Emotion-aware cross-modal domain adaptation in video sequences

EMOTION-AWARE CROSS-MODAL DOMAIN ADAPTATION IN VIDEO SEQUENCES

EMOTION-AWARE CROSS-MODAL DOMAIN ADAPTATION IN VIDEO SEQUENCES

Dissertation

to obtain the degree of Doctor at the Maastricht University,
on the authority of the Rector Magnificus Prof. dr. Pamela Habibovic
in accordance with the decision of the Board of Deans,
to be defended in public on
Thursday, October 20, 2022, at 10:00 hours

by

Christos ATHANASIADIS

Faculty of Science and Engineering
Department of Advanced Computing Sciences

Promotor: Dr. Stylianos (Stelios) Asteriadis
Promotor: Prof. Dr. Gerhard Weiss
Co-promotor: Dr. Enrique Hortal

Assessment Committee:

Prof. dr. Nava Tintarev (chair),	Maastricht University
Prof. dr. Andre Dekker,	Maastricht University
Prof. dr. Miguel Cazorla,	University of Alicante
Prof. dr. Federico Alvarez,	Universidad Politécnica de Madrid
Dr. Jerry Spanakis,	Maastricht University

This research was supported by the Horizon 2020 funded project MaTHiSiS (Managing Affective-learning THrough Intelligent atoms and Smart InteractionS) (http://www.mathisis-project.eu/), under Grant Agreement nr. 687772.

Dissertation Series No. 2022-27 The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Keywords: Affective Computing, Machine Learning, Audio-Visual Emotion Recognition, Shallow and Deep Metric Learning, Attention Mechanisms

Printed by: ProefschriftMaken, de Bilt

Front & Back: Designed by Gizem Ustuner, Amsterdam, the Netherlands.

An electronic version of this dissertation is available at
`http://repository.maastrichtuniversity.nl`.

To my beloved family.

CONTENTS

LIST OF FIGURES

List of Tables

ACKNOWLEDGEMENTS

First of all, I would like to deeply thank my supervisor Dr. Stylianos Asteriadis for giving me complete freedom in pursuing my research interests while also providing guidance whenever I needed it, for sharing his knowledge with me, and for reading my paper drafts. I am very grateful to Dr. Enrique Hortal (my co-supervisor) as well for creating a great working atmosphere and giving helpful insights throughout my Ph.D. period. Enrique thanks for being patient and always keen to help. Moreover, I would like also to thank Prof. Gerhard Weiss my secondary supervisor, for his support and valuable comments during all this period.

Of course, in this part of the dissertation I could not omit Esam and Dario. Starting our Ph.D. on the same day and working together on the same European project, Esam and I faced similar challenges and stressful times. He helped me significantly during my research but helped me also to maintain and balance my mental health as well. Dario was already in the office when I started and he provided a significant help to me to understand how to navigate academia and how to be a “pro”. Mirella and Gerasimos were always available to share their valuable insights in research but in life as well. Furthermore, I would like to thank my girlfriend Gizem for her unconditional emotional support and encouragement during all this struggle. She was always there to console me.

Lastly and more importantly, I would like to thank my family for their support and love. I would like to thank my parents Silvia Provata and Giorgos Athanasiadis, and my brothers Efstathios and Grigorios who have given me a lifetime of love and care.

This work was supported by the Horizon 2020 funded project MaTHiSiS (Managing Affective-learning THrough Intelligent atoms and Smart InteractionS) nr. 687772 (http://www.mathisis-project.eu/). Furthermore, we want to thank all students that participated in our experiments.

SUMMARY

In his "Treatise of Human Nature", David Hume argued that: " Reason is, and ought only to be the slave of the passions!!".

Emotion detection and recognition play a fundamental role in humans, especially in our everyday social engagement, when we are experiencing social behaviors (for example, in relationship maintenance). Moreover, they can prove really useful when addressing people in public (for instance, during a political debate or discussion). Not only do emotions play an important role in everyday life and social interactions of each person, but also they are crucial in human perception and cognition. Therefore, developing an automatic interface to rationalize about human emotions could prove to be very helpful, especially to those people that lack the ability to properly recognizing other people's emotions.

At the same time, developing intelligent systems that are capable of recognizing emotions is an important, but a rather challenging task. Humans experience a wide range of emotions when interacting with an interface. Previous research has provided firm evidence that emotions vigorously affect motivation and hence play a crucial role when users interact with interfaces (such as learning or video games, for instance). However, it was not until recently that modern technology started to embody users' affect analysis and not only use their conventional interaction methodology with systems (such as button clicks, feedback, etc.).

In this dissertation, a study of emotion recognition by employing sensorial modalities (such as auditory signals) with an insufficient amount of data is performed. Domain adaptation approaches are applied to leverage emotion-related information with great availability (such as facial expressions) and transfer this knowledge to emotion cues with less available information to increase their emotion recognition performance. In particular, we shed light on the audio-visual cross-modal relationship. Having a system performing emotion recognition from modalities such as audio (with limited access in audio features), the task is to leverage availability in face datasets (where there are more available data) with the purpose of enhancing the classification efficiency.

The main pillars of this research concern the implementation of metric learning and generative models to facilitate the task of cross-modal domain adaptation between audio and face within the affect context. Furthermore, an extensive study in the ways that temporal dynamics between audio and face modalities could contribute to domain adaptation is employed. Finally, we facilitate also the inverse task that is to perform face-based emotion recognition by eliciting auditory source information.

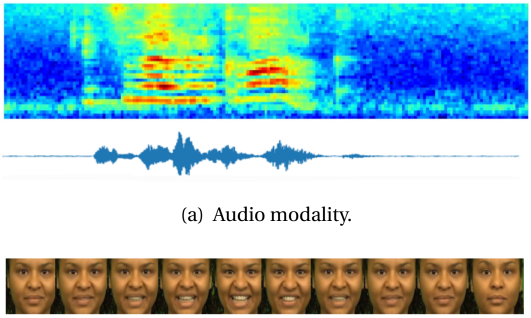
1

INTRODUCTION TO EMOTION-AWARE DOMAIN ADAPTATION

In the current thesis, research in domain adaptation for cross-modal emotion recognition is performed. The applied modalities represent input information such as facial expressions, audio utterances, body posture, textual, and health-related signals. Emotion-related information can be extracted by humans or automatically from machines when examining these channels. In this dissertation, our primary focus is to investigate audio-visual cross-modal relationships within emotion context and explore multiple ways of mapping and projecting face-related to audio-related features and vice versa. To measure the performance of our task, we tried to leverage source data from one modality with the purpose of improving the emotion recognition efficacy in the second modality. In Figure 1.1, a visualized instance of the modalities under study is presented.

The core inspiration for the dissertation originated from academic studies on cross-modal relationships within the cognitive psychology field and the analysis regarding the way that audio and video are connected in the human brain. Furthermore, we were also inspired by literature findings in [1] that show that the emotion recognition performance among different modalities varies significantly. There are dominant modalities such as facial cues, for example, while there are modalities such as auditory signals where the human, as well as the computer capacity to recognize emotions, is inferior, according to conducted research in cognitive psychology [1].

Scientists in the domain of cognitive psychology have long studied the relationship between facial and vocal cues in humans [1] [2]. In particular, researchers suggested that infants, during the development of their auditory and visual perception, fuse facial cues together with audio information to better discriminate and recognize emotions. Along the same line, authors in [3] cited an extremely interesting phenomenon (called "The McGurk Effect") regarding how human visual perception functions to calibrate the way that the brain perceives auditory signals. This study states that there is a difference



(a) Audio modality.

(b) Face modality.

Figure 1.1: Example of the under study modalities.

in the way that the auditory input signal is interpreted when the subject has open or closed eyes. For example, the sentence*My dad pop me poo brive* is interpreted by the human brain as *My dad taught me to drive* when the subject has open eyes. Authors suggest that both audio and visual perceptual systems function together and calibrate these mistakes.

In the same spirit, authors in [4] performed a study on how people with hearing problems perceive the sounds of phonemes by triggering their visual perceptual system with the purpose of performing lipreading (or speech-reading). In a similar manner, concerning emotional cross-modal relationships, prosodic speech information (linguistic variation in speech like pitch tempo, loudness, etc.) and its correlation with facial features have been intensively studied [5] [6] [7]. The cardinal outcome of these works was that speech prosodic information is associated with other social cues such as facial expression, body language or tone tempo. In particular, authors in [7] suggested that speech prosodic information could be extracted merely using facial cues. Hence, a worthwhile research question that is inspired by the theoretical conducted research is whether the connection between the audio-visual information could be examined from the emotional point of view. In this regard, we re-frame the question as: *Does the emotional state content of a persons' voice correlate with their facial expression?*

In the light of the aforementioned cognitive research, we intend to transform the same questions posed by cognitive psychologists into the domain of emotion recognition and Domain Adaptation (DA) from the computer science perspective and pose the following question: *Is it possible to transfer knowledge between facial expressions and audio-features (which are derived from the same videos and are governed by the same expressions)?* The importance of this question lies in the fact that, while emotion recognition through facial expressions has been studied extensively [8] [9] [10], emotion recognition through other modalities such as audio has produced fewer advancements concerning classification results [11]. An important reason behind this is the fact that there are not a lot of publicly available datasets for audio-based emotion recognition compared to the abundance of data that exists regarding facial expressivity. Therefore, generating training models for emotion recognition through this modality can be a rather

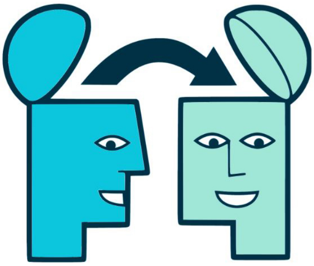


Figure 1.2: Transfer of knowledge between humans.

challenging task and requires the generation of new robust datasets. Meanwhile, the engineering of such big and complex corpora is not always a straightforward and feasible task.

To overcome these limitations, domain adaptation (defined also as transfer learning) algorithms are fostered by researchers with the objective of developing classification methods for specific modalities by exploiting data from other similar ones that come from rich available datasets [12]. Therefore, by eliminating the source and target domain distributions, inherent differences, we can transfer knowledge across modalities with different distributions by projecting both into a new latent domain [13], or by learning a transformation that projects the one domain onto the other. The essential objective is to efficiently perform classification tasks by leveraging data from the modalities with rich available datasets and transferring that knowledge into different modalities with different distributions (with sparsely available datasets), performing the same in-hand task [12][13]. These techniques were inspired by human behavior and the way that the learning process is materialized in the human brain by “re-using” previous knowledge to handle new situations or the way in which knowledge is transferred through socializing and learning from one human to another (Figure 1.2).

In the current dissertation, two domain adaptation strategies were employed which are **metric learning based techniques** and **domain adaptation based on generative models**. Both strategies are introduced in subsections 1.3 and 1.4 and further analyzed in Chapters 1, 2 and 3.

The remainder of this chapter is structured as follows. First, Section 1.1 gives an introduction to domain adaptation as well as the terminology adopted through the whole dissertation. Section 1.2 introduces emotion recognition, as well as state-of-the-art techniques for the modalities that we will focus on in the current thesis. Subsequently, in Sections 1.3 and 1.4, the techniques applied in domain adaptation are discussed, as well as state-of-the-art techniques that inspired our work. In section 1.5, the datasets utilized in all the techniques that are established throughout the dissertation are described. Next, in Section 1.6, the problem statement of this dissertation is analyzed and the five research questions of this thesis are formulated. Finally, Section 1.7 gives an overview of the whole dissertation.

1.1. DOMAIN ADAPTATION

During the last years, very significant progress has been made regarding supervised and unsupervised learning techniques that led to astonishing advances in the field of machine learning. Nowadays, we experience AI innovations that include autonomous vehicles, intelligent agents and healthcare detection and prediction systems with human-level or even super-human efficiency.

Yet, the great majority of these sophisticated learning models are in need of data sources and their performance depends heavily on the quality and the amount of data available for training. However, in practise, these large amounts of labeled data are not always available or are costly to obtain, which in turn leads to poor classification performances in the learning models. On top of this, when training on a specific type of data, the extracted model very often lacks the ability to generalize to any new situation beyond these experiences that are described by training data distributions, and therefore, there is a limitation of what they can achieve.

Inspired by all previously mentioned examples and in general by the way that the human brain is capable of performing knowledge transfer, researchers in the Artificial Intelligence community have shifted their attention to the area of domain adaptation to overcome these issues. In comparison with the traditional machine learning approaches where the learning process occurs in distinctively, without taking into account knowledge from any other domain, domain adaptation uses knowledge from other existing domains during the learning process.

The field of domain adaptation has been inspired by human behavior and the way that the learning process occurs. We humans find it easy to transfer gained knowledge from one domain or task to another. An analogous paradigm for the performed “domain adaptation” in humans is the following: Let us consider the case of two people who want to learn the programming language of Python. One person has no previous experience in programming languages, while the second one has knowledge of Java and C++ and is already familiar with concepts like loops, recursion, objects, inheritance, abstraction, functions, etc. It will be much easier for the second person to study and learn Python by using and transferring the knowledge that has been already gained from Java and C++. Furthermore, it will be much easier for the first person to learn Python by getting help from the Java and C++ expert.

A more concrete example deriving from the machine learning domain (which is closely related to the work conducted in the current dissertation) is as follows: Let us assume the task of guessing emotions using sound clips of people speaking. If there exists an abundance of available labeled data for that task, traditional machine learning algorithms can be implemented to create a robust emotion recognition system through those sound clips. However, assume the case of sparse datasets available from the audio modality and dense datasets available from the face modality. In both cases, the data are linked since they are referring to the same emotions and, thus, to the same classification task of emotion recognition. A strategy that could elicit data from closely related domains with audio like face modality (for instance, in the case of audio-visual video symbiosis) to improve the emotion recognition performance should be applied. This methodology is coined with the term “transfer learning” or “domain adaptation”. An alternative way to conceptualize the two different domains of the example (face and au-

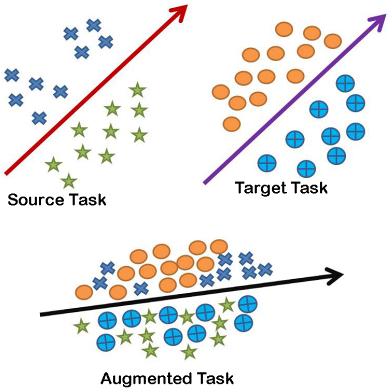


Figure 1.3: In the top left figure, we can see a binary classification task for the source domain (source task). In the top right figure, we can a binary classification task in the target domain (target task). Finally, in the bottom figure, we can see a binary classifier that binds the two domains into a common augmented classification task.

dio) is that both domains can be represented in a higher common sub-space where the same classifier can be employed for both source and target tasks. Figure 1.3 visualizes this concept. In particular, it is displayed how a binary classifier for source and target domains (with a common task, for example, emotion recognition from audio and face) can be fused together into a common augmented domain and perform classification for both domains simultaneously.

1.1.1. DOMAIN ADAPTATION DEFINITIONS

For readability purposes, this sub-section introduces a domain adaptation terminology dictionary which will help readers understand the concepts and methodologies used in the remainder of this thesis. Primarily, the term domain $\mathscr{D}$ is defined by using the following two notions. Firstly, the feature space $\mathscr{X}$, which contains all possible instances while $X = \{x_1, x_2, ..., x_n\} \in \mathscr{X} \subset D$ is a subset of the feature space and contains the in-hand available learning sample vectors. Secondly, the feature probability distribution of the learning sample features is defined as $P(X)$. Then, the classification task denoted as T, is introduced. Two new terms should be defined in that case. Firstly, the label space $\mathscr{Y}$ and, secondly, the classification function $f(X)$, which can be used to map a new unknown input feature vector $x^{'} \in \mathscr{X}$ to the label space $\mathscr{Y}$. In particular, if the machine learning problem under study is the classification task of emotion recognition through facial expressions using the six basic emotional states (happiness, sadness, fear, disgust, anger and neutral) [66], then the feature space $\mathscr{X}$ is represented by all possible values (e.g. between 0-255) that the pixels of the images from the facial domain can take, while the labels $y_i \in \mathscr{Y}$ are represented by the aforementioned six basic emotions. It should be also stressed that, for two different classification tasks, the feature domains and the feature probability distribution can be vastly different.

In addition, the terms source and target domain are introduced [12]. The source domain is considered a space which contains data that will be used to perform the transfer of knowledge. In the current work, the facial expression modality is defined as the source domain. Formally, the source domain can be defined as $D_S = (X_S, Y_S, P(X_S))$ (with $X_S \subset \mathscr{X}_{\mathscr{S}}$, the feature set which is subset of the feature space that represents the source data). On the other hand, the target domain is the sub-domain that needs to be enhanced through transferred knowledge stemming from the source domain. This domain is defined as: $D_T = (X_T, Y_T, P(X_T))$ (with $X_T \subset \mathscr{X}_{\mathscr{T}}$, the feature set that is subset of the feature space that represents the target data). In the same spirit, a definition for source and target classification tasks can be defined as follows: Insofar as the first is concerned, the source task T_S is the classification task that can be trained using the data from the source domain D_S. Similarly, the classification task T_T is the one that can be applied using the target domain data D_T. The classification task for each domain consists in calculating the predictive classification function for each case: $f_S(X_S)$ and $f_T(X_T)$. This is done by incorporating the feature vectors into the training set and learning the relation between the feature vectors and the corresponding labels. By definition, the scope of domain adaptation is to extract the knowledge from the source task and to apply this knowledge to the target task. This transfer of knowledge is implemented with the purpose of improving the performance of the classification task in the target domain by incorporating knowledge from the source domain, thus improving the performance of the predictive classification function $f_T(X_T)$.

Having defined basic terms associated with Domain Adaptation (DA), several emerging scenarios arise regarding the nature of the available data and the way that DA can be utilized [12][13]. For the source and target domains $D_S = (X_S, Y_S, P(X_S)$ and $D_T = (X_T, Y_T, P(X_T))$, the emerging cases for DA are very often subject to the following conditions: $\mathscr{X}_{\mathscr{S}} \neq \mathscr{X}_{\mathscr{T}}$ and $P(X_S) \neq P(X_S)$. In the case that the source and target domains are not the same ($\mathscr{X}_{\mathscr{S}} \neq \mathscr{X}_{\mathscr{T}}$), the approaches are defined as *Heterogeneous Domain Adaptation*. While if $\mathscr{X}_{\mathscr{S}} = \mathscr{X}_{\mathscr{T}}$ and $P(X_S) \neq P(X_S)$ the approaches are defined as *Homogeneous Domain Adaptation*. These categories are illustrated in Figure 1.4. As an example of *Homogeneous Domain Adaptation* can be mentioned the knowledge transfer between two different datasets of images that contain for instance people's faces, however, with different subjects, illumination conditions and poses. On the other spectrum, in the case of *Heterogeneous Domain Adaptation*, as an example, we can have two datasets with completely different types of information like images and audio.

To perform knowledge transfer, for both Homogeneous and for Heterogeneous Domain Adaptation, there are two main strategies: symmetric and asymmetric approaches. Insofar as the first is concerned, both the source and target domains are transformed into a common latent subspace, while in the latter case, only the source domain is transformed into the target domain. In the current work, we employ both strategies since in the case of metric learning we performed symmetric domain adaptation while in the case of generative networks we performed asymmetric domain adaptation.

When trying to shed light on the literature on domain adaptation, it is becoming evident that the terminology and definitions are often inconsistent [16]. A prime example of this discrepancy can be the use of the terms domain adaptation and transfer learning, commonly but not always used to refer to the same concept [13]. Another common inconsistency is how domain adaptation problems are grouped. Traditionally, do-

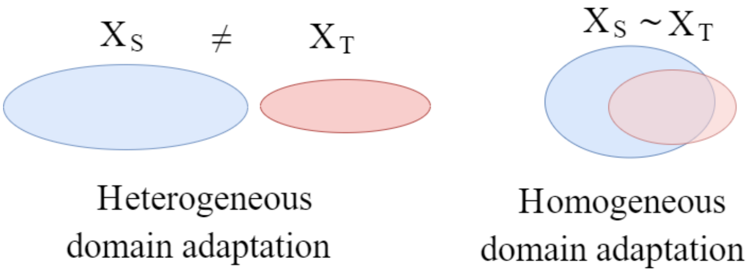

$X_S \neq X_T$
Heterogeneous domain adaptation
$X_S \sim X_T$
Homogeneous domain adaptation

Figure 1.4: Homogeneous and heterogeneous domain adaptation.

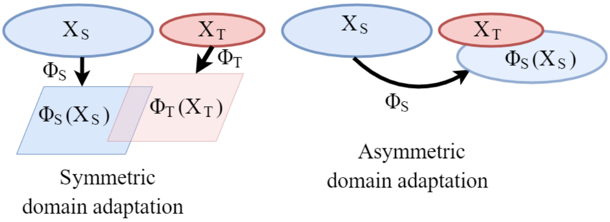


Figure 1.5: Symmetric and asymmetric domain adaptation.

main adaptation problems are categorized based on the similarity between domains, from whether deep learning is implemented or not and also based on the availability of labeled and unlabeled data [16]. Another categorization can lead to three main groups: “Inductive”, “transductive”, and “unsupervised domain adaptation” (the review work done in [12] analyzed these categories in detail). More specifically:

- As far as the first is concerned, both source and target domains contain dense datasets with fully available label information. Hence, the cross-domain projection is taking into account the tuples of annotated samples from all the involved domains [12].

- In the second case, there exists some small amount of label information or there is no label information regarding the target domain but there is an auxiliary way to infer them (for instance with the usage of a classifier). This category can be noted alternatively as “unsupervised domain adaptation” [12].

- Finally, in the last case (“unsupervised domain adaptation”), there is a lack of information concerning labels in the target domain and any auxiliary classifier. DA is performed by explicitly using the distributions of the different modalities without any annotation information [12].

The employed techniques of this dissertation lie in the first two cases.

Another important term related to domain adaptation is the so-called *negative knowledge transfer* [16]. This term expresses the case where the applied domain adaptation contributes negatively to the performance of the target task. For instance, by leveraging information from the source domain (for instance face modality) and trying to transfer it to the target domain (audio modality) for applying the target task classification (emotion recognition), the classification performance is deteriorating.

1.1.2. RECENT ADVANCES IN DOMAIN ADAPTATION

In this subsection, the State-Of-The-Art (SOTA) approaches regarding domain adaptation are introduced. The related overview papers discussed in [12][13] are considered among the most popular surveys in the fields of transfer learning and domain adaptation. These works introduced a widely accepted terminology and some comparison between the performance of the approached involved. Moreover, the works in [15][16] are recent surveys that introduce SOTA approaches for visual domain adaptation in particular. In the following subsections, we describe in detail the approaches that provided fertile inspiration for the current work. Most of these approaches can be categorized into two groups depending on whether they are deep-learning-based or not. For clarity purposes, we will define the second category as *classical machine learning for domain adaptation*. In this case, the feature extraction process from all the involved domains is occurring in an extra step using a dedicated feature extraction algorithm. Therefore, this approach can be coined also as *classic domain adaptation* approach. In contrast, the input for the deep learning approaches are raw data and the feature extraction is occurring in an *end-to-end* fashion.

DEEP LEARNING DOMAIN ADAPTATION APPROACHES

Deep-learning-based approaches can be grouped into the following categories: fine-tuning deep architectures, adversarial approaches, data reconstruction based methods, and discrepancy based methods [15][16].

Fine-tuning deep architectures. One of the most common deep learning domain adaptation approaches, is to fine-tune the deep network model using data from source domain D_S and then update several added last layers (or one last layer) using data only from the target domain D_T. Fine-tuning requires normally, a large amount of data from D_S and the limited amount of data from the target domain D_T. Hence, the model is in general augmented with, when available, the few labeled target instances - which allows adjusting the deep model to the new domain. In this case, we can also work even if the targeted classes do not belong to the classes used to pre-train the deep model. However, if the domain divergence between the D_S and D_T is large, the fine-tuning process could lead to over-fitting the deep model. In this case, the performance of the fine-tuned model can lead to negative transfer learning.

A prominent example of this category is the work done in [31], where an analysis in the so-called *parameter-based domain adaptation* field is performed. In that case, the scope of domain adaptation is to transfer the knowledge of the calculated parameters between the source and target task. More specifically, the core idea of this work is to train Convolutional Neural Networks (CNNs) in two steps. Firstly, the intermediate layer and their parameters of the CNNs algorithm are tuned using the source domain dataset. Secondly, the last layers of the CNN architecture are tuned by using solely the information from the target domain.

Adversarial approaches. The target of these techniques is to provoke domain confusion by utilizing an adversarial objective with respect to a domain “discriminator”. These approaches are normally implemented by using two-stream approaches. In this sense, one neural network will try to perform the domain confusion, while a second neural network called the “discriminator” will be judging whether the domain confusion is efficient or not. These approaches can be further divided in two categories depending on whether they encompass a generative strategy (as in the case of GANs) or not. In this case, the target is not only to minimize the source and target divergence but also to generate samples from the target domain.

For example, authors in the work titled “Domain-Adversarial Training of Neural Networks” (DANN) [17] facilitated domain adaptation from the learning representation perspective. They attempted to jointly learn representations for both source and target domain samples by introducing a neural network that is having as loss function a domain divergence loss ($\mathscr{H}$-divergence) that calculates the distance between the two domains. This loss is coupled together with classification loss that uses the supervised information exclusively from the source domain. Inspired by this work, authors in [18] proposed the so-called Variational Recurrent Adversarial Deep Domain Adaptation (VRADA) model, which employs variational recurrent adversarial networks. This is done in order to capture and transfer temporal latent dependencies across domains using domain-invariant representations for real-world health-care time-series data. On this ground, they tested their approach using the MIMIC-III [19] and a Pediatric ICU (PIC) [20] dataset.

Authors in [33] proposed a covariant multimodal attention method based on multimodal domain adaptation neural networks (MDANN). In this work, the authors investigate whether it is possible to perform domain adaptation that can transfer knowledge from one multimodal dataset to another one. This is done by trying to learn a common feature representation for multiple modalities and mitigate inter-domain divergence by applying jointly adversarial loss among the different modalities.

The work done in [32] facilitates the implicit discourse classification problem in a principled adversarial manner. The term implicit discourse in the linguistic domain connotes sequential sentences that are connected semantically but without the use of any explicit grammatical connectivity. A deep learning architecture is established which is composed of a network called implicit relation (i-CNN) that extracts embeddings related to implicit input and a network called connective-augmented relation (a-CNN) that extracts embeddings for the same implicit input enhanced with explicit connectors. Furthermore, an adversarial network (discriminator) judges whether the inputs come from i-CNN or a-CNN. Finally, on top of the previous networks, a CNN network performed the discourse classification.

Discrepancy based methods. These methods employ a discrepancy loss which is normally applied for measuring the distance between two distributions [16] (for instance Maximum-Mean-Discrepancy MMD [138]). A prominent example of this category is the Siamese architecture [21] where the loss is computed between the corresponding activation layers of two neural networks (streams). The first stream-network represents the source domain while the second one is the target domain. Both are unified together in a common loss which is calculated based on their domain discrepancies. These types of models are analyzed further in Section 1.3.

In the same spirit, the authors in [22] proposed the Deep Adaptation Network (DAN) that regards the sum of MMDs loss defined between several layers for two stream networks (similarly to work done in [21]). Furthermore, the authors investigated several kernels for adapting the extracted deep representations from the two streams, which substantially enhances adaptation effectiveness compared to a single kernel method used in [23]. This is even improved by the study called Joint Adaptation Networks [24], which instead of the sum MMDs defined between different layers use the joint distribution discrepancies of these representations. In the same spirit, CORAL methodology [26] aims at minimizing domain shift by aligning the second-order statistics of source and target distributions (covariance). The employed loss which is expressed by the distance between the covariance matrices of the source and target domain distributions (C_S and C_T) is used to minimize the discrepancy between the domains. These covariance matrices express the variation of the multi-dimensional distributions for each domain. The CORAL loss can be summarised with the following Equation:

$$\lambda_{CORAL}||C_S - C_T||^2 \tag{1.1}$$

The Deep CORAL [25] is an extension of the CORAL using deep learning. In particular, the core idea is to calculate a nonlinear transformation that aligns correlations of activation layers between the two streams networks (likewise to Siamese networks). This idea is similar to DAN approach with the exception that instead of MMD loss, the CORAL loss is applied.

Authors in [27] introduced a technique called Hybrid Heterogeneous Transfer Learning (HHTL) through deep learning to perform heterogeneous transfer learning between multilingual documents. The target is to perform knowledge transfer and to create a text sentiment classifier for documents in a specific language by using only an annotated set of documents in another language. To tackle that issue, the authors implemented a technique called Marginalized Stacked Denoised Autoencoders (mSDA) [28] to transform both source and target domains into common representations. *Denoised autoencoders* are one-layer neural networks trained to reconstruct input data from partial and random corruption. These “autoencoders” can be “stacked” into a big deep learning architecture (therefore called “stacked”). The term marginalised in the name of the approach refers to the adopted greedy layer-by-layer training process. Finally, a deep learning architecture is applied in order to bridge these different common representations of the source and target domains by making use of the “document-correspondences” information that links the source and the target domain (which are generated using “Google Translate”).

Data reconstruction (encoder-decoder) based methods. These approaches, in contrast with the previous ones, combine a convolutional neural network that normally performs predictions for the source domain X_S (denoted as “encoder”) with a second deconvolutional neural network for target data reconstruction (“decoder”). In this case, encoder extracts a latent space representation of the X_S while decoder network reconstructs the target domain X_T. The loss of the whole process can be still the normal prediction or classification loss (cross-entropy for instance) bound together with a reconstruction loss.

The Domain Separation Networks (DSN) topology [34] introduces the notion of a private subspace for the source X_S and target domain X_T and we can capture domain specific properties, such as background and low level image statistics (for a dataset consists of images). Then, a shared feature space between the domains, enforced by using autoencoders and explicit loss functions, extracts the shared features between the domains. The model involves a reconstruction loss using a shared decoder, that aims at reconstructing the input sample by using both private (domain specific) and source representations. The evaluation of the proposed approach is performed in several domain adaptation benchmarks for image-to-image domain adaptation.

All methods in this category are in need of a big data corpus to properly tune the employed deep neural networks that are consisted of [47]. Moreover, these approaches comes with a high computational cost and therefore are in need of more hardware resources than the *classic machine learning approaches*. However, state-of-the-art results are observed also in several cases using these approaches [16] in domains where these big corpora are available. Unfortunately, this is not always the case and there are several domains in which it is not possible to apply these techniques. However, in the field of audio-visual domain adaptation, which is the focus of the current dissertation, there are various big corpora that can be leveraged to apply deep learning. Therefore, these techniques can be employed to study the relations between these two modalities.

Classical machine learning for domain adaptation

In the following paragraphs, the classical DA methods are reviewed. These approaches can be categorized into the following groups: graph-based methods, parameter adaptation methods, feature augmentation, feature space alignment, and metric learning based feature transformation.

Graph based methods. In this category, X_S and X_T are treated as graphs, and similarities between the instances of both domains are calculated. These instances are representing the nodes of these graphs. The variations between the several methodologies are lying in the way that the similarities between the graphs are computed.

Authors in [35] tackled the homogeneous domain adaptation task in an unsupervised manner by trying to spot correspondences between samples in the source and target domains. The correspondences are obtained by treating the source and target domains as graphs and using a convex criterion to match them. The criteria which are used are first and second-order similarities between the graphs as well as a class-based regularization. Experiments are performed on several image classification datasets as well as on toy datasets. Similarly, as an extension of the same work, authors in [36] accommodated the same task by considering also higher-order similarities. In the work presented in [37], titled *Optimal Transport for domain adaptation*, authors accommodated the unsupervised and semi-supervised domain adaptation problem as in the graph-matching problem described in [35], by trying to bring close to each other the source and the target modalities, by using the Monge-Kantorovich (alternatively called as Wasserstein) distance that is coupled together with several regularizers and by using a generalized conditional gradient (GCG).

In [38], a hybrid version of deep neural networks and graph matching approaches is developed. Neural networks are employed to extract domain invariant representations that used a graph matching loss as the domain discrepancy metric.

Authors in [41] proposed a symmetric approach for domain adaptation called Heterogeneous Spectral Mapping (HeMAP). This approach handles the cases where the target and a source domain are not linked with a common class-label set. This approach employs a spectral mapping algorithm to learn transformation projections. For each domain, a transformation matrix B_S is introduced for the source and B_T for the target domain. The inferred objective function consists of two components: the first one tests whether the structure of the target and source data is retained after the transformation using matrices B_S and B_T while the second one calculates the difference between the target and source data in the projected space. By minimizing the objective functions, the algorithm learns target and source transformation mappings which retain their data structure and at the same time minimizes their differences in the common sub-space. In the end, a dedicated HeMAP algorithm classification approach is employed for the target classification task.

Feature augmentation. One of the simplest approaches to tackle domain adaptation which called *frustratingly easy domain adaptation* is proposed in [29]. The original representation of source domain $\mathbf{x}_S \subset X_S$ is fused with itself and a vector of the same size is filled with zeros (zero padding). Each feature from each domain, for instance for source domain, (with dimension $\mathbf{x}_S \in \mathbb{R}^D$) is mapped onto an augmented space of dimension $\mathbf{x}_S^{'} \in \mathbb{R}^{3D}$ simply by duplicating the feature vectors. Thus, the features from the source domain are mapped into $[\mathbf{x}_S, \mathbf{x}_S, \mathbf{0}]^T$ while the features from the target domain are represented by $[\mathbf{x}_T, \mathbf{0}, \mathbf{x}_T]^T$ with $\mathbf{x}_T \in \mathbb{R}^D$. Finally, an SVM classifier is trained using a fused dataset which contains both augmented domains. While this approach is seemingly really straight-forward and not a particularly sophisticated one, surprisingly it is proven to work in many cases.

From a similar perspective, the authors in [39] proposed an algorithm that is titled Heterogeneous Feature Augmentation (HFA) to handle the case of knowledge transfer for domains of heterogeneous nature. In their approach, features which belonged to two different domains (either images or text with different representations and distributions) are represented by features with different dimensionalities. Thus, the very first thing that the approach needed to cater for is the transformation of the source and target domain features to the same dimensionality. To perform so, authors propose two transformation matrices (P and Q) and by fusing the transformed data with the initial features and vectors of zeros. This can be seen in Equation 1.2:

$$\phi_S(\mathbf{x}_S) = \begin{bmatrix} P\mathbf{x}_S \\ \mathbf{x}_S \\ \mathbf{0} \end{bmatrix} \text{ and } \phi_S(\mathbf{x}^T) = \begin{bmatrix} P\mathbf{x}_T \\ \mathbf{x}_T \\ \mathbf{0} \end{bmatrix} \tag{1.2}$$

The ϕ transformation represents this new sub-space for both the source and target domains. P and Q are learned by using Support Vector Machines (SVM) [40] (and are part of the Karush–Kuhn–Tucker constraints [40]). With the purpose of simplifying the optimization equation in HFA, the authors propose, instead of calculating both transformation matrices P and Q, to calculate an intermediate metric $H = [P, Q]^T [P, Q]$. In this way, it is more trivial to optimize and solve for H instead for both variables P and Q, and thus, we have less constraints in the optimization scheme. Moreover, the authors proved that it is not necessary to calculate P and Q separately. When learning the intermediate metric H the system is able to transform both source and target domains into the same subspace and perform traditional classification. This transformation matrix H can be applied directly in a new test sample.

Feature space alignment. In this category, the task is to perform a transformation and align features from both domains.

Authors in [44] proposed a heterogeneous domain adaptation technique based on manifold alignment. The core idea of this approach is to construct mappings to bridge different feature spaces in order to perform knowledge transfer across different domains. This approach can reuse supervised information from multiple source domains in a target domain even when the input source domains do not share any common features or instances. The key idea is to transform different source domains into a sub-latent space using manifold alignment by making use of the available supervised input information. The method minimizes the divergence between the domains by using second-order statistics of the source and target domain distributions ($P(X_S)$ and $P(X_T)$). The main idea is a whitening of the source data using its covariance followed by a “re-coloring” by using the target covariance matrix.

Metric learning approaches for domain adaptation. These approaches require at least a limited number of labeled data in the target domain and they implement a metric learning technique to bridge the source and target domains. The core part of these approaches is based on the implementation of metric techniques for bridging the source D_S with the target D_T domains.

Authors in [42] proposed a method called Asymmetric Regularized Cross-Domain Transformation (ARC-t). This approach considers the case that both the source and the target domain (different visual databases) have the same class-label set and the target objective is to perform object recognition. The purpose of this approach is to calculate a transformation matrix W that maps the source domain to the target domain. This matrix W is learned by the implementation of a non-linear RBF kernel. A minimization task is introduced which aims to minimize matrix W by making use of constraints imposed by pairs of source and target features that share common information (derived from the same class). This is achieved after considering this matrix W as a similarity function and therefore metric learning approaches are employed.

Authors in [43] proposed a method called Sparse Heterogeneous Feature Representation (SFHR), which is an asymmetric heterogeneous domain adaptation technique, assuming that both the source and the target domains have the same class-label set. This approach is relevant to ARC-t [42]. However, the difference is that the source and target datasets are first represented in a code space based on some class decomposition schema and then the transformation matrix W is learned. In SFHR, this is done through the implementation of non-negative “Lasso optimization”.

In this spirit, the metric learning for Domain Specific Class Means (DSCM) [45] is introduced. This approach is an extension to Nearest Class Means (NCM) [134] which is an approach dedicated to a specific domain for calculating a new distance and improving the image classification task. Given an image $\mathbf{x}_i$, this approach similarly to k-means assigns this image to the class $c^* \in y_c = \{1, 2, ..., c\}$ whose mean is the closest:

$$c^* = \underset{c \in Y_c}{\arg\min}\, d_{\mathbf{w}}(\mathbf{x}_i, \boldsymbol{\mu}_c), \text{ with } \boldsymbol{\mu}_c = \frac{1}{c} \sum_{i:y_i=c} \mathbf{x}_i \tag{1.3}$$

where $d_{\mathbf{w}}(\mathbf{x}_i, \boldsymbol{\mu}_c) = ||\mathbf{W}(\mathbf{x}_i - \boldsymbol{\mu}_c)||^2$ is the squared distance between an instance $\mathbf{x}_i$ and the class mean $\boldsymbol{\mu}_c$ in a projected feature space given by the transformation matrix $\mathbf{W}$. If

$\mathbf{W} = \mathbb{I}$ where $\mathbb{I}$ is the identity matrix, the above equation corresponds to the Euclidean distance in the original feature space. This definition can be interpreted as the posterior probability of a sample $\mathbf{x}$ to belong to class c. Authors have shown that the equation 1.3 can be analyzed as a softmax function:

$$p(c|\mathbf{x}_i) = \frac{exp(-\frac{1}{2}d_{\mathbf{W}}(\mathbf{x}_i, \boldsymbol{\mu}_c))}{\sum_{c=1}^{N_c} exp(-\frac{1}{2}d_{\mathbf{W}}(\mathbf{x}_i, \boldsymbol{\mu}_c))} \tag{1.4}$$

Finally, the class assignment occurs using the $c^* = \arg\max_{c \in Y_c} p(c|\mathbf{x}_i)$. More details about this can be found in [45]. DSCM, is an extension of NCM for domain adaptation. It calculates the means of each class by taking into account the data of both source and target domains. It realizes a transformation of the feature space which minimizes the weighted softmax distances (which likewise NCM represents the posterior probability of a sample $\mathbf{x}$ to belong to class c) from the corresponding domain-specific class means (average instances) using data from both domains. In this way, the approach manages to decrease the intraclass and to increase the interclass distances.

The work described above has been extended with the use of an active learning component by the Self-adaptive Metric Learning Domain Adaptation (SaML-DA) [45] framework. In this method, the training set for the target domain is iteratively increased with the labels extracted by using with DSCM and used to fine-tune the under study metric.

SaML-DA is motivated by the Naive Bayes Nearest Neighbor based Domain Adaptation (NBNN-DA) [46] approach that blends a distance metric learning technique with a Naive Bayes Nearest Neighbor classifier to model the sample-to-class distances by slightly making the metric more suitable for the target domain. The core idea for SaMLDA and NBNN-DA methods, is to substitute at each iteration the source sample which the classifier is less confident for each class by the target sample, where the classifier (DSCM respectively NNBA) is the most confident for the given class.

Distance metric learning approaches are analyzed further in Section 1.3.

Parameter based approaches. Another set of early DA methods investigates a different strategy for adapting a classifier trained using datasets from the source domain D_S (for instance an SVM classifier) in order to perform better on the D_T domain. It is important to note that, at least some small annotated information is requited from both domains in these approaches, hence they can only be applied in the semi-supervised DA scenario. These approaches are trying to take into account the training parameters extracted from the source domain task and apply them to the target domain task.

A prominent work in this category is the adaptive SVM (A-SVM) [48]. This approach aims at adjusting the decision boundaries optimized for the source domain classifiers with the help of a set of delta functions that are added in the primal optimization of the SVM by exploiting predictions on the available labeled target examples.

The work done in [50], referred as Domain Selection Machine (DSM), regards the task of event recognition in online available videos. In this approach, the target domain consists of unlabeled videos while the source information is obtained from annotated images found via Flickr platform. The DSM methodology is realized as follows: SVM classifiers are created using multiple source domain datasets by extracting Scale-invariant feature transform SIFT [84] image features. These datasets are related to different semantic classes and are obtained through Flickr. The target classifier (SVM) consists of two parts. The first part is a weighted sum of the predictions of the source classifiers using key frames from the input video. The second part is a learning function which input consists of spatio-temporal features [51] from the input video and is trained from target domain data. This part selects the most relevant source domains by using a data-dependent regularizer.

Classical machine learning methods in comparison with the deep learning approaches, require less computational cost and there is not a need for big datasets [16]. Moreover, due to their low complexity, they are considered an easily explainable framework than the deep learning architectures [47]. In this sense, it is easier to track how specific decisions are performed from the algorithm in contrast with the state-of-the-art deep learning algorithms where the increasing depth of the networks makes the network explainability a challenging task [16]. All the above renders these approaches an attractive candidate for starting our research in domain adaptation. Hence, we decided to begin our research with a *classical machine learning* domain adaptation approach. Then, as a built-up step to proceed with a more sophisticated deep learning approach.

1.2. Emotion Recognition

The target of the current dissertation is to study domain adaptation techniques to uncover hidden relations between audio and face modalities from videos that contain emotional responses. Further to this, to evaluate the performance of the introduced approaches, emotion recognition (ER) was introduced as our main evaluation measurement tool. We would like to answer the question of whether domain adaptation improves the ER performance in the target modality X_T by making use of datasets distributed in the source domain X_S. However, in order to proceed with the development of our whole approach, it seems imperative to proceed first with an explanation of how emotion is determining and measured by humans. In this way, it will be much easier to analyze and understand the state-of-the-art ER strategies from a machine learning perspective. In Figure 1.6, instances of facial expressions and the corresponding emotion are illustrated (these instances were extracted from CREMA-D dataset).

Humans have the ability to rationalize about emotions which are crucial when experiencing social behaviors, which can help in relationship maintenance and could be beneficial in artistic endeavors, for political power scheming [52]. What is more, emotions are crucial in human perception and cognition.

Cognitive psychology researchers are studying the beneficial functions within the human brain that show the obvious role that emotions play during the process of decision making [59]. The collection of all cognitive processes for reasoning about others' emotions can be summarized by the term *affective cognition.*

People's emotion recognition mechanism is significantly varied in their efficiency at judging the affects of other people [52]. This aforesaid uncertainty that humans encounter when trying to identify emotions, especially from non-facial cues such as audio, gestures, or sentiments from the text is mitigated using modern technology. Additionally, this ambiguity is evident from the difficulty that the community of cognitive psychology is facing when trying to rigorously define and measure what exactly constitutes emotions [53]. Therefore, the question whether emotional states can be empirically measured, remains a controversial issue.

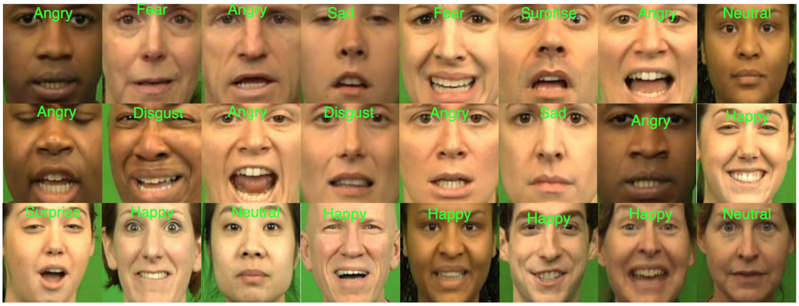


Figure 1.6: Examples of emotion recognition extracted from CREMA-D.

Firstly, regarding the definition ambiguity, an important obstacle is that emotional processes and states are complicated and can be studied from several points of view, therefore, a complete definition is virtually impossible. A safe way to approach this term is to include several well-known accepted definitions for the term "emotions". A major work towards this end is the work done in [54], where authors gathered 92 different statements for the term "emotions" and provide an analysis for these definitions. From our perspective the most interesting definitions that we found in the bibliography are the following ones:

- Emotions are expressions of the human internal states of mind, thinking, and feelings. They are linked to decision making, mood, motivation, and many aspects of cognition and intelligence [57].
- Authors in [58] suggested that affective states are enclosed body expressions, neurological and physiological responses, and cognitive and meta-cognitive states.
- A subjective feeling state involves physiological arousal, accompanied by characteristic behaviors [54].
- Scottish philosopher David Hume argued that: "Reason is, and ought only to be the slave of the passions ..." By "passions", Hume meant what we now call emotions [55].
- Darwin view of "emotions" as a way to convey information which helps organisms adapt and evolve [56].

1.2.1. EMOTION CATEGORIZATION

Regarding the ambiguity in measuring emotions, during the past decades, many studies have been performed to describe the way humans measure and attain an accurate understanding of others' emotional states [53]. Furthermore, a plethora of divergent models has been introduced that are trying to grasp the fundamental categorization in which emotions can be grouped. These theories can be grouped into two distinct categories namely, *manifestation* and *structure* (see Figure 1.7). The first is related to how emotions

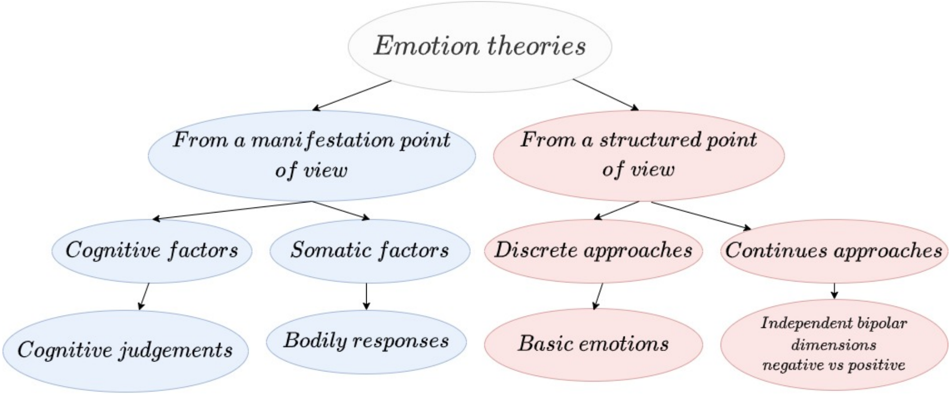


Figure 1.7: The fundamental categorization in which emotions theories can be grouped [52].

are experienced or perceived, hence, they are related more about the definition and the subjective understanding of emotions. The second category mainly relates to how different emotion responses can be measured. Each group can be better described by its sub-categories.

MANIFESTATION

In the first group, there are two sub-categories, namely, cognitive and somatic factors. The emotion theories based on cognitive factors consider cognition as a necessary element of emotion that takes a form of a thought or judgment. Under this category, emotions are experienced and perceived from the subjective manifestation perspective (a term that was analyzed by Karl Jaspers, and more details can be found in [60]). In this case, emotions are not directly measurable and can be only processed by humans through empathetic understanding. Furthermore, from the subjective manifestation perspective, emotions can be intentional or unintentional, conscious or subconscious, and can be formed as a judgment or a thought. In the same spirit, the authors in [61] analyzed and coined the term “cognitive appraisal” which they associated with a particular subject and the way that this person is perceiving its surrounding environment. A major advocate of this theory can be found in [62], where the importance of cognitive evaluations in reasoning about specific stimuli was stressed. Another example of a cognitive approach is the work done in [63], where authors tried to define emotions as a reaction to an affect-related event that consists of awareness of emotional stimuli and further, as being aroused and ready to act due to that stimulus.

The other pillar of emotion manifestations, namely, somatic theories, argues that somatic responses are more important than cognitive factors, and regard somatic reactions as the reason of emotional responses. The author in [64] considers the affect system of humans as the primary motivation system that can trigger and arouse other physical and bodily functions. Moreover, the work done in [69] analyzes the connection between evolutionary theories and emotions during the study of the primal behaviour of animals. Ekman in [66], who shares a similar perspective, regards emotions as psychosomatic conditions that have developed over time in a Darwinian manner due to their

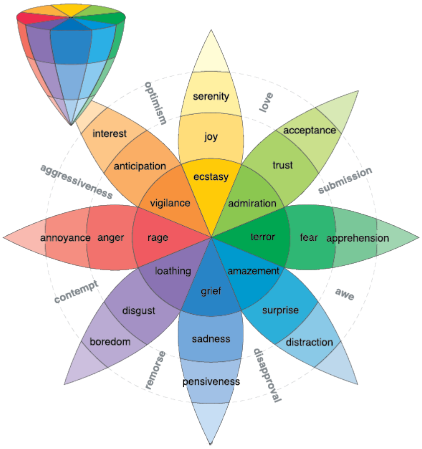


Figure 1.8: The emotion wheel introduced in [69].

adaptive value when tackling novel unseen life tasks. He proposed that the role which emotions play is to prepare a subject in a manner to handle new challenging situations by retrieving information of how similar events were encountered in the past.

STRUCTURE

From the *measurement* or the so-called *structured manifestations* point of view, we encounter the division between discrete or continuous approaches [52]. Discrete emotion approaches regard the existence of several distinct and universally recognized basic emotions. A prominent example of this type of categorization can be found in the work done by Ekman and al. in [67], as well as in the work done in [68], where authors proposed six basic emotions (namely fear, anger, disgust, happiness, sadness, and surprise). In these studies, it was stated that all people worldwide express and recognize basic emotions in a similar manner. Furthermore, by combining these six emotions, some more sophisticated emotions are derived. As an example, the emotion of agony is conceived as a variation of the basic emotion of fear and surprise. In the same spirit, the research held by Plutchik et al. in [69] claimed that there are eight basic emotions: joy, trust, fear, surprise, sadness, anticipation, anger, and disgust. In particular, an emotion wheel schema was proposed (this can be seen in Figure 1.8) and illustrates these eight basic emotions and the relations among them, such as intensity, combination and the converse relation. In particular, the different colors represent the intensity of the emotion. Moreover, it can be seen from the figure that we can perform mathematical operations to emotions such as addition. For instance when joy and trust (sequential emotions) are combined they produce as a result the emotion of love. Finally, joy and sadness are considered to be opposite emotions (they lie anti-symmetrical in the diagram).

However, there are some disadvantages that the models of discrete emotions are facing too. Firstly, there is not a consensus in the scientific community about the number and type of the basic emotions [59]. Furthermore, some wide criticism on the discrete models concerns their incapability to capture some other human emotions. However, discrete emotion models are widely used because of their simplicity and high plausibility.

The second sub-category of the structured theory models concerns emotion theories based on continuous approaches with the occurrence of two or more dimensions instead of strict emotion categorization. Russell in [70], instead of a small number of discrete emotions, introduced independent dipolar dimensions of emotion such as pleasure–displeasure. Another continuous affect model is the so called *Theory of Flow* [71][72][73]. This model is used to describe the emotion states and the engaging experiences of individuals during the execution of a specific task. The term “flow” can be defined as the cognitive state in which an individual experiences motivation, immersion and happiness. This model is mainly introduced to describe the emotion flow during the learning process and has been proven critical when designing games [74][75] as well as computer-based education tools [76]. The learning experience from the *Theory of Flow* perspective model can be represented by two dimensions, the skill of the learners and the challenge presented to them. Attaining a balance between these two parameters, a positive effect on the interaction process can emerge, which will potentially enhance the knowledge acquisition process. Another prevalent example is the bipolar valence-arousal [77] model. Emotions in this case are measured by using two questions. The first one concerns the degree of arousal (feelings of stimulation, excitation, arousal). The second one regards the degree of pleasure versus displeasure. Finally, the Pleasure, Arousal, and Dominance or PAD [78] emotional state model, is a psychological model to describe and measure emotional states. PAD employs three numerical dimensions, Pleasure, Arousal, and Dominance to represent all emotions.

All the above categories (see Figure 1.7) in the field of cognitive psychology is leveraged from the modern technology point of view to perform automatic emotion recognition. Terms like *affective computing* [59] and *emotional design* are related to the realization of emotions when designing computer systems in an attempt to make them more natural for humans to understand and use [79]. Picard in [80] coined and defined affective computing as *computing that relates to, arises from, or deliberately influences emotions.* Towards this end, recent progress has been made in developing *affective systems* which are able to detect and befittingly respond to human emotions and drive human–computer interaction towards a more enjoyable and efficient fashion.

1.2.2. AUTOMATIC EMOTION RECOGNITION

Utilizing machines to perform emotion recognition is a relatively recent research area that is becoming widely popular, especially during the last couple of decades in conjunction with the proliferation of deep learning techniques. Generally, as in the case of humans, technology works best when it leverages multiple modalities in a context [81]. Currently, research is mostly focusing on performing emotion recognition through facial expressions, audio segments, sentiments from written text, and health signals as measured by wearables. The goal is to predict high level cognitive content from the aforesaid low-level human-oriented emotion-related raw signals [82][100][83].

One way of classifying existing automatic Emotion Recognition (ER) approaches in the recent literature is based on the way that emotion information (or technically-speaking emotion features, which are information patterns that are associated with a specific emotion) are extracted from the input source. Hence, we have the following two main categories: handcrafted feature-based techniques and end-to-end learning methods.

Firstly, *handcrafted feature-based* techniques (or as they alternatively are called *conventional* approaches [88]) are based on manual feature extraction strategies and can be categorised into domain knowledge feature extraction and image description-based techniques.

- Regarding the first category, domain knowledge from the cognitive psychology perspective is employed to extract features for each different modality (which leads to the detection of certain emotion patterns). For each modality, different features can be extracted and applied in an attempt to retrieve information from the input sources of information such as video, image, audio, wearable sensors, or text. That extracted information is directly leveraged using rule-based models for retrieving emotions from the input modalities or using supervised learning methods. For instance, regarding facial features, Facial Action Coding System (FACS) [90] or Action Units (AUs) [91] are used in an attempt to extract features that are associated with emotions. Both FACS and AUs are concerned with facial movements and facial anatomy. The FACS system classifies, for example many human expressions in real life, and is the definitive reference standard for muscle movements in facial expressions today. One of the advantages of handcrafted feature-based strategies is the simplicity in planning and developing regarding the feature extraction process. Furthermore, these approaches do not require a normally high computational cost to be utilized. A limitation of this category of techniques is their inability to handle sophisticated patterns and complex features from the available datasets which cannot be extracted using just *conventional approaches*.

- Regarding the second category, the target of these approaches is to extract descriptors (from the whole image) that describe the image, in a way that emotion patterns can be accurately deduced. These descriptors correspond normally to low level image characteristics such as colors, image-texture, edges, and corners. In this category, techniques such as Scale-invariant feature transform (SIFT) [85], Speeded Up Robust Features (SURF) [86] and Local binary patterns (LBP) [87] are some dominant strategies. These approaches, while computationally more expensive than the domain knowledge feature extraction, still they require relatively lower computational cost than the deep learning approaches [92].

Secondly, end-to-end learning methods is a topic that receives increasing attention in the Machine Learning community with the proliferation of Deep Neural Network's (DNNs) [92]. In this case, a large set of labeled data is supplied to the system with the purpose of learning and predicting the appropriate emotion type feature patterns. While learning approaches provide normally reasonable classification performance, they normally need to be fed large training datasets. Some of the most commonly used machine learning algorithms originated from the proliferation of deep learning. Some widespread applied paradigms are the following: Convolutional Neural Networks (CNN) [93], Long Short-term Memory networks (LSTM) [94], Generative Adversarial Networks(GANs) [95],

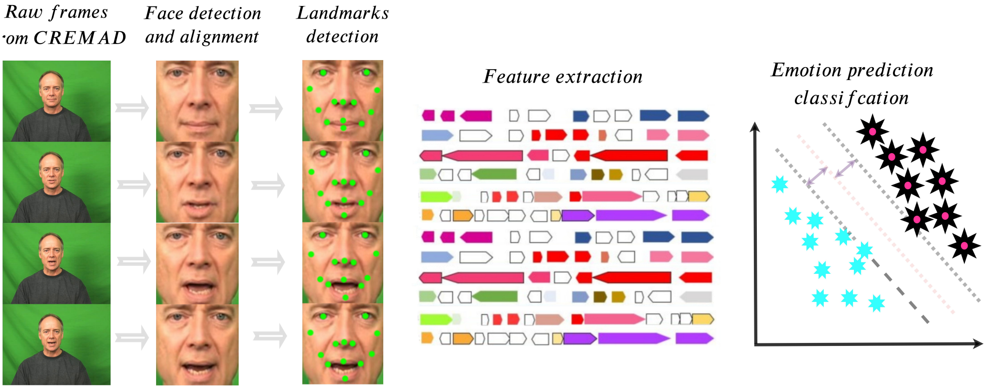


Figure 1.9: Handcrafted feature-based schema. The image samples are extracted from CREMA-D dataset.

autoencoders [96], Restricted Boltzmann machines [97] and Extreme Learning Machines (ELM) [98], as well as variations of those.

1.2.3. STATE-OF-THE-ART EMOTION RECOGNITION APPROACHES

Given that, in the current dissertation, we are analyzing audio and face emotion recognition (AER and FER), we will introduce the state-of-the-art for both modalities. In the following paragraphs, the most stimulating works that inspired the current dissertation are presented in detail.

FACE-BASED EMOTION RECOGNITION

Regarding the automatic FER systems, various types of approaches which lie within the categories of handcrafted feature-based techniques or end-to-end learning techniques have been studied.

As far as the first type is concerned, the common ground for all these approaches regards some pre-processing steps. These are the following: Firstly, the detection of the face region is performed to remove possible noise and superfluous background from the video. Secondly, alignment is applied to establish common coordinate references across all data samples (in case that the extracted faces from the previous step are not aligned). Subsequently, the extraction of geometric features, appearance features, or the combination of them is implemented. Finally, a classifier is implemented, where a learning scheme or some rational extracted rules are employed. This is done with the purpose of categorizing the extracted features into some basic emotions. These steps are better reflected in Figure 1.9.

Handcrafted feature-based techniques: In this paragraph, an introduction to state-of-the-art approaches concerning the handcrafted feature-based techniques is given. Authors in [101] studied FER in videos. They use two types of geometric features based on the position and angle of 52 standard facial landmark points. Firstly, for each frame, facial landmark detection is performed. Then, the angle and Euclidean distance between the detected and the standard landmarks within a specific frame are calculated.

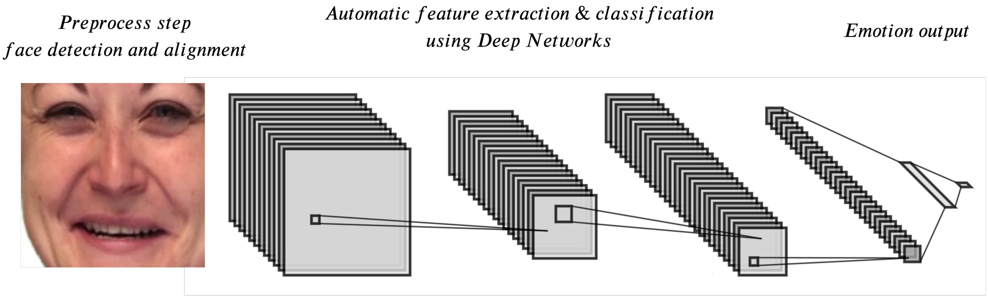

Preprocess step
face detection and alignment
Automatic feature extraction & classification
using Deep Networks
Emotion output

Figure 1.10: End-to-end learning. The face is extracted from the RAVDESS dataset [171].

Secondly, the subtraction between these calculated distances and angles and the corresponding distances and angles from the first frame of the video sequence is performed. Having extracted the aforementioned features, then, during the classification process, two different approaches are tested, namely multi-class AdaBoost and SVM.

For feature extraction, authors in [102] employed a local binary pattern (LBP) histogram of various block sizes from a global face region to extract feature representations. In addition to that, principal component analysis PCA-based classifier is utilized for each emotion. Although this method can be implemented in real time, the recognition accuracy tends to be decreased because it does not take into account local variations of the facial components in the extracted feature vector, and therefore, is an important to be taken into account in the extracted representation. Towards this end, the authors in [103] computed region-specific features by dividing the face region into domain-specific local regions. Local regions with high significance are extracted by employing an incremental search approach. In this way, the approach results in a decreasing number of feature dimensions and an increasing emotion recognition performance.

An approach for analysing and representing facial dynamics is introduced in [104]. This approach calculates optical flow caused by facial expressions in an attempt to capture the direction of motion. In [105], an optical flow-based approach is employed to uncover emotional expression by automatically recognising subtle changes in facial expressions.

Authors in [106] conducted a study by analyzing multi-view facial expressions. Three local patch descriptors (histogram of gradients, local binary patterns, and SIFT) are applied to capture facial features representations, which constitute the inputs to a nearest-neighbor indexing method that identifies facial expressions. Authors performed a study on the influence of dimensionality reduction on features when using principal component analysis (PCA) and linear discriminant analysis (LDA). These approaches are evaluated on multi-view data generated from BU-3DFE 3d facial expression database that includes 100 subjects with 6 emotions and 4 intensity levels.

In [88], a survey on conventional and deep learning face emotion recognition approaches, as well results from these approaches can be found. The main employed datasets are the following: Acted Facial Expressions in the Wild Dataset (AFEW) [169] containing the six basic emotions (fear, anger, joy, sadness, disgust, and surprise), Extended Cohn–Kanade Dataset (CK+) containing the basic emotions plus contempt [107], the Japanese Female Facial Expressions (JAFFE) containing the six basic emotions plus the neutral [108], NVIE Dataset [109] (it is labelled with six facial emotions, expression intensity, and Arousal–Valence label) and CMU Multi-PIE Database (Multi-PIE) [110]. As can be seen in Table 3 from [88]), with the recent proliferation of the deep learning approaches, it is possible to obtain better results by employing modern architectures and create more robust emotion recognition systems. That is mainly contributed to the end-to-end nature of deep learning approaches. These approaches provide an automatic way for extracting features which are task-related (for instance, emotion recognition) since these features are learned using task-related datasets. On the other hand, the conventional approaches employ strategies for feature extraction that are not task-related, and therefore, there is a divergence in the performances. Furthermore, deep learning approaches reduce the reliance on pre-processing steps, and hence, can be considered a more elegant alternative.

End-to-end learning approaches: A typical example of these approaches can be seen in Figure 1.10. In this example a Convolutional Neural Network (CNN) 3.1 classifier is employed to an input image in an effort to classify the image. The performed operations of this model include convolution with learnable parameters) and max pooling which are typical operators of the CNN (Appendix 3.1). By employing this model, we can extract features that capture better the input image distributions from our datasets. That can be done by using the last fully connected layer of the architecture.

In the same category, authors in [114] introduce CNN visualization methodology to comprehend the learned model using various FER datasets, and illustrated the efficiency of the trained models on emotion recognition. Authors in [115] are using two different CNN schemes: the first one is implemented to extract temporal features from videos, while the second one is applied to extract temporal geometry features from temporal facial landmark points. These two models are fused using an integration schema to improve the performance of FER. The work that is done in [116] is proposing a deep learning methodology called deep region and multi-label learning (coined as DRML). This approach is based on a region layer that used feed-forward functions to extract essential facial regions and enforces the trained weights to capture structural features from the face.

A CNN-based facial emotion recognition method is proposed in [117]. This work employs FACS and AUs feature representation, which showed a good generalisation capability for the networks for both cross-data and cross-task related to FER. A well-performed recognition rate is obtained when utilising Extended Cohn-Kanade (CK+), NovaEmotions and FER2013 datasets.

The work is done in [88], contains a review on conventional as well as end-to-end approaches. Furthermore, the standard FER approaches and their results from these studies are analyzed as well. As we have discussed before, these approaches provide multiple advantages over the conventional methods, however, there are several noteworthy disadvantages. Firstly, these approaches in order to avoid overfitting require a massive amount of data [88], which are not always available. Secondly, these approaches require an enormous computational power which can become an obstacle when designing such a system. Finally, while end-to-end approaches outperform the conventional methods, due to their black-box nature [111], sometimes it is a rather challenging task to explain their behaviour. Naturally, this raises several ethical questions [112], especially when these algorithms are employed in real-world applications [112]. There are applications where it is more crucial to explain the behavior of the algorithm and how it makes decision than its performance [111]. For instance, in self-driving cars, the decision can be made by algorithms that can harm pedestrians or drivers. In this case, it is really crucial for developers to be able to explain the behavior of the algorithms.

AUDIO-BASED EMOTION RECOGNITION

In this Subsection, the most influential works regarding audio-based emotion recognition (AER) are presented. AER is the task of automatically recognizing emotions from input audio clips that contain human audio or speech segments. From our literature review, it is obvious that the most prominent recent approaches are using spectrograms as for representing the audio clips, which, as a next step, are given as an input to a deep learning model. In [89] an analysis of the most prominent works in AER is performed. This paper contains the performance of state-of-the-art approaches in datasets such as eNTERFACE [125] and CREMA-D [170] (see Section 1.5).

Authors in [119] transformed speech input channel information into spectrograms, this is a visual representation of the spectrum of frequencies of a signal as it fluctuates over time. The implemented spectrograms were used as input to a Convolutional Neural Network. The proposed architecture is composed of three convolutional layers and three fully connected layers, which were utilized in order to extract distinct features from spectrograms and predictions for seven basic emotions. While in [121], a speech emotion recognition strategy based on phoneme sequences is introduced. In a similar way, in [119], authors converted the input speech signals into spectrograms. Furthermore, they performed various experiments with different kinds of deep neural networks with phoneme and spectrograms as inputs. Finally, an evaluation of all proposed network architectures is conducted.

Results from state-of-the-art AER approaches, as well as the employed datasets can be found in [88], [113] and [89]. A noteworthy observation from these studies (especially from [113]) is that recent approaches which employ deep learning models mainly make use of RAVDESS [171] and CREMA-D [170] datasets. Hence, this is a reason that we decided to use these datasets as well in our study. Another extracted observation is that in recent years there is the widely used implementation of attention mechanisms for AER, which are mainly employed for the feature extraction process. This provided our inspiration for performing a similar study which can be found in Section 4.

MUTLI-MODAL EMOTION RECOGNITION

Research in the cognitive psychology domain renders that the human brain employs a multimodal strategy for emotion recognition [1]. Therefore, multimodal approaches are proven useful for the field of affective computing as well. Moreover, it is found that multimodal emotion recognition outperforms single modal emotion recognition [123][124]. On the other end, a generic drawback of the multi-modal analysis is the lack of datasets in comparison with the single modality case. However, in the case of emotion recognition there are several available multimodal datasets. Some typical datasets for multimodal emotion recognition are AFEW [169], CREMA-D [170], RAVDESS [171] and eN-

TERFACE [125]. The surveys in [131][88] contain state-of-the-art research for multimodal emotion recognition for swallow and deep learning approaches.

Bengio et al. in [127] employ a dataset called Acted Facial Expressions In The Wild [132]. This work participated in the "EmotiW 2016 challenge" which is described in [169]. Their approach combined different deep neural network architectures from different data modalities, including: 1) a CNN that performs emotion recognition using facial expressions within video frames 2) a deep belief net to extract features from the audio 3) a deep autoencoder to model the spatio-temporal relations of the human actions in the entire scene and 4) a shallow network architecture focused on extracting features from the mouth region from the first human subjects in the scene.

The work done in [128] presented a methodology built for facilitating the Emotion Recognition in the Wild 2017 video-based sub-challenge. The goal of this sub-challenge is to classify the basic emotions, namely, anger, sadness, happiness, surprise, fear, disgust, and neutral. In the proposed solution, the authors applied three state-of-the-art techniques to tackle the emotion recognition task. Deep network transfer learning is used for feature extraction. Spatio-temporal model fusion is applied to augment the different networks. Finally, semi-auto reinforcement learning is applied for the optimization of the fusion strategy.

The work done in [129] was motivated by the recent proliferation of deep learning in bridging the semantic gap between audio and video. Authors proposed a multimodal CNN which fuses the audio and visual cues in a deep model. This multimodal deep model was trained in two phases: Firstly, two deep CNN models, which are pre-trained on large-scale image data, are fine-tuned to perform audio and visual emotion recognition tasks, respectively, on the corresponding labeled speech and face data. Secondly, the output of these two networks was embodied in a final augmentation network which was developed using fully-connected layers. This augmented network was trained with the purpose of obtaining a joint audio-visual feature representation for emotion recognition.

Authors in [130] accommodated the following research questions: Firstly, how to robustly combine information from different modalities, such as audio and face, and secondly, how to deal with incremental emotion relations on these modalities. On this ground, a data-driven unified multimodal incremental deep learning approach was introduced in order to explore the variation of emotion expression over time. Their proposed method was evaluated on two datasets, CREMA-D and RAVDESS.

1.3. DISTANCE METRIC LEARNING

In the current dissertation, Distance Metric Learning (DML) [133][135] is employed as the first strategy to facilitate the task of domain adaptation within the emotion context. By using DML techniques, we are able to project face and audio-related features into a common hyperspace where the domain distribution divergence is minimized. The main reason that DML is chosen as a candidate approach is due to the essence of the approach. In particular, it regards the calculation of a new distance metric that binds together similar objects while conversely, increases the distance of dissimilar objects as much as possible [133]. In our research, we employ this approach to bind together similar instances from X_S and X_T and push away conversely dissimilar instances. In this

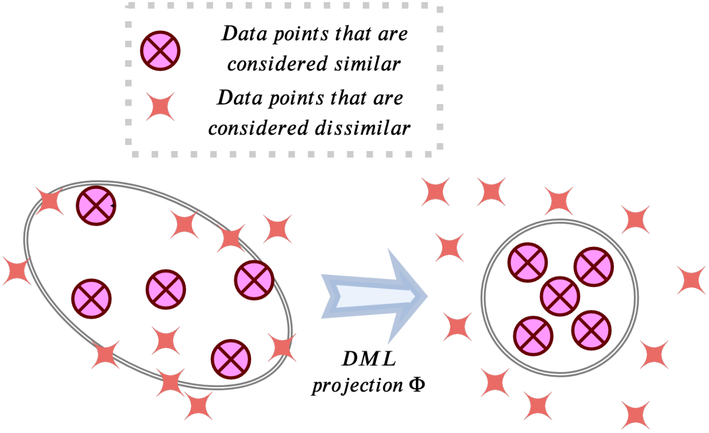

Data points that are considered similar
Data points that are considered dissimilar
DML projection Φ

Figure 1.11: DML projection learning.

way, the divergence between two different domains is minimized when using this new distance metric.

A more rigorous definition of DML [133] is concerned with learning a distance function designed for a specific task by making use of some supervised information. A possible intuitive formulation for the distance metric learning task is the following: Firstly, we have an initial distance function $d(x, y)$ between samples x and y (this could be, as an example, the simple Euclidean distance), coupled together with some supervised information that demonstrates which data pairs are indeed semantically close and which are not considered close [135]. DML's target is to frame a new distance function $\hat{d}(x, y)$ that is more "efficient" than the initial distance function $d(x, y)$ [135]. All the metric learning methods that we study in our work and which are a major inspiration for our work, assume that there is some supervised information available in hand. In the current dissertation, the hypothesis that the annotations from both domains can work as supervised input information for computing and optimizing the new distance function $\hat{d}$ is placed. The data samples from both domains are derived from the same events (as an example could be that the face and audio are extracted from the same video sequence). As an example of the aforesaid procedure, let us hypothesize that there are available data from the face domain $X_F \subset D_F$ with $y_i,\ i \in 1,...,6$ the samples' annotation (the six basic emotions) and data from audio domain $X_A \subset D_A$ with similar annotation $y_i,\ i \in 1,...,6$. One possible strategy for domain adaptation is to learn a new more sophisticated distance $\hat{d}()$ to bring closer samples from X_F and X_A domains that are considered semantically similar (having the same annotation). At the same time, to increase the distance between samples from both domains with dissimilar annotation.

The whole procedure is explained in more detail in Chapter 2. Furthermore, the conducted process is better visualized in Figure 1.11 where, in the left part of the figure, it can be observed that the similar and dissimilar data points could not be easily separated but after the implementation of the DML new distance $\hat{d}(X_F, X_A)$ (displayed in the right part of the Figure), it is much easier to find a separation between these two categories.

The utilization of the aforesaid proposed strategy is introduced in order to drive both domains closer. Having calculated the new distance function, we can apply this distance and project the input datasets into a common hyperspace. There, both modalities are fused into a new augmented dataset, and emotion recognition is implemented. As we already mentioned, our task is to figure out whether we can improve the performance of emotion recognition by using that fused dataset.

1.3.1. OVERVIEW OF DISTANCE-METRIC LEARNING TECHNIQUES

In this subsection, a revision of the state-of-the-art techniques for Distance Metric Learning that provide fruitful inspiration for our work is presented. Firstly, the surveys in [133][135] introduced the state-of-the-art approaches of *distance metric learning*. In the same manner with Section 1.2 this approach can be divided into *classical distance metric learning* and *deep distance metric learning* techniques.

Classical distance metric learning: Authors in [136] proposed a supervised Mahalanobis distance metric by incorporating pairwise constraints. These pairwise constraints are represented with feature vectors $[(a, b), (c, d)]$ where (a, b) pairs are considered to be similar (for instance belongs to the same class) while (c, d) are not considered similar. Thus, the distance within the first pair is significantly smaller than the distance within the second pair of vectors: $d(a, b) < d(c, d)$. A convex objective function is proposed to minimize the sum of squared residuals of the constraints. By learning and then employing the Mahalanobis distance, it is assumed that the distance is more accurate between the pairwise feature vectors.

Authors in [137] introduced a technique called Domain Adaptation Metric Learning (DAML). The proposed approach established an input data dependent regularization for the conventional distance metric learning and its representation in the kernel Hilbert space (RKHS). Data-dependent regularization is introduced with the purpose of minimizing the distribution difference by employing the empirical maximum mean discrepancy [138] between source and target domain data in RKHS. The optimization method that is introduced for calculating the Mahalanobis distance and consequently the reproduced RKHS is similar to the straight-forward Distance Metric Learning. However, the distance between the distributions of datasets in the RKHS space is also incorporated in the optimization formula. Authors performed extensive experiments using four different popular face recognition datasets and a large scale Web image annotation dataset for cross-domain face recognition and image annotation tasks under several settings. Finally, from the experimental results, the effectiveness of the DAML technique against state-of-the-art Distance Metric Learning techniques is grounded.

Deep distance metric learning: Recent advancements in machine learning led academia to foster deep learning networks for DML as well. These approaches normally introduce a deep learning architecture with which the input data can be transformed into a feature space where the distance of similar objects will be as small as possible while the distance of dissimilar objects will be maximized. In [139], an extensive analysis on the deep distance metric learning field is introduced, as well as the state-of-the-art approach for this domain is analyzed. In this work, authors group the state-of-the-art in two categories, namely, in *Siamese networks* and secondly, in *triplet loss-based* approaches.

In the first category, a pair of similar $(\mathbf{x}_i,\mathbf{x}_j)$ or dissimilar samples (based on some "supervised" or "unsupervised" information) is processed from a two-stream neural network where each sample processed from a different network, and then, the constructive loss is employed to optimize the network and learning a projection which will bring similar samples closer and push dissimilar samples further away.

In the second category, again, a two-stream network is also trained, however this time, by minimizing a triplet loss objective. This loss exploits the training annotation to create triplets. More concretely, given a triplet $(\mathbf{x}_i,\mathbf{x}_i^+,\mathbf{x}_i^-)$, where $\mathbf{x}_i$ is similar with $\mathbf{x}_i^+$ and dissimilar to $\mathbf{x}_i^-$. The target is this case is to minimize the following hinge loss:

$$L = \sum_i h(\tau + d_f(\mathbf{x}_i,\mathbf{x}^+) - d_f(\mathbf{x}_i,\mathbf{x}^-)) \tag{1.5}$$

where $h(\mathbf{x}) = max(0,\mathbf{x})$ is the hinge loss function, and $\tau > 0$ is a margin between $d_f(\mathbf{x}_i,\mathbf{x}^+)$ and $d_f(\mathbf{x}_i,\mathbf{x}^-)$. The triplet network pulls similar samples closer to the reference and pushes dissimilar samples further away.

Regarding related work for the first category and siamese networks, Hu et al. [140], implemented a deep distance metric learning for the task of face verification. In particular, they project face pairs into the same dimensionality subspace, in which the distance between a pair of face pair samples is less than a threshold. In the same way, the distance of the negative pair should be higher than a second larger threshold. In this way, *the discriminative information* can be exploited by a deep network. The proposed method achieved high face verification performance when implemented with the LFW [142] and YouTube Faces (YTF) [143] datasets.

Taigman et al. [141] introduced an end-to-end metric learning method called DeepFace, which employs a siamese network for face recognition. In contrast to the previous work, the loss is calculated using several parts of the networks, using the convolutional, pooling, and fully connected layers. Finally, the parameters of the siamese network are trained by the standard cross-entropy loss and back-propagation method.

Regarding the triplet loss-based methods, Hoffer et al. [144] employed a triplet network architecture for DML, where the target is to extract meaningful representations by distance comparison. The "real-world application" of this methods is a deep ranking similarity function for image retrieval. They also performed an extensive study of the triplet architecture, and indicated that the triplet loss-based methodology is an efficient alternative to the siamese methodology.

Authors in [145] developed the "FaceNet deep model" which learns a mapping from the input feature space to a projected Euclidean space (represented by CNN embeddings of the deep model) by using a triplet loss. Once this space is developed, face recognition and clustering which are the main tasks of this work can be applied more efficiently. Furthermore, triplets of aligned matching and non-matching faces are deployed for training the model by optimizing the triplet loss. The target is the squared distance between all faces, independent of imaging conditions, of the same identity to be small, whereas the squared distance between a pair of face images from different identities to be large.

1.4. Generative Adversarial Networks

By performing our first strategy to facilitate the posed domain adaptation problem using a preliminary non-deep learning approach, we extracted several worthwhile deductions about how we can further proceed. Firstly, the technique was critically dependant on the initial feature selection, as well as the clustering schema. Furthermore, while the method led to some initially promising results, after experimentation with novel deep learning approaches, it was evident that the amount of knowledge transfer can be improved further. That is in the same spirit with the research found in [131][88].

Hence, a more sophisticated approach is adopted to tackle the challenging task of domain adaptation within emotion context and is based on the implementation of Generative Adversarial Networks (GANs) techniques. A variety of innovative GANs topologies were deployed as in the case of DML, to study these cross-modal relationships between the symbiotic modalities of video (shown in Figure 1.1) and perform domain adaptation.

Over the recent years, an explosive popularity has emerged in the domain of GANs [146], which became one of the most promising developments in Deep Learning. The preliminary idea of GANs can be framed as follows: Given a vector of random noise z, the whole process endeavors to accomplish a good approximation of the data distribution in hand by learning a mapping between the noise distribution and that domain. GANs usually consist of two different neural networks which compete in a min-max manner. These networks are called Generator G and Discriminator D. An example of this methodology is depicted in Figure 1.12. An illustrative example of how GANs functions is introduced in [146], where the basic idea of the GANs technique is rendered. In this work, the target is to train a network G that, given a noise vector z, will be able to generate new samples derived from the MNIST dataset domain (target domain) by trying to approximate the desired distribution. In the meantime, D tries to decide whether the generated samples are genuine or not.

Starting from the above-mentioned architecture (the so-called vanilla architecture), the scope of the proposed research is to modify it and adapt it to the needs of our goals. The desired objective is to develop a framework that will be able to not only generate data in the target domain but also to convert source samples into target domain ones. Thereby, several modifications are needed in the classical version of GANs for the sake of formalizing a system that will be able to perform as such. In Chapter 3 a detailed analysis of the way that these modifications from the initial vanilla architecture to the more sophisticated architecture that is used to study these cross-modal relationships is provided.

1.4.1. State-of-the-art Generative Adversarial Networks techniques

In this sub-section, state-of-the-art techniques for Generative Adversarial Networks which influence our work are presented. These works help us to shape our understanding of the generative models and work as a compass since they provide a direction for the current dissertation.

In conditional GANs that are introduced in [148], networks G and D are conditioned to some variables c that represent the label information of the class. The model not only managed to generate data that robustly represent the distributions associated with their

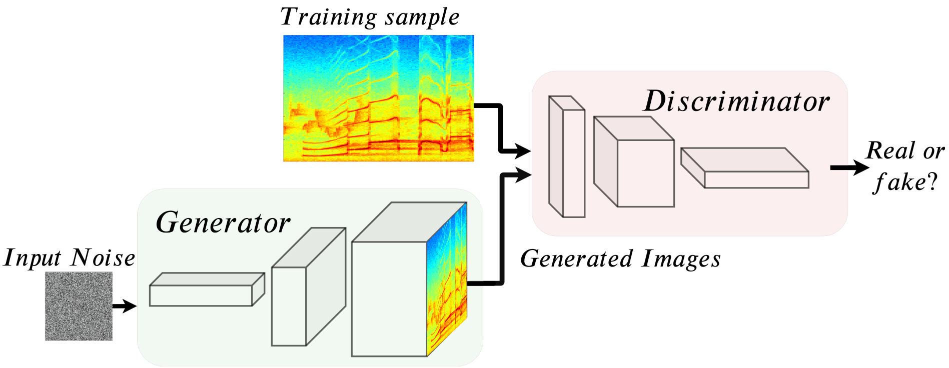


Figure 1.12: Vanilla architecture of the Generative Adversarial Networks.

labels, but also improved the quality of the generated data. Similarly, in [147], a modified version of the initial GANs which makes use of Deep Convolutional Neural Networks for the G and D networks is proposed. Authors in [153] presented an approach to learn how to translate an image from a source domain X_S to a target domain X_T, without having any available paired information among these two domains. The main objective of the approach is to learn a mapping $G: X_S \rightarrow X_T$ such that the distribution of images from $G(X_S)$ domain is equivalent with the distribution X_T using an adversarial loss. Since that mapping is highly under-constrained, the authors banded it together with an inverse mapping $F: X_T \rightarrow X_S$ and introduced a cycle consistency loss to force $F(G(X_S)) \approx X_S$ (and vice versa). Qualitative and quantitative results were delivered on several tasks where paired training data were not available. Furthermore, in [149], a conditional deep Generative Adversarial Network is proposed with the aim of performing image-to-image translation.

In DiscoGAN [154], authors performed a study of cross-domain relations given unpaired data (representing two different image domains). Their aim is to map characteristics from the source domain and transfer them to the target domain. For instance, in the case of face conversion, they keep some characteristics (accessories, color of hair) from the input image and transfer them to the target image. They propose a method based on GANs whose target is to expose connections between different domains. Using these relations, the proposed network effectively transfers “style” from one domain to another while maintaining important image features such as orientation and face identity. The term “style” is used to denote some semantics that are related to the implemented dataset from the input source domain X_S. For instance, in the case of face conversion, the set $X_M \subset D_M$ represents male celebrities while set $X_F \subset D_F$ female celebrities. The task is by keeping some characteristics from X_S (for instance the color of hair, the facial pose or whether the image contains glasses) to generated samples from the target domain X_M by using this semantic information. In [149], authors proposed an image-to-image translation schema that is based on a CNN architecture that is called U-Net [150]. The architecture is based on an encoder and a decoder module that have linked layers, thus, early information from the convolution layers of the encoder is still preserved, since that information is merged with the layers of the decoder module.

The work titled Cycle-Consistent Adversarial Domain Adaptation, or CyCADA [155], introduced a novel discriminatively-trained technique. Authors suggested that GANs, combined with cycle-consistency loss, are surprisingly effective at mapping images between domains, even without the use of aligned image pairs. The idea of cycle-consistency is that an image output by the first generator is used as input to the second generator and the output of the second generator should match the original image. CyCADA transforms image domains at both pixel and feature levels and enforces cycle-consistency constraints while leveraging a task loss. To validate this approach, the authors applied their model in the task of facial expression recognition by using the CK+ facial expression database.

In [156], authors focused on establishing a bridge between the definitions of GANs and Variational Autoencoders (VAE) by reformulating the definition of GANs. They perceived the generation of samples as performing posterior inference (similar to the way that VAE is formulated). That is done by adding a regularization loss into the latent space of the generator, using KL-divergence (in a similar fashion to VAEs). In that way, several state-of-the-art strategies already used for VAE can be implemented for GANs as well, and vice versa. For instance, the adversarial learning can be established for the VAE approach, while the importance weighting method in VAE can be implemented for GANs.

The state-of-the-art audio-visual studies that mainly influence our study are also presented in this paragraph. An captivating study of cross-modal relationships of audio and visual cues is introduced in [157], where conditional GANs are employed with the purpose of generating data from one modality while another modality is given as an input. For this purpose, authors use two separate networks (image-to-sound and sound-to-image) in order to perform cross-modal generations in both ways. They explored different encoding methods for audio (by making use of spectrograms derived from Mel-Frequency Cepstral Coefficients (MFCC) [164] or Short-Time Fourier Transform (STFT) [165]) and visual signals. To test the proposed approach, authors developed two separate novel datasets with pairs of images and sounds of musical performances of different instruments. Inspired by this work, authors in [163], built a model called Cross-Modal Cycle Generative Adversarial Model to perform cross-modal mappings between image and audio. This approach is composed of four kinds of subnetworks: audio-to-visual, visual-to-audio, audio-to-audio, and visual-to-visual subnetworks respectively, which are organized in a cycle architecture, and the training approach took place simultaneously. The tested datasets were the ones introduced in [157].

Another slightly different study is performed in [158], which aims in tackling speech-driven facial animation. In this problem, given an image and an audio clip, the aim is to generate a video clip of facial animation representing the input face and the information extracted from the audio clip. In an attempt to do so, authors made use of temporal GANs. In particular, the proposed architecture contains three different discriminators focused on achieving detailed frames, audio-visual synchronization, and realistic expressions.

In terms of the whole training stability of the GANs architecture, authors in [159] proposed a modified version of GANs (called Wasserstein Generative Adversarial Networks wGANs) with the aim of defining a more stabilized version concerning the optimization convergence of GANs by using Wasserstein distance as the loss function. An extension of that work is established in [160] where some improvements are introduced for wGANs. In the work that is done in [161] called Loss-Sensitive Generative Adversarial Networks on Lipschitz Densities introduced a novel loss based on Lipschitz regularization.

In a nutshell, GANs are a novel and powerful tool for generating samples and approximating distributions, or, as in the case of the current work, for reducing the discrepancy between two different distributions. It is the most efficient approach in terms of generating distributions, it is easy and intuitive to understand and has led to an enormous number of interesting applications such as generating images of human faces, objects, artistic images, high resolution pictures and many more [162].

However, this approach came with some notorious drawbacks [95]. The training process of the approach is focusing on solving the minimax game between G and D. As it is evident from several studies, the existence of the minimax converging point (the Nash equilibrium [95]) of the GANs game is challenging to be discovered, and the training of GANs is highly unstable and difficult to converge [95]. Furthermore, another important challenge for GANs is the mode collapse issue. This issue is very common when training a GANs model. In particular, a mode collapse restricts GANs ability of diversity. The G is only forced to deceive the D, by generating samples with limited variety without being able to reproduce the versatile distributions of the real data [146].

1.5. BENCHMARK DATASETS FOR MULTIMODAL EMOTION ANALYSIS

The previous sections introduced several approaches that were implemented in the current dissertation namely **Metric Learning** and **Generative Adversarial Networks**. These approaches were needed to be evaluated in the task of emotion recognition. Therefore, in order to perform so, we decided to make use of four state-of-the-art datasets, namely AFEW [169], CREMA-D [170], RAVDESS [171] and VoxCeleb [172], which were published in the academic field of affective computing. All datasets contain, as annotations, a discrete number of emotions (except for VoxCeleb which does not contain any annotation), and therefore, our study lies in discrete emotion modeling approaches. We did so, since this type of annotation (discrete emotion categorization) is widely used and accepted, and furthermore, we were able to make use of large amounts of annotated data that were necessary to properly train our methods.

1.5.1. ACTED FACIAL EXPRESSIONS IN THE WILD

The firstly described dataset is the popular Acted Facial Expressions In The Wild (AFEW) dataset. Samples from AFEW can be seen in Figure 1.13. AFEW [1] is a challenging dataset for emotion recognition since the data gathering procedure occurred in uncontrolled environments (from popular movies) and not within a controlled lab environment. Furthermore, it consists of data with occlusions, varying illumination, and head poses that normally can be met in real world conditions. The dataset consists of 1127 videos (of varying duration) annotated with the six basic emotions, namely anger, disgust, fear,

[1] `https://cs.anu.edu.au/few/AFEW.html`

Figure 1.13: Examples that are extracted from the AFEW dataset [169].

happiness, sadness and surprise plus neutral. The videos were extracted from professional actors in movies mimicking spontaneous human expressions. Moreover, the authors covered the evolution of the user emotions of the same actors in different ages (e.g. the emotion evolution of actors in Harry Potter movies). The range of ages of the actors varies from 1 to 70 years old. In total, 37 different movies were used to extract video clips from a great variety of different movie-genres and, furthermore, a great variety of expressions. Finally, information about movies and actors is contained in an XML schema. More specifically, the dataset contains the actor's expressions, the start time, the clip length, actor's name, the current facial pose of the actor, age of the character, age of the actor and their gender.

1.5.2. CREMA-D

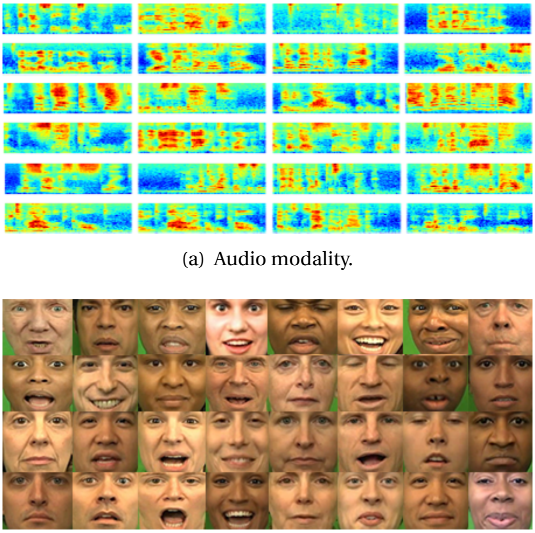
The second introduced dataset is CREMA-D. It is an audio-visual emotion expression database which is publicly available[2]. It encompasses 7442 videos (which have an average length of 2.63±0.53 seconds) from 91 actors (43 females and 48 males). Their age ranged from 20 to 74 years old and they stem from a diversity of races and ethnicities (African, American, Asian, Caucasian and Hispanic). Actors were requested to pose 12 sentences that are associated with six different emotions (anger, disgust, fear, happy, neutral, and sad) with four different levels of intensity ("low", "medium", "high" and "unspecified"). The dataset is a result of an effort to generate standard emotional stimuli for neuro-imaging studies, which require a wide range of intensities and separation for visual and auditory modalities presentation. The dataset's annotation is based on the presented videos that were shown to the actors.

1.5.3. RAVDESS

The next described dataset is the so-called *Ryerson Audio-Visual Database of Emotional Speech and Song* (RAVDESS). This corpus is a large-scale multimodal emotion expression dataset derived from speech and song segments[3]. The database is gender-balanced consisting of 24 actors posing the following expressions: neutral, calm, happy, sad, angry,

[2] github.com/CheyneyComputerScience/CREMA-D

[3] https://zenodo.org/record/1188976$#$.XHPO8-hKi70



(a) Audio modality.

(b) Face modality.

Figure 1.14: Examples from CREMA-D dataset [170].

fear, surprise, and disgust. Each expression is produced at two intensity levels. All cases are available in face-and-voice, face-only, and voice-only formats. In our experiments, we use the speech segments. The 7356 recordings (videos have an average duration of 3.82±0.34) were rated 10 times on emotional validity, intensity, and genuineness. Ratings were provided by 247 individuals. Finally, we decided to keep the clips from the dataset with the emotional expressions that overlap with the ones which can be also found in CREMA-D (thus, skipping Surprise and Calm).

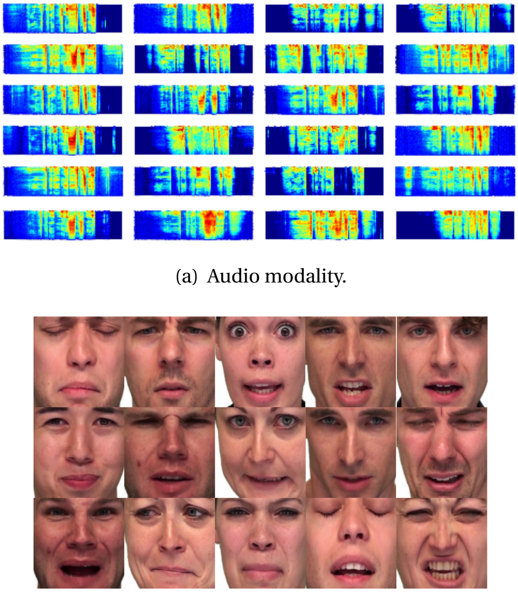
1.5.4. VOXCELEB

Finally, the VoxCeleb [172] dataset was selected[4]. The recording videos in the VoxCeleb dataset are videos based on interviews of 1,251 celebrities uploaded to YouTube with over 100,000 audio-visual clips (7.8±0.65). The speakers cover a wide diversity of different ages, nationalities, professions and accents. The dataset is roughly gender-balanced. The clips also contain audio in different languages. While the identities of the speakers are available, the dataset has no emotion-related labels. Alongside the data, the authors implemented and provided a classification model for emotion recognition through the facial modality, which is described in [172].

1.6. PROBLEM STATEMENT AND RESEARCH QUESTIONS

The previous sections discussed the main pillars of the current doctoral dissertation that are *domain adaptation*, *emotion recognition*, *distance metric learning*, *generative adversarial networks*, and the datasets which are used to evaluate the performance of

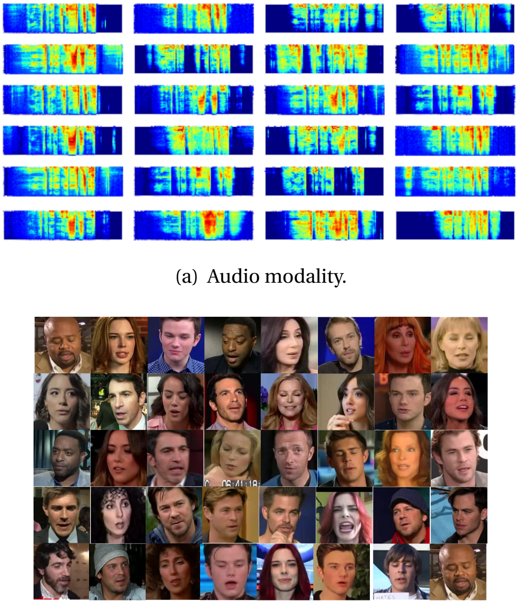
[4] http://www.robots.ox.ac.uk/~vgg/research/cross-modal-emotions/



(a) Audio modality.

(b) Face modality.

Figure 1.15: Examples that were extracted from RAVDESS dataset [171].



(a) Audio modality.

(b) Face modality.

Figure 1.16: Examples that were extracted from the VoxCeleb dataset.

the introduced approaches. As it was already mentioned, the undertaking quest which summarizes the core research of the current dissertation could be formulated using the following problem statement:

> **Problem statement:** Is it possible to transfer knowledge between the face and the audio modalities from the machine perspective when the emotion context is taken into account?

The core inspiration of the current study lies in the cognitive psychology domain and the conducted research in the domain of knowledge transfer between the face and audio modalities from the computer science perspective. Furthermore, the deduction that was inferred during the literature review that the emotion recognition from face modality performs better that modalities like audio [11]. To answer the posed problem statement, five research questions have been formulated. They deal with the steps in which we could face and decompose the posed problem and also with the steps that we followed and presented in the following chapters. The first posed question is framed as:

> **Research question 1:** How could domain adaptation approaches be employed to improve emotion recognition in one modality and perform knowledge transfer from the other modality?

Having stated the fundamental question of the conducted research, we should proceed to answer it by introducing more detailed questions about the nature of our approaches. Therefore, the next research question concerns the implementation of metric learning strategies (which are analyzed in Chapter 2) for our audio-visual domain adaptation task. This question can be framed as follows:

> **Research question 2:** How could a metric learning framework be deployed to model audio-visual cross modal relationships?

Subsequently, a more sophisticated deep-learning based approach is introduced to model the more complex patterns that govern this cross-modal relationship. That approach used a deep learning architecture which is based on Generative Adversarial Networks (GANs) and the posed research question is the following:

> **Research question 3:** How could Generative Adversarial Networks be deployed to uncover static audio-visual cross modal relationships?

The answer to the aforesaid question can be found in Chapter 3. Regarding the technical aspect of the posed question and the way that the training process of GANs architecture is conducted, we can pose another more sophisticated question. This question is associated with the way that the optimization process of the whole network is held. On this ground, by unfolding the previously posed question, we could state the following question:

> **Research question 4:** How could we learn and train the proposed Generative Neural Networks in a progressive manner? Or in simple terms, how can we initialize the weights of the network in such a way that we will optimize the training process?

Furthermore, inspired by several proposed works that can be found in the literature, it is evident that an investigation of the temporal cross-modal relationships between the face and audio modalities is imperative in our study. Therefore, on top of the previously posed tasks, we decided to proceed with the study of the time dynamics between both domains. While that is partially done and described in Chapter 2, during the implementation of the DML strategy, it is developed in more detail and structured way in Chapter 4. The whole research endeavour of this study can be framed from the following posed question:

> **Research question 5:** Could the temporal dynamics between audio and visual information be modeled and help us in the domain adaptation task?

Finally, having extensively studied the way that face modality can help in improving the audio emotion recognition in all the previous questions, in Chapter 5, it is chosen to focus on researching the converse task, by answering the following posed question:

> **Research question 6:** Is it possible to improve face emotion recognition by eliciting information from the audio modality?

1.7. THESIS OVERVIEW

This thesis is organized into 6 chapters. Chapter 1 introduces the basic terms such as *domain adaptation*, *emotion recognition*, *distance metric learning*, *generative adversarial networks*, and the utilized datasets which are analyzed. Furthermore, an explanation of our current posed problems regarding this thesis is introduced and formulated using five different research questions that guide the rest of conducted research.

Chapter 2 describes a metric learning framework for facilitating the task of audio-visual cross-modal domain adaptation. In particular, a framework that consists of four steps is deployed. These steps are namely: feature extraction, feature selection, clustering, and finally, distance metric learning. The study aims at performing symmetric domain adaptation, therefore, the goal is to project the input data points of face and audio modalities into a common hyper-space where the divergence between the distributions of both modalities is minimized. The evaluation scenario is based on the improvement of the classification performance of audio emotion recognition when using the projected data from face-related datasets. Furthermore, an evaluation measurement that examines how close the domains are before and after the implementation of the current framework is also examined. Finally, the advantages and disadvantages of this approach are discussed in the conclusions of the chapter.

Chapter 3 addresses the third and fourth raised research questions. It establishes another way of performing domain adaptation. At this stage, instead of using metric learning, we decided to modify our strategy and employ a deep learning architecture with the purpose of performing cross-modal mapping. The methodology is based on Generative Adversarial Networks. We analyze the proposed architecture and different ways of calibrating the weights of the architecture. That approach lies in the category of asymmetric domain adaptation and could be seen as a generative approach where the one modality is generated when the other modality it is given as input to the approach. Besides the description of the method, multiple ways of evaluating the proposed approach are analyzed in the chapter. Furthermore, the capability of that approach for generating *high quality* samples and approximating the target domain is introduced and examined. Finally, an ablation study is discussed in an effort to show how we were led to use the current topology.

Chapter 4 addresses the fifth research question. It concerns the modification of the proposed approach for modeling the temporal dynamics that govern audio-visual cross-modal relationships in a video clip. We firstly modified the proposed architecture of Chapter 3 which is based on a static image-to-image translation approach, to take into account temporally dynamics between the features of the face from video clips. We did keep the same generative strategy, however, we decided to modify the embodied optimization schema. In this manner, we not only target to improve the performance of the employed approach but also the time complexity by improving the way that the proposed approach is optimized. Therefore, the Wasserstein distance (or as it called alternatively, Earth-Mover distance) is analyzed. The evaluation of the aforesaid approaches is done in a similar manner as in Chapter 2 and 3. In this way, we ensure that the reader can deduce noteworthy conclusions about the method in comparison with the approaches that are introduced in the previous chapters. Finally, the capability of that approach for generating high quality samples and approximating the target domain is introduced and examined.

In Chapter 5, we are grouping together all the aforementioned strategies from the previous chapters. However, the objective of the chapter is to highlight the performance of the converse DA task that is related to improving the face emotion recognition (when leveraging knowledge from the audio modality). That is done in an attempt to evaluate the versatility of the proposed approaches. In this respect, we are measuring the performances of each approach and we are trying to determine which of these techniques are performing better regarding the time complexity and the classification performance. The capability of that approach for generating high quality samples from the face is also examined. Chapter 6 contains the conclusion of the whole conducted research, as well as it provides an overview of possible future research directions. Finally the Appendix contains an explanation to basic notions that are used in this dissertation such machine learning, neural networks, deep learning and support vector machines. The explanation for these serves as an introduction and can help the non-experienced reader in several parts of the dissertation.

REFERENCES

[1] T.Grossman, The development of emotion perception in face and voice during infancy, Resorative Neurology and Neuroscience, Volume 28, Pages: 219–236, 2010.

[2] D.W.Massaro and M.M.Cohen, Perceiving Talking Faces, Current Directions in Psychological Science, Volume 4, Number 4, Pages: 219–236, 1995.

[3] H.McGurk and J.MacDonald, Hearing lips and seeing voices, Nature, Volume 264, Number 5588, Pages: 746–748, 1976.

[4] C.Bayard, C.Colin and J.Leybaert, How is the McGurk effect modulated by Cued

Speech in deaf and hearing adults?, Frontiers in Psychology, Volume 5, Number 416, Pages: 746–748, 2014.

[5] E.Cvejic, J.Kim and C.Davis, Prosody off the top of the head: Prosodic contrasts can be discriminated by head motion, Journal of Speech Communication, Volume 52, Number 6, Pages: 555–564, 2010.

[6] M.Swerts and E.Krahmer, Facial expression and prosodic prominence: Effects of modality and facial area, Journal of Phonetics, Volume 36, Number 2, Pages: 219-238, 2008.

[7] M.D.Pell, Prosody–face Interactions in Emotional Processing as Revealed by the Facial Affect Decision Task, Journal of Nonverbal Behavior, Volume 29, Number 4, Pages: 193-–215, 2005.

[8] C.A.Corneanu, M.Oliu, J.F.Cohn and S.Escalera, A Survey on RGB, 3D, Thermal, and Multimodal Approaches for Facial Expression Recognition:History, Trends, and Affect related Applications, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 38, Number 8, Pages: 1548–1568, 2016.

[9] J.Kumari, R.Rajesh and K.M.Pooja, Facial Expression Recognition: A Survey, Procedia Computer Science Volume 58, Pages 486-491, 2015.

[10] P.Liu, S.Han, Z.Meng and Y.Tong, Facial Expression Recognition via a Boosted Deep Belief Network, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[11] S.Albanie, A.Nagrani, A.Vedaldi and A.Zisserman, Emotion Recognition in Speech using Cross-Modal Transfer in the Wild, ACM Multimedia 2018.

[12] K.Weiss, T.M.Khoshgoftaar and D.D.Wang, A survey of transfer learning, Journal of Big Data, Volume 4, Number 1, 2017.

[13] S.J.Pan and Q.Yang, A Survey on Transfer Learning, IEEE Transactions on Knowledge and Data Engineering, Volume 22, Number 10, Pages: 1345–1359, 2009.

[14] P.Ekman, Expression and the nature of emotion, Approaches to emotion. Hillsdale, New Jersey: Lawrence Erlbaum Associates, Pages: 319–344, 1984.

[15] G.Csurka, Domain Adaptation for Visual Applications: A Comprehensive Survey, Domain Adaptation in Computer Vision Applications, Pages: 1–35, 2017.

[16] M.Wang, W.Deng, Deep Visual Domain Adaptation: A Survey, Neurocomputing 2018.

[17] Y.Ganin, E.Ustinova, H.Ajakan, P.Germain, H.Larochelle, F.Laviolette, M.Marchand and V.Lempitsky, Domain-Adversarial Training of Neural Networks, Journal of Machine Learning Research, Volume 17, Pages: 1–35, 2016.

[18] S.Purushotham, W.Carvalho, T.Nilanon and Y.Liu, Variational Recurrent Adversarial Deep Domain Adaptation, International Conference on Learning Representations (ICLR), 2017.

[19] A.E.W.Johnson, T.J.Pollard, L.Shen, L.Lehman, M.Feng, M.Ghassemi, B.Moody, P.Szolovits, L.A.Celi, and R.G.Mark, Mimic-iii, a freely accessible critical care database. Scientific Data, Volume 3, Number 160035, 2016.

[20] X.Zeng, G.Yu, Y.Lu, L.Tan, X.Wu, S.Shi, H.Duan, Q.Shu and H.Li, PIC, a paediatric-specific intensive care database, Scientific Data, volume 7, Number 14, 2020.

[21] J.Bromley, I.Guyon, Y.LeCun, E.Sackinger and R.Shah, Signature verification using a Siamese time delay neural network, Advances in Neural Information Processing Systems, Volume 6, Pages: 737–744, 1994.

[22] M. Long, Y. Cao, J. Wang, and M. Jordan, Learning transferable features with deep adaptation networks, in International Conference on Machine Learning (ICML), 2015.

[23] E.Tzeng, J.Hoffman, N.Zhang, K.Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance", on arXiv:1412.3474, 2014.

[24] B.Sun and K.Saenko, "Deep coral: Correlation alignment for deep domain adaptation", in ECCV Workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV), 2016.

[25] B.Sun and K.Saenko, Deep coral: Correlation alignment for deep domain adaptation, in ECCV Workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV), 2016.

[26] B.Sun, J.Feng, and K.Saenko, Return of frustratingly easy domain adaptation, in AAAI Conference on Artificial Intelligence (AAAI), 2016

[27] J.T.Zhou, S.J.Pan, I.W.Zang and Y.Yan, Hybrid Heterogeneous Transfer Learning through Deep Learning, Proceedings of the Twenty-Eighth Conference on Artificial Intelligence (AAAI), 2014.

[28] M.Chen, Z.Xu and K.Q.Weinberger, Marginalized Denoising Autoencoders for Domain Adaptation, n Proceedings of the 29-th International Conference on Machine Learning, Edinburgh, Scotland, UK, 2012.

[29] H.Daume, Frustratingly Easy Domain Adaptation, Annual Meeting of the Association for Computational Linguistics, 2007.

[30] R.Raina, A.Battle, H.Lee, B.Packer and A.Y.Ng, Self-Taught Learning: Transfer Learning from Unlabeled Data, Proceedings of the 24^{th} International Conference on Machine Learning, 2007.

[31] M.Oquab, L.Bottou, I.Laptev and J.Sivic, Learning and Transferring Mid-Level Image Representations using Convoltional Neural Networks, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[32] L.Qin, Z.Zhang, H.Zhao, Z.Hu and E.P.Xing, Adversarial Connective-exploiting Networks for Implicit Discource Relation Classification, Conference: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017.

[33] F.Qi, X.Yang and C.Xu, A Unified Framework for Multimodal Domain Adaptation, Proceedings of the 26th ACM international conference on Multimedia, 2018,

[34] K.Bousmalis, G.Trigeorgis, N.Silberman, N.Krishnan and E.Dumitru, Domain Separation Networks, Advances in Neural Information Processing Systems 29 (NIPS), 2016.

[35] D.Das. C.S.G.Lee, Sample-to-sample correspondence for unsupervised domain adaptation, Journal for Engineering Applications of Artificial Intelligence, Volume 73, Pages: 80–91, 2018.

[36] D.Das. C.S.G.Lee, Unsupervised domain adaptation using regularized hyper-graph matching, International Conference on Image Processing, 2018.

[37] N.Courty, R.Flamary, D.Tuia and A.Rakotomamonjy, Optimal Transport for Domain Adaptation, IEEE Transactions of Pattern Analysis and Machine Intelligence, 2016.

[38] D.Das and S.G.Lee, Graph Matching and Pseudo-Label Guided Deep Unsupervised Domain Adaptation, Graph Matching and Pseudo-Label Guided Deep Unsupervised Domain Adaptation, 2018.

[39] L.Duan, D.Xu and I.W.Tsang, Learning with Augmented Features for Heterogeneous Domain Adaptation, Proceedings of the 29^{th} International Conference of Machine Learning, Edinburgh, Scotland, UK, 2012.

[40] C.Cortes and V.Vapnik, Support-vector networks, Machine Learning, 20, Pages: 273–297, 1995.

[41] X.Shi, Q.Liu, W.Fan, Y.Philip and R.Zhu, Transfer learning on heterogenous feature spaces via spectral transformation, Proceedings IEEE International Conference on Data Mining, ICDM, Pages: 1049–1054, 2010.

[42] B.Kulis, K.Saenko, and T.Darrell, What You Saw is Not What You Get: Domain Adaptation Using Asymmetric Kernel Transforms, Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[43] B.Kulis, K.Saenko and T.Darrell, Learning with Augmented Features for Heterogeneous Domain Adaptation, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[44] C.Wang and S.Mahadevan, Heterogeneous Domain Adaptation Using Manifold Alignment, Proceedings of the Twenty Second International Joint Conference on Artificial Intelligence Volume Volume Two, IJCAI 11, 2011.

[45] G.Csurka, B.Chidlovskii, and F.Perronnin, Domain adaptation with a domain specific class means classifier, in ECCV Workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV), 2014.

[46] T.Tommasi and B.Caputo, Frustratingly easy NBNN domain adaptation, in IEEE International Conference on Computer Vision (ICCV), 2013.

[47] N.Xie, G.Ras, M.V.Gerven and D.Doran, Explainable Deep Learning: A Field Guide for the Uninitiated, on arXiv, 2020.

[48] J.Yang, R.Yan, and A.G.Hauptmann, Cross-domain video concept detection using adaptive SVMs, in ACM Multimedia, 2007.

[49] L.Duan, I.W.Tsang, D.Xu, and S.J.Maybank, Domain transfer SVM for video concept detection, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[50] L.Duan, D.Xu, and S.F.Chang, Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[51] H.Wang, A.Klaser, C.Schmid, C.L.Liu, Action recognition by dense trajectories. In: IEEE 2011 conference on computer vision and pattern recognition (CVPR), 2011.

[52] I.Lopatovska and I.Arapakis, Theories, methods and current research on emotions in library and information science, information retrieval and human–computer interaction, Information Processing and Management, Volume 47, Pages: 575-–592, 2011.

[53] B.D.Gelder and J.Vroomen, The perception of emotions by ear and eye, Cognition and Emotion, Volume 14, Number Issue 3, Pages: 289–311, 2000.

[54] P.Kleinginna and A.Kleinginna, A categorized list of motivation definitions, with a suggestion for a consensual definition, Motivation and Emotion, Volume 5, Number 3, Pages: 263–291, 1981.

[55] David Hume, A Treatise of Human Nature: Being an Attempt to Introduce the Experimental Method of Reasoning into Moral Subjects and Dialogues Concerning Natural Religion, 1739–40.

[56] Charles Darwin, The Expression of the Emotions in Man and Animals, London: J.Murray, 1872.

[57] S.D'Mello, A.Kappas, and J.Gratch, The Affective Computing Approach to Affect Measurement, Emotion Review, Volume 10, Number 2, Pages: 174–183, 2018.

[58] A.Kappas, Social regulation of emotion: messy layers, Frontiers in Psychology, Volume 4, Number 51, 2013.

[59] R.W.Picard, Affective computing, M.I.T Media Laboratory Perceptual Computing Section Technical Report Number 321, 1997.

[60] D.T.Correia, S.Saraiva and J.G.Marques, Jaspers' Phenomenology, Folia Medica, Volume 60, Number 3, Pages: 373—380, 2018.

[61] S.Folkman, R.S.Lazarus, R.J.Gruen, and A.DeLongis, Appraisal, coping, health status, and psychological symptoms, Journal of Personality and Social Psychology, Volume 50, Number 3, Pages: 571—57, 1986.

[62] R.S.Lazarus, Thoughts on the relations between emotion and cognition, American Psychologist, Volume 37, Number 9, Pages: 1019–1024, 1982.

[63] N.H.Frijda, Varieties of affect: Emotions and episodes, moods, and sentiments. The nature of emotion, New York: Oxford University Press, Pages: 197–202, 1994.

[64] S.S.Tomkins, Affect theory. Approaches to emotion, Hillsdale, New Jersey: Lawrence Erlbaum Associates, Pages: 163-–197, 1984.

[65] R.Plutchik, A general psychoevolutionary theory of emotion, Theories of emotion, New York Academic, Volume 1, Pages: 3–33, 1980.

[66] P.Ekman, Expression and the nature of emotion, Approaches to emotion. Hillsdale, New Jersey: Lawrence Erlbaum Associates, Pages: 319–344, 1984.

[67] P.Ekman, Basic emotions, The handbook of cognition and emotion. UK: John Wiley and Sons, Ltd., Pages: 45–60, 1999.

[68] C.Darwin, The Expression of the Emotions in Man and Animals, Kessinger Publishing, 2005.

[69] R.Plutchik, The Nature of Emotions: Clinical Implications, Emotions and Psychopathology, Pages: 1–20, 1988.

[70] J.A.Russell, Is there universal recognition of emotion from facial expression?, Psychological Bulletin, Volume 115, Number 1, Pages: 102-141, 1994.

[71] M.Csikszentmihalyi, *"Beyond boredom and anxiety"*, The Jossey-Bass behavioral science series, San Francisco: Jossey-Bass Publishers, 1975.

[72] J.Nakamura and M.Csikszentmihalyi, *"The concept of flow"* Emotion focused approaches, 2004.

[73] M.Csikszentmihalyi, *"Creativity, Flow and the psychology of discovery and invention"* Emotion focused approaches, 2004.

[74] B.Cowley, D.Charles, M.M.Black and R.J.Hickey, Toward an understanding of flow in video games, Computers in Entertainment, Volume 6, pages: 1–20, 2008.

[75] L.Michailidis, E.B.Ballester and X.He, Flow and Immersion in Video Games: The Aftermath of a Conceptual Challenge, Frontiers in Psychology, Volume 9, 2018.

[76] K.Kiili, S.Freitas, S.Arnab and T.Lainema, The Design Principles for Flow Experience in Educational Games, Volume 15, pages: 78-91, 2012.

[77] J.A.Russell, Measures of emotion. In R.Plutchik & H.Kellerman (Eds.), Emotion: Theory, research, and experience, Volume 4, 1989.

[78] A.Mehrabian, Framework for a comprehensive description and measurement of emotional states. Genetic, Social, and General Psychology Monographs, Volume 121, Number 3, pages: 339-–361, 1995.

[79] R.W.Picard, Building hal: Computers that sense, recognize, and respond to human emotion, In Society of photo-optical instrumentation engineers. Human vision and electronic imaging VI, Volume 4299, Pages: 518-523, 2001.

[80] R.W.Picard, Affective computing: Challenges, International Journal of Human-Computer Studies, Volume 59, Number 1-2, Pages 55–64, 2003.

[81] H.G.Wallbott and K.R.Scherer, Cues and Channels in Emotion Recognition, Journal of Personality and Social Psychology, Volume 51, Number 4, Pages: 690–699, 1986.

[82] C.A.Corneanu, M.Oliu, J.F.Cohn and S.Escalera, E Survey on RGB, 3D, Thermal, and Multimodal Approaches for Facial Expression Recognition: History, Trends, and Affect related Applications, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016.

[83] C.H.Wu, J.C.Lin and W.L.Wei, Survey on audiovisual emotion recognition: databases, features, and data fusion strategies, Transactions on Signal and Information Processing, Volume 3, Number 12, Pages: 1–18, 2016.

[84] D.G.Lowe, Object recognition from local scale-invariant features, Proceedings of the International Conference on Computer Vision (ICCV), 1999.

[85] C.Shan, S.Gong, and P.W.McOwan, “Facial expression recognition based on local binary patterns: A comprehensive study”, Image and Vision Computing, Volume 27, Number 6, Pages: 803—816, 2009.

[86] H.Bay, T.Tuytelaars, L.V.Gool, SURF: Speeded Up Robust Features, European Conference on Computer Vision (ECCV), 2006.

[87] T.Ojala, M.Pietikainen and D.Harwood, A comparative study of texture measures with classification based on feature distributions, Volume 29, Number 1, Pages: 51–59, 1996.

[88] Y.Huang, F.Chen, S.Lv and X.Wang, Facial Expression Recognition: A Survey, Symmetry Volume 11, Number 1189, 2019.

[89] H.Zhao, N.Ye and R.Wang, A Survey on Automatic Emotion Recognition Using Audio Big Data and Deep Learning Architectures, 4th IEEE International Conference on Big Data Security on Cloud, 2018.

[90] C.H.Hjortsjo, Man’s face and mimic language, Studentlitteratur, Lund, Sweden 1969.

[91] P.Ekman and W.Friesen, Facial Action Coding System: A Technique for the Measurement of Facial Movement, Palo Alto: Consulting Psychologists Press, 1978.

[92] Y.LeCun, Y.Bengio and G.Hinton, Deep learning, Nature, Volume 521, pages: 436-444, 2015.

[93] Y.LeCun, P.Haffner, L.Bottou and Y.Bengio, Object Recognition with Gradient-Based Learning, Shape, Contour and Grouping in Computer Vision, pages: 319-345, 1999.

[94] S.Hochreiter and J.A.Schmidhuber, Long Short-Term Memory, Neural Computation, Volume 9, Number 8, pages: 1735–1780, 1997,

[95] I.J.Goodfellow, J.P.-Abadie, M.Mirza, B.Xu, D.W.Farley, S.Ozair, A.Courville and Y.Bengio, Generative Adversarial Networks, 27th conference on Advances in Neural Information Processing Systems (NIPS), 2014.

[96] D.H.Ballard, Modular Learning in Neural Networks, AAAI, 1987.

[97] P.Smolensky, Information processing in dynamical systems: Foundations of harmony theory, Technical report, 1986.

[98] G.B.Huang and C.K.Siew, Extreme learning machine: A new learning scheme of feedforward neural networks, Conference: Neural Networks, 2004.

[99] R.Jenke, A.Peer and M.Buss, Feature Extraction and Selection for Emotion Recognition from EEG, IEEE Transactions on Affective Computing, Volume 5, Number 3, 2014.

[100] H.Zacharatos, C.Gatzoulis and Y.L.Chrysanthou, Automatic Emotion Recognition Based on Body Movement Analysis: A Survey, IEEE Transactions of Computer Graphics and Applications, Volume 34, Number 6, 2014.

[101] D.Ghimire and J.Lee, Geometric feature-based facial expression recognition in image sequences using multi-class AdaBoost and support vector machines, Sensors, Volume 13, Number 6, Pages: 7714-7734, 2013.

[102] S.L.Happy, A.George and A.Routray, A real time facial expression classification system using local binary patterns, In Proceedings of the 4th International Conference on Intelligent Human Computer Interaction, 2012.

[103] D.Ghimize, S.Jeong, J.Lee and S.H.Park, Facial expression recognition based on local region specific features and support vector machines, Multimedia Tools and Applications, Volume 76, Pages: 7803–7821, 2017.

[104] Y.Yacoob and L.S.Davis, Recognizing human facial expressions from long image sequences using optical flow, IEEE Transactions Pattern Analysis Machine Intelligence, Volume 18, Pages: 636—642, 1996.

[105] J.F.Cohn, A.J.Zlochower, J.J.Lien and T.Kanade, Feature-point tracking by optical flow discriminates subtle differences in facial expression. In Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998.

[106] Y.Hu, Z.Zeng, L.Yin, X.Wei, X.Zhou and T.S.Huang, Multi-view facial expression recognition, 8th IEEE International Conference on Automatic Face & Gesture Recognition, 2008.

[107] P.Lucey, J.F.Cohn, T.Kanade, J.Saragih, Z.Ambadar and I.Matthews, The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, 2010.

[108] M.Lyons, S.Akamatsu, M.Kamachi and J.Gyoba, Coding facial expressions with gabor wavelets, In Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998.

[109] S.Wang, Z.Liu, S.Lv, Y.Lv, G.Wu, P.Peng, F.Chen and X.Wang, A natural visible and infrared facial expression database for expression recognition and emotion inference. IEEE Transaction Multimedia, Volume 12, Pages: 682–691, 2010.

[110] R.Gross, I.Matthews, J.Cohn, T.Kanade and S.Baker, Multi-PIE. Image Visual Computation, Volume 28, Pages: 807–813, 2010.

[111] N.Xie, G.Ras, M.V.Gerven and D.Doran, Explainable Deep Learning: A Field Guide for the Uninitiated, arXiv, 2020.

[112] A.B.Arrieta, N.D.Rodríguez, J.D.Ser, A.Bennetot, S.Tabik, A.Barbado, S.Garcia, S.G.Lopez, D.Molina, R.Benjamins, R.Chatila and F.Herrera, Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Volume 58, Pages: 82–115, 2020.

[113] B.J.Abbaschian, D.S.Sosa and A.Elmaghraby, Deep Learning Techniques for Speech Emotion Recognition, from Databases to Models, Sensors, Volume 21, Number 4, 2021.

[114] R.Breuer and R.Kimmer, A deep learning perspective on the origin of facial expressions, 2017.

[115] H.Jung, S.Lee, J.Yim, S.Park and J.Kim, Joint fine-tuning in deep neural networks for facial expression recognition, In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 2015.

[116] K.Zhao, W.S.Chu and H.Zhang, Deep region and multi-label learning for facial action unit detection, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016.

[117] R.Breuer and R.Kimmel, A deep learning perspective on the origin of facial expressions, arXiv 2017.

[118] M.Liu, S.Li, S.Shan, R.Wang and X.Chen, Deeply learning deformable facial action parts model for dynamic expression analysis. In Asian Conference on Computer Vision, 2014.

[119] A.M.Badshah, J.Ahmad, N.Rahim and S.W.Baik, Speech Emotion Recognition from Spectrograms with Deep Convolutional Neural Network, International Conference on Platform Technology and Service, 2017.

[120] P.Yenigalla, A.Kumar, S.Tripathi, C.Singh, S.Karl and J.Vepa, Speech Emotion Recognition Using Spectrogram and Phoneme Embedding, Interspeech, 2018.

[121] S.Zhang, S.Zhang, T.Huang and W.Gao, Multimodal deep convolutional neural network for audio-visual emotion recognition, Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, 2016.

[122] S.A.Bargal, E.Barsoum, C.C.Ferrerand and C.Zhang, Emotion recognition in the wild from videos using images, Proc. of the 19th ACM International Conference on Multimodal Interaction, 2016.

[123] A.Dhall, R.Goecke, J.Joshi, J.Hoey and T.Gedeon, EmotiW 2016: Video and Group Level Emotion Recognition Challenges, Procs. of the 18th ACM International Conference on Multimodal Interaction, ACM, Pages: 427–432, 2016.

[124] A.Dhall, R.Goecke, J.Joshi, J.Hoey and T.Gedeon, EmotiW 2016: Video and Group Level Emotion Recognition Challenges, Proceedings of the 18th ACM International Conference on Multimodal Interaction, ACM, Pages: 427–432, 2016.

[125] O.Martin, I.Kotsia, B.Macq and I.Pitas, The eNTERFACE'05 Audio-Visual Emotion Database, 22nd International Conference on Data Engineering Workshops (ICDEW'06), 2006.

[126] S.Zhalehpour, O.Onder, Z.Akhtar, Z.Akhtar and C.Erdem, BAUM-1: A Spontaneous Audio-Visual Face Database of Affective and Mental States, IEEE Transactions on Affective Computing, Volume 8, Number 3, Pages: 300–313, 2016.

[127] S.E.Kahou, C.Pal, X.Bouthillier, P.Froumenty, R.Memisevic, P.Vincent, A.Courville and Y.Bengio, Combining Modality Specific Deep Neural Networks for Emotion Recognition in Video, Proceedings of the 15th ACM International Conference on Multimodal Interaction, ACM, Pages: 543–550, 2013.

[128] X.Ouyang, S.Kawaai, E.Goh, S.Shen, W.Ding, H.Ming and D.Y.Hang, Audio-visual emotion recognition using deep transfer learning and multiple temporal models, Proceedings of the 19th ACM International Conference on Multimodal Interaction ICMI, 2017.

[129] S.Zhang, S.Zhang, T.Huang and W.Gao, Multimodal deep convolutional neural network for audio-visual emotion recognition, Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, 2016.

[130] E.Ghaleb, M.Popa and S.Asteriadis, Multimodal and Temporal Perception of Audio-visual Cues for Emotion Recognition, International Conference on Affective Computing & Intelligent Interaction (ACII), 2019.

[131] S.Zhalehpour, Z.Akhtar and C.E.Erdem, Multimodal emotion recognition based on peak frame selection from video, Signal, Image and Video Processing, Volume 10, Number 5, Pages: 827-834, 2016.

[132] A.Dhall, R.Goecke, S.Lucey and T.Gedeon, Collecting large, richly annotated facial expression datasets from movies, IEEE MultiMedia, Volume 19, Number 3, Pages: 34–41, 2012.

[133] F.Wang and J.Sun, Survey on distance metric learning and dimensionality reduction in data mining, Transactions on Data Mining and Knowledge Discovery, Volume 29, Number 2, 2014.

[134] R.Tibshirani, T.Hastie, B.Narasimhan and G.Chu, Diagnosis of multiple cancer types by shrunken centroids of gene expression, Proceedings of the National Academy of Sciences, 2002.

[135] B.Kulis, Metric Learning: A Survey, Transactions on Foundations and Trends in Machine Learning, Volume 5, Number 4, Pages: 287–364, 2013.

[136] E.Y.Liu, Z.Guo, X.Zhang, V.Jojic and W.Wang, Metric Learning from Relative Comparisons by Minimizing Squared Residual, International Conference on Data Mining, Sheraton Brussels, Belgium, 2012.

[137] B.Geng, D.Tao and C.Xu, DAML: Domain Adaptation Metric Learning, IEEE Transactions on Image Processing, Voluem 20, Number 10, Pages: 2980 - 2989, 2011.

[138] K.M.Borgwardt, A.Gretton, M.J.Rasch, H.P.Kriegel, B.Scholkopf and A.J.Smola, Integrating structured biological data by Kernel Maximum Mean Discrepancy, Transactions on Bioinformatics, Volume 22, Number 14, Pages: 49—57, 2006.

[139] J.Lu, J.Hu, and J.Zhou, Deep Metric Learning for Visual Understanding, IEEE Signal Processing Magazine, Volume 34, Number 6, Pages: 76–84, 2017.

[140] J.Hu, J.Lu and Y.P.Tan, Discriminative Deep Metric Learning for Face Verification in the Wild, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014

[141] Y.Taigman, M.Yang, M.Ranzato, and L.Wolf, Deepface: Closing the gap to human-level performance in face verification, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[142] G.B.Huang, M.Mattar, H.Lee and E.Learned-Miller, Learning to Align from Scratch, NeurIPS, 2012.

[143] L.Wolf, T.Hassner and I.Maoz, Face recognition in unconstrained videos with matched background similarity, Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[144] E.Hoffer and N.Ailon, Deep metric learning using triplet network, in Proceedings Similarity-Based Pattern Recognition, Third International Workshop, 2015.

[145] F.Schroff, D.Kalenichenko, and J.Philbin, Facenet: A unified embedding for face recognition and clustering, Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[146] I.J.Goodfellow, J.P.-Abadie, M.Mirza, B.Xu, D.W.Farley, S.Ozair, A.Courville and Y.Bengio, Generative Adversarial Networks, 27th conference on Advances in Neural Information Processing Systems (NIPS), 2014.

[147] A.Radford, L.Metz and S.Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, International Conference on Learning Representations (ICLR), 2016.

[148] M.Mirza and S.Osindero, Conditional Generative Adversarial Nets, Computing Research Repository (CoRR), 2014.

[149] O.Ronneberger, P.Fischer and T.Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015.

[150] P.Isola, J.Yan, Z.Tinghui, Z.Alexei and A.Efros, Image-to-Image Translation with Conditional Adversarial Networks, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[151] G.Shafer and V.Vovk, A Tutorial on Conformal Prediction, The Journal of Machine Learning Research archive Volume 9, Pages 371-421, 2008.

[152] J.Zhu, T.Park, P.Isola and A.A.Efros, Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, IEEE International Conference on Computer Vision (ICCV), 2017.

[153] J.Zhu, T.Park, P.Isola and A.A.Efros, Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, IEEE International Conference on Computer Vision (ICCV), 2017.

[154] T.Kim, M.Cha, H.Kim, J.K.Lee and J.Kim, Learning to Discover Cross-Domain Relations with Generative Adversarial Networks, International Conference on Machine Learning (ICML), 2017.

[155] J.Hoffman, E.Tzeng, T.Park, J.Y.Zhu, P.Isola, K.Saenko, A.Efros and T.Darrell, CyCADA: Cycle-Consistent Adversarial Domain Adaptation, International Conference on Learning Representations (ICLR), 2018.

[156] Z.Hu, Z.Yang, R.Salakhutdinov, E.P.Xing, On Unifying Deep Generative Models, International Conference on Learning Representations (ICLR), 2018.

[157] L.Chen, S.Srivastava, Z.Duan and C.Xu, Deep Cross-Modal Audio-Visual Generation19, Thematic Workshops '17 Proceedings of the on Thematic Workshops of ACM Multimedia, 2017.

[158] K.Vougioukas, S.Petridis and M.Pantic, Realistic Speech-Driven Facial Animation with GANs, International Journal of Computer Vision, Pages: 1–19, 2019.

[159] M.Arjovsky, S.Chintala and L.Bottou, Wasserstein GAN, arXiv, 2017.

[160] I.Gulrajani, F.Ahmed, M.Arjovsky, V.Dumoulin and A.Courville, Improved Training of Wasserstein GANs, NIPS, 2017.

[161] G.J.Qi, Loss-Sensitive Generative Adversarial Networks on Lipschitz Densities, International Journal of Computer Vision (2019).

[162] H.Alqahtani, M.K.Thorne and G.Kumar, Applications of Generative Adversarial Networks (GANs): An Updated Review, Archives of Computational Methods in Engineering, 2019.

[163] W.Hao, Z.Zhang and H.Guan, CMCGAN: A Uniform Framework for Cross-Modal Visual-Audio Mutual Generation, Association for the Advancement of Artificial Intelligence (AAAI), 2018.

[164] J.Foote, Content-based retrieval of music and audio, In Multimedia Storage and Archiving Systems II, 1997.

[165] E.Sejdic, I.Djurovic and J.Jiang, "Time-frequency feature representation using energy concentration: An overview of recent advances", Digital Signal Processing, Volume 19, Number 1, Pages: 153–183, 2009.

[166] L.Metz, B.Poole, D.Pfau and J.S.Dickstein, Unrolled Generative Adversarial Networks, International Conference on Learning Representations (ICLR), 2017.

[167] J.S.Chung and A.Zisserman, Out of time: automated lip sync in the wild, Workshop on Multi-view Lip-reading, ACCV, 2016.

[168] B.Korbar, D.Tran and L.Torresani, Cooperative Learning of Audio and Video Models from Self-Supervised Synchronization,Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS), 2018.

[169] A.Dhall, R.Goecke, S.Lucey, and T.Gedeon, Collecting large, richly annotated facial expression datasets from movies, IEEE Transactions of MultiMedia, Volume 19, Number 3, 2012.

[170] H.Cao, D.G.Cooper, M.K.Keutmann, R.C.Gur, A.Nenkova and R.Verma, CREMA-D: Crowd-Sourced Emotional Multimodal Actors Dataset, IEEE Transactions on Affective Computing, Volume 5, Number 4, Pages: 377–390, 2014.

[171] S.R.Livingstone and F.A.Russo, The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English, Volume 13, Number 5, Pages: 1—35, 2018.

[172] A.Nagrani, J.S.Chung and A.Zisserman, VoxCeleb: a large-scale speaker identification dataset, Interspeech, 2018.

[173] S.Albanie, A.Nagrani, A.Vedaldi and A.Zisserman, Emotion Recognition in Speech using Cross-Modal Transfer in the Wild, ACM Multimedia, 2018.

2

METRIC LEARNING FOR DOMAIN ADAPTATION

The research presented in this chapter is supported by the following publication:

- *C.Athanasiadis, E.Hortal, and S.Asteriadis, Bridging face and sound modalities through Domain Adaptation Metric Learning, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), 2019.*

This chapter introduces the first approach developed in order to accommodate the task of audio-visual domain adaptation. This approach was the first one in chronological order that was attempted during the current dissertation. Therefore, our scope was to take a step for accommodating cross-modal domain adaptation within the emotion analysis context. The first attempt is based on a basic well-known technique to get acquainted with the problem to be tackled. On this basis, an intuitive framework based on Distance Metric Learning is introduced. The main research question that is posed in this chapter is the following: *How can a metric learning framework be deployed to model audio-visual cross modal relationships?*.

In said manner, this chapter introduces a technique based on Distance Metric Learning (DML) and the pre-processing steps applied for the face and audio related datasets. The main idea is to exploit robust modalities such as face (from which we can access dense datasets) to improve the classification performance of weak modalities such as audio (from which we can only access sparsely available datasets). In this respect, facial features from video are used in order to represent the dense dataset (robust modality), while features from audio are utilized to represent the sparse datasets (weak modality). We perform the whole approach in distinct parameterized modules that can be easily tuned and function separately. In such a way, we are able to understand and explain easier the functionality of each of these steps.

The proposed framework is implemented as five distinct modules:

- Firstly, a *pre-processing step* for both modalities is performed. In this manner, face detection, alignment and cropping on images is applied. Moreover, a normalization scheme is conducted for the audio signal.
- Secondly, temporal *feature extraction* is performed for both modalities.
- Thirdly, a *feature selection* module is carried out, where only the most significant features of the datasets are chosen and kept.
- Subsequently, a *clustering approach* is introduced with the purpose of transforming both source and target domains into a common dimensionality.
- Finally, “Distance Metric Learning” (DML) approaches are implemented with the purpose of establishing a bridge between the two domains.

All these steps are carefully analysed and several techniques are evaluated. More details about the conducted work can be found in Section 2.1. The whole implemented approach is summarized and visualized in Figure 2.1. This image renders the whole flow of the approach for introducing the derived metric.

Having established a transformation between the two modalities, the performance of the introduced framework is tested using the Support Vector Machine (SVM) classification [23] for the target task (emotion recognition from audio). This approach is described in detail in Appendix 4. In Figure 2.1, the way that the projected features from the aforesaid procedure are evaluated, using a multi-class SVM classification scheme, is illustrated (the visualized example consists of four different classes).

Over and above, an evaluation tool is utilized to measure the distance between the transformed distributions of audio and face modalities and compare them with the distance between the distributions of the initial datasets (which are derived from raw features).

The main contributions that are introduced in the current chapter are three-fold:

1. Up to our knowledge, this is the first study that performs “domain adaptation” for the domain of emotion recognition using two heterogeneous modalities.
2. This is the first time that an audio-visual projection is performed with respect to emotion recognition.
3. This is the first time that the current DML framework is introduced and tested in the task of emotion recognition.

The structure of the remainder of this chapter is as follows: Section 2.1 describes the Heterogeneous Domain Adaptation (HDA) technique that is applied to AFEW, CREMA-D and RAVDESS datasets, while in Section 2.2 the experimental results are analyzed. Finally, Section 2.3 contains the conclusions and the extracted deductions that are analysed to proceed further with the improvement of our domain adaptation approach.

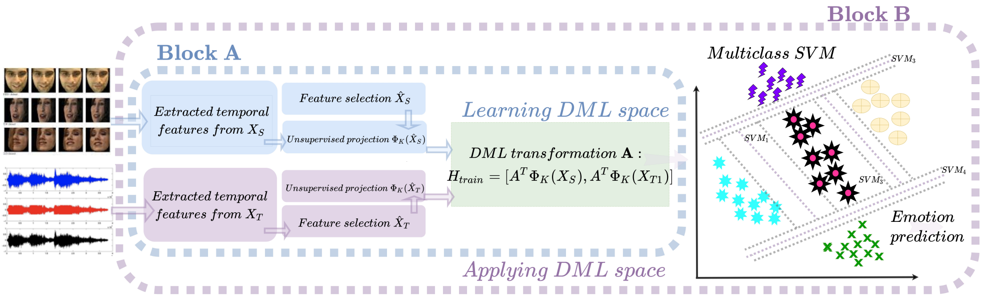
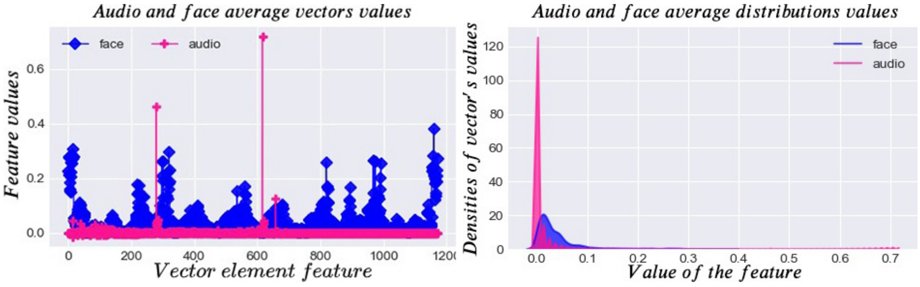

Block A
Block B
Extracted temporal features from X_S
Feature selection $\hat{X}_S$
Unsupervised projection $\Phi_K(\hat{X}_S)$
Learning DML space
DML transformation $\mathbf{A}$:
$H_{train} = [A^T\Phi_K(X_S), A^T\Phi_K(X_{T1})]$
Extracted temporal features from X_T
Unsupervised projection $\Phi_K(\hat{X}_T)$
Feature selection $\hat{X}_T$
Applying DML space
Multiclass SVM
SVM_1
SVM_2
SVM_3
SVM_4
Emotion prediction

Figure 2.1: Visualization of the whole DML framework. X_S corresponds to the face modality while X_T to audio modality. “Block A” contains all the necessary step to learn a DML projection matrix **A**. Having learned this mapping, in “Block B”, the classification step is employed, where this projection **A** is applied to input domains (after applying again the *feature extraction*, *selection* and the *unsupervised step* steps from “Block A” using the same parameters). Then, multi-class SVM is applied in a common set with projected audio and facial features.



(a) Mean vector average values for face and audio modalities.

(b) Distribution densities for the average vector's values.

Figure 2.2: The feature distributions for the mean face and audio vectors for the AFEW dataset after feature selection (described in Section 2.1.2). For each average vector the values for each dimension are represented. Note that this analysis conducted with all samples from the happy class.

2.1. DISTANCE METRIC LEARNING BASED APPROACH

In the current section, the proposed affect-augmented framework, which performs audio emotion recognition through leveraging annotated data related to face modality is analyzed. The whole framework is visualized in Figure 2.1. Both modalities are governed by different distributions, however, both could potentially be used for performing emotion recognition either by eliciting facial expressions or by extracting emotions from audio features. These two modalities cannot be easily bridged though, because of the inherent differences in their distributions. Figure 2.2 illustrates the difference in the distribution of the average vectors for each modality. It is clear that the distributions of the mean face and audio data features can be vastly different from each other. Moreover, their initial feature dimensionality is also different and it depends on the initial feature extraction strategy.

Therefore, to exploit the source domain dataset for the target classification task, a transformation that will bridge both domains needs to be established. Firstly, for the needs of current research, the face modality is defined as source domain X_S (representing the pixels of the images) while the audio modality is defined as target domain X_T (sampled value from the amplitude of the raw audio signal). This is done due to the lower availability of emotion-related datasets from audio. The proposed framework is divided into five distinct modules:

- As has been explained already in Section 1.2.3, certain pre-processing steps for each domain need to be performed for face and audio before proceeding with the domain adaptation approach. For the face modality, face detection, cropping, and alignment are performed. For the audio modality, segments of 500ms from the beginning and the end of the audio are removed. These steps are performed in order to extract only the necessary information (the face bounding box or the audio clip without silence parts) from both modalities regarding emotion recognition.

- Secondly, the feature extraction module is built. This is done to extract useful information from both modalities that relates to the emotions that govern both domains, and which can help in obtaining higher performance regarding the face and audio emotion recognition.

- Thirdly, the feature selection module is performed. In this dissertation, we define the extracted features for face and audio after this step $\hat{X_S}$ and $\hat{X_T}$ respectively. Finally, the most dominant features are kept from both modalities. In such a way, we discard redundant features that do not contribute or negatively impact the emotion recognition.

- Subsequently, the framework proceeds with the unsupervised learning module. In this step, a transformation (based on clustering) of the extracted features is performed. The primary task of this transformation is to bring both domains into a common dimensionality.

- Finally, the last module is the distance metric learning. In this case, a "new distance" is calculated to minimize the divergence between the two domains.

Briefly, this new distance regards with the learning of a distance $\hat{d}(x,y)$ between samples $\{x,y\}$, based on a dataset that contains pairs of samples that are considered to be close (similar) based on their class information and pairs of samples that are considered to be distant (dissimilar pairs). This information (relation between pairs of vectors) can be mathematically framed with the following equation:

$$C_{ij} = \begin{cases} 1 & \text{if } (x_i, x_j) \in S \\ -1 & \text{if } (x_i, x_j) \in D \end{cases} \tag{2.1}$$

where the sets S and D correspond to similar and dissimilar pairs of subjects. The generation of the distance $\hat{d}(x,y)$ is based on these constraints. The goal is to learn the matrix M in the following equation 2.2:

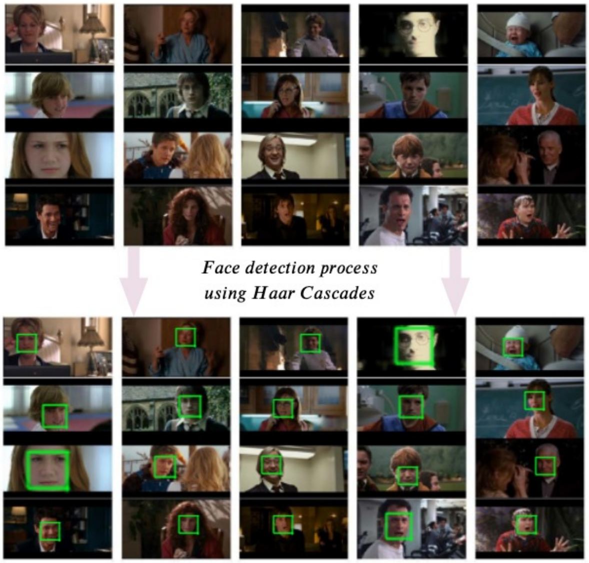


Figure 2.3: The way that face detection is conducted in AFEW dataset. The bounding box in green colour contains each time the detected face for each frame.

$$
\begin{aligned}
\hat{d}_m(x_i, x_j) &= ||(A^T x_i - A^T x_j)^2||_2^2 \\
&= (x_i - x_j)^T A A^T (x_i - x_j) \\
&= (x_i - x_j)^T M (x_i - x_j) \qquad (2.2)
\end{aligned}
$$

Where the goal is to learn matrix $M = A \cdot A^T \in \mathbb{S}_+^d$. M needs to be an $n \times n$ symmetric positive semi-definite (PSD) matrix [19][21], where n is the number of input features (both for face and audio domains after the clustering coding). The matrix A^T can be used to transform the input to the distance metric learning domain. More details can be found in Section 2.1.4.

In the following subsections, all the previously mentioned modules of our approach are analyzed in detail.

2.1.1. PRE-PROCESSING AND FEATURE EXTRACTION MODULES

In this sub-section, we first explain the pre-processing phase of facial images and how we obtain facial features from a video. Then, a discussion of a pre-processing step regarding audio is following. Finally, we present the low-level feature extraction methods implemented in our framework for both modalities.

FACIAL FEATURES

With respect to face modality, the first pre-processing module is the face detection process. To address the issue of face detection, the system implements Haar Cascade classifiers which are provided by OpenCV [1]. This algorithm is based on the extraction of

Haar features which, in principle, involves the convolution of the initial image with Haar Cascade filters [1] of varying sizes and orientations. Subsequently, a feature selection process that filters all calculated features and a classification step are performed by making use of AdaBoost algorithm [2]. Adaboost selects the best features and trains several classifiers by using them. This algorithm constructs a strong classifier, which is a linear combination of weighted simple weak classifiers [1]. Having as an input a single image, the resulted output is a bounding box with the position of the face within the image.

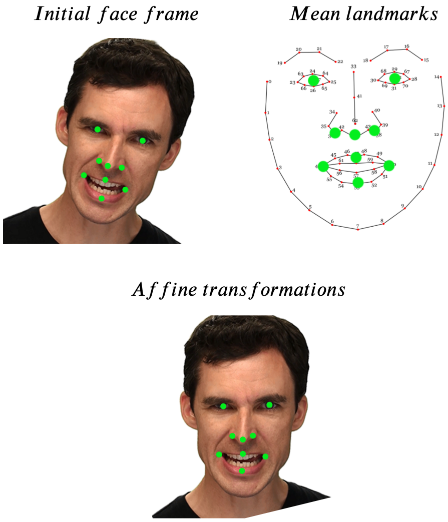
By using this algorithm, for every frame of a video clip, the system returns facial bounding boxes in the video sequence. Every video clip contains only one person in the AFEW, CREMA-D, and RAVDESS datasets. In this way, we are able to remove superfluous information regarding the background and visual information that does not (or even negatively) contribute to emotion recognition performance. An example of how this process can be applied in the AFEW dataset is visualized in Figure 2.3. For each frame, only the information within the green bounding box is employed in the subsequent frames.

The next pre-processing step of the framework is the face alignment process. This phase is essential for the system since the faces in the video frames can be found in different poses and angles which can affect the performance of an emotion recognition classifier in a negative way. With the purpose of performing facial alignment, the framework first should proceed with the face landmark detection. On this ground, the Supervised Descent Method (SDM) [3] approach is applied. The calculated SDM landmark points then are used and a similarity transformation is applied with the purpose of aligning faces with a fixed canonical frame based on eye center positions. Then, all the extracted faces from the video are cropped and re-sized to a fixed size of 224×224 pixels. Figure 2.4 renders an example of how the face alignment is utilized using a sample from the RAVDESS dataset.

As soon as the faces are aligned and cropped, the system proceeds with the temporal feature extraction phase. To that end, Volume Local Binary Patterns (VLBP) [4] are employed to extract features from the whole sequence of frames that correspond to a specific emotion. This approach is the expansion of the local binary patterns (LBP) but for video (3d-LBP). The main difference between LBP and VLBP is that the extracted histograms in the latter case is not just a spatial representation of the pixels but is extended to the spatio-temporal domain. For each video, the extracted dimensionality for the VLBP approach is $\mathbb{R}^{1018}$. This approach is performing LBP on 3d tensors of facial frames from the whole video. More details on this can be found in [4].

AUDIO FEATURES

Regarding the audio modality, the methodology described in [14] for extracting frequency-based features is applied in our work. In particular, our system is using the audio analysis openSMILE toolkit [10] for audio feature extraction. We are following the same strategy for extracting audio features as in [11]. This approach is utilized for extracting features from the audio that contain both the voice quality (represented by jitter, segments and Harmonics-to-Noise Ratio HNR [12]) and the prosodic characteristics of the speaker. The set of audio features that are extracted contains 1.941 features. This set is composed based on two categories of features, namely low-level descriptors (LLD), and functional features. LLD are features extracted by using audio-signal



Initial face frame

Mean landmarks

Affine transformations

Figure 2.4: An example of face alignment from the detected faces from the initial frame (top-left) to the aligned version (bottom). The top-left picture shows the initial image with the detected landmarks. The top-right shows the canonical aligned landmarks. The example is derived from the RAVDESS dataset [7].

processing methods such as Fourier transformation and by extracting features related to zero-crossing and pitch. The functionals are features extracted after processing the LLD features by employing statistical analysis features such as mean, variance, median, minimum, and maximum values. Secondly, Lasso regression analysis [13] is performed, which is a common tool for feature selection.

In particular, the final feature vector is composed of 25 energy and spectral related low-level descriptors (LLD) × 42, 6 voicing related LLD × 32 functionals, 25 delta coefficients of the energy/spectral LLD × 23 functionals, 6 delta coefficients of the voicing related LLD × 19 functionals, and 10 voiced/unvoiced durational features. The set of LLD covers a standard range of commonly used features in audio signal analysis and emotion recognition. In [10], a description of the employed LLD and functional features is provided.

2.1.2. FEATURE SELECTION

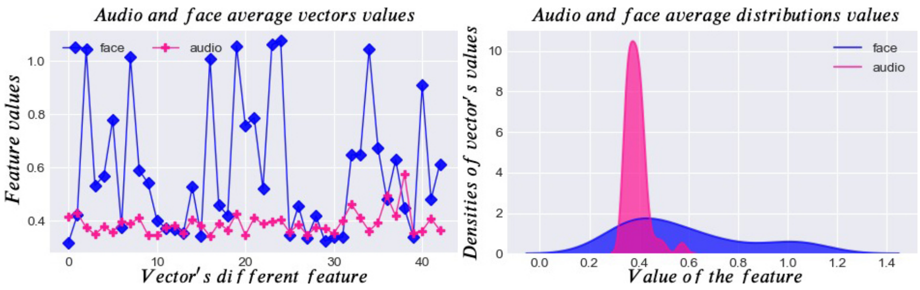
The second part of the approach is performed with the purpose of implementing a feature selection technique using the extracted features from both modalities. This step is implemented to keep only the features from both domains which are significant to the classification task in hand (emotion recognition) and discard less relevant features. Feature selection is performed for extracted facial and audio features separately. Three different ways to perform feature selection are tested. In particular:

- Firstly, an implementation solely based on the variance of the features is performed. This methodology can be assigned to the so-called *filter feature selection* family of techniques [9]. In this category, statistical tools are employed to determine the relationship between each input variable and the target variable. Then, these statistical tools are the basis to filter the variables (features) that will be used in the model. In our case, only the features from the source and target vectors with variance greater than a threshold are chosen. This threshold is calculated based on the classification performance (for emotion recognition) for both modalities after feature selection on a validation set. Building on the results obtained, it is found that, by keeping the 75% more significant dimensions, for both modalities, the classification performance remained practically unchanged. For instance, the dimensionality of both modalities (audio and face) after the implementation of variance-based feature selection is $\mathbb{R}^{600}$ and $\mathbb{R}^{379}$ respectively (for the AFEW dataset). As we already mentioned before, the initial dimension (after feature extraction module) for face modality is $x_S \in \mathbb{R}^{1018}$ with $x_S \in X_S$ while for audio modality is $x_T \in \mathbb{R}^{1941}$ with $x_T \in X_T$.

- Secondly, L1-based feature selection, using logistic regression to perform feature selection is tested. The whole approach is described in [15]. This approach belongs to the so-called *wrapper feature selection* family of techniques [9]. Wrapper methods measure the "utility" of features based on the classifier performance. On this ground, several models that are employing different subsets of input features are computed and the subgroup with the features that result in the best performance according to a prediction metric are chosen to be the final selected features. For instance, the dimensionality of both modalities (audio and face) after the implementation of L1-based feature selection is $\mathbb{R}^{621}$ and $\mathbb{R}^{358}$ respectively (for the AFEW dataset).

- Finally, tree-based estimators (random forests) are used to compute features' importance, which, in turn, are used with the purpose of discarding irrelevant features. This methodology belongs to the so-called *embedded feature selection* family of techniques [9]. Embedded feature selection methods combine both filter and wrapper methods. More specifically, they realize feature selection during the training of the model. That is why they are most often referred to with the term embedded methods. For instance, the dimensionality of both modalities (audio and face) after the implementation of tree-based feature selection is $\mathbb{R}^{1120}$ and $\mathbb{R}^{682}$ respectively (for the AFEW dataset).

Initially, as it has been explained in Section 1.1.1, the available dataset from the source domain is denoted as $X_S = \{x_1^S, x_2^S, .., x_k^S\}$ (a set of k different instances), while for the target domain $X_T = \{x_1^T, x_2^T, .., x_p^T\}$ a set of p different instances. After the implementation of feature selection modules, the source and the target domain can be denoted as $\hat{X}_S = \{\hat{x}_1^S, \hat{x}_2^S, .., \hat{x}_k^S\}$ (with $\hat{x}_k^S$ being a vector containing a subset of features from X_k^S). Moreover, $\hat{X}_T = \{\hat{x}_1^T, \hat{x}_2^T, .., \hat{x}_p^T\}$ (with $\hat{x}_p^T$ being a vector containing a subset of features from X_p^T) the dataset for D_T after the implementation of feature selection module. The employed datasets contain videos, hence we have the same amount of samples from audio and video modalities, thus, $k = p$. In Figure 2.2, the values of the average vector and the distribution density for face and audio modalities after the performance of feature selection are depicted. It is obvious that the distribution divergence between these



(a) Mean vector average values for face and audio modalities.

(b) Distribution densities for the average vector's values.

Figure 2.5: Feature distributions after the implementation of the clustering approach for the mean face and audio vectors for the AFEW dataset [5]. For each average vector the values for each dimension are represented. Note that this analysis conducted with all samples from the happy class.

modalities is still large, given that the selected features are still coming from different distributions.

2.1.3. UNSUPERVISED LEARNING OF FEATURES

Subsequent to feature selection, an unsupervised learning technique is performed to transform the already filtered features. During this approach, K clusters are calculated for both modalities using a clustering technique. Then, every input feature vector for both domains is represented by the distance of each feature vector to each calculated center (and this process employed distinctively for each domain) by the clustering approach. For example if the number of calculated centers for audio are 100 and for face is 88, then the dimensionality of the extracted representations is 100 for audio and 88 for face. In the end, ($\Phi_{K_S}(\hat{X}_S)$ and $\Phi_{K_T}(\hat{X}_T)$) are representing the transformed features in source and target domains. Φ_K represents the transformation of the features to the distance from the calculated centers. In that case, the k-means algorithm is implemented for calculating the clusters [16].

The motivation behind the aforementioned procedure is manifold: Primarily, both domains need to be transformed into the same dimensionality (in order to proceed with the transformation distance metric learning module). Moreover, transformation techniques and distance metric learning cannot work well in high dimensions, since their inhibiting complexity relates to the dimensionality of the feature vectors [17]. The best performed size of clustering centers during the experimental phase has been found to be within the range of 40-60. Thus, the feature dimensionality after performing *clustering transformation* reduces significantly the dimensionality of the features. Hence, our feature dimensionality dropped from $\hat{x}_S \in \mathbb{R}^{1018}$ and $\hat{x}_T \in \mathbb{R}^{1941}$ to a range of $\Phi_{K_S}(\hat{X}_S), \Phi_{K_T}(\hat{X}_T) \in \mathbb{R}^{40-60}$. In the Experimental results Section 2.2 a tuning process is performed to find the optimal number of clusters. The desired projection between the two modalities is easier to be calculated using the clustering approach due to the decreased number of clusters in comparison to the initially large feature space.

As Figure 2.5 shows (which illustrates the mean face and audio samples after the im-

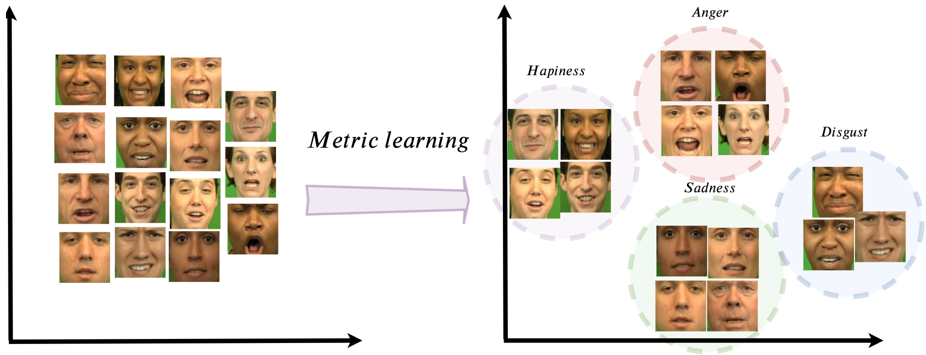


Figure 2.6: A simple instance of how the DML approach can be applied for the face modality. The depicted images are extracted from the CREMA-D dataset [6].

plementation of the clustering approach, its values, and densities), the distributions of mean face and audio cannot be considered comparable yet. A transformation is needed in order to construct the desired bridge between the two domains and reduce the gap between the inherent differences of the distributions. In the current work, Distance Metric Learning techniques are chosen to fill this gap.

2.1.4. DISTANCE METRIC LEARNING FOR AUDIO-VISUAL DOMAIN ADAPTATION

The Distance Metric Learning (DML) problem relates to learning a distance function which is tuned to a specific task using supervised information [17]. In our case, this task is emotion recognition.

A visual example of this approach is depicted in Figure 2.6. This figure illustrates the way that DML operates by calculating a new more advanced distance between input samples. This is done in such a way that samples coming from the same class are projected to a new coordinate space where they are closer to each other. However, in our work, the effort lies in bridging samples with the same emotion context (*affect label*), but from two inherently different domains (face and audio modalities).

All tested methods in this chapter regarding DML assume that we need to have some annotation information available. In this study, since the effort lies in binding together samples from face $X_S \subset D_S$ and audio $X_T \subset D_T$ modalities, we foster the assumption that the label information from both domains can work as supervised input for calculating and optimizing the distance metric. Furthermore, since DML needs to have pairs of samples with the same dimensionality, we decided to use the extracted features after the feature selection and unsupervised (see Section 2.1.3) steps. Therefore, the learned distance can be framed as $\hat{d}(\Phi_{K_S}(\hat{x}^S), \Phi_{K_T}(\hat{x}^T))$, where $\Phi_{K_S}(\hat{x}^S) \in \Phi_{K_S}(\hat{X}^S)$ and $\Phi_{K_T}(\hat{x}^T) \in \Phi_{K_T}(\hat{X}^T)$. The supervised information that connotes the relation between pairs of vectors can be mathematically framed with the following equation:

$$C_{ij} = \begin{cases} 1, & \text{if } (\Phi_{K_S}(\hat{x}_i^S), \Phi_{K_T}(\hat{x}_j^T)) \in S_{im} \\ -1, & \text{if } (\Phi_{K_S}(\hat{x}_i^S), \Phi_{K_T}(\hat{x}_j^T)) \in D \end{cases} \tag{2.3}$$

where the sets S_{im} and D correspond to similar and dissimilar pairs of subjects, respectively. Therefore, S_{im} contains samples from the face and audio modality with the same class, while D contains the same amount of pairs from both modalities coming from different classes. The set S_{im} can be defined as:

$$S_{im} = (\Phi_{K_S}(\hat{x}_i^S), \Phi_{K_T}(\hat{x}_j^T)) \text{ , if } i, j\text{-th samples belong to the same class} \tag{2.4}$$

while the set D can be defined as:

$$D = (\Phi_{K_S}(\hat{x}_i^S), \Phi_{K_T}(\hat{x}_j^T)) \text{ , if } i, j\text{-th samples belong to different classes} \tag{2.5}$$

The generation of the distance $\hat{d}(\Phi_{K_S}(\hat{x}_i^S)), \Phi_{K_T}(\hat{x}_j^T))$ is based on these constraints. This "new distance" can be framed using the following equation:

$$\begin{aligned} \hat{d}(\Phi_{K_S}(\hat{x}_i^S), \Phi_{K_T}(\hat{x}_j^T)) &= ||(A^S\Phi_{K_S}(\hat{x}_i^S) - A^T\Phi_{K_T}(\hat{x}_j^T))^2||_2^2 \\ &= (\Phi_{K_S}(\hat{x}_i^S) - \Phi_{K_T}(\hat{x}_j^T))^T A A^T (\Phi_{K_S}(\hat{x}_i^S) - \Phi_{K_T}(\hat{x}_j^T)) \\ &= (\Phi_{K_S}(\hat{x}_i^S) - \Phi_{K_T}(\hat{x}_j^T))^T M (\Phi_{K_S}(\hat{x}_i^S) - \Phi_{K_T}(\hat{x}_j^T)) \end{aligned} \tag{2.6}$$

where the goal is to learn the matrix $M = A \cdot A^T \in \mathbb{S}_+^d$ where M has to be a $n \times n$ symmetric positive semi-definite (PSD) matrix [19][21]. The dimensionality of the matrix n is the number of input features (both for face and audio domains and corresponds to the number of clusters K). Matrix A^T can be used to transform the input to the distance metric learning domain. The dimensionality of matrix A^T is $n \times o$ where o is a parameter of DML to be tuned. When $o < n$ then DML achieves dimensionality reduction as well. $M \in \mathbb{S}_+^d$ ensures that the "new distance" satisfies the properties of pseudo-distance which are:

- $\hat{d}(\Phi_{K_S}(\hat{x}_i^S), \Phi_{K_T}(\hat{x}_j^T)) \geq 0$ (non nonnegativity),
- $\hat{d}(\Phi_{K_S}(\hat{x}_i^S), \hat{x}_i^T) = 0$ (identity),
- $\hat{d}(\Phi_{K_S}(\hat{x}_i^S), \Phi_{K_T}(\hat{x}_j^T)) = \hat{d}(\Phi_{K_T}(\hat{x}_j^T), \Phi_{K_S}(\hat{x}_i^S))$ (symmetry),
- $\hat{d}(\Phi_{K_S}(\hat{x}_i^S), \Phi_{K_T}(\hat{x}_j^T)) \leq d(\Phi_{K_S}(\hat{x}_i^S), v) + d(v, \Phi_{K_T}(\hat{x}_j^T))$ (triangle inequality)

The next step of DML is to minimize the distances between those adjacent examples indicated in Equation 2.3. This minimization can be described with the following loss function:

$$\mathscr{L}(M, S_{im}, D) = \frac{1}{2}\sum_{i,j}^{n} ||(\Phi_{K_S}(\hat{x}_i^S) - \Phi_{K_T}(\hat{x}_j^T))^T M (\Phi_{K_S}(\hat{x}_i^S) - \Phi_{K_T}(\hat{x}_j^T))||C_{ij} \tag{2.7}$$

The target is to find parameters for the metric such that best agrees with these constraints, in an attempt to approximate the underlying semantic metric. This can be framed as a constrained optimization problem with the following general form:

$$min_M \mathscr{L}(M, S_{im}, D) + \lambda R(M) \tag{2.8}$$

where $\mathscr{L}(M, S_{im}, D)$ is the loss introduced in 2.7, and it incurs a penalty when training constraints are violated, $R(M)$ is some regularizer on the parameters M of the learned metric and λ is the reguralization parameter. All the implemented approaches for learning the DML space essentially differ by their choice of metric, constraints (S_{im} and D), loss function, and regularizer $R(M)$.

Finally, the calculated matrix A is used to project data on the DML space, where the new distance between the pairs that are considered similar will decrease, while it increases for the pairs that are considered dissimilar. In the current approach, the new distance is calculated using the same number of pairs (of transformed features from source $\Phi_{K_S}(\hat{X}_S)$ and target domain $\Phi_{K_T}(\hat{X}_T)$) that have the same or different labels. The cardinality of set S_{im} and D is equal ($|S_{im}| = |D|$). In the end, the projected source domain is denoted as $H_S = A^T \cdot (\Phi_{K_S}(\hat{X}_S))$ while the projected target domain is denoted as $H_T = A^T \cdot (\Phi_{K_T}(\hat{X}_T))$.

In order to find the optimal matrix M and generate a robust bridge between the domains of face and audio, several techniques are tested. The tested techniques are:

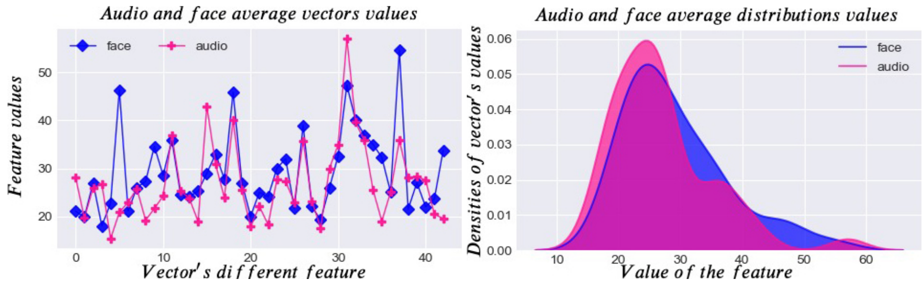
- *Sparse Determinant Metric Learning* (SDML) [19]: In this approach, the LogDet [20] divergence and L_1-regularization on the off-diagonal elements of M are utilized to minimize the total loss in equation 2.8.

- *Relative Components Analysis* (RCA) [21]: This approach leverages only the positive pairs and makes use of subsets of training examples that are denoted as “chunklets”. These are acquired from the set of positive pairs by performing a transitive closure: for instance, if $(x_1, x_2) \in S_{im}$ and $(x_2, x_3) \in S_{im}$, then x_1, x_2 and x_3 belong to the same chunklet. Points in a chunklet share the same label.

- *Local Fisher Discriminant Analysis* (LFDA) [22]: LFDA is a linear supervised dimensionality reduction method. It is particularly useful when dealing with multimodality, where one or more classes consist of separate clusters in the input space. In this case, the core optimization problem of LFDA is solved as a generalized eigenvalue problem.

In Figure 2.7, the mean face and audio vectors after the implementation of a DML transformation (in this example, Sparse Determinant Metric Learning) are depicted (their values and densities). It should be noted that, after the implementation of the DML bridge, the gap between the two mean vectors is reduced as can be observed from Figure 2.7. That is serve as a mere indication of the proximity between the two domains. The results of all under test approaches are presented in Section 2.2.

Finally, a multiclass SVM [23] is employed to measure the classification performance in the target domain (audio) for the task of emotion recognition. More information about SVM method can be found in Appendix 4. The SVM classifier is trained using the



(a) Mean vector average values for face and audio modalities.

(b) Distribution densities for the average vector's values.

Figure 2.7: Feature distributions after the transformation using SDML approach for the mean face and audio vectors for the AFEW dataset [5]. For each average vector the values for each dimension are represented. Note that this analysis conducted with all samples from the happy class.

transformed merged dataset which contains the projected data from both modalities in the DML space. Therefore, the final dataset is a fusion of the transformed audio and face modalities after the DML transformation.

The overall framework of training and testing (of learning a DML space, as well as, the classification schema) is visualized in Figures 2.1. In Figure 2.1, the process for calculating the DML space is displayed in "Block A". While "Block B" illustrates the performed classification schema. "Block A" concerns learning the DML space. Having learned this space, it can be used to project audio and face datasets into this common sub-space and perform classification using SVM classifiers ("Block B").

2.2. EXPERIMENTAL RESULTS USING DISTANCE-METRIC LEARNING FOR AUDIO-VISUAL EMOTION RECOGNITION

The proposed metric learning framework of this chapter has been validated on the challenging datasets Acted Facial Expressions In The Wild (AFEW) [5], Crowd-sourced Emotional Multimodal Actors Dataset (CREMA-D) [6] and The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [7] datasets. All these datasets, as mentioned in Section 1.5, contain discrete emotion labels, namely, six basic emotions except RAVDESS which contains seven (anger, calmness, disgust, fear, happiness, sadness and surprise).

EVALUATION PROTOCOL

During the experimental process, two different scenarios that are related to the learning process (of the DML projection) of our approach are evaluated for all three datasets:

- Firstly, the scenario of *sparsely* available data from the target domain.
- Secondly, the scenario of *densely* available information from the target domain.

Before proceeding with explanation of these two scenarios, it is important to note that the AFEW dataset is split in training and test sets for the target modality $S_T =$

X_T, X_{Test} for all datasets. For the AFEW dataset the training set X_T consists of 773 video samples while the test set consists of 383 videos. For RAVDESS and CREMA-D dataset, we create three different sets for the training, validation, and test sets.

SPARSE SCENARIO

The first configuration is the main scenario of the experimental results and it is referred to as "Semi-supervised Domain Adaptation" in the literature [24]. In this case, a sparse pair from the face ($X_{S1} \subset X_S$) and the audio modality ($X_{T1} \subset X_T$) are used with the purpose of learning the "new distance". The number of samples from both modalities needed to be identical. The reason behind this is the fact that the DML framework is fed with pairs of instances from both modalities. For feature selection and clustering, X_{T1} and X_S sets are used. Subsequently, following a 4-fold cross validation procedure, pairs of samples (X_{S1}, X_{T1}) that correspond to part of the dataset (experiments with considering sparsity levels of 20%, 30%, 40% and 50%) of the dataset are randomly selected. For applying domain adaptation, as described in Section 2.1, the similarity correspondence is established by using their label information. Having learned the DML transformation with the sparse pairs, we can use it to map the training dataset from the source modality X_S and the sparse training dataset from the target modality X_{T1} into the common DML space using the calculated DML projection matrix A^T. Then, both datasets are merged and a new training dataset emerges, namely $H_{train} = (H_S, H_{T1})$.

DENSE SCENARIO

Secondly, in order to perform a broader evaluation, a second test for the calculation of the matrix M was performed by incorporating the full source X_S and target domain X_T training sets. This allows us to evaluate the performance of the HDA system under the dense availability of both datasets. This scheme is referred to as "Fully-supervised Domain Adaptation" in the literature [24].

BASELINE

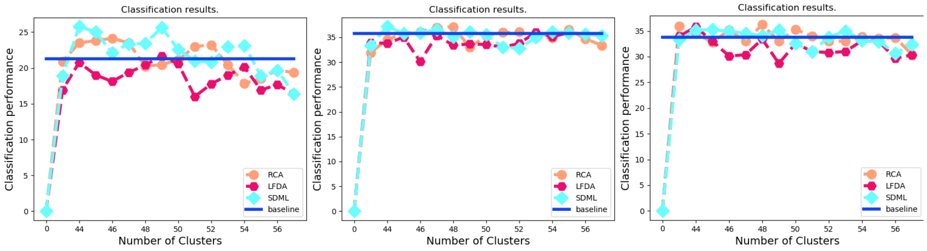
Since the task is to prove that audio emotion recognition can be improved when eliciting information from the face domain, as a baseline, SVM classifiers are trained on the transformed features after the unsupervised learning step 2.1.3 is utilized. This approach is considered as the "baseline method". Training and testing for the SVM is performed using only data from target domain X_T. In particular, for the training process, $\Phi_{K_{T1}}(\hat{X}_{T1})$ are used for sparse scenario ($X_{T1} \subset X_T$) and $\Phi_{K_T}(\hat{X}_T)$ are employed for the dense one. The prediction process is performed using $\Phi_{K_{T_{test}}}(\hat{X}_{T_{test}})$) set.

MAXIMUM MEAN DISCREPANCY

Moreover, with the purpose of illustrate the way the DML transformation operates, Maximum Mean Discrepancy (MMD) was employed [25] in order to investigate the extent to which the distance between the two domains reduces after the applied DML transformation. MMD can be framed using the following equation:

$$D_{MMD}(Q,P) = ||E_Q[\mathscr{F}(x)] - E_P[\mathscr{F}(x)]||^2 \tag{2.9}$$

where Q and P are two different domains, or more analytically in our case:



(a) AFEW dataset. (b) CREMA-D dataset. (c) RAVDESS dataset.

Figure 2.8: Classification results for the evaluation scenarios for the 20% sparsity scenario (which analyzed in Section 2.2) for different datasets and different cluster center number. The blue line represents the baseline SVM performance.

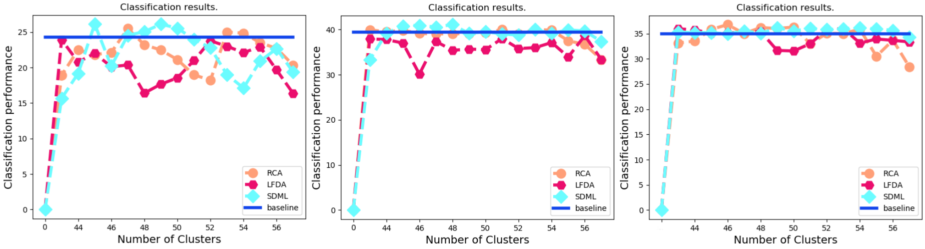
$$D_{MMD}(X_{S_1},X_{T_1}) = ||\frac{1}{m}\sum_{k=1}^{m}\mathscr{F}(x_k^S) - \frac{1}{n}\sum_{k=1}^{m}\mathscr{F}(x_k^T)||^2 \tag{2.10}$$

where $X_{S_1} = [x_1^S, x_2^S, ...x_m^S]$ and $X_{T_1} = [x_1^T, x_2^T, ..., x_m^T]$ represent feature vectors from both domains that are governed from these distributions and $\mathscr{F}$ is the calculated mapping function for the MMD method, which, in our case, is the DML transformation. Therefore, the MMD metric calculates the distance between the means of the source and target domains in the transformed space.

An exhaustive search is performed for the type of feature selection, for the chosen type of clustering method, for the DML technique implemented and finally for the SVM parameters for both scenarios. Furthermore, SVM classifiers are trained using the audio features after performing the unsupervised step to compare them with the proposed technique. This approach, as we mentioned before, connotes the baseline approach. The training is performed using $\Phi_{K_{T1}}(\hat{X}_{T1})$ for the sparse scenario or $\Phi_{K_T}(\hat{X}_T)$ for the dense one, while the prediction is conducted using $\Phi_{K_{T_{test}}}(\hat{X}_{T_{test}})$). For the baseline approach, the SVMs are trained for both scenarios and cases. For the tuning process, a greedy approach is followed by searching all typical parameters for feature selection, k-means, DML and SVM.

Table 2.1: Best extracted classification performance for varying sparsity (20%, 30%, 40%, 50% and 100%) scenarios for the baseline approach and the proposed algorithm. For AFEW and CREMA-D the results are extracted using SDML while for RAVDESS using RCA.

Case	AFEW		CREMAD		RAVDESS	
	Baseline	SDML	Baseline	SDML	Baseline	RCA
20%	21.29%	23.77%	35.82%	37.16%	33.77%	36.20%
30%	21.25%	25.90%	36.13%	37.36%	34.70%	36.83%
40%	22.60%	26.12%	38.36%	38.95%	34.99%	36.77%
50%	24.32%	26.17%	39.44%	40.06%	34.91%	36.90%
Dense	28.84%	29.11%	40.14%	41.58%	37.45%	37.66%



(a) AFEW dataset.

(b) CREMA-D dataset.

(c) RAVDESS dataset.

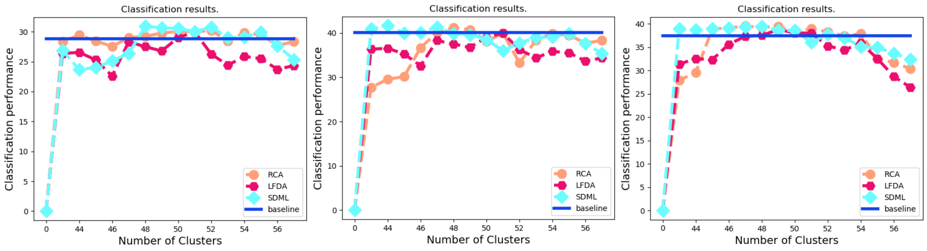
Figure 2.9: Classification results for the evaluation scenarios for the 50% sparsity scenario (which are analyzed in Section 2.2) for different datasets and different cluster numbers. The blue line represents the baseline SVM performance.

EVALUATION DISCUSSION

As it is depicted in Table 2.1 where the best results for different sparsity scenarios (20%, 30%, 40%, 50% and 100%) for all datasets are illustrated, the whole framework succeeds in the target objective. That is the improvement of the classification performance in the target domain by incorporating information from the source dataset (and outperform the baseline method) in all scenarios. This is also depicted from the Figures 2.8, 2.9 and 2.10 (where the results for all datasets are visualized). Moreover, in the case of SDML, a parameter that regularizes the sparsity of the calculated matrix M during the DML optimization process λ by using L_1-regularization M, is tuned [19]. In this case, L_1-regularizer it gives high penalty in big values of the matrix M on the off-diagonal elements.

It is found that by defining this parameter equal to $\lambda = 0.05$ we can extract the best results for AFEW while $\lambda = 0.01$ for CREMA-D.

Firstly, regarding AFEW, it is found that, by using tree-based feature selection, a limited number of dimensions can be kept without decreasing the classification performance in our experiments. In the same spirit, SDML algorithm has led to the best performance for domain adaptation for the sparse scenario for all cases (20%, 30%, 40% and 50% target domain sparsity). More specifically, the best performance regarding the classification results is found by using SDML combined with k-means (and using 53 as a number of clusters in the case of 20% sparsity). For the case of the 30%, 40% and 50% sparsity, the best performance is achieved by using SDML with k-means with 50, 50 and 46 clusters, respectively. For the dense scenario, the proposed technique managed to outbid (again using SDML and k-means with 46 clusters) the performance of the baseline SVM classification. The proposed framework's performance is 29.11% while the baseline performance is 28.84% for the best-performing baseline k-means SVM using only the audio features. The approach reaches similar values with the reported state-of-the-art performance for audio recognition in the dataset which was extracted using Deep Belief Networks [26]. A striking observation extracted throughout the experimental phase is the fact that the proposed method (for each sparsity case) is always converging to the same classification performance using the same number of clusters and the sparsity parameters.



(a) AFEW dataset.

(b) CREMA-D dataset.

(c) RAVDESS dataset.

Figure 2.10: Classification results for the evaluation scenarios for the dense scenario (which are analyzed in Section 2.2) for different datasets and different cluster numbers. The blue line represents the baseline SVM performance.

For the baseline approach, due to the randomness in picking samples for the sparse scenario, and to improve the reliability of the analyses, the results are based on several iterations of the classification task using different sample splitting processes. This observation denotes the efficiency of the knowledge transfer of the face modality over audio modality through the proposed domain adaptation approach.

Furthermore, to better generalize the aforesaid observations, we proceed with two bigger datasets, namely, CREMA-D and RAVDESS. In such a manner, a generalization for the classification performance of the proposed approach using different datasets can be achieved. With respect to CREMA-D, it is found again that, by using the SDML algorithm, the framework obtains the best performance. These results are found when using 46, 43, 45, and 44 clusters for 30%, 40%, 50% and 100% sparsity levels, respectively (see Figures 2.8, 2.9 and 2.10). Regarding the feature selection approach, the tree-based strategy is chosen since it leads to the best performance. In all sparsity evaluations, it is observed that the SDML algorithm is more stable and, in most cases, (for several parameters of the whole approach), the obtained results are superior than in the case of the RCA algorithm. On the other side, the LFDA approach is the weakest and did not manage to perform as well as the rest ones.

Concerning the RAVDESS dataset, the best results are extracted when using tree-based estimators for feature selection and when using 43, 43, 46 and 43 clusters for the 30%, 40%, 50% and 100% sparsity cases. However, in that case, the best performed DML approach is the RCA approach (see Figures 2.8, 2.9 and 2.10 where the results for the three different algorithms and with different sparsity levels are displayed). In the same vein, the LFDA algorithm cannot perform as good as the other two approaches.

For all three datasets, it is evident that the approach is very much dependent on the employed algorithm for DML and, furthermore, on the number of clusters chosen for the clustering step. Even in the same datasets, by changing the sparsity level of available datasets, the number of clusters needs to be tuned again for the best results to be found. Therefore, the tuned parameters of this approach are data-dependent.

Figures 2.9 and 2.10 shows the fluctuation of the performance for the three datasets for different values of the hyper-parameter k for the clustering approach for the baseline and several approaches applied in this chapter. In both sparse and dense scenarios, SDML approach is proven to be more robust in comparison with the other methods.

Regarding the MMD distance between the distributions of the approach, Table 2.2 illustrates the distance between the averages after clustering and the MMD distance after the DML transformation for several k (numbers of clusters). The table is illustrative of the fact that the distance after the performed DML transformation is always smaller than the distance only after the transformation to the clustering procedure (for all datasets and sparsity levels) which corresponds to Euclidean distance. That is observed for all three datasets and validates the behaviour that is noticeable in Figures 2.10(a), 2.10(b), and 2.10(c).

Table 2.2: The distances between domains for several sparsity scenarios.

Dataset	AFEW		CREMAD		RAVDESS	
	Baseline	SDML	Baseline	SDML	Baseline	RCA
20%	1.87	1,19	3.50	2.99	4.13	4.69
30%	1.99	1.55	3.71	3.50	4.81	4.27
40%	1.81	1.64	3.81	3.83	4.81	4.27
50%	1.81	1.64	3.85	3.80	4.81	4.27
Dense	2.27	1.53	3.99	3.92	4.81	4.27

2.3. CONCLUSION

In this chapter, a preliminary strategy for accommodating the challenging task of domain adaptation is introduced. The study focused mainly on the capability of the proposed technique based on simple distinct modules to enhance the emotion recognition performance of the audio modality when leveraging data from the face modality. The goal is to develop a link between both modalities and to eliminate the gap between their distributions. The proposed approach incorporates an *unsupervised learning* approach related to clustering and a metric learning technique with the purpose of establishing a bridge between the two different domains. Two test scenarios are analyzed and an evaluation schema using three different datasets is conducted. Firstly, knowledge transfer is performed from the source to the target domain, with the target domain availability being reduced to 20%,30% and 50% with regards to the original dataset. Secondly, an investigation in the case of a densely available target domain is conducted. To validate this approach, three different datasets are considered, namely AFEW, CREMA-D, and RAVDESS. The experimentation phase demonstrated that, for all scenarios, the performance of the DA algorithm surpassed the performance of a baseline SVM algorithm trained solely on the target domain features. Having established a successful bridge between the two modalities, consequently, this transformation can be exploited with the purpose of improving the target classification task of emotion recognition for the audio modality.

On the whole, the approach is shown to be efficient and very simple to implement with the training process lasting only a short period of time. On the other hand, for RAVDESS and CREMA-D datasets, the improvement when using the proposed approach is rather marginal and in some cases, even negative knowledge transfer is observed between the two modalities. Furthermore, it is found that even the classification performance of the baseline strategy for these datasets is not really close to the state-of-the-art. What is more, the whole approach is dependant on the hyper-parameters and even with a small modification to them, the obtained results are radically different. That is observed when changing the number of clusters, or the parameters, or the feature selection method. The performance of the approach is radically different. The same fluctuations in performance are experienced when using the same hyper-parameters for different datasets. For different datasets, these parameters need to be tuned again.

The current approach focuses on traditional machine learning unsupervised and supervised tools. It provides a proof of concept idea that it is possible to perform emotion-aware audio-visual domain adaptation. Inspired by this and, considering the fact that deep learning can provide versatile solutions to a multitude of in-the-wild problems, we decided to look into a new direction of research. Hence, it was decided to apply deep learning approaches to study audio-visual relationships. The motivation behind this decision is two-fold. As it is found in [27], recent deep learning approaches perform really well in the task of Audio Emotion Recognition (AER). Furthermore, deep learning approaches are implemented in an end-to-end manner. In such a way, the feature extraction process is tackled automatically from the learning algorithm (for instance when using CNN) and powerful expressivity features related to emotions can be uncovered. Chapters 3, 4 and 5 contain the work done in this direction.

REFERENCES

[1] P.Viola and M.Jones, Rapid object detection using a boosted cascade of simple features, IEEE Conference on Computer Vision and Pattern Recognition CVPR, 2001.

[2] P.Viola and M.Jones, Rapid object detection using a boosted cascade of simple features, IEEE Conference on Computer Vision and Pattern Recognition CVPR, 2001.

[3] X.Xiong and F.D.Torres, Supervised Descent Method and its Applications to Face Alignment, IEEE Conference on Computer Vision and Pattern Recognition CVPR, Pages: 532–539, 2013.

[4] G.Zhao and M.Pietikainen, Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 29, Number 6, 2007.

[5] A.Dhall, R.Goecke, S.Lucey, and T.Gedeon, Collecting large, richly annotated facial expression datasets from movies, IEEE Transactions of MultiMedia, Volume 19, Number 3, 2012.

[6] H.Cao, D.G.Cooper, M.K.Keutmann, R.C.Gur, A.Nenkova and R.Verma, CREMA-D: Crowd-Sourced Emotional Multimodal Actors Dataset, IEEE Transactions on Affective Computing, Volume 5, Number 4, Pages: 377–390, 2014.

[7] S.R.Livingstone and F.A.Russo, The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English, Volume 13, Number 5, Pages: 1–35, 2018.

[8] E.Ghaleb, M.Popa, E.Hortal and S.Asteriadis, Multimodal Fusion Based on Information Gain for Emotion Recognition in the Wild, IEEE Intelligent Systems Conference, London, UK, 2017.

[9] A.Jovic, K.Brkic, and N.Bogunovic, A review of feature selection methods with applications, 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015.

[10] F.Eyben, F.Weninger, F.Gross and B.Schuller, Recent Developments in openSMILE, the Munich Open Source Multimedia Feature Extractor, Proceedings of the ACM Multimedia, 2013.

[11] B.Schuller, M.Valstar, F.Eyben, G.McKeown, R.Cowie, and M.Pantic, Avec 2011 the first international audio/visual emotion challenge, in International Conference on Affective Computing and Intelligent Interaction, Pages: 415–424, 2011.

[12] J.P.Teixeira, C.Oliveira and C.Lopes, Vocal Acoustic Analysis – Jitter, Shimmer and HNR Parameters, Procedia Technology, Volume 9, Pages: 1112–1122, 2013.

[13] R.Tibshirani, Regression Shrinkage and Selection via the lasso, Journal of the Royal Statistical Society, Volume 58, Number 1, Pages: 267–288, 1996.

[14] B.Schuller, M.Valstar, F.Eyben, G.McKeown, R.Cowie and M.Pantic, AVEC 2011 The First International Audio Visual Emotion Challenge, Affective Computing and Intelligent Interaction, Pages: 415–424, 2011.

[15] R.Baraniuk, Compressive Sensing, Lecture notes in IEEE signal processing magazine, 2007.

[16] J.A.Hartigan and M.A.Wong, A K Means Clustering Algorithm, of the Royal Statistical Society. Series C (Applied Statistics), Volume 28, Number 1, Pages: 100–108, 1975.

[17] F.Wang and J.Sun, Survey on distance metric learning and dimensionality reduction in data mining, Transactions on Data Mining and Knowledge Discovery, Volume 29, Number 2, 2014.

[18] B.Kulis, Metric Learning: A Survey, Transactions on Foundations and Trends in Machine Learning, Volume 5, Number 4, Pages: 287–364, 2013.

[19] Q.G.Jun, J.Tang, J.T.Zha, T.S.Chua, and H.J.Zhang, An Efficient Sparse Metric Learning in High dimensional Space via L1 penalized Log determinant Regularization, 26-th Annual International Conference on Machine Learning, ICML, Pages: 841–848, 2009.

[20] A.Cichocki, S.Cruces and S.I.Amari, Log-Determinant Divergences Revisited: Alpha–Beta and Gamma Log-Det Divergences, Entropy Volume 17, Number 5, 2014.

[21] E.Y.Liu, Z.Guo, X.Zhang, V.Jojic and W.Wang, Metric Learning from Relative Comparisons by Minimizing Squared Residual, International Conference on Data Mining, Sheraton Brussels, Belgium, 2012.

[22] M.Sugiyama, Local Fisher Discriminant Analysis for Supervised Dimensionality Reduction, 23rd International Conference on Machine Learning, Pages: 905–912, 2006.

[23] C.Cortes and V.Vapnik, Support vector networks, Transactions on Machine Learning , Volume 20, Number 3, Pages: 273–297, 1995.

[24] K.Weiss, T.Khoshgoftaar and D.Wang, A survey of transfer learning, Transactions of Big Data, 2016.

[25] K.M.Borgwardt, A.Gretton, M.J.Rasch, H.P.Kriegel, B.Scholkopf and A.J.Smola, Integrating structured biological data by Kernel Maximum Mean Discrepancy, Transactions on Bioinformatics, Volume 22, Number 14, Pages: 49–57, 2006.

[26] C.J.Pal and Y.Bengio, EmoNets: Multimodal deep learning approaches for emotion recognition in video, Transactions on Multimodal User Interfaces, Volume 10, Pages: 99–111, 2015.

[27] S.Zhang, S.Zhang, T.Huang and W.Gao, Multimodal deep convolutional neural network for audio-visual emotion recognition, Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, 2016.

3

Domain adaptation using Generative Networks

The research presented in this chapter is supported by the following publications:

- *C.Athanasiadis, E.Hortal, and S.Asteriadis, Audio–visual domain adaptation using conditional semi-supervised Generative Adversarial Networks, Neurocomputing Elsevier* [1]*, Volume 397, Pages: 331–344, 2020.*
- *C.Athanasiadis, E.Hortal, and S.Asteriadis, Audio based Emotion Recognition enhancement through Progressive GANs, 27th international conference on image processing (ICIP), Abu Dabi, 2020.*

The core insight established in Chapter 2 is related to the validation of knowledge transfer when using the learned distance metric learning (DML) approach between modalities. The learned projection onto a common subspace was managed to minimize the divergence between the modalities and, consequently, emotion recognition classification performance in the fused datasets was improved. The whole technique was validated during the evaluation process in Section 2.2 and in particular in Table 2.1 where the best results for the baseline and different approaches are illustrated.

In particular, chapter 2 established a “proof of concept” idea that it is possible to perform emotion-aware audio-visual domain adaptation. From Table 2.1 is clear that this approach has improved performance in comparison to “the baseline”. Inspired by this and, considering the fact that deep learning can provide versatile solutions to a multitude of in-the-wild problems, this chapter is now looking into a new direction of research. Hence, it was decided to apply deep learning approaches for studying the audio-visual relationships as a next step. The motivation behind this decision is two-fold. As it was found in in [1], deep learning recent approaches performs really well in the task of Audio Emotion Recognition (AER). Furthermore, deep learning approaches are implemented in an end-to-end manner. In such a way, the feature extraction process is tackled

[1] https://doi.org/10.1016/j.neucom.2019.09.106

automatically from the learning algorithm (for instance, when using CNN) and powerful expressivity features related to emotions can be uncovered. The conclusions from the emotion recognition survey in [1], were validated after experimentation with deep learning architectures such as Convolutional Neural Networks. It was observed that there is a discrepancy of approximately $8\% - 10\%$ in the classification accuracy for audio emotion recognition (for CREMA-D [2] and RAVDESS [3] datasets) in comparison with the results from Chapter 2.

Therefore, it is chosen to proceed with the more sophisticated deep learning approach with the purpose of learning the desired cross-modal projection. This architecture, in principle, is formed by two sub-models. The first one is assigned to perform the cross-modal transformation from source X_S to target domain X_T ($X_S \rightarrow X_T$). While the second one is tasked with judging whether the above-learned projection leads to meaningful results regarding the target domain X_T.

However, this approach comes with one evident drawback. As is expected, this drawback is the need of these algorithms for big amounts of data. The introduced deep learning system is able to be trained only when using the CREMA-D [2] and RAVDESS [3] datasets which contain a big number of samples. On the contrary, when using the AFEW dataset [4], the system is not able to smoothly converge and train properly. Therefore, we decided to proceed only with CREMA-D and RAVDESS datasets and skip the AFEW dataset in the deep learning-related approaches.

The fostered deep learning approach is the so-called Generative Adversarial Networks (GANs) architecture. Over the recent years, explosive popularity has emerged in the domain of GANs [5] which have become one of the most promising developments in deep learning. The preliminary idea of GANs can be framed as follows: Given a vector of random noise z, the whole process endeavors to accomplish a good approximation of the data distribution in hand (in our case, represented as the target domain) by learning a mapping between the noise distribution and this domain. GANs usually consist of two different neural networks which compete in a min-max manner. These networks are called Generator G and Discriminator D and they are depicted in Figure 3.1. The basic idea behind this approach (for studying cross-modal relationships between the symbiotic modalities of video) is shown in Figure 1.1. An illustrative work that renders in detail the way that GANs function is introduced in [5]. In that work, the target is to train a network G that, given a noise vector z, is able to generate new samples derived from the MNIST dataset domain (target domain) by trying to approximate the desired distribution. In the meantime, D tries to decide whether the generated samples are genuine or not.

Starting from the above-mentioned architecture (the so-called vanilla architecture), the scope of the proposed research is to modify it and adapt it to the needs of our goals. The desired objective is to develop a framework that will be able to not only generate data in the target domain but also convert source samples into target domain ones. Thereby, several modifications are needed in the classical version of GANs for the sake of formalizing a system that will be able to perform as such. Recent advancements in GANs suggest several modifications that make them more suitable for the field of domain adaptation and audio-visual cross-modal mapping and provide fertile inspiration to the current work. A cardinal influence is the work done in [6] [7] and, particularly, the one in

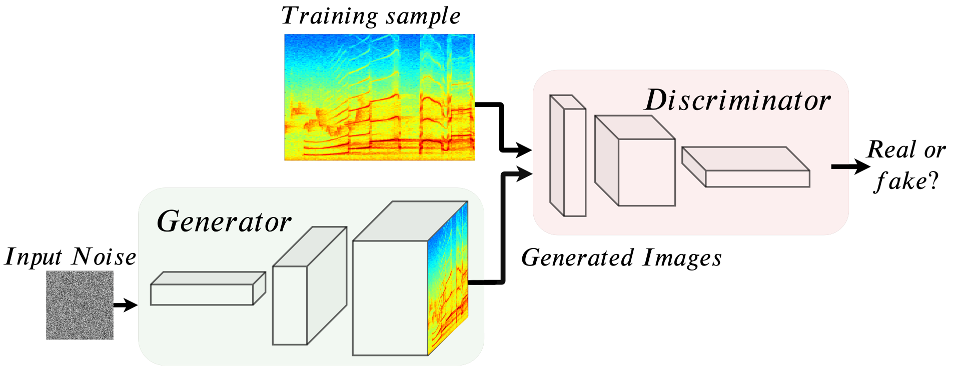


Figure 3.1: Initial version of the Generative Adversarial Networks. The two basic blocks of the architecture, namely, the Generator G and the Discriminator D are rendered.

[8], where a conditional deep Generative Adversarial Network is proposed with the aim of performing image-to-image translation. In this modified version of GANs, a U-Net [9] architecture is proposed with a view to learning the domain shift between two different image datasets that share some characteristics. Contrary to the work done in [8], we propose a semi-supervised architecture, the so-called dacssGAN (Domain Adaptation Conditional Semi-Supervised Generative Adversarial Network) where the input to the generator contains, apart from the source modality data, conditional semi-supervised information extracted using a facial expression classifier (based on convolutional neural networks) and it is processed using conformal prediction (CP) [10] [11]. Conformal prediction is a framework for credible machine learning, constituting a methodology for obtaining error calibration in classification and regression tasks. This framework is based on hypothesis assumptions to provide rigorous error calibration. It allows obtaining confidence values for any class label given a test instance. In the current work, the implementation of CP is performed in order to provide robust conditional information as input to the proposed dacssGAN architecture.

However, directly learning this aforesaid projection based on GANs is still a very complicated procedure that requires sophisticated deep architectures and access to enormous datasets that are also consisted of linked annotations related to the emotion states of the subjects from videos. For this reason, a *progressive training* of the whole architecture is also introduced to improve the learning capacity of the whole framework. This method consists of two steps. Insofar as the first step is concerned, the so-called *weight initialization*, a large data corpus of unlabeled audio-visual clips, derived from the VoxCeleb dataset [12], is used in an attempt to uncover low-level hidden relationships between the face and audio modalities. VoxCeleb has proven to be really useful for the proposed task since it contains a massive unlabelled correspondence of audio and face samples. As a step further, for the sake of calibrating the network in an emotion-wise manner, we made use of emotion-labeled datasets, namely, CREMA-D and RAVDESS.

As it has already been mentioned in Chapter 1, the main research questions that we address in this chapter are two: Firstly, *How can Generative Adversarial Networks be deployed to uncover audio-visual cross modal relationships?*. Secondly, *How can we learn* *and train the proposed Generative Neural Networks in a progressive manner? Or in simple terms, how can we initialize the weights of the network in such a way that can allow us to optimize the training process?.* On the whole, a synopsis of the current Chapter's contributions is summarized as follows:

- The challenging task of heterogeneous semi-supervised domain adaptation between the symbiotic audio-visual modalities in the affective understanding context is explored.
- A novel label-agnostic architecture for GANs based on conditional information extracted using a classifier's prediction scheme is introduced.
- Conformal prediction is utilized as an extra step to update the classifier's prediction. Inductive conformal prediction [13] is evaluated as well, with a view to remedying the high implementation cost of the traditional conformal prediction approach.
- A regulation mechanism over the generator that consists of an auxiliary classifier is opted to impose the emotion states over the generated samples.
- The implementation of the knowledge transfer process occurs by using a progressive calibration of the weights.
- An ablation study is performed in an attempt to investigate the capability of different architectures, loss functions and the performance of different conditional inputs on the presented GAN approach.
- The evaluation of the domain adaptation procedure is performed by implementing a data augmentation schema (similar to [14] and [15]), where generated and real samples are fused together and emotion recognition is performed in this expanded dataset.

Finally, the structure of the remainder of this Chapter is as follows: Section 3.1 describes the introduced Domain Adaptation (DA) method that is based on the proposed GANs architecture and the progressive training of it, while in Section 3.2 the experimental protocol, dataset and results are presented and analyzed. Finally, Section 3.3 discusses the conclusion and the future work of this study.

3.1. DOMAIN ADAPTATION BASED ON GENERATIVE ADVERSARIAL NETWORKS APPROACH

In this section, the proposed GANs approach which is denoted as "dacssGAN" is discussed in detail. The overall architecture of the approach can be seen in Figure 3.2. Figures 3.3(a) and 3.3(b) are displaying two different architectures which are implemented for the network G (and are examined during the ablation study in Section 3.2.3). GANs, as mentioned before, consist of two networks, a generator G and a discriminator D. Given a noise vector z as input to the network G and a dataset of samples that come from the

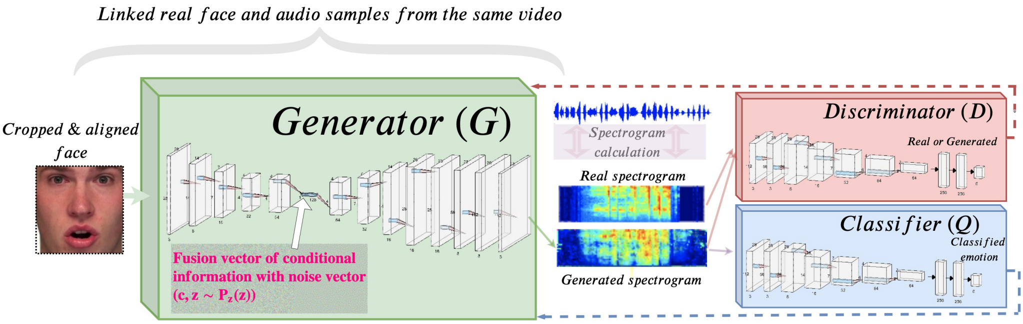


Figure 3.2: Complete architecture of the dacssGAN approach. The diagram contains the three basic blocks of the approach namely, the network G, the network D and the classifier network Q. Furthermore, the input and output, as well as, the architecture for each block are also displayed.

target domain distribution $X_T = \{x_1, x_2, ..., x_n\} \subseteq D_T$, network G is calibrated to generate unseen samples that resemble that distribution. While for network D, the input is the samples distributed in the target domain X_T and the generated samples $G(z)$. It is tasked to learn how to distinguish between genuine and synthetic (generated) samples. The whole training procedure is occurring in an adversarial fashion implemented as a min-max algorithm. The initial formalization of the process can be framed with the following equation:

$$\min_G \max_D V_1(D, G) = \mathbb{E}_{t \sim X_T}[\log D(t)] + \mathbb{E}_{z \sim P_z}[\log(1 - D(G(z)))] \tag{3.1}$$

where $t \in X_T$ (samples coming from target domain D_T) corresponds to the data that we want to approximate while $z \in P_z$ corresponds to the distribution domain that the noise vector z is sampled from. However, the objective in the current work is to implement a domain shift and calculate a transformation between the source (D_s) and the target domain (D_T). Thereby, instead of having as input to the G network the noise vector $z \in P_z$, samples that are distributed from the source domain $X_S = \{x_1, x_2, ..., x_n\} \subseteq D_S$ are also utilized. However, if we proceed by neglecting completely the noise vector z, that may result in the development of a network that only produces deterministic outputs. Thus, the noise vector z should be used also as an input together with the source domain samples in G [16]. Consequently, Equation 3.1 becomes:

$$\min_G \max_D V_2(D, G) = \mathbb{E}_{t \sim X_T}[\log D(t)] + \mathbb{E}_{z \sim P_z, x \sim X_S}[\log(1 - D(G(x, z)))] \tag{3.2}$$

where x are samples derived from the source domain ($x \in X_S$) and t are samples belonging to the target domain ($t \in X_T$). Additionally, since the goal is to generate data that approximate the target domain $X_T \subseteq D_T$ conditioned to emotional information, Equation 3.2 is easily re-framed for the conditional scenario as:

$$\min_G \max_D V_3(D, G) = \mathbb{E}_{t \sim X_T}[\log D(t)] + \mathbb{E}_{z \sim P_z, x \sim X_S}[\log(1 - D(G(x, c, z)))] \tag{3.3}$$

where the input in G network is conditioned to the variable information c. In the current framework, we examined the possibility of having three alternative sources of conditional information as input to the network G namely: class label information, prediction of a classifier (trained using datasets that derived from source domain X_S), and conformal predictions [11]. This strategy is analyzed in more detail in Section 3.1.3. Furthermore, similar to [9], in our study, we investigated the possibility of combining the initial GANs objective with a more classical loss, such as the L1 distance [17]. The discriminator's task remains the same, however, the generator is deputed to not only fool the discriminator but also to be near the ground truth output in an L1 manner (that, in our case, is calculated in a pixel-wise manner). It is found [9] that L1 norm encourages less blurring in the results than other metrics like L2. The implemented loss function that is formulated in our framework is as follows:

$$L_1(G(x,c,z),t) = E_{t\sim X_T, x\sim X_S, z\sim p_z(z)}[||t - G(x,c,z)||_1] \tag{3.4}$$

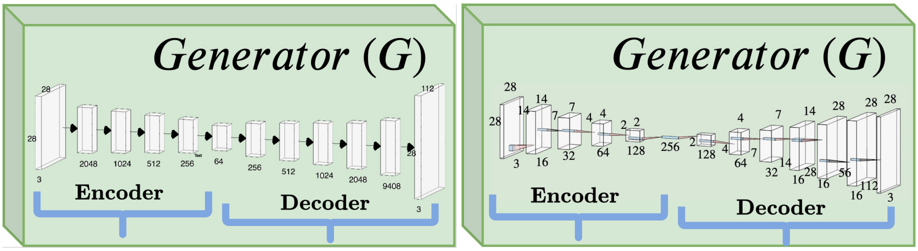
The complete optimization schema that derives after combining Equations 3.3 and 3.4 is formulated as:

$$\min_G \max_D (V_3(D, G(x,c,z)) + L_1(G(x,c,z),t)) \tag{3.5}$$

Hitherto, the system uses some conditional information about the label, however, during the experimental phase, it is found that by just adding this conditional variable c is not efficient enough to produce genuine samples from the target domain (that also represent the desired emotional states). In Section 3.2.3 a manifestation about the previous statement is provided. Thereby, we decided to add an extra network $Q = f_T(x \in G(x,c,z))$ that is producing an error based on the correct or wrong classification of the emotional states. This model Q is presented in Figure 3.2 as “Classifier”. The proposed network Q is a CNN network with an architecture similar to that used in the network D. However, they differ in the last layer that in the classifier case outputs the predicted emotion state for the input audio samples instead of the binary decision provided by the network D. The input of this network is the output samples of G and the cross-entropy error is passed to the generator optimization in tandem with Formula 3.5. Intuitively, the target is to calibrate and influence the output of the network G by considering its capability to reproduce samples that are not only governed from the target domain X_T distribution but also represent as good as possible the class information that the samples of the X_T are associated with. In the work presented in [18] and [19], authors already introduced an extra classifier in the whole training process of GANs. In our approach, the introduced error of the classifier, the cross-entropy of the generated samples from the network G, is denoted as:

$$L_C = \mathbb{E}[\log(P(c = y^k | x^{'}]))] \tag{3.6}$$

where $P(c = y^k)$ denotes the probability for the specific class k and $(P(c = y^k | x^{'}]))$ the probability of the sample $x^{'}$ to belong to class k. This sample belongs either to real dataset $x^{'} \in X_T$ or in the generated samples $x^{'} \in D(G(x,c,z))$. Eventually, the complete loss function is the summary of Equations 3.5 and 3.6:



(a) The architecture of the network G (ED).

(b) The architecture of the network G (U-NET).

Figure 3.3: Different tested architectures for the network G. In both architectures, we denoted as encoder the first part that transforms the input image to a latent space, while as decoder the inverse operation.

$$\min_G \max_D (V_3(D, G(x, c, z)) + L_1(G(x, c, z), t) + \mathbb{E}[\log(P(c = y^k|x^{'}])) \tag{3.7}$$

All the details about the models G, D and Q can be found in Section 3.1.2.

3.1.1. Mode collapse problem and remedies

One of the most profound drawbacks of GANs is the so-called *mode collapse* problem. Mode collapse is the phenomenon where the network G generates a limited diversity of samples, or even the same samples, regardless of the input. Authors in [23] presented this issue and provided a precise explanation of the reasons why this phenomenon occurs. According to them, since training is a stochastic process, due to the randomness introduced with vector z, during the early stages in training, the generated samples will deviate depending on $z \in P(z)$ and the samples drawn from $x \in \mathscr{X}$. In other words, the gradients back-propagated to the network G will deviate between training steps relying on input information. In practice, there exists a single fixed point for the weights that network G considers as the optimum ones for the generation process regardless of the input information we fed into it. However, there is nothing in the objective function that explicitly forces the network G to generate different samples given different input. For its part, Network D eventually is not imposing any more variety in the generated samples or forcing the partially collapsed G towards a different direction.

Possible remedies that are proposed in the literature (mainly in [24] [25] [26]) and were proven extremely useful during the experimental procedure of the current chapter for mitigating the mode collapse phenomenon, are:

- Firstly, the normalization of the input in our network G is utilized. That is, in general, an intuitive method confirmed to be useful in deep learning. In this manner, the optimization process stabilizes, and the risk of exploding gradient which leads to mode collapse is minimized.

- Secondly, another approach is the implementation of *batch normalization* or *batchnorm* for each layer of the employed architecture (Figure 3.2). In this way, not only the input of the networks but also the intermediate layer are properly normalized. Batch normalization is applied during training by computing the mean and standard deviation of the input variables for each layer per mini-batch and using these statistics we can normalize the activation's of each layer. More information can be found in Appendix 3.1. In this manner, we ensure a smooth gradient in each layer which mitigates the problem of mode collapse.

- Thirdly, the Leaky REctified Linear Unit *LeakyRelu* [28] is utilized as the activation function for all three networks (G, D and Q) of the proposed architecture. This activation function is introduced to mitigate the issue of dying ReLU which is described in Appendix 3.1, which is the most commonly applied activation function in deep learning [28]. The problem with ReLU is that all the negative values become zero immediately. In this way, the capability of the model to fit or train the data properly decreases. This means that any negative input given to the ReLU activation function turns the value into zero immediately, which in turn affects the result by not mapping the negative values appropriately. Therefore, a small variation of ReLU is proposed in order to be able to not loose the negative values (LeakyReLU). Both activation functions are visualized in Figures .10(c) and .10(d).

- Furthermore, the application of soft and noisy labels is employed (in the case that the conditional variable c is formed by the real label information, see "Supervised conditional GAN" in Section 3.2.2). The inspiration for this approach is the work in [26] where the labels, in the same way, are input to the network G.

- Another remedy that can be utilized, is the adoption of Adagrad [27] as the optimizer for all networks of the dacssGAN architecture. According to [26], the Adagrad optimizer helps the GANs system becoming more flexible and less prone to *mode collapse*. This is due to the adjustable nature of the learning rate during the training process.

- As a final remedy, the addition of some noise to the input data [26] for each batch step has been also shown to provide a solution, since in this way we are able to add some stochasticity in the data themselves and can push the generator into less deterministic results.

3.1.2. Networks' topology

Taking into account the aforesaid remedies, we designed all the networks of our architecture accordingly. In Figure 3.2 the topology of the network G, is displayed, as well as the input and output for each network. In this figure, we mainly display the tensor sizes after the applied operations involved in the CNN network. The operations are *convolution*, *max pooling*, *dropout* and *batch Normalization*, an activation function and a fully connected layer (depicted in Figure 3.2 with one dimensional tensors) operations which are described in Appendix 3.1. For each tensor, as described in Appendix 3.1, firstly, the convolution operator is applied, then batch normalization and dropout. The resulted tensor then is filtered using an activation function.

This topology is influenced from the image to image conditional Deep Convolutional Generative Adversarial Networks topology, found in [20]. However, several modifications are performed to simplify the training procedure by taking into account the aforesaid

Table 3.1: Details about the employed networks. The variable θ stands for the dropout parameter (see Appendix 3.1).

Network	learning rate	Optimizer	θ
G	0.001	RMSProp	0.5
D	0.001	RMSProp	0.5
Q	0.001	RMSProp	0.5

remedies for the mode collapse problem. On this ground, we have chosen Leaky REctified Linear Unit (LeakyReLU) as the activation function for each of the layers (for more details see Appendix 3.1). As an exception, in the output layer, we made use of the hyperbolic tangent function in order to normalize the output to be between $[-1,1]$. Furthermore, batch normalization and the dropout operations are performed in all the layers of all three networks except the output layer. Table 3.1 contains details related to the networks hyper-parameters such as learning, optimizer and dropout parameter θ.

Table 3.2 contains all the details of the generator network G, with all the dimensionalities of the input and output tensors, as well as all the involved operators. Similarly, Table 3.3 renders all these details for the discriminator network D. Classifier network Q is actually similar with the discriminator D with only difference the last layer (is a dense layer with k dimensions instead of a binary layer as in case of the discriminator).

3.1.3. Semi-supervised GANs

The conditional information c that has been applied as a supplementary input to network G (together with the vector noise z) in the classical version of conditional GANs [5] is mainly associated with the label information ($c = y_i \in \mathscr{Y}$) of the target domain samples ($x \in X_T$). This label information is represented by categorical vector with its dimensionality to be the number of classes.

However, these labels are not always available, and being able to construct a network G regardless of the label information is one of the main objectives in the current chapter. Having said that, the very first approach that is fostered to displace the conditional label information is to use a trained classifier $f_S(X_S)$ that relates to source classification task T_S. This classifier (displayed in Figure 3.4) will output the prediction confidence that depicts the probability for each sample of being associated with each class ($y_i \in \mathscr{Y}$ that represents in our case every different emotion, denoted in Equation 3.7 as c). In this respect, by neglecting the class information in the input of the network G, we can state that our generator is operating in a semi-supervised manner.

Conformal prediction

As an alternative to the analyzed semi-supervised approach of Section 3.1.3, we investigate the efficiency of conformal prediction (CP) [11] as a way to output better class-confidence and use it as the conditional information that is fed into the network G instead of simply using the classifier prediction output. The CP framework is a probabilistic approach focused on post-processing of classification results for more reliable pre-

Table 3.2: The model employed for the generator network *G*.

Layer type	tensor input	tensor output
Flatten	3 × 28 × 28	1 × 2352
Dense	1 × 2352	1 × 1024
Batch Normalization	1 × 1024	1 × 1024
Dropout	1 × 1024	1 × 1024
Dense	1 × 1024	1 × 512
Batch Normalization	1 × 512	1 × 512
Dropout	1 × 512	1 × 512
Dense	1 × 512	1 × 256
Batch Normalization	1 × 256	1 × 256
Dropout	1 × 256	1 × 256
Dense	1 × 256	1 × 64
Batch Normalization	1 × 64	1 × 64
Dropout	1 × 64	1 × 64
Dense	1 × 64	1 × 256
Batch Normalization	1 × 256	1 × 256
Dropout	1 × 256	1 × 256
Dense	1 × 256	1 × 512
Batch Normalization	1 × 512	1 × 512
Dropout	1 × 512	1 × 512
Dense	1 × 512	1 × 1024
Batch Normalization	1 × 1024	1 × 1024
Dropout	1 × 1024	1 × 1024
Dense	1 × 1024	1 × 2048
Batch Normalization	1 × 2048	1 × 2048
Dropout	1 × 2048	1 × 2048
Dense	1 × 1024	1 × 2048
Batch Normalization	1 × 2048	1 × 2048
Dropout	1 × 2048	1 × 2048
Dense	1 × 2048	1 × 4096
Batch Normalization	1 × 4096	1 × 4096
Dropout	1 × 4096	1 × 4096
Flatten	1 × 2352	3 × 112 × 28

Table 3.3: The model employed for the discriminator network *D*.

Layer type	tensor input	tensor output
Conv2D	3 × 112 × 28	64 × 112 × 28
Batch Normalization	64 × 112 × 28	64 × 112 × 28
Dropout	64 × 112 × 28	64 × 112 × 28
MaxPooling	64 × 112 × 28	64 × 56 × 28
Conv2D	64 × 56 × 28	32 × 56 × 28
Batch Normalization	32 × 56 × 28	32 × 56 × 28
Dropout	32 × 56 × 28	32 × 56 × 28
MaxPooling	32 × 56 × 28	32 × 28 × 28
Conv2D	32 × 56 × 28	16 × 56 × 28
Batch Normalization	16 × 56 × 28	16 × 56 × 28
Dropout	16 × 56 × 28	16 × 56 × 28
MaxPooling	16 × 56 × 28	16 × 28 × 28
Conv2D	8 × 28 × 28	8 × 28 × 28
Batch Normalization	8 × 28 × 28	8 × 28 × 28
Dropout	8 × 28 × 28	8 × 28 × 28
MaxPooling	8 × 28 × 28	8 × 14 × 14
Conv2D	8 × 14 × 14	4 × 14 × 14
Batch Normalization	4 × 14 × 14	4 × 14 × 14
Dropout	4 × 14 × 14	4 × 14 × 14
MaxPooling	4 × 14 × 14	4 × 7 × 7
Conv2D	4 × 7 × 7	4 × 4 × 4
Batch Normalization	4 × 4 × 4	4 × 4 × 4
Dropout	4 × 4 × 4	4 × 4 × 4
MaxPooling	4 × 4 × 4	4 × 2 × 2
Dense layer	4 × 2 × 2	1 × 16]
Dense layer	1 × 16	1 × 512
Dense layer	1 × 512	1 × 6

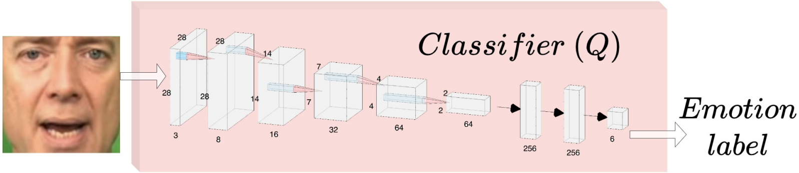


Figure 3.4: The face classifier applied in the case of semi-supervised GANs. In the same manner as in Section 3.1.2, the classifier contains the similar activation function and operations such as batch normalization and dropout.

dictions. It combines a methodology of algorithmic randomness and hypothesis testing to provide error calibration in online settings. It is important to note here, that an identically independently distributed (i.i.d.) assumption for the data samples of the dataset is made. Conformal prediction uses past experience to determine precise levels of confidence in predictions.

CP is a classical frequentist method and concerns a hypothesis testing which creates error regions for each sample. This is done, without employing prior probabilities as it would have been the case for *Bayesian modelling*. CP goal is to produce error regions around the predictions. The underlying rationale is that samples less similar to the training set should lead to less certain estimates: this is captured by *nonconformity scoring function*.

On the whole, the methodology is summarized in Algorithm 3. In particular, an illustrative example of the way CP functions is described as follows: given a dataset $X_D = \{(x_1, y^p), (x_2, y^p), ..., (x_n, y^p)\}$ (where $p \in \{1, 2, ..., w\}$ with w the number of classes), a classifier $f(X)$ and a new test data point x_{n+1}, the hypothesis that x_{n+1} is assigned to a specific class label $c = y^p \in \mathcal{Y}$ is adopted. Having defined the test hypothesis, a re-training process for the adopted classifier $f(X)$, with $X_D \cup \{x_{n+1}, y^p\}$ is performed. Subsequently, a nonconformity function for all the data points $\{(x_1, x_2, ..., x_{n+1})\}$ is recomputed assuming this hypothesis true. This nonconformity function is correlated with the chosen classifier that is selected. In [21], several nonconformity measures that correspond to several classifiers are presented. For the case of CNN, the nonconformity measure for sample x_j is framed by using two different types [13]. Firstly, the so-called *hinge nonconformity*:

$$a_j^{y^p} = 1 - o_j^{y^p} \tag{3.8}$$

and secondly, the so-called *margin nonconformity*:

$$a_j^{y^p} = -o_j^{y^p} + \max_{i=1,..,w, i\neq p} o_j^{y^i} \tag{3.9}$$

where $o_j^{y^p}$ denotes the output of the last layer of the CNN for the specific class p and in particular, it corresponds to the softmax output function of the CNN architecture for the j-th sample (trained in the whole dataset X_D). Then, $\max_{i=1,..,M, i\neq p} o_j^{y^i}$ corresponds to the higher value among all conformity hypotheses excluding the case where $i = p$. Both measures try to measure the degree of *strangeness* for the new data. We can say for both measures that the sample x_j does not conform to class p when $a_j^{y^p}$ is low. In the current approach, we used Equation 3.8 as the established nonconformity measure. The next step of the approach is to define a p-value function measurement for each posed hypothesis, given in Equation 3.10:

$$p(a_{n+1}^{y^p}) = \frac{count\{k \in \{1, ..., n+1\} : a_k^{y^P} \geq a_{n+1}^{y^P}\}}{n+1} \tag{3.10}$$

where $a_{n+1}^{y^P}$ denotes the nonconformity measure of x_{n+1} when it is assumed that it belongs to class label $c = y^p$. This test hypothesis is performed with all available classes (and the corresponding p-value for each hypothesis is calculated). It is obvious that the p-value is highest when all nonconformity measures of training data belonging to the class $c = y^p$ are higher than that of the new test point x_{n+1}, which points out that x_{n+1} is most conformal to the class $c = y^p$. This process is repeated by performing the null hypothesis for all class labels, and the highest p-value is used to decide the actual class label to be assigned to x_{n+1}. Considering p_j as the highest p-value and p_k as the second-highest p-value, p_j is called the credibility of the decision while 1 - p_k represents the confidence of the classifier's decision.

Algorithm 1 Pseudo-code for the conformal prediction process [21].

```
Given a training set $D = (x_i, y^p), ..., (x_n, y^p), x_i \in X$, number of classes $y^p \in Y = y^1, y^2, ..., y^w$ and a classifier $f(X)$:
Get a new unlabeled sample $x_{n+1}$.
for all class labels $y^j$, where j = 1, ..., w do
    Assign label $y^j$ to $x_{n+1}$.
    Re-train the classifier $f(X)$, with $D \cup \{x_{n+1}, y^{(j)}\}$.
    Compute nonconformity measure value, $a_i^{y^j}$ with $i = 1...., n+1$ to compute the p-value according to Equation 3.10.
end for
Output all the conformal prediction based on the p-value prediction for each of the w class according to Equation 3.10.
```

However, as it was already mentioned before, in the current approach, we use the inductive conformal predictions that are introduced in [13], where a set of size $l = n - r$ is denoted as the training set and a set of size r is denoted as the calibration set of the conformal prediction. For a new x_{n+1} now the calculation of p-value is occurring without the re-training of the classifier $f(X)$ but just by directly comparing the nonconformity value of that sample with the nonconformity values of the calibration set. This method is preferred rather than the classic CP method described in Algorithm 3 to reduce the complexity of the approach.

3.1.4. Conformal prediction in GANs for domain transfer

In the light of the above, the p-values are calculated for the training and test datasets by implementing the mentioned inductive CP approach. This is done by using the CREMA-D and RAVDESS datasets and by applying the same CNN network (to calculate the nonconformity measures) as in the previous semi-supervised case (which is displayed in Figure 3.4) and its details are analyzed in Section 3.1.2. More details, about the way that the datasets are split and applied for calculating the p-values for the inductive CP (into the train, calibration, and test subsets), as well as the training and prediction phase of GANs can be found in Experimental phase Section 3.2.

The calculated p-values of the training dataset mentioned before, are given as conditional input c to the proposed dacssGAN architecture (to the generator network G). Hence, the conditional input is modified, instead of directly using the label information or the outcome of the classifier $f_S(X_S)$, as illustrated in Figure 3.2. Then, the whole training procedure for GANs proceeds as before. Regarding the test procedure for generating samples using the test subset, we can apply the trained model for generator G as before. By giving as input to the G the p-values again instead of the labels we can generate spectrograms related to the source domain.

The rationale behind using inductive CP is the high computational complexity cost of the initial conformal prediction algorithm in combination with a GAN architecture. This approach only requires that the underlying algorithm be employed once to generate a classification rule, and thus, the re-training process for each new x_{n+1} sample is not necessary as in the case of the classic CP. Additionally, experimental tests resulted in similar behaviors when using inductive CP in our dacssGAN approach in comparison with the plain version of the CP algorithm. The way that the dataset can be split by forming the training and calibration subsets (hence, determining the size of r) is based on techniques (query functions) which are described in [13]. These techniques provide an efficient way to split the training and calibration subsets for the inductive CP and are based on picking the most uncertain samples (based on the nonconformity measurement) or the most diverse samples in the training set, or randomly choosing the samples for the calibration subset.

As it is explained in more detail in the experimental phase (Section 3.2.3), we demonstrate that in order to reduce the computational cost, the most straight-forward way is to employ as a query function the *random split* of the subsets.

3.1.5. PROGRESSIVE TRAINING

Directly training the proposed deep learning architecture introduced in Section 3.1 may be a challenging task. Furthermore, it is observed, that the proposed architecture needs to be complex enough to learn a meaningful projection between both domains. In total, during the training procedure, our architecture needs to calibrate more than 50 million parameters. Directly learning this number of parameters requires an extremely large data corpus of correlated face and audio samples annotated with the emotional states. Unfortunately, such a big dataset, with these specifications, could not be easily met in the affective computing field. To facilitate this task, a **progressive learning** procedure is introduced in this work.

This approach is proposed to better tune the weights of the architecture and increase the efficiency of the method. It consists of two steps:

- Firstly, a corpus of unlabeled data (VoxCeleb dataset [22]) is used with the purpose of initializing the weights of the networks G and D. In this sense, with this process, our goal is the network G to learn a general audio-visual mapping ($X_S \rightarrow X_T$) without placing any additional conditional constraints (such as the class information). The network G is fine-tuned to generate samples from the target domain conditioned merely upon the low-level pixel information of the source domain X_S. In this manner, low-level, not related to emotions, correlations among features can be retrieved and pave the path for domain adaptation at a later stage. In this unlabelled training of the whole architecture, the same GANs architecture is applied with all the implemented details described in Section 3.1.2.
    - Firstly, since we do not use any label information for the tuning of the network, we need to modify the input to the network G. Instead of using as in-

put the conditional information $[x, z, c]$ with $x \in X_S$, $z \sim P(z)$ the noise vector and c the conditional information, we just make use of $[x, z]$.

  - The second modification concerns the removal of the classifier Q from the whole process. Since we do not possess any label information in VoxCeleb corpus, the classifier Q is unnecessary. We proceed with the GANs training using only the generator G and the discriminator D.

- Then, having initialized the weights of G to produce unlabelled data from the target domain, the next step is to continue the optimization of the whole network (as described in the previous subsections) by using a fully annotated audio-visual dataset (the conditional information $[x, z, c]$ and the classifier network Q) with a view to calibrating the weights and to expose high-level emotion-wise relationships between the face and the audio domain. In this manner, having tuned the weights emotion-wise, we can evaluate whether the previous step is helpful regarding knowledge transfer evaluation.

- The evaluation of the whole progressive training is performed at the end of the second step. Having trained our GANs approach by using these two steps, we can calculate then the performance of the approach as it is performed for the normal training of GANs (without the weight initialization process). More details about the evaluation and the metrics implemented can be found in the experimental phase of this chapter in Section 3.2.4.

The evaluation of progressive training is introduced in Section 3.2.

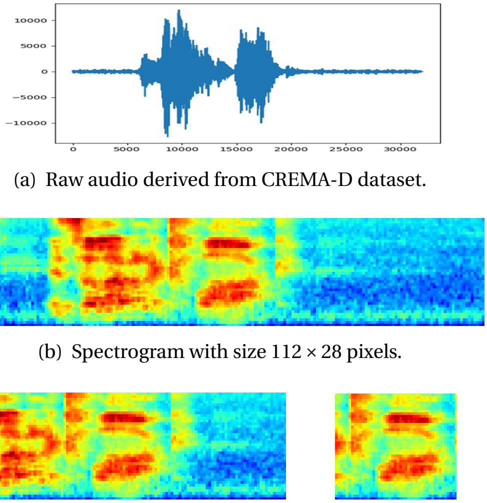
3.1.6. Pre-processing steps for face and audio modalities

In this Section, the main pre-processing steps that are followed in respect to face and audio modalities are analyzed. As it has been analytically presented in the introductory chapter, Section 1.2.2, these steps are performed to extract the most relevant information that relates to emotion labels. In this way, we discard unnecessary noisy information from both modalities and transform the raw signals into more compact representations that will be easily handled by our GANs architecture. Hence, regarding the pre-processing steps for the face, we follow the same strategy as in Chapter 1, in order to extract the bounding box for the faces. While for the audio modality, we are using spectrogram representation. Having performed these steps, then, the proposed GANs architecture is implemented using the transformed modalities.

In [40], an analysis with the common way of performing pre-processing for both modalities for emotion recognition can be found. It is evident, for the audio-modality the most recent techniques employ the spectrogram representation as well.

Calculation of spectrograms

Motivated by several works dealing with audio classification tasks [29] [30] [31], it was decided to extract and make use of a spectrogram representation (instead of making use of the raw audio signals for the target domain X_T) for the feature extraction process. A spectrogram is a visual depiction of the spectrum of frequencies of a signal (audio signal in our case) and its fluctuations over time. In this manner, the whole approach of



(a) Raw audio derived from CREMA-D dataset.

(b) Spectrogram with size 112 × 28 pixels.

(c) 56 × 28 pixels.

(d) 28 × 28 pixels..

Figure 3.5: The initial raw audio and the correspondent transformed spectrogram.

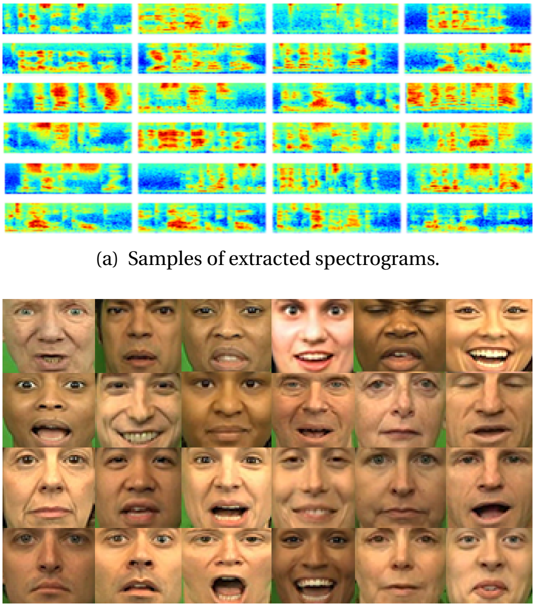
knowledge transfer is transformed into an image-to-image translation task that will also make easier the implementation of our GAN architecture. Additionally, it also facilitates the qualitative inspection of the generated spectrograms produced during experimental results. The whole process of extracting spectrograms given raw audio is detailed in [32]. In Figure 3.5, a raw audio signal and its correspondent spectrogram are visualized.

Finally, from the resulting spectrograms, only the central parts of the clips are kept. For this purpose, three different configurations have been considered: 28 × 28, 56 × 28 and 112 × 28 pixels. This fixed size is established by always starting from the middle part of the spectrogram and symmetrically keeping the surrounding regions (that roughly corresponds to auditory information of 0.2, 0.5, and 1 second respectively) to avoid the silence appearing in the beginning and in the end of the file. In Figure 3.6(a), samples of spectrograms extracted from the CREMA-D dataset are visualized.

Data augmentation for the spectrograms A data augmentation technique with which it is possible to increase the number of samples from the target domain is also applied. This technique is based on SpecAugment [33].

SpecAugment is an approach for performing data augmentation in spectrograms. It modifies the spectrogram by warping it in the time direction, masking blocks of consecutive frequency channels, and masking blocks of utterances in time.

In this way, by having as input one spectrogram, 10 new spectrograms are generated and in this way, we are able to increase the number of data samples in our dataset.



(a) Samples of extracted spectrograms.

(b) Real samples derived from the CREMA-D.

Figure 3.6: Samples extracted from CREMA-D dataset.

Face cropping and alignment

As described in Section 1.5, in this work, we use the CREMA-D and RAVDESS datasets. Using the data included in these datasets, the same strategy that was followed in Chapter 2 for obtaining aligned faces from the initial videos is utilized. The extracted images are cropped and re-sized to a fixed resolution chosen to be 28 × 28. The main rationale for proceeding with the aforementioned size is that we need a reasonably small size images as input to our network G. In this way, the system is able to restrict also the complexity of the whole training process. In Figure 3.6(b), instances of the CREMA-D database after the whole cropping and alignment steps are shown.

3.2. Experimental phase for using Generative adversarial networks for emotion-aware domain adaptation

In this section, the empirical results for the evaluation of our approach are displayed. As mentioned before, we tested the proposed approach using the Crowd-sourced Emotional Multimodal Actors Dataset (CREMA-D) and the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) datasets. CREMA-D consists of facial and vocal emotional expressions in sentences spoken in a range of basic emotional states (happy, sad, anger, fear, disgust, and neutral). 7,442 clips of 91 actors with diverse ethnic backgrounds were rated by multiple raters in three modalities: audio, visual, and audio-visual. The RAVDESS is a validated multimodal database of emotional speech and songs. The database is gender balanced consisting of 24 professional actors, vocalizing lexically-matched statements in a neutral North American accent. Speech includes calm, happy, sad, angry, fearful, surprise, and disgust expressions, and the song segments contain calm, happy, sad, angry, and fearful emotions.

Furthermore, *progressive learning* process makes use of the VoxCeleb dataset as well. It should be noted that the CREMA-D and RAVDESS datasets, during the whole experimental phase, are balanced with the purpose of containing approximately the same amount of data samples for each class. Both datasets are split into four different subsets for both modalities. The splitting process is performed using in each subset, different subjects from CREMA-D and RAVDESS datasets, respectively.

- Firstly, the subset which is denoted as S_1, is utilized in order to train the classifier from the source domain $f_S(x_S)$ to perform facial expression recognition, with the purpose of using it for the semi-supervised GANs. This subset consisted of the 60% of the whole dataset. This is the classifier that is analyzed in Section 3.1.3.

- Secondly, the subset S_2 is used to calibrate the classifier (defined in Section 3.1.4) by calculating the p-values of conformal prediction (Equation 3.10). This subset consisted of the 20% of the whole dataset

- Thirdly, the samples in a subset denoted as S_3 are used for retrieving their p-values and for training the dacssGAN architecture (networks G, D and Q). This subset consisted of the 10% of the whole dataset

- Finally, the rest of the subjects, grouped as the subset S_4 are used for the testing of the whole approach. This subset consisted of the 10% of the whole dataset

3.2.1. Metrics for evaluating generated samples

Four different metrics have been applied to evaluate the quality of the generated samples [2]. These four metrics are: *augmentation scheme using real and generated data*, *inception score*, *fréchet inception distance* and finally *structural similarity index*.

Augmentation scheme using real and generated data: Firstly, the classification performance of the data augmentation scheme is assessed as the first metric. We fuse real audio samples from the original dataset (from the S_1 subset) with the ones generated from our architecture and we extract the classification performance of the expanded subset (from the S_4 subset). This is done with the aim of testing whether the generated spectrograms encompass efficiently the emotion recognition performance and whether they can improve the classification performance of the initial real dataset. In this sense, we can perceive data augmentation as an affordable alternative to easily expand audio-related datasets. The training for this scheme, in the end, has the double size of S_1. It

[2] For this purpose, a separate research item was picked up, in collaboration with the Master student Adam Eljasiak in his Master Thesis: “Can You Hear a Face? – Measuring The Quality of Spectrograms Generated by Generative Adversarial Network From Faces”, Maastricht University, Bachelor Thesis Conference, June 25, 2019

is important to note here that this approach should not be confused with the approach described in Section 3.1.6 (SpecAugment [33]).

Qualitative metrics Secondly, in an attempt to evaluate the quality of the generated samples, three metrics are established. The first one, the Inception Score (IS) [34] is utilized as an evaluation measurement. The approach is using an *Inception network* pre-trained on performing emotion recognition on real spectrogram datasets. Training is performed using data from dataset A, while the metric is applied to dataset B and vice versa. These datasets A and B in our case are CREMA-D and RAVDESS. This pre-trained model is applied to the generated samples in an effort to compare the conditional label distribution with the marginal label distribution. The marginal distribution inidicates how much variety there is in the generated samples, while the condition distribution whether there is high confidence in the class prediction in the generated samples (which can be interpreted whether each image looks distinctly like something). IS score combines these two measures in order to evaluate the quality of the generated samples. To combine these two factors, authors in [34], made use of the Kullback-Leibler (KL) divergence [35]. This is framed in the following equation:

$$IS(x) = \exp(E_x[KL(p(y|x)p(y))]) \tag{3.11}$$

where x is a generated sample and $p(y|x)$ represents the distribution of the classes for this sample. We want the posterior probability $p(y|x)$ to be highly predictable so to have low entropy. Furthermore, the marginal probability $p(y)$ is the overall distribution of classes across the sampled data and should have a high entropy which means the absence of dominating classes and a well-balanced training set. Altogether, the higher the IS score, the better the quality of the generated samples.

The second qualitative metric that is employed is the so-called Fréchet Inception Distance (FID) [36]. This metric compares the statistics of generated samples to real ones, instead of only evaluating generated ones. This approach is based on the same Inception model (previously used for the IS) and it is applied to the generated and real images to calculate the prediction using the Inception network. In more detail, FID could be framed as:

$$FID(\mathscr{X}_R, \mathscr{X}_G) = ||\mu_R - \mu_G|| + Tr(\Sigma_R + \Sigma_G - 2(\Sigma_R \Sigma_G)^{(\frac{1}{2})}) \tag{3.12}$$

where $\mathscr{X}_R$ and $\mathscr{X}_G$ are distributions of real and generated images (after the utilization of the Inception network) respectively and $\mu_{R,G}$ and $\Sigma_{R,G}$ correspond to the mean and covariance of the real and generated datasets, respectively. Lower FID values mean better image quality and diversity.

Finally, the Structural SIMilarity index (SSIM), which is described in [37] compares two images based on perceptual differences. It is expressed as a floating-point number ranging from -1 to 1, where 1 indicates exact similarity and -1 means a complete dissimilarity. The comparison is based on the following three measurements (between the two data samples): luminance, contrast, and structure. Calculating the SSIM for two grayscale images (x, y) is as follows:

$$L(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \tag{3.13}$$

$$C(x,y) = \frac{2\sigma_x\sigma_y + c_1}{\sigma_x^2 + \sigma_y^2 + c_1} \tag{3.14}$$

$$S(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \tag{3.15}$$

with $c_1 = (K_1 \cdot L^2)$, $c_2 = (K_2 \cdot L^2)$ and (with $K_1, K_2 < 1$) to be constants, L being the dynamic range of the pixel values (255 for 8-bit grayscale images), and $c_3 = \frac{c_2}{2}$. Variables μ, σ are the mean and variance operations for the input images. Finally the SSIM measure can be formulated as:

$$SSIM(x,y) = L(x,y)^{\alpha} \cdot C(x,y)^{\beta} \cdot S(x,y)^{\gamma} \tag{3.16}$$

where $\alpha > 0$, $\beta > 0$ and $\gamma > 0$ are the parameters used to adjust the relative importance of the three components. In our case, for simplicity reasons, as in [37], we defined these three values to be equal to 1 ($\alpha = \beta = \gamma = 1$). In this case, Equation 3.16 can be re-written as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_x\sigma_y + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{3.17}$$

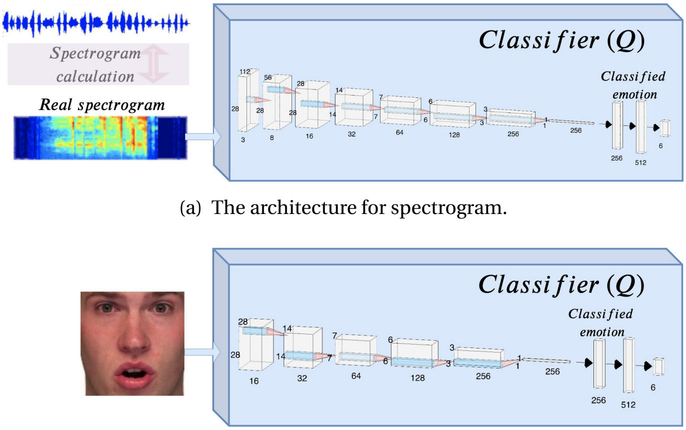
SSIM is calculated with a one-pixel offset over a 11 × 11 sliding window containing Gaussian weighting function, and then the final metric of an image is calculated as an average of the SSIMs of each window [37]. For color images, the average SSIM for each color channel is computed.

3.2.2. EXPERIMENTAL PROTOCOL

In the experimental phase, two basic scenarios are assessed:

- Firstly, the simple tuning of the whole architecture without any weight initialization (the so-called *moderate training* scenario) is applied.
- Secondly, the whole approach is utilized with the implementation of the weight initialization strategy (*progressive training* scenario) analyzed in Section 3.1.5.

For each of these two scenarios, the proposed architecture is evaluated in several steps. The objective of the current work is to evaluate the capacity and the amount of knowledge transferred between the source and target domains as well as to inspect the quality of the generated samples. For this reason, we employ the metrics described in the previous sub-section for each step of the experimental protocol.



(a) The architecture for spectrogram.

(b) The architecture for face.

Figure 3.7: The implemented CNN network for the classifier Q.

Baseline

Primarily, the emotion recognition classification performance of audio ($f_T(X_T)$) and visual ($f_S(X_S)$) domains is established as the baseline in the evaluated datasets (CREMA-D and RAVDESS) accordingly, without utilizing any domain adaptation strategy. For each of these datasets, we are employing the provided annotation information. The classifier employed to establish this baseline (for both modalities, audio and face) is a network similar to the one used as part of the dacssGAN topology and represented in Figure 3.7 as “Classifier” (network Q) for both cases (a more detailed description about the topology can be found in Section 3.1.2). The mere difference between them is two extra convolutional layers that are added to tackle the different sizes of the two domains (as it is also depicted in Figure 3.7). The rest of the layers and the implemented operations (*convolutions, max pooling, batch normalization* and *dropout*) in the network are the same. For training the baseline for face and audio, the S_1 subset is utilized while for testing purposes, we made use of the S_4 subset.

As a preliminary step, three different experiments are performed, evaluating three different spectrogram datasets with corresponding sizes of $28 \times 28, 56 \times 28$ and 112×28 pixels. In this analysis, it is found that the best results are obtained in the case of spectrograms with a resolution of 112×28 (see Table 3.4). This outcome is anticipated and is aligned with the assumption that the largest the spectrogram is (regarding its time duration) the better the classification results will be. From this point on, all the audio-related experiments are referring to this case [3].

The results of this baseline evaluation are compared with the evaluation of all the performed GANs approaches, and more specifically with the classification performance of the “GANs augmentation scheme using real and generated data” explained in Section 3.2.1. To ensure a fair comparison, for the baseline evaluation, an extra basic data aug-

[3] Normally, this analysis should have been included in the ablation study (Section 3.2.3), however, it is placed here to ensure a smooth transition in the corpus.

Table 3.4: Audio emotion recognition classification performance from spectrograms datasets of different size.

Spectrogram dimensionality	Performance
28 × 28	41.60%
56 × 28	46.77%
112 × 28	**49,34**%

mentation is performed using the SpecAugment technique (described in Section 3.1.6) in order to increase the number of samples to be equal with the baseline approach. Finally, both approaches consisted of the same amount of training data.

That being said, the classification performance using the CNN classifier is established as 49,34% and 64,50% for the audio and face modalities respectively for the CREMA-D dataset while for the RAVDESS dataset these values are established as 46,28% and 59,79% respectively. These aforesaid results will be noted henceforth as the *baseline* scores for the whole evaluation schema.

GANS TRAINING PROTOCOL

The next step of our evaluation schema is to train the whole GAN architecture (shown in Figure 3.2) by making use of the available training S_3 subset. To that end, three different cases are considered:

- **Supervised conditional GAN**: In this case, we have as input to the network G the samples from the target domain. Additionally, part of the input is the noise vector z and the conditional information vector c (represented by the label information).
- **Semi-supervised conditional GAN**: In this case, we explored the possibility of replacing the label information that is given as input to the generator G with the output of a classifier $f_S(X_S)$ trained using S_1 to perform emotion recognition on the source domain. The output of this classifier is a six-featured vector that contains, for each feature, the probability of the input sample to be derived from a specific emotional label.
- **Semi-supervised conditional CP GAN**: Finally, in this case, we explore the possibility of replacing the classification conditionality with the calibrated version provided using inductive conformal prediction. As was already mentioned, for calculating the inductive CP, we make use of the subset denoted as S_2 as a validation bucket that helped us to calculate the p-values for the denoted subset S_3 that played the role of the conditional information.

In the first case, we will evaluate the performance of the whole approach by using explicitly annotated information. However, as a step further, we want to show whether it is possible to train the whole network without explicitly using any label information. Hence, in the second case, an evaluation of whether the label information can be replaced by an emotion recognition classifier is examined. The outcome of this experiment is that the whole GANs architecture can be trained in a semi-supervised manner

Table 3.5: Results from the ablation study.

Classification performance for different experiments	
Omitting noise vector $z \in P_z$	41.40%
Omitting conditional vector c	39.83%
Omitting the source domain input $X_S \in D_S$	49.61%
All three inputs to network G	**52.52**%

by avoiding explicitly having annotation information as input. Finally, a way to improve this conditional information further is proposed.

3.2.3. Ablation study

Over the abovementioned evaluation approaches, an ablation study has been performed by appraising the performance of different architectures for the generator network G, different loss functions, different types of input for the network G, different sparsity levels concerning the data availability in the target domain and different algorithms for conformal predictions. The motivation for the ablation study is originated due to the overwhelming number of possible tests that can potentially be performed. Since the complexity of the approach is high, it is imperative to find elegant ways to reduce the total experimental phase duration. Therefore, an efficient way to prune the search space of the possible tests needs to be performed. The idea is to evaluate, in steps, the abovementioned framework setups, and each time, keep the best-performing combinations. Having found the best-performing combination for the loss function, input composition for the network G and architecture of the network during the ablation study in the case of moderate learning using supervised GANs we test the same variation in the case of progressive learning and the rest of the conditionality steps. It is important to note that the ablation study was performed solely on the CREMA-D dataset.

Firstly, a deep convolutional U-Net and a structure that resembles an encoder and decoder approach (EncDec) using dense layers are evaluated as the possible architectures for network G. Figures 3.3(a) and 3.3(b) depict these two architectures for network G. Moreover, Figure 3.2 reveals all the details for the EncDec architecture. Table 3.6 shows that U-Net does not outperform the simplified (EncDec) architecture. In this table, the performances of the three mentioned scenarios (supervised, semi-supervised, and semi-supervised CP GANs) when using CREMA-D are depicted. A possible assumption for this outcome is that U-Net, due to a very large number of free parameters to be trained, needed much more time and computational power (that was not practically possible given our setup) to be tuned properly than the simple EncDec approach. Henceforth, all the conducted experiments are using exclusively the EncDec architecture.

Additionally, the standard GANs optimization loss function is regularized by applying the L_1 norm (Equation 3.4). This loss during the training process compares pixel-wise the generated images with the real ones. The addition of L_1 is shown to be decisive for

Table 3.6: The classification performances of the target domain task T_T for the performed ablation study.

Baseline	49,34%		
Case	EncDec	U-Net	EncDec_without L1
Supervised GANs	52.52%	50.24%	38.11%
Semi-Supervised GANs-classifier	49.92%	50.12%	31.67%
Semi-Supervised GANs-CP	50.29%	50.09%	31.69%

the outcome of the dacssGAN topology (as displayed in Table 3.6). When this part is omitted from the optimization function, it is observed that the results are deteriorating not only classification-wise but also and most notably regarding the forfeiting of visual fidelity of the generated images. A possible explanation is that pixel to pixel distance as a loss is helping the calibration of network G and is forcing the generated distributions to be closer to the real ones. Therefore, for the rest of the conducted evaluations, L1 is integrated into all under test frameworks.

Moreover, we performed some further experiments wherein combinations regarding the input to the network G are formed and evaluated. In particular, as mentioned above, the input to the generator is the noise vector $z \sim P(z)$, the conditional information c and the source domain $x \in X_S$. Therefore, it is chosen to perform three different experiments where, in each case, one of the three inputs is omitted. This is done in an effort to evaluate the importance of each input to the network G. In particular, in the case that $x \in X_S$ is omitted, we are using a different EncDec architecture, where we drop the encoding part for the input $x \in X_S$. These results are displayed in Table 3.5.

- Firstly, it is found that by neglecting the label conditionality c, the classification performance is deteriorating, while *negative knowledge transfer* is experienced. This phenomenon is analyzed in Section 1.1.1 and is related to decreasing results when leveraging data from the source domain.

- Secondly, when the noise vector $z \sim P(z)$ is omitted, the mode collapse phenomenon re-occurred. During training, similar patterns in generated data re-occur and, therefore, including the noise vector helps alleviate this phenomenon.

- Finally, in the last combination, where the source domain $x \in X_S$ is omitted from the input, the obtained results deteriorate (with respect to the classification performance that is explained in Section 3.2.1), while convergence becomes slower.

As a conclusion, it is inferred that each of the inputs considered has a positive impact on the overall performance of the framework and, consequently, all of them are included in subsequent experimental procedures (see Section 3.2.4 for further details).

Another performed study is concerned with the conformal prediction conditionality. In this study, it is decided to monitor the performance of the whole approach by having different sparsity availability in the target domain. In this case, we decided to make use of the 50% and 20% of the initial datasets and extract the results for the Semi-supervised conditional CP GAN case (analyzed in Section 3.1.3). During this experiment, we want to determine how crucial is the availability of data for the proper training of the whole GAN approach. This study is explained in more details in Section 3.2.5.

It is important to note here that in the case of conformal prediction, we validate both the initial algorithm (explained in Algorithm 3) and the inductive conformal prediction (analyzed in Section 3.1.3). It is found that the performance of both techniques is similar, however, the computation complexity of the initial version of conformal prediction in comparison to inductive conformal prediction is large. Therefore, it is decided to stick with the results of inductive conformal prediction and introduce it as our semi-supervised technique. Furthermore, since the classification performance of the inductive conformal prediction had similar results with the results using a random split for the training and calibration subsets, it is decided that it is not necessary to employ a more sophisticated way to split the subsets.

To sum up the findings of the ablation study, the best-obtained observations that resulted from the related experiments are the following:

- Firstly, spectrograms of size 112×28 pixels are chosen to represent the audio modality.
- Furthermore, also the EncDec architecture is used (instead of U-Net). The basic topologies of these networks can be found in Section 3.1.2.
- It is shown also that the L_1 norm as part of the framework's loss function is necessary for the outcome of the results.
- Moreover, it is deduced, that all three inputs to the network G are crucial for the performance of knowledge transfer of our system.
- Finally, regarding conformal prediction, inductive CP is chosen as an algorithm, while random splitting is chosen as the way to split our datasets into the subsets $\{S_1, ..., S_4\}$ introduced at the beginning of this Section.

3.2.4. EXPERIMENTAL RESULTS AND DISCUSSION

The best-performing combinations extracted from the ablation study are employed to analyze the performance of moderate and progressive learning and also by applying the three conditionality approaches (in total, six possible configurations are evaluated). Firstly, the results for the moderate learning case are illustrated in the following paragraphs. For both scenarios, our analysis is divided into two inspections, namely: quantitative and qualitative results.

MODERATE LEARNING

Firstly, the classification performance and the qualitative metrics of moderate learning are presented. In this case, the training of the whole GANs architecture occurs once by using the emotion-aware datasets (CREMA-D or RAVDESS) and without any pre-training of the architecture's networks. Furthermore, the visual results (generated spectrogram samples) extracted when performing the moderate learning are presented as well.

QUANTITATIVE RESULTS

In this Section, the quantitative results, obtained during the evaluation phase for all the three cases of the experimental protocol (supervised conditional GAN, semi-supervised conditional GAN and semi-supervised conditional CP GAN) are described. In Table 3.7, the performance regarding all four metrics of all the aforementioned cases of the experimental protocol is presented (Section 3.2.1). Firstly, in the initial GAN case, the so-called supervised conditional GAN is evaluated. In this case, the approach reached the best performance (52.52% for CREMA-D and 47.11% for RAVDESS). The same behavior is demonstrated by the results extracted using the other evaluation metrics, namely, the IS, FID and SSIM (see Table 3.7).

However, in the current work, our main effort is focused on the much more interesting semi-supervised case where the goal is to generate annotated audio samples coming from rich but not necessarily annotated video samples. As a subsequent step, the semi-supervised conditional GAN is evaluated. In this case, the obtained results are 49.92% for CREMA-D and 46.23% for RAVDESS.

Finally, the evaluation of the semi-supervised conditional CP GAN is conducted. The extracted results are 50.29% for CREMA-D and 46.55% for RAVDESS and are slightly better than the baseline results for both datasets. The rationale behind this improvement in the results (in contrast to the previous case) is mainly that, after the application of conformal prediction, the calculated p-values contain better-distributed confidences in the rest of the labels in comparison to the confidences derived using merely the classifier.

Table 3.7: Classification, FID and IS performance for all experimental protocol scenarios (moderate GANs).

Case	CREMA-D				RAVDESS			
	Classification	FID	IS	SSIM	Classification	FID	IS	SSIM
Bas.	49.34%				44.73%			
Sup. GANs	**52.52%**	**59.44**	**2.16**	**0.77**	**47.11%**	**49.77**	**2.21**	**0.90**
S-Sup.GANs	49.92%	60.13	2.01	2.01	46.23%	50.33	2.05	2.05
S-Sup.GANs-CP	50.29%	60.10	2.00	2.00	46.55%	49.95	2.01	2.01
DML approach	41.58%	-	-	-	37.66%	-	-	-

Further stimulating observations can be found in the confusion matrices (CM) extracted when using the CREMA-D dataset for the baseline case (displayed in Figure 3.8(a)), supervised GAN approach (Figure 3.8(b)) and semi-supervised CP (Figure 3.8(d)). Firstly, all figures show that the considered emotions are well discriminated since the diagonal elements of the matrix have (in all cases) the highest classification performances. Also, in all three cases, it is evident that the strongest captured emotion is anger. It is noteworthy that this behavior is consistent with the study performed in [38] concerning the human accuracy in audio emotion recognition, where it was stated that the best-performing emotion label was anger for the CREMA-D dataset. In [40], the authors observed a similar finding when using also other state-of-the-art audio-based datasets. In the case of the data augmentation schema of the supervised case, in Figure 3.8(b), we can observe that, while the anger emotion performance drops, the efficiency of the rest of the emotions roughly increases and, thus, more uniform recognition is achieved by making use of supervised learning.

Altogether, a captivating observation in this chapter can be extracted from the comparison between the results of the best-performing approach (Supervised GANs, see Table 3.7) with the best-obtained results from the previous Chapter (that can be seen in Table 2.1). We extracted two main observations:

- Firstly, both the baseline emotion recognition for CREMA-D and RAVDESS datasets are improved when using CNN architecture like the one in Figure 3.4 instead of simple SVM classification.
- Secondly, the experienced transfer knowledge is more acute in the case of the approach based on Generative Adversarial Networks.

To ensure a fair comparison between the results of these two different methods, we are employing the same datasets. The training and test sets are partitioned in the same manner. Furthermore, regarding the pre-processing steps, we followed the same process regarding the face extraction, by performing face detection, alignment and cropping. However, for the feature extraction process, both for face and audio we could not follow the same process, since the GANs network input and output are raw images instead of features.

Table 3.8: Classification, FID and IS performance for all experimental protocol scenarios (progressive GANs). The last row contains the results from the DML approach from the previous Chapter from Table 2.1. We add this information for comparison reasons.

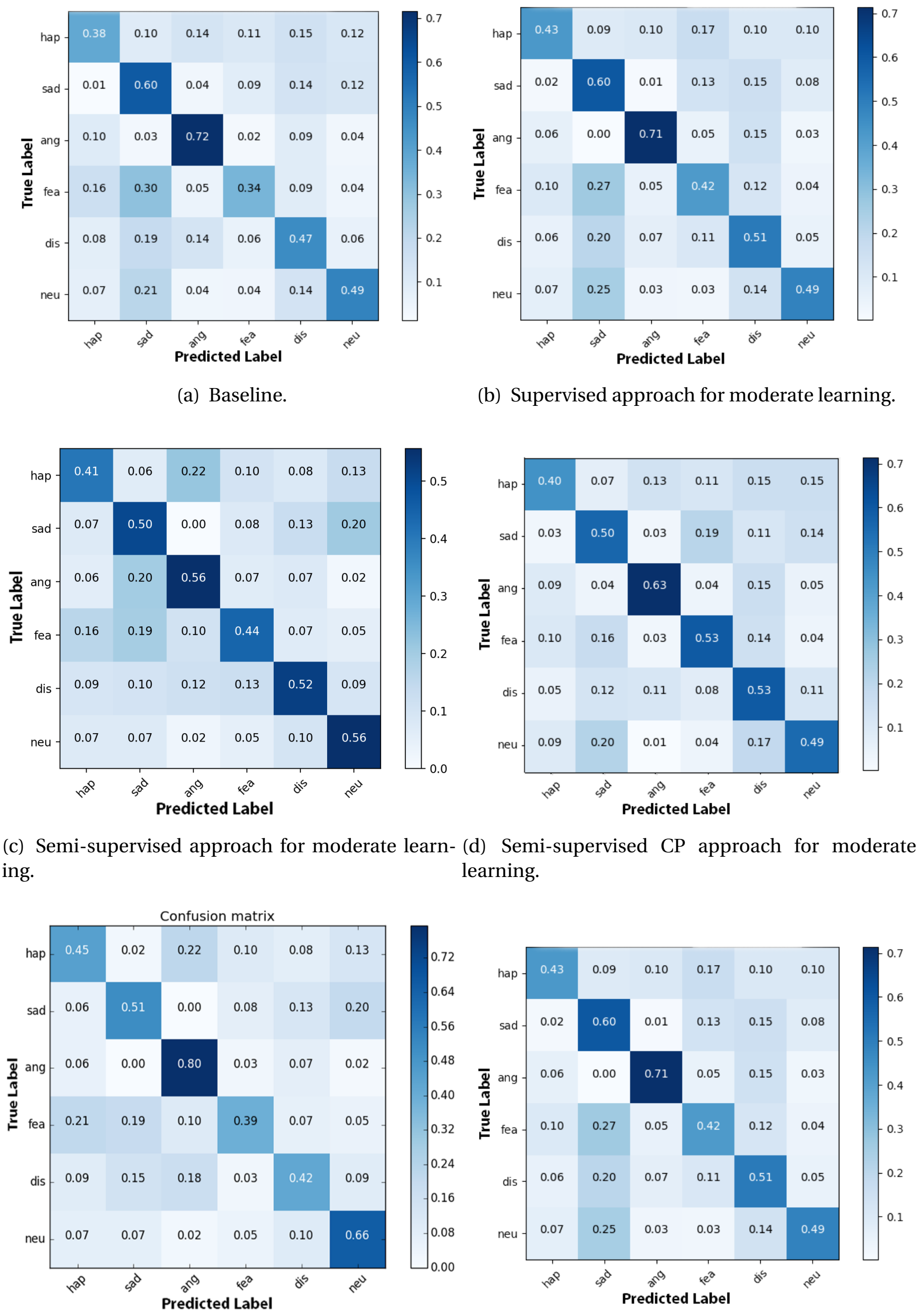
Case	CREMA-D				RAVDESS			
	Classification	FID	IS	SSIM	Classification	FID	IS	SSIM
Bas.	49.34%				44.73%			
Sup. GANs	**53.71**%	57.22	2.16	**0.86**	**47.55**%	49.59	**2.15**	0.90
s.Sup. GANs CP	50.40%	59.91	2.22	0.81	46.77%	48.98	2.05	0.95
DML approach	41.58%	-	-	-	37.66%	-	-	-

QUALITATIVE RESULTS

With respect to the qualitative analysis, visual generated samples from the performed experiments that are introduced in the ablation study, as well as the results for the best-performing architectures regarding moderate learning (which are also evaluated in the quantitative study) are displayed. These experiments constitute the so-called qualitative research that is performed for this chapter.

Firstly, the results from the mentioned under investigation frameworks introduced in the quantitative study are rendered. In Figure 4.14(a) generated samples from the supervised conditional GANs are illustrated. Whereas, Figure 4.14(b) represents the generated spectrograms that are derived from the semi-supervised conditional GANs. In Figure 3.11(a), the results in the scenario where conformal prediction is used as part of the input to the networks *G* are visualized.

From all these figures that represent the generated spectrograms for all three cases (described in 3.2.2), we can deduce that the dacssGANs approach in all three configurations managed to approximate the target domain in each case while the visual results



(a) Baseline.

(b) Supervised approach for moderate learning.

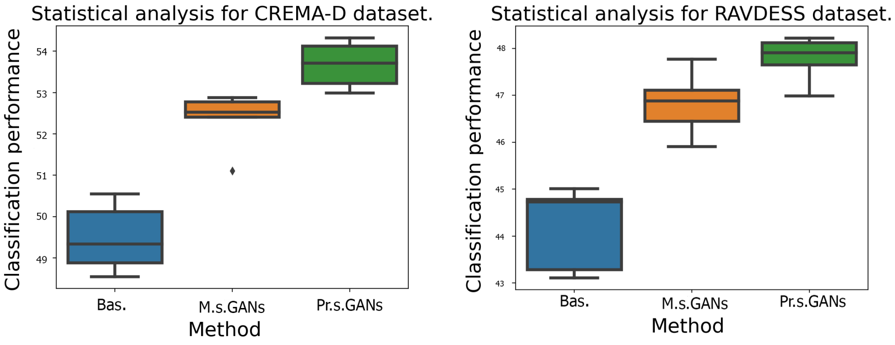
(c) Semi-supervised approach for moderate learning.

(d) Semi-supervised CP approach for moderate learning.

(e) Supervised approach for progressive learning.

(f) Semi-supervised approach for progressive learning.

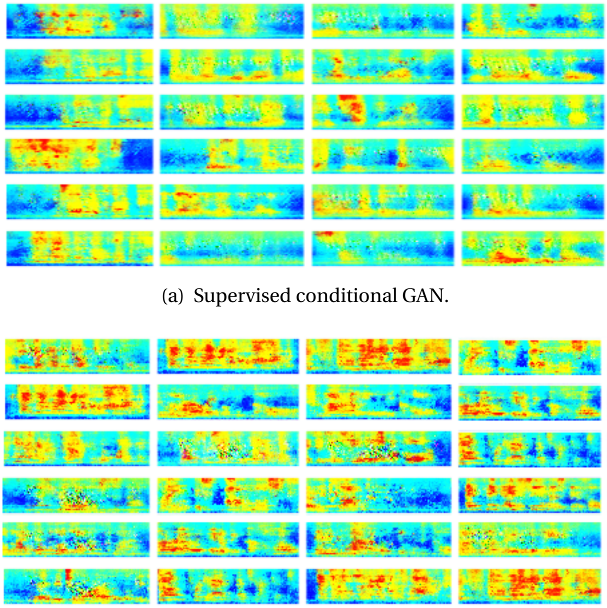
Figure 3.8: Confusion matrix for the emotion recognition classifier through audio spectrograms using different approaches for the six distinct emotion in CREMA-D.



(a) Methodologies for the CREMA-D .

(b) Methodologies for the RAVDESS .

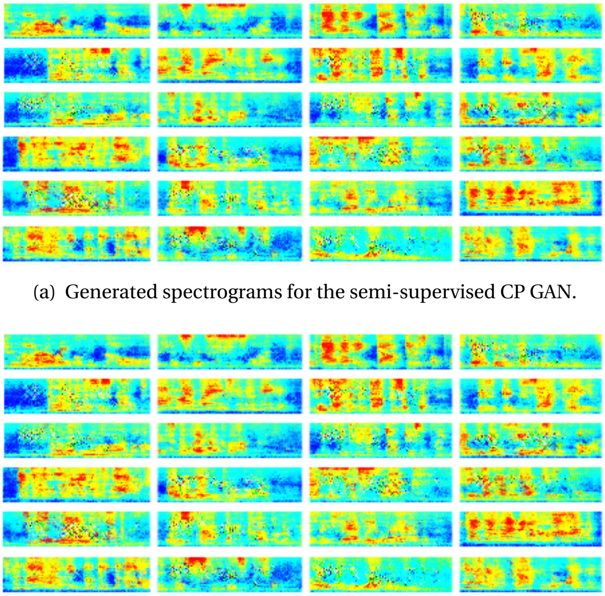
Figure 3.9: Std error bars for the baseline, the moderate and progressive GANs methods for the four different subsets of the set S_4.



(a) Supervised conditional GAN.

(b) Semi-supervised conditional GAN.

Figure 3.10: Generated spectrograms of the GAN approach calibrated using the CREMA-D dataset.



(a) Generated spectrograms for the semi-supervised CP GAN.

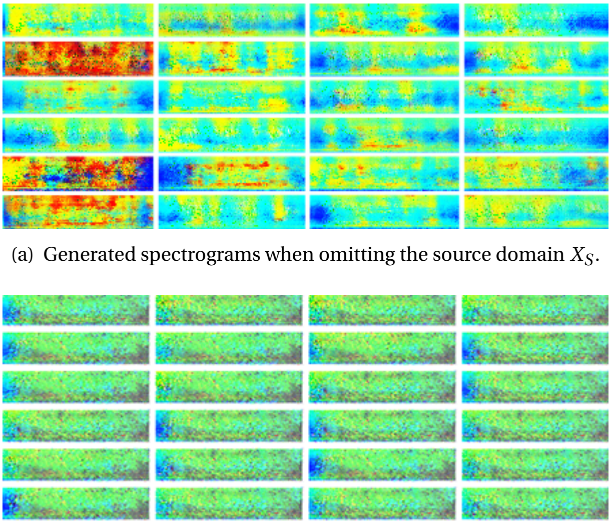
(b) Generated spectrograms for the semi-supervised CP GAN..

Figure 3.11: Generated samples of the GAN approach that calibrated using CREMA-D dataset.

can be considered faithful representations of the target distribution domain. The approaches in all cases managed to track meaningful patterns that represent similar patterns that are found in the original spectrograms. However, from the mere visual inspection, it is not possible to infer deductions regarding a visual comparison between these three categories. Furthermore, it is difficult to extract conclusions concerning the wealth of affect context in the generated results. This comparison is mainly established using the extracted results from the quantitative analysis.

With regard to the conducted experiments for the ablation study, qualitative results are displayed here. Firstly, the generated spectrograms for the experiment conducted by omitting the source domain X_S from the network G input are rendered in Figure 3.12(a). As it has already been discussed in Section 3.2.3, in this case, we *negative transfer* knowledge is observed since the performance of the fused audio emotion classification is lower than the baseline. This behavior is validated also by the visual results where it can be observed that there is a lot of unexpected artifacts and noise in the generated samples.

Regarding the training of the whole approach without the implementation of the regularization L_1 norm in the loss function of the framework (see Section 3.1) the visual results are depicted in Figure 3.12(b). In this case, the whole approach failed completely to capture any meaningful pattern with respect to the audio signal spectrum and led to very noisy visual results.



(a) Generated spectrograms when omitting the source domain X_S.

(b) Generated samples when omitting the L_1 norm from the optimization function.

Figure 3.12: Generated samples of the GAN approach for the ablation study (moderate GANs).

PROGRESSIVE LEARNING

Furthermore, the evaluation of progressive learning is performed. In this case, as it is analyzed in Section 3.1.5, the training of the whole framework is performed in two-steps. Firstly, an unlabelled dataset is applied to initialize the weights of the frameworks and, secondly, two emotion-annotated datasets are applied to calibrate the weights emotion-wise. Again the same two categories of investigation are deployed, namely the quantitative and qualitative studies.

QUANTITATIVE ANALYSIS

Regarding the quantitative results, we tested the progressive training of GANs for both datasets and all three possible scenarios. The obtained results are 53.71% for CREMA-D in the supervised scenario, 50.40% for the semi-supervised one, and 52.55% in the case of CP. For the RAVDESS dataset, the performance obtained is 47.55%, 46.77% and 47.37% respectively. In Table 3.8, the performance for this case is displayed. Moreover, figures 3.8(e) and 3.8(f) depict the confusion matrices for the supervised GANs and semi-supervised GANs for the progressive learning. From the quantitative results, it is obvious that by using this approach we are able to improve the performance of the algorithm for at least 1.2% for the CREMA-D and 0.5% for RAVDESS in the supervised GANs case between moderate and progressive learning. Regarding the divergence between the two approaches in the semi-supervised CP case, the improvement is rather marginal and we could not extract any safe observations.

In an attempt to better understand the performance of the proposed semi-supervised CP GAN in comparison with the baseline, a statistical analysis has been performed. In this analysis, the statistical results (mean and standard deviation) for the classification performance are extracted from different folds (each fold containing different subjects of the dataset). The results obtained for both datasets are illustrated in Figure 3.9. From this figure, we can observe that, in the case of RAVDESS, the standard deviation is narrow and there exists a significant difference in the mean value. In the case of CREMA-D, the deviation from the mean is higher which could be attributed to the bigger variety of the subjects included in the dataset.

Since performing cross-validation was not feasible due to the high computational power, it was decided to test our models solely with the test set by splitting this subset (S_4) into 4 different subsets (that contains different subjects) and then we extracted the prediction performance per subset. The average and standard deviation (also the outliers) of this prediction are depicted in Figure 3.9. In this way, we were able to assess and monitor the performance of the model from different subjects in the dataset and hence to perform the evaluation in a more efficient way.

Qualitative analysis

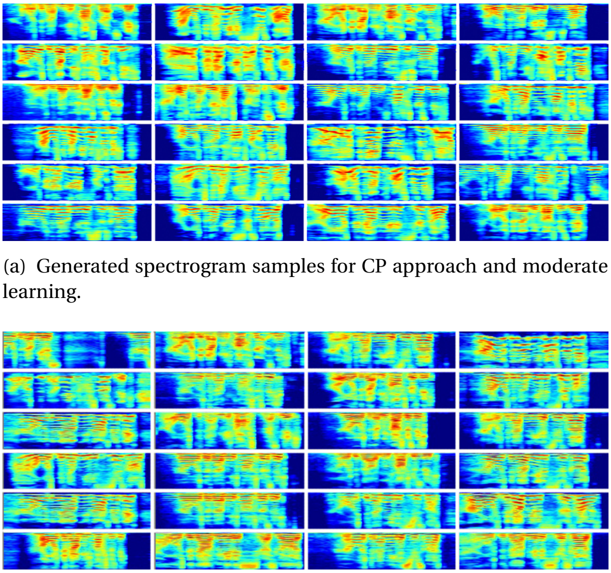
To supplement the quantitative results, qualitative evaluation of progressive training is discussed as well. As can be seen in Figure 3.13, the quality of the visual results is high, and visually approximating the real spectrograms. It is obvious, especially for the case of the RAVDESS dataset, that the generated results after our subjective visual inspection keep similar patters in the generated results in comparison with the real samples, as well as, some color information. However, from the extracted visual results it is difficult to analyze about the affect load of these samples merely using visual inspection.

Therefore, the quality of the extracted images could not be assessed directly. Nonetheless, a comparison between all our applied methods except the baseline (moderate and progressive GANs) using the three introduced quality metrics (see Tables 3.7 and 3.8) can be examined. It is obvious that the best extracted results for both datasets can be found in the case of the progressive GANs.

3.2.5. Training process evaluation

In order to examine the performance of the training process of our dacssGANs algorithm, we visualize the loss function of the *G*, *D* and *Q* networks, as well as, the IS and FID scores during the training procedure. Figures 3.14(a) and 3.14(b) render the loss function of the three networks for the CREMA-D and RAVDESS datasets, respectively. Figures 3.15(a) and 3.15(b) display the IS and FID scores for the CREMA-D and RAVDESS datasets (using the test subset S_4) and also the evaluation performance for the sparsity test. The aim of this experiment is to investigate the training performance of the method when using different *sparsity* levels (20%, 50% and 100% of the initial training subset) for the training data of the whole GANs architecture for the supervised case. In Chapter 2, the sparsity test is performed to illustrate the robustness of the new DML distance with different sparsity levels. In this chapter, it is performed to depict the relation of GANs architecture with the available datasets.

As aforementioned, deep learning approaches require massive amount of data. Hence, with this experiment, we wanted to identify the importance of the size of our training datasets. From the correspondent figures, we can deduce that during training, even using 50% of the dataset leads to noticeably poorer results in qualitative perfor-

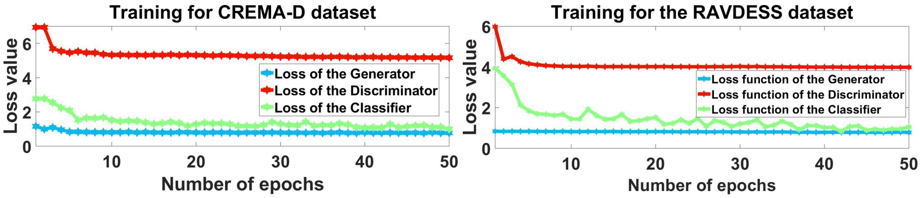


(a) Generated spectrogram samples for CP approach and moderate learning.

(b) Generated spectrogram samples for CP approach and progressive learning.

Figure 3.13: Generated visual results when the framework was trained using the RAVDESS dataset.

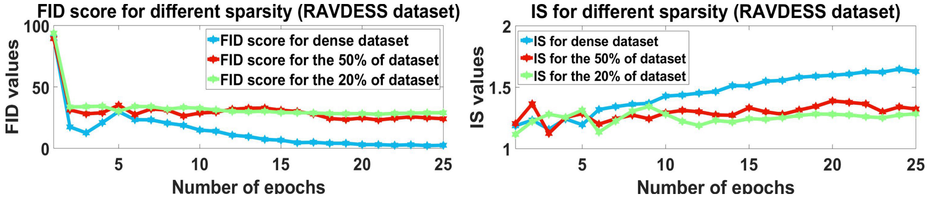
mance than using the whole dataset. On the whole, an evident observation during the training procedure is that the approach steadily converges and the quality of the visual results improves as the number of epochs increases. Finally, regarding the complexity of our approach, all the conducted experiments are performed by using the same hardware (as mentioned before). For a single experiment, the total time duration is approximately 82 and 102 hours (for RAVDESS and CREMA-D respectively) that is equated approximately to 100 epochs. We chose to proceed with this number of epochs for all the conducted experiments to reduce the time of our experiments.



(a) CREMA-D dataset.

(b) RAVDESS dataset.

Figure 3.14: Loss function during the training procedure of dacssGANs CP (for moderate learning).



(a) Féchet Inception Distance.

(b) Inception Score.

Figure 3.15: FID and IS values during the training procedure of dacssGANs CP (for moderate learning).

3.3. CONCLUSION

The first research question posed in the current dissertation is related to whether it is possible to transfer knowledge between facial expressions to the audio information from the same sequences. In the previous chapter, an approach that was based on distance metric learning was introduced to accommodate the domain adaptation task.

In this chapter, we introduce a new approach to studying the cross-modal relationships between audio and visual modalities, which is based on Generative Adversarial Networks. In particular, the name of the approach is *domain adaptation conditional semi-supervised Generative Adversarial Networks or dacssGANs*. The core objective of the approach is to tune the weights of the generator network G which will generate samples that are distributed from the target domain (audio domain in the presented case) and represent specific emotion states. The input to network G is a random noise vector $z \in Z$, fused together with samples from the source domain ($x \sim X_S$) corresponding to emotionally expressive faces and with conditional information (c) corresponding to the nature of the emotion. It is important to note here that the source domain input has a static nature since it corresponds to a single frame. The conditional information is calculated using a semi-unsupervised technique called Conformal Prediction. We proposed the use of these confidence values, instead of labels, as a softer and more reliable manner to introduce knowledge into the generator. Furthermore, we investigated the efficiency of a network Q that works as a classifier in the target domain and calibrates the generated samples from network G.

Therefore, in this chapter, we deal with the third and fourth research questions posed in Section 1.6. The first one is concerned with whether domain adaptation using GANs could be used to uncover audio-visual relationships. Furthermore, the second posed question is associated with the progressive training of the topology and whether this process can improve further the domain adaptation. In the progressive learning schema, a weight initialization process is introduced, which uses an abundance of unlabelled audio-visual information to tune the weights of our G and D networks. By having initialized the weights to map audio-visual relationships (without using any explicit emotion information), the next step is to calibrate the weights emotion-wise.

On this ground, an investigation regarding the improvement of the knowledge transfer approach in comparison with the work done in Chapter 2 is performed. In particular, we tried to figure out whether we can improve the knowledge transfer by using the GANs scheme than using the DML technique. Furthermore, an evaluation that is associated with the improvement of the approach under the progressive learning process is also performed.

The efficiency of our approach is established during the experimental phase. We propose a data augmentation schema dealing with the hypothesis that it is possible to generate emotionally enhanced audio data from visual cues. As a further step, it is shown that by using progressive learning we are able to improve the results even further and, as a consequence, the knowledge transfer performance. The introduced measurement for the knowledge transfer implements the classification performance as an evaluation tool (similarly to the DML approach from Chapter 2). Furthermore, some metrics that measure the quality of the generated images are introduced as well. In comparison to the previous chapter, the approach adopted in this chapter improves the knowledge transfer results and generalizes better in the challenging CREMA-D and RAVDESS datasets. Moreover, it has been shown that the deep learning architecture, when combined with spectrograms (as audio information) performs better than using simple SVMs on the extracted features (analyzed in Chapter 2). In summary, our main observations extracted from this chapter are:

- We confirmed that performing domain adaptation using deep learning architectures, and in particular GANs, led to an increased transfer of knowledge.
- Furthermore, we can replace the supervised conditional information of the approach by employing a semi-supervised methodology based on the CP algorithm, without decreasing the performance significantly.
- Finally, we have shown that it is possible to increase the transfer of knowledge with the *progressive training* of the whole network.

However, some noticeable limitations that are faced during the implementation of the proposed approach are the following:

- The introduced network led to better classification results (for the fused real and generated data used jointly in training) compared to the knowledge transfer using DML. However, its sophisticated nature required a very complex deep learning architecture. In order for this architecture to be properly tuned, a training process that ranging from 3-5 days took place. Therefore, the time complexity is significantly higher than that of the previous light-weighted DML approach.
- Secondly, the whole approach did not take into account the temporal nature of facial expressivity within the video. In fact, in the whole approach, we map the middle face of a video sequence with the corresponding, expected audio spectrogram. However, in this manner, a lot of useful information concerning the temporal dynamics of the face (such as muscle movements) goes astray. Therefore, a mechanism to study these dynamics should be provided.
- Finally, while the generated results are quite promising, from our subjective observations it is deduced that we need to proceed using a more efficient training process in order to obtain visually better results.

REFERENCES

[1] P. V. Rouast, M. T. P. Adam and R. Chiong, Deep Learning for Human Affect Recognition: Insights and New Developments, IEEE Transactions on Affective Computing, 2018.

[2] H.Cao, D.G.Cooper, M.K.Keutmann, R.C.Gur, A.Nenkova and R.Verma, CREMA-D: Crowd-Sourced Emotional Multimodal Actors Dataset, IEEE Transactions on Affective Computing, Volume 5, Number 4, Pages: 377–390, 2014.

[3] S.R.Livingstone and F.A.Russo, The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English, Volume 13, Number 5, Pages: 1—35, 2018.

[4] A.Dhall, R.Goecke, S.Lucey, and T.Gedeon, Collecting large, richly annotated facial expression datasets from movies, IEEE Transactions of MultiMedia, Vol. 19, Iss. 3, 2012.

[5] I.J.Goodfellow, J.P.-Abadie, M.Mirza, B.Xu, D.W.Farley, S. Ozair, A.Courville and Y. Bengio, Generative Adversarial Networks, 27th conference on Advances in Neural Information Processing Systems (NIPS), 2014.

[6] A.Radford, L.Metz and S.Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, International Conference on Learning Representations (ICLR), 2016.

[7] M.Mirza and S.Osindero, Conditional Generative Adversarial Nets, Computing Research Repository (CoRR), 2014.

[8] O.Ronneberger, P.Fischer and T.Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015.

[9] P.Isola, J.Yan, Z.Tinghui, Z.Alexei and A. Efros, Image-to-Image Translation with Conditional Adversarial Networks, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[10] G.Shafer and V.Vovk, A Tutorial on Conformal Prediction, The Journal of Machine Learning Research archive Volume 9, Pages 371-421, 2008.

[11] V.Vovk, Conformal prediction, Algorithmic Learning in a Random World, Springer US, Pages: 17–51, 2005.

[12] A.Nagrani, J.S.Chung and A.Zisserman, VoxCeleb: a large-scale speaker identification dataset, in Interspeech, 2017.

[13] S.Matiz, K.E.Barner, Inductive conformal predictor for convolutional neural networks: Applications to active learning for image classification, Volume 90, Pages: 172–182, 2019.

[14] X.Liu, Y.Zou, L.Kong, Z.Diao, J.Yan, J.Wang, S.Li, P.Jia and J. You, Data Augmentation via Latent Space Interpolation for Image Classification, International Conference on Pattern Recognition (ICPR), 2018.

[15] S.Sahu, R.Gupta and.E.Wilson, On Enhancing Speech Emotion Recognition using Generative Adversarial Networks, Interspeech, 2018.

[16] X.Wang and A.Gupta. Generative image modeling using style and structure adversarial networks, European Conference on Computer Vision (ECCV), 2016.

[17] D.Pathak, P.Krahenbuhl, J.Donahue, T.Darrell, and A.A. Efros. Context encoders: Feature learning by inpainting, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[18] A.Odena, C.Olah and J.Shlens, Conditional Image Synthesis with Auxiliary Classifier GANs, Proceedings of the 34th International Conference on Machine Learning, (PMLR), 2017.

[19] M.Lee and J.Seok, Controllable Generative Adversarial Network, Published in ArXiv, 2017.

[20] J.Zhu, T.Park, P.Isola and A.A.Efros, Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, IEEE International Conference on Computer Vision (ICCV), 2017.

[21] V.N.Balasubramanian, A.Baker, M.Yanez, S. Chakraborty, and Sethuraman Panchanathan, PyCP: An Open-Source Conformal Predictions Toolkit.

[22] E.Cvejic, J.Kim and C.Davis, Prosody off the top of the head: Prosodic contrasts can be discriminated by head motion, Journal of Speech Communication, Volume 52, Number 6, Pages: 555—564, 2010.

[23] L.Metz, B.Poole, D.Pfau and J.S.Dickstein, Unrolled Generative Adversarial Networks, International Conference on Learning Representations (ICLR), 2017.

[24] I.Goodfellow, NIPS 2016 Tutorial: Generative Adversarial Networks, Conference on Neural Information Processing Systems (NIPS), 2016.

[25] T.Salimans, I.Goodfellow, W.Zaremba, V.Cheung, A.Radford and X.Chen, Improved Techniques for Training GANs, Conference on Neural Information Processing Systems (NIPS), 2016.

[26] I.Tolstikhin, S.Gelly, O.Bousquet, C.Johann, S.Gabriel and B.Schölkopf, AdaGAN: Boosting Generative Models, Advances in Neural Information Processing Systems (NIPS), 2017.

[27] J. Duchi, E. Hazan and Y. Singer, Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research, Volume 12, Pages: 2121-–2159, 2011.

[28] K.He, X.Zhang, S.Ren and J.Sun, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, IEEE Computer Vision and Pattern Recognition (CVPR), 2015.

[29] C.Donahue, J.McAuley and M.Puckette, Adversarial Audio Synthesis, International Conference on Learning Representations (ICLR), 2019.

[30] C.Donahue, J.McAuley and M.Puckette, Synthesizing Audio with GANs, International Conference on Learning Representations (ICLR), 2018.

[31] P.Yenigalla, A.Kumar, S.Tripathi, C.Singh, S.Kar and J.Vepa, Speech Emotion Recognition Using Spectrogram & Phoneme Embedding, in Interspeech, 2018

[32] H.M.Fayek, M.Lech and L.Cavedon, Towards Real-time Speech Emotion Recognition using Deep Neural Networks, 9th International Conference on Signal Processing and Communication Systems (ICSPCS), 2015.

[33] D. S. Park, W. Chan, Y. Zhang, C.C. Chiu, B. Zoph, E. D. Cubuk, Q. V. Le, SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition, Interspeech, 2019.

[34] T.Salimans, I.Goodfellow, W.Zaremba, V.Cheung, A.Radford and X.Chen, Improved Techniques for Training GANs, In Advances in Neural Information Processing Systems (NIPS), 2016.

[35] KS. ullback, R.A. Leibler, On information and sufficiency, Annals of Mathematical Statistics, Volume 22, Number 1, Pages: 79—86, 1951.

[36] M.Heusel, H.Ramsauer, T.Unterthiner, B.Nessler and S.Hochreiter, Gans trained by a two time-scale update rule converge to a local Nash equilibrium, In Advances in Neural Information Processing Systems (NIPS), (2017).

[37] Z.Wang, A.C.Bovik, H.R.Sheikh and E.P.Simoncelli, Image Quality Assessment: From Error Visibility to Structural Similarity, EEE Transactions on image processing, Volume 13, Number 4, 2004.

[38] H.Cao, D.G.Cooper, M.K.Keutmann, R.C.Gur, A.Nenkova and R.Verma, CREMA-D: Crowd-Sourced Emotional Multimodal Actors Dataset, IEEE Transactions on Affective Computing, Volume: 5, Number: 4, Pages: 377–390, 2014.

[39] E.Ghaleb, M.Popa, E.Hortal and S.Asteriadis, Multimodal Fusion Based on Information Gain for Emotion Recognition in the Wild, IEEE Intelligent Systems Conference 2017.

[40] E.Ghaleb, M.Popa, E.Hortal and S.Asteriadis, Multimodal Fusion Based on Information Gain for Emotion Recognition in the Wild, IEEE Intelligent Systems Conference 2017.

4

Temporal conditional Wasserstein GANs for studying cross-modal relationships

The research presented in this chapter is supported by the following publication:

- *C.Athanasiadis, E.Hortal, S.Asteriadis, Temporal wGANs for investigating audio-visual affect ties,* 9^{th} *International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW), 2021.*

The core insights established in Chapter 3 were associated with audio-visual transfer of knowledge when using a strategy based on Generative Adversarial Networks (GANs). In particular, an image-to-image approach (a facial frame to spectrogram) was investigated for studying the cross-modal emotion recognition. Having as input to the proposed GANs architecture ("dacssGANs") a vector of noise, some conditional information and an image from the source domain X_S, the task of the proposed GANs is to approximate the distribution of the target domain X_T (that is represented again by images-spectrograms). The annotations could be either given directly or can be calculated using an auxiliary classifier. Hence, the whole approach regards a "supervised" (or "semi-supervised") "image-to-image" approach.

This approach has been shown to be robust, while performance in terms of domain adaptation was validated using several state-of-the-art metrics. However, several deductions were extracted through the experimental phase:

- Firstly, the duration of the training process can take approximately 5-6 days (120-144h). The conducted experiments were applied in a Titan XP GPU and the total number of training epochs was 100. Tuning the whole architecture was proven to be a very challenging task since each time a parameter needs to be tuned, a new training process must be executed, which entails another additional 120-144 computer hours to obtain new measurements of the framework performance. Thus, the whole architecture needs a large amount of time to be tuned properly and find the performed parameters.

- Secondly, the loss function considered in Chapter 3, while it performs well when approximating distributions, leaves plenty of space for improvement, especially concerning training stability and the convergence to optimal solutions and finding the Nash equilibrium. To this end, related approaches (such as Wasserstein distance [1]) can be incorporated with the current approach to improve training stability and, as a consequence, to improve the knowledge transfer.
- Furthermore, from the experimental phase observations from Chapter 3, while the quality of the generated images is quite convincing, still, we believe that there is a potential space for improvement. GANs in the vanilla architecture has shown to be an effective approach in regard to approximating visual distributions. However, several state-of-the-art architectures are allegedly able to perform better training (by improving the train stability) for their networks and enhance the quality of the generated results. Hence, by using these new approaches in training GANs, we can potentially improve the performance of the knowledge transfer and the quality of the generated samples.
- Finally, as said before, the study conducted in Chapter 3 is concerned with the static image-to-image audio-visual domain adaptation. This approach presents a main constraint. While the constructed spectrograms entail temporal information, the input X_S, represented by the face modality, is composed of a static frame. However, as research in cognitive psychology suggests [2], studying the audio-visual relationships for emotion recognition should entail the temporal relations that govern face and audio. In particular, a study that tries to associate the movements of muscles with temporal information is researched in [2]. On this ground, while by using static image-to-image GANs some promising results were extracted, we decided to proceed with further investigation into the temporal connections between the face and audio using a modified GANs architecture.

Contrary to the work done in the previous chapter, in this work, by summarizing the previous deductions, we mainly focus on studying two research topics: Firstly, we study the implementation of an alternative loss function for the whole GANs training instead of the conventional loss implemented in Chapter 3. Secondly, the implementation of temporal information as input to the network G to represent the source domain is performed instead of using raw images (as we did in the previous chapter). Hence, this is the motivation behind the title of this chapter.

Regarding the first research topic of this chapter, the applied new loss function is based on Earth-Mover distance (or as alternatively called, Wasserstein distance). The motivation behind using this function is two-fold:

- Firstly, this is done to examine whether the Wasserstein loss can improve further the performance of the conducted knowledge transfer and the quality of the generated samples. Our motivation is the fact that Wasserstein loss was proved to be a much more stable approach regarding GANs training convergence and can lead to better performance regarding approximating distributions [1]. Therefore, we pose the assumption that it could be shown to be suitable and efficient regarding our cross-modal study as well.

- The second motivation behind this loss is associated with several indications that Wasserstein loss can speed up the training process. This behavior is analyzed in [1] and [6] and is an effect of the training stability of this approach described in the previous point.

Regarding the second research topic and the implementation of temporal features, the stimulation behind this choice is motivated firstly, with the intuition that the human brain processes audio-visual signals in a temporal manner [3]. On this ground, several studies in the field of cognitive psychology displayed temporal connections between the two modalities [3][4], especially within the emotion context [2].

To study the temporal dynamics between the two modalities, it is decided that temporal features will be extracted from the source domain X_S which represents the face modality. Furthermore, these extracted features will be added as an input to the whole GANs network as conditional information (bound together again with a random noise vector $z \in P(z)$ and the annotations for each video c). For the sake of efficiency, and to keep the complexity of the whole approach in low-level, it is decided, as a first step, to extract the temporal information and, as a second step, to train the whole GANs architecture by giving the extracted temporal features as input to the network G. Training everything on an end-to-end basis would have increased dramatically the complexity of the whole training process and the demand for more resources (RAM memory, CPUs and GPUs).

With respect to the temporal feature extraction, a pre-processing step is performed in an attempt to extract meaningful representations for both modalities. Toward this end, the same steps that were performed in the previous chapters are conducted also here (are displayed in Section 3.1.6).

Regarding the face modality, face detection, cropping, and alignment are performed on the video clips. Having extracted the faces and spectrograms from each video, several steps regarding temporal feature extraction are performed to capture the correspondences of sequences of frames with spectrograms. Furthermore, an approach to increase the number of samples in the dataset is performed using data augmentation. All these steps are analyzed further in Section 4.1.

Regarding feature extraction from the face modality, for each video, the extracted frames from the video are grouped into consecutive sets of frames (or sequences of frames). Then, several techniques are introduced to extract common representations, in particular:

- Firstly, 3d Convolution Neural Networks (3dCNN [16]) are implemented. The kept set of frames constitute the input to the 3dCNN topology. For each of these sets, the algorithm outputs one common representation denoted henceforth as 3d-embeddings.

- Secondly, a Long Short-Term Memory network (LSTM [17]) is tested. The algorithm combines these sets of frames into a common representation using LSTM architecture (LSTM-embedding).

- Thirdly, a strategy based on a transformer attention mechanism [18] is implemented. The algorithm outputs embeddings that combine all these frames into a common representation in a similar way with LSTM (which henceforth will be denoted as transformer-embeddings). More details will be given in Section 4.1.4.

- Finally, a combination between the transformer and 3dCNN approaches is applied for extracting the temporal information. In this case, firstly, we extract the 3d embeddings from sets of frames. Secondly, since each video consists of several sets of frames, we combine them using the transformer approach.

More details for all the above methods can be found in Section 4.1.4.

Regarding the audio modality, we proceed with the implementation of spectrograms and perform identical pre-processing steps to the ones presented in Chapter 3.

Moreover, it is important to note that we decided to compare the introduced technique which has as input to GANs architecture features with temporal information with the conventional GANs (proposed in Chapter 3) performed under the supervised scenario and by using the moderate training (described in Section 3.1.5 and denoted as dacssGANs). Firstly, we decided to skip the semi-supervised training (explained in detail in Section 3.1.3), since, while it has proven to perform similarly with the supervised case, still we were able to obtain the best results in the latter case. Regarding the progressive training (analyzed in Section 3.1.5), we found that while it is a promising approach and can improve the performance of the network, the training of the whole approach needs considerable time and computational power, and therefore, we decided to skip it and proceed with the moderate scenario (which is the training of the GANs architecture without performing the progressive training).

Finally, the structure of the remainder of this chapter is as follows: Section 4.1 describes the introduced DA method that is based on the proposed temporal GANs architecture and the way that we extract temporal features from the source domain information. In Section 4.2, the implemented metrics, the experimental protocol, the experimental results, and the training process analysis are presented and analyzed. Finally, Section 4.3 contains the conclusions and possible future directions related to this study.

4.1. TEMPORAL GENERATIVE ADVERSARIAL NETWORKS APPROACH

As stated before, the core research conducted in this chapter is directly related to two research questions:

- *Can we improve the performance of the whole system (demonstrated in Chapter 3) further by making use of the Wasserstein loss?*

- *Can the audio-visual knowledge transfer increase if we take into account temporal information extracted from the face modality?*

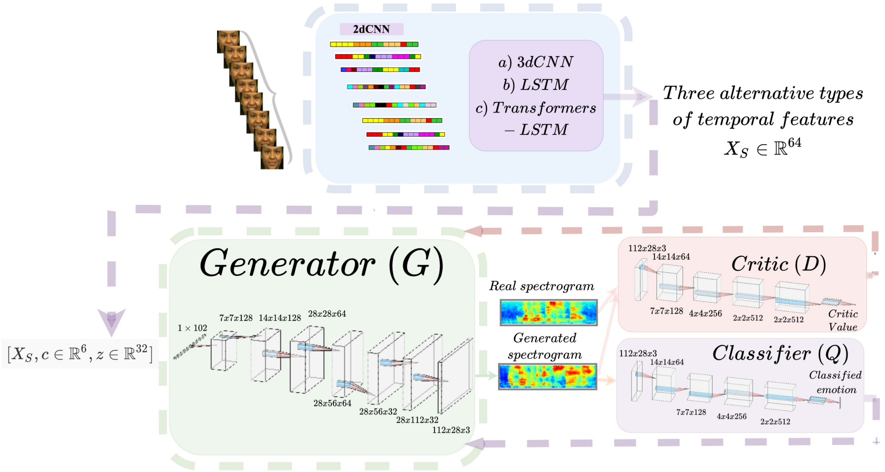


Figure 4.1: Temporal Wasserstein GANs architecture. The input to the generator are temporal features extracted from the upper-block of the figure. The features are extracted either using a) 3dCNN b) LSTM or c) transformer encoder (using as input 2dCNN embeddings).

The following paragraphs contain the analysis for the first research question. Furthermore, a brief summary of the way that GANs architecture is introduced in our work is also placed.

The overall architecture of the approach can be seen in Figure 4.1. Initially, we needed to alter the vanilla GANs architecture to meet the demands of our current research and investigate the two posed questions. Hence, a brief reminder of how GANs are operated will be presented in the following paragraphs. However, this will be analyzed from a slightly different point of view (a more probabilistic aspect). In this way, we will be able to connect the loss function of the GANs approach with the Wasserstein loss. The initial formalization of the GANs min-max game could be framed with the following equation:

$$\min_G \max_D \mathop{\mathbb{E}}_{x \sim \mathbb{P}_r} [log(D(x)] + \mathop{\mathbb{E}}_{\hat{x} \sim \mathbb{P}_g} [1 - log(D(\hat{x})] \tag{4.1}$$

where $x \sim \mathbb{P}_r$ is a real sample coming from the target domain and $\hat{x} \sim \mathbb{P}_g$ is a generated sample (that our approach targets to bring as close as possible to the real distribution). The input to the generator G for the vanilla GANs as mentioned already is the vector $z \sim P(z)$. More compactly, we could frame Equation 4.1 as:

$$\min_G \max_D S_1(D,G) = \mathbb{E}_{x \sim X_T(x)} [\log D(x)] + \mathbb{E}_{z \sim P_z} [\log(1 - D(G(z)))] \tag{4.2}$$

where $x \in X_T$ (is a sample distributed in the target domain D_T) corresponds to the data that we want to approximate. In a more theoretical analysis, our network aims at minimizing the divergence between our unknown real data distribution P_r which is represented from the $X_T \in D_T$ set, which is the distribution of the targeted samples and the distribution of the generated samples denoted as P_g represented by all samples $G(z)$. Using a different notation it can be re-written as:

$$\min_G \max_D S_1(D,G) = \mathbb{E}_{x \sim P_g}[\log D(x)] + \mathbb{E}_{x \sim P_g}[\log(1 - D(x))] \tag{4.3}$$

Now after calculating these expected values by marginalizing over x then also by replacing using some simple auxiliary variables ($\tilde{x} = D(x)$, $\alpha = P_r(x)$ and $\beta = P_g(x)$) it can be easily shown (for this proof you could check [8]) that the optimal discriminator D^* has the shape of:

$$D^* = \frac{P_r(x)}{P_r(x) + P_g(x)} \tag{4.4}$$

and furthermore, by replacing back in Equation 4.3 (can be seen in [8]) it was proven that the loss function (that is denoted as L) from Equation 4.1 could be framed as:

$$L(D^*, G^*) = 2JSP(P_r||P_g) - 2log2 \tag{4.5}$$

where JSP corresponds to Jensen-Shannon divergence [11] which can be formulated as:

$$JSP(P_r||P_g) = \frac{1}{2}KL(P_r||P_A)\frac{1}{2}KL(P_g||P_A) \tag{4.6}$$

where A is the average distribution with density:

$$P_A = \frac{P_r + P_g}{2} \tag{4.7}$$

while the Kullback-Leibler (KL) divergence [12] between two distributions is described by the following equation:

$$KL(P_r||P_g) = \int_{\mathscr{X}} P_r(x) log \frac{P_r(x)}{P_g(x)} dx \tag{4.8}$$

In this way, the work done in [8] shown that optimizing the GANs loss function is equated with optimizing the Jensen-Shannon divergence as described in Equation 4.6.

LOSS THAT INCORPORATES SOURCE DOMAIN, CONDITIONAL INFORMATION AND CLASSIFIER Q

The objective in the current research is to implement a temporal domain shift and calculate a transformation between source ($X_S \subset D_S$ which is represented by temporal-related facial features) and target domain $X_T \subset D_T$ (represented by spectrograms) which is denoted as $X_S \rightarrow X_T$. Thereby, instead of having as input to the generator G merely the noise vector $z \in P_z$, we coupled it together with samples that are distributed from the source domain $X_S = \{x_1, x_2, ..., x_n\} \subseteq D_S$. Additionally, since the goal is to generate data that approximate the target domain $X_T \subseteq D_T$ conditioned to emotional information, the conditional information that is denoted as $\mathbf{c}$ is added to the equation as well. The loss function of the GANs (denoted as $J_1(D, G(\mathbf{x}, \mathbf{c}, \mathbf{z}))$) can be formulated as:

$$\min_{G}\max_{D} J_1(D,G) = E_{y\sim X_T}[\log D(y)] + E_{\mathbf{z}\sim P_z, x\sim X_S}[\log(1 - D(G(\mathbf{x},\mathbf{c},\mathbf{z})))] \tag{4.9}$$

where x are samples derived from the source domain ($x \in X_S$) and y are samples belonging to the target domain ($y \in X_T$). Finally, the next step is (as explained in Section 3.1), to incorporate into the whole architecture a classifier Q and the L1 norm (the loss between real and generated samples during training) denoted as $L_1(G(x,\mathbf{c},\mathbf{z}))$. In this manner, the formulation of the final joint loss (denoted as $J(D,G(x,\mathbf{c},\mathbf{z}))$) could be framed as:

$$\min_{G}\max_{D}(J_1(D,G(x,\mathbf{c},\mathbf{z})) + L_1(G(x,\mathbf{c},\mathbf{z})) + \mathbb{E}[\log(p(c = y^k|x^{'}]))]) \tag{4.10}$$

where the last part of the loss function represents the cross-entropy for the sample $x^{'}$ which is either a real or a generated sample $x^{'} \in G(x,\mathbf{c},\mathbf{z})$ to belong to a specific class $c = y^k$ where y^k the categorical vector represent the class.

4.1.1. ISSUES WITH THE CONVENTIONAL GANS APPROACH

However, even if GANs in this initial version, are powerful generative models, they suffer from training instability. As it has been demonstrated by previous works [7], GANs (in their original form) suffer from the following drawbacks:

- **It is hard to achieve a Nash equilibrium**: The works presented in [7] and [8] established a theoretical analysis regarding the problem with GAN's gradient-descent-based training procedure. Two models are trained at the same time to reach a Nash equilibrium to a two-player non-cooperative min-max game. However, each model updates its cost independently and irrespectively of the other player. Updating the gradient of both models concurrently cannot guarantee a convergence [8].

- **Vanishing gradients**: GANs can optimize network D much easier than G [8]. Minimizing the GANs objective function with an optimal D is equivalent to minimizing the Jensen–Shannon-divergence loss as shown in Equation 4.6. An optimal network D back-propagates the proper gradient for network G to be tuned. However, if G is not properly learning, the gradient for G diminishes and this network is not tuning properly as well. Furthermore, the work conducted in [7] suggests that the divergence function (between real and generated data distributions) that GANs architecture typically aims at minimizing is potentially not continuous with respect to the generator's parameters, leading to training instability and difficulty in training convergence.

- **Mode collapse**: This problem is analyzed in Section 3.1.1. Briefly, the generator may collapse to a setting where it always produces a reduced set of outputs. Even though the generator might be able to trick the corresponding discriminator, it fails to learn a meaningful representation of the complex real-world data distribution and gets stuck in a small space with extremely low variety.

- **Lack of a proper evaluation metric**: GANs, in their initial form, do not have a proper loss function that is informative regarding the training process. While the cross-entropy is an indication for the distance, still is not indicative of the visual quality of the generated results [8]. Without having a good evaluation metric during training, the network does not have any good indication of whether it reached a convergence or not. Furthermore, it is not easy to compare the method with other methodologies.

- **Time complexity**: From some preliminary conducted tests, we figured out that there are different types of loss functions (like Wasserstein distance) which, when incorporated into GANs, managed to accelerate the training process. Through experimentation, we discovered that conventional GANs converge much slower than approaches like Wasserstein GANs.

With this in mind and by taking into account the work done in [1][6], we decided to foster Wasserstein distance and incorporate it in our audio-visual cross-modal GANs architecture. Our hypothesis is that by doing so, the training procedure will stabilize further and the overall quality of generated samples, as well as the efficiency of knowledge transfer will improve.

4.1.2. Wasserstein Generative Adversarial Networks

Wasserstein GANs [1] are introduced in an attempt to mitigate the shortcomings that escort the conventional GANs architecture. This approach is based on the “Earth-Mover’s” distance [9], which can be described as follows: It is the minimum cost of moving and transforming a pile of mass in order to match the shape of one probability distribution with the shape of another one [1]. In order to better grasp this definition, an example is provided. Let us assume that we have the following four samples from two different distributions P_r and P_g (representing real and generated data respectively):

$$\begin{aligned} P_r &= \{P_r^1 = 1, P_r^2 = 5, P_r^3 = 4, P_r^4 = 5\} \\ P_g &= \{P_g^1 = 5, P_g^2 = 3, P_g^3 = 6, P_g^4 = 1\} \end{aligned} \tag{4.11}$$

where in P_i^d, d represents whether the distribution is real or generated and i the index of the value in the distribution. Hence, to calculate the Earth Mover’s distance in our example means that we need to recalculate the sample values of P_g to match the distribution of P_r. On this ground, if we move four piles from ($P_g^1 \rightarrow P_g^2$), then, (P_g^1, P_r^1) have the same value. In the same manner, we move two piles from ($P_g^2 \rightarrow P_g^3$) and four from ($P_g^3 \rightarrow P_r^4$). The cost for each move, could be defined as: $\delta_{i+1} = \delta_i + P_g^i - P_i^r$, while the total cost:

$$c = \sum \delta_i \tag{4.12}$$

This c cost defines the total amount of mass moved in order to perform that transformation. In the case of the continuous probability domain, the distance could be framed as:

$$W(P_r, P_q) = \inf_{\gamma \sim \Pi(P_r, P_g)} \mathbb{E}_{x,y}[||x - y||] \tag{4.13}$$

In this equation, $\Pi(P_r, P_g)$ corresponds to the set of all possible joint probability distributions between the probability distributions P_r and P_g. A single joint probability distribution $\gamma \in \Pi(P_r, P_g)$ describes one possible strategy for mass moving. In particular, $\gamma(x, y)$ describes the percentage of mass that needs to be transported from point x to y so as to make x distribution to approximate the distribution of y. However, it is infeasible to search and track all the possible joint distributions in $\Pi(P_r, P_g)$ to compute $\inf \gamma \sim \Pi(p_r, p_g)$. Hence, a common approach [1] is to alter the formula based on the Kantorovich-Rubinstein duality (and transfer it closer to the initial GANs formula found in Equation 3.1) and constrain Equation 4.13 to:

$$W(P_r, P_q) = \min_{G} \max_{D \in \mathscr{D}} \mathbb{E}_{x \sim P_r}[D(x)] - \mathbb{E}_{y \sim P_g}[D(y)] \tag{4.14}$$

where $\mathscr{D}$ is a set of K-Lipschitz functions. A real-valued function D is called K-Lipschitz continuous if there exists a real constant $K \geq 0$ such that, for all $x_1, x_2 \in R^D$ (with D to be an arbitrary dimension):

$$|D(x_1) - D(x_2)| \leq K|x_1 - x_2| \tag{4.15}$$

It is important to note that Equation 4.14 can be used for the GANs strategy (and it is connoted as wGANs). Samples $x_1, x_2 \in R^D$ could be samples from real and generated distributions, while, function D could be represented by the discriminator of GANs network. Furthermore, the task of the generator is still to generate samples from the D_T in the same way as in Chapter 3. However, the task of the discriminator network D (in the literature it is connoted also as *critic function*), instead of judging whether a sample comes from the real or generated distribution, is to minimize the distance found in Equation 4.14. Hence, its eventual purpose of the wGANs is to minimize the distance between the distributions of real and generated datasets, and in this manner, to learn a proper function $\mathscr{D}$. In other words, the critic is trained to learn a K-Lipschitz continuous function to help compute the Wasserstein distance. As the loss function (Equation 4.14) decreases during training, the Wasserstein distance gets smaller and the generator model's output approximates better the real data distribution. The wGANs loss function results in a critic which outputs a better-behaved gradient with respect to its input than its conventional GANs counterpart, making the optimization of the network G easier. This claim can be intuitively explained by comparing Wasserstein distance with Jensen-Shannon divergence. When comparing two different distributions, even when these are to be found in lower dimensional manifolds without overlaps, Wasserstein distance can still yield a meaningful in-between distance. While with JS we will receive zero difference [1][6].

Empirically, it was also observed that the wGANs loss value allegedly correlates with sample quality of the generated instances, which is not the case for the conventional GANs [1][6].

To enforce the Lipschitz constraint on the critic, authors in [1] proposed to restrict the weights of the critic to lie within a specific range (weight clipping). The set of functions satisfying this constraint is a subset of the K-Lipschitz functions for some K. Yet, this approach still is proved to be problematic (as explained in [6]), regarding finding optimal K-Lipschitz functions. Therefore, to circumvent the tractability issues (for finding

K-Lipschitz functions), an alternative strategy for enforcing the K-Lipschitz needs to be fostered. The shortcomings of enforcing the K-Lipschitz by weight clipping exposed in [8] are:

- The whole wGANs approach is proved to have unstable training while it leads to slow convergence of the weights of network G and D as well.
- The wGANs optimization process is difficult because of interactions between the weights' constraint and the cost function, which result in either vanishing or exploding gradients without carefully tuning and choosing the clipping threshold c.
- Implementing a K-Lipschitz constraint via weight clipping biases the critic by calculating much simpler Lipschitz functions.

Algorithm 2 Wasserstein GANs implementing the gradient loss. Initially, the following parameters are initialized randomly: λ, n_c, λ_r, β_1 and β_2. The variables α and β are variables related to AdaGrad algorithm [10]. Moreover, G_θ represents the weights of generator G while D_w represents the weights of critic D.

```
Require: The gradient penalty coefficient $\lambda$, the number of critic iterations per generator iteration $n_c$, the batch size M, AdaGrad hyper-parameters learning rate $\lambda_r$, $\beta_1$ and $\beta_2$.
Require: Initialize generator $G_\theta$ and critic $D_w$ weights randomly.
  while The weights of the generator $G_\theta$ did not converge yet do
    for i = 1, 2, 3, ..., $n_c$ do
      for j = 1, 2, 3, ..., M do
        Sample real data $x \sim P_r$, a noise vector $z \sim P_z$ and some uniformly distributed noise: $\epsilon \sim \mathscr{U}[0,1]$
        $\tilde{x} \leftarrow G_\theta(z)$ {$\hat{x}$ represents the generated samples}
        $\hat{x} \leftarrow \epsilon x + (1-\epsilon)\tilde{x}$ {adding some noise $\epsilon \sim U(1,2)$}
        $L^j \leftarrow D_w(\hat{x} - D_w(x) + \lambda(||\nabla_{\hat{x}} D_w(\hat{x})||_2 - 1)^2$
      end for
      $w \leftarrow AdaGrad(\nabla_w \frac{1}{M}\sum_{j=1}^{M} L^j, w, \alpha, \beta_1, \beta_2)$
    end for
    Sample a batch of latent variables $\{z^j\}_{j=1}^{M} \sim P_z$
    $\theta \leftarrow AdaGrad(\nabla_\theta \frac{1}{M}\sum_{j=1}^{M} -D_w(G_\theta(z)), \lambda_r, \beta_1, \beta_2)$
  end while
```

As mentioned before, a differentiable function is K-Lipschitz if and only if it has gradients with norm at most K everywhere (shown in equation 4.15). Hence, another strategy can be implemented to directly constrain the gradient norm of the critic's output with respect to its input. On this ground, as proposed in [6], we enforce a K-Lipschitz constraint with a penalty on the gradient norm when using random samples $\hat{x} \in P_z$. Therefore, Inequation 4.15 can be framed as:

$$\mathscr{L} = \mathbb{E}_{\hat{x}\sim P_g}[D(\hat{x})] - \mathbb{E}_{x\sim P_r}[D(x)] + \lambda\, \mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2] \tag{4.16}$$

Table 4.1: Details about the employed networks. The variable θ stands for the dropout parameter (see Appendix 3.1).

Network	learning rate	Optimizer	θ
G	0.01	RMSProp	0.5
D	0.01	RMSProp	0.5
Q	0.01	RMSProp	0.5

where the variable λ corresponds to the regularization parameter regarding the enforced gradient penalty. Again, P_r represents the distribution of real samples, P_g the distribution of generated samples and P_x a random combination of real and generated samples. The whole strategy for this version of Wasserstein GANs is summarized in Algorithm 2. As we can observe, firstly, the parameters regarding the regularizer of the gradient and the optimizer (AdaGrad) of the whole process are initialized. Then, the weights of the critic are tuned (multiple times based on a parameter denoted as n_c). Finally, the weights of the network G are trained. The whole training process is looped until the weights of G converge.

INCORPORATING CLASSIFIER Q AND CONDITIONAL INFORMATION IN WASSERSTEIN LOSS
Finally, Equation 4.16 is modified in order to incorporate (as in the case of the conventional GANs approach analyzed Chapter 3) the classifier Q and the conditional information $\mathbf{c}$. Therefore, the equation can be framed as:

$$\mathscr{L} = \mathbb{E}_{\hat{\mathbf{x}}\sim P_g}[D(\hat{x})] - \mathbb{E}_{x\sim p_r}[D(x)] + \lambda\, \mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2] + \mathbb{E}[\log(p(c = y^k|x^{'}]))]) \quad (4.17)$$

where $p(c = y^k|x^{'}])$ denotes the probability of a sample $x^{'}$ to belong to the specific class $c = y^k$.

4.1.3. NETWORK TOPOLOGY

In this paragraph, we analyze the topology for all the involved networks in our Wasserstein GANs scheme. Firstly, the tensor dimensionalities (that are mainly referring to convolution and max pooling operations) are displayed in Figure 4.1. More details can be found for generator and discriminator in Tables 4.2 and 4.3 where all the input outputs are depicted. However, besides the tensor dimensionalities, in the same way as in Chapter 3, we make use of the same operations (namely, “dropout”, “batch normalization”). The activation function for each layer for all networks is “LeakyRELU”, while the activation function in the output layer is “hyperbolic tangent” function.

Finally, Table 4.1 contains details related to all networks hyper-parameters such as learning, optimizer and dropout parameter θ.

4.1.4. TEMPORAL FEATURE EXTRACTION

In this subsection, the focus lies on the second posed question of this chapter and in particular: *Can the audio-visual knowledge transfer be increased if we take into ac-*

Table 4.2: The model employed for the generator network *G*.

Layer type	tensor input	tensor output
Dense	1 × 102	1 × 6272
Reshape	1 × 6272	7 × 7 × 128
Cov2d	7 × 7 × 128	7 × 7 × 128
Upsampling2d	7 × 7 × 128	14 × 14 × 128
Batch Normalization	14 × 14 × 128	14 × 14 × 128
Dropout	14 × 14 × 128	14 × 14 × 128
Cov2d	14 × 14 × 128	14 × 14 × 128
Upsampling2d	14 × 14 × 128	28 × 28 × 128
Batch Normalization	28 × 28 × 128	28 × 28 × 128
Dropout	28 × 28 × 128	28 × 28 × 128
Cov2d	28 × 28 × 128	28 × 28 × 64
Upsampling2d	28 × 28 × 64	28 × 56 × 64
Batch Normalization	28 × 56 × 64	28 × 56 × 64
Dropout	28 × 56 × 64	28 × 56 × 64
Cov2d	28 × 56 × 64	28 × 56 × 32
Upsampling2d	28 × 56 × 32	28 × 128 × 32
Batch Normalization	28 × 128 × 32	28 × 128 × 32
Dropout	28 × 128 × 32	28 × 128 × 32
Cov2d	28 × 128 × 32	28 × 128 × 3
Tanh	28 × 128 × 3	28 × 128 × 3

Table 4.3: The model employed for the discriminator network *D*.

Layer type	tensor input	tensor output
Conv2D	3 × 112 × 28	64 × 112 × 28
Batch Normalization	64 × 112 × 28	64 × 112 × 28
Dropout	64 × 112 × 28	64 × 112 × 28
MaxPooling	64 × 112 × 28	64 × 56 × 28
Conv2D	64 × 56 × 28	32 × 56 × 28
Batch Normalization	32 × 56 × 28	32 × 56 × 28
Dropout	32 × 56 × 28	32 × 56 × 28
MaxPooling	32 × 56 × 28	32 × 28 × 28
Conv2D	32 × 56 × 28	16 × 56 × 28
Batch Normalization	16 × 56 × 28	16 × 56 × 28
Dropout	16 × 56 × 28	16 × 56 × 28
MaxPooling	16 × 56 × 28	16 × 28 × 28
Conv2D	8 × 28 × 28	8 × 28 × 28
Batch Normalization	8 × 28 × 28	8 × 28 × 28
Dropout	8 × 28 × 28	8 × 28 × 28
MaxPooling	8 × 28 × 28	8 × 14 × 14
Conv2D	8 × 14 × 14	4 × 14 × 14
Batch Normalization	4 × 14 × 14	4 × 14 × 14
Dropout	4 × 14 × 14	4 × 14 × 14
MaxPooling	4 × 14 × 14	4 × 7 × 7
Conv2D	4 × 7 × 7	4 × 4 × 4
Batch Normalization	4 × 4 × 4	4 × 4 × 4
Dropout	4 × 4 × 4	4 × 4 × 4
MaxPooling	4 × 4 × 4	4 × 2 × 2
Dense layer	4 × 2 × 2	1 × 16]
Dense layer	1 × 16	1 × 512
Dense layer	1 × 512	1 × 6

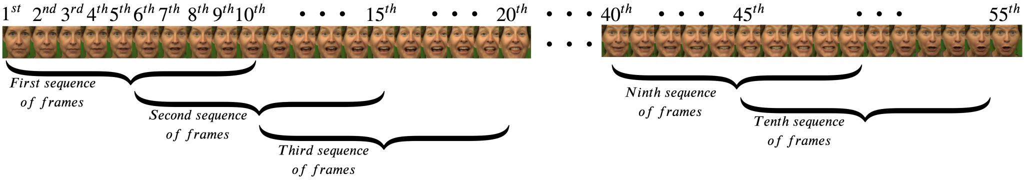

1st 2nd 3rd 4th 5th 6th 7th 8th 9th 10th • • • 15th • • • 20th • • • 40th • • • 45th • • • 55th
First sequence of frames
Second sequence of frames
Third sequence of frames
Ninth sequence of frames
Tenth sequence of frames

Figure 4.2: The way that the video frames are grouped into a sequence of frames.

count the temporal information extracted from the face modality? Therefore, in this new approach, we analyze the extracted temporal information regarding the face. On this ground, before studying the cross-modal relationships with the proposed architecture, a pre-processing step is performed to extract the targeted temporal information both from face and audio.

Briefly, regarding the audio modality, as described in Section 3.1.6, spectrograms are extracted for the target domain X_T as its temporal information representation. The rationale of choosing spectrograms are based firstly on previous findings in the literature [13] [14] and, secondly, on visualization purposes. This is a key point since we can easily infer the quality of the generated images from GANs architecture using visual inspection. As a final step, an augmentation strategy is utilized in the same spirit as in [15] to increase the number of samples in the dataset. This strategy is analyzed in more detail in Section 3.1.6. This approach modifies the spectrograms by warping them in the time direction, masking blocks of consecutive frequency channels, and masking blocks of utterances in time. In this way, for each input spectrogram, we are able to construct 9 different new ones.

With respect to pre-processing steps for face modality: firstly it is decided to keep the middle 55 frames of each video. In this way, we wanted to exclude frames that do not contain important information from the beginning and the end of the video clips. Then, face detection and alignment are performed (following the strategy previously described in Section 3.1.6). Subsequently, for each video, overlapping time-windows sequences of 10 frames are created. In this manner, the windows are slid by five frames each time, which results in an overlap of five frames between the time-windows. This process is portrayed in Figure 4.2 where all the extracted sequences for one specific video clip are shown. Since, for each video, there are 10 sequences (consisting of 10 frames) and 10 different spectrograms (generated after the data augmentation approach described in Section 3.1.6), by combining them, 100 different new pairs are obtained.

Subsequently, the temporal feature extraction process for the face is performed. This is displayed in Figure 4.1 and more particularly in the upper block. Four different approaches are investigated for extracting temporal features. In particular, 3dCNN, LSTM, attention mechanisms and a combination between 3dCNN and attention mechanisms.

3D Convolutional Neural Networks

3dCNNs constitute an extension of conventional CNNs (see Appendix 3.1), expanded in three dimensions, mainly to include the notion of time. The main applications of this

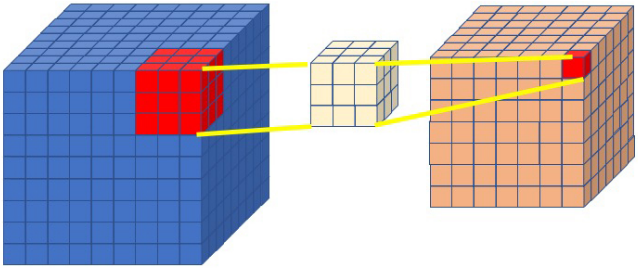


Figure 4.3: An example of how the 3d convolution is performed in the 3dCNN topology. The input to the operation is a sequence of frames while the kernel applied is 3d.

approach can be found in the domain of computer vision or graphics and especially for volumetric video analysis [22][23] or for 3d imaging.

The input to a 3dCNN is 3d data and more specifically, a 3d tensor (a set of frames derived from a video in our case), while the output is a common representation (the extracted temporal features). This approach extracts spatio-temporal features by performing 3d convolutions, and, in particular, by capturing the motion information encoded in multiple adjacent frames. Furthermore, in the same spirit as with the input, the learned convolution kernels that are applied and the max pooling operation are performed in a 3d manner as well. This operation is visualised in Figure 4.3. Having as input a 3 × 3 region, then, we apply the feature kernel which has the same dimension. By sliding the filter all over the input tensor, we can calculate the extracted feature map.

Long short-term memory

Long short-term memory (LSTM) networks [17] algorithm was employed as follows: Given as input ten frames (their 2dCNN features) to the algorithm, it process them in different output timestamps. Then, it outputs one common representation which is tuned using a classification schema Facial Emotion Recognition (FER). That output comes from the final cell of the LSTM. More details about the LSTMs topology and information about the algorithm (cell input or output) are analyzed further in Appendix 3.2. We apply LSTMs in our case, since it is a very popular technique for extracting temporal information in general in time signals but also when handling videos. For the training process as well as for the test process the same exact topology is applied.

Attention mechanisms and the transformer attention case

The transformer attention mechanism [18] is a type of neural network architecture introduced mainly to tackle machine translation and sequence to sequence tasks by placing attention on related and semantically involved parts of a sentence. In our work, it is modified and implemented for video analysis and more specifically for extracting temporal embeddings that are placing attention in different parts of the input video signal (which consist of facial expressions). Before explaining how this framework is incorporated in our work, a description of the overall technique is presented.

In its core essence, the framework operates as shown in Figure 4.4. This is a simplification of how the initial framework functions for the machine translation problem. This

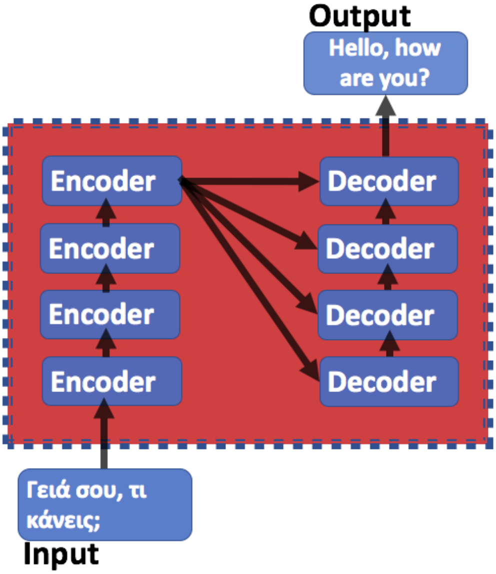


Figure 4.4: Basic example of the transformer architecture. It consists of stacked encoders networks and stacked decoders networks. In this example, the transformer framework accommodates a machine translation problem. The input sequence corresponds to a Greek phrase while the output sentence corresponds to its translation in English.

task concerns the linguistic translation of an input sentence from the input language (for example, Greek source domain D_S) into the targeted language (for example English, target domain D_T). Normally, the input sentences, instead of string representations (using actual words) are portrayed by numerical representations of the initial words, which alternatively are called embeddings. This approach is called word vectorization, which is a methodology in Natural Language Processing (NLP) to map words or phrases from vocabulary to a corresponding vector of real numbers by using techniques such as bag of words, term frequency-inverse document frequency (tf-idf), Neural models and so forth [25].

However, in our approach, our task is related to video, and in particular, in extracting features from sequences of frames. Hence, instead of having words and sentences (or textual embeddings) as input to the transformer framework, we have sequences of frames from a video. The input is visualised in Figure 4.2. Each time, the input to the framework is one sequence (consisting of 10 frames). Furthermore, each frame is represented by its extracted features (by a 2dCNN network using $\mathbf{x}_i \in \mathbb{R}^{128}$). The implemented network used to extract these features is using the same topology implemented as face classifier and rendered in Figure 3.4.

In further detail, the initial transformer network consists of a set of stacked encoder networks and a set of stacked decoder networks. This is illustrated in Figure 4.4. Firstly, we should explain the term “stacked”, a ubiquitous term in the deep learning domain which is associated with putting deep learning networks in a sequence (as depicted in

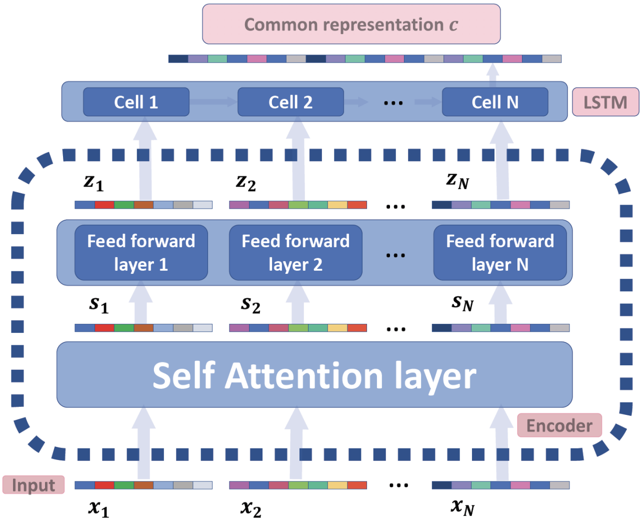


Figure 4.5: The three assembled modules that the feature extraction procedure based on transformer architecture (self-attention layer, feed-forward layer and LSTM). The positional encoding is performed to the input. In the presented approach, the dimensionality for all the embeddings is: $\mathbf{x}_i \in \mathbb{R}^{128}$, $\mathbf{s}_i \in \mathbb{R}^{256}$, $\mathbf{z}_i \in \mathbb{R}^{256}$ and $\mathbf{c} \in \mathbb{R}^{64}$. The total number of frames applied as input to the network in our approach was $N = 10$.

Figure 4.4). Furthermore, the notion of "encoder" in essence is a network that, given as input information from a specific domain, learns as an output a feature representation for this domain. While the "decoder" network has as input the feature representation (derived from the encoder) and its task is to transform this input to a specific output domain.

All these encoder and decoder networks are identical to each other (they have the same architecture) with an exception of the weights sharing. Essentially, the encoder's task is to extract features from the input data, while the decoder's task is to translate these features to the targeted output. With respect to the encoder network, it is visualized in Figure 4.5, while it can be decomposed into three modules:

- Firstly, the so-called **self-attention** module, which allows the input frames (or words in case of machine translation) of the transformers framework to interact with each other, and discover to which other frames of input they should pay more attention to. The outputs are aggregates of these interactions and attention scores. In this way, multiple representations for each input are constructed. Part of the self-attention, is also the multi-head mechanism which is introduced for extracting better representations and is analyzed further later in this section.

- Secondly, the **feed-forward layer** module, which combines multiple representations for each input extracted from the self-attention module in a single representation.

- Finally, the **positional encoding layer** module, which is a method to memorize the

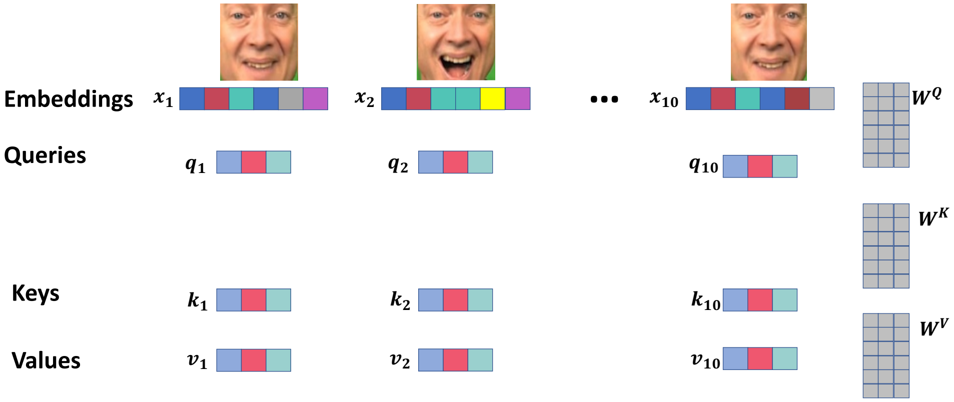


Figure 4.6: Illustration of some details regarding the self-attention mechanisms related to query Q key K and value V weights. These matrices are learned during the training of the whole scheme. Then, each input $\mathbf{x}_i \in \mathbb{R}^{128}$ is transformed by applying these weights to vectors: $\mathbf{q}_i \in \mathbb{R}^{64}$,$\mathbf{k}_i \in \mathbb{R}^{64}$and $\mathbf{v}_i \in \mathbb{R}^{64}$. This example is again for the case that we have as input $N = 10$ frames and represents one head.

order of the sequence frames.

Before analyzing these modules in details we should give some more details about the input we applied to them. Since the task of the current work is to combine several input frames ($\mathbf{x}_i$, with $i \in \{1,...,10\}$) into one common representation, the decoder part mentioned before is neglected. The reason behind this decision is that our task extracts features and does not perform domain translation directly. Therefore, we need to use only the outputs of the stacked encoders (denoted as $\mathbf{z}_i$), which will be combined together using an LSTM scheme in order to produce one common representation $\mathbf{c}$.

In the following paragraphs, the basic blocks of the encoder networks are analyzed in further detail.

Self-Attention module: As it was mentioned before, the self-attention layer helps the encoder look at other frames in the input sequence as it encodes a specific frame. Therefore, for each input frame, it performs some specific operations in an attempt to learn a distributed vector of weights that associates each frame with the rest of the input frames.

The first operation of the self-attention module is tasked to calculate three vectors, namely key $\mathbf{k}$, query $\mathbf{q}$ and value $\mathbf{v}$, for each of the different encoder input frames $\mathbf{x}_i$. This can be seen in both Figures 4.6 and 4.7. These vectors, while they do not have any physical meaning, are helpful abstractions for calculating the attention weights for each frame. These vectors are created by multiplying the input embeddings $X = \{\mathbf{x}_1,...,\mathbf{x}_i,...,\mathbf{x}_{10}\}$ by three matrices (feature maps) W^K, W^Q and W^V that are network weights tuned during the training process. The dimensionality of these weights is the same with $W^K \in \mathbb{R}^{d\times l}$ where d is the new dimensionality that we want to transform the input while l is related to the number of input frames (in this case, $l = 10$). In the following paragraphs, it will be shown that in practice, multiple weight matrices need to be calculated for the key, query and value.

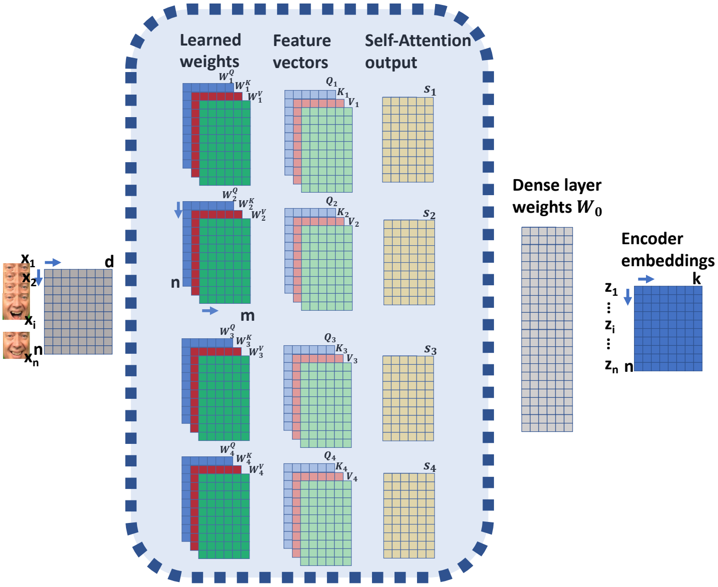


Figure 4.7: This figure visualizes the whole encoder process. The multi-head mechanisms are composed of four heads as depicted in the figure (each row in the blue box). Each input $\mathbf{x}_i \in \mathbb{R}^{128}$ is transformed by applying all this process to the encoder embeddings, where output $\mathbf{z}_i \in \mathbb{R}^{64}$ corresponds to each frame. The dimensionality of the output of the self-attention mechanism is $\mathbf{s}_i \in \mathbb{R}^{256}$

It is important to note that the dimensionality d of these vectors could be either smaller or bigger than the size of the embeddings and it is the architecture choice that should be tuned during the training process.

Having trained the aforesaid weights, the three vectors $\mathbf{k}$, $\mathbf{q}$ and $\mathbf{v}$ can be calculated for each input frame. Subsequently, the second operation of the self-attention layer is to calculate the score $\tilde{s} \in \mathbb{R}$. This value reveals how much focus needs to be placed on different parts of the input sequence of frames as we encode a frame at a certain position. Finally, the score $\tilde{s} = \mathbf{q}^T\mathbf{k}$ can be found by calculating the dot product of the query vector with the key vector of the respective frame which is under examination.

The next calculation is the division of the calculated score $\tilde{s}$ with the square root of the dimensionality of the key vector size d_k, $\hat{s} = \tilde{s}/d_k$. This is done in an effort to normalize the output and stabilize the gradients of the network. Then, a softmax process normalizes these scores $\sigma(\hat{s})$ values within 0-1 range. This softmax score $\sigma(\hat{s})$ determines how much each frame will be expressed at this position. It is obvious that each time the current frame will have the highest softmax score, but the interesting part is the softmax score regarding the rest of the input frames.

The next operation, for a specific input frame $\mathbf{x}_i$, is to multiply each value vector $\sigma(\hat{s}) * \mathbf{v}_i$ for each frame by the softmax score of the input frame. The final operation of the self-attention is to sum up all these weighted value vectors. This produces the output of the self-attention layer at this position. All these aforesaid steps are performed for each frame. Therefore, for $\mathbf{x}_k$ frame the sum-up of these weighted value vectors is framed as $\mathbf{z}_k = \sum_{i=1}^{10} \sigma(\hat{s}_k) * \mathbf{v}_i$, where, we sum-up the values vectors for all the frames. Moreover, it is important to note that the score of $\mathbf{x}_k$ for the rest of frames are calculated as: $\tilde{s} = \mathbf{q}_k^T \mathbf{k}_i$

That concludes the self-attention calculation. The resulting vector is the input to the feed-forward layer. In the actual implementation, however, this calculation is done in a matrix perspective for faster processing. It is worth noting that Q **query**, K **key** and V **value** matrices must be calculated after applying some weight matrices (feature map weight matrices) that are learned during the learning procedure. We can denote as $Q = X \times W^Q$, $K = X \times W^K$ and $V = X \times W^V$ all these above-mentioned operations, where W^Q, W^K and W^V are the matrices to be learned and $X = \{\mathbf{x}_1, ..., \mathbf{x}_{10}\} \in \mathbb{R}^{128\times 10}$ is a matrix which combines all input embeddings vectors $\mathbf{x}_i$. On the whole, we can frame the whole approach using the following equation:

$$z = \sigma(\frac{Q \times K^T}{\sqrt{K_d}}) \times V \tag{4.18}$$

where K_d is the dimensionality of the key matrix.

Multi-headed Attention: Additionally, authors in [18] introduced a mechanism called multi-headed attention. This methodology is part of the self-attention mechanism. In its essence, this mechanism produces several attention outputs z_i (see Equation 4.18) called heads for a single frame input. Instead of having just one set of W^Q, W^K and W^V, as it was described before, we can perform this process several times (each being a different head). Hence, if we decide to have four different heads in our approach, four different z matrices for each input frame $\mathbf{x}_i$ will be calculated. This is visualized in Figure 4.7, where we have as input N frames (each frame is represented by an embedding with d feature dimensionality). Furthermore, we can see the different heads of the encoder. The same is depicted in Figure 4.7.

This improves the performance of the attention layer in two ways:

- It expands the model's ability to focus on different positions.
- It gives the attention layer multiple representation sub-spaces.

Feed-forward layer: The matrices Q, V and K, are calculated after the **Multi-headed Attention** step for all the heads and frames, are concatenated into one matrix. Eventually, the weights of a feed-forward layer are applied to this matrix and the final output is another matrix that contains one embedding $\mathbf{z}_i$ for each input frame $\mathbf{x}_i$. This process is represented by Figure 4.7 from the dense layer weights W_0.

Positional Encoding: One fundamental operation involved in the transformer framework is the so-called positional encoding mechanism which is a way to account for the order of the input sequences. The positional encoding step allows the model to recognize which part of the sequence an input belongs to. That is a pre-processing step that is applied to the input embeddings of the transformer's network. It is necessary to note here that this operation is not part of the learning process meaning that it is just an operation and not a trainable procedure.

At a higher level, the positional embedding is a set of vectors, where each of the vectors represents the position of a specific input embedding in the sequence. These vectors are added to the input embeddings to produce a final embedding vector with order information. This operation in our proposed transformer network is added before the transformer encoding part.

As we mentioned before, our sequence of frames are represented by feature embeddings (extracted using 2dCNN) $\mathbf{x}_i \in \mathbb{R}^{N \times h}$, where N represents the number of frames in a sequence, while the h represents the dimensionality of the input embeddings. Then, we define i to be the position of the frame in the sequence, and $j \in [0, h-1]$ the index that spans the dimensions of the input embeddings. Moreover, the positional encoding procedure can be framed from the following equations:

$$P_{i,2j} = sin\left(\frac{i}{10000^{\frac{2j}{h}}}\right) \tag{4.19}$$

$$P_{i,2j+1} = cos\left(\frac{i}{10000^{\frac{2j}{h}}}\right) \tag{4.20}$$

It is easy to see that the frequency of the sine and cosine functions is determined by the dimensional index i. Once the positional encoding vectors are calculated, the last procedure is to add these vectors to the initial embeddings $\mathbf{x}_i$ (via standard element-wise addition) $\mathbf{x}_i^{'} = \mathbf{x}_i + \mathbf{p}_i$ (where $\mathbf{p}_i$ is a specific vector from matrix P).

To capture positional information, each element of the positional embedding varies according to a frame's position and the index of the element within the dimension of the frame embedding h. This is achieved by varying frequencies, as mentioned above.

Embedding combination and training: Finally, in order to combine all the extracted embeddings outputs $\mathbf{z}_i$ of the encoder network (visualized in Figure 4.5), an **LSTM layer** is employed which outputs a common dense layer ($\mathbf{x} \in \mathbb{R}^{256}$), when given as an input representation for each frame.

As mentioned before, our purpose is to tune the weights of the whole transformer network in such a way that we will be able to extract a good temporal representation of facial frames. With the purpose of tuning the whole transformer encoder, the emotion recognition task is utilized. To this end, for each input video, the common dense layer can be returned as the temporal embeddings. In Figure 4.5, the input $\mathbf{x}_i$ represents one of the frames from a set of 10 frames. Instead of giving the raw frames as input, we transformed the set of 10 frames into 2dCNN features. As a corollary, the output of the whole approach is one common representation for all 10 frames in the set.

Therefore, by having as input 10 frames, the encoder network returns one common representation of these 10 frames as seen in Figure 4.7. Finally, for each video, from the 100 possible sets of 10 frames (that described at the beginning of the current sub-section) we can extract 100 different embeddings based on the transformer encoder. Figure 4.5 displays how the whole architecture looks like in the case of one stacked encoder. In the case of multiple stacked encoders, the output of the first one is processed as input to the second one and so forth. The final output of the whole network is the embedding that we keep for using it in the GANs network. Finally, it is important to note that in total, for our network, we applied a topology with 8 stacked encoders.

Combination of transformer and 3dCNN

Finally, a modified version of the transformer framework is also performed. This time, instead of extracting 2dCNN embeddings, we use the 3dCNN network to extract feature embeddings.

This combination could be better understood by making reference to Figure 4.2. For each of the 10 sequences of frames represented in this figure, we extract their 3dCNN embeddings. Finally, for each video, we have 10 temporal 3dCNN extracted embeddings. These embeddings can be used subsequently, as our input to the transformer framework. The transformer encoder uses these embeddings and outputs one common representation (following the same procedure applied in the previously described case of the transformer framework).

For each video, in the end, we can extract one embedding from the faces that corresponds to one spectrogram.

Simple data augmentation approach for spectrograms

It is important to mention here that we followed the same strategy with Chapter 3.1.6, for data augmentation in order to increase the number of audio samples and thus, increase the size of our datasets. Briefly, as it was explained before, we made use of SpecAugment [19] approach which, given as input a spectrogram, can generate new ones by performing some time wrapping and by applying some block masking. This approach is denoted as **simple data augmentation**.

4.2. Experimental phase for using temporal Generative adversarial networks for emotion-aware domain adaptation

This chapter's main target is to study two core research questions:

- The first posed question concerns the efficiency of the Wasserstein loss in comparison with the loss implemented in the framework (consisting of minimax GAN Loss and the L1 loss) described in Section 3.1.

- In the second posed question, we want to scrutinize whether modelling the temporal dynamics that govern the audio-visual relationship could help to improve the performance of knowledge transfer.

In the rest of the current section, the conducted experiments that were performed to study these two research topics are presented. To better display the experimental phase and clearly illustrate the conducted experiments, we segregate this section into the following subsections: Firstly, we discuss the evaluation metrics which were applied to calculate the performance of the tested approaches (Section 4.2.1). Secondly, we analyze the experimental protocol (for all the performed cases) which is applied for answering the two posed questions (Section 4.2.2). In the experimental results (Section 4.2.3) section, we discuss about the observed results for the whole experimental protocol. Finally, in the training process subsection, some results that are related to the training process and concern the time and stability of the process are analyzed (Section 4.2.4).

4.2.1. Evaluation metrics

To be able to compare the conducted research of this chapter with the work of the previous ones (especially for Chapter 3), we employ the metrics introduced in Section 3.2.1. In particular, we implement the same four evaluation metrics:

- **GANs augmentation scheme using real and generated data**: A **GANs data augmentation** scheme where real and generated samples are fused. The scope is to evaluate whether, by fusing the generated samples to the initial dataset, the conducted Audio Emotion Recognition (AER) could achieve greater performance and improve the classification rate. This approach is the same as implemented in Section 3.2.1.

- **Inception Score (IS)**: This score is measured based on two criteria: firstly, whether the generated spectrograms have diversity and, secondly, whether they have good quality. The higher the IS score, the better the quality of the generated samples.

- **Fréchet Inception Distance (FID)**: the FID score compares the statistics of generated samples to real ones. Lower FID values mean better image quality and diversity in the generated samples.

- **Structural Similarity Index (SSIM)**: the SSIM score allows comparing two images based on perceptual differences. It is expressed as a floating-point number ranging from -1 to 1, where 1 indicates exact similarity and -1 means a complete dissimilarity.

In addition to these metrics, since we wanted to evaluate the efficiency of Wasserstein GANs and the conventional approach time-wise, we decided to add, as an extra measurement tool, the total amount of time needed for training each of the conducted approaches (conventional GANs or Wasserstein GANs). The training was deployed on a Titan XP GPU and the total amount of time corresponds to 100 epochs. From the experimental phase (for the wGANs using features extracted using 3dCNN as it is analyzed in Section 4.2.4), while we run our experiments in total for 100 epochs, we deduced that we could apply an early stopping operation and quit the training process at 55 epochs. In this way, we could reduce even more the training complexity and computational time. More details regarding how this metric is deployed can be found in Section 4.2.3.

4.2.2. Experimental protocol

In this subsection, the structure of all conducted experiments conducted to study the two aforesaid research questions is analyzed. The core three parts of the experimental protocol are the baseline evaluation, the evaluation of the Wasserstein loss, and the evaluation of the temporal information.

It is important to state that in the same way as in Chapter 3, CREMA-D and RAVDESS datasets were applied. Furthermore, we are using the same scheme to split both datasets (to $S = \{S_1, S_2, S_3, S_4\}$) where S is the complete dataset. However, sets S_2 and S_3 are merged and form the validation sets. Set S_1 is as in Chapter 3 applied for training our GANs scheme. S_4 is utilized for performing the test. An extra definition which is useful for further reading is the size of each subset which can be framed as $N = \{N_1, N_2, N_3, N_4\}$.

Baseline method

Before analyzing the experimental protocol for the two core research questions, for comparison purposes, we developed a CNN network for performing audio emotion recognition (AER) using only the real spectrograms from both datasets (CREMA-D and RAVDESS) respectively. The architecture that is implemented is the one displayed in Figure 3.7 (classifier Q). More details about the topology can be found in Section 4.1.3. This evaluation will be denoted henceforth as “baseline evaluation” (in the same way as in Chapter 3), since our primary focus is to investigate whether, by fusing the generated samples with the real ones, we can outbid the performance of this baseline. This approach is explained in Section 3.2.2. Before performing the training of this approach, we make use of the simple augmentation scheme (Section 3.2.1) to increase twice the size of the training subset ($2 \cdot N_1$). This step is implemented in order to be able to fairly compare the baseline with the GANs augmentation scheme (analyzed in Section 3.2.1). The training of this method and the methods that we are comparing with have the same training size.

Evaluation of Wasserstein loss

To answer the first posed research question of this chapter, we perform multiple tests. All these tests are compared against the performance of the baseline evaluation. The rationale behind these tests is to evaluate the efficiency of the Wasserstein GANs and investigate whether we can increase the cross-modal “knowledge transfer” in comparison with conventional GANs. In particular, this evaluation consists of the following steps:

- Firstly, the best-performed approach derived from Chapter 3 is utilized. This approach implements the conventional GANs version using an architecture based solely on dense layers (called EncDec and explained in Section 3.1), an extra L1-norm for the loss function, and the classifier Q for enforcing affect information in the generated spectrograms. All these details are described in Section 3.2.3. In this approach (denoted as **dacssGANs**) the input to the generator G is a noise vector bind together with conditional information and one image from the source domain. The whole architecture of this approach is rendered in Figure 3.2.

- Secondly, we incorporate into the architecture of the conventional GANs the Wasserstein loss (by replacing the loss applied in Section 3) as can be seen in Equation 4.17. This approach, denoted as **wGANs-im2im**, is based on the EncDec architecture. As explained in Section 4.1.1, with Wasserstein loss, the training process has proven to be more stable (regarding the convergence) and reaches an optimal solution much more often. Therefore, we are expecting, by using this loss, to increase the performance of the knowledge transfer as well.

- As an extra evaluation, the performance of the Wasserstein GANs, but this time without taking into account information from the source domain D_S (and particular extracted facial features), is evaluated. This approach is denoted as **wGANs-decoder**. In this version, the only information we add in the generator G is noise and label information. In this case, we needed to modify the architecture that is shown in Figure 3.2. We remove completely the encoder part of the generator network G and we kept only the decoder part of G. In such a manner, we could add directly the input (the noise vector and the conditional information). The role of the generator encoder part in the network G is mainly to analyze the input images into latent space, while the role of the decoder is to transform the latent space to target size images. The input of the network is a vector of noise and some conditional information related to emotion labels. It is important to note, that the terms encoder and decoder are referring explicitly to the network G and we should not confuse them with the ones analyzed in the transformer networks in Section 4.1.4. The rest of the architecture is similar to the one displayed in Figure 3.2. This evaluation is performed to ensure a smooth transition and validate whether an architecture with just the decoding part (of the network G) performs as good as the previous evaluation.

In all these evaluations, our goal is to investigate whether the transferred knowledge between the two domains could be further increased when the Wasserstein loss is incorporated in the GANs approach. On this ground, we employ all the metrics discussed in Section 4.2.1.

Evaluation of temporal information

Furthermore, the second core pillar of the conducted research is associated with the study of temporal information as input to the whole network. To test the efficiency of temporal information, several approaches for extracting related features are applied. More specifically, as already mentioned: 3dCNN, LSTM, attention transformers, and a combination between 3dCNN and transformers.

More details can be found for all these approaches in Section 4.1.4.

Moreover, in an attempt to compare the performance when using Wasserstein loss with the dacssGANs method (the employed method of Chapter 3) an extra test is applied. In this case, we modified the architecture of the dacssGANs to be the same with all the topologies associated with the Wasserstein GANs. Therefore, by taking as a reference point the architecture found in Figure 3.3(a) we kept only the decoder part. The input to this generator G is again a noise vector $z \sim P(z)$ coupled with a conditional vector c and a vector representing the temporal information $\mathbf{c}$. This is represented by 3dCNN features from the face modality. The applied loss function can be seen in Equation 3.7.

By performing the above-mentioned test, we want to compare whether Wasserstein loss (when having as input the temporal information) performed better than the conventional GANs when also leveraging the temporal information. This approach is denoted as **dacssGANs-tmp**.

Before analyzing the experimental results, it is essential for clarity purposes, to organize all the aforesaid topologies. In this way, straightforward navigation through the experiments is guaranteed. All these topologies can be found in Table 4.4.

Table 4.4: Information about the features embodied in each of the approaches evaluated in this chapter. For each approach, we can see some of the inputs, features, and types of loss functions that are embodied in these approaches.

Case/Feature	Raw input	w.loss	LSTM	3dCNN	transformer	2dCNN
dacssGANs	✓	-	-	-	-	-
wGANs-3d	-	✓	-	✓	-	-
wGANs-LSTM	-	✓	✓	-	-	✓
wGANs-trs	-	✓	-	-	✓	✓
wGANs-trs-3d	-	✓	-	✓	✓	-
wGANs-im2im	✓	✓	-	-	-	-
wGANs-decoder	-	✓	-	-	-	✓
dacssGANs-tmp	-	-	-	-	-	-

4.2.3. Experimental results

In this section, we present the quantitative results for the first posed question (related to the evaluation of the Wasserstein loss research question) and then the quantitative results for the second posed question (related to the evaluation of the temporal information research question). Moreover, we present qualitative results from both cases which are displayed by the generated samples and qualitative interpretation of them.

Table 4.5: Classification performance, FID, IS and SSIM for all the methods associated with the first posed question.

Case	CREMA-D				RAVDESS			
	clas.	FID	IS	SSIM	clas.	FID	IS	SSIM
Baseline	49.34%				44.73%			
dacssGANs	52.52%	59.44	2.16	0.77	47.11%	49.77	2.21	0.90
wGANs-im2im	52.88%	51.55	2.50	0.90	49.81%	41.34	2.27	0.95
wGANs-decoder	51.17%	50.82	2.55	0.91	46.23%	42.11	2.32	0.94

Quantitative results regarding the first research question

In Table 4.5, all the results from the conducted experiments to investigate whether by incorporating the Wasserstein loss in our GANs scheme we can improve the efficiency of our approach are illustrated. The table contains the following:

- The first row of the table (referred as **Baseline**) contains the results of both datasets for the baseline evaluation approach.

- In the second row, the results obtained when using the approach implemented in Section 3.1 for the conventional GANs, denoted as dacssGANs, can be seen. As stated before, this approach managed to outbid the baseline evaluation regarding the classification metric (see in Section 3.1 and the statistical analysis in Figure 3.9). The classification performance is based on GANs augmentation scheme using real and generated data, while the assessment of the quality metrics is based only on the generated samples. The same goes for the rest of the cases.

- The third row (referred as **wGANs-im2im**) of the table is associated with the image-to-image implementation of the Wasserstein GANs. This approach reached slightly better results than the conventional GANs approach.

- Finally, the fourth row (**wGANs-decoder**) concerns the Wasserstein version but this time without the usage of the source domain.

Regarding the classification performance, when we apply the Wasserstein loss, we are able to slightly improve the gained results (approximately 0.3%) in comparison with the conventional GANs case. This is experienced though, only when the source domain is taken into account. For the **wGANs-decoder**, our approach failed at capturing emotion patterns during the generation of the spectrogram samples. Regarding the quality metrics on the generated images, there is a significant discrepancy in all three metrics between the conventional GANs and the Wasserstein GANs cases (see Table 4.5).

The best-obtained results (regarding the quality metrics) are found when the source domain is not taken into account (**wGANs-decoder**). Firstly, as a reminder regarding these metrics, the highest the IS and SSIM and the lowest the FID score the better the quality of the generated images. These insights are contradictory to the classification behaviour which analyzed in the previous paragraph. However, our hypothesis regarding this behavior is that **wGANs-decoder** approach managed better at learning the distributions of spectrograms in general, however, it failed at incorporating nuanced emotion patterns when approximating the audio distributions. This behaviour demonstrates the importance of the source domain information when training our wGANs approach.

As a supplementary evaluation, we measured the time discrepancy in the training of all the aforesaid approaches (the training process lasted for 100 epochs for all approaches). The training process occurred for all cases using the subset S_1 which is defined in Section 3.2 (the training subset). For each approach, the hyper-parameters analyzed in Section 4.2.2 are chosen. The training subset is processed through mini-batches of 64 samples for all performed experiments.

That is displayed in Table 4.6, where we can observe a huge time discrepancy between the conventional GANs and the approaches that implement Wasserstein loss. We further observed that there is a noticeable time discrepancy between the Wasserstein GANs (**wGANs-im2im**) and in the case where we alleviate the source domain from the input of the generator G (**wGANs-decoder**). We attribute that to two factors. Firstly, the architecture in the first example is more complicated (in terms of the amount of training

Table 4.6: The time performance for the experiments conducted regarding the first posed research question. In all cases, we measure the total amount of time after 100 epochs. All the experiments were performed in the same hardware (Titan XP GPU).

All conducted methods	Time in hours
dacssGANs	112h
wGANs-im2im	58h
wGANs-decoder.	43h

parameters and layers) and secondly, by adding the source domain as input, the convergence of the whole approach is slower.

Quantitative results regarding the second research question

In Table 4.7, all the results from the conducted experiments regarding the evaluation of temporal information are depicted. This table contains the following:

- The first row of the table contains the baseline results of both datasets (denoted as **Baseline**). These results are included for the convenience of the reader since they are the same results presented in Chapter 3.

- Secondly, the results obtained when using the approach implemented in Section 3.1 can be seen (denoted as dacssGANs). This approach managed to outbid the baseline approach. These results are exactly the same as in the case of Chapter 3 (for the supervised dacssGANs). The classification performance is based on the GANs augmentation scheme using real and generated data, while the assessment of the quality metrics is based only on the generated samples. The same goes for the rest of the cases.

- The third row (denoted as **wGANs-3d**), contains the results extracted from the Wasserstein GANs with input to the network *G* temporal features extracted using the 3dCNN approach.

- The fourth row (denoted as wGANs-LSTM), contains the result extracted from the Wasserstein GANs with input to the network *G* temporal features by using LSTM framework.

- The fifth row (denoted as wGANs-trs), contains the results extracted from Wasserstein GANs with input to the network *G* temporal extracted using transformer mechanism.

- The sixth row (denoted as wGANs-trs-3d) contains the results extracted from Wasserstein GANs with input to the network *G* temporal information extracted using the combination of transformer architecture and the 3dCNN algorithm.

- Finally, the last row (denoted as dacssGANs-tmp) contains the result of a modified version of conventional GANs. In this case, we keep only the decoder part of the network which is fed with features that are extracted using 3dCNN. It is obvious that this is the worst performed approach. The whole approach failed at generating samples which can help to improve the AER performance.

Table 4.7: Classification performance, FID, IS and SSIM for all the methods analysed in the current chapter for the Wasserstein GANs with temporal information as input associated with the second posed research question.

Case	CREMA-D				RAVDESS			
	clas.	FID	IS	SSIM	clas.	FID	IS	SSIM
Baseline	49.34%	-	-	-	44.73%	-	-	
dacssGANs	52.52%	59.60	2.16	0.77	47.11%	49.95	2.13	0.90
wGANs-3d	**55.87**%	38.55	2.65	0.91	**51.81**%	41.34	2.27	0.95
wGANs-LSTM	51.07%	**39.03**	2.60	0.87	49.84%	41.12	**2.32**	**0.96**
wGANs-trs	54.72%	39.15	**2.66**	**0.93**	51.76%	**41.10**	2.30	0.95
wGANs-trs-3d	53.09%	40.12	2.51	0.91	50.10%	41.12	2.21	0.95
dacssGANs-tmp	49.91%	69.77	1.97	0.65	41.90%	61.45	1.81	0.72

There are several insights obtained from the experimental phase, presented in Table 4.7. Firstly, the version of conventional GANs using temporal information (denoted as dacssGANs-tmp) failed to generate samples that will approximate the distributions of the real spectrograms and incorporate a rich emotional context. This became apparent from the classification performance, but also from the visual quality metrics.

Regarding the rest of the approaches, the best extracted results are found when using 3dCNN to extract the temporal information (in the third row, denoted as **wGANs-3d**). In comparison with the results of the approach implemented in Chapter 3, there is a classification improvement of 3.3% for CREMA-D and 3.0% for RAVDESS datasets. Regarding the quality metrics, it is obvious that with this approach we obtained superior results in comparison with conventional GANs, presenting the best results for some metrics (FID and SSIM for both CREMA-D and RAVDESS). That is also validated from the visual results that can be found between Figure 4.13 and Figures 4.9, 4.10, 4.11 and 4.12. However, with some other approaches like **wGANs-3d** and **wGANs-trs** we obtained better results for the quality metrics.

Then, slightly less efficient (regarding the classification metric) is the case of temporal information using transformer encoder (with input 2dCNN embeddings, denoted as **wGANs-trs**). This approach reached similar performance regarding the quality metrics, however, it performed slightly worse classification-wise than **wGANs-3d** regarding the GANs augmentation scheme.

Subsequently, in the sixth row of Table 4.7, where the temporal features are extracted using a combination of transformer and 3dCNN (denoted as **wGANs-trs-3d**), the performance is 53.09% for CREMA-D and 48.10% for RAVDESS. This approach managed to slightly overbid dacssGANs. However, this approach did not perform as well as the previous two approaches. Our hypothesis is that the transformer's encoder failed to capture correlations within different frames in a sequence by using 3dCNN embeddings.

Table 4.8: The time performance for the experiments conducted regarding the second posed research question.

All the conducted methods	Time in hours
wGANs-3d	46h
wGANs-LSTM	44h
wGANs-trs	48h
wGANs-trs-3d.	50h
dacssGANs-tmp.	104h

We believe that this contributes to the fact the temporal information from the input is not cascaded properly during the time back-propagation due to the complexity of the approach.

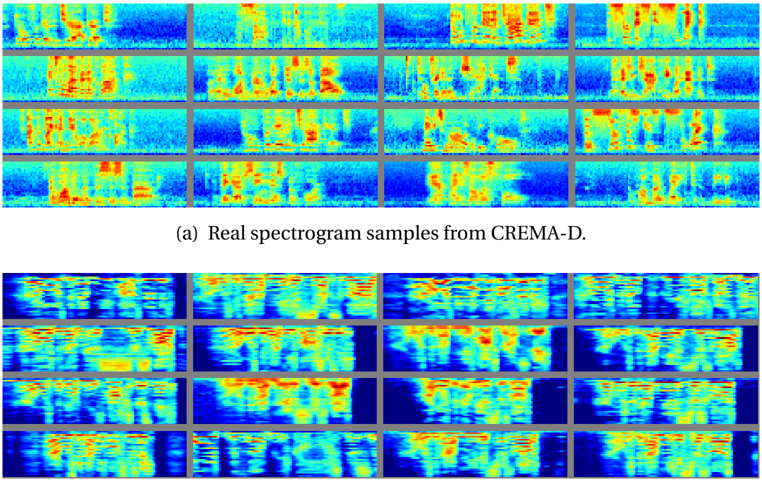
The last case concerns the extracted temporal features using LSTM (denoted as **wGANs-LSTM**). In this version, the methodology failed to properly generate samples that approximate the distribution of the real spectrograms with a rich emotional context. Yet, the generated samples performed well in the quality metrics. Again we can assume that the approach managed at learning the distributions of spectrograms in general, however, it failed when it comes to generating spectrograms with intense emotion information.

All things considered, it is apparent that the extracted results, when using Wasserstein loss, managed to approximate the distribution of the real spectrograms much better than conventional GANs. However, regarding the Wasserstein GANs, we do not observe a significant difference between the quality metrics. Hence, strong conclusions cannot be extracted regarding this aspect. Based on our findings, it appears that, by using 3dCNN to extract features, there is a small improvement in the IS and FID scores, while in the case of the SSIM, the best-performed case is when implementing LSTM as the temporal information extractor (denoted as **wGANs-LSTM**).

The duration of training of all the aforesaid approaches is measured to address the second posed question. Results can be seen in Table 4.8, where again it can be observed a significant improvement between the conventional GANs and the Wasserstein loss cases. Concerning the Wasserstein loss case, we measured the time needed to complete 100 epochs of training running in an Nvidia Titan XP. In particular, the training for the **wGANs-LSTM** lasted approximately 44 hours. Secondly, **wGANs-3d** and **wGANs-trs** performed similarly with 46h and 48h respectively. Finally, the most time-consuming case is that corresponding to **wGANs-trs-3d** with approximately 50h training time which attributed to the complicated nature of the model. The relation between time and performance is also studied (just for our best-performed approach the **wGANs-3d** case). It is analyzed in Section 4.2.4 (and displayed in Figure 4.15).

Qualitative results

The approach described in this section (with Wasserstein loss and using extracted temporal information) appears to be more promising in comparison to conventional GANs and the image-to-image approach for the generation of spectrograms, while it reduced significantly the training process duration by using the Wasserstein loss. On top of this,



(a) Real spectrogram samples from CREMA-D.

(b) Real spectrogram samples from RAVDESS

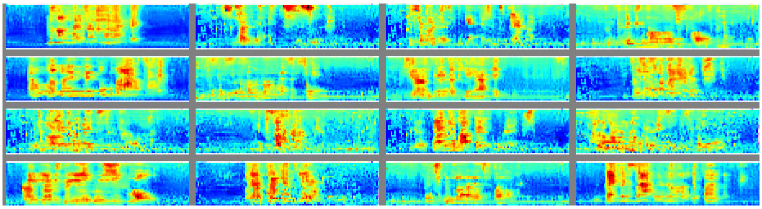
Figure 4.8: Real spectrogram samples from our implemented datasets.

it also managed to generate visual results of high quality. In comparison with the visual results obtained in Chapter 3, it is apparent that our approach managed to generate spectrograms with higher visual clarity and with more detailed patterns. This can be validated from all the quality metrics described in Table 4.7 and from the visual inspection of the generated results as well. Firstly, in Figure 4.8, real spectrograms from the CREMA-D and RAVDESS datasets are depicted. In Figures 4.9 and 4.11 the results for **wGANs-3d** and **wGANs-LSTM** approaches for CREMA-D and RAVDESS respectively, are shown. By comparing (by visual inspection) these images with the real spectrograms in Figures 4.8(a) and 4.8(b), we can conclude that both approaches managed to generate results that are close to the real ones. In Figures 4.10 and 4.12, results obtained using the **wGANs-trs** and **wGANs-trs-3d** methods are rendered. Again, it is noticed that the visual fidelity of the generated results is high and the visual inspection goes in parallel with the obtained results using the image quality metrics (FID, IS and SSIM).

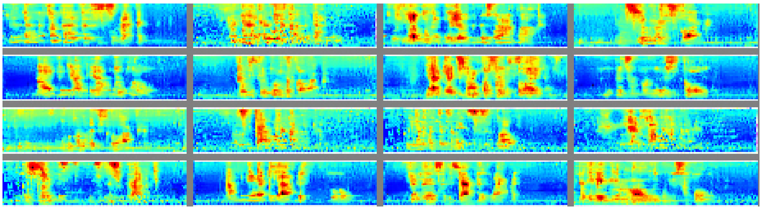
Finally, in Figure 4.13 we can see the results from the dacssGANs-tmp where it is obvious that this approach failed at approximating the distributions of spectrograms and any emotion-related information.

4.2.4. Training process

In order to examine the performance of the training process of Wasserstein GANs, we visualize the loss function of the Wasserstein loss during training (see Figure 4.15(b)), as well for the validation set. Furthermore, Figure 4.15(a) shows the accuracy during the training process (for the training and the validation set) when using **wGANs-3d** on the CREMA-D dataset. From both figures, it can be deduced that, after some epochs, the classification performance on the training and validation set, as well as the FID score are stabilized. For the IS score is harder to judge since there is high fluctuation during the

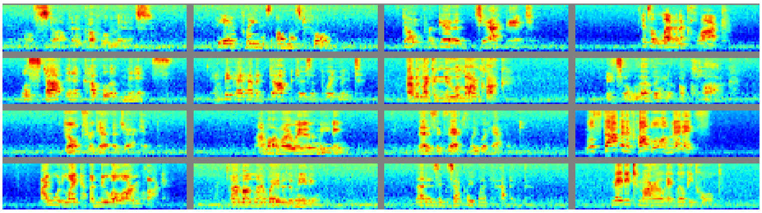


(a) Generated spectrogram samples produced using **wGANs-3d**.

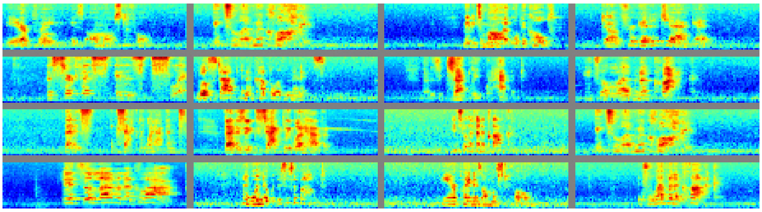


(b) Generated spectrogram samples produced using **wGANs-LSTM**.

Figure 4.9: Generated samples.

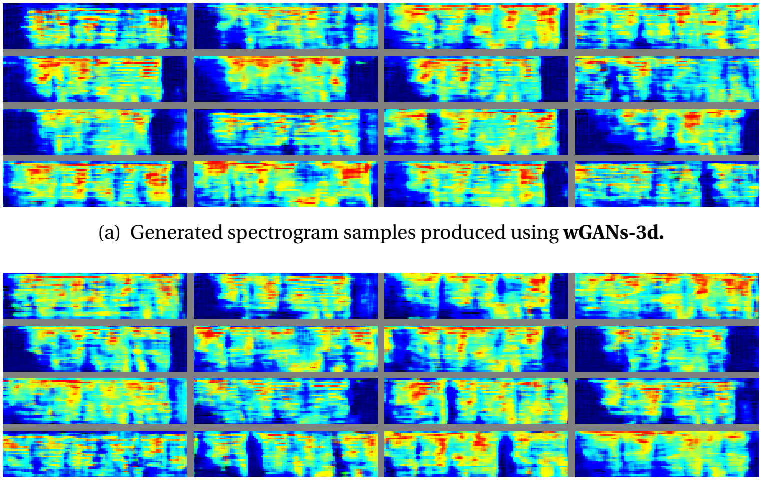


(a) Generated spectrogram samples produced using **wGANs-trs.**



(b) Generated spectrogram samples produced using **wGANs-trs-3d.**

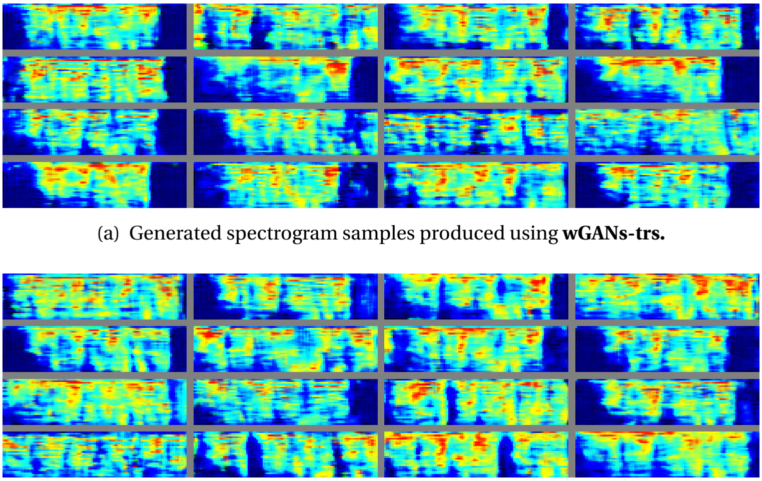
Figure 4.10: Generated samples.



(a) Generated spectrogram samples produced using **wGANs-3d.**

(b) Generated spectrogram samples produced using **wGANs-LSTM**.

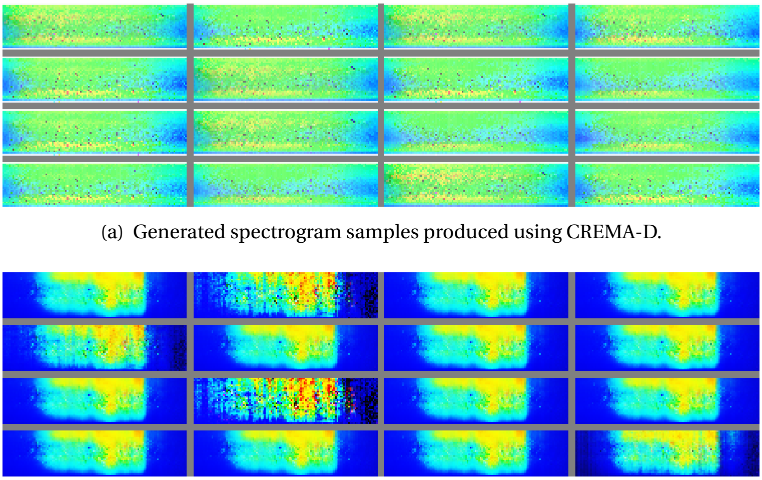
Figure 4.11: Generated samples.



(a) Generated spectrogram samples produced using **wGANs-trs.**

(b) Generated spectrogram samples produced using **wGANs-trs-3d.**

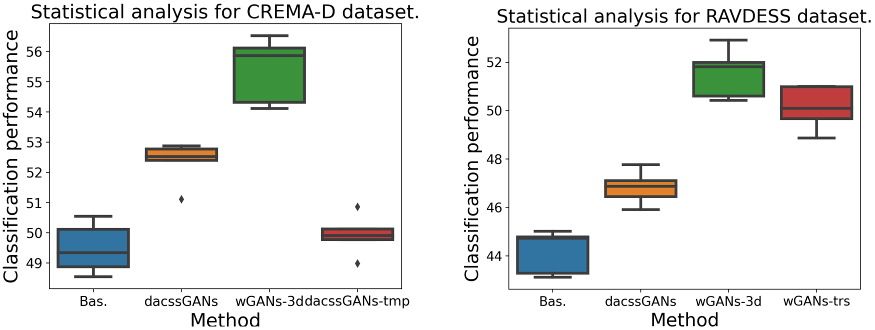
Figure 4.12: Generated samples.



(a) Generated spectrogram samples produced using CREMA-D.

(b) Generated spectrogram samples produced using RAVDESS.

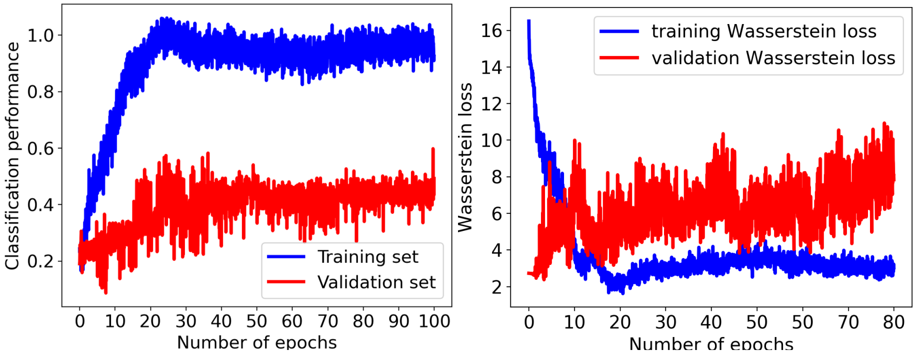
Figure 4.13: Generated samples.



(a) Methodologies for the CREMA-D .

(b) Methodologies for the RAVDESS .

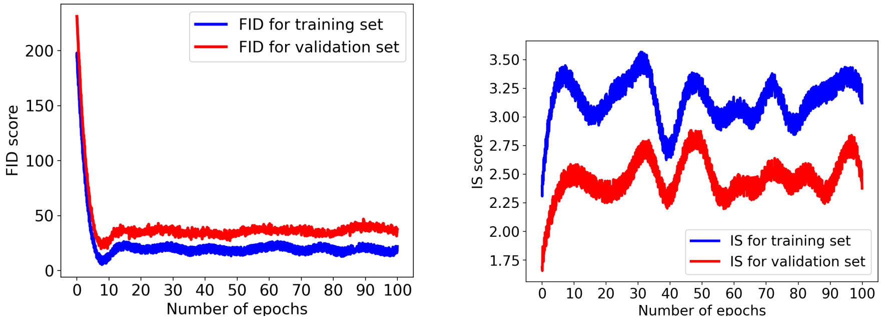
Figure 4.14: Std error bars for several applied methodologies.



(a) Classification accuracy.

(b) Wasserstein loss.

Figure 4.15: Training and validation information of **wGANs-3d** for CREMA-D.



(a) Féchet Inception Distance.

(b) Inception Score.

Figure 4.16: FID and IS values during the training procedure of **wGANs-3d** for CREMA-D.

training process.

For that reason, it is decided to perform early stopping after 55 epochs. That is aligned with the observation that the test performance on 100 and 55 epochs is approximately the same. Furthermore, the time needed for training can be further reduced. On this ground, we need to state that while we train in total for 100 epochs and report the time results for this number of epochs in Figure 4.8, when it comes to the performance (classification accuracy and metrics) we employ trained models for 55 epochs since the performance is the same.

To supplement the previous results, it is decided to perform an analysis of the quality of the generated results during the training procedure. In this manner, the training and validation performance is calculated using FID and IS during training (when using **wGANs-3d**). The performance for CREMA-D dataset can be seen in Figures 4.16(a) and 4.16(b) when using **wGANs-3d**. We can see for both measures and evaluations (in training and validation set) that, after the fifth epoch, there is a significant increase in the quality for the generating samples taking into account both FID and IS.

4.3. Conclusion

One of the main objectives of this dissertation is to study the problem of audio-visual domain adaptation and examine whether we can transfer knowledge from face to audio modalities to improve the AER efficiency. In Chapter 3, an approach based on generative adversarial networks (GANs) was introduced to accommodate the domain adaptation task. The goal of that approach, similarly to this chapter, is to generate spectrograms using, as input to the generator G, information from the source domain (static facial images) coupled together with conditional information $\mathbf{c}$ and a noise vector $\mathbf{z} \sim P(z)$. The final core task was to investigate whether AER could be improved by incorporating generated spectrogram samples from GANs to actual spectrograms, to expand the available training set.

In the current chapter, audio-visual domain adaptation within emotion-rich contexts is expanded by introducing a temporal conditional Wasserstein GANs approach. Conversely to the previous chapter, in this work, we focused on examining two key research questions. Firstly, our main focus is the study of temporal dynamics that govern audio and face modalities. To study the related cues, we modified our initial GANs architecture (presented in Chapter 3). In this case, instead of having as input static images from the source domain, we input temporal features extracted from sets of frames. Several approaches are studied for temporal feature extraction (3dCNN, LSTM, a transformer framework, and a combination of 3dCNN and transformer). Consequently, several additional modifications in the architecture of G needed to be performed to tackle our new task. After the experimental phase, it is observed that by using 3dCNN embeddings, the highest results could be obtained, achieving an improvement equal to 2.5% and 4.5% (for dataset CREMA-D and RAVDESS, respectively), in comparison to the models proposed in the previous chapter, where static visual features are employed.

Secondly, we examine whether training can be further improved by incorporating a state-of-the-art loss function called Wasserstein loss. As a result, we need to change the nature of the discriminator network D to encompass the new loss. From the experimental phase, it is proved that by implementing this new loss function, we could improve the complexity of the approach and reduce significantly the time of the training process. Furthermore, the quality of the generated results is improved significantly as well. This can be seen through visual observation of the obtained samples, but also by taking into account the image quality metrics introduced in Chapter 3, namely Inception Score (IS), Fréchet Inception Distance (FID) and Structural Similarity Index Measurement (SSIM).

From the experimental phase, we validated both claims raised at the beginning of the chapter. However, this chapter raises several future research lines as well. More specifically:

- The introduced transformer network is integrated into our GANs approach by extracting features from the source domain D_S (face modality) using only the stacked encoders of the initial transformer scheme introduced in [18]. Another possibility would be to investigate the same problem, making use of transformer attention mechanisms by employing the overall sequence-to-sequence framework. This would involve making use of both the encoder and decoder networks, similarly to the original machine translation problem [18]. In the problem tackled in this dissertation, instead of translating between two different languages, the goal would be to translate between facial and audio time series data.

- Thus far, in this dissertation, we studied transfer learning by always defining the face modality as the source domain D_S and the audio modality as target modality D_T. Therefore, we studied the generation of audio samples with the help of GANs using input data extracted from the face modality. However, the inverse problem was not addressed: how can we use the lessons learned so far in order to use audio information for generating emotionally expressive facial images not seen before? Chapter 5 deals with this problem, proposing necessary amendments to the topologies proposed so far, while results on the datasets used in previous chapters are presented.

REFERENCES

[1] M.Arjovsky, S.Chintala and L.Bottou, Wasserstein GAN, on arXiv, 2017.

[2] B.D.Gelder and J.Vroomen, The perception of emotions by ear and eye, Cognition and Emotion, Volume 14, Number Issue 3, Pages: 289–311, 2000.

[3] B.D.Gelder , J.P.Teunisse and J.B.Philip, Categorical Perception of Facial Expressions: Categories and their Internal Structure, cognition and emotion, Volume 11, Number 1, Pages: 1–23, 1997.

[4] W.J.Adams, The Development of Audio-Visual Integration for Temporal Judgements, PLOS Computational Biology, Volume 12, Number 4, Pages: 1–17, 2016.

[5] B.D.Gelder and J.Vroomen, The perception of emotions by ear and eye, Cognition and Emotion, Volume 14, Number Issue 3, Pages: 289–311, 2000.

[6] I.Gulrajani, F.Ahmed, M.Arjovsky, V.Dumoulin and A.Courville, Improved Training of Wasserstein GANs, Advances in Neural Information Processing Systems 30 (NIPS), 2017.

[7] G.J.Qi, Loss-Sensitive Generative Adversarial Networks on Lipschitz Densities, International Journal of Computer Vision, 2019.

[8] M.Arjovsky and L.Bottou, Towards principled methods for traning generative adversarial networks, ICLR, 2017.

[9] E.Levina and P.Bickel, The EarthMover's Distance is the Mallows Distance: Some Insights from Statistics, Proceedings of ICCV, Vancouver, Canada, 2001.

[10] J.Duchi, E.Hazan and Y.Singer, Adaptive Subgradient Methods for Online Learning and Stochastic Optimization, Journal of Machine Learning Research Volume 12, pages: 2121-2159, 2011.

[11] D.Ido, L.Lee and F.Pereira, Similarity-Based Methods For Word Sense Disambiguation, Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics:, 1997.

[12] S.Kullback and R.A.Leibler, On information and sufficiency, Annals of Mathematical Statistics, Volume 22, Number 1, Pages: 79—86, 1951.

[13] C.Donahue, J.McAuley and M.Puckette, Adversarial audio synthesis, ICLR 2019.

[14] J.Engel, K.K.Agrawal, S.Chen, I.Gulrajani, C.Donahue, and A.Roberts, GANSYNTH:Adversarial neura audio synthesis, ICLR 2019.

[15] D.S.Park, W.Chan, Y.Zhang, C.C.Chiu, B.Zoph, E.D.Cubuk and Q.V.Le, SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition, Interspeech 2019.

[16] S.Ji, W.Xu, M.Yang and K.Yu, 3D Convolutional Neural Networks for Human Action Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 35, Number 1, 2012.

[17] S.Hochreiter and J.Schmidhuber, Long short term memory, Neural Computation, 9(8), Pages: 1735-1780, 1997.

[18] A.Vaswani, N.Shazeer, N.Parmar, J.Uszkoreit, L.Jones, A.N.Gomez, L.Kaiser and I. Polosukhin, Attention Is All You Need, 31st Conference on Neural Information Processing Systems (NIPS), 2017.

[19] D. S. Park, W. Chan, Y. Zhang, C.C. Chiu, B.Zoph, E. D. Cubuk, Q. V. Le, SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition, Interspeech, 2019.

[20] J.Huang, J.Tao, B.Liu, Z.Lian and M.Niu, Multimodal Transformer Fusion for Continuous Emotion Recognition, Acoustics, International Conference on Speech, and Signal Processing (ICASSP), 2020.

[21] M.Bilkhu, S. Wang and T. Dobhal, Attention is all you need for videos: Self-attention video summarization using universal transformers, arXiv 2020.

[22] S.Zhi Y. Liu, X. Li, Y. Guo, Toward real-time 3D object recognition: A lightweight volumetric CNN framework using multitask learning, Computers and Graphics, Volume 71, Pages: 199–207, 2018.

[23] O.Cicek, A. Abdulkadir, S. S. Lienkamp, T. Brox and O. Ronneberger, 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation, International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2016.

[24] T. Zhou, S. Ruan and S. Canu, A review: Deep learning for medical image segmentation using multi-modality fusion, Array, Volume 3-4, Pages: 100004, 2019.

[25] D. W. Otter, J. R. Medina, and J. K. Kalita, A Survey of the Usages of Deep Learning for Natural Language Processing, IEEE transactions on neural networks and learning systems, Pages: 1–21, 2020.

5

DOMAIN ADAPTATION FOR IMPROVING THE FACE EMOTION RECOGNITION

The core insights which were obtained from the previous chapters can be summarized into the following key points:

- It was observed that it is possible to *transfer knowledge* from the face to audio modality by implementing a distance metric learning-based algorithm (Chapter 2). In this way, we can elicit information from the face modality and project it into a common sub-space with audio where a common classification can be held.
- Additionally, it is possible to increase the transfer of knowledge performance if we employ a deep learning architecture by utilizing Generative Adversarial networks (GANs) (Chapters 3 and Chapter 4).
- The conducted research in the previous chapter consolidated the performance of the Wasserstein loss which can improve both the training stability and the performance.
- Finally, the importance of taking into account temporal information from the source domain (face) X_S was also established (Chapter 4).

However, in order to accommodate the last posed research question from Section 1.6 (*Is it possible to improve the face emotion recognition by eliciting information from the audio modality?*), in this chapter, the objective is to perform the converse task. In this manner, we will investigate whether features from the source domain D_S (represented by audio modality this time) can be leveraged to improve Facial Emotion Recognition (FER). Our main motivation for this approach is two-fold:

- We would like to assess the performance of the most successful approaches for transfer learning found in the previous chapters when implemented for the converse task, namely face generation using audio information as input. In this way, it will be also easier to visually verify the performance of these domain adaptation approaches since it is much natural for humans to evaluate face samples rather than audio samples (especially in its spectral representation through spectrograms).

- Secondly, this study is also inspired by the work done in the cognitive psychology domain. From this perspective, authors in [1] performed several experiments regarding the audio-visual relationship within the emotion context and the way it is perceived by the human brain. Their results strongly suggest the existence of bidirectional links between affect detection structures in visual and auditory human systems. In greater prominence, it is the work associated with the first performed experiment (see in [1]), in particular: *the identification of emotion in the face and, whether it is biased in the direction of the simultaneously presented tone of voice.* The findings of this experiment validate the work of McGurk and MacDonald [2], illustrating that speech from a video is perceived in a different manner when the correspondent face from the video is also observed or not by the visual system of a human subject.

Having said the above, in the current chapter we intend to transform these cognitive psychology research questions to an affective computing perspective. We are mainly focusing on the research of performing domain adaptation from audio modality to face modality. In particular, the generation of emotion-driven faces by having as input to our system audio-driven information. In terms of related work, plenty of research work can be found in the domain of audio-driven generation of the face modality.

One of the most popular and provocative works is the notorious "Synthesizing Obama"[1] in [3]. This application became very popular in recent years, but it provoked a lot of controversy as well. The core idea is as follows: given the audio of President Barack Obama, the task is to synthesize high-quality videos of him speaking with accurate lip sync, composited into a target video clip. Trained by using a large amount of real videos of Obama speeches, an RNN learns the mapping from raw audio features to mouth shapes. Given the mouth shape at each time step, their application synthesizes high-quality mouth texture, which when is combined with a proper 3d pose can change what this person appears to be saying in a target video.

Authors in [4] proposed a technique called WAV2PIX which aims at generating face images of a speaker by using conditional Generative Adversarial Networks (cGANs) with raw speech input. The proposed system is composed of three distinctive modules. Firstly, the encoder of the input speech aims at extracting features from the raw audio signal. Secondly, the generator network G, which, given the extracted features from audio, is able to generate an image of size 128×128. Finally, a discriminator network D aims at judging whether the generated faces are real or not.

In the work done in [5], authors, research is focused on generating high-quality talking face videos by having as input an audio speech of a source person and a short video

[1] `https://www.youtube.com/watch?v=9Yq67CjDqvw&feature=emb_logo`

of a target person. Moreover, to perform the projection from the audio speech domain to face-lip motion with facial expressions, the talking face generation also takes into account the information from the target talking head such as head pose. The approach is mainly composed of two stages. Firstly, the mapping from audio-visual information to 3d facial animation. Then, the goal of the second stage is to perform realistic talking face video generation using the 3d facial animation from the previous stage.

Authors in [6] approached the video generation of "talking heads" by employing an end-to-end system by using as input a still image of the person and an audio clip containing speech, without relying on handcrafted intermediate features.

Their method generates videos which have (a) "lip movements" which are synchronized with the audio speech segments and (b) natural facial expressions such as blinks and eyebrow movements. The whole system is based on temporal GANs. This approach consists of three different discriminators focused on achieving increased frame quality, a proper synchronization between audio and video, and realistic generated expressions, respectively. The evaluation of the generated videos is performed based on their sharpness, reconstruction quality, lip-reading accuracy, synchronization, as well as their ability to generate natural blinks.

However, all these approaches are tasked to synthesize faces in a different setup and for different reasons than our current study. They are assigned not to improve Face Emotion Recognition (FER) but are mainly tasked to generate high-quality video sequences. On this ground, we decided to proceed with an architecture similar to the one proposed in the previous chapters. In particular, in this chapter, we employ Wasserstein GANs for generating samples from the face modality given as input features extracted from audio modality bound together with the noised vector $z \sim P(z)$ and conditional information related to annotations. However, during the process of structuring and organizing the research of the current chapter, new key challenges were encountered. These challenges are framed in the following questions:

- Firstly, what modifications should be done to the network's architecture and input size in order to incorporate the modifications of this current research (the audio is now the *source domain* while the face is the *target domain*)?
- We also need to determine what is the ideal input to the network G in an attempt to generate faces that properly represent the targeted emotions. For this reason, the following question would be addressed: what will be the format of the audio modality that we provide as an input to our generator G?
- What will be the output of our generator? Is it going to be a single frame image or a sequence of frames that represents the whole or part of the video clip?
- Finally, an evaluation schema must be established. In particular, we should answer the following questions: how can we evaluate the amount of the expressivity and emotional context of the generated samples?

On this ground, several modifications needed to be done in an effort to accommodate the necessary changes in comparison with Chapter 4. Firstly, we will use Wasserstein distance as a loss function to the whole architecture. Secondly, the same architecture with the previous chapter is utilized (as in Section 4.8) but this time we will modify

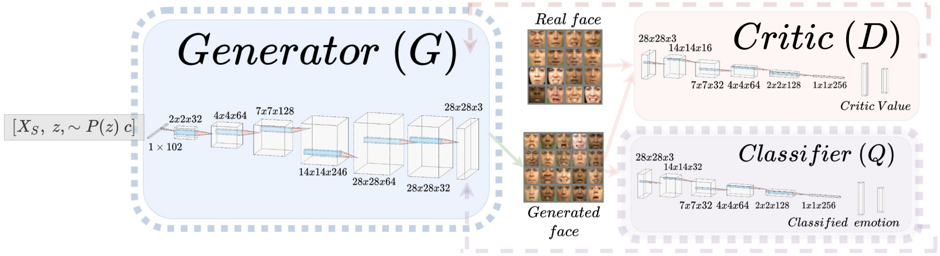


Figure 5.1: The core architecture that is implemented for face generation (single frame) GANs using information from the audio spectrograms.

the interior tensor shapes of our models (G, D, and Q) to meet the demands of the current task. In Figure 5.1, the proposed topology for this chapter can be seen.

Furthermore, a second strategy which aims at generating a sequence of facial frames instead of a single frame is also performed. Briefly, this is depicted in Figure 5.2. More information on this technique will be provided in Section 5.1.

In terms of evaluation of the proposed approaches, similar to Chapters 3 and 4, we perform the same evaluation schemes to gauge the performance of knowledge transfer between audio and face modalities. More specifically:

- The data augmentation scheme is performed as well. In this sense, a fusion between the real faces from the datasets with the generated ones using our GANs approach is performed (GANs augmentation scheme). More details for this comparison can be found in Section 5.1.2.
- Secondly, the quality metrics are also evaluated. In this vein, the IS, FID, and SSIM quality metrics are applied to the generated samples to measure their visual fidelity performance.
- An extra approach to evaluate the facial expressivity of the generated samples is used in an effort to determine how well the generated images contain facial poses with intense emotions (by employing the work done in [7]).
- Eventually, a visual inspection of the generated results was performed.

The structure of the remainder of this chapter is as follows: Section 5.1 describes the introduced DA method that is based on the proposed GANs architecture, while in Section 5.2 the experimental protocol, dataset, and results are presented and analyzed. Finally, Section 5.3 contains the conclusion and the future work of this study.

5.1. Domain adaptation for face emotion recognition approach

The core research conducted in this chapter can be framed by the following research question:

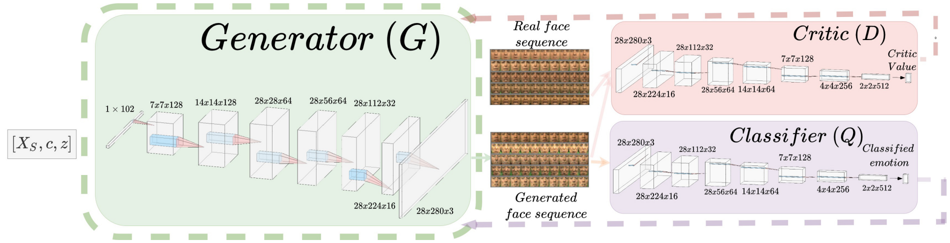


Figure 5.2: The core architecture in the second strategy of our approach for generating sequences of frames.

- Is it possible to improve face-based emotion recognition by eliciting information from the audio modality?

To answer this question, the best-performed findings from Chapter 4 are employed. In particular, Wasserstein GANs (wGANs) are applied by leveraging temporal information as input to our generator system in this chapter. On this account, we implement wGANs to create a system for generating data from the face domain given as input information from the source domain (this time, audio modality). To study this converse scenario, two distinct strategies are implemented:

- Firstly, a single face-frame wGANs approach with input audio-related features extracted from spectrograms representation using 2dCNN and as output, a single generated face-frame is utilized. In Figure 5.1, the architecture implemented for this strategy is visualised. We can observe that the input of the network G is the source domain X_S (represented by audio-related features) bound together with a noise vector and condition information related to the annotation of the audio.

- In the second strategy, instead of using the single face-frame approach, the network outputs a sequence of frames by giving as input to the network again the same features. This strategy's architecture is visualized in Figure 5.2.

5.1.1. NETWORK'S TOPOLOGY

Figures 5.1 and 5.2 depict the architectures for the two proposed strategies (the generation of a single frame and sequence of frames respectively). In these figures, we can see the tensor dimensionalities calculated after the convolution and max-pooling operations which change the size of the tensors. The output of the generator is either a single frame (in Figure 5.1) or a sequence of frames (in Figure 5.2). However, besides the convolutional and max-pooling operations, in the same spirit as with the previous chapters, we add the same operations: "dropout", "batch normalization" and activation function. "LeakyRELU" is added in the output of each convolution layer, while the activation function in the output layer is the hyperbolic tangent function (for more details see Sections 3.1, 3.1 and 3.1). The details regarding the generator G and discriminator D can be seen also in Tables 5.2 and 5.3. We need to note that the classifier network Q is the same network with the discriminator with the only difference between the last layer where instead of a binary classification we perform multi-class classification.

Table 5.1: Details about the employed networks. The variable θ stands for the dropout parameter (see Appendix 3.1).

Network	learning rate	Optimizer	θ
G	0.001	RMSProp	0.5
D	0.001	RMSProp	0.5
Q	0.001	RMSProp	0.5

Regarding the second strategy, the generation of sequences, for the implemented scheme, we need to modify the network G output and the input of the discriminator D. In particular, the desired output is a combination of 10 frames with size $28 \times 28 \times 3$ pixels. In this sense, for the adopted approach, the dimensionality of the output was $28 \times 280 \times 3$ pixels.

At this point, we need to highlight that the split of our datasets occurred in the same spirit likewise to Section 4.1.4 (for both strategies). However, there is one difference since we flipped the notation for the source and target domains. In this chapter, the source domain X_S is the audio while, the target domain X_T is the face. Moreover, for the second strategy, the source domain is represented by spectrograms, while the target domain is represented by extracted sequences of 10 frames. Each of these sequences of 10 frames was combined in this bigger image of $28 \times 280 \times 3$ pixels. An example of the combined 10 frames from a specific video of CREMA-D can be seen in Figure 5.3.

Finally, the critic D and the classifier Q networks received as input a tensor of size $28 \times 280 \times 3$ pixels. Both networks process this input using the following steps:

- They split the input into 10 distinct frames and then these networks are applied separately for each frame.

- For each of these frames, we apply both networks in the same way as in the case of the first strategy.

- Finally, an average of the loss for all these frames is performed, which in fact is the loss that these networks are back-propagating.

In Table 5.1 the learning rate, optimizer and the θ parameters for the dropout can be seen. In Tables 5.2 and 5.3, all the inputs and outputs for the generator and discriminator networks are depicted.

5.1.2. PRE-PROCESSING STEPS

In the following paragraphs, the basic pre-processing techniques for the audio and face modalities in this chapter are developed. Most of the pre-processing steps are similar to those described in previous chapters, however, some important key differences should be applied.

Table 5.2: The model employed for the generator network *G*.

Layer type	tensor input	tensor output
Dense	1 × 102	1 × 6272
Reshape	1 × 6272	7 × 7 × 128
Cov2d	7 × 7 × 128	7 × 7 × 128
Upsampling2d	7 × 7 × 128	14 × 14 × 128
Batch Normalization	14 × 14 × 128	14 × 14 × 128
Dropout	14 × 14 × 128	14 × 14 × 128
Cov2d	14 × 14 × 128	14 × 14 × 128
Upsampling2d	14 × 14 × 128	28 × 28 × 128
Batch Normalization	28 × 28 × 128	28 × 28 × 128
Dropout	28 × 28 × 128	28 × 28 × 128
Cov2d	28 × 28 × 128	28 × 28 × 64
Batch Normalization	28 × 28 × 64	28 × 28 × 64
Dropout	28 × 28 × 64	28 × 28 × 64
Cov2d	28 × 28 × 64	28 × 28 × 32
Batch Normalization	28 × 28 × 32	28 × 28 × 32
Dropout	28 × 28 × 32	28 × 28 × 32
Cov2d	28 × 28 × 32	28 × 28 × 3
Batch Normalization	28 × 28 × 3	28 × 28 × 3
Dropout	28 × 28 × 3	28 × 28 × 3

Table 5.3: The model employed for the discriminator network *D*.

Layer type	tensor input	tensor output
Conv2D	3 × 28 × 28	8 × 28 × 28
Batch Normalization	8 × 28 × 28	8 × 28 × 28
Dropout	8 × 28 × 28	8 × 28 × 28
MaxPooling	8 × 28 × 28	8 × 14 × 14
Conv2D	8 × 14 × 14	4 × 14 × 14
Batch Normalization	4 × 14 × 14	4 × 14 × 14
Dropout	4 × 14 × 14	4 × 14 × 14
MaxPooling	4 × 14 × 14	4 × 7 × 7
Conv2D	4 × 7 × 7	4 × 4 × 4
Batch Normalization	4 × 4 × 4	4 × 4 × 4
Dropout	4 × 4 × 4	4 × 4 × 4
MaxPooling	4 × 4 × 4	4 × 2 × 2
Dense layer	4 × 2 × 2	1 × 16]
Dense layer	1 × 16	1 × 512
Dense layer	1 × 512	1 × 6

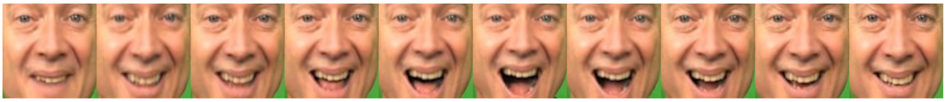


Figure 5.3: Sequence of 10 real frames from the CREMA-D dataset.

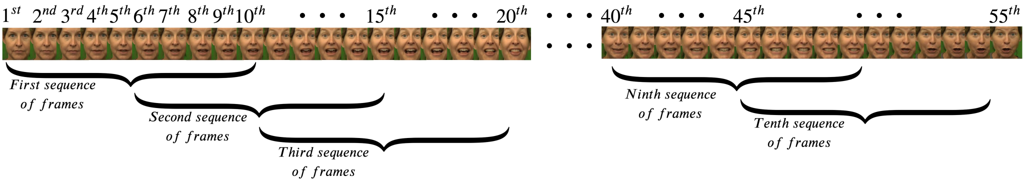


Figure 5.4: The way that the video frames are grouped into a sequence of frames.

FEATURE EXTRACTION FROM AUDIO MODALITY

As we already mentioned, we use spectrograms to represent the audio modality. This choice is made to keep the same image-to-image spirit introduced in Chapter 3. However, learning this image-to-image transformation in an end-to-end manner is proven to be a time-consuming task (and would have required big computation power). As a consequence, it is decided (similarly to Chapter 3) to firstly extract features from the spectrogram, and then, use these features as input to the wGANs generator G.

To extract these audio features, we use a 2dCNN classifier (which is depicted in Figure 3.7(a)). The stored model is retrieved. However, this time we want to extract features rather than perform an emotion classification using the spectrograms. Towards this end, the output layer of the model (that is responsible for the emotion classification) needs to be omitted. We are keeping the rest of the layers of the network (with the last kept layer to be a dense layer of 512 neurons). In this way, by applying this classifier, we could extract for each input spectrogram, embeddings (or otherwise denoted as feature vector) with the dimensionality to be 512.

PRE-PROCESSING STEPS FOR FACE MODALITY

In the same way as in the previous chapters, the pre-processing steps performed for each video, we are following the same strategy as the one explained in Section 4.1.4.

Firstly, it is decided to keep several frames for the middle part of each video. Subsequently, for each video, overlapping time-windows sequences of 10 frames are created. In this manner, the windows are slid by five frames each time, which results in an overlap of five frames between time-windows. This process is portrayed in Figure 5.4 where all the extracted sequences for one specific video clip are shown. For this chapter, we use a different number of middle frames (instead of only using the 55 frames used in previous approaches). This is done to examine whether we can increase the performance by keeping a larger amount of frames. In particular, four different approaches are utilized:

- By keeping the middle 55 face frames of the video in the same manner, as it was explored in Section 4.1.4.

- By keeping the 120 middle facial frames for each video while discarding the rest ones.

- By keeping the 10 middle facial frames for each video.

- By keeping 10 frames for each video. This time the frames are chosen based on their emotional content. To perform this, we employ an emotion classifier which is tasked to evaluate how strong is the emotion in each frame. This classifier is depicted in Figure 3.7(a) and its architecture details can be found in Section 5.1.1 and Table 5.3. In this table, the architecture of a discriminator network is depicted. The only difference with the classifier network is the last layer.

Simple data augmentation approach for face

Finally, for both strategies (generation of single and multiple frames), to increase the number of samples and for evaluation purposes, we need to perform a simple data augmentation approach. That is a necessary process to be in the position to properly train our deep learning networks.

In this simple augmentation approach for face modality, given as an input a single face (which is paired with a corresponding spectrogram), some noise is fused in order to be able to create new samples. The way that this is implemented is the following: Each time, we randomly pick pixels (from 1-10% of the image) and replace the pixel values with uniform noise. In this way, we are able to generate ten new samples and increase the number of data samples in our dataset. Figure 5.5 shows how the noise is fused with the input image. For one input image (the first image of the figure) this approach generates another nine.

Each of these new samples is paired with the same corresponding spectrogram from the initial sample. Therefore, we can generate 10 new face-spectrogram pairs.

In this way, we are able to increase the number of samples in the dataset, when we are in need of more data. This is particularly important for our second implemented strategy (the generation of a face sequence), and particularly in the case when we keep 10 middle frames. That is because in this strategy we need even more data for training, since we have fewer pairs of source and target domain samples. Furthermore, this augmentation approach is also important, since we will be able to fairly compare the proposed GANs augmentation scheme (where real and generated samples are fused) and the baseline data augmentation approach (where we increase the number of samples by performing this simple noise data augmentation). In this way, the GANs augmentation scheme for FER and our implemented "baseline" (based on this simple data augmentation scheme) should have the same amount of samples.

In order to disentangle potential conflicts in terminology, we are defining this approach as "simple data augmentation". This technique was applied and used for extracting the baseline FER. In the experimental results, this approach will be compared with a GANs-based augmentation scheme.

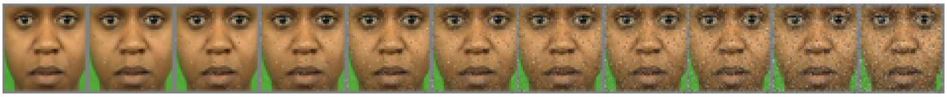


Figure 5.5: Creating 10 new samples using a simple data augmentation scheme based on noise.

5.2. Experimental phase for using temporal Generative adversarial networks for emotion-aware domain adaptation

In this section, we will discuss all the experiments conducted to support and evaluate the main objective of this chapter. This objective is related to the sixth research question found in Section 1.6 (regarding face generation) and the two proposed strategies analyzed in Section 5.1. It is important to state that in this chapter (similarly with all the previous ones), we employed the CREMA-D and RAVDESS datasets. Furthermore, the datasets were partitioned following the same scheme as in Chapter 3.

5.2.1. Experimental metrics

Firstly, the following experimental metrics are applied in this chapter, in similar terms with previous ones:

- A *GANs data augmentation* scheme (where the real and the generated samples are fused) is employed. It is important to state here that this data augmentation scheme is different from the one which is analyzed in Section 5.1.2 which is related to just increasing the sample number of the real datasets and do not use the generated samples.
- The *Inception Score* (IS), *Fréchet Inception Distance* (FID), and *Structural Similarity Index* (SSIM) are employed for evaluating the quality of the generated results using both our strategies.
- Moreover, the total amount of time regarding the training process was calculated. In this way, we are able to evaluate the efficiency of the approach time-wise.
- Finally, as an extra metric dedicated to the evaluation of the generated faces, we adopted the so-called expression net model (exp-net). Given a frame with a face as input, this technique is able to reconstruct the 3d pose of the face (as a 3d avatar) in the case that it contains a wealth amount of expressivity features.

Expression Net Model

This approach was introduced in [7], and is applied to evaluate the emotional expressivity of generated face samples by detecting the pose of the faces. The performed experiments made use of a pre-trained model publicly available on Github [2]. By applying this pre-trained model, the output is a 3d facial avatar where the emotion and facial pose of the input frames become apparent. Using this 3d model, it is easier to judge whether

[2] https://github.com/fengju514/Expression-Net

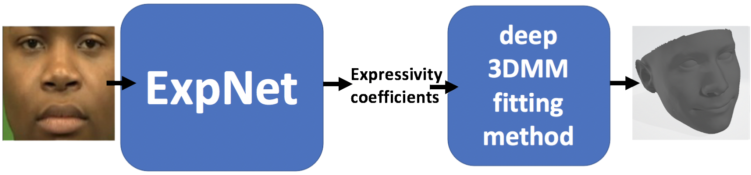


Figure 5.6: The way that our model was applied to generate the expressivity avatars. In this way, we made use of 3DMM-CNN model described in [8].

the generated faces consist of poses that are highly associated with specific emotions. Consequently, more concrete deductions related to the quality of the generated results can be extracted visually.

In particular, the authors introduced a system that estimates the facial expression coefficients (3D morphable model coefficients [8]) for a given face sample. A Deep Convolutional Neural Network (DCNN) model was introduced to perform the robust estimation of the facial expressivity features, directly from an unconstrained face image and without the use of face landmark detectors. Having calculated the facial expressivity features, the model is able to reconstruct a 3d graphic model of the input image by properly representing the emotion and using a simple 3d graphic avatar. This approach was performed by applying AFEW [10] and Cohn-Kanade (CK+) [11] datasets which use the basic emotion categories. More details about the implementation of the expression net model can be found in [7].

However, after experimentation, in our work, it was proven a constraint to apply this approach in images of low dimensionality. For instance, generated (or real) images with resolution 28 × 28 pixels did not provide accurate results. With this resolution, it was impossible for the approach to infer the poses of the input images. Through testing, it was proven to be applicable only to high-resolution images (higher than 100 × 100 pixels). On this ground, and just for performing this evaluation using exp-net, it was chosen as the lowest possible resolution, 112 × 112 × 3 pixels. Hence, the performed experiments just for the exp-net occurred to generate samples of this dimension.

As a result, we needed to perform some modifications in our proposed GANs architecture (Figure 5.1) in order to generate samples of this resolution. For this reason, we added a new set of operations in the proposed GANs architecture, and in particular, for the generator network of Figure 5.1. Thus, we added in the network two more convolution layers and two un-pooling layers together with the corresponding batch normalization, dropout, and activation function (LeakyRelu) operations. For the critic and the classifier, it was not necessary to add any new layer or operations.

As a consequence, due to all these extra operations (for generating samples of this resolution), it was obvious that the computational complexity of this experiment was significantly increased. Hence, it was chosen to perform this evaluation only for the first strategy of our current approach, namely for the generation of a single frame. Regarding the generation of a sequence of frames, in this case, this approach would have required the generation of multiple frames with size 112 × 112 × 3 pixels that would have exploded our needs for computational power and, furthermore, was not feasible to be done using our available computational resources.

5.2.2. EXPERIMENTAL PROTOCOL

In this section, we are analyzing all the experiments conducted to demonstrate the efficiency of the transfer of knowledge from audio to face modality when using the wGANs approach. Mainly, the performed experiments were related to the two strategies introduced in Section 5.1. However, several additional experiments were applied to evaluate the validity of these cases. In more details:

- Firstly, we extracted the FER performance using only real face samples from both datasets after performing the simple data augmentation scheme described in 5.1.2. Henceforth, this approach is denoted as *baseline*. To perform this experiment, we use the S_1 subset described already in Section 3.2 for training the classifier. In particular, after the aforesaid augmentation method the cardinality of the employed set is 2 times the cardinality of the set S_1. The performance of the baseline is measured using set S_4. The employed classifier is described in Section 5.1.1.

- Secondly, an implementation that made use of wGANs architecture, where the input was noise conjugated with the conditional information related to emotions from each video was conducted. This experiment was employed in an effort to measure the importance of the absence of the source domain (audio modality) to the input of the generator G (denoted as wGANs with a single frame without source information or **wGANs-sf-wt**).

- A natural extension of the previous experiment is to add the extracted features from the audio modality (2dCNN features) as the input to the generator (denoted as wGANs with a single frame in the source domain **wGANs-sf-wi**) in order to perform the image-to-image translation and domain shift.

- Furthermore, regarding the generation of the sequence of faces, we proceeded with the same two experiments (as in the case of the single-frame generation) by omitting or not the source domain denoted as **wGANs-se-wt** and **wGANs-se-wi** related to sequences and with/without source domain information.

- An extra performed evaluation, the quality of the generated results using the expnet model was also assessed. In this case, we only tested the **wGANs-sf-wi** using a modified version of the architecture, since the task was to generate samples with size 112 × 112 × 3 pixels (denoted in this Section as **wGANs-sf-wi-big**) by given as input to the generator G features from audio domain coupled together with a noise vector and conditional information c related to the annotation. Figure 5.6 shows how this approach is applied to an input image.

- Finally, for comparison purposes, the performance of the generated faces when using the dacssGANs architecture from Chapter 3 is applied.

Table 5.4: Emotion recognition using four different approaches for the CREMA-D dataset.

All the tested methods	Accuracy
55 middle frames	54.11%
120 middle frames	49.08%
10 middle frames	53.98%
10 frames with biggest emotion context	52.01%

5.2.3. EXPERIMENTAL RESULTS

Having analyzed the proposed structure and organization of the experimental protocol, in the following paragraphs, we will analyze the experimental results. By keeping the same spirit as in the previous chapter, we divide this section into three main parts, namely: ablation study, quantitative and qualitative results.

5.2.4. ABLATION STUDY

We are starting with insights from an ablation study related to facial frame extraction from the input video are presented. This approach examines which combination of frames from the input video will lead, during training, to the best FER performance.

In particular, we are exploring which combination of frames contributes the most to emotion recognition performance when they are used to train the above-mentioned GANs networks. This study is associated with the four different scenarios related to face frame selection (in the face pre-processing step) that are introduced and analyzed in Section 5.1.2. These scenarios regard the implemented policy for keeping some of the frames of each input video. In specific, these options are to keep the middle 55, 120, and 10 middle frames and finally, keep 10 frames based on their emotional context.

After training a FER model using all four aforesaid methodologies (using the same dataset split that was performed for the rest of the cases in this chapter), the obtained deduction during the FER testing phase is that the best approach is to keep 55 frames from the middle frames from the video where the best-obtained results are obtained (54.11%). The results for these four experiments can be found in Table 5.4.

This ablation study was performed just by using the CREMA-D dataset. The chosen methodology for extracting frames (using 55 frames) was implemented for the CREMA-D and RAVDESS datasets.

QUANTITATIVE RESULTS

On the whole, the quantitative results from the experimental protocol can be found in Table 5.5.

It is evident that when using the Wasserstein loss to generate single frames coupled together with source information (as in the cases of **wGANs-sf-wi**), we obtained the best classification results (concerning the GANs augmentation scheme using real and generated data) for CREMA-D dataset, while, for RAVDESS the best-performed approach is **wGANs-sf-wt** with the **wGANs-sf-wi** to perform very close regarding the data augmen-

Table 5.5: Classification performance, FID, IS, and SSIM for all the methods analyzed in the current chapter for the wGANs with temporal information as the input for the CREMA-D and RAVDESS datasets.

Case	CREMA-D				RAVDESS			
	clas.	FID	IS	SSIM	clas.	FID	IS	SSIM
Baseline	54.11%	-	-	-	49.12%	-	-	
dacssGANs	54.08%	59.60	2.16	0.77	50.30%	49.95	2.13	0.90
wGANs-sf-wt	53.20%	44.32	2.49	0.96	51.81%	39.88	2.30	0.92
wGANs-sf-wi	**55.97**%	46.09	2.57	0.96	**51.85**%	39.86	2.31	0.92
wGANs-sf-wi-big	54.12%	**43.23**	**2.77**	**0.97**	47.11%	**37.77**	**2.51**	**0.95**
wGANs-se-wt	52.66%	56.33	2.66	0.89	49.99%	45.57	2.13	0.82
wGANs-se-wi	51.10%	58.94	2.51	0.88	50.10%	45.58	2.01	0.83

tation classification metric. In particular, it is important to note that using the source domain extracted from the audio modality, we experienced an improvement in the performance of approximately ~ 1.9% and ~ 2.7% for CREMA-D and RAVDESS in comparison with the *baseline*.

Furthermore, it is also apparent that our second strategy, generating a sequence of frames, failed at generating robust results (regarding the GANs augmentation scheme using real and generated data) and the results were inferior in comparison with all the conducted experiments with single frame generation. In some cases, we even experience "negative transfer learning" for this approach. This could be attributed to the way that the critic D and classifier Q networks combined the frames in order to evaluate the performance of these multiple frames. This was confirmed by the extracted results when using this technique only on real data samples, where the same way of combining the frames was implemented to extract the classification performance. In this case, we obtained 52.44% for the CREMA-D and 43.79% for the RAVDESS dataset. A more sophisticated approach (like LSTM) should have been deployed for combining the sequence of frames for critic and network Q to improve the performance. However, due to our hardware limitations, this approach would have required access to more hardware resources, that were not available, and hence, we decided to perform it as future work. It is important to state here that it was decided to keep this strategy due to the interesting visual results.

Regarding the quality metrics (FID, IS, and SSIM), it is obvious again that wGANs for the single frame strategy (**wGANs-sf-wt** and **wGANs-sf-wi**) performed better than in the case of sequential generation (**wGANs-se-wt** and **wGANs-se-wi**) as well as for the dacssGANs approach in almost all metrics and for both datasets. That is attributed to the implementation of Wasserstein loss (in comparison with the **dacssGANs**). While in comparison with the sequential approaches (**wGANs-se-wt** and **wGANs-se-wi**) the difference is attributed again to the way that the critic D and classifier Q networks are combining the sequence of frames.

The best-obtained results regarding all quality metrics are obtained when using the **wGANs-sf-big** approach. That is validated also after the visual inspection of the generated results.

Table 5.6: The time performance for the experiments conducted in this chapter. In all cases, we measure the total amount of time after 100 epochs. All the experiments performed in the same hardware (Titan XP GPU)

All the conducted methods	Time in hours
dacssGANs	108h
wGANs-sf-wt	17h
wGANs-sf-wi	21h
wGANs-se-wt	98h
wGANs-sf-wt	113h

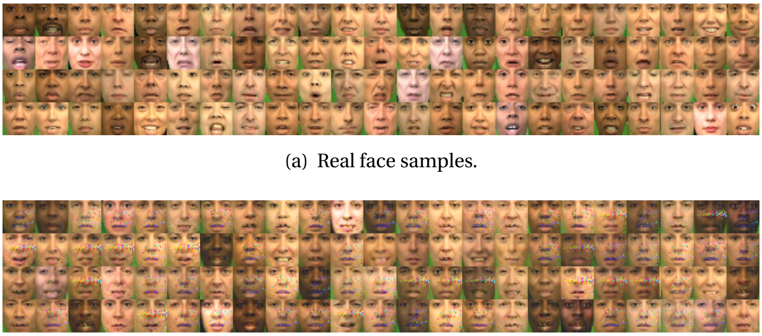
Furthermore, following the strategy described in Chapter 4, it was decided to calculate the total training time for the different approaches. In this way, we were able to assess the performance of all under-test approaches time-wise. These results can be seen in Table 5.6. These results were extracted when training using CREMA-D to give some intuition about the time-performance of the algorithms. Again, in a similar fashion as in Chapter 4, the total number of epochs for training was limited to 100 while all experiments were deployed on the same hardware, in particular using a Titan XP GPU.

It is noteworthy that when performing the sequential strategy, there is an increase in time duration in a ratio of 4-5 times in comparison with the single-frame generation methodology. This behavior was expected due to the more complicated network that is employed to handle this experiment.

QUALITATIVE RESULTS

Regarding the visual inspection of the generated results, several approaches were tested in order to evaluate the performance of the method. The performed evaluation contains the following examined cases:

- Generated samples using the dacssGANs approach (the implementation was based on the architecture found in Chapter 3).
- Extracted faces using the architecture based on wGANs with the single-face as an output (**wGANs-sf-wi**).
- The generation of sequences of frames instead of a single frame again using the wGANs architecture (**wGANs-se-wi**).
- These techniques (**wGANs-sf-wi** and **wGANs-se-wi**), evaluated with the implementation of source information.
- Furthermore, we inspected the performance of **wGANs-sf-wi** when generating images with their dimensionality to be 112 × 112 instead of 28 × 28 pixels (denoted as **wGANs-sf-wi-big**).
- The exp-net model was also applied to evaluate the expression wealthiness (for the **wGANs-sf-wi-big** approach).



(a) Real face samples.

(b) Generated faces samples.

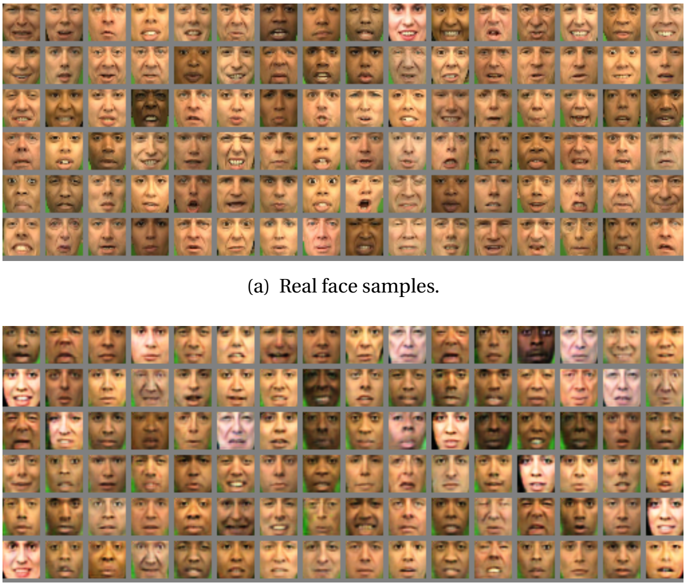
Figure 5.7: Generated visual faces when using CREMA-D during the test process for 28 × 28 × 3 pixels size using dacssGANs.

Single frame: Firstly, the visual results regarding the single frames generation is depicted in Figure 5.7 when generating faces using the **dacssGANs**. We can observe that, while the generated results have high visual fidelity and approximate the real ones, there are also several artifacts and deformations that reduce the aesthetic performance (as it was perceived from a subjective visual inspection). This observation is in parallel with the quantitative metrics performance of **dacssGANs** as it is depicted in Table 5.5.

As a further step, the performance of the approach proposed using the first strategy namely, in the case of the single-face generation (**wGANs-sf-wt**), when having as input to network *G* the features from the audio domain can be seen in Figure 5.8. Again from a subjective visual inspection perspective, it is obvious that the quality of the visual result is significantly improved (a claim that is supported by the quantitative metrics of Table 5.5 as well). We can observe that both the issue with the visual artifacts and the deformations experienced in the generated results in the case of “dacssGANs” were mitigated. In Figure 5.9, the extracted visual results of the same approach when using samples from the RAVDESS dataset can be seen.

Furthermore, in Figure 5.10, the generated “big” faces (with size 112 × 112 × 3 pixels) can be seen. These samples were generated in an effort to evaluate the performance of Exp-Net when inferring the emotional content of generated faces. However, we can also add here a visual analysis for the results. While it is a subjective impression, in our view, the generated results are of high quality.

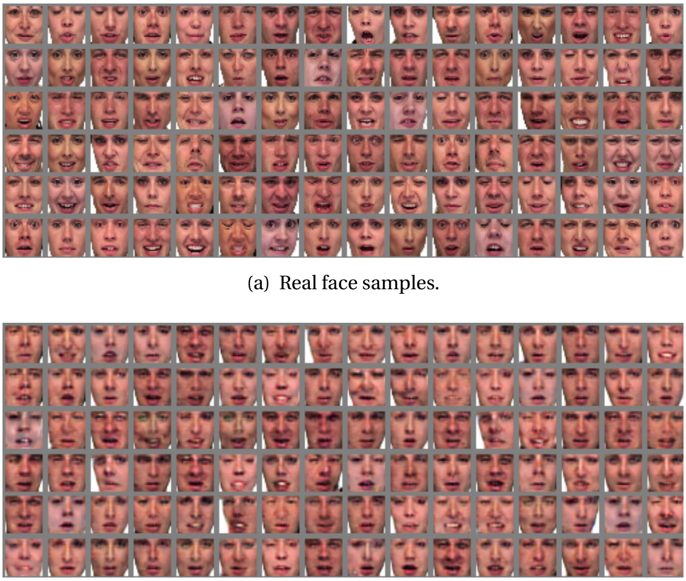
Finally, regarding single face generation, we performed one extra evaluation step by using the exp-net model. In this way, as it was mentioned in Section 5.2.1, this model was applied to the generated and, real images to estimate the amount of expressivity of these samples. In Figure 5.11 we can see the obtained results for both cases. The exp-net model was applied to the generated samples of **wGANs-sf-wi-big** (with targeted generated sample dimensionality to be 112 × 112 × 3 pixels). The derived inference from this experiment was that our approach can successfully generate samples that contain a wealth amount of facial expressivity features. On this ground, we can validate that the emotional context of the audio samples was transferred successfully to the face samples



(a) Real face samples.

(b) Generated faces samples.

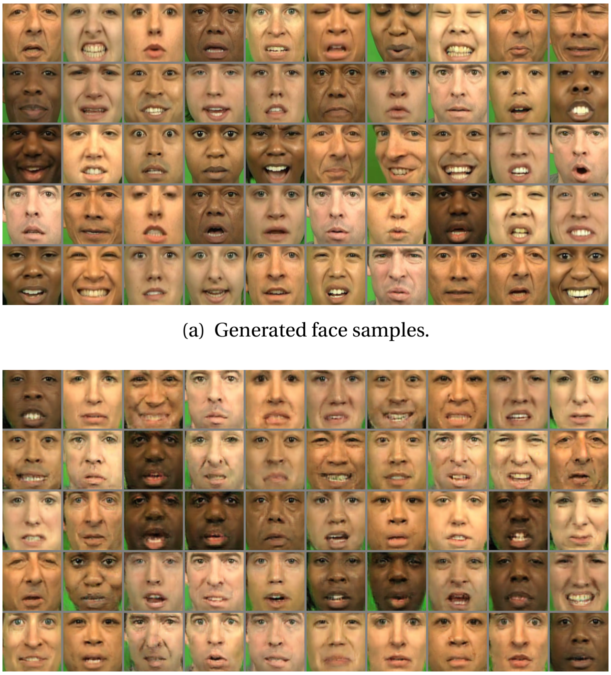
Figure 5.8: Real and generated faces when using CREMA-D during the test process for 28 × 28 × 3 pixels size using wGANs.



(a) Real face samples.

(b) Generated faces samples.

Figure 5.9: Real and generated faces when using RAVDESS during the test process for 28 × 28 × 3 pixels size using wGANs.



(a) Generated face samples.

(b) Real faces samples.

Figure 5.10: Generated visual faces when using CREMA-D and during the train process for 112 × 112 × 3 pixels size using wGANs.

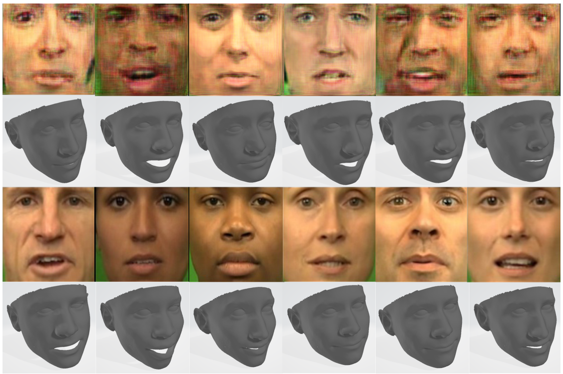
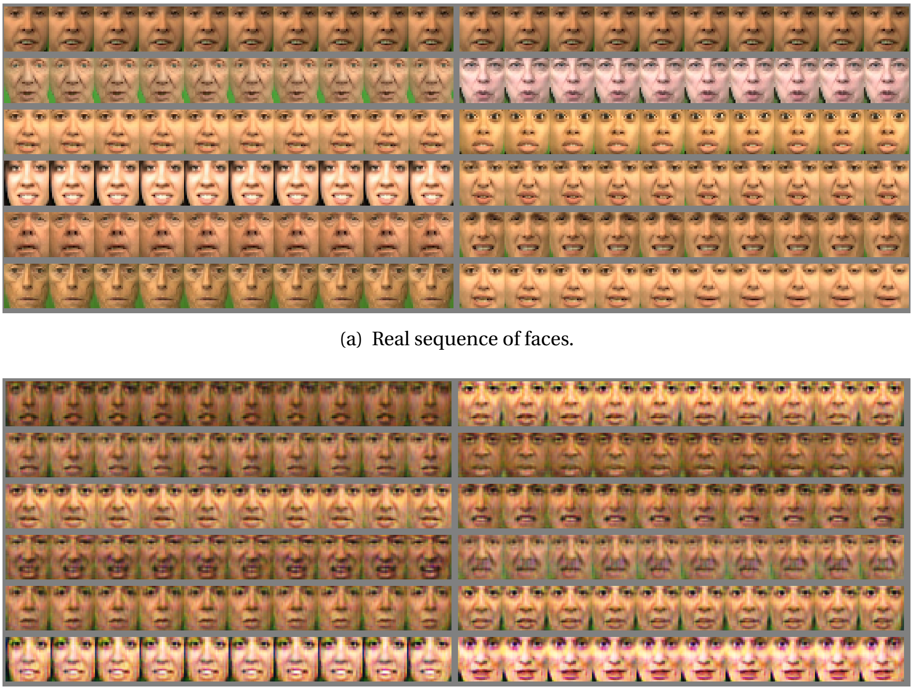


Figure 5.11: The extracted poses when using exp-net model to output the corresponding 3d model can be seen in the second and fourth row. These correspond to the real face samples which can be seen in the first and third rows.



(a) Real sequence of faces.

(b) Generated sequence of faces.

Figure 5.12: Generated visual sequences of faces for the CREMA-D dataset.

as well.

Sequence of frames: With respect to the second pillar of the current approach, the generation of sequences of frames, in Figure 5.12 we can see the associated visual results (for **wGANs-se-wt**). The same behavior was experienced in the case of **wGANs-se-wi**. It is evident that while the GANs augmentation scheme did not perform well for that case, still the generated results were close to the real ones. This approach clearly suffers from mode-collapse and over-fitting issues. The generated sequence samples were similar (even if the annotation and the noise vector were different). Moreover, during the training, the generated results were identical to the real ones (with the addition of some artifacts). We attribute this behaviour to the limited number of data samples for this second scenario.

5.3. CONCLUSION

The main effort of this dissertation is to study audio-visual domain adaptation and examine whether we can transfer knowledge from the source modality (face) into the target one (audio). The effort is to improve the classification performance of the target domain by leveraging information from the source domain.

In Chapters 3 and 4, two core approaches based on generative adversarial networks

(GANs) are introduced to accommodate the domain adaptation task. These approaches are employed to generate spectrograms using as input to the generator G information from the source domain coupled together with conditional information c and a noise vector $z \sim P(z)$. The final core task was to investigate whether Audio Emotion Recognition (AER) could be further improved by incorporating the generated samples from the GANs to model the real spectrogram datasets.

Alternatively, in the current chapter, we perform the converse task, where we aim at improving Face Emotion Recognition (FER) by leveraging information from the audio modality. In this sense, audio is defined as the source modality while face is applied as the target modality. The motivation for this research is mainly to prove the robustness of the domain adaptation model introduced in Chapters 3 and 4. Furthermore, in this case, it will be easier to visually investigate the obtained results of the proposed generative approach. Two different strategies are followed and tested, namely, single face generation and sequence of frame generation. Furthermore, to establish the performance of this approach, the introduced metrics from Chapters 3 and 4 are employed.

During the experimental phase, the hypothesis that FER can be improved after the GANs fusion scheme (where real and generated samples are fused) is shown to be valid. On this ground, the final research question posed in this dissertation is: *Is it possible to improve face emotion recognition by eliciting information from the audio modality?*. In order to confirm this hypothesis, a thorough experimental procedure is performed where several key deductions are obtained. More concretely, in this chapter:

- The performed architecture using wGANs and as input audio extracted information using 2dCNN, managed to create generated samples that, when fused with the real ones, improved the FER by approximately 1.9% for CREMA-D and 3.9% for RAVDESS datasets. Such finding further corroborates our belief that using as source domain audio-related information can enhance the FER performance.

- After a close visual inspection of the extracted generated results, we can deduce that the generated distributions of face images approximate well the real distributions of faces. Especially, the approach has proven to be very robust when it comes to generating “big images” of 112 × 112 pixels.

- Furthermore, regarding the second experiment (generation of a sequence of frames), while it is not possible to significantly increase the transfer learning performance, the extracted generated results are very promising visually but they lack of variety. Our hypothesis is that this behavior is attributed to the mode collapse issue. But also, in the way that we combine the sequence of frames in the classifier Q and critic D of the network. As an alternative, for future work, it will be to replace the whole generator G with more sophisticated architecture to incorporate the 3d temporal face information (for instance, 3dCNN or LSTM).

To conclude, we can state that the last research question posed in this dissertation is affirmed, where we investigated whether FER can be improved by using generated face samples from a wGANs architecture. At the same time, we validated that it is possible to transfer knowledge between audio and face modalities. However, several noticeable obstacles are faced. The “naive” approach for combining the facial frames failed in the generation of sequences of frames with quality that can be translated to improvements in FER performance. For this case, while the generation of a single frame has proven to be robust, it did not come with very competitive FER performance as well.

As future work, we can explore further the generation of frame sequences. In this respect, more sophisticated approaches for capturing the temporal information in consecutive frames needs to be investigated using algorithms such as LSTM or 3dCNN.

REFERENCES

[1] B.D.Gelder and J.Vroomen, The perception of emotions by ear and eye, Cognition and Emotion, Volume 14, Number Issue 3, Pages: 289–311, 2000.

[2] H.McGurk and J.MacDonald, Hearing lips and seeing voices. Nature, Volume 264, Pages: 746-–748, 1976.

[3] S.Suwajanakorn, S.M.Seitz, I.K.Shlizerman, Synthesizing Obama: learning lip sync from audio, ACM Transactions on Graphics, Volume 95, Number 4, 2017.

[4] A.Duarte, F.Roldan, M. Tubau, J.Escur, S.Pascual, A. Salvador, E. Mohedano, K.McGuinness, J.Torres, X.Giro-i-Nieto, Wav2Pix: Speech-conditioned Face Generation using Generative Adversarial Networks, International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019.

[5] R.Yi, Z.Ye, J.Zhang, H.Bao and Y.J.Liu, Audio-driven Talking Face Video Generation with Learning-based Personalized Head Pose, 2020.

[6] K.Vougioukas, S.Petridis, M. Pantic, Realistic Speech-Driven Facial Animation with GANs, International Journal of Computer Vision, 2020.

[7] F.J.Chang, A.T.Tran, T.Hassner, I.Masi, R.Nevatia and G.Medioni, ExpNet: Landmark-Free, Deep, 3D Facial Expressions, IEEE International Conference on Automatic Face and Gesture Recognition, 2018.

[8] A.T.Tran, T.Hassner, I.Masi, E.Paz, Y. Nirkin, and G.Medioni, Extreme 3D face reconstruction: Looking past occlusions. Computer Vision and Pattern Recognition (CVPR), 2018.

[9] P.Ekman and W.Friesen, Facial Action Coding System: A Technique for the Measurement of Facial Movement. Palo Alto: Consulting Psychologists Press, 1978.

[10] A.Dhall, R.Goecke, J.Joshi, J.Hoey and T.Gedeon, EmotiW 2016: Video and Group Level Emotion Recognition Challenges, Procs. of the 18th ACM International Conference on Multimodal Interaction, ACM, Pages: 427–432, 2016.

[11] P.Lucey, J.F.Cohn, T.Kanade, J.Saragih, Z.Ambadar and I.Matthews, The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression, IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2010.

6

Conclusions and Future Research

This thesis investigates how domain adaptation can be employed to perform audio-visual knowledge transfer to improve audio emotion recognition (AER) and face emotion recognition (FER). In Chapters 2, 3 and 4 we study the case of AER, while in Chapter 5 the case for FER is explored. The main effort in all chapters is to elicit information from the "source domain" and improve the classification performance in the "target domain". Towards this end, several research questions have been formulated to investigate cross-modal domain adaptation that should be addressed before answering the overall problem statement of this dissertation. This chapter provides a summary of answers to these posed questions (see Section 1.7). An overall answer to the posed research statement is given in Section 6.2. Finally, the last part of this chapter consists of recommendations for future research (see Section 6.3).

6.1. Answers to the Research Questions

In this section, a synopsis of the answers to each of the research questions posed in this thesis can be found. These posed questions and their answers help us to better grasp the formulated statement and the essence of the current research. The core inspiration of the current study lies in the domain of cognitive psychology, and the conducted research in the domain of knowledge transfer between the face and audio modalities from the affective computing perspective.

6.1.1. First research question

The first posed question is framed as:

> **Research question 1:** How could domain adaptation approaches be employed to improve emotion recognition in one modality and perform knowledge transfer from the other modality?

This is a rather general question and concerns all the main chapters of this dissertation (namely, Chapters 2, 3, 4 and 5). It refers to the possibility of applying domain adaptation techniques to perform knowledge transfer between two different modalities. As we have seen already in each chapter, multiple approaches are followed to tackle this question. In Chapter 2, a distance metric learning algorithm is employed to learn a common projection of these two modalities (face and audio). This approach, while simple, can provide a proof of concept for the fact that “knowledge transfer” is possible between the audio and face modalities. While it performs worse that the deep learning approach that is deployed in the following chapters, it is still a straightforward and elegant technique and its performance can be more easily explained than in the case of the “more” sophisticated deep learning techniques.

In Chapter 3, a different approach is followed to address the same problem. More specifically, we employed a generative machine learning technique (GANs), where the task is to generate novel audio samples with a wealth of emotion information by providing as input face samples. This approach validates that deep learning approaches can improve model performance in comparison to the deployed technique in the previous chapter. This motivated us to work towards this direction. In particular, in Chapter 4, an approach that takes into account the dynamics of the input face and a state-of-the-art approach for performing GANs (Wasserstein GANs) is employed. The observed performance in this work validates the fact that studying temporal dynamics is essential when modelling the cross-modal relations between face and audio domains. Finally, in Chapter 5, the best architectures found from the previous chapters are implemented to perform the converse task, that is to generate faces giving audio features as input to the GANs model. From the experimental phase of each chapter, it is proven that it is possible to elicit information from one modality to improve the AER or FER.

6.1.2. SECOND RESEARCH QUESTION

As we have seen in Chapter 1, to answer the above-mentioned generic question, we posed more detailed research questions that are associated with the nature of the employed techniques. In this spirit, the following question concerns the implementation of metric learning strategies (which are analyzed in Chapter 2) for our audio-visual domain adaptation task. This question can be framed as follows:

> **Research question 2:** How could a metric learning framework be deployed to model audio-visual cross-modal relationships?

In Chapter 2, a simple way to perform cross-modal domain adaptation is applied. This approach is based on a distance metric learning supervised approach, which is implemented to project both domains into a common space where the distance between samples from the face and audio modalities with common emotion context is minimized. This is performed in three basic steps: a) Feature extraction and selection, choosing emotion-related features from both modalities that contribute the most to the domain adaptation, b) secondly, an unsupervised technique is employed to transform both modalities into a common sub-space and finally, c) the implementation of the distance metric learning. The evaluation of this approach validated that it is possible to transfer knowledge from the source to the target modality. This is shown by using data from the source modality to improve the emotion recognition performance in the target modality. This observation confirmed our hypothesis that the model can be employed to perform cross-modal domain adaptation. Furthermore, given the simplicity of this approach this research worked for us as a proof of concept (and also a boost of confidence) for our dissertation claim that "knowledge transfer" is possible between the audio and face modalities.

6.1.3. THIRD RESEARCH QUESTION

Subsequently, a more sophisticated deep learning based approach is introduced in an effort to model the complex patterns that govern these cross-modal relations. This strategy is deployed since, after some preliminary experimentation, it is observed that better results can be obtained for audio emotion recognition using deep learning techniques. The employed architecture is based on Generative Adversarial Networks (GANs) and the posed research question is the following:

> **Research question 3:** How could Generative Adversarial Networks be deployed to uncover static audio-visual cross modal relationships?

An analytic answer to the aforesaid question can be found in Chapter 3. This approach's strategy is to generate novel audio samples by giving as input face samples coupled together with noise and label information related to emotion. Having given as input the face samples, a noise vector, and conditional information, our task is to generate audio spectrogram samples that are expected to convey emotional information. A fusion scheme is employed to evaluate the performance of domain adaptation in this case. It is shown that we can generate novel audio samples using a variation of the initial vanilla GANs architecture and deploy it to improve the AER of the initially limited datasets (by increasing the amount of data samples). This observation shows that there are hidden patterns that connect both modalities and therefore, we can model them by employing a deep neural network such as GANs.

Furthermore, now we are in the position to compare the results of this part of the research with the research from the Chapter 2. It is obvious that with the introduced deep learning approach the performance of the FER and AER was improved, as well as, the performance of the domain adaptation. Hence, this led to adoption of a deep learning strategy for the rest of the dissertation.

6.1.4. FOURTH RESEARCH QUESTION

Having performed the training of GANs for audio-visual domain adaptation, an obvious insight that is extracted is the difficulty and the large amount of time required to perform training. Hence, a new question is posed. This question is associated with the way that the optimization process of the whole network is held. On this ground, we pose the following question:

> **Research question 4:** How could we train the proposed Generative Neural Networks in a progressive manner? Or in simple terms, how can we initialize the weights of the network in such a way that will optimize the training process?

An analytical answer is given in Chapter 3. It is shown that an improvement of the process can be achieved by using a progressive way of training the whole GANs system. In this case, we first pre-trained the generator network *G* using an enormous unlabeled audio-visual dataset, while in a second step, we calibrated these weights emotion-wise. This observation validates our expectations, since the audio-visual modelling occurs in two steps by employing two enormous datasets and especially through the employment of Voxceleb in the first step.

6.1.5. FIFTH RESEARCH QUESTION

Furthermore, inspired by several proposed works that can be found in the literature, it is evident that an investigation of the temporal cross-modal relationships between face and audio modalities is imperative in our study. Towards this end, on top of the previously posed tasks, we decided to proceed with the study of the time dynamics between both domains. While this is partially done and described in Chapter 2, during the implementation of the DML strategy, it is developed in a more solid way in Chapter 4. The whole research endeavour of this Chapter can be framed based on the following posed question:

> **Research question 5:** Could the temporal dynamics between audio and visual information be modeled and help us in the domain adaptation task?

This research question is addressed in Chapter 4. By leveraging a novel algorithm of machine translation from Natural Language Processing, we model the dynamic connection between the face and audio modalities. It is confirmed that indeed the audio-visual temporal dynamics can be modeled and help improve the amount of transfer of knowledge between these two modalities. An improvement in the audio-visual domain adaptation is observed. This can be validated by the evaluation metrics employed in this dissertation such as data augmentation classification, as well as the introduced quality metrics.

6.1.6. SIXTH RESEARCH QUESTION

As is shown from the experiments conducted in Chapters 2, 3, and 4, audio emotion recognition performance is less than the face emotion recognition performance and this is validated also from the human performance in all of our implemented datasets. This is one of the reasons that in this dissertation we focused mainly on trying to improve audio modality performance by leveraging information from face modality. In the last research question of this thesis, we address the converse task. In particular, to improve FER by taking into account samples from the audio modality and performing the transfer of knowledge between these two modalities. The reason for performing this research is to evaluate the proposed architectures from previous chapters in a different task that can contribute to visual results that are easier to inspect. Hence, we formulated the following research question:

> **Research question 6:** Is it possible to improve face emotion recognition by eliciting information from the audio modality?

To that end, the architecture from Chapter 4 is modified to tackle the opposite task. In particular, the extracted 2dCNN features from audio modalities are given as input to GANs (as conditional information) and the output of the network is a face image having the same emotion with the input audio sample. This research question is validated in Chapter 5. It is observed that indeed it is possible to generate results with a great amount of emotion information by leveraging audio modality samples. This chapter works as a proof-of-concept. While the extracted observations are promising, some extra work needs to be done by addressing the shortcomings of our methodology, such as working in more depth towards sequence-to-sequence analysis. In this regard, an attention based architecture (such as transformers) can be employed for the generator network that will focus on extracting dynamic features from the input audio signal and connecting them with the output of the network *G* that will be a sequence of face frames. Hence, by using an approach like that, we can leverage the dynamic relationship between the two modalities.

6.2. Answer to the Problem Statement

After addressing the six research questions, an answer to the problem statement can be provided. Firstly, to remind our reader about the research statement, we can summarize it as follows:

> **Problem statement:** Is it possible to transfer knowledge between the face and audio modalities from the machine perspective when the emotion context is taken into account?

The answer to this problem statement is based on the answers to the research questions given above. We have shown that the DML approach can reduce the discrepancy between the face and audio modalities. This is validated by the experimental results where a metric to calculate the distance between the two modalities as well as the AER trained on a fused dataset with transformed faces and audio in the DML space is performed. Then, a more advanced approach based on progressive training of GANs is utilized. With this deep learning approach, we are able to perform better in AER and also validate the transfer of knowledge between the two domains. As we stated before, by studying the temporal dynamics and using a novel loss function for the GANs approach (Wasserstein loss), we experience even higher knowledge transfer.

Finally, our research statement is validated also in the converse task which is to improve FER by eliciting information from the audio modality. In this case, 3dCNN and Wasserstein GANs are employed to perform this projection.

6.3. Recommendations for Future Research

In future research, it would be interesting to investigate more techniques for performing cross-modal domain adaptation. All these techniques are related to the ones we currently researched and are extensions of our research with small modifications. In particular:

- Regarding Chapter 2, it will be interesting to investigate deep metric learning techniques instead of the standard DML approaches. Based on these techniques, neural networks are employed to learn a common representation of the two modalities and learn an ideal metric learning using efficient deep metric approaches such as “Siamese networks” or “triplet loss-based”. Thus, we hope that we will be able to learn a more meaningful sub-space where the two domains divergence will be minimized and the transformed face modality could be deployed to help increase the AER performance further.

- With respect to generative techniques for yielding new samples for the target samples by giving the source domain as input, a different approach to the problem could be our new focus. In particular, deep probabilistic generative models such as variational auto-encoders or probabilistic GANs such as ProbGANs, where the focus is on the probabilistic perspective of the generation process, are promising research directions. In this way, we could provide another remedy to the notorious mode collapse problem and improve the performance of the knowledge transfer.
- Another interesting idea would be to perform the whole learning scheme (from Chapter 4) end-to-end. Therefore, instead of using different architectures to extract features (especially temporal features from the face modality), a more sophisticated approach to include the feature extraction in the generator G could be analyzed. In particular, to model the temporal dynamics of face and audio in a common network, a solution based on attention mechanisms could be considered. This network would be our new generator G which can be tuned as usual in the *min-max* fashion.

IMPACT PARAGRAPH

In this addendum, a discussion is presented to introduce the scientific and social impact of the conducted research in this dissertation, its results, and the proposed methodologies. The core research of this dissertation is domain adaptation, that is applied mainly in Human-Computer Interaction (HCI) and Affective Computing (AC). However, while the main experimentation was conducted in the spectrum of these fields, in principle, the applied methodologies could be easily transferred to a plethora of diverse applications where domain adaptation could be useful.

All these aforesaid applications have an enormous social and economic impact on society. On this ground, according to Maastricht University's "Regulations for obtaining the doctoral degree Maastricht University", dissertations should encompass an impact section which should include the "short-term" and "long-term" contributions of the conducted research and its results in relation to shifting insights and stimulating science, methodologies, results, theory, and applications. On the other hand, the social impact relates to the short and long-term contributions of the conducted research to changes in the development of social sectors and to social challenges. This paragraph addresses the drafted four questions in the doctorate regulations, which are related to the main objective of the research and its relevance, its target groups, and activities.

> **Research:** *What is the main objective of the research described in the thesis and what are the most important results and conclusions?*

The main objective of this dissertation is to address an important research problem in machine learning, that is: performing domain adaptation from audio and visual cues. It approaches the task from different perspectives with various methodologies with the end goal of enhancing the performance of Emotion Recognition (ER) when it is gauged in one modality by leveraging information from the other. For instance, the task can be to improve Audio Emotion Recognition (AER) by leveraging information from the face modality. In particular:

- Chapter 1 introduces the task under study and the state-of-the-art approaches in the fields of domain adaptation, emotion recognition (FER and AER) with the focus on the ones that widely inspired this dissertation. Furthermore, Chapter 1 presents state-of-the-art technologies, datasets, applications, modalities' representations, and learning schemes.

- In Chapter 2, the domain adaptation study is performed from the Distance Metric Learning (DML) perspective. In this case, a proof-of-concept algorithm is developed to model the audio-visual relations and study whether face modality can help improve AER. This approach is composed of several modules such as: feature extraction and selection, clustering and the core DML projection. From the experimental phase, it is shown that it is indeed possible to transfer knowledge from face to audio modality.

- In Chapter 3, a deep learning direction is pursued. A study on Generative Adversarial Networks (GANs) is performed, with the purpose of discovering the correlation between face and audio modalities. Several methods are studied with the aim to build the proper architecture for the GANs network, and a proper way to tune the networks is also performed.
- As a follow-up research, a method to perform temporal analysis and study the temporal connection between face and audio modalities is applied in Chapter 4. This methodology makes use of 3d extracted features from face modality and attention mechanisms. A way to improve the training procedure of GANs architecture is also suggested.
- Finally, Chapter 5 studies the inverse task which is the improvement of FER using the audio modality. It is shown that it can be possible to increase the performance of the face modality by leveraging audio.

From the experimental phase, from all chapters it is clear that domain adaptation can be successfully applied to improve the performance of the audio or face modality by leveraging the other modalities and improving the AER and FER correspondingly. Chapter 2 provided a compact framework to perform domain adaptation, however, when we employed more sophisticated deep learning architectures (as in Chapters 3 and 4) we managed to outperform our initial results from Chapter 2. Moreover, we observed that it is really crucial to study the temporal relations between the two domains, which can lead to a more efficient "transfer" of knowledge between them. Finally, in Chapter 5 we performed a preliminary study on domain adaptation for the face modality. In this case, it was proven that it is possible to improve FER by employing audio information.

The next question that this chapter addresses is the following:

> **Relevance:** *What is the (potential) contribution of the results from this research to science, and, if applicable, to social sectors and social challenges?*

Our current era is mainly shaped from the so-called "digital revolution" in which the fields of data and computer science play a leading role. In the last decades, we are experiencing a constantly increasing interest in the fields of machine learning and deep learning in academia but also in industry. The "corporate world" has shown a great interest in investing in these fields and most of the big IT companies have already created their own dedicated "artificial intelligence" research and development department. This interest of the corporate world is also redeemed in our society since it equates to a shift in everyday life, which is now shaped through the omnipresence of modern technology. From mobile phones, smart TV's and electronic devices, to wearable health sensors and software that analyzes data for companies, states or individuals. In the near future, it seems likely that this tendency will increase and continue changing our lives. This tendency also fuels the popularity of research in machine learning and artificial intelligence in academia.

However, one of the notorious shortcomings of this cutting edge research is the so-called "lack of generalization". The developed machine learning algorithms need an enormous amount of datasets to learn how to perform a specific task, while, at the same time, they lack the flexibility to be employed in related tasks with slightly different characteristics and input datasets. Hence, in this dissertation we investigated a remedy for this notorious drawback which can be drawn from the research of the domain adaptation field.

Each chapter of this dissertation demonstrates the ability of the proposed solutions to perform DA efficiently between two inherently different modalities. This methodology can be used in a broader context, by applying it to different modalities. The importance of domain adaptation in research and as a consequence in society is beyond doubt.

By employing domain adaptation our purpose is to develop a more efficient framework that is able to combine data of a different nature to generate efficient models. In this scheme, we can leverage a big amount of data from different cues, which is crucial when developing a deep learning model since using only narrow data distributions is not really possible. Several popular deep learning and machine learning algorithms (object detection, language translation, face recognition, and so forth) can be benefited from this application.

The second contribution of this study is related to emotion recognition where the focus is to enhance emotionally incapable machines with emotional intelligence to improve human-machine interaction. Particularly, when the task is to perform emotion recognition in modalities for which we do not possess plenty of data. While the main focus of this dissertation is to perform domain adaptation, the task under study in each chapter is emotion recognition. We are modelling whether it is possible to perform domain adaptation with the purpose of enhancing the classification performance of a modality for instance, audio, by leveraging information from another modality. Hence, Chapters 2, 3 and 4 provide methodologies for performing emotion recognition from audio. While, in Chapter 5 we provide methodologies for performing face emotion recognition.

> **Target group:** *To whom are the research results interesting and/or relevant? And why?*

The conducted research concerns developers, practitioners, and researchers in the fields of "machine learning" and affective computing. In this work, we provide several frameworks for performing "heterogeneous" domain adaption between two inherently different modalities such as face and audio. We hope that researchers in the same and in similar fields will be inspired to continue research in this direction and will expand our research and ideas in new and interesting paths.

As aforementioned, this work was tested on the domain of affective computing and concerns the study of audio-visual relations. However, it can be easily transferred to different tasks and different modalities by performing the necessary modifications in the corresponding parts of the approach. For instance, in the case of performing a different classification than emotion recognition, we will need to change and retrain all the involved classifiers. One simple example is the following: performing person identification from audio that lacks annotated datasets by leveraging the huge availability of face recognition datasets. Towards this end, we can make use of the introduced techniques in this dissertation to generate audio samples by giving as input face samples, and leverage these generated samples to perform person identification.

Furthermore, our work can be useful to industry and developers that would like to develop robust classifiers in domains that lack large annotated datasets. In particular, they can leverage our approach by transferring knowledge from "close-related" domains to enhance the performance of the classifier at hand. A real-life tool for domain adaptation (part of the conducted research of this dissertation) was developed for the European Horizon research project called "MaTHiSiS" [1]. The scope of this tool was to improve emotion recognition performance from cues for which we do not have access to large datasets. Mainly, this tool was tested for performing audio emotion recognition by leveraging information from the face modality. However, this tool provides a friendly interface that can handle easily different modalities other that face and audio.

Some other applications where domain adaptation can be applied in industry is "the task of language translation", image classification for unseen objects, in gaming, in education applications and many more.

> **Activity:** *In what way can these target groups be involved in and informed about the research results, so that the knowledge gained can be used in the future?*

This thesis is article-based, where the studies in Chapters 2, 3 and 4 are published in various conferences and journal proceedings. At the beginning of each chapter, the papers which are parts of the corresponding chapter are listed. Moreover, throughout the course of the Ph.D. research, the proposed methodologies and the conclusions of their findings have been presented in the respective scientific venues. Besides, a tool for performing domain adaptation was developed for European Horizon2020 project called MaTHiSiS and was part of the whole learning framework.

[1] http://mathisis-project.eu/

Appendices

APPENDIX

In this Appendix, the basic principles of Machine Learning (ML) and deep learning (DL) are analyzed. We explore algorithms that were widely used in this dissertation and were not explained in depth in the previous chapters. The scope of this appendix is to provide the basic information of these algorithms to the reader. For more details regarding "Machine Learning" (ML) algorithms, an extremely influential and recommended reference is Bishop's book titled Pattern Recognition and Machine Learning [4] and Mathematics for machine learning [4]. For deep learning (DL) algorithms, an excellent book that helped us give a shape to this chapter is the book "Deep Learning" by Aaron Courville, Ian Goodfellow, and Yoshua Bengio [5] as well as [6].

1. MACHINE LEARNING

Firstly, it is important to introduce the reader to the basic terminologies of modern "Artificial Intelligence" (AI). One of the most commonly coined terms is "Machine Learning" (ML). From our perspective, the most efficient way to define this, is to cite popular quotes for this term such as:

- *Machine learning is the science of getting computers to act without being explicitly programmed, but instead letting them learn a few tricks on their own* [1].

- *Machine Learning at its most basic is the practice of using algorithms to parse data, learn from it, and then make a determination or prediction about something in the world* [2].

- *Machine learning research is a part of research on artificial intelligence, seeking to provide knowledge to computers through data observations and interacting with*

[1] https://www.coursera.org/learn/machine-learning

[2] https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/an-executives-guide-to-machine-learning

the world. Acquired knowledge allows computers to correctly generalize to new settings [3].

- Arthur Samuel described it as: “the field of study that gives computers the ability to learn without being explicitly programmed”. This is an older, informal definition.
- Tom Mitchell provides the following definition: “A computer program is said to learn from experience E with respect to some class of tasks T and a performance measure P, if its performance on tasks in T, as measured by P, improves with experience E”.

1.1. MACHINE LEARNING CATEGORIZATION

Having defined ML, we can also group ML into several categories. One common way of categorization is based on the way ML algorithms perform the learning process which are: **supervised**, **unsupervised** and **reinforcement learning** [8] ML techniques.

- In **supervised learning**, ML algorithms process the data examples (or alternatively called “data samples” or “data instances”) along with annotation information denoted as “labels” or “targets” ($y \in \mathbb{R}$ or $y \in \mathbb{R}^K$ in the case of categorical annotation) for each sample $x \in \mathbb{R}^D$. These labels represent the semantic category to which the corresponding sample x belongs. Two of the most common supervised machine learning tasks are **classification** and **regression**. For classification, the aim is to assign unseen data samples to a category (for instance in emotion recognition the samples are assigned to emotion classes like happiness, sadness, neutral and so forth). For regression, given a sample x the goal is to estimate a target value (for instance, the temperature forecast, and so forth) based on a set of features.
- In the **unsupervised learning** methodology, the employed algorithms attempt to uncover hidden patterns (in a new “latent space”) from the data themselves. There is a lack of annotation information for the given input samples $x \in \mathbb{R}^D$. Some common tasks in this category involve grouping similar examples together, called **clustering**, or performing **dimensionality reduction**.
- Regarding **reinforcement learning** (RL), it is different from the previous two cases, in the sense that RL performs actions directly, without learning a set of data first. In this case, RL directly performs actions and learns from these actions to re-calibrate its decision making strategies using rewards or penalties.

Another interesting way of grouping ML techniques is between a **probabilistic** and **non-probabilistic** point of view. In the first category, we seek to interpret ML algorithms using probabilistic modelling by either examining the data likelihood or by inferring posterior distributions. While in the second category, the target is to minimize directly a cost function that is related to the expected behaviour of the algorithm (in comparison with its real behaviour) without explicitly employing any probabilistic interpretation.

Finally, another possible way to categorize ML approaches is related to whether they are based on **neural network** approaches (and as an extra sub-categorization whether

[3] `https://becominghuman.ai/what-is-machine-learning-d292114cc6ce/`

they are related to **deep neural networks**) or not. This categorization is analyzed in further detail in Sections 2 and 3 where more insights into neural networks and deep learning are presented.

2. NEURAL NETWORKS

"Neural Networks" (NNs) is a sub-category of ML algorithms that are the reason for the recent hype about AI and ML and are responsible for the most state-of-the-art applications (face recognition, object classification, deep fakes, hand-digit recognition and so forth).

NNs are a family of algorithms whose structure reminisces an abstracted and simplified version of human brain topology. Similarly to the human brain networks that is composed of layers of simple entities called "neurons" and connections between these entities.

To explain better the functionality of these networks, we first introduce the so-called basic linear classification models that provide insights about the holistic structure of neural networks. In particular these algorithms try to model single neuron functionality.

BASIC LINEAR CLASSIFICATION MODELS

Before presenting the details behind these linear models, we need firstly to provide some generic terminology related to machine learning and neural networks. The following definitions will be helpful for the rest of the Appendix chapter (and for all chapters of the current dissertation):

- **Employed dataset**: A collection of data $\mathbb{D} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ with $\mathbf{x} \in \mathbb{R}^D$ (D-dimensional vectors) that are related to a problem that needs to be handled by the ML algorithm. For example, for a binary classification task related to the face detection problem, the dataset will be represented by a collection of images containing faces, and images that do not contain any face. In the case that each of these images has a size, for instance 28×28 (as in the case of MNIST [9]) then, we can represent each image by a vector of 784 dimensions.

- **Dataset sample**: One sample from the MNIST dataset $\mathbf{x}_i \in \mathbb{R}^D$ is an image that lies in $x_i \in \mathbb{R}^{784}$ space. Normally, these samples are represented by a vector of values which in our example represents all the pixels of the image. We can of course use a feature transformation ϕ (denoted also as "basis functions") to extract another meaningful representation of our input image (for instance, we can apply a filtering technique such as edge detection to obtain a new representation).

- **Sample features**: This term is associated with the values of the aforesaid vector ($x_{id} \in \mathbb{R}$, index i corresponds to the i-th sample of the dataset and d to d-th dimension of the data sample). In our example, for the MNIST dataset, each of these variables represent a single pixel of the images.

- **Target value**: As aforementioned, in the case of supervised ML the target is to estimate a specific class for the given problem. Therefore, the standard datasets, besides the samples, contain also some annotation information (that can be denoted alternatively as "labels" or "target values") which is related to the task that the ML needs to perform. It is normally denoted as $\mathbf{t} = \{t_1, ..., t_n\}$ (or as y) and represents information related to the classification task. For instance, for the MNIST dataset, the target value is all digits (0, 1, 2, ..., 9).

- **Weights**: "Weights" are trainable parameters that are deployed from ML algorithms and are associated to a corresponding task. In most algorithms, the values of these variables are randomly initialized as a first step. Moreover, during the training phase these values are updated to represent the input dataset as good as possible. The task is to optimize these values by minimizing a target error function that compares the target value with the algorithm output. More details about this process will be presented in the next paragraphs. These weights can be denoted as: $w = \{w_1, w_2, w_3, .., w_n\}$.
- **Bias**: A "bias" value w_0 is an extra trainable parameter that provides ML algorithms extra flexibility. As an example, in the case of a binary classifier, which is tasked to learn a decision boundary to split data in two classes, it allows to shift the decision boundary in a specific direction. In this way, it helps the training of the model to be faster and with better quality.

Having defined all these terms, we can now proceed with the explanation of linear classification algorithms. These approaches normally assume a model framed by the following form:

$$y(\mathbf{x}) = f(\mathbf{w}^T\mathbf{x} + w_0) \tag{1}$$

where $\mathbf{w}$ are the weights of the algorithm and w_0 is the bias. The function $f(.)$ is the so-called "activation function" that, given the inner product of weights with the input data samples and the bias term, returns one scalar value. Moreover, Equation 1 can be easily extended to:

$$y(\mathbf{x}) = f(\mathbf{w}^T\phi(\mathbf{x}) + w_0) \tag{2}$$

where ϕ is a transformation of the initial space ("basis functions"). This can be a transformation related to a feature extraction process, or the implementation of a standard basis function like a polynomial, for example. This can help to separate non-linear datasets using the linear operation of Equation 2.

PERCEPRTON ALGORITHM

One of the most prominent examples in the family of linear classification algorithms is the perceptron algorithm. It corresponds to a two-class approach which makes use of the model from Equation 3. The activation function in this case can be expressed by the following expression:

$$f(\mathbf{w}^T\phi(\mathbf{x}) + w_0) = \begin{cases} +1, & \mathbf{w}^T\phi(\mathbf{x}) + w_0 \geq 0 \\ -1, & \mathbf{w}^T\phi(\mathbf{x}) + w_0 < 0 \end{cases} \tag{3}$$

where $\{+1,-1\}$ corresponds to two possible target classes. The main functionality of the algorithm is to find values for $\mathbf{w}$ that can linearly separate data points that belong to two different classes.

A sane strategy in this case is to design an error function and perform a gradient-based optimization to find the optimal weights. One intuitive way to choose the error function is to use the total number of the mis-classified samples from the given dataset: $\mathcal{D} = \{\{\mathbf{x}_1, y_1\}, ..., \{\mathbf{x}_n, t_n\}\}$ with $t_n \in \{+1,-1\}$. However, this is proven to be meaningless since, in this case, we cannot calculate the derivative over $\mathbf{w}$.

Another strategy is to utilize the following error function:

$$E_{n\in\mathcal{M}} = -\sum_n \mathbf{w}^T \phi(\mathbf{x}_n)\, t_n \tag{4}$$

where $n \in \mathcal{M}$ are the mis-classified indexes of the corresponding data points. In this way, the error function is a positive value and its amount is related to how many how many samples were misclassified during the training process. To optimize its values $\mathbf{w}$, the algorithm makes use of a gradient-based technique, for example gradient descent:

$$\mathbf{w}^{\iota+1} = \mathbf{w}^{\iota} - \eta \nabla E(\mathbf{w}) \tag{5}$$

where the derivative of the error function E over $\mathbf{w}$ can be easily calculated, and in a vectorized form is $-\phi(\mathbf{x}_n)^T t_n$. This error can be calculated in steps after passing each sample from the dataset.

To make the algorithm complete, an initialization step is defined during which the weights $\mathbf{w}$ are randomly picked, and the iteration loop is performed. This algorithm can be run after processing the available samples from the dataset one or multiple times (epochs).

The perceptron learning algorithm loops through the training data samples in turn, and for each sample x_n the algorithm evaluates the error function. When the sample is properly classified, then the weight vector does not change, while when it is incorrectly classified, then for class C_1 it adds the vector $\phi(\mathbf{x}_n)$ onto the current estimate of weight vector $\mathbf{w}$ while for class C_2 it subtracts the vector $\phi(\mathbf{x}_n)$ from $\mathbf{w}$. This approach is called "training" since the algorithm learns the optimal weights gradually in steps.

Convergence theorem: One important characteristic of this algorithm is *the perceptron convergence theorem* which states that if there exists an exact point solution for the weights, then the algorithm is guaranteed to discover the exact solution in a finite number of steps.

Algorithm drawbacks: However, on the negative side, the drawbacks of the perceptron algorithm can be summarized as follows:

- This algorithm cannot be interpreted probabilistic perspective since it does not output a probability but a simple output y.

- Also the algorithm is restricted to two classes and cannot handle multi-class datasets.

- The feature mapping ϕ needs to be defined beforehand. For each different dataset we need to employ a different basis function.

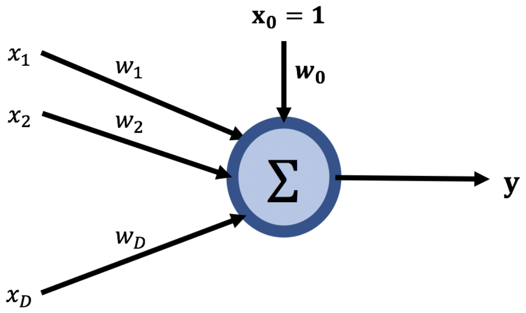


Figure 1: Visualization of a single neuron with x_i to be the input to the neuron, w_i the learning parameters and y the output of the algorithm. The bias term is represented by w_0 and in order to easily vectorize the linear calculations we add as $x_0 = 1$.

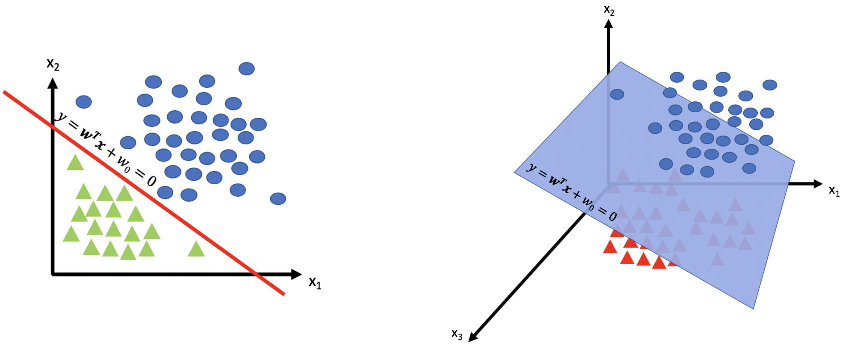
- Most importantly, this approach can handle only datasets that are linearly separable (or can be linearly separable when applying a specific combination of the basis function). For data sets that are not linearly separable, the perceptron learning algorithm will never converge.

- Even when the dataset is linearly separable, there may be many solutions.The found solution quality will depend on the parameter's initialization and the order in which the data points are presented to the algorithm.

Geometric motivation The input vector $\mathbf{x}_i = \{x_{i1}, x_{i2}, .., x_{iD}\}$ (representing here a data sample) can be visualized as points in the D-dimensional space. Then, the weight summation $\mathbf{w}^T\mathbf{x}_i$ defines a hyperplane in D-dimensions that can be shifted by using the bias parameter. Actually, the bias term w_0 is embedded in the weight summation by multiplying with a new variable $x_i 0 = 1$. Figure 1 provides a visual representation of this weight summation procedure.

In Figure 2, the hyperplane that is calculated using the weight summation for the two and three-dimensional space is displayed. The final task of perceptrons is to segregate the space in two half-spaces as can be seen from this image using these hyperplanes. In this sense, the perceptron can be seen as an algorithm that is tasked with learning a linear function (represented by the hyperplane) which is able to perform a binary classification.

Another useful representation of the perceptron algorithm (and in general of linear classifiers and SVM as well) can be found in Figure 3. Here again we can see the same hyperplane as in the case of .2(a). We can apply also some intuition from linear algebra, and more specifically, we can state that the weight $\mathbf{w}$ is orthogonal to the hyperplane and therefore it can define its direction, while the bias value amounts for the distance of the hyperplane to the origin. Finally, for a specific data sample $\mathbf{x}$ its distance to the hyperplane amounts to: $\frac{y(\mathbf{x})}{||w||}$.



(a) A hyperplane in case of two dimensional input space.

(b) A hyperplane in case of three dimensional input space.

Figure 2: The way that the weight summation operation.

LOGISTIC REGRESSION

Similarly to the perceptron algorithm, logistic regression is another type of linear classification model which tackles some of the limitations of the perceptron. One fundamental difference is the use of another activation function which, given the weight summation, outputs probabilities. In particular:

$$y(\phi(\mathbf{x})) = \sigma(\mathbf{w}^T \phi(\mathbf{x}) + w_0) \tag{6}$$

where σ function is the "sigmoid activation" given by the following expression:

$$\sigma(\alpha) = \frac{1}{1 + exp(-\alpha)} \tag{7}$$

The motivation for this activation function is derived from the calculated posterior probability for classification, which, by applying Bayesian rule for a sample $\mathbf{x}$ can be framed as:

$$p(C_1|\mathbf{x}) = \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_1)p(C_1) + p(\mathbf{x}|C_2)p(C_2)} \tag{8}$$

which can be easily shown that is of sigmoid form if we divide everything with $p(\mathbf{x}|C_1)p(C_1)$ and make the proper replacements.

Having defined our model and the activation function, in a similar fashion with perceptron, we need to define the error function to calculate the gradient over $\mathbf{w}$. The error function that is implemented for the case of logistic Regression is the *cross-entropy* and has the following shape:

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln(y_n) + (1 - t_n) \ln(1 - y_n)\} \tag{9}$$

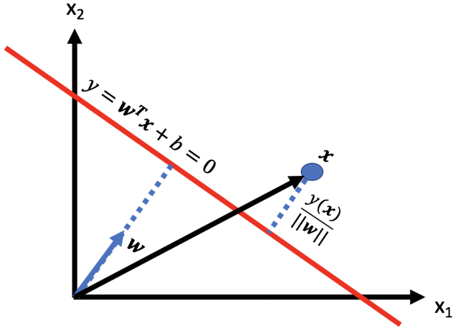


Figure 3: Geometric interpretation of perceptron

where $y_n = \sigma(\mathbf{w}^T\phi(\mathbf{x_n}) + w_0)$ is the prediction of our model. The importance of this error function can be explained form the "Information Theory" perspective. For more details, see Section 1.6 in Bishop's book [4].

Multi-class Logistic Regression Logistic Regression can be expanded also to take into account datasets with multiple classes. Towards this end, instead of the sigmoid function we make use of the softmax activation function, which in fact, is the extension of the sigmoid function for the multi-class case. Note here, that in this case instead of one output and one vector of weights, we have one output for each class and one vector of weights for each class. Therefore, the output y for the $k-th$ class can be re-formulated as:

$$y_k(\phi(\mathbf{x})) = \frac{e^{\alpha_k}}{\sum_j e^{\alpha_j}} = \frac{e^{\mathbf{w}_k^T\phi(\mathbf{x})+w_0}}{\sum_j e^{\mathbf{w}_j^T\phi(\mathbf{x})+w_0}} = \zeta(k) \tag{10}$$

with $\alpha_k = \mathbf{w}_k^T\phi(\mathbf{x}) + w_0$, and we can see that the process is calculated once for each of the K different classes, and therefore, we have K different weights $\mathbf{w}$. The derivative of this function can be analyzed for a specific index i to be the following:

$$\frac{\partial y(\phi(\mathbf{x}))_k}{\partial \mathbf{w}_i} = \frac{\partial \zeta(k)}{\partial \mathbf{w}_i} = \zeta(k)(\mathbb{I}_{kj} - \zeta(i))) \tag{11}$$

where $\mathbb{I}_{kj}$ is equal to one when $k = j$ and zero otherwise. While the cross entropy in this case can be expressed as follows:

$$E(\mathbf{w}_1, ..., \mathbf{w}_k) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}ln(y_{nk}) \tag{12}$$

For this form of the cross-entropy is important to note that we use categorical vectors for the labels t_{nk} (with n to represent the n-th sample). The dimensionality of these vectors is K (where K is the total number of classes). Finally, the gradient of the error function can be calculated and it is:

$$\nabla_{\mathbf{w}_j} E(\mathbf{w}_1, ..., \mathbf{w}_K) = \sum_{n=1}^{N} (y_{nj} - t_{nj})\phi_n \tag{13}$$

The update of the weights can occur in a similar fashion with the perceptron algorithm using a gradient-based iterative algorithm like stochastic gradient descent.

Finally, by using this approach we can overcome two of the perceptron's drawbacks namely:

- Firstly, with logistic regression, we can have a probabilistic interpretation of the output by making use of the sigmoid function as activation function.

- Secondly, by applying the softmax activation function, the LR algorithm can be extended to the multi-class case.

2.1. Multi-layer perceptron

All the aforesaid basic operations are very interesting and intuitive to understand, however, they have been proven to be simplistic, and, they can only model linear representations. Furthermore, when it is necessary to learn a more complicated and meaningful separation plane for the input data (which cannot be expressed by linear separable classes), we need to employ manually a basis function ϕ before performing the training process, which as we mentioned before, needs to be different given different datasets. In this case, we need to be familiar with the input datasets and the underlying distributions of these datasets, which are not always easy to obtain.

To mitigate this challenge, and to solve more sophisticated problems, an automatic way to learn these basis functions should be employed. On this ground, the "Multi-Layer Perceptron" (MLP) algorithm, also called "Neural Networks" (NN), is introduced. This approach is an extension of the previously analyzed linear models, and in particular, it consists of multiple neurons structured in several layers. Neural networks are nested functions performing linear or non-linear (depending on the activation function) operations to some input, that produce an output in the same way as logistic regression.

A simple multi-layer perceptron with one hidden layer is visualized in Figure 4. Firstly, we can see that the network input has the same dimensionality as the data sample ($\mathbf{x} = \{x_1, ..., x_D\}$, where $\mathbf{x} \in \mathbb{R}^D$ here is a data sample). Secondly, the hidden layer is represented by $\mathbf{h} = \{h_1, ..., h_M\} \in \mathbb{R}^M$, where M is the total amount of hidden neurons and is a hyper-parameter to be tuned during the training process. Finally, the output layer is represented by $\mathbf{y} = \{y_1, ..., y_K\}$ where, K is the number of classes.

A neuron, in a similar fashion to perceptron and logistic regression, combines linearly input $\phi(\mathbf{x})$ with a set of weights $\mathbf{w}$, that either amplify or downplay that input, thereby assigning significance to inputs with regard to the task the algorithm is aiming to learn.

The tuning of these weights occurs during the "training process". As depicted in Figure 4, in this example there is only one hidden layer, however, there could be more. The network in this figure contains K output neurons, a number that is suitable for K-class classification.

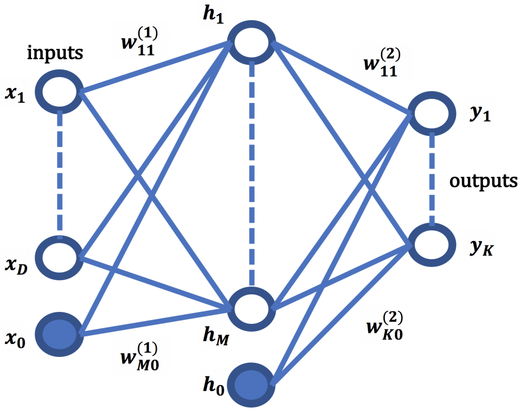


Figure 4: A multi-layer Perceptron or Neural Network topology. In the similar spirit as in case of perceptron, each neuron h_j (from the hidden layer) in the network contains a weight w_{ji} that connects its inputs (i represents the input layer.

The output of its network is calculated in the same manner as in the logistic regression algorithm and is based on the weight summation which is then passed to the activation function. The same process occurs for the hidden layer, where, the neuron can determine whether and to what extent the input signal should progress further through the network and cascade information to the next layer to affect the ultimate outcome. If the signals pass through, the neuron has been activated. The weight summation and activation operations for all neurons of the network (from all layers) is denoted as the **feed-forward pass**.

Moreover, we will provide a simple example with three layers which are the input, hidden, and output layers (as depicted in Figure 4). Firstly, for the input vector $\mathbf{x} \in \mathbb{R}^D$ we can construct M linear combinations of the input variables with the hidden layer:

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)} = \mathbf{w}_{(1)}^T \mathbf{x} \tag{14}$$

where a_j is the the output of a j-th neuron from the hidden layer and is called “activation”, with $j = 1, ..., M$. Finally, the superscript (1) denotes the weights of the hidden layer. Moreover, with $w_{j0}^{(1)}$ we define the bias of the neuron where the superscript defines the layer and we add in the data sample one extra dimension $x_0 = 1$ to be able to vectorize the expression (as in case of logistic regression). To calculate the final output of the neuron, we should apply an activation function $h(.)$ (for instance, it can be a sigmoid function) and we can have $z_j = h(a_j)$. This entity is defined as a “hidden unit”. For the output layer, we have:

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)} \tag{15}$$

where $k = \{1, ..., K\}$ is the index for the specific output class of the output layer. This layer is denoted by (2) superscript while its weights are denoted by the following $w_{kj}^{(2)} z_j$.

Again, $w_{k0}^{(2)}$ corresponds to the bias of the output layer. In a similar way as in hidden layer neurons, this activation also needs to be passed from an activation function like the sigmoid. The choice of this activation function depends on the nature of our data and can also be a hyperparameter that needs to be tuned during the training process.

Having a dataset of data samples, we need to pass them through the network and calculate the output of the hidden neurons. This output is compared with the initial annotation and an error score based on the predictive power of the output is calculated. This error score can be used to re-calibrate the weights of the whole network. The core idea is to always move the weights in the direction that will minimize the "error score". To do so, the "back-propagation" algorithm is proposed to re-update the weights. Back-propagation is based on the the gradient descent algorithm. However, before explaining this algorithm, an explanation of the basic error function is introduced.

Error function in Neural networks: The error function is a way to evaluate how well a neural network performs. Given a training sample, the error function compares the expected targeted output and the actual network output. The error function is a single value (scalar), not a vector, that denotes how well the neural network does as a whole. A simple version of this error function (denoted as squared error) can be framed as:

$$E(\mathbf{w}, w_0, \mathbf{x}, y) = \frac{1}{2}\sum_{i=1}^{m}\{y_{nk} - t_{nk}\}^2 \tag{16}$$

where y_{nk} is the target value and t_{nk} the predicted one, similarly to "logistic regression". However, this error has proven to be problematic with outliers.

From a more rigorous probabilistic point of view and in a similar way with the logistic regression algorithm, we can apply the binary loss function ("cross entropy") as:

$$E(\mathbf{w}, w_0) = -\sum_{n=1}^{n}\sum_{k=1}^{K}\{t_{nk}\ln(y_{nk}) + (1 - t_{nk})\ln(1 - y_{nk})\} \tag{17}$$

Or in case of K mutually exclusive classes we can write it as:

$$E(\mathbf{w}, w_0) = -\sum_{n=1}^{n}\sum_{k=1}^{K} t_{nk}\ln(y_{nk}) \tag{18}$$

Gradient Descent Algorithm Having designed our network and error function, a method to update the whole network weights should be also employed. Towards this end, we employ again Gradient Descent Algorithm (GDA).

This algorithm is associated with the way that NNs algorithm updates its weights each time the error function is calculated. This occurs after one pass of the whole dataset in one epoch (or possibly in a different way, for instance, after one data sample or a batch of samples). Having calculated the error function $E_{\mathbf{w},w_0}$, this value can be back-propagated and utilized to modify the values of NN's parameters $\{\mathbf{w}, w_0\}$ in such a way that the $E_{\mathbf{w},w_0}$ error will be minimized.

GDA is associated with calculating the gradient of the error function $\nabla E(\mathbf{w}, w_0)$ at a specific point and is a vector, tangential to the surface pointing in the direction where the function increases most rapidly. The reason for this lies in the specific design of the error function (otherwise it could have also been the point of rapid decrease). Conversely, the negative gradient $-\nabla E(\mathbf{w}, w_0)$ will point in the direction in which the function decreases most rapidly. By taking this into account, we can denote the gradient update of the weights as:

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} - \lambda \nabla E_{\mathbf{w}} \tag{19}$$

In the equation above, λ is a constant parameters that is denoted as "learning rate". This parameter reflects the importance that the algorithm gives to the gradient vector during the updating of the weights. If it is set too low, the weights will be adjusted slowly and the convergence to a local minimum will take a lot of time. On the contrary, if it is set too high, the algorithm might not be able to track the local minimum since the update of the weights is too big. In a more analytical form of Equation 19 and using denominator-layout notation, GDA can be framed as:

$$\nabla E_{\mathbf{w}} = \begin{bmatrix} \frac{\partial E_{\mathbf{w}}}{\partial w_1} \\ \frac{\partial E_{\mathbf{w}}}{\partial w_2} \\ \dots \\ \frac{\partial E_{\mathbf{w}}}{\partial w_D} \end{bmatrix} \tag{20}$$

and when combined with Equation 19 we can develop it as:

$$\begin{bmatrix} w_1^{\tau+1} \\ w_2^{\tau+1} \\ \dots \\ w_D^{\tau+1} \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_D \end{bmatrix} - \lambda \begin{bmatrix} \frac{\partial E_{\mathbf{w}^{\tau}}}{\partial w_1} \\ \frac{\partial E_{\mathbf{w}^{\tau}}}{\partial w_2} \\ \dots \\ \frac{\partial E_{\mathbf{w}^{\tau}}}{\partial w_D} \end{bmatrix} \tag{21}$$

The most compelling thing about the GDA algorithm is that it will try to adjust those weights that are in most need, while the weights in less need will change less (due to the partial derivatives calculation). This is closely connected to the fact that the negative gradient vector points exactly in the direction of maximum descent. This is also visualized in Figure 5 in a two-dimensional space.

So far the partial derivatives of the error function E are calculated over the variable $\mathbf{w}$, that represents the weights. For simplicity reasons, the variable w_0 that representing the bias term is omitted. However, in reality, the partial derivative of the error function over the bias variable needed to be calculated as well: $\frac{\partial E_w}{\partial w_0}$.

Back propagation With GDA, the way that the weights and bias variable needs to be modified to optimize the error function is shown. This modification is based on the partial derivative of the error function over the weights and bias. However, calculating these partial derivatives is not a straightforward task (due to the nested nature of the weights) and is related to implementation of the "chain rule", since we need to calculate nested derivations.

To be more precise, the total error after adopting the independent and identically distributed assumption (i.i.d) can be formulated as:

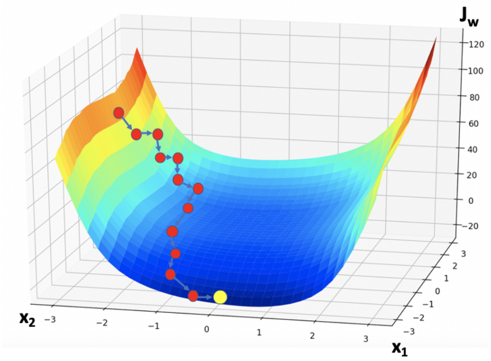


Figure 5: A visualization of the gradient procedure in a 3d space.

$$E(\mathbf{w}) = \sum_{n=1}^{N} E_n(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} (y_{nk} - t_{nk})^2 \tag{22}$$

where the error function from Equation 16 is applied. The goal is to find the optimal weights $\mathbf{w}^*$ (by including also the bias term) such that if we perform a feed pass of the input dataset the error function value will be minimum. We can formulate this mathematically in the following way:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} E(\mathbf{w}) \tag{23}$$

In general, for the feed-forward Neural Network, each output unit computes a weighted sum of the following form:

$$a_j = \sum_{i=0}^{M} w_{ji} z_i \tag{24}$$

with $z_j = h(a_j)$ being the activation of this specific neuron. In our simple example with one input, one hidden, and one output layer $z_i = x_i$ (for the input layer). Therefore, for the neuron $z_j = h(a_j)$ can be either one in the output layer or one of hidden layer (in this case: $a_j = \sum_{i=0}^{M} w_{ji} x_i$).

Having calculated the feed-forward step, we need to calculate the gradient of the error function over these weights. The whole process of calculating the gradient of the error function over w_{ji} for sample n (where j is the j-th neuron in the hidden layer) can be deconstructed using the chain rule:

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}} \tag{25}$$

where it can be easily shown that $\frac{\partial a_j}{\partial w_{ji}} = z_i$. When the activation function is the identity function $z_i = x_i$. Moreover, we can disentangle the derivations if we define $\delta_j = \frac{\partial E_n}{\partial a_j}$

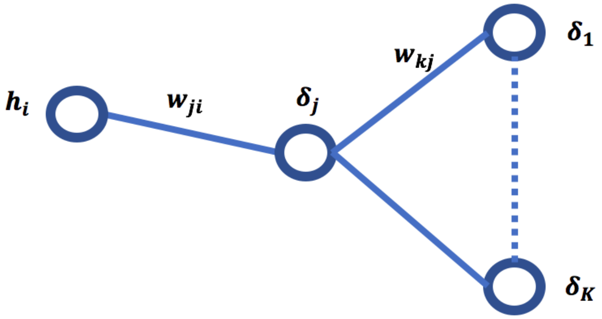


Figure 6: A visualization of the gradient procedure in a 2d space. Being in a current position the goal is to modify the weights w in such as way that the algorithm will reach the minimum error function $E_{\mathbf{w}}$.

and therefore, the result of the whole derivation is $\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i$. Hence, by calculating δ_j we can calculate the desired derivative. For the output layer we have:

$$\delta_k = \frac{\partial E_n}{\partial a_k} = y_k - t_k \tag{26}$$

since the neuron is connected to all output neurons as is depicted in Figure 6, we derive:

$$\delta_j = \sum_{k=1}^{K} \frac{\partial E_n}{\partial a_k} \frac{\partial a_k}{\partial a_j} = h^{'}(a_j) \sum_k w_{kj} \delta_k \tag{27}$$

where $\frac{\partial E_n}{\partial a_k} = \delta_k = y_k - t_k$ and $\frac{\partial a_k}{\partial a_j} = w_{kj} h^{'}(a_j)$. Here we should also develop the equation for y_k:

$$y_k = h(\sum_{k=0}^{K} w_{ik} z_j) \tag{28}$$

For an unseen sample $x^{'}$, we can use Equation 28 to calculate its output layer activation for each class. Then, we can assign the test sample in the class with the highest activation.

3. DEEP LEARNING

In a nutshell, deep learning networks are differentiated from single-hidden-layer neural networks by the number of node layers that they consist of. The basic versions of NNs such as the one explained in the previous section are the so-called shallow networks, composed of one input and one output layer, and at most one hidden layer in between. Networks with more than three layers (including input and output) qualify as deep learning. In this section, we will delve into deep learning architectures. For instance, in the following sub-sections we will analyze Convolutional Neural Networks (CNNS), Recurrent Neural Networks (RNNs), and some useful features of deep learning.

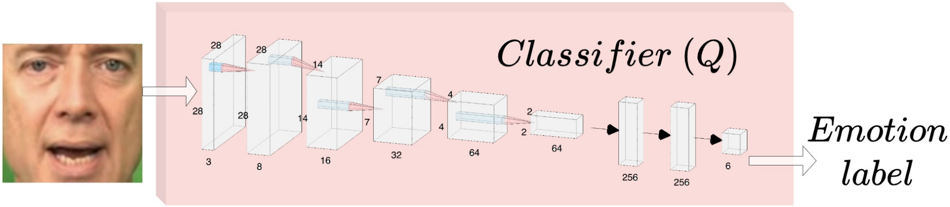


Figure 7: A diagram of Convolutional Neural Network (CNNs) classifying basic emotion from an input face. In this figure we can see the tensor sizes in each step of the operation that are implemented in Chapters 3, 4 and 5. In between each layer we apply max pooling, batch normalization and dropout.

3.1. CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (“ConvNets“ or CNNs) [15] [16] are a special category of “sparse” neural networks that has been proven to be very effective especially for computer vision applications. These networks are considered to be sparse since in comparison with NNs and fully connected layers, only some of the connections are activated due to the convolution operation.

CNNs have been successfully implemented to perform face recognition, object recognition, and in general in classification applications related to images, video or graphics [14] [17] [18].

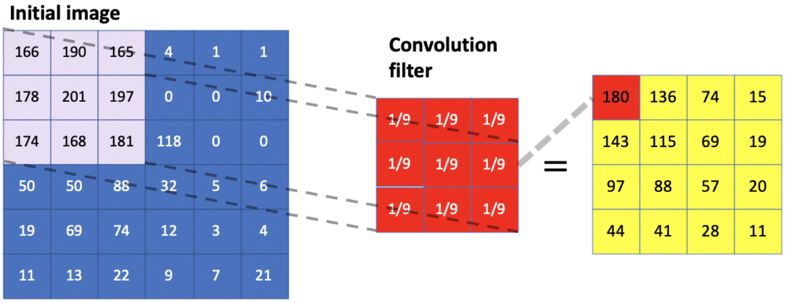
The CNN in Figure 7, is based on the LeNet [19] architecture and classifies a face input image into six categories of emotion: anger, disgust, fear, happy, neutral, and sad. The figure visualises how an input image can be transformed through the CNN operations and lead to the output layer. The operations involved will be analysed into further details in the following paragraphs. In the visualised example of the figure, when the system receives an angry face image as input, the network correctly assigns the highest probability for anger (0.94) and the remaining 0.06 is distributed across all five remaining categories. The sum of all probabilities in the output layer should be one.

In more detail, regarding the feed-forward pass, the operations that are applied to the input tensor (in the previous example represented by an image) are the *Convolution operation*, the **activation function**, the *pooling operation* and finally, the *fully connected layer* or as called alternatively fully connected layer. In more details:

- **Convolution operation**: During this step, the convolution operation is performed to the input image using a set of filters (whose values are to be learned during the training). The basic idea of the CNNs algorithm is that a set of convolution filters are applied to the image with unknown trainable parameters. In this way, the network tunes the parameters of these filters, which lead to learn robust features from the image after the convolution operation.

- **Pooling or Sub Sampling**: This operation is applied to decrease the dimensionality of the tensors, and, as a result, to reduce the computational complexity of the whole training approach.

- **Classification (Fully Connected Layer)**: After performing multiple convolution and pooling layers, in the end a fully connected layer (similar to a swallow neural



166	190	165	4	1	1
178	201	197	0	0	10
174	168	181	118	0	0
50	50	88	32	5	6
19	69	74	12	3	4
11	13	22	9	7	21

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

180	136	74	15
143	115	69	19
97	88	57	20
44	41	28	11

Figure 8: An example of how convolution operates in 2d image.

network) is applied to easily connect the result of the whole network to an output layer.

These operations are the basic building blocks for the CNNs. Finally, the training procedure is performed by using the standard backprogragation algorithm and tuning the weights of the whole network, while the error function depends of the application that the algorithm needs to perform.

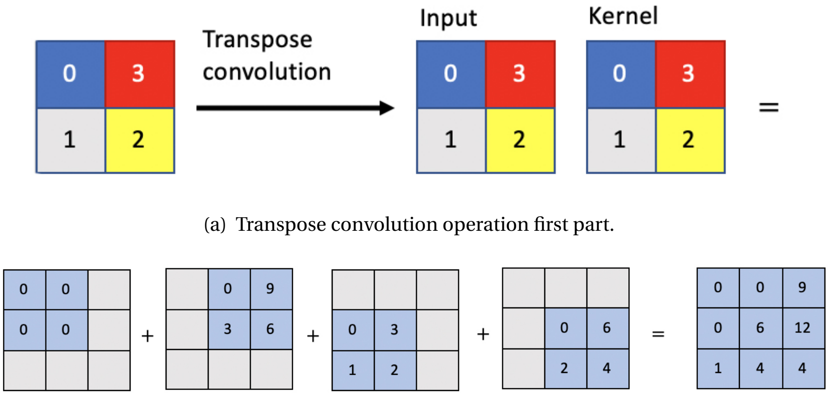
CONVOLUTION OPERATION

The basic operation that gives CNNs their name is the "convolution" operation. The purpose of convolution is to extract features from the input image. To do so, there are several convolutional layers in the network. Each of these layers consists of several filters that are used to perform the convolution operation on the inputs of the layer. These filters, in a similar way to the dense layer in the multi-layer perceptron, contain weights that are randomly initialized. These filters are applied to the input data (by performing the convolution operation as shown in Figure 8). During the training procedure, their values are tuned by the back-propagation process. By learning the weights that represent the values of the convolution operation, the network preserves the spatial relationship between pixels by learning image features using small squares of input data.

To give a better intuition about the convolution process, Figure 8 displays the way that the convolution process operates for a 2d image. In the specific example, we make use of a static filter that is called "average filter". In the case of CNNs, the values of the filter are tuned during the training process and are fixed values.

Transposed convolution layer: In several occasions, it is very useful to perform, the converse convolution operation called "transposed convolution" or otherwise "deconvolution". This is normally the case (when using GANs or auto-encoder inspired architectures) when it is desired to map an input (for instance an input image) to a target image.

The basic explanation of converse convolution can be as follows: let us assume an input image and a convolution filter and the output after the application of the convolution operation. The intuitive idea behind the converse convolution is the following: after



(a) Transpose convolution operation first part.

(b) Transpose convolution operation second part.

Figure 9: Transpose convolution operation.

applying the operator to the input image the output image emerges. If we then apply a normal convolution to this output image it should lead to the initial input image.

The basic principle of the transposed convolution layer can be summarized using the example of Figure 9. In this Figure, we have a 2 × 2 input, as well as a filter with the same pixel values. The result of the operation can be seen in the end of this figure. While this operation is also called deconvolution, it does not exactly perform the converse operation of the convolution itself, but it is an up-sampling operation that simulates the operation of the convolution itself.

ACTIVATION FUNCTION

A basic operation that provides non-linearity in neural networks and in deep learning is the application of an "activation function". This function is applied to the activation of each artificial neuron. Some typical implemented activation functions are:

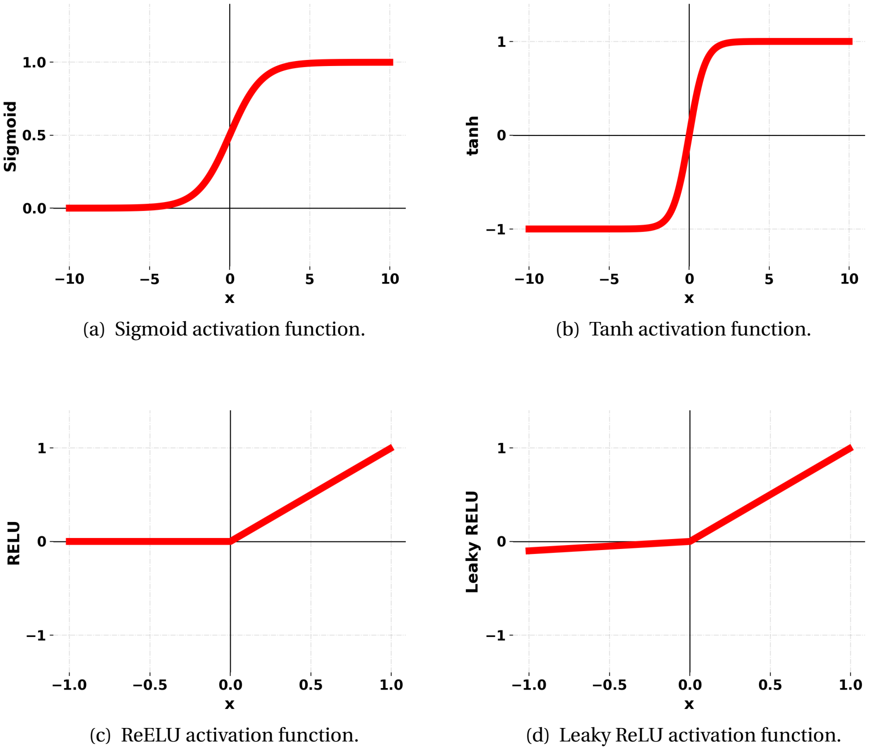
Sigmoid:

The sigmoid function is a mathematical formula that is shaped to be like an S-shape function. It can be framed using the following formula:

$$h = \frac{1}{1 + \exp^{-x}} = \frac{\exp^{x}}{1 + \exp^{x}} \tag{29}$$

The main reason that the sigmoid function is employed is because it transforms all input values so that they are between zero and one. Hence, it is useful for models that needs to output probabilities. Since the probabilities are valid only in the range of 0 to 1, sigmoid is a logical option. Another significant feature is that the function is differentiable. That is to say that the slope of the sigmoid curve can be calculated at any two points. This activation function is visualized in Figure .10(a).

Hyperbolic tangent:



(a) Sigmoid activation function.

(b) Tanh activation function.

(c) ReELU activation function.

(d) Leaky ReLU activation function.

Figure 10: Activation functions.

$$h = \frac{\exp^{x} - \exp^{-x}}{\exp^{x} + \exp^{-x}} \tag{30}$$

The hyperbolic tangent or tanh framed by Equation 3.1 is also a simgoid style function with an S-shape. The fundamental difference to the simple sigmoid is framed in Figure 10. It actually projects the negative inputs to be mapped as strongly negative and the zero inputs will be mapped near zero in the tanh graph.

ReLU: Rectified Linear Unit is another type of activation function. Mathematically, it is framed in Equation 31. Visually, it is displayed in Figure .10(c):

$$h = \begin{cases} max(0,x) & \text{if } x \geq 0 \\ \text{undefined} & \text{if } x = 0 \\ 0 & \text{if } x < 0 \end{cases}. \tag{31}$$

The ReLU is widely applied as an activation function in deep learning state-of-the-art approaches. In particular, it has been used in almost all convolutional neural networks or deep learning approaches. One notorious drawback with this activation function is that the negative values turn into zero steeply. This characteristic affects the training process of the network and the ability of the model to properly fit the training datasets.

Leaky ReLU: In an attempt to solve the dying ReLU problem a new modified version of the ReLU activation is proposed, the so-called Leaky ReLU. This activation function is displayed in Figure .10(d) and framed in Equation 32.

$$h = \begin{cases} max(0,x) & \text{if } x \geq 0 \\ \text{undefined} & \text{if } x = 0 \\ \alpha x & x < 0 \end{cases}. \tag{32}$$

MAX POOLING OPERATION

Spatial Pooling (also called sub-sampling or down-sampling) is an essential operation that is part of the CNNs architectures, that aims at reducing the dimensionality of the feature maps by keeping only essential information. Spatial Pooling has different mode types: Max, Average, Sum, and so forth. In the case of Max Pooling, we define a spatial neighborhood (for example, a 2×2 window), and for this window, we keep the largest element that results from the convolution filter after the application of the activation function. Instead of taking the largest element, we could also take the average (Average Pooling) or the sum of all elements in that window. Empirically, Max Pooling has been shown to work better. In Figure 11, an example of how this operation functions is depicted. It is important to state here that there are no trainable weights for this operation in the network. Furthermore, for each convolution layer, normally there is one max-pooling layer as well.

Up-sampling operation This operation (in a similar spirit to transpose convolution) is proposed for performing the reverse task regarding the max polling operation. This

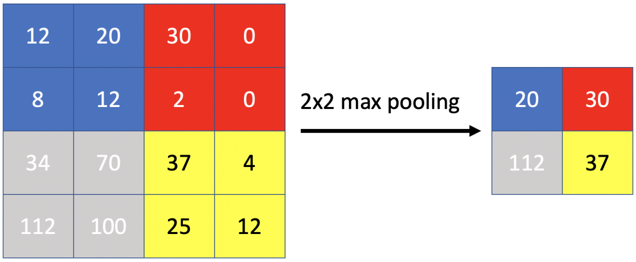


Figure 11: The max pooling process example for a 2d input image matrix.

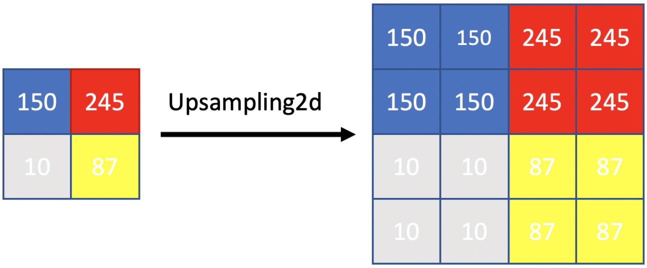


Figure 12: The up-sampling operation example for a 2d input image matrix.

operation simply takes as input one image of dimensions $k \times k$ and quadruples its size by replicating each pixel 4 times (as a 2×2 region).

BATCH NORMALIZATION

Batch normalization [20] is the idea of normalizing the intermediate layers of the CNNs during the training process. Normally, in machine learning, the input layer is normalized by scaling the input feature values. For example, when an input variable ranges from 0 to 1 while another ranges from 1 to 1000, then a standard normalization process occurs to speed up the learning. Since it has been proven that this process for the input layer has beneficial effects, the same can be applied to the values in the hidden layers, which are constantly changing.

To accelerate the convergence of a neural network, batch normalization "normalizes" the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation. However, after this shift of the activation outputs, the weights in the next layer are no longer optimal. Consequently, batch normalization adds two trainable parameters to each layer, so that the process will be smoother. The whole approach is summarized in Algorithm 3.

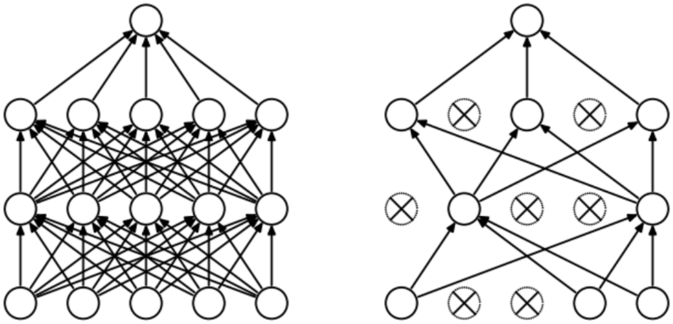


Figure 13: Dropout operation for the dense connected layer in a specific batch.

Algorithm 3 Batch normalization approach implemented in the output of the activation function x .

```
Input: Values of $\mathbf{x}$ over a mini batch $\mathscr{B} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m\}$ and parameters to be learned $\gamma$ and $\beta$.
Output: The normalized output $\hat{x}_i = \phi_{(\gamma,\beta)}(\mathbf{x}_i)$ with $i = 1, ..., m$ and $\phi$ the batch normalization function.
$\mu_B = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i$
$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{x}_i - \mu_B)^2$
$\tilde{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$
$\hat{x}_i = \gamma\tilde{x}_i + \beta$
```

Dropout Operation

Dropout operation [21] regards ignoring several neurons (which are chosen randomly) in NNs by using skip connections [4] during the training phase of the network. This approach can be applied to Neural Networks and in general in deep learning approaches. This means, that during the forward and backward training pass, some neurons of the weights are neutralized. More technically, at each training stage, individual nodes are either randomly dropped out or kept. That decision is based on a Bernoulli distribution with probability p to keep the node and probability $1 - p$ to keep drop the node.

The reason that this operation is applied is to avoid over-fitting. During training, there is a chance that the network collapses and only some of the neurons of the network really fire up. In these situations, no matter what is the input, there are neurons that are never influenced by it and lead to a problematic output. On this ground, by neutralizing different parts of the network during each epoch, we are trying to uniformly enforce that all neurons learn some meaningful information. In Figure 13, the way that this approach operates in the case of dense connected layers is displayed. In the case of the CNNs, the "dropout" operation operates in a similar fashion. However, in this case, it occurs in the filters of a CNNs layer.

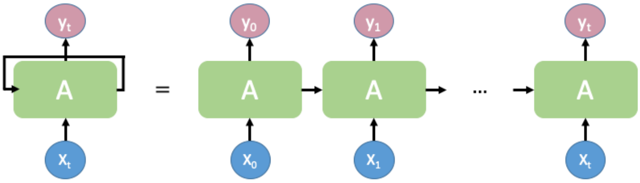


Figure 14: A diagram of Recurrent Neural Network (RNNs).

3.2. Recurrent Neural Networks

All the previous ML algorithms can be categorized as non-sequential-based networks since they always process a single input (for example, a single image frame) independently of other frames. However, in several cases, when we are trying to model data that embed time information, a modified strategy needs to be applied. Hence, recurrent neural networks (RNNs) [23] [24] are proposed as the natural extension of simple NNs for sequential-related problems to time signals (video classification, stock market prediction, machine translation, text generation, and so forth [23][24]). To explain the way that RNNs operate, we will proceed with the following example: Let us assume that a viewer is watching an animation movie, and a new scene from this animation is about to start. Having already watched several scenes of the animation, the viewer can already rationalize about the story of the animation and start connecting the dots of the whole movie. Hence, the current scene will be interpreted by human perception by taking into account the existing information.

However, when it comes to the analysis of the animation by ML algorithms such as CNNs, this is not the case. The reason is that the CNNs are always processing one state of the problem at a time (by processing just one frame), and therefore, the information from previous frames is lost. One possible way to deal with this, is to apply the network in a self-loop by giving as input frames from different time sequences and by keeping a memory information from previous frames.

On this ground, RNNs are introduced to mitigate the insufficiency of CNNs to handle temporal information. Since now the input is data over time, we can denote the input data (frames in our example) as $\mathbf{x}_i$ where index i stands for the time order in the sequence of the frames. An RNN network takes as input not only the input frames $\mathbf{x}_i$ but also the network's output from the previous instance $\mathbf{h}_{i-1}$ (where $\mathbf{h}$ represents the hidden state of the NNs or CNNs). Hence, if the output of the network at $t = 1$ is $\mathbf{h}_0$ and the input $\mathbf{x}_0$, the training for the network at $t = 2$ will also consider $\mathbf{h}_0$ (the output received from the previous instance of time). If we unroll the network, we will get the structure visualized in Figure 14.

It is important to note that the sequential units A (depicted in Figure 14) are the same over time. This unit A actually can be represented by a dense Neural Network, a CNN, or any other architecture. Again, for the error function, it is chosen depending on the nature of the input datasets and the task that we want to perform. Finally, the feed-forward and back-propagation are occurring in the same fashion as in the case of NNs.

Unfortunately, RNNs notoriously face a severe issue regarding the way that back-propagation operates over time. This could notoriously lead to “vanishing gradient” or

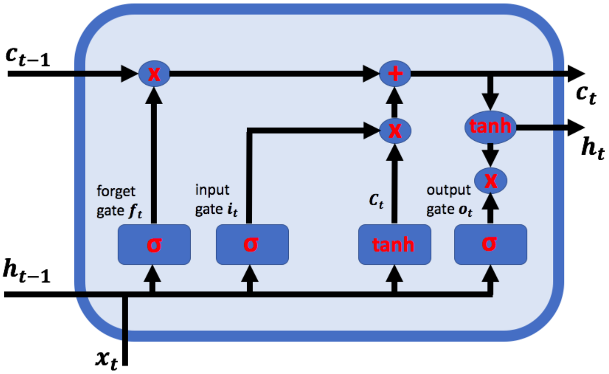


Figure 15: Long short-term memory (LSTM) cell or unit.

"exploding gradient" problems [4] and to a de-stabilized training process due to the unrolled performance of the chain rule over time. In simple words, RNNs can not keep in memory important information (from early previous time sequences) that may provide important insights to later timestamps. To overcome these problems, Long Short Term Memory (LSTM) is proposed by Hochreiter and Schmidhuber in [25].

LONG SHORT-TERM MEMORY

LSTM is a special type of RNNs that can learn long-term dependencies that do not suffer from the "vanishing or exploding gradient" to the same extent as in the case of RNNs. The LSTM unit has four different neural network layers which interact differently with the input data, unlike RNNs that only have a single neural network layer.

In particular, the LSTM unit consists of the **cell state** (current information flow of the unit) and three Gates - **forget gate**, **input gate**, and **output gate**. These operations can be seen in Figure 15. In more detail:

- **Cell State**: The cell state plays the role of the continuous flow of information over various instances of time. At each instance of time it needs to be decided how much information the cell state will maintain or modify. The reason why LSTMs are introduced in the first place, is to retain the importance of the information that comes from a particular sequence in time that is time-distant from the current sequence (in contrast with RNNs that take into account only the immediately preceding frame while the information from older sequences vanishes). Naturally, in the first time step there is not any cell state $\mathbf{c}_{t-1}$ from the previous step to be given as input.

- **Gate**: The LSTM has the ability to let information through the network by either adding or removing information from previous states by using the "gates".

[4] shorturl.at/efzBL

- **Forget gate**: In order to decide which information the LSTM will pay more attention to (at each stage) the **forget gate** is introduced. Firstly, let us take a closer look at the various notations we have:

  - $\mathbf{c}_{t-1}$ is the previous cell state, while c_t the current cell state.
  - $\mathbf{h}_{t-1}$ is the output from the previous state, while h_t is the output of the current state.

  The forget gate regulates the amount of information from the previous cell state that will be used and how much will be discarded. The output from the last state h_{t-1} is concatenated with x_t and passed through a sigmoid function. This operation can be formalized as: $f_t = \sigma(\mathbf{w}_f[\mathbf{h_{t_1}}, \mathbf{x_t}] + w_{0f})$. Sigmoid activation provides an output between 0-1. Intuitively, 0 can be translated as a "complete" forget while one means that everything should be kept in memory.

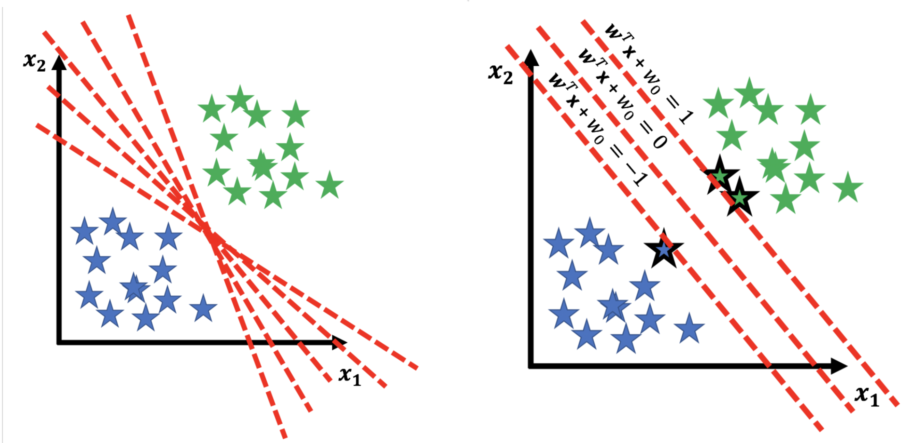
- **Input gate**: The input gate regulates which new information will be added to the cell state. The concatenated x_t and h_{t-1} is sent over a sigmoid unit that decides what value we will be updated. This value is also processed by a hyperbolic tangent layer which normalizes its value between -1 to $+1$. Then, we multiply the hyperbolic tangent output with the sigmoid output and add it to the cell state. After performing all these operations the output of the cell state can be extracted. This operation can be formalized as: $i_t = \sigma(\mathbf{w}_i[\mathbf{h_{t_1}}, \mathbf{x_t}] + w_{0i})$.

- **Output gate**: The output gate decides which information will be passed to the next instance of time. Firstly, the input x_t and h_{t-1} are sent over a sigmoid unit. Then, the current cell state is processed by the hyperbolic tangent function (point wise). This is formalized as: $\tilde{C}_t = \tanh(\mathbf{w}_c[\mathbf{h_{t_1}}, \mathbf{x_t}] + w_{0c})$. Both of these outputs are multiplied and sent over to the next instance of time. We can frame this as: $C_t = f_t C_{t-1} + i_t \tilde{C}_t$.

It is important to state here that the weights of LSTM are shared among the different timestamps. In this way, previous information from previous states still remains in memory. Finally, by using this RNNs architecture, we are able not only to take into account the temporal information from previous frames, but to also alleviate the vanishing gradient problem that is introduced by RNNs.

4. SUPPORT VECTOR MACHINES

Support Vector Machines (SVM) is a popular supervised classification and regression ML algorithm. It lies in the non-probabilistic and non-deep learning approaches. One important feature that makes SVM a popular algorithm is that it can be framed as a convex optimization problem, which means that we can have only one optimal solution. Another characteristic that makes SVM a good classification option is that it is not sensitive to outliers. In this section, we will analyze its basic form that is related to binary classification tasks. It can be easily expanded for multi-class classification and regressions tasks as well.



(a) The initial two datasets.

(b) Support vectors and the calculated hyperplane.

Figure 16: The objective of the SVM algorithm when trying to classify two different data points in two different classes.

The goal of SVM is to define a hyperplane in a multidimensional space that separates two different classes from a given dataset. The algorithm is considered to be a maximum margin classifier since its task is to find a maximum marginal hyperplane that best divides the dataset into classes. To calculate the aforesaid hyperplane, its task is to determine the correspondent training data points which can specify a hyperplane that can divide the two classes in the most optimal way by maximizing the margin between the two classes. These data points that defined the hyperplane are called **support vectors**.

Firstly, we will denote out dataset as $\{\mathbf{x}_n, t_n\}$ with $\mathbf{x}_n \in \mathbb{R}^D$ to be a D-dimensional vector, $n = \{1, ..., N\}$ to be the indexes of the data points and $t_n \in \{-1, 1\}$ the class information associated with the data samples. SVM in its core is a linear and binary classifier, and thus, we can use the following model:

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + w_0 \tag{33}$$

where ϕ denotes a fixed feature transformation (coined alternatively as basis functions, which is a possible transformation from the initial pixel space to a targeted space) and $\mathbf{w}$ and w_0 denote the associated linear parameters of the SVM algorithm. The decision of the network is determined by the sign of $y(\mathbf{x})$. SVM employs the term "margin" which is coined as *the perpendicular distance from the decision boundary to the closest data points* [4], as illustrated in Figure .16(b).

The distance from any of the points in this figure, to the boundary decision defined as $y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + w_0 = 0$ is given by:

$$d = \frac{|y(\mathbf{x})|}{||w||} \tag{34}$$

That is shown before in Figure 3. Since we are interested only in the decision boundaries where all points are correctly classified, therefore, $t_n y(\mathbf{x}_n) > 0$ that means $y(\mathbf{x}_n) \geq 0$ if $t_n = 1$ or $y(\mathbf{x}_n) < 0$ if $t_n = -1$. Moreover, we can re-write the previous equation as:

$$\frac{t_n y(\mathbf{x}_n)}{||w||} = \frac{t_n(\mathbf{w}^T \phi(\mathbf{x_n}) + w_0)}{||w||} \tag{35}$$

The definition of the term **margin** in SVM (that needs to be maximized) can be as follows: the perpendicular distance to the closest point x_n from the data set. Moreover, as we mentioned before, we wish to optimize the parameters $\mathbf{w}$ and w_0 to maximize this distance. Thus, the maximum margin solution is found by solving:

$$\underset{\mathbf{w}, w_0}{\arg\max} \left\{ \frac{1}{||\mathbf{w}||} \min_n [t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + w_0)] \right\} \tag{36}$$

From Equation 36, it follows that there are several values of weights that satisfy this optimization. If we replace $\mathbf{w}$ with $\kappa\mathbf{w}$ it is obvious that we can lead to the same optimization problem. However, to make the problem tractable, we can assume that the following is true:

$$t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + w_0) = 1 \tag{37}$$

In this case, all data points $\mathbf{x_n}$ need to satisfy the following constraint:

$$t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + w_0) \geq 1, \text{ with } n = \{1, ..., N\} \tag{38}$$

Maximizing Equation 36 is equivalent to minimizing the following expression:

$$\underset{\mathbf{w}, w_0}{\arg\min} \frac{1}{2} ||\mathbf{w}||^2 \tag{39}$$

where the utility of adding the fraction lies in its convenience for the numerical derivations, while the square was added to guarantee the final optimization scheme to be convex. Bare in mind that the minimization of Equation 39 is subjected still to the constraint of Equation 37, therefore the whole optimization, ends up being a constrained "quadratic optimization". To solve this constraint optimization problem, Lagrangian multipliers need to be introduced and to form the following equivalent Lagrangian function:

$$L(\mathbf{w}, w_0, \mathbf{a}) = \frac{1}{2} ||w||^2 - \sum_{n=1}^{N} a_n \{t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + w_0) - 1\} \tag{40}$$

For each of the data points $\mathbf{x}_i$ corresponds one Lagrangian multiplier a_i. It is important to note here, that the minus sign of the Lagrangian function is due to the fact that we want to perform minimization and not maximization. Then, the stationary points of the Lagrangian can be calculated. Deriving over $\mathbf{w}$ will lead to $\mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n)$ and deriving over w_0 will lead to $0 = \sum_{n=1}^{N} a_n t_n$. The previous optimization is subject also to Karush-Kuhn-Tucker constraints (KKT):

$$a_n \geq 0 \tag{41}$$

$$t_n y(\mathbf{x}_n) - 1 \geq 0 \tag{42}$$

$$a_n(t_n y(\mathbf{x}_n) - 1) = 0 \tag{43}$$

with the first constraint being the dual feasibility the second one being the primal feasibility, and the last one to be the complementary slackness. Having calculated the stationary points, we can eliminate $\mathbf{w}$ and w_0 from the Lagrangian, which leads to the dual Lagrangian representation:

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m) \tag{44}$$

which is subject to Karush-Kuhn-Tucker constraints (KKT): $a_n \geq 0$ and $\sum_{n=1}^{N} a_n t_n = 0$.

The main reason for the dual representation is that we can easily replace the expression: $k(x, x^{'}) = \phi(\mathbf{x})^T \phi(\mathbf{x})$ with a kernel and take advantage of the kernel trick and the powerful expressivity that comes along with it. The prediction function for a new data sample can be calculated using the following formula:

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_b k(\mathbf{x}, \mathbf{x}_n) \tag{45}$$

As is obvious from Equation 45, if the Lagrangian multiplier for a point is equal to $a_n = 0$ then it will not influence the calculation of the prediction decision so it will be considered inactive, while the rest of the samples for which $a_n \geq 0$ will be considered the active samples, or otherwise called **support vectors**. From the complementary slackness we know that if $a_n \geq 0$ then we have $t_n y(\mathbf{x}_n) = 1$ and thus, the support vector points lie on the maximum margin hyperplane. Hence, in the end, we can discard most of the points and we are only interested in a set of points that lie in the maximum margin hyperplane. On this ground, SVM can be considered as a Sparse Machine kernel method.

As aforesaid, we employed in SVM basis functions to map the initial input space (which is not linearly separable) into a feature space in which the same classification can be linearly separable and proceeded to calculate the decision boundary. Furthermore, we explained how in the dual Lagrangian representation the basis function can be replaced by kernels and in order to reduce the calculation complexity.

However, in this introduced feature space (in which we mapped our problem using kernels), still, we cannot always guarantee that our problem will be linearly separable. After applying the kernels though, the resulting decision boundary will give exact separation of the training data even if the correspondent decision boundary is not linear. On this ground, the non-linear separable problem will have class-distributions that have significant overlap and therefore, the decision boundary will lead to problematic generalization behaviours.

To cope with this issue, Vapnik and Cortes in [26] employ the slack variables $\xi_i \geq 0$, where each one of these variables corresponds to each of the data points $\mathbf{x}_i$. For $\xi_i = 0$ then the correspondent data point $\mathbf{x}_i$ lies in the correct margin boundary and when $\xi_i = 1$ then the data point lies exactly in the decision boundary and when $\xi_i > 1$ the data point is misclassified. After introducing these slack variables, we can modify the constraints of the SVM optimization correspondingly:

$$t_n y(\mathbf{x}_n) \geq 1 - \xi_n, \text{ with } i = \{1, ..., n\} \tag{46}$$

In this way, the algorithm relaxes the hard margin constraint and gives a soft margin which allows some misclassified training data samples. We can pose our soft minimization now as:

$$C \sum_{n=1}^{N} \xi_n + \frac{1}{2} ||\mathbf{w}||^2 \tag{47}$$

where the relaxation parameter C accounts for trade-off between the minimization of the margin and the slack variables. Setting C to a value close to zero gives a lot of emphasis to slack variables, while setting $C >> 0$ poses again our initial SVM optimization. Having introduced the slack variables, we can reformulate our primal and dual Lagrangian forms again. In this case, the primal Lagrangian can be framed as:

$$L(\mathbf{w}, w_0, \xi\mathbf{a}, \mu) = \frac{1}{2} ||w||^2 + C \sum_{n=1}^{N} \xi_n - \sum_{n=1}^{N} a_n \{t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + w_0) - 1 + \xi_n\} - \sum_{n=1}^{N} \mu_n \xi_n \tag{48}$$

with $\mu_n >> 0$ to be Lagrangian multipliers, in order to take into account the new constraint that is $\xi_n >> 0$. That is reflected in the last part of the Equation 48. We can re-write the KKT conditions as:

$$a_n \geq 0 \tag{49}$$

$$t_n y(\mathbf{x}_n) - 1 + \xi_n \geq 0 \tag{50}$$

$$a_n (t_n y(\mathbf{x}_n) - 1 + \xi_n) = 0 \tag{51}$$

$$\xi_n \geq 0 \tag{52}$$

$$\mu_n \geq 0 \tag{53}$$

$$\mu_n \xi_n = 0 \tag{54}$$

And finally, by calculating the stationary points we can write the dual form as:

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m) \tag{55}$$

REFERENCES

[1] M. Deisenroth, A.Faisal, and C.Ong, Mathematics for Machine Learning, Cambridge University Press, 2020.

[2] K.P.Murphy, Machine Learning: a Probabilistic Perspective, MIT Press, 2012

[3] C.E.Rasmussen and C.K.I.Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006.

[4] C.M.Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), New York, NY, Springer, 2006.

[5] A.Courville, I.Goodfellow, and Y.Bengio, Deep Learning, MIT Press, 2016.

[6] Charu C. Aggarwal, Neural Networks and Deep Learning, Springer, 2018.

[7] C.M.Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), New York, NY, Springer, 2006.

[8] K.P.Murphy, Machine Learning: A Probabilistic Perspective, Adaptive Computation and Machine Learning series 1st Edition, 2012.

[9] Y.LeCun, L.Bottou, Y.Bengio, and P.Haffner, Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86, 2278–2324, 1998.

[10] W.S.McCulloch and W.Pitts, A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biophysics, Volume 5, Number 4, Pages: 115–133, 1943.

[11] F.Rosenblatt, The Perceptron: A Perceiving and Recognizing Automaton (Project PARA), 1957.

[12] D.G.Kleinbaum, Logistic regression : a self-learning text, New York : Springer, ebook: 10.1007/978-1-4757-4108-7, 1994.

[13] D.E.Rumelhart, G.Hinton and R.J.Williams, Learning representations by back-propagating errors, Nature, Volume 323 Number 6088, pages 533–536, 1986.

[14] Y.LeCun, Y.Bengio and G.Hinton, Geoffrey, Deep learning, Nature, Volume 521, Number 7553, pages 436–444, 2016.

[15] J.Schmidhuber, Deep Learning in Neural Networks: An Overview, Technical Report, 2014.

[16] A.Krizhevsky, I.Sutskever and G.Hinton, ImageNet Classification with Deep Convolutional Neural Networks, Advances in Neural Information Processing Systems 25 (NIPS), 2012.

[17] K.Simonyan and A.Zisserman, Very deep convolution networks for large-scale image recognition, International Conference on Learning Representations (ICLR), 2015.

[18] C.Szegedy, W.Liu, Y.Jia, P.Sermanet, S.Reed, D.Auguelov, D.Erhan, V.Vanhoucke and A.Rabinovich, Going Deeper with Convolutions, Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[19] Y.Lecun, L.Bottou, Y.Bengio and P.Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE , Volume 86, Number 11, Pages: 2278–2324, 1998.

[20] S.Ioffe and C.Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, arXiv 2015.

[21] N.Srivastava, G.Hinton, A.Krizhevsky, I.Sutskever and R.Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research, Volume 15, Pages: 1929–1958, 2014.

[22] N.Srivastava, G.Hinton, A.Krizhevsky, I.Sutskever and R.Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research, Volume 15, Pages: 1929–1958, 2014.

[23] S.Dupond, A thorough review on the current advance of neural network structures, Annual Reviews in Control, Volume 14, Pages: 200—230, 2019.

[24] A.Sherstinsky, Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network, Physica D: Nonlinear Phenomena, Volume 404, 2020

[25] S.Hochreiter, J.Schmidhuber, Long Short-Term Memory, Neural Computation, Volume 9, Number 8, Pages: 1735-–1780, 1997.

[26] C.Cortes and V.N.Vapnik, Support-vector networks, Machine Learning, 20, 1995

CURRICULUM VITÆ

Christos Athanasiadis was born in February 1988 in Drama, Greece. In 2010 he completed his bachelor degree in the department of computer science in Aristotle University in Greece, where he also earned his master degree as well in the *Digital Media, Computational Intelligence* program. In his master thesis, he studied machine learning methods for developing intelligent agents in strategy video game *StarCraft*. In particular, the conducted research was about the decision making process of the agent, and in particular, the employment of neural networks (trained from multiple hours of gaming of real or artificial agents) for controlling important aspects of the *bot*. Early academic work experiences includes working as a research assistant in *SThe Centre for Research and Technology, Hellas CERTH* in the *Informatics and Telematics Institute ITI* from November 2013 to March 2016 at Visual Computing Lab (VCL). Part of my role was to conduct research, develop software and write deliverables for Horizon2020 European projects.

In March 2016, Athanasiadis started his Ph.D. in the department of data science and knowledge engineering (DKE) at Maastricht University under the supervision of Associate Professor Stelios (Stylianos) Asteriadis. He has been working in the Affective and Visual Computing Lab (AVCL), which was previously embedded in the Robots, Agents and Interaction (RAI) group. His research interests include computer vision, machine learning, Explainable AI (XAI), emotion recognition and human computer interaction. During the course of his Ph.D., he carried out research on cross-modal emotion recognition based on domain adaptation. His work utilized methods to bridge expressivity features from the audio and visual modalities and ways to elicit both in order to improve emotion recognition performance. Athanasiadis's Ph.D. research was supported by the Horizon 2020 project Managing Affective-learning THrough Intelligent atoms and Smart InteractionS (MaTHiSiS). Besides his research and project work, he was a teaching assistant for multiple courses in DKE at Maastricht University.

Since August 2020, he has been working as a junior lecturer at the Artificial intelligence master program at University of Amsterdam (UvA) in multiple courses such as: Machine learning, Deep learning, Advanced Deep Learning, Fairness Accountability Confidentiality and Transparency in AI and Explainability and Interpretability in AI. Part of his role is to prepare materials, lectures and tutorials for all these courses, but also to work in research projects with students and Ph.D. candidates from the UvA by conducting research in the recent advances in these fields.

List of Publications

- *C.Athanasiadis, E.Hortal, D.Koutsoukos, C.Zarco Lens, S. Asteriadis, Personalized, Affect and Performance-driven Computer-based Learning, the International Conference on Computer Supported Education (CSEDU), Porto, 2017.*

- *C.Athanasiadis, E.Hortal, M.Amestoy and S. Asteriadis, e-3 learning: a Dataset for Affect-driven Adaptation of Computer-Based Learning*[5]*, IEEE Multimedia, Volume 27, Number 1, Pages: 49-60, 2019.*

- *C.Athanasiadis, E.Hortal, and S.Asteriadis, Bridging face and sound modalities through Domain Adaptation Metric Learning, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), 2019.*

- *C.Athanasiadis, E.Hortal, and S.Asteriadis, Audio–visual domain adaptation using conditional semi-supervised Generative Adversarial Networks, Neurocomputing Elsevier*[6]*, Volume 397, Pages: 331–344, 2020.*

- *C.Athanasiadis, E.Hortal, and S.Asteriadis, Audio based Emotion Recognition enhancement through Progressive GANs, 27th international conference on image processing (ICIP), Abu Dabi, 2020.*

- *C.Athanasiadis, E.Hortal, S.Asteriadis, Temporal wGANs for investigating audio-visual affect ties,* 9^{th} *International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW), 2021.*

[5] https://10.1109/MMUL.2019.2945716

[6] https://doi.org/10.1016/j.neucom.2019.09.106

SIKS Dissertation Series

2011

1. Botond Cseke (RUN), Variational Algorithms for Bayesian Inference in Latent Gaussian Models
2. Nick Tinnemeier (UU), Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language
3. Jan Martijn van der Werf (TUE), Compositional Design and Verification of Component-Based Information Systems
4. Hado van Hasselt (UU), Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference
5. Bas van der Raadt (VU), Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.
6. Yiwen Wang (TUE), Semantically-Enhanced Recommendations in Cultural Heritage
7. Yujia Cao (UT), Multimodal Information Presentation for High Load Human Computer Interaction
8. Nieske Vergunst (UU), BDI-based Generation of Robust Task-Oriented Dialogues
9. Tim de Jong (OU), Contextualised Mobile Media for Learning
10. Bart Bogaert (UvT), Cloud Content Contention
11. Dhaval Vyas (UT), Designing for Awareness: An Experience-focused HCI Perspective
12. Carmen Bratosin (TUE), Grid Architecture for Distributed Process Mining
13. Xiaoyu Mao (UvT), Airport under Control. Multiagent Scheduling for Airport Ground Handling
14. Milan Lovric (EUR), Behavioral Finance and Agent-Based Artificial Markets
15. Marijn Koolen (UvA), The Meaning of Structure: the Value of Link Evidence for Information Retrieval
16. Maarten Schadd (UM), Selective Search in Games of Different Complexity
17. Jiyin He (UVA), Exploring Topic Structure: Coherence, Diversity and Relatedness
18. Mark Ponsen (UM), Strategic Decision-Making in complex games
19. Ellen Rusman (OU), The Mind's Eye on Personal Profiles
20. Qing Gu (VU), Guiding service-oriented software engineering - A view-based approach
21. Linda Terlouw (TUD), Modularization and Specification of Service-Oriented Systems
22. Junte Zhang (UVA), System Evaluation of Archival Description and Access
23. Wouter Weerkamp (UVA), Finding People and their Utterances in Social Media
24. Herwin van Welbergen (UT), Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior
25. Syed Waqar ul Qounain Jaffry (VU), Analysis and Validation of Models for Trust Dynamics
26. Matthijs Aart Pontier (VU), Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots
27. Aniel Bhulai (VU), Dynamic website optimization through autonomous management of design patterns

28. Rianne Kaptein (UVA), Effective Focused Retrieval by Exploiting Query Context and Document Structure
29. Faisal Kamiran (TUE), Discrimination-aware Classification
30. Egon van den Broek (UT), Affective Signal Processing (ASP): Unraveling the mystery of emotions
31. Ludo Waltman (EUR), Computational and Game-Theoretic Approaches for Modeling Bounded Rationality
32. Nees-Jan van Eck (EUR), Methodological Advances in Bibliometric Mapping of Science
33. Tom van der Weide (UU), Arguing to Motivate Decisions
34. Paolo Turrini (UU), Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations
35. Maaike Harbers (UU), Explaining Agent Behavior in Virtual Training
36. Erik van der Spek (UU), Experiments in serious game design: a cognitive approach
37. Adriana Burlutiu (RUN), Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference
38. Nyree Lemmens (UM), Bee-inspired Distributed Optimization
39. Joost Westra (UU), Organizing Adaptation using Agents in Serious Games
40. Viktor Clerc (VU), Architectural Knowledge Management in Global Software Development
41. Luan Ibraimi (UT), Cryptographically Enforced Distributed Data Access Control
42. Michal Sindlar (UU), Explaining Behavior through Mental State Attribution
43. Henk van der Schuur (UU), Process Improvement through Software Operation Knowledge
44. Boris Reuderink (UT), Robust Brain-Computer Interfaces
45. Herman Stehouwer (UvT), Statistical Language Models for Alternative Sequence Selection
46. Beibei Hu (TUD), Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work
47. Azizi Bin Ab Aziz (VU), Exploring Computational Models for Intelligent Support of Persons with Depression
48. Mark Ter Maat (UT), Response Selection and Turn-taking for a Sensitive Artificial Listening Agent
49. Andreea Niculescu (UT), Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality

2012

1. Terry Kakeeto (UvT), Relationship Marketing for SMEs in Uganda
2. Muhammad Umair (VU), Adaptivity, emotion, and Rationality in Human and Ambient Agent Models
3. Adam Vanya (VU), Supporting Architecture Evolution by Mining Software Repositories
4. Jurriaan Souer (UU), Development of Content Management System-based Web Applications
5. Marijn Plomp (UU), Maturing Interorganisational Information Systems
6. Wolfgang Reinhardt (OU), Awareness Support for Knowledge Workers in Research Networks
7. Rianne van Lambalgen (VU), When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions
8. Gerben de Vries (UVA), Kernel Methods for Vessel Trajectories
9. Ricardo Neisse (UT), Trust and Privacy Management Support for Context-Aware Service Platforms
10. David Smits (TUE), Towards a Generic Distributed Adaptive Hypermedia Environment

11. J.C.B. Rantham Prabhakara (TUE), Process Mining in the Large: Preprocessing, Discovery, and Diagnostics
12. Kees van der Sluijs (TUE), Model Driven Design and Data Integration in Semantic Web Information Systems
13. Suleman Shahid (UvT), Fun and Face: Exploring non-verbal expressions of emotion during playful interactions
14. Evgeny Knutov (TUE), Generic Adaptation Framework for Unifying Adaptive Web-based Systems
15. Natalie van der Wal (VU), Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes.
16. Fiemke Both (VU), Helping people by understanding them - Ambient Agents supporting task execution and depression treatment
17. Amal Elgammal (UvT), Towards a Comprehensive Framework for Business Process Compliance
18. Eltjo Poort (VU), Improving Solution Architecting Practices
19. Helen Schonenberg (TUE), What's Next? Operational Support for Business Process Execution
20. Ali Bahramisharif (RUN), Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing
21. Roberto Cornacchia (TUD), Querying Sparse Matrices for Information Retrieval
22. Thijs Vis (UvT), Intelligence, politie en veiligheidsdienst: verenigbare grootheden?
23. Christian Muehl (UT), Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction
24. Laurens van der Werff (UT), Evaluation of Noisy Transcripts for Spoken Document Retrieval
25. Silja Eckartz (UT), Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application
26. Emile de Maat (UVA), Making Sense of Legal Text
27. Hayrettin Gurkok (UT), Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games
28. Nancy Pascall (UvT), Engendering Technology Empowering Women
29. Almer Tigelaar (UT), Peer-to-Peer Information Retrieval
30. Alina Pommeranz (TUD), Designing Human-Centered Systems for Reflective Decision Making
31. Emily Bagarukayo (RUN), A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure
32. Wietske Visser (TUD), Qualitative multicriteria preference representation and reasoning
33. Rory Sie (OUN), Coalitions in Cooperation Networks (COCOON)
34. Pavol Jancura (RUN), Evolutionary analysis in PPI networks and applications
35. Evert Haasdijk (VU), Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics
36. Denis Ssebugwawo (RUN), Analysis and Evaluation of Collaborative Modeling Processes
37. Agnes Nakakawa (RUN), A Collaboration Process for Enterprise Architecture Creation
38. Selmar Smit (VU), Parameter Tuning and Scientific Testing in Evolutionary Algorithms
39. Hassan Fatemi (UT), Risk-aware design of value and coordination networks
40. Agus Gunawan (UvT), Information Access for SMEs in Indonesia
41. Sebastian Kelle (OU), Game Design Patterns for Learning
42. Dominique Verpoorten (OU), Reflection Amplifiers in self-regulated Learning
43. (Withdrawn)
44. Anna Tordai (VU), On Combining Alignment Techniques

45. Benedikt Kratz (UvT), A Model and Language for Business-aware Transactions
46. Simon Carter (UVA), Exploration and Exploitation of Multilingual Data for Statistical Machine Translation
47. Manos Tsagkias (UVA), Mining Social Media: Tracking Content and Predicting Behavior
48. Jorn Bakker (TUE), Handling Abrupt Changes in Evolving Time-series Data
49. Michael Kaisers (UM), Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions
50. Steven van Kervel (TUD), Ontologogy driven Enterprise Information Systems Engineering
51. Jeroen de Jong (TUD), Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching

2013

1. Viorel Milea (EUR), News Analytics for Financial Decision Support
2. Erietta Liarou (CWI), MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing
3. Szymon Klarman (VU), Reasoning with Contexts in Description Logics
4. Chetan Yadati (TUD), Coordinating autonomous planning and scheduling
5. Dulce Pumareja (UT), Groupware Requirements Evolutions Patterns
6. Romulo Goncalves (CWI), The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience
7. Giel van Lankveld (UvT), Quantifying Individual Player Differences
8. Robbert-Jan Merk (VU), Making enemies: cognitive modeling for opponent agents in fighter pilot simulators
9. Fabio Gori (RUN), Metagenomic Data Analysis: Computational Methods and Applications
10. Jeewanie Jayasinghe Arachchige (UvT), A Unified Modeling Framework for Service Design.
11. Evangelos Pournaras (TUD), Multi-level Reconfigurable Self-organization in Overlay Services
12. Marian Razavian (VU), Knowledge-driven Migration to Services
13. Mohammad Safiri (UT), Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly
14. Jafar Tanha (UVA), Ensemble Approaches to Semi-Supervised Learning Learning
15. Daniel Hennes (UM), Multiagent Learning - Dynamic Games and Applications
16. Eric Kok (UU), Exploring the practical benefits of argumentation in multi-agent deliberation
17. Koen Kok (VU), The PowerMatcher: Smart Coordination for the Smart Electricity Grid
18. Jeroen Janssens (UvT), Outlier Selection and One-Class Classification
19. Renze Steenhuizen (TUD), Coordinated Multi-Agent Planning and Scheduling
20. Katja Hofmann (UvA), Fast and Reliable Online Learning to Rank for Information Retrieval
21. Sander Wubben (UvT), Text-to-text generation by monolingual machine translation
22. Tom Claassen (RUN), Causal Discovery and Logic
23. Patricio de Alencar Silva (UvT), Value Activity Monitoring
24. Haitham Bou Ammar (UM), Automated Transfer in Reinforcement Learning
25. Agnieszka Anna Latoszek-Berendsen (UM), Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System
26. Alireza Zarghami (UT), Architectural Support for Dynamic Homecare Service Provisioning

27. Mohammad Huq (UT), Inference-based Framework Managing Data Provenance
28. Frans van der Sluis (UT), When Complexity becomes Interesting: An Inquiry into the Information eXperience
29. Iwan de Kok (UT), Listening Heads
30. Joyce Nakatumba (TUE), Resource-Aware Business Process Management: Analysis and Support
31. Dinh Khoa Nguyen (UvT), Blueprint Model and Language for Engineering Cloud Applications
32. Kamakshi Rajagopal (OUN), Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development
33. Qi Gao (TUD), User Modeling and Personalization in the Microblogging Sphere
34. Kien Tjin-Kam-Jet (UT), Distributed Deep Web Search
35. Abdallah El Ali (UvA), Minimal Mobile Human Computer Interaction
36. Than Lam Hoang (TUe), Pattern Mining in Data Streams
37. Dirk Börner (OUN), Ambient Learning Displays
38. Eelco den Heijer (VU), Autonomous Evolutionary Art
39. Joop de Jong (TUD), A Method for Enterprise Ontology based Design of Enterprise Information Systems
40. Pim Nijssen (UM), Monte-Carlo Tree Search for Multi-Player Games
41. Jochem Liem (UVA), Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning
42. Léon Planken (TUD), Algorithms for Simple Temporal Reasoning
43. Marc Bron (UVA), Exploration and Contextualization through Interaction and Concepts

2014

1. Nicola Barile (UU), Studies in Learning Monotone Models from Data
2. Fiona Tuliyano (RUN), Combining System Dynamics with a Domain Modeling Method
3. Sergio Raul Duarte Torres (UT), Information Retrieval for Children: Search Behavior and Solutions
4. Hanna Jochmann-Mannak (UT), Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation
5. Jurriaan van Reijsen (UU), Knowledge Perspectives on Advancing Dynamic Capability
6. Damian Tamburri (VU), Supporting Networked Software Development
7. Arya Adriansyah (TUE), Aligning Observed and Modeled Behavior
8. Samur Araujo (TUD), Data Integration over Distributed and Heterogeneous Data Endpoints
9. Philip Jackson (UvT), Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language
10. Ivan Salvador Razo Zapata (VU), Service Value Networks
11. Janneke van der Zwaan (TUD), An Empathic Virtual Buddy for Social Support
12. Willem van Willigen (VU), Look Ma, No Hands: Aspects of Autonomous Vehicle Control
13. Arlette van Wissen (VU), Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains
14. Yangyang Shi (TUD), Language Models With Meta-information
15. Natalya Mogles (VU), Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare
16. Krystyna Milian (VU), Supporting trial recruitment and design by automatically interpreting eligibility criteria

17. Kathrin Dentler (VU), Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability
18. Mattijs Ghijsen (UVA), Methods and Models for the Design and Study of Dynamic Agent Organizations
19. Vinicius Ramos (TUE), Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support
20. Mena Habib (UT), Named Entity Extraction and Disambiguation for Informal Text: The Missing Link
21. Kassidy Clark (TUD), Negotiation and Monitoring in Open Environments
22. Marieke Peeters (UU), Personalized Educational Games - Developing agent-supported scenario-based training
23. Eleftherios Sidirourgos (UvA/CWI), Space Efficient Indexes for the Big Data Era
24. Davide Ceolin (VU), Trusting Semi-structured Web Data
25. Martijn Lappenschaar (RUN), New network models for the analysis of disease interaction
26. Tim Baarslag (TUD), What to Bid and When to Stop
27. Rui Jorge Almeida (EUR), Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty
28. Anna Chmielowiec (VU), Decentralized k-Clique Matching
29. Jaap Kabbedijk (UU), Variability in Multi-Tenant Enterprise Software
30. Peter de Cock (UvT), Anticipating Criminal Behaviour
31. Leo van Moergestel (UU), Agent Technology in Agile Multiparallel Manufacturing and Product Support
32. Naser Ayat (UvA), On Entity Resolution in Probabilistic Data
33. Tesfa Tegegne (RUN), Service Discovery in eHealth
34. Christina Manteli (VU), The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems.
35. Joost van Ooijen (UU), Cognitive Agents in Virtual Worlds: A Middleware Design Approach
36. Joos Buijs (TUE), Flexible Evolutionary Algorithms for Mining Structured Process Models
37. Maral Dadvar (UT), Experts and Machines United Against Cyberbullying
38. Danny Plass-Oude Bos (UT), Making brain-computer interfaces better: improving usability through post-processing.
39. Jasmina Maric (UvT), Web Communities, Immigration, and Social Capital
40. Walter Omona (RUN), A Framework for Knowledge Management Using ICT in Higher Education
41. Frederic Hogenboom (EUR), Automated Detection of Financial Events in News Text
42. Carsten Eijckhof (CWI/TUD), Contextual Multidimensional Relevance Models
43. Kevin Vlaanderen (UU), Supporting Process Improvement using Method Increments
44. Paulien Meesters (UvT), Intelligent Blauw. Met als ondertitel: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden.
45. Birgit Schmitz (OUN), Mobile Games for Learning: A Pattern-Based Approach
46. Ke Tao (TUD), Social Web Data Analytics: Relevance, Redundancy, Diversity
47. Shangsong Liang (UVA), Fusion and Diversification in Information Retrieval

2015

1. Niels Netten (UvA), Machine Learning for Relevance of Information in Crisis Response
2. Faiza Bukhsh (UvT), Smart auditing: Innovative Compliance Checking in Customs Controls

3. Twan van Laarhoven (RUN), Machine learning for network data
4. Howard Spoelstra (OUN), Collaborations in Open Learning Environments
5. Christoph Bösch (UT), Cryptographically Enforced Search Pattern Hiding
6. Farideh Heidari (TUD), Business Process Quality Computation - Computing Non-Functional Requirements to Improve Business Processes
7. Maria-Hendrike Peetz (UvA), Time-Aware Online Reputation Analysis
8. Jie Jiang (TUD), Organizational Compliance: An agent-based model for designing and evaluating organizational interactions
9. Randy Klaassen (UT), HCI Perspectives on Behavior Change Support Systems
10. Henry Hermans (OUN), OpenU: design of an integrated system to support lifelong learning
11. Yongming Luo (TUE), Designing algorithms for big graph datasets: A study of computing bisimulation and joins
12. Julie M. Birkholz (VU), Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks
13. Giuseppe Procaccianti (VU), Energy-Efficient Software
14. Bart van Straalen (UT), A cognitive approach to modeling bad news conversations
15. Klaas Andries de Graaf (VU), Ontology-based Software Architecture Documentation
16. Changyun Wei (UT), Cognitive Coordination for Cooperative Multi-Robot Teamwork
17. André van Cleeff (UT), Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs
18. Holger Pirk (CWI), Waste Not, Want Not! - Managing Relational Data in Asymmetric Memories
19. Bernardo Tabuenca (OUN), Ubiquitous Technology for Lifelong Learners
20. Lois Vanhée (UU), Using Culture and Values to Support Flexible Coordination
21. Sibren Fetter (OUN), Using Peer-Support to Expand and Stabilize Online Learning
22. Zhemin Zhu (UT), Co-occurrence Rate Networks
23. Luit Gazendam (VU), Cataloguer Support in Cultural Heritage
24. Richard Berendsen (UVA), Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation
25. Steven Woudenberg (UU), Bayesian Tools for Early Disease Detection
26. Alexander Hogenboom (EUR), Sentiment Analysis of Text Guided by Semantics and Structure
27. Sándor Héman (CWI), Updating compressed colomn stores
28. Janet Bagorogoza (TiU), Knowledge Management and High Performance; The Uganda Financial Institutions Model for HPO
29. Hendrik Baier (UM), Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains
30. Kiavash Bahreini (OU), Real-time Multimodal Emotion Recognition in E-Learning
31. Yakup Koç (TUD), On the robustness of Power Grids
32. Jerome Gard (UL), Corporate Venture Management in SMEs
33. Frederik Schadd (TUD), Ontology Mapping with Auxiliary Resources
34. Victor de Graaf (UT), Gesocial Recommender Systems
35. Jungxao Xu (TUD), Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction

2016

1. Syed Saiden Abbas (RUN), Recognition of Shapes by Humans and Machines
2. Michiel Christiaan Meulendijk (UU), Optimizing medication reviews through decision support: prescribing a better pill to swallow

3. Maya Sappelli (RUN), Knowledge Work in Context: User Centered Knowledge Worker Support
4. Laurens Rietveld (VU), Publishing and Consuming Linked Data
5. Evgeny Sherkhonov (UVA), Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers
6. Michel Wilson (TUD), Robust scheduling in an uncertain environment
7. Jeroen de Man (VU), Measuring and modeling negative emotions for virtual training
8. Matje van de Camp (TiU), A Link to the Past: Constructing Historical Social Networks from Unstructured Data
9. Archana Nottamkandath (VU), Trusting Crowdsourced Information on Cultural Artefacts
10. George Karafotias (VUA), Parameter Control for Evolutionary Algorithms
11. Anne Schuth (UVA), Search Engines that Learn from Their Users
12. Max Knobbout (UU), Logics for Modelling and Verifying Normative Multi-Agent Systems
13. Nana Baah Gyan (VU), The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach
14. Ravi Khadka (UU), Revisiting Legacy Software System Modernization
15. Steffen Michels (RUN), Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments
16. Guangliang Li (UVA), Socially Intelligent Autonomous Agents that Learn from Human Reward
17. Berend Weel (VU), Towards Embodied Evolution of Robot Organisms
18. Albert Meroño Peñuela (VU), Refining Statistical Data on the Web
19. Julia Efremova (Tu/e), Mining Social Structures from Genealogical Data
20. Daan Odijk (UVA), Context & Semantics in News & Web Search
21. Alejandro Moreno Célleri (UT), From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground
22. Grace Lewis (VU), Software Architecture Strategies for Cyber-Foraging Systems
23. Fei Cai (UVA), Query Auto Completion in Information Retrieval
24. Brend Wanders (UT), Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach
25. Julia Kiseleva (TU/e), Using Contextual Information to Understand Searching and Browsing Behavior
26. Dilhan Thilakarathne (VU), In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains
27. Wen Li (TUD), Understanding Geospatial Information on Social Media
28. Mingxin Zhang (TUD), Large-scale Agent-based Social Simulation - A study on epidemic prediction and control
29. Nicolas Höning (TUD), Peak reduction in decentralised electricity systems - Markets and prices for flexible planning
30. Ruud Mattheij (UvT), The Eyes Have It
31. Mohammad Khelghati (UT), Deep web content monitoring
32. Eelco Vriezekolk (UT), Assessing Telecommunication Service Availability Risks for Crisis Organisations
33. Peter Bloem (UVA), Single Sample Statistics, exercises in learning from just one example
34. Dennis Schunselaar (TUE), Configurable Process Trees: Elicitation, Analysis, and Enactment
35. Zhaochun Ren (UVA), Monitoring Social Media: Summarization, Classification and Recommendation
36. Daphne Karreman (UT), Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies

37. Giovanni Sileno (UvA), Aligning Law and Action - a conceptual and computational inquiry
38. Andrea Minuto (UT), Materials that Matter - Smart Materials meet Art & Interaction Design
39. Merijn Bruijnes (UT), Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect
40. Christian Detweiler (TUD), Accounting for Values in Design
41. Thomas King (TUD), Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance
42. Spyros Martzoukos (UVA), Combinatorial and Compositional Aspects of Bilingual Aligned Corpora
43. Saskia Koldijk (RUN), Context-Aware Support for Stress Self-Management: From Theory to Practice
44. Thibault Sellam (UVA), Automatic Assistants for Database Exploration
45. Bram van de Laar (UT), Experiencing Brain-Computer Interface Control
46. Jorge Gallego Perez (UT), Robots to Make you Happy
47. Christina Weber (UL), Real-time foresight - Preparedness for dynamic innovation networks
48. Tanja Buttler (TUD), Collecting Lessons Learned
49. Gleb Polevoy (TUD), Participation and Interaction in Projects. A Game-Theoretic Analysis
50. Yan Wang (UVT), The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains

2017

1. Jan-Jaap Oerlemans (UL), Investigating Cybercrime
2. Sjoerd Timmer (UU), Designing and Understanding Forensic Bayesian Networks using Argumentation
3. Daniël Harold Telgen (UU), Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines
4. Mrunal Gawade (CWI), Multi-core Parallelism in a Column-store
5. Mahdieh Shadi (UVA), Collaboration Behavior
6. Damir Vandic (EUR), Intelligent Information Systems for Web Product Search
7. Roel Bertens (UU), Insight in Information: from Abstract to Anomaly
8. Rob Konijn (VU) , Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery
9. Dong Nguyen (UT), Text as Social and Cultural Data: A Computational Perspective on Variation in Text
10. Robby van Delden (UT), (Steering) Interactive Play Behavior
11. Florian Kunneman (RUN), Modelling patterns of time and emotion in Twitter #anticipointment
12. Sander Leemans (TUE), Robust Process Mining with Guarantees
13. Gijs Huisman (UT), Social Touch Technology - Extending the reach of social touch through haptic technology
14. Shoshannah Tekofsky (UvT), You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior
15. Peter Berck (RUN), Memory-Based Text Correction
16. Aleksandr Chuklin (UVA), Understanding and Modeling Users of Modern Search Engines
17. Daniel Dimov (UL), Crowdsourced Online Dispute Resolution
18. Ridho Reinanda (UVA), Entity Associations for Search
19. Jeroen Vuurens (UT), Proximity of Terms, Texts and Semantic Vectors in Information Retrieval

20. Mohammadbashir Sedighi (TUD), Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility
21. Jeroen Linssen (UT), Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)
22. Sara Magliacane (VU), Logics for causal inference under uncertainty
23. David Graus (UVA), Entities of Interest — Discovery in Digital Traces
24. Chang Wang (TUD), Use of Affordances for Efficient Robot Learning
25. Veruska Zamborlini (VU), Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search
26. Merel Jung (UT), Socially intelligent robots that understand and respond to human touch
27. Michiel Joosse (UT), Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors
28. John Klein (VU), Architecture Practices for Complex Contexts
29. Adel Alhuraibi (UvT), From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT"
30. Wilma Latuny (UvT), The Power of Facial Expressions
31. Ben Ruijl (UL), Advances in computational methods for QFT calculations
32. Thaer Samar (RUN), Access to and Retrievability of Content in Web Archives
33. Brigit van Loggem (OU), Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity
34. Maren Scheffel (OU), The Evaluation Framework for Learning Analytics
35. Martine de Vos (VU), Interpreting natural science spreadsheets
36. Yuanhao Guo (UL), Shape Analysis for Phenotype Characterisation from High-throughput Imaging
37. Alejandro Montes Garcia (TUE), WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy
38. Alex Kayal (TUD), Normative Social Applications
39. Sara Ahmadi (RUN), Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR
40. Altaf Hussain Abro (VUA), Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems
41. Adnan Manzoor (VUA), Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle
42. Elena Sokolova (RUN), Causal discovery from mixed and missing data with applications on ADHD datasets
43. Maaike de Boer (RUN), Semantic Mapping in Video Retrieval
44. Garm Lucassen (UU), Understanding User Stories - Computational Linguistics in Agile Requirements Engineering
45. Bas Testerink (UU), Decentralized Runtime Norm Enforcement
46. Jan Schneider (OU), Sensor-based Learning Support
47. Jie Yang (TUD), Crowd Knowledge Creation Acceleration
48. Angel Suarez (OU), Collaborative inquiry-based learning

2018

1. Han van der Aa (VUA), Comparing and Aligning Process Representations
2. Felix Mannhardt (TUE), Multi-perspective Process Mining
3. Steven Bosems (UT), Causal Models For Well-Being: Knowledge Modeling,

Model-Driven Development of Context-Aware Applications, and Behavior Prediction

4. Jordan Janeiro (TUD), Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks
5. Hugo Huurdeman (UVA), Supporting the Complex Dynamics of the Information Seeking Process
6. Dan Ionita (UT), Model-Driven Information Security Risk Assessment of Socio-Technical Systems
7. Jieting Luo (UU), A formal account of opportunism in multi-agent systems
8. Rick Smetsers (RUN), Advances in Model Learning for Software Systems
9. Xu Xie (TUD), Data Assimilation in Discrete Event Simulations
10. Julienka Mollee (VUA), Moving forward: supporting physical activity behavior change through intelligent technology
11. Mahdi Sargolzaei (UVA), Enabling Framework for Service-oriented Collaborative Networks
12. Xixi Lu (TUE), Using behavioral context in process mining
13. Seyed Amin Tabatabaei (VUA), Computing a Sustainable Future
14. Bart Joosten (UVT), Detecting Social Signals with Spatiotemporal Gabor Filters
15. Naser Davarzani (UM), Biomarker discovery in heart failure
16. Jaebok Kim (UT), Automatic recognition of engagement and emotion in a group of children
17. Jianpeng Zhang (TUE), On Graph Sample Clustering
18. Henriette Nakad (UL), De Notaris en Private Rechtspraak
19. Minh Duc Pham (VUA), Emergent relational schemas for RDF
20. Manxia Liu (RUN), Time and Bayesian Networks
21. Aad Slootmaker (OUN), EMERGO: a generic platform for authoring and playing scenario-based serious games
22. Eric Fernandes de Mello Araujo (VUA), Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks
23. Kim Schouten (EUR), Semantics-driven Aspect-Based Sentiment Analysis
24. Jered Vroon (UT), Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots
25. Riste Gligorov (VUA), Serious Games in Audio-Visual Collections
26. Roelof Anne Jelle de Vries (UT),Theory-Based and Tailor-Made: Motivational Messages for Behavior Change Technology
27. Maikel Leemans (TUE), Hierarchical Process Mining for Scalable Software Analysis
28. Christian Willemse (UT), Social Touch Technologies: How they feel and how they make you feel
29. Yu Gu (UVT), Emotion Recognition from Mandarin Speech
30. Wouter Beek, The "K" in "semantic web" stands for "knowledge": scaling semantics to the web

2019

1. Rob van Eijk (UL),Web privacy measurement in real-time bidding systems. A graph-based approach to RTB system classification
2. Emmanuelle Beauxis Aussalet (CWI, UU), Statistics and Visualizations for Assessing Class Size Uncertainty
3. Eduardo Gonzalez Lopez de Murillas (TUE), Process Mining on Databases: Extracting Event Data from Real Life Data
4. Ridho Rahmadi (RUN), Finding stable causal structures from clinical data
5. Sebastiaan van Zelst (TUE), Process Mining with Streaming Data
6. Chris Dijkshoorn (VU), Nichesourcing for Improving Access to Linked Cultural Heritage Datasets
7. Soude Fazeli (TUD), Recommender Systems in Social Learning Platforms

8. Frits de Nijs (TUD), Resource-constrained Multi-agent Markov Decision Processes
9. Fahimeh Alizadeh Moghaddam (UVA), Self-adaptation for energy efficiency in software systems
10. Qing Chuan Ye (EUR), Multi-objective Optimization Methods for Allocation and Prediction
11. Yue Zhao (TUD), Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs
12. Jacqueline Heinerman (VU), Better Together
13. Guanliang Chen (TUD), MOOC Analytics: Learner Modeling and Content Generation
14. Daniel Davis (TUD), Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses
15. Erwin Walraven (TUD), Planning under Uncertainty in Constrained and Partially Observable Environments
16. Guangming Li (TUE), Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models
17. Ali Hurriyetoglu (RUN),Extracting actionable information from microtexts
18. Gerard Wagenaar (UU), Artefacts in Agile Team Communication
19. Vincent Koeman (TUD), Tools for Developing Cognitive Agents
20. Chide Groenouwe (UU), Fostering technically augmented human collective intelligence
21. Cong Liu (TUE), Software Data Analytics: Architectural Model Discovery and Design Pattern Detection
22. Martin van den Berg (VU),Improving IT Decisions with Enterprise Architecture
23. Qin Liu (TUD), Intelligent Control Systems: Learning, Interpreting, Verification
24. Anca Dumitrache (VU), Truth in Disagreement - Crowdsourcing Labeled Data for Natural Language Processing
25. Emiel van Miltenburg (VU), Pragmatic factors in (automatic) image description
26. Prince Singh (UT), An Integration Platform for Synchromodal Transport
27. Alessandra Antonaci (OUN), The Gamification Design Process applied to (Massive) Open Online Courses
28. Esther Kuindersma (UL), Cleared for take-off: Game-based learning to prepare airline pilots for critical situations
29. Daniel Formolo (VU), Using virtual agents for simulation and training of social skills in safety-critical circumstances
30. Vahid Yazdanpanah (UT), Multiagent Industrial Symbiosis Systems
31. Milan Jelisavcic (VU), Alive and Kicking: Baby Steps in Robotics
32. Chiara Sironi (UM), Monte-Carlo Tree Search for Artificial General Intelligence in Games
33. Anil Yaman (TUE), Evolution of Biologically Inspired Learning in Artificial Neural Networks
34. Negar Ahmadi (TUE), EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES
35. Lisa Facey-Shaw (OUN), Gamification with digital badges in learning programming
36. Kevin Ackermans (OUN), Designing Video-Enhanced Rubrics to Master Complex Skills
37. Jian Fang (TUD), Database Acceleration on FPGAs
38. Akos Kadar (OUN), Learning visually grounded and multilingual representations

2020

1. Armon Toubman (UL), Calculated Moves: Generating Air Combat Behaviour
2. Marcos de Paula Bueno (UL), Unraveling Temporal Processes using Probabilistic Graphical Models
3. Mostafa Deghani (UvA), Learning with Imperfect Supervision for Language Understanding

4. Maarten van Gompel (RUN), Context as Linguistic Bridges
5. Yulong Pei (TUE), On local and global structure mining
6. Preethu Rose Anish (UT), Stimulation Architectural Thinking during Requirements Elicitation - An Approach and Tool Support
7. Wim van der Vegt (OUN), Towards a software architecture for reusable game components
8. Ali Mirsoleimani (UL),Structured Parallel Programming for Monte Carlo Tree Search
9. Myriam Traub (UU), Measuring Tool Bias and Improving Data Quality for Digital Humanities Research
10. Alifah Syamsiyah (TUE), In-database Preprocessing for Process Mining
11. Sepideh Mesbah (TUD), Semantic-Enhanced Training Data Augmentation-Methods for Long-Tail Entity Recognition Models
12. Ward van Breda (VU), Predictive Modeling in E-Mental Health: Exploring Applicability in Personalised Depression Treatment
13. Marco Virgolin (CWI), Design and Application of Gene-pool Optimal Mixing Evolutionary Algorithms for Genetic Programming
14. Mark Raasveldt (CWI/UL), Integrating Analytics with Relational Databases
15. Konstantinos Georgiadis (OUN), Smart CAT: Machine Learning for Configurable Assessments in Serious Games
16. Ilona Wilmont (RUN), Cognitive Aspects of Conceptual Modelling
17. Daniele Di Mitri (OUN), The Multimodal Tutor: Adaptive Feedback from Multimodal Experiences
18. Georgios Methenitis (TUD), Agent Interactions & Mechanisms in Markets with Uncertainties: Electricity Markets in Renewable Energy Systems
19. Guido van Capelleveen (UT), Industrial Symbiosis Recommender Systems
20. Albert Hankel (VU), Embedding Green ICT Maturity in Organisations
21. Karine da Silva Miras de Araujo (VU), Where is the robot?: Life as it could be
22. Maryam Masoud Khamis (RUN), Understanding complex systems implementation through a modeling approach: the case of e-government in Zanzibar
23. Rianne Conijn (UT), The Keys to Writing: A writing analytics approach to studying writing processes using keystroke logging
24. Lenin da Nobrega Medeiros (VUA/RUN), How are you feeling, human? Towards emotionally supportive chatbots
25. Xin Du (TUE), The Uncertainty in Exceptional Model Mining
26. Krzysztof Leszek Sadowski (UU), GAMBIT: Genetic Algorithm for Model-Based mixed-Integer opTimization
27. Ekaterina Muravyeva (TUD), Personal data and informed consent in an educational context
28. Bibeg Limbu (TUD), Multimodal interaction for deliberate practice: Training complex skills with augmented reality
29. Ioan Gabriel Bucur (RUN), Being Bayesian about Causal Inference
30. Bob Zadok Blok (UL), Creatief, Creatieve, Creatiefst
31. Gongjin Lan (VU), Learning better – From Baby to Better
32. Jason Rhuggenaath (TUE), Revenue management in online markets: pricing and online advertising
33. Rick Gilsing (TUE), Supporting service-dominant business model evaluation in the context of business model innovation
34. Anna Bon (MU), Intervention or Collaboration? Redesigning Information and Communication Technologies for Development
35. Siamak Farshidi (UU), Multi-Criteria Decision-Making in Software Production

2021

1. Francisco Xavier Dos Santos Fonseca (TUD),Location-based Games for Social Interaction in Public Space
2. Rijk Mercuur (TUD), Simulating Human Routines:Integrating Social Practice Theory in Agent-Based Models
3. Seyyed Hadi Hashemi (UVA), Modeling Users Interacting with Smart Devices
4. Ioana Jivet (OU), The Dashboard That Loved Me: Designing adaptive learning analytics for self-regulated learning
5. Davide Dell'Anna (UU), Data-Driven Supervision of Autonomous Systems
6. Daniel Davison (UT), "Hey robot, what do you think?" How children learn with a social robot
7. Armel Lefebvre (UU), Research data management for open science
8. Nardie Fanchamps (OU), The Influence of Sense-Reason-Act Programming on Computational Thinking
9. Cristina Zaga (UT), The Design of Robothings. Non-Anthropomorphic and Non-Verbal Robots to Promote Children's Collaboration Through Play
10. Quinten Meertens (UvA), Misclassification Bias in Statistical Learning
11. Anne van Rossum (UL), Nonparametric Bayesian Methods in Robotic Vision
12. Lei Pi (UL), External Knowledge Absorption in Chinese SMEs
13. Bob R. Schadenberg (UT), Robots for Autistic Children: Understanding and Facilitating Predictability for Engagement in Learning
14. Negin Samaeemofrad (UL), Business Incubators: The Impact of Their Support
15. Onat Ege Adali (TU/e), Transformation of Value Propositions into Resource Re-Configurations through the Business Services Paradigm
16. Onat Ege Adali (TU/e), Transformation of Value Propositions into Resource Re-Configurations through the Business Services Paradigm
17. Esam A. H. Ghaleb (UM), Bimodal emotion recognition from audio-visual cues
18. Dario Dotti (UM), Human Behavior Understanding from motion and bodily cues using deep neural networks
19. Remi Wieten (UU), Bridging the Gap Between Informal Sense-Making Tools and Formal Systems - Facilitating the Construction of Bayesian Networks and Argumentation Frameworks
20. Remi Wieten (UU), Bridging the Gap Between Informal Sense-Making Tools and Formal Systems - Facilitating the Construction of Bayesian Networks and Argumentation Frameworks
21. Roberto Verdecchia (VU), Architectural Technical Debt: Identification and Management
22. Masoud Mansoury (TU/e), Understanding and Mitigating Multi-Sided Exposure Bias in Recommender Systems
23. Pedro Thiago Timbó Holanda (CWI), Progressive Indexes
24. Sihang Qiu (TUD), Conversational Crowdsourcing
25. Hugo Manuel Proença (LIACS), Robust rules for prediction and description
26. Kaijie Zhu (TUE), On Efficient Temporal Subgraph Query Processing
27. Jelte van Waterschoot (UT), Personalized and Personal Conversations: Designing Agents Who Want to Connect With You
28. Christoph Selig (UL), Understanding the Heterogeneity of Corporate Entrepreneurship Programs

2022

1. Judith van Stegeren (UT), Flavor text generation for role-playing video games
2. Paulo da Costa (TU/e), Data-driven Prognostics and Logistics Optimisation: A Deep Learning Journey
3. Ali el Hassouni (VUA), A Model A Day Keeps The Doctor Away: Reinforcement Learning For Personalized Healthcare
4. Ünal Aksu (UU), A Cross-Organizational Process Mining Framework
5. Shiwei Liu (TU/e), Sparse Neural Network Training with In-Time Over-Parameterization

6. Judith van Stegeren (UT), Flavor text generation for role-playing video games
7. Reza Refaei Afshar (TU/e), Machine Learning for Ad Publishers in Real Time Bidding
8. Sambit Praharaj (OU), Measuring the Unmeasurable? Towards Automatic Co-located Collaboration Analytics
9. Maikel L. van Eck (TU/e), Process Mining for Smart Product Design
10. Oana Andreea Inel (VUA), Understanding Events: A Diversity-driven Human-Machine Approach
11. Felipe Moraes Gomes (TUD), Examining the Effectiveness of Collaborative Search Engines
12. Mirjam de Haas (TiU), Staying engaged in child-robot interaction, a quantitative approach to studying preschoolers' engagement with robots and tasks during second-language tutoring
13. Guanyi Chen (UU), Computational Generation of Chinese Noun Phrases
14. Xander Wilcke (VUA), Machine Learning on Multimodal Knowledge Graphs: Opportunities, Challenges, and Methods for Learning on Real-World Heterogeneous and Spatially-Oriented Knowledge
15. Jelmer Jan Koorn (UU), Work in Process: Unearthing Meaning using Process Mining
16. Pieter Gijsbers (TU/e), Systems for AutoML Research
17. Laura van der Lubbe (VUA), Empowering vulnerable people with serious games and gamification
18. Paris Mavromoustakos Blom (TiU), Player Affect Modelling and Video Game Personalisation
19. Bilge Yigit Ozkan (UU), Cybersecurity Maturity Assessment and Standardisation
20. Fakhra Jabeen (VUA), Dark Side of the Digital Media - Computational Analysis of Negative Human Behaviors on Social Media
21. Seethu Mariyam Christopher (UM), Intelligent Toys for Physical and Cognitive Assessments
22. Alexandra Sierra Rativa (TiU), Virtual Character Design and its potential to foster Empathy, Immersion, and Collaboration Skills in Video Games and Virtual Reality Simulations
23. Ilir Kola (TUD), Enabling Social Situation Awareness in Support Agents
24. Samaneh Heidari (UU), Agents with Social Norms and Values - A framework for agent based social simulations with social norms and personal values
25. Anna L.D. Latour (LU), Optimal decision-making under constraints and uncertainty
26. Anne Dirkson (LU), Knowledge Discovery from Patient Forums: Gaining novel medical insights from patient experiences
27. Christos Athanasiadis (UM), Emotion-aware cross-modal domain adaptation in video sequences
28. Onuralp Ulusoy (UU), Privacy in Collaborative Systems
29. Jan Kolkmeier (UT-EEMCS), From Head Transform to Mind Transplant: Social Interactions in Mixed Reality