

QATAR UNIVERSITY

COLLEGE OF ENGINEERING

STATIC AND DYNAMIC FACIAL EMOTION RECOGNITION USING NEURAL

NETWORK MODELS

BY

EALAF SAYED AHMED HUSSEIN

A Thesis Submitted to

the College of Engineering

in Partial Fulfillment of the Requirements for the Degree of

Masters of Science in Computing

June 2020

© 2020 Ealaf Sayed Ahmed Hussein. All Rights Reserved.

COMMITTEE PAGE

The members of the Committee approve the Thesis of
Ealaf Sayed Ahmed Hussein defended on 30/04/2020.

Dr. Uvais Qidwai
Thesis Supervisor

Dr. Mohamed Al-meer
Thesis Co-Supervisor

Prof. Sumaya Al-Maadeed
Committee Member

Prof. Lee Yoot Khuan
Committee Member

Dr. Tamer Khattab
Committee Member

Approved:

Khalid Kamal Naji, Dean, College of Engineering

ABSTRACT

Hussein, Ealaf, S., Masters : June : 2020, Masters of Science in Computing

Title: Static and Dynamic Facial Emotion Recognition Using Neural Network Models

Supervisor of Thesis: Uvais, A., Qidwai.

Emotion recognition is the process of identifying human emotions. It is made possible by processing various modalities including facial expressions, speech signals, biometric signals, etc. With the advancements in computing technologies, Facial Emotion Recognition (FER) became important for several applications in which the user's emotional state is required, such as emotional training for autistic children. The recent years witnessed a major leap in Artificial Intelligence (AI), specially neural networks for computer vision applications. In this thesis, we investigate the application of AI algorithms for FER from static and dynamic data. Our experiments address the limitations and challenges of previous works such as limited generalizability due to the datasets. We compare the performance of machine learning classifiers and convolution neural networks (CNNs) for FER from static data (images). Moreover, we study the performance of the proposed CNN for dynamic FER (videos), in addition to Long-Short Term Memory (LSTM) in a CNN-LSTM hybrid approach to utilize the temporal information in the videos. The proposed CNN architecture outperformed the other classifiers with an accuracy of 86.5%. It also outperformed the hybrid approach for dynamic FER which achieved an accuracy of 74.6%

DEDICATION

*To my family and my friends
for their unconditional love and support*

ACKNOWLEDGMENTS

I thank Allah for giving me the knowledge and strength to accomplish this work.

I would like to deeply thank my supervisors Dr. Uvais and Dr. Al-Meer for their patience and continuous support, and for their valuable feedback throughout this thesis.

I sincerely appreciate all the support and love I got from my family and friends during this journey, for always pushing me to be the best I can, and for always believing in me.

TABLE OF CONTENTS

DEDICATION	iv
ACKNOWLEDGMENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ACRONYMS	xii
Chapter 1: Introduction.....	1
Research Objectives	2
Research Questions	3
Motivation.....	3
Thesis Structure	4
Chapter 2: Background and Foundations	5
Emotion Models.....	5
<i>Categorical Model</i>	5
<i>Dimensional Model</i>	6
FER Datasets.....	7
<i>Datasets Used</i>	11
Artificial Intelligence	13
<i>Support Vector Machine</i>	13
<i>Convolutional Neural Networks</i>	14
<i>Recurrent Neural Networks</i>	16

Chapter 3: Related Work	18
Legacy Facial Emotion Recognition	19
Static FER - Convolutional Neural Networks	21
Dynamic FER - Recurrent Neural Networks	24
Chapter 4: Static FER Using Artificial Intelligence	28
introduction	28
<i>Transfer Learning</i>	28
<i>Depth-wise Separable Convolutions</i>	29
<i>Residual Learning</i>	32
<i>Dense Block</i>	33
Methodology	34
<i>Pre-processing and Dataset Preparation</i>	34
<i>Machine Learning</i>	38
<i>Transfer Learning</i>	39
<i>CNN Architectures</i>	40
Experiments and Results	45
<i>Machine Learning</i>	47
<i>Transfer Learning</i>	49
<i>Proposed CNN</i>	52
Chapter 5: Dynamic Facial Emotion Recognition Using Artificial Intelligence	57
Introduction	57
<i>Proposed CNN model for Dynamic FER</i>	57

<i>A hybrid CNN-RNN Approach</i>	57
Methodology	58
<i>Dataset Preparation</i>	58
<i>Feature Extraction</i>	59
Experiments and Results	60
<i>Proposed CNN for Dynamic FER</i>	60
<i>Hybrid CNN-RNN</i>	62
Real-life Experiment: Training Session Evaluation	65
<i>Participants</i>	65
<i>Questionnaire</i>	65
<i>Observations</i>	66
Chapter 6: Discussion	67
Chapter 7: Conclusion and Future Work	70
References	72
Appendix A: CNN Architectures	81
Appendix B: Extended Results	83
Appendix C: Training Session Evaluation Experiment	85
Pre-training Questionnaire	85
Post-training Questionnaire	85

LIST OF TABLES

Table 3.1. Static FER Summary	26
Table 3.2. Dynamic FER Summary.....	27
Table 4.1. Relabelled Classes	35
Table 4.2. Data Augmentation Operations	37
Table 4.3. Pre-trained CNN Architectures.....	39
Table 4.4. Configurations of Proposed CNN Architecture.....	44
Table 4.5. Environment Specifications	45
Table 4.6. Class Weights.....	46
Table 4.7. Results of Machine Learning Classifiers	47
Table 4.8. Results of Machine Learning Classifiers with CNN Features	48
Table 4.9. VGG-16 Architecture for Transfer Learning	49
Table 4.10. ResNet Architecture for Transfer Learning	50
Table 4.11. Base Model Classification Report	54
Table 4.12. Proposed Model Classification Report	56
Table 5.1. LSTM Model Parameters.....	62
Table 6.1. FER Results Summary.....	67
Table A.1. VGG-16 and ResNet-50 for 3-Class on FER2013	81
Table A.2. VGG-16 and ResNet-50 for 7-Class on FER2013	81
Table A.3. VGG-16 and ResNet-50 for 3-Class on CK+.....	81
Table A.4. VGG-16 and ResNet-50 for 7-Class on CK+.....	81
Table A.5. VGG-16 and ResNet-50 for Cross-dataset.....	82

LIST OF FIGURES

Figure 2.1. Valence-Arousal dimensional emotion model.	6
Figure 2.2. Example of lab-controlled datasets.	8
Figure 2.3. Example of in-the-wild datasets.....	9
Figure 2.4. Example of dynamic and static datasets.....	10
Figure 2.5. Class distribution for the FER2013 dataset.....	12
Figure 2.6. Class distribution for the CK+ dataset.	12
Figure 2.7. Support Vector Machine (SVM).	13
Figure 2.8. 2D convolution.....	15
Figure 2.9. A simple RNN module.	16
Figure 2.10. An LSTM module.	17
Figure 3.1. Generic FER pipeline.....	19
Figure 3.2. FACS Sample [32].....	20
Figure 4.1. Convolution over a volume.	30
Figure 4.2. DC filtering stage.	31
Figure 4.3. PW combination stage.....	31
Figure 4.4. Residual block.	32
Figure 4.5. Dense block.....	33
Figure 4.6. Data preparation pipeline.	35
Figure 4.7. Class distribution for relabeled data.....	36
Figure 4.8. Data augmentation.	38
Figure 4.9. Base CNN model.	41

Figure 4.10. Dense block configuration.....	42
Figure 4.11. Proposed CNN model.	43
Figure 4.12. Transfer learning results.....	51
Figure 4.13. Confusion matrix for 3-class classification using base CNN model.....	53
Figure 4.14. Confusion matrix for 3-class classification using proposed CNN model.....	55
Figure 5.1. System diagram of hybrid CNN-RNN approach.	58
Figure 5.2. A sample of mis-classified frames.	60
Figure 5.3. LSTM training VS validation plots.	64
Figure B.1. Confusion matrix for 3-class FER2013.....	83
Figure B.2. Confusion matrix for 7-class FER2013.....	83
Figure B.3. Confusion matrix for 3-class CK+.	84
Figure B.4. Confusion matrix for 7-class CK+.	84
Figure B.5. Confusion matrix for cross-dataset.....	84

LIST OF ACRONYMS

AC Affective Computing.

AI Artificial Intelligence.

AU Action Unit.

CK+ Cohn-Kanade Extended Dataset.

CNN Convolution Neural Networks.

DC Depth-wise Convolution.

FACS Facial Action Coding System.

FC Fully-Connected.

FER Facial Emotion Recognition.

FER2013 Facial Emotion Recognition 2013 Dataset.

HOG Histogram of Oriented Gradients.

KNN K-Nearest Neighbour.

LBP Local Binary Pattern.

Leaky-Relu Leaky Rectified Linear Unit.

LSTM Long-Short Term Memory.

ML Machine Learning.

PC Point-wise Convolution.

PCA Principal Component Analysis.

ReLU Rectified Linear Unit.

RNN Recurrent Neural Network.

SVM Support Vector Machine.

CHAPTER 1: INTRODUCTION

The last few decades witnessed huge innovations that have opened the doors for automatically analysing human behaviour, such as gesture analysis, activities, etc. [1]. Human emotion is an important behavioural factor in human-human interaction as it relays the unvoiced state of a person, in addition to weighing heavily in many aspects of our daily lives such as learning and reasoning [2], [3]. With the advancements of computer technologies, emotion recognition gained momentum in numerous fields of study such as human-centred design in Human-Computer Interaction (HCI), in fact, several studies show the importance of machines interpreting human emotion for different applications [4]. Affective computing (AC) is a widely researched field that studies new approaches to enhance the communication between humans and machines [5]. The rapid growth in AC takes advantage of the different types of data such as text, speech, gestures, etc. for emotion recognition.

Emotion recognition is the process of identifying the emotional status of a subject, this can be achieved via several modalities including facial expression, speech signals, biometric signals, etc. [6]. Facial Emotion Recognition (FER) is not a new research field, it gained popularity when Paul Ekman adopted the Facial Action Coding System (FACS) to set a universal framework of the basic human emotions; anger, fear, disgust, sadness, surprise, happiness, neutral [7]. From there, researchers started handcrafting features and fed them to machine learning algorithms that performed well for FER [8]. These systems, however, have shortcomings for faces in the wild as they are trained using data collected in a controlled setting, with controlled posture, illumination and contrast, background, etc [9].

Emotional Stability is another term associated with emotion recognition, it is defined as the change in the displayed emotional state in reaction to a change in the perceived surroundings. In FER, emotional stability can be quantified by measuring the change of the facial emotion in a given time frame, T . In this research, we focus on FER for *negative, neutral and positive* emotions.

The importance of FER lies in the variety of applications and fields that could benefit from accurate and robust emotion recognition. Some of the recent applications target driver drowsiness detection, in which the FER module is integrated into an alerting system for cases of drowsiness [10]. Another field of application is more shifted towards psychological and medical applications; for instance, FER is used for monitoring the pain levels of bedridden patients. Additionally, there exist several works that study the emotion comprehension levels of people with several medical conditions and disorders such as Autism Spectrum Disorder (ASD) and Huntington's Disease [11], [12].

1.1. Research Objectives

In this thesis, we aim to develop an accurate and robust facial emotion classifier. For that, we investigate the effectiveness of using Artificial Intelligence (AI) for the task of Facial Emotion Recognition (FER). We first study different approaches to recognize facial emotion from static images such as machine learning algorithms, transfer learning from well-known pre-trained CNNs, in addition to proposing modifications that improve an existing CNN architecture. Moreover, we investigate the efficacy of CNNs for dynamic FER from image-sequences for the application of FER on videos. Finally, we explore the application and performance of hybrid CNN-RNN approach for video FER.

1.2. Research Questions

1. Do deep CNNs outperform machine learning approaches for FER?
2. How effective are CNNs for FER from videos?
3. How can RNNs improve classification for FER from videos?
4. Can a hybrid approach be used for accurate and robust classification of facial emotions?

1.3. Motivation

Despite the magnitude of research on FER, there exist several challenges that could arise when tackling such a problem, these challenges are mostly related to the datasets, deep learning models, and the nature of human emotion itself. The main challenges in FER are:

Challenge 1 – Most of the available datasets are collected in a controlled studio setting, referring to the posture of the subjects, background, illumination and contrast when collecting the data. Moreover, these datasets display exaggerated/extreme emotions rather than how emotions are actually exhibited in reality. This could limit the generalizability of the deep learning model.

Challenge 2 – For optimal classification of emotions, the intra-class class variance must be minimal, while inter-class variance must be maximal [13]. This could be a challenge due to the nature of human emotion, where some emotions appear to be similar such as fear and surprise.

Challenge 3 – Since machine learning is a data-centered study field, the performance and accuracy of the algorithms depend almost completely on the quality of the data. One aspect of data quality is diversity. There have been instances in the literature where machine learning algorithms resulted in biased classification. Hence, it is very important to have a dataset that is diverse in terms of race, age, gender, etc. to create an algorithm that generalizes well in real-life settings.

Challenge 4 – Most of the times, human emotion is not an instantaneous reaction to a stimulus, it is rather a prolonged change that happens to the facial features of a human over a period of time. Hence, it is very important to consider the temporal correlations in emotion.

1.4. Thesis Structure

This thesis is structured as follows; chapter 2 covers the background on the datasets and the techniques used in this thesis. Chapter 3 covers the literature and the current trends in FER. Chapters 4 and 5 cover static and dynamic FER, which include methodology, experiments and results for each approach. A thorough discussion reflecting on the results and findings is detailed in chapter 6. Finally, chapter 7 covers the conclusion and future work.

CHAPTER 2: BACKGROUND AND FOUNDATIONS

2.1. Emotion Models

In psychology, human emotion is defined as a psychological change that reflects thoughts, leading to physical change. Although it is quite subjective to standardize a scale for the quantification of human emotion, several studies and theories in the field of human psychology have been conducted [14]. Generally, an emotion model offers standards that facilitate the classification of human emotions. The most well-known models of emotion in the literature are the categorical (discrete) model, and the dimensional model.

2.1.1. Categorical Model

In this theory, the emotions are treated as discrete classes, typically six classes labelling the six basic emotions: anger, fear, disgust, joy, sadness and surprise. However, it is very common to have a domain-specific emotion classes. For example, in the context of emotions in a classroom setting, the emotion classes could be confusion, boredom and flow; because in such a situation, fear or disgust are less relevant to the context. Although this emotion theory is the most used for FER research, several studies argue that some factors such as cultural, linguistic, environmental differences impose difficulties on categorizing emotions. Other variations of the categorical model modify the classes to become more descriptive, such as bored-angry.

2.1.2. Dimensional Model

In this emotion model, the emotion is modelled in a multi-dimensional space where each dimension specifies a factor in categorizing emotions. Usually in such models, the emotion is defined within 2D or 3D spaces such as valence and arousal, or valence, arousal and power. The valence dimension reflects the polarity of the emotion, whether it is a positive or a negative emotion, hence, reflecting the sense of pleasantness of the emotion. The arousal dimension, on the other hand, reflects the intensity of the emotion, it could range from boredom to excitement. Other variations to this model also exists, where the dimensions are energy and stress. Fig. 2.1 shows an example of a 2D emotion model [15].

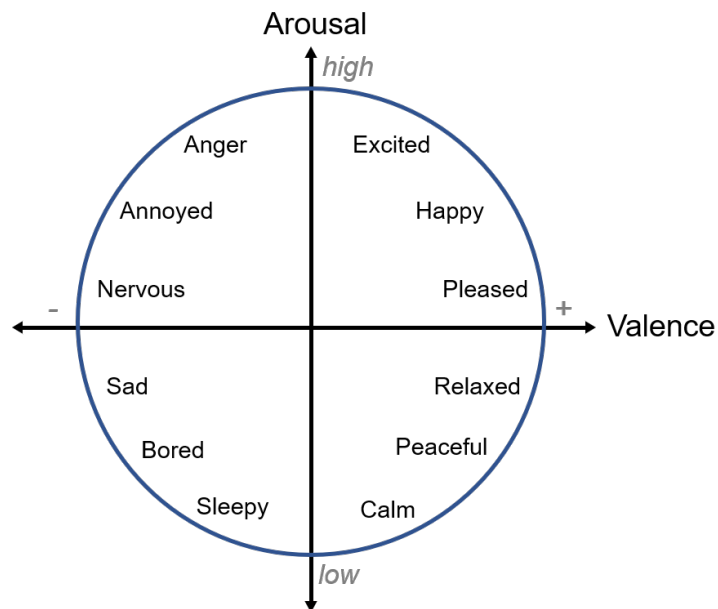


Figure 2.1. Valence-Arousal dimensional emotion model.

Although psychologists argue that the dimensional model is more inclusive and representative of human emotions, the FER research field lacks such datasets, as all datasets follow the categorical emotion model for both static and sequence-based datasets.

2.2. FER Datasets

There are several public datasets for FER, however, the size of these datasets remains a challenge for deep learning models for FER. Generally, deep CNNs require huge amounts of data to learn and recognize patterns from, specifically given that FER is not a simple classification problem due to the differences in how emotion is naturally expressed. In this section, the different types of datasets and the datasets used in this thesis are introduced.

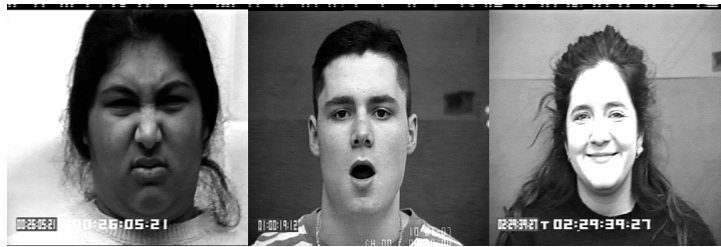
FER datasets can be classified on two levels: a) nature of the collection environment, and b) the type of data or format. Another classification can be done based on the image acquisition method, however, it is not relevant to the scope of this thesis as it only handles camera-acquired images. Other types of image acquisition include infrared images, light-field images, etc. [16].

In terms of the collection environment, FER datasets can generally be classified as either lab-controlled datasets, or in-the-wild datasets.

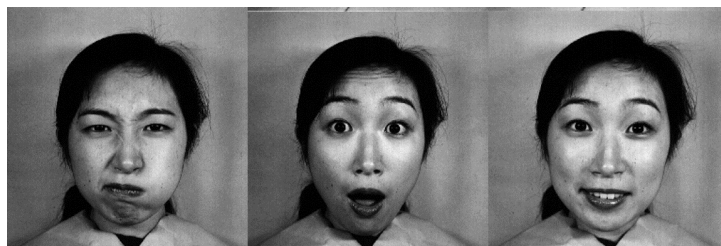
- **Lab-controlled datasets**

As the name suggests, these datasets are typically collected in a walk-in experiment manner in which the participants are asked to exhibit certain emotions [17]. The emotions exhibited in these types of datasets are posed and often exaggerated and they could be an extreme representation of how the subject would normally exhibit this emotion under real-life circumstances. The images in these datasets

are considered “easy” for machine learning as the subjects are facing the cameras, the lighting conditions and the contrast of the images are perfect and uniform across the dataset. Some of the most used datasets of this class are: *Cohn-Kanade (CK+) dataset* [18], JAFFEE [19], shown in Fig. 2.2. One thing to note here is that as these datasets are not complex, they are usually classified with high accuracy with machine learning, however, when employed in real life they do not perform as well as reported, as the system is trained on simple images where the subjects are facing the camera, such that any deviation from the training images limits the performance.



(a) CK+ dataset.

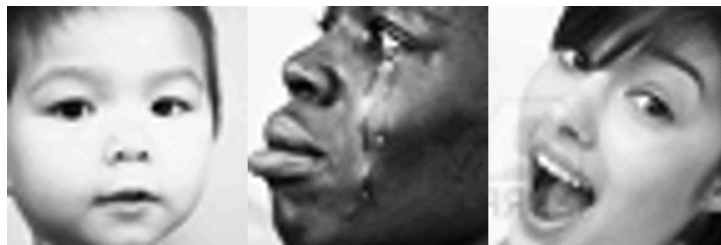


(b) JAFFE dataset.

Figure 2.2. Example of lab-controlled datasets.

- **In-the-Wild datasets**

This type of datasets is different from the other as it is not collected in a laboratory setting. In fact, the images in these datasets are usually crawled from the web, movies, advertisements, etc. As the emotions in these datasets are not collected on demand, the emotions here are more spontaneous and less exaggerated, depicted in Fig. 2.3. Hence, they are more representative of real-life emotions which makes these datasets “complex” and a difficult task for classification using machine learning due to the differences in pose, light, backgrounds, etc. Unlike lab-controlled, usually these datasets only have one image per emotion per subject. Some of the most used in-the-wild sets are the *FER2013 dataset* [20] and the *EmotiW dataset* [21].



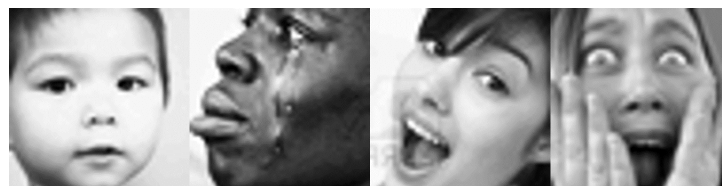
(a) FER2013 dataset.



(b) EmotiW 5.0 dataset.

Figure 2.3. Example of in-the-wild datasets.

The second point of classification is based on the type of data, or the final format of the dataset, that is static and dynamic datasets. Dynamic datasets include videos, frames, or image sequences. Mostly, dynamic datasets are represented in the lab-controlled datasets, as it is more convenient to record videos for each emotion. Static images are usually one image per emotion and is mostly common in the in-the-wild datasets. Dynamic datasets can also be used as static images if the frames containing the peak emotion are used as single images. An example of a dynamic dataset is the CK+ dataset. FER2013 and JAFFE datasets are examples of static datasets. Fig. 2.4 shows the difference between the two formats; in dynamic datasets, the temporal correlation with the change of facial expressions can be observed (Fig. 2.4b).



(a) Samples from a static dataset (FER2013).



(b) Samples from a sequence-based dataset (CK+).

Figure 2.4. Example of dynamic and static datasets.

2.2.1. Datasets Used

In this thesis, a cross-dataset approach is used for static FER in which the two types of datasets are used for training. The reason behind using two types of datasets is that models trained with lab-controlled datasets only perform well for posed emotions but do not maintain the same performance for faces in the wild. On the other hand, training with in-the-wild only is challenging; the models trained with it only do not achieve high accuracy compared to other datasets. As for dynamic FER, only the sequence-based datasets can be used.

- **Facial Expression Recognition 2013 (FER2013) Dataset**

The FER2013 is an in-the-wild static dataset collected using Google image search to crawl images that follow 184 specific emotion keywords. The search also included other keywords to make the dataset as diverse as possible, such as gender, race, and age. The final dataset consists of 35,887 non-posed images of expressions captured in the wild of size 48×48 . It follows a categorical emotion model that classifies the 7-basic human emotions: {anger, disgust, fear, happiness, sadness, surprise, neutral}. Fig. 2.5 shows the distribution of the classes; the dataset is highly imbalanced [20].

- **Extended Cohn-Kanade Dataset (CK+)**

The CK+ dataset is a lab-controlled dynamic dataset containing posed expressions from 123 subjects totalling 593 image sequences of different duration varying from 10 to 60 frames [18]. Although this dataset follows a categorical model, it is different from the FER2013 dataset in terms of the neutral emotion. Here, the emotions are: {*anger, contempt, disgust, fear, happiness, sadness, and surprise*}. As it does

not consider a neutral class per se, the neutral emotion can be extracted from the first few frames of the contempt emotion. Fig. 2.6 shows the class distribution for the CK+ dataset.

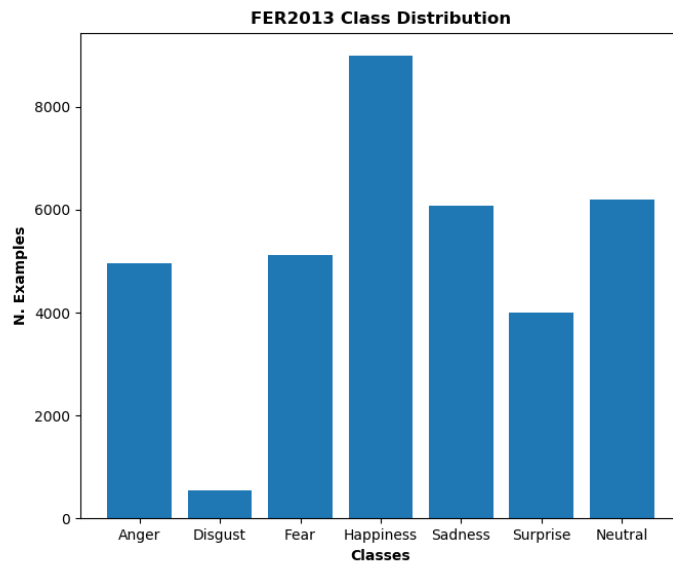


Figure 2.5. Class distribution for the FER2013 dataset.

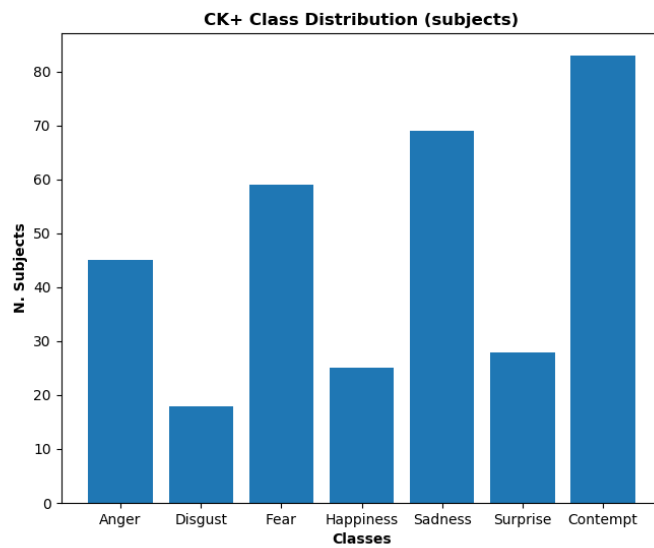


Figure 2.6. Class distribution for the CK+ dataset.

It is important to note that the class distribution in Fig. 2.6 refers to the number of subjects per class. As CK+ is a lab-controlled dataset, there are multiple emotions per subject.

2.3. Artificial Intelligence

2.3.1. Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression [22]. This algorithm aims to find the optimal hyperplane between two classes such that the margin, denoted m_o in Fig. 2.7, between them is maximized. Primal SVM is applicable for linearly separable datasets, as seen in Fig. 2.7; for higher dimension datasets, the data is projected on a new dimension to create a linearly separable format.

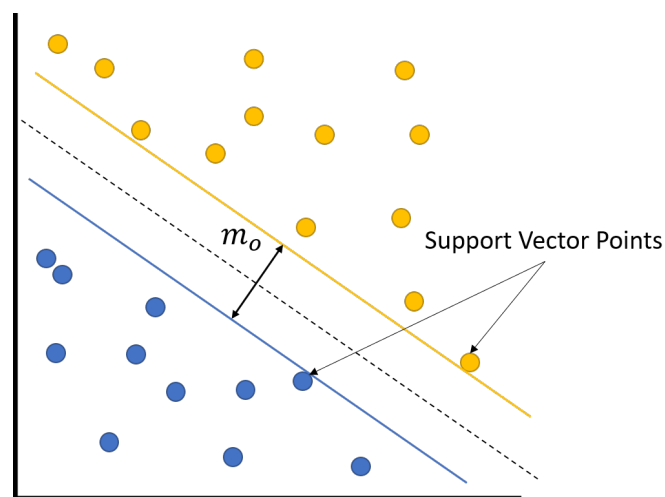


Figure 2.7. Support Vector Machine (SVM).

2.3.2. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a special class of neural networks that have proven to be exceptionally efficient for computer vision applications. CNNs are named so after the mathematical operation employed in the networks, that is convolution. Simply put, CNNs are “neural networks that deploy convolution operation instead of conventional matrix multiplication” [23]. Typically, a CNN consists of three types of layers; convolution, pooling and fully-connected (FC) layers. The combination of convolution and pooling layers are responsible for feature extraction, hence, CNNs eliminate the need for feature extraction or feature engineering [24].

- **Convolution**

The convolution layer is the fundamental component of a CNN. Equation 2.1 demonstrates the mathematical representation of the convolution operation.

$$s(t) = (x * w)(t) \tag{2.1}$$

Convolution is a mathematical process used in many image processing operations. The output of the convolution $s(t)$ reflects how function $x(t)$ is changed by $w(t)$. Due to this, it is used for applying filters to images for many purposes such as edge detection and feature extraction. In convolution, the filter or the convolving function is typically flipped, if it is not, it is referred to as cross-correlation. Although the operation used in CNNs is cross-correlation, it is conventionally referred to as convolution, and we follow the same convention in this thesis [23].

Fig. 2.8 illustrates the convolution operation in the context of CNNs. The original

image is referred to as the input, the convolving argument is the kernel, and the output of the operations is known as the feature map. In the convolution layer, a number of filters (kernels) is convolved with the input from the previous layer to extract some features, such as vertical or horizontal lines (e.g. edges). The values of the kernels are learned through the training of the network, hence, they ‘learn’ the features relevant to the classification.

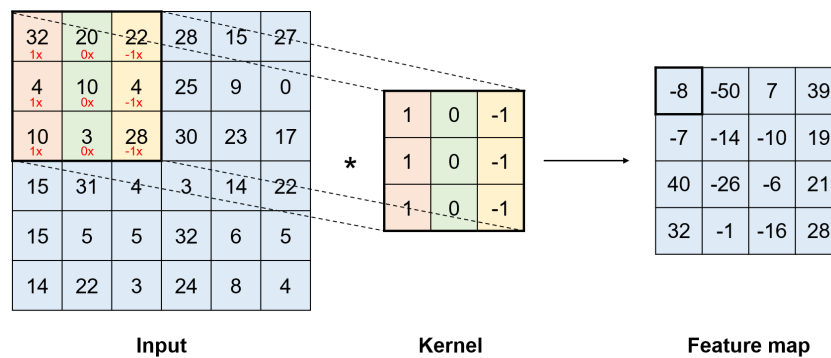


Figure 2.8. 2D convolution.

- **Pooling**

This layer is responsible for reducing the dimensionality of the input, the down-sampling is usually done either by maximum or average pooling using a window of size $p \times p$. In max pooling, the window is moved over the input to produce a down-sampled output in which each pixel is the maximum of the corresponding window in the original input. Intuitively, in average pooling, the pixels in the output correspond to the average of the corresponding sliding window.

- **Fully-connected**

The fully connected layer, sometimes referred to as a Dense layer, is a fully-connected neural network that is used for classification and is typically at the end of the network architecture, preceded by a flattening layer that changes the dimension of the input to 1D.

2.3.3. Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a class of neural networks most suitable for sequential data processing [23]. Unlike CNNs, the input to RNNs is a sequence of data $\{x^{(1)}, x^{(2)}, \dots, x^{(\tau)}\}$ rather than a single data instance x . For example, the input to an RNN could be an image sequence of an exhibited emotion (the change over time) whereas the input to a CNN is just a single image showing the exhibited emotion. Put simply, an RNN iterates through the input sequence and keeps a *state* of the extracted patterns, and then resets the state between sequences [25]. It can be considered as several copies of a module that has a very simple structure as shown in Fig. 2.9.

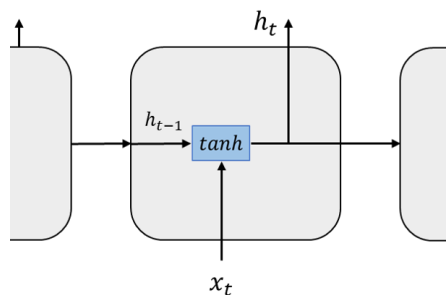


Figure 2.9. A simple RNN module.

For the application of FER, RNNs can be used to extract temporal features of emotion. Since an emotion is expressed by a change in facial expression over a period of time, RNNs could theoretically learn the temporal correlations for better and more robust FER [9].

Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are a subclass of recurrent neural networks that are better suited for long-term dependencies due to their key component which is the cell state C_t . Fig. 2.10 shows the three types of gates in an LSTM module. The forget-gate, f_t , outputs either a 0 or 1 to decide whether to keep or forget the current input. The input-gate i_t updates the current state C_t by adding the new candidate value. Finally, the output-gate o_t propagates the final state.

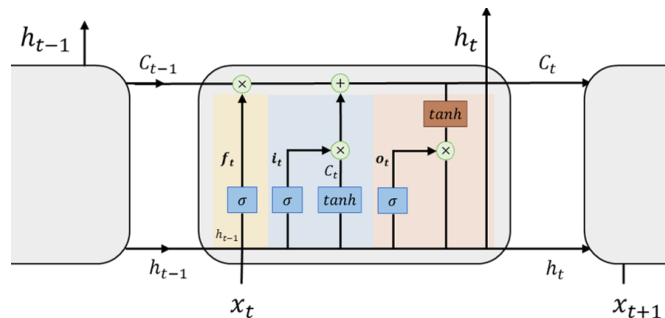


Figure 2.10. An LSTM module.

CHAPTER 3: RELATED WORK

This chapter studies the current state of art with regards to Facial Emotion Recognition (FER). As introduced earlier, FER evolved from machine learning techniques to, the current trend, deep learning. To study the existing works, this section is structured as follows; first, we overview the legacy methods for FER to establish a benchmark for FER, followed by sections 3.2 and 3.3 that cover the recent trends for static and dynamic FER, respectively.

Generally, FER is conducted through multiple steps, illustrated in Fig. 3.1. The first step is face extraction, which is sometimes referred to as facial landmarks extraction. In this step, the main regions of the face are extracted and passed as an input to the feature extraction. In feature extraction, facial regions of interest are extracted such as eyes, eyebrow, etc. and there are two types of features [26]:

- Intransient: this type of features are always present in face; however, they could be deformed due to the facial expression. Examples: eyes, eyebrows, mouth.
- Transient: this type of features occur due to the facial expression, such as wrinkles and bulges around the eyes and mouth.

The last step in a typical FER pipeline is classification; here, the model is trained based on the extracted features and classifies the emotion. Different classification algorithms can be applied here, as to be explained in the following sections.

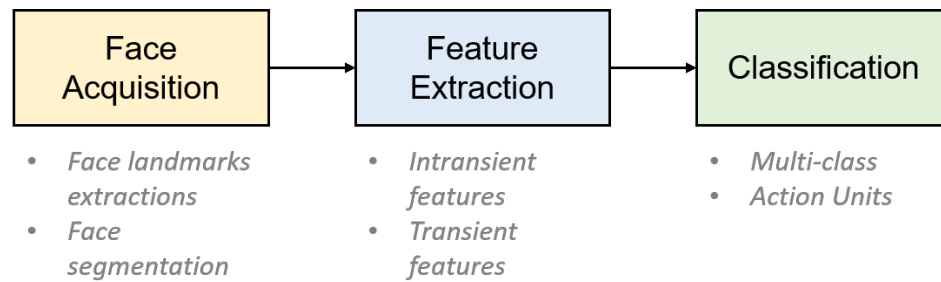


Figure 3.1. Generic FER pipeline.

3.1. Legacy Facial Emotion Recognition

One of the first works in FER is by Paul Ekman and Wallace Friesen in which they adopted the Facial Action Coding System (FACS) and further improved it in 2002 [27]. FACS is a system for describing (coding) facial emotions, it provides a comprehensive description of muscle movements that makeup the facial emotions, these descriptors are known as Action Units (AUs), shown in Fig. 3.2 [8]. Afterwards, researchers started developing machine learning based approaches using hand-crafted features for emotion recognition. Some of the techniques for feature extraction are LBP, HOG, PCA, Gabor filters, etc. As for classifications, the mostly used classifiers are Support Vector Machine (SVM), K-Nearest Neighbours (K-NN) [28]–[31].









Upper Face Action Units (AUs)			
AU 1	AU 2	AU 4	AU 6
Inner Brow Raiser	Outer Brow Raiser	Brow Lowerer	Cheek Raiser
			
Lower Face Action Units (AUs)			
AU 9	AU 12	AU 13	AU 14
Nose Wrinkler	Lip Corner Puller	Cheek Puffer	Dimpler
			

Figure 3.2. FACS Sample [32]

A hybrid pre-processing approach is proposed to improve the classification of emotion using SVM [33]. The authors use a combination of Gabor filters and Histogram of Oriented Gradients (HOG) for feature extraction. When Gabor filters are applied to an image, the edges and textures in the image are extracted by analysing the light changes in the image, hence, extracting the prominent features. The output of the Gabor filters is then processed using HOG. Before classification, the Gabor-HOG processed images are processed using Principal Component Analysis (PCA) to transform the data to a new coordinate system to reduce the number of features.

Finally, the paper compares the classification between SVM and Neural Networks (NNs) which result in the same accuracy of 97.7%. The justification behind the NN not outperforming the SVM is that the number of data samples is very little for a NN that is only 213 images.

3.2. Static FER - Convolutional Neural Networks

There are two types of FER datasets as explained in chapter 2; static images are the most researched with regards to FER, and in some cases, dynamic (sequence-based) are used for static FER by extracting the final video frames that represent the exaggerated emotion. This section reviews some of the recent trends in static FER. Table 3.1 presents a summary of these trends.

With the rapid advancements in deep learning and the outstanding performance exhibited by CNNs for computer vision tasks, neural networks came into play for FER. As the convolution layers in a CNN are the feature extractors, only simple pre-processing is required before training, such as histogram equalization and normalization [9]. However, an important step is detecting the face to reduce the dimensionality of the image and to simplify the task for the classifier by eliminating the regions of the image that are irrelevant to the task. Viola-Johns [34] is the most used face detector in the recent FER works. One recent work proposed a 3-level cascaded face-detector combining Joint-cascade Detection and Alignment (JDA), Deep-CNN and Mixture of Trees (MoT) detectors [35]. For their work, stochastic pooling showed better results in the proposed model, in contrary to the typically used max and average pooling. This approach achieved 55.96% classification accuracy on the SFEW2.0 dataset.

Another approach using CNNs which utilizes transfer learning as it is proven efficient for tasks with limited size of the samples in datasets [36]. The proposed CNN model in [37] uses AlexNet and CNN-M2048 model for FER; they propose a cascaded-finetuning process which shows better results compared to finetuning over a cross-dataset. In their

work, they finetune the model using FER2013, then it is followed by another fine-tuning process using EmotiW yielding an overall classification accuracy of 55.6%.

In [38], a CNN architecture based on the Xception model is proposed for FER. The authors train and test the proposed model on the FER2013 dataset achieving a test accuracy of 66%. In the proposed architecture, the authors replace traditional 2D convolutions with *depth-wise separable convolutions* to reduce the number of trainable parameters, thus, yielding a smaller model. Another step to reduce the number of parameters is eliminating the fully-connected layers usually present at the end of a CNN architecture. Instead, the authors propose using a Global Average Pooling layer along with a convolution layer in which the number of filters is the number of classes. With these modifications introduced, the model achieves a 66% test accuracy on the FER2013 dataset with a relatively small model (~60K trainable parameters). *This model is used in this thesis as a base model. It is further explained in chapter 4.*

The authors in [39] propose a CNN architecture for emotion recognition. They use the CK+ and JAFFE datasets. Along the typical CNN layers, they use residual blocks of 4 convolution layers. They do perform pre-processing such as cropping, intensity normalization, and image normalization. Their proposed model yields high accuracies (0.32% – 0.89% higher than state-of-art).

A very recent research proposes using Attentional CNNs for FER [40], the main idea is to add a spatial transformer module in addition to the typical convolution/pool/FC layers of a CNN. This module is used to transform the feature maps to focus on the most relevant regions of the image, without requiring additional training nor modifications to the optimization [24]. Their approach achieved high accuracy compared to the existing

works with regards to the CK+ (98%) and FER2013 (70%) datasets.

Ming et al. [41] propose a CNN architecture based on the Inception-ResNet architectures for multi-task classification. The model is trained for face authentication, verification, and facial expression recognition to create an end-to-end network named faceLiveNet+. The data used in this paper is CK+ dataset and the OuluCASIA. The final multi-task model slightly outperforms the reported accuracies on the used datasets. They achieve higher accuracy using multi-task model which proves the effectiveness of dynamic weight sharing; compared to single-task FER. The reported accuracies are 99% and 89.6% on the CK+ and OuluCASIA datasets.

This paper improves the work on the faceLiveNet+ and the new models is named 'Dense_faceLiveNet' [42]. One of the modifications they introduce to the model is replacing the residual block with a dense block. Additionally, they use Swish activation which increases the accuracy by 0.9% compared to ReLU [43]. They use transfer learning from easy datasets such as CK+ to complex ones such as FER2013 dataset. Dense_faceLiveNet achieves an accuracy of 69.9% on the FER2013 dataset. Another experiment conducted in this study is transfer learning from FER2013 to another dataset they collected; their results show that transfer learning results in an accuracy of 91.9% which greatly outperforms training on the dataset alone, which yielded an accuracy of ~79%.

Most of the existing FER works perform 7-class classification on the basic emotions. Authors of [44] use a CNN to recognize emotions on 7 classes and on the three classes mentioned. They use the eNTERFACE database and a database they developed for the training, they achieve an accuracy of 75% for the 3-class classification.

3.3. Dynamic FER - Recurrent Neural Networks

As introduced earlier, dynamic FER refers to the classification of emotions from videos. Usually, the type of data for training is either videos (frames) or image sequences. This section overviews the recent trends in dynamic FER, summarised in table 3.2.

A light-field images technique is proposed for dynamic FER. This paper proposes a novel approach for FER [45]. Instead of using conventional cameras, they use light-field cameras which is another type of image acquisition, as used in the IST-EURECOM Light Field Face Database (LFFD). In this approach, they fuse a CNN with LSTM for the classification of dynamic images; and they compare their solution to machine and deep learning techniques. Their approach proved to outperform the existing and state-of-art solution. The input to the LSTM is the features extracted by a pre-trained VGG-face model with 4096 features. Overall, their solution achieves a recognition accuracy of 78.12%.

Chao et al. propose a multimodal system for emotion recognition using facial expressions and audio signals [46]. The authors use a CNN for feature extraction and then feed the extracted features to the LSTM network. For the CNN feature extractor, the network is trained on the EmotiW dataset; and then for LSTM, different outputs from the network are tested to see which CNN output results in better recognition accuracy. Using the output of the 3 lowest layers in the network proved to increase the recognition accuracy to 46.39%.

A hybrid CNN-RNN approach for dynamic emotion recognition is proposed in [47]. The datasets used are the MMI and JAFFE datasets. They first train a CNN for feature extraction where the model is not very complex as the datasets are relatively simple

(i.e. lab controlled). The CNN achieves an accuracy of 76.1%, however, this work also proves that the temporal correlations play a significant role in the classification of emotion, in addition to the spatial features. Hence, they introduce LSTM layers to the model that takes an input the extracted features from the CNN after max pooling. The overall accuracy of the hybrid model is 94.91% and loss of 3.98%.

An interesting work applies multiple deep learning approaches for dynamic emotion recognition [48]. The system proposed in this work is multimodal since it uses facial expressions and audio signals. For audio signals, an SVM with linear kernel is trained and it shows that using audio signals along with FER can increase the classification accuracy by almost 3%.

One approach is proposing a hybrid CNN-RNN approach on the AFEW 6.0 dataset. They fine-tune the VGG-face net with the FER2013 dataset, then, an LSTM is trained with the features extracted by layer FC6 of the VGG-face, and 16 features of 16 frames are stacked for training the LSTM. With this approach, they achieve an accuracy of 45.43% using 128 nodes in the LSTM layer. The other approach adopted in this work is a 3D CNN. The C3D net has 8 convolutions, 5 max-pooling layers and 2 fully connected layers with a softmax output layer. An accuracy of 39% is achieved using the 3D CNN alone, however, with a fused model the accuracy can increase up to 46.5%.

Table 3.1. Static FER Summary

Ref	Pre-processing	Algorithm	Dataset	Accuracy
[33]	Gabor filters HOG with PCA	SVM and NNs	JAFFE	97.7%
[35]	3-level cascaded face-detector using JDA, Deep-CNN and MoT detectors	CNN with stochastic pool	SFEW2.0	55.96%
Ref	Approach	Model	Dataset	Accuracy
[37]	2-level cascaded finetuning	AlexNet CNN-M2048	FER2013 -> EmotiW	55.6%
[38]	DW separable convolutions Residual learning Global Avg Pool	miniXception	FER2013	66%
[39]	Intensity/image normalization	Residual learning CNN	CK+ JAFFE	0.32%+ 0.89%+
[40]	Spatial transformer	Attentional CNNs	CK+ FER2013	98% 70%
[41]	Multitask CNN	Inception-ResNet based	CK+ OuluCASIA	99% 89.6%
[42]	Dense blocks Swish activation	Dense_faceLiveNet	Private dataset FER2013	91.9% 69.9%
[44]	-	CNN	eNTERFACE	75%

Table 3.2. Dynamic FER Summary

Ref	Approach	Model	Dataset	Accuracy
[45]	CNN with LSTM	VGG-face model	LFFD	78.12%
[46]	CNN with LSTM	—	EmotiW	46.39%
[47]	CNN with LSTM	CNN features + LSTM	MMI JAFFE	76.1% 94.91%
[48]	Multimodal (with audio)	CNN + SVM		3%+
	CNN-RNN	VGG-face net	FER2013 and AFEW 6.0	45.43%
	3D CNN	—	AFEW6.0	39%
	Fused model (weighted)	CNN-RNN with 3D CNN		46.5%

CHAPTER 4: STATIC FER USING ARTIFICIAL INTELLIGENCE

This chapter presents FER from still images using artificial intelligence. We cover the concepts used in our experiments, in addition to the results of these experiments.

4.1. introduction

Several approaches and modifications have been proposed in the recent years for different challenges in deep learning such as transfer learning, dense and residual blocks. In this section, these concepts are explained in detail.

4.1.1. Transfer Learning

A natural behaviour of CNNs that has been observed in the literature is that the lower layers in a CNN typically learn low-level features that are similar regardless of the classification task [23]. More precisely, similar features to Gabor filters are learned in the first layers of the architecture. Intuitively, the last layers in any CNN are specific to the classification task; thus, it can be observed that the features become more specific to the task deeper in the network. This nature of CNNs allow transfer-ability of features. Transfer learning is essentially promising to solve the challenge of insufficient data. As deep CNNs require huge amounts of data, it is quite challenging to train such networks with a limited number of examples [37].

In other terms, transfer learning is transferring the knowledge from one task to another. Although theoretically it seems straightforward, in practice, several factors play in the selection of the network to transfer from. For a new classification task T_n with dataset S_n , an already trained model for a task T_o on dataset S_o , transfer learning helps by transferring the knowledge from T_o to T_n where $S_n \neq S_o$ and/or $T_n \neq T_o$. In

practice, this is usually done by only training the FC layers in the pre-trained model since we assume the convolution layers have already learned the features; hence, we *freeze* them. This makes the feasibility of transfer learning depend greatly on similarity between T_o and T_n , the amount of data in S_n , and lastly, the computation resources available [49].

4.1.2. Depth-wise Separable Convolutions

In chapter 2, we introduced convolution and how it is computed for 2D matrices. Although convolution itself is not a costly operation for small networks, when architectures become bigger and deeper, the cost to do the simple multiplication and addition to calculate the convolutions become very expensive and very slow to compute. It also increases the number of trainable parameters as they are the learned kernel values throughout the network. One approach to avoid this challenge is by using a different method of calculating convolutions over volumes known as Depth-wise Separable Convolution [50]. This technique is also used in other applications to change the depth of a volume (i.e. number of channels) without changing the spatial features.

In traditional CNNs, the convolution operation is applied across the channels in the input image, shown in Fig. 4.1. This process is computationally costly and results in large number of trainable parameters in cases of deep networks. For an image of size $D_i \times D_i \times N_c$, the number of multiplication required to convolve it by N_f filters of size $n \times n$ is calculated by equation 4.1.

$$total_{std} = N_f \times (D_o \times D_o) \times (n \times n) \times N_c \quad (4.1)$$

where D_o is the dimension of the output volume and N_c is the number of channels.

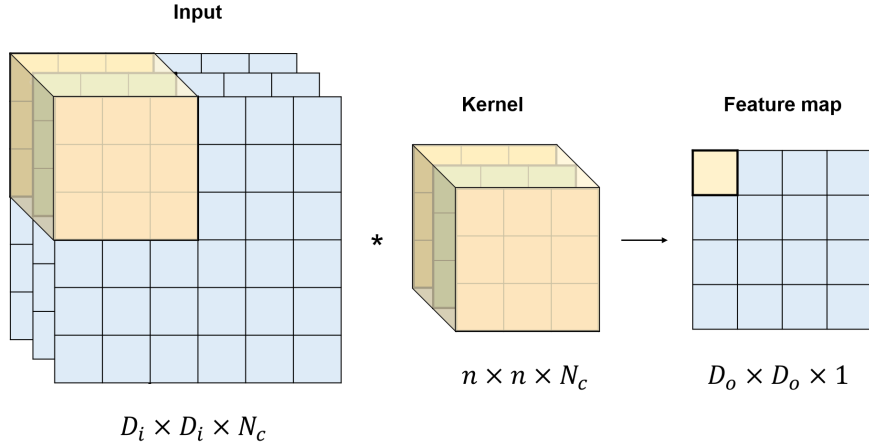


Figure 4.1. Convolution over a volume.

Depth-wise Separable convolutions are used to reduce the cost of convolution and consequently, reduce the number of parameters in CNNs. It is computed in two phases, depth-wise convolution (DC) and point-wise convolution (PC) [38].

- Depth-wise Convolution - Filtering stage:** Unlike standard convolutions, here, the convolution operation is done per channel. Instead of convolving a kernel of $n \times n \times N_f$ over a volume $D_i \times D_i \times N_c$, a kernel of $n \times n \times 1$ is convolved with a single channel of the input $D_i \times D_i \times 1$. The output of this convolution is a 2D matrix of size $D_o \times D_o$. Repeating this for all N_c channels and stacking them results in a volume of $D_o \times D_o \times N_c$. Fig. 4.2 illustrates DC filtering stage.
- Point-wise Convolution - Combination stage:** This stage is computed to create a linear combination of the channels resulted by DC. Here, a $1 \times 1 \times N_c$ filter is convolved with the volume to produce a volume of $D_o \times D_o \times N_c$, while preserving the spatial features; this operation results in an output of shape $D_o \times D_o \times 1$. By performing this convolution N_f times, the output volume becomes of shape $D_o \times D_o \times N_f$. To change the depth of the volume, N_f should be changed.

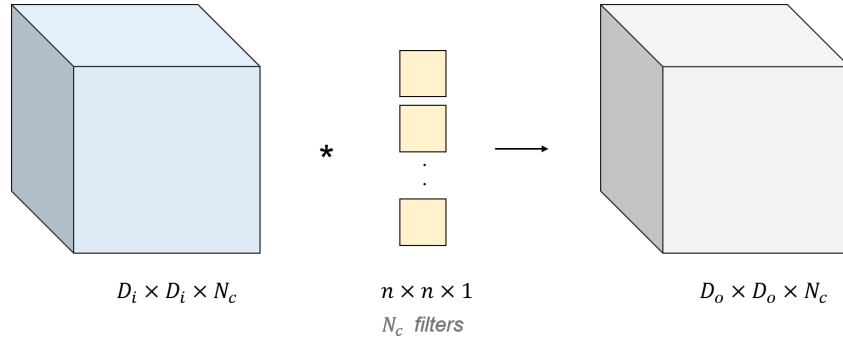


Figure 4.2. DC filtering stage.

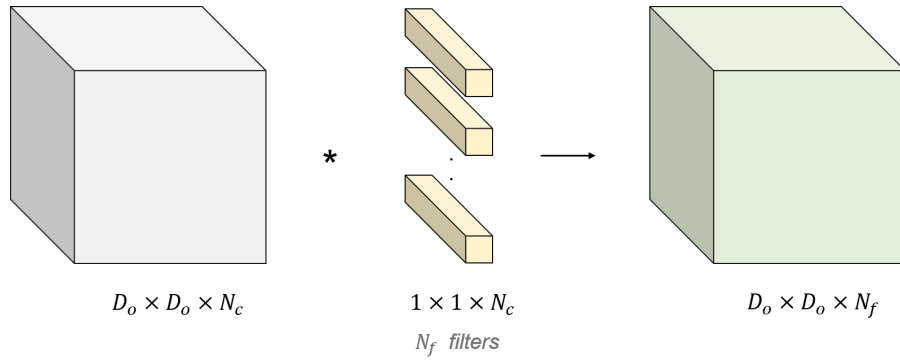


Figure 4.3. PW combination stage.

To compare the performance of standard convolutions against depth-wise separable convolution, the total number of multiplications is calculated by summing the number of multiplications to result in a volume of shape $D_o \times D_o \times N_f$.

The number of multiplications of depth-wise separable convolutions is the sum of the multiplications in the two phases, for the same image $D_i \times D_i \times N_c$:

$$DC = (D_o \times D_o) \times N_c \times (n \times n) \quad (4.2)$$

$$PC = (D_o \times D_o) \times N_f \times N_c \quad (4.3)$$

Using equation 4.4, the total number of multiplications and trainable parameters is thus reduced by $\frac{1}{N_f} + \frac{1}{n^2}$, where N_f is the number of filters (kernels) and n^2 is the size of the filters.

$$\frac{DC + PC}{total_{std}} = \frac{1}{N_f} + \frac{1}{n^2} \quad (4.4)$$

4.1.3. Residual Learning

In residual deep learning, the output of one layer is passed as an input to another layer further in the architecture [51]. This is known as “skip connections” or “shortcut path”. Fig. 4.4 shows an example of skip connections. By doing that, the learned features are the difference between the original and desired feature maps. Residual blocks do not add complexity to the model and do not increase the number of trainable parameters.

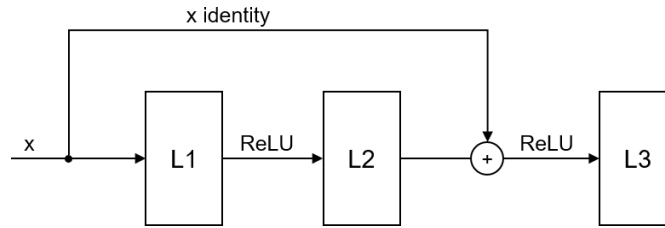


Figure 4.4. Residual block.

For a network of l layers, a non-linear transformation is applied at each layer to calculate the output x_l as shown in equation 4.5. In residual blocks, however, since the output of a layer is passed to another layer deeper in the architecture, the identity function is added, as shown in equation 4.6.

$$x_l = H_l(x_{l-1}) \quad (4.5)$$

$$x_l = H_l(x_{l-1}) + x_{l-1} \quad (4.6)$$

4.1.4. Dense Block

Another modification to traditional CNNs to overcome the vanishing gradient problem is introduced in DenseNets [52]. The dense block architecture allows for maximum gradient flow, since every layer is connected to the following ones directly, as shown in Fig. 4.5.

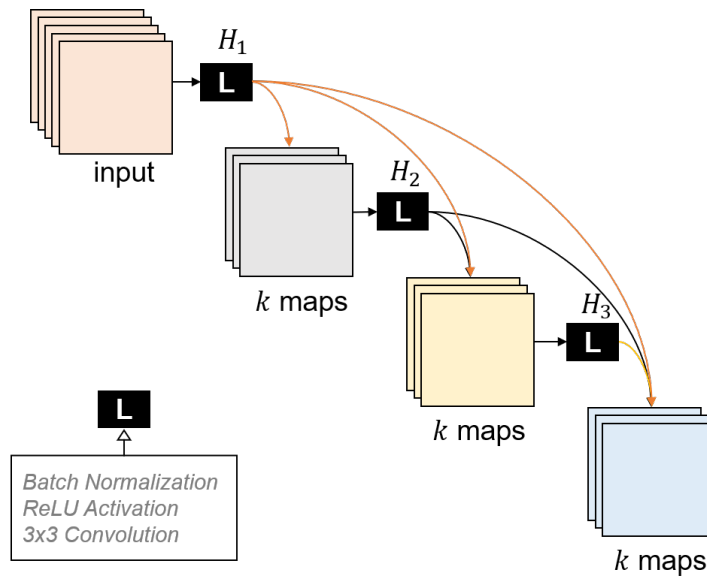


Figure 4.5. Dense block.

Dense blocks concatenate the feature maps rather than adding them, unlike residual blocks. The output x_l of a layer l is calculated by equation 4.7.

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]) \quad (4.7)$$

Since every layer in a dense block has connection with its preceding layers, it has access to the previous feature maps. In their paper, Huang et al. refer to the features maps as a state of the network, and every layer adds its k maps to this state; this represents the "collective knowledge" of the network [52]. The hyperparameter k is referred to as the *growth rate*, and it indicates how much new knowledge each layer adds to the existing collective knowledge. The output of each layer in the dense block is $k_0 + k \times (l - 1)$ feature maps, where k_0 is the input layer channels.

4.2. Methodology

4.2.1. Pre-processing and Dataset Preparation

As demonstrated in Fig. 4.6, preparing the dataset for training is done in two stages, offline and online. Offline is done before training, and online is done as the data is passed to the training model.

Offline data preparation mostly handles the relabeling of the data. Since the emotion model followed in this thesis is a modified categorical model, the emotion classes here are: *{negative, neutral, and positive}*. For that reason, the data classes are relabeled as shown in table 4.1. It must be noted that the *surprise* emotion is not included in the redefined classes since it does not fall under any of the three classes in this emotion model. It is worth mentioning that due to the original sizes of the two datasets, the

cross-dataset consists of 90% FER2013 images and 10% CK+ images.

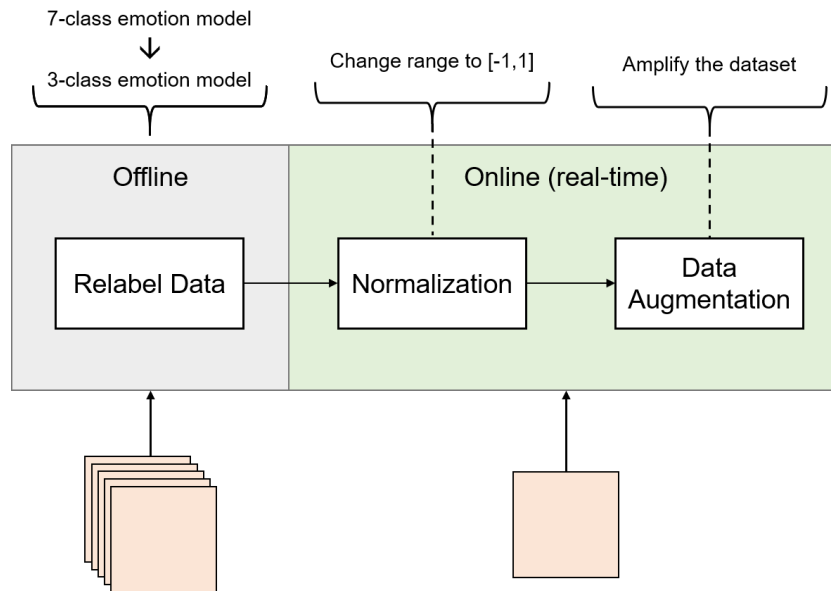


Figure 4.6. Data preparation pipeline.

Table 4.1. Relabelled Classes

New Label	Old Label(s)
Negative	anger, fear, disgust, sadness
Neutral	neutral
Positive	happiness (smiling)

Given that the original classes are relabeled by grouping the negative emotions, this means that there are more examples from the negative class. This consequently results in a new class distribution as shown in Fig. 4.7.

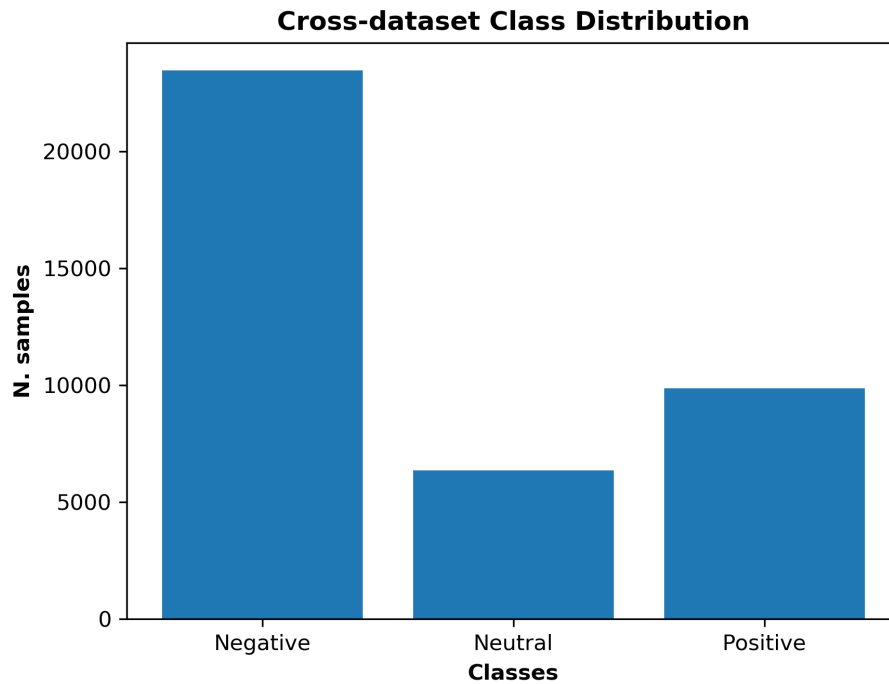


Figure 4.7. Class distribution for relabeled data.

Another step in offline data preparation is pre-processing. It is already established that CNNs do not require intensive pre-processing, however, some image processing operation are reported to improve the performance of the models. Normalization is performed to improve the performance of the CNN, however, it is done *online*. Since it is applied on each image separately, it can be processed as the data is passed to the training model.

Data Imabalnce

As shown in Fig. 4.7, the final dataset is highly imbalanced which can cause the model learn the patterns from the dominant class and hence overfit to this class because it has the highest number of samples. There are several ways to handle data imbalance such as oversampling/undersampling the dataset, collecting more data samples for the

minority class, etc. One way to handle imbalanced data is by assigning weights to each class, corresponding to its number of samples, and incorporating the weights in the loss function. Intuitively, the minority classes will be assigned higher weights to have more impact on the loss function.

Data Augmentation

As deep CNNs require huge amounts of data, data augmentation is used to amplify the size of the dataset without the need to collect new data. *Data Augmentation* is simply applying some image processing operations to the images such that the resulting images are still the same to a human, however, are considered new samples to the classifier. The operations applied are rotation, width and height shift, zoom, and horizontal flip, illustrated in table 4.2. Fig. 4.8 shows a sample of the augmented images.

It is worth noting that gray-scale images are used for training. Since RGB images require more channels in the input layer of the architecture, they add an unneeded overhead as color is not relevant for emotion classification.

Table 4.2. Data Augmentation Operations

Operation	Value
Rotation range	10
Width shift range	0.1
Height shift range	0.1
Zoom range	0.1
Horizontal flip	True



Figure 4.8. Data augmentation.

4.2.2. Machine Learning

Since most of existing works follow a 7-class emotion model, we use machine learning algorithms to create a benchmark for the 3-class emotion model. For that, MATLAB's *Classification Learner Toolbox* is used to train several machine learning classifiers such as SVM, K-Nearest Neighbor (KNN), etc. The input data to these classifiers are the label of the image along with the features which are the pixel values.

The second method using machine learning is to use SVM for the classification together with a CNN. Essentially, the features are extracted from the convolution layers in the network. Instead of classifying using fully connected layers, an SVM classifier is trained. The features for this method can be extracted from different layers in the architecture, hence, can be of different lengths.

4.2.3. Transfer Learning

Transfer learning and its advantages are introduced in chapter 2; to leverage the pre-trained models, a set of dense layers are added to the model for the new classification task. As we expect the convolution layers to have previously learned features, we freeze them. To further customize the model to FER task, it is possible to unfreeze some of the convolution layers to relearn task-specific features from the training datasets, however, this might be very resource-intensive, as the pre-trained networks are very deep.

In this work, we freeze the convolution layers of two pre-trained networks, then we add and fine-tune a set of fully-connected layers. Table 4.3 shows a summary of the used network architectures.

- VGG-16 [53]: a CNN architecture that is trained on the ImageNet dataset and achieves an accuracy of 92.7%. VGG-16 improves AlexNet by using smaller filter sizes such as 3×3 compared to 5×5 and 11×11 in the latter.
- ResNet-50 [51]: Another architecture trained on the ImageNet challenge with an accuracy of 95.51%. It efficiently used skip connections to improve the performance of CNNs by facilitating the training of deep networks.

Table 4.3. Pre-trained CNN Architectures

Model	# Parameters	Accuracy
VGG-16	~138 million	92.7%
ResNet-50	~25.5 million	95.51%

4.2.4. CNN Architectures

In chapter 4 we introduced some CNN architecture features that facilitate training deeper models by avoiding the vanishing gradient problem. In this section, We train and fine-tune an existing model as a *base model* and propose modifications that improve the performance of the this model.

Base Model

The base model proposed in this thesis is inspired by the mini Xception model proposed in [38] based on the Xception model [50]. The model uses combinations of residual blocks and depth-wise separable convolutions to reduce the number of trainable parameters as shown in Fig. 4.9. The total number of trainable parameters is approximately 52K parameter.

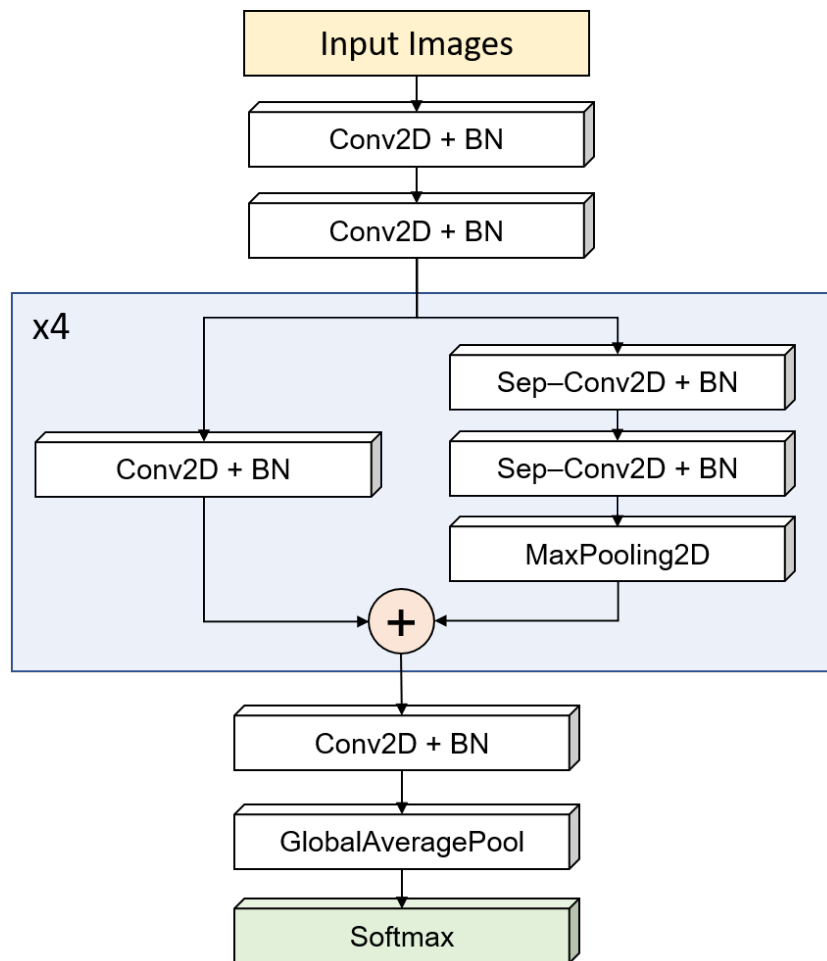


Figure 4.9. Base CNN model.

Proposed CNN

As mentioned previously, several experiments were conducted to improve the performance of the base model. Dense blocks with configuration presented in Fig. 4.10 are introduced to the model, to produce a final CNN model as shown in Fig. 4.11. In addition to that, the number of layers and kernels per layer are modified empirically.

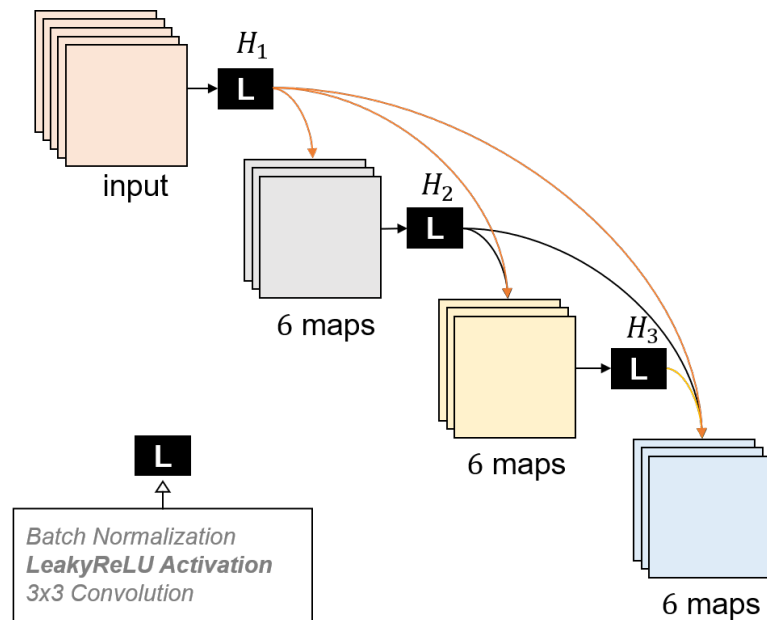


Figure 4.10. Dense block configuration.

The proposed dense block replaces the activation function ReLU by LeakyReLU as, with experimentation, it was evident that it leads to faster convergence. Additionally, the number of feature maps per layer in the dense block is changed to six 3×3 feature maps. The dense connection is repeated 4 times per dense block.

Fig. 4.11 illustrates a high-level view of the proposed CNN. It is important to note that using depth-wise separable convolutions (equation 4.4), the overall number of train-

able parameters is reduced from approximately 420K to 94K trainable parameter. The residual block in this model is similar to the one in the base model. Other modifications are done to the number of layers and filters in some layers.

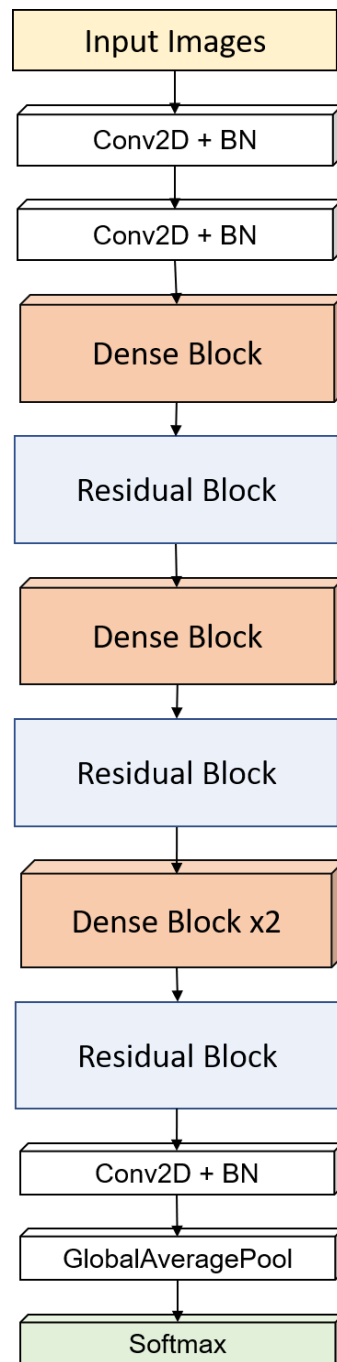


Figure 4.11. Proposed CNN model.

Table 4.4 summarises the configuration of the model, it presents the number of filters per layer, and the output shape of each layer.

Table 4.4. Configurations of Proposed CNN Architecture

Layer	N. filters	Output shape
Input	0	$48 \times 8 \times 1$
Convolution	8	$48 \times 48 \times 8$
Convolution	16	$48 \times 48 \times 16$
Dense Block	6 (x4)	$44 \times 44 \times 40$
Residual Block	32	$22 \times 22 \times 32$
Dense Block	6 (x4)	$22 \times 22 \times 56$
Residual Block	64	$11 \times 11 \times 64$
Dense Block	6 (x4)	$11 \times 11 \times 88$
Dense Block	6 (x4)	$11 \times 11 \times 112$
Residual Block	64	$6 \times 6 \times 64$
Convolution	3	$6 \times 6 \times 3$
Global Avg Pool	0	3
Softmax	0	3

In the following section, we present the experiments conducted to train and evaluate the performance of the different models adopting residual and dense blocks in addition to transfer learning for static FER. Although we focus on 3-class emotion classification, we also experiment with 7-class emotion as a comparison with the existing works.

4.3. Experiments and Results

As mentioned previously, a toolbox in MATLAB is used to train several machine learning classifiers. All experiments discussed in this chapter are conducted and tested on a computer with specifications presented in table 4.5.

Table 4.5. Environment Specifications

Specification	Version
Operating System	Windows 10 – version (1803)
Processor	Intel i7-8850 – 16 GB RAM
Python version	3.6
Tensorflow-gpu	1.12
GPU	NVIDIA GeForce 940MX

Python packages were used for the development, mainly, Keras with Tensorflow backend. Moreover, all CNN models are trained with *categorical cross-entropy* for loss and *ADAM Optimizer*, and L2 Regularization is used to prevent overfitting. In machine learning, a loss function is a measure of how different the prediction from the true label is. It dictates how much and in which direction the weights should change during the back-propagation [25].

Categorical Cross-Entropy Loss is a loss function that computes the average difference between the true and the predicted probability distributions for a task of n-class classification. Equation 4.8 demonstrates the mathematical formula of the categorical cross-entropy loss where N is the number of classes, y_c is the true label for the class,

and \hat{y}_c is the CNN output score for class c .

$$CE = \sum_{c=1}^N y_c \times \log(\hat{y}_c) \quad (4.8)$$

In section 4.2, we introduced the concept of incorporating class weights in the loss function to handle imbalanced datasets. Thus, the loss function is simply modified by introducing the term w_c which denotes the weight for class c , given in equation 4.9, with weights shown in table 4.6.

$$CE = \sum_{c=1}^N w_c \times y_c \times \log(\hat{y}_c) \quad (4.9)$$

Table 4.6. Class Weights

Class	Negative	Neutral	Positive
N. Examples	23,469	6,358	9,855
Weight	0.5671	2.0485	1.3363

4.3.1. Machine Learning

Benchmark

MATLAB Classification Learner trains different machine learning classifiers. The data input to the classifiers are of 2,304 features, with a corresponding class label. Table 4.7 summarizes the trained classifiers and their test accuracy. The highest accuracy is 62% achieved by Medium Gaussian SVM, nevertheless, Quadratic and Cubic SVM classifiers also achieve comparable accuracy.

Table 4.7. Results of Machine Learning Classifiers

Classifier	Accuracy (%)
Linear SVM	57.8
Quadratic SVM	62.2
Cubic SVM	62.3
Medium Gaussian SVM	62.5
Bagged Trees Ensemble	59.9

SVM Classifier-CNN Features

The other hybrid approach is to train an SVM classifiers with the features extracted from the CNN. It is expected that this approach performs better compared to learning from just the pixels as features; this is because the CNN features are expected to represent the most relevant information to the classification. The Classification Learner toolbox allows for the use of PCA to reduce the number of features such that the final feature vector is only 22 features out of 128.

Table 4.8 summarizes the results of this approach. The highest test accuracy was achieved by Linear SVM and Quadratic SVM classifiers. With PCA, Quadratic SVM achieved and accuracy of 74.4%. Furthermore, Linear SVM achieved the highest accuracy for this approach with an accuracy of 76.9% without using PCA.

Table 4.8. Results of Machine Learning Classifiers with CNN Features

Classifier	Accuracy (PCA) (%)	Accuracy (%)
Linear SVM	73.1	<u>76.9</u>
Quadratic SVM	74.4	73.1
Cubic SVM	65.4	69.4
Fine Gaussian SVM	73.5	73.5
Medium Gaussian SVM	73.5	73.5

4.3.2. Transfer Learning

The pre-trained models mentioned in chapter 4 have implementations in Keras. For the experiments, we freeze the layers of the models to preserve the learned weights, then, we add 3 fully connected layers with dropouts to prevent overfitting. We train the modified VGG-16 and ResNet-50 for a total of 5 experiments each; for each dataset, both models are trained, and then we train them on the cross-dataset to measure the performance.

The VGG-16 network is implemented in Keras, three fully-connected layers are added on top of the model. All the convolution layers in the model are frozen. The FC configurations are attached to appendix A.

Table 4.9 summarizes the new layers and the number of trainable parameters in each layer for the VGG-16 model on the used dataset (cross-dataset).

Table 4.9. VGG-16 Architecture for Transfer Learning

Layer (type)	# Parameters
vgg_base (Model)	14714688
dense_4 (Dense)	262656
dropout_3 (Dropout)	0
dense_5 (Dense)	262656
dropout_4 (Dropout)	0
dense_6 (Dense)	1539
Total parameters	15,241,539
Trainable parameters	526,851
Non-trainable parameters	14,714,688

ResNet-50 model was also fine-tuned with the same parameters mentioned, however, the final model is significantly bigger than the VGG-16. A summary of the modified ResNet-50 model is given in table 4.10.

Table 4.10. ResNet Architecture for Transfer Learning

Layer (type)	# Parameters
resnet_base (Model)	23587712
dense_4 (Dense)	4194816
dropout_3 (Dropout)	0
dense_5 (Dense)	262656
dropout_4 (Dropout)	0
dense_6 (Dense)	1539
Total parameters	28,046,723
Trainable parameters	4,459,011
Non-trainable parameters	23,587,712

Transfer Learning Results

Fig. 4.12 shows the resulting accuracies of transfer learning from the two models on the two datasets and the cross-dataset. Extended results are available in appendix B.

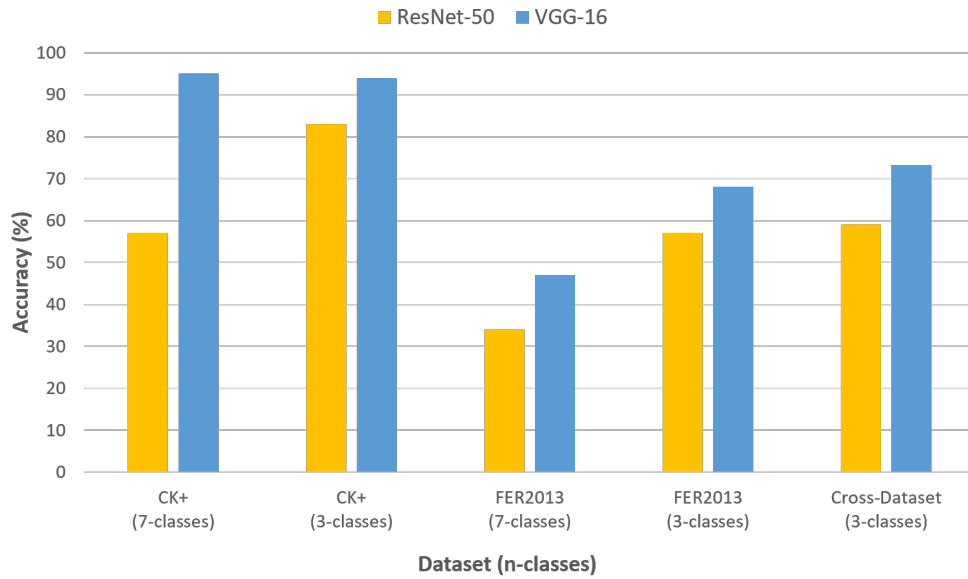


Figure 4.12. Transfer learning results.

Generally, VGG-16 performed better for FER from FER2013 and CK+ datasets, as well as for the cross-dataset. From the comparison, it can be observed that the classification on CK+ dataset is higher with both networks, as anticipated. For the cross-dataset, VGG-16 achieves an accuracy of 73.2% compared to 59% with ResNet-50.

Additionally, the classification of emotion for 3-classes is done with higher accuracy compared to 7-classes for both models. This can be explained by looking into the data examples, typically, the inter-class variance between some of the negative classes is not significant in 7-class classification. Hence, several instances of *fear* class, for instance, are classified as *anger*. Additionally, restructuring the dataset to three classes consequently increases the number of images per class, in comparison to seven classes.

Another observation is that using a cross-dataset approach predictably improved the performance of the classification compared to only using FER2013. Evidently, it increased the classification accuracy from 68% to 73.2%.

4.3.3. Proposed CNN

This section reports the results of the base model and the modifications in the proposed model for 3-class FER. Both models were trained for 100 epochs with *ReduceLROnPlateau* callback. This callback monitors the validation loss and reduces the learning rate by a factor of 0.1 if it does not improve for 12 epochs. The dataset is split 80%-10%-10% as train-test-validation, with a batch size of 128.

In addition to the classification accuracy, other evaluation metrics are computed. For a class x , *precision* measures the ratio of the correctly classified x samples, to the number of samples classified as x , it is calculated by equation 4.10. *Recall* calculates the ratio of the correctly classified x samples to the total number of x samples. *Recall* is calculated by equation 4.11. Finally, *F1-score* reflects the overall performance of the model by combining both the precision and recall. It is calculated by 4.12.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (4.10)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (4.11)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4.12)$$

Base Model

The model is trained on the cross-dataset after re-structuring and re-labeling the datasets to map the three emotions. The model was trained for 100 epochs with a batch size of 128. Fig. 4.13 shows the confusion matrix for the 3-class classification on the cross-dataset.

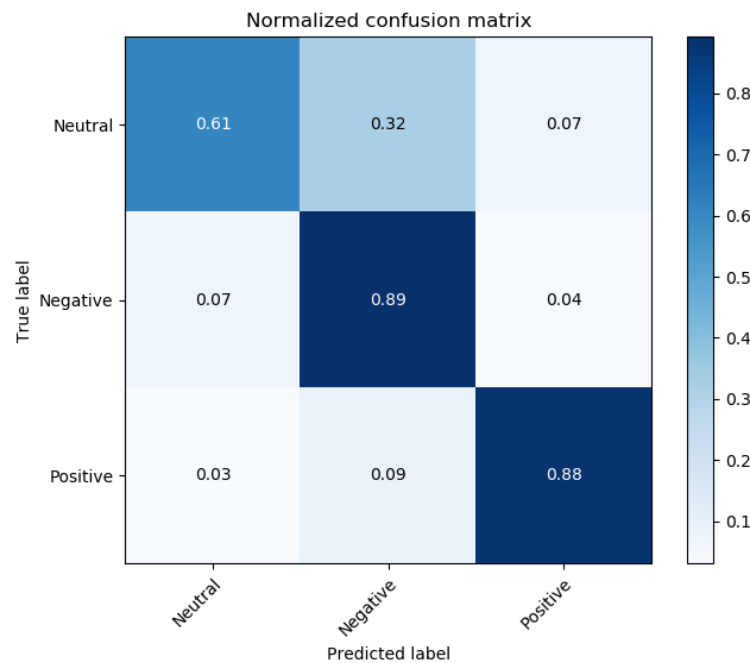


Figure 4.13. Confusion matrix for 3-class classification using base CNN model.

Overall, this model achieves a test accuracy of 84%. From the confusion matrix, it can be observed that the recognition rates for positive and negative are considerably higher than neutral emotion. It can be observed that 32% of the *neutral* images were classified as *negative*; this is because in some cases, the neutral emotion can be interpreted as a

negative emotion; this was present on the training data and hence the low recognition rate.

Table 4.11 summarizes the classification report of the base model. As shown, the *neutral* class had the lowest precision, recall and f1-score. The f1-score of the neutral class is 0.62. As for the *negative* and *positive* classes, they both achieved similar metrics.

Table 4.11. Base Model Classification Report

Class	Precision	Recall	f1-score
Neutral	0.64	0.61	0.62
Negative	0.89	0.89	0.89
Positive	0.87	0.88	0.87

Proposed CNN

The modifications introduced to the base model, such as adding dense blocks, have proved to improve the performance of the model. The overall accuracy of the model increased to 86.5%. In Fig. 4.14, the confusion matrix of this model is illustrated. We can observe that, although the accuracy of classification increased per class, the *neutral* emotion classification is mis-classified as *negative* for 28% of the images.

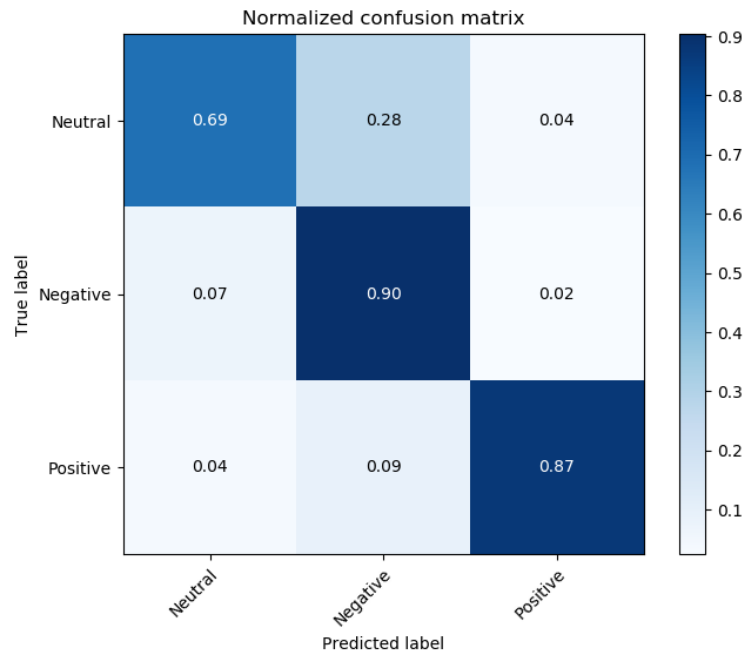


Figure 4.14. Confusion matrix for 3-class classification using proposed CNN model.

Table 4.12 summarizes the evaluation metrics of the proposed model. From the results, we can notice that the proposed model increased the recall to 0.69; this implies a better ability at recognizing *neutral* emotion compared to the base model. Similarly, the performance measures increased in comparison to the base model. We can notice that the precision increased by 0.04 for the *positive* class. However, the base model had a 0.1 higher recall for the *positive* class.

Table 4.12. Proposed Model Classification Report

Class	Precision	Recall	f1-score
Neutral	0.65	0.69	0.67
Negative	0.90	0.90	0.90
Positive	0.91	0.87	0.89

CHAPTER 5: DYNAMIC FACIAL EMOTION RECOGNITION USING ARTIFICIAL INTELLIGENCE

It is established that leveraging the spatio-temporal features might possibly improve the performance of dynamic FER. In this chapter, we introduce the different approaches to dynamic FER using Artificial Intelligence.

5.1. Introduction

5.1.1. Proposed CNN model for Dynamic FER

As the proposed CNN model is relatively small in size since it uses depth-wise separable convolution, in real-time, classification using this model can be a quick process (i.e. near instantaneous). Hence, it is used for video FER by processing real-time camera input. To speedup the deployment, every frame is skipped to minimize the processing time. The face is detected using a CNN proposed in [54] which achieves an accuracy of 99% for face detection. This model outperforms CV2's Cascade detector as the former detects faces in the wild, and from different angles rather than just the front.

5.1.2. A hybrid CNN-RNN Approach

Another potential approach for dynamic FER is leveraging the trained CNN model, in addition to using a recurrent network to integrate temporal correlation in the recognition. Fundamentally, this is done by using the CNN to extract features from each sample in the image sequences; then, the extracted feature-sequences are passed to an RNN, as illustrated in Fig. 5.1.

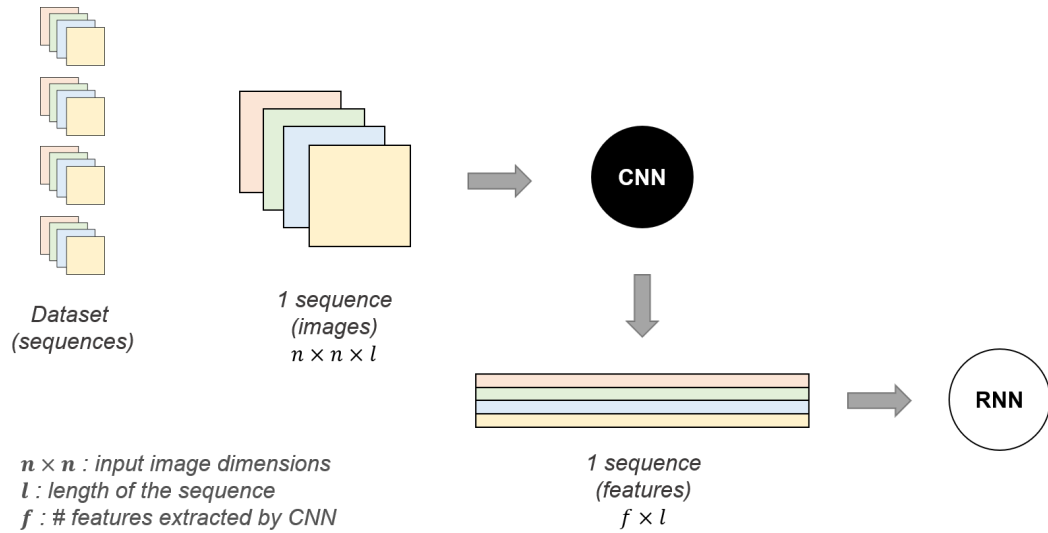


Figure 5.1. System diagram of hybrid CNN-RNN approach.

5.2. Methodology

5.2.1. Dataset Preparation

The dataset used for dynamic FER is the CK+ dataset, introduced in chapter 2, since it is a sequence based dataset. One challenge is length of the sequence; the CK+ dataset is variable in terms of the number of images per emotion. The number of samples per sequence vary from 4 to 77. This makes it quite challenging from two aspects: 1) if the RNN takes fixed-length sequences, and 2) the significantly different amount of temporal information that can be interpreted from a 4-frame sequence compared to a 77-frame sequence.

To overcome this, the number of samples in a sequence must be reconstructed to a sequence of a given fixed-length l . This is done by down-sampling sequences longer than l , and by up-sampling sequences less than l , as shown in algorithm 1.

Algorithm 1: Reconstruct Image Sequences

Result: A sequence of length l

Input: sequences, l

```
for sequence in sequences do
     $n = \text{length}(\text{sequence})$  ;
    if  $n > l$  then
         $s = \text{int}(\text{ceil}(n / l))$  ;
         $\text{counter} = 0$ ;
        while  $\text{counter} < l$  do
            skip every  $s$  sample ;
        end
    else
        if  $n < l$  then
             $r = \text{int}(\text{ceil}(l / n))$  ;
             $\text{counter} = 0$  ;
            while  $\text{counter} < r$  do
                repeat every sample  $r$  times ;
            end
        else
            do not reconstruct ;
        end
    end
end
```

5.2.2. Feature Extraction

After reconstructing the full dataset to a new dataset of n sequences of l frames, the features can now be extracted using the CNN. In our approach, we use the features extracted by the last convolution layer of shape $6 \times 6 \times 3$, which is 108 features per frame. This way, the dataset is of size $327 \times 108 \times 40$; for each sequence of the 327, there are 108 features per frame (40 frames). Output from other layers can also be used.

5.3. Experiments and Results

5.3.1. Proposed CNN for Dynamic FER

To test the proposed CNN model for dynamic FER, we test the classification on a sample of CK+ videos. Another experiment we conducted for dynamic FER is a real-life experiment detailed later in this chapter (see section 5.4).

Overall, the classifier classified 84% of the frames correctly. For the negative emotion videos, the model correctly classified all the frames. By looking into the predictions, we notice that the model correctly classified 100% of the frames in which the emotion is peaking, however, some of the initial frames are mis-classified.

A sample of the mis-classified frames is shown in Fig. 5.2. As demonstrated in the evaluation of the model in chapter 4, the model originally classified 28% of *neutral* samples as *negative*. In this experiment, almost all of the mis-classification are of *neutral* emotion interpreted as *negative*.

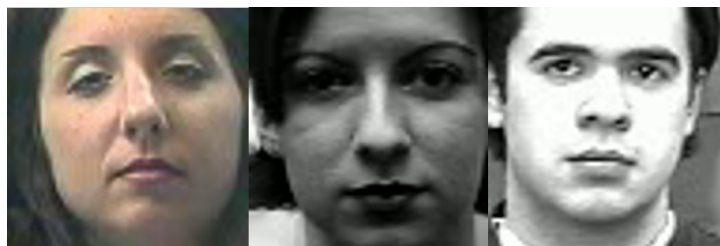


Figure 5.2. A sample of mis-classified frames.

For the middle and right samples, it can be noticed that the intensity is lower in the eyebrows region, which could possibly be interpreted as a frown by the model. Moreover, the left sample's *neutral* face shows a downward pull in the mouth region which might have lead to the sample getting classified as *negative*.

It is worth mentioning that the CNN used for face detection is able to detect multiple faces in a frame, hence, the real-time classification also be used for group FER. Algorithm 2 explains the steps to detect facial emotion in videos. In the algorithm, *video source* can either be a video file, or the camera input. The second case is used for real-time application.

Algorithm 2: Video FER Using Static CNN Model

Result: Real-time emotion classification

Input: video source: vid, CNN: model

```
while true do
    vid  $\leftarrow$  frame
    temp = resize(frame) ;
    temp = toGrayScale(temp) ;
    faces  $\leftarrow$  get_faces(temp)
    for face in faces do
        | classification = model(face) ;
    end
end
```

5.3.2. Hybrid CNN-RNN

As we explained previously, leveraging spatio-temporal correlations could possibly improve FER. With that goal, we train an RNN by using LSTM layers along with fully-connected layers on the reconstructed dataset. Table 5.1 summarizes the architecture of the LSTM model. The data was split 80-20% for training and testing, respectively.

Table 5.1. LSTM Model Parameters

Model Layers	Parameters
LSTM layer	108 units
Dropout layer	25%
Dense layer	256 units
Dropout layer	25%
Dense layer	3 units

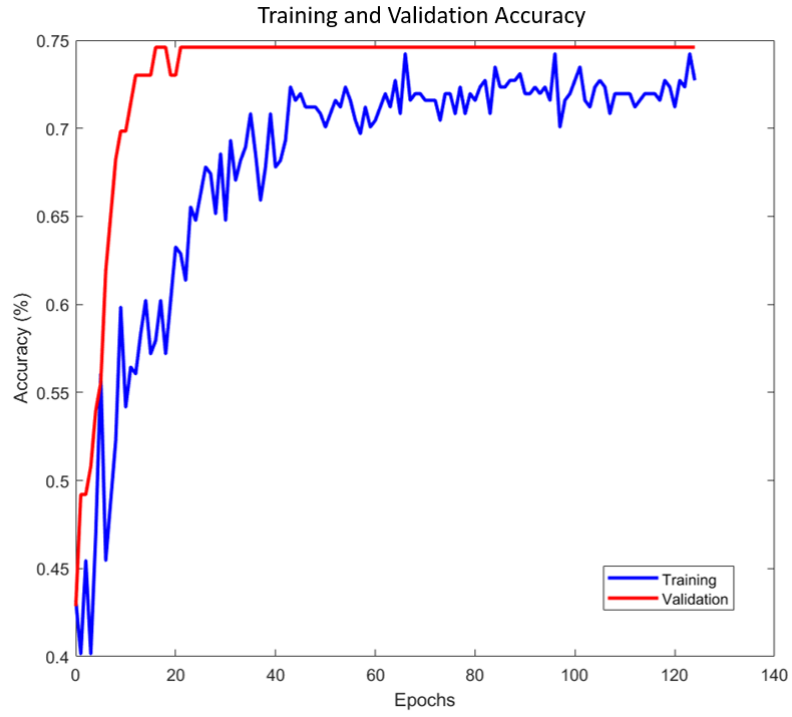
The model is trained for 300 epochs with ADAM optimizer and categorical cross-entropy. Moreover, EarlyStopping is used with a patience of 50. It monitors the validation loss and stops if it does not change for 50 epochs. Fig. 5.3 show the accuracy and loss plots of the hybrid model. The overall accuracy obtained by the model is 74.6%. This low accuracy can be justified by several points:

1. The model is trained on the CK+ dataset which only consists of 327 sequences. This is a very small dataset and is considered insufficient for deep learning.
2. Although the dataset is made up of image sequences, the sequences are of different lengths and are not actually a full consecutive sequence, rather frames sampled

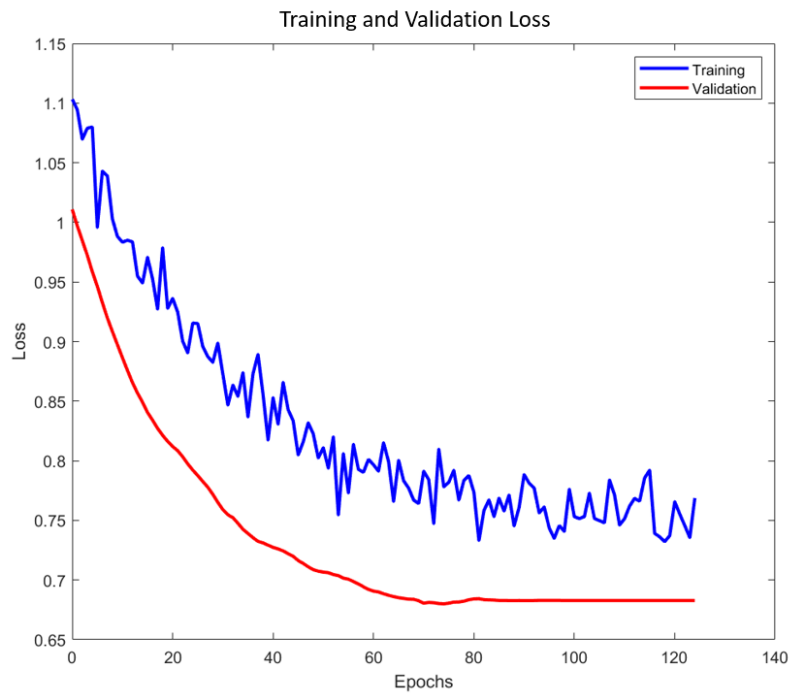
from a sequence. This could affect the FER as the sequence does not exactly reflect the natural transition in facial emotion. A variable-length LSTM could possibly perform better.

3. The pre-processing done to unify the number of frames by either up-sampling or down-sampling could have altered the transition in the facial emotion as some frames were dropped or repeated.

It can be observed in Fig. 5.3 that the model's validation loss and accuracy are always higher than the training's. This is possibly due to the use of regularization techniques such as dropout. In drop-out, a number of neurons are disabled through the training process to prevent overfitting. However, all neurons are active during validation, hence, the model is better at extracting those features.



(a) Accuracy plot.



(b) Loss plot.

Figure 5.3. LSTM training VS validation plots.

5.4. Real-life Experiment: Training Session Evaluation

One application of FER is in the education process, there are various examples of FER deployed in classrooms to evaluate the understanding and attention of students [55]–[57]. For this thesis, we conducted an experiment to evaluate a training session conducted by a local company. The experiment aimed to evaluate the attendees emotional reaction to a set of questions regarding the training and the trainer, to measure and assess the overall effectiveness of the training. This is achieved by analysing video-recordings provided by the training company.

The training session was an introduction to a software called PowerBI that the trainees are expected to use for their jobs, hence, the interviews included some technical questions to evaluate the participants' understanding.

5.4.1. Participants

The experiment is considered racially diverse as it consisted of six participants with different racial facial features (East Asians, Arabs, Caucasians, Africans). Each participant was asked a set of questions pre-training and post training. Moreover, there were equal number of female and male participants. It is important to have this diversity in race and gender to test the model's ability to generalize.

5.4.2. Questionnaire

The questions were set by the trainer and asked by an interviewer that is not acquainted with the participants, as to eliminate any bias in the answers and reactions. The pre-training questionnaire was made up of five questions that target the participants' previous knowledge on the training topic. As for the post-training questionnaire, it targeted 1) the

participants' understanding of the explained topic, and 2) their opinion on the session and the trainer. The questionnaires are provided in appendix C. We recorded each participant's answers to analyze their facial emotion as they answered the pre and post training questionnaire.

5.4.3. Observations

For the FER classifier, we used the proposed CNN architecture. It is important to note that due to the models relatively small size (1.4MB), the classification was instantaneous with minimum latency. As for the observations from this experiments, the following can be deduced:

1. The face recognition module used allowed for face recognition even when the participants were not directly facing the camera (i.e. different postures)
2. Humans exhibit emotion differently, this lead to mis-classification for some participants. For instance, some individuals *frown* when thinking, which is labeled as *negative* according to the data fed to the classifier during training. This reflects the complexity of FER from facial expression as the way we express emotions differ greatly from one person to another, even from one situation to another.

CHAPTER 6: DISCUSSION

In this chapter, we review and discuss the results and limitations in this thesis. Table 6.1 highlights the highest results achieved in each experiment, for 3-class FER.

Table 6.1. FER Results Summary

Model	Input data	Dataset	Accuracy
Medium Gaussian SVM	48×48 images	FER2013	62.5%
Linear SVM	CNN features	Cross dataset	76.9%
Quadratic SVM	CNN features (PCA)	Cross dataset	74.4%
VGG-16	48×48 images	Cross dataset	73.2%
ResNet-50	48×48 images	Cross dataset	59%
Base model	48×48 images	Cross dataset	84%
Proposed Model	48×48 images	Cross dataset	86.5%
LSTM Model	CNN features (sequences)	CK+ (dynamic)	74.6%

In this thesis, we conducted several experiments to compare the performance of AI algorithms for the task of FER. As existing works are trained on lab-controlled datasets which limits the generalizability of the algorithms in real-life, we follow a cross-dataset approach to overcome that, additionally, to increase the size of the dataset. To create a baseline, we evaluated the performance of machine learning classifiers which were outperformed by SVM with an accuracy of 62% with a Medium Gaussian kernel. Another approach using SVM is done by training the classifier on the features extracted by CNN model. This increased the classification accuracy to ~77%.

Another experiment was to test the feasibility of transfer learning from pre-trained

models for FER. We fine-tuned and tested VGG-16 network and ResNet-50. For both models, we freeze the convolution layers and add fully-connected layers, detailed in appendix A. Overall, VGG-16 performed better for all cases in comparison to ResNet-50. For 3-class FER on the cross-dataset, VGG-16 achieves an accuracy of 73.2% with a model size significantly smaller than ResNet-50 (approx. 63 MB vs 144 MB).

The main contribution is the modifications applied to the CNN proposed in [38]. We choose this model as it uses depth-wise separable convolution, and we follow the same approach, considering it significantly reduces the number of trainable parameters, hence, the size of the model. The size of the proposed model is only (1.45 MB), this is an important feature as it would be more suitable for real-life application because: 1) smaller network classifies input faster, and 2) the small size facilitates deploying the model in hardware-constrained platforms. The proposed architecture increases the accuracy up to 86.5% compared to 84% by the original model.

As for dynamic FER, the proposed CNN model was also used for video FER and was tested in a real-life experiment as explained previously. Another approach is to leverage the spatio-temporal correlation in videos to improve FER, however, this method achieved an accuracy of 74%. This can be explained by the difference in the amounts of training data. For static FER, the dataset consists of 39K images, whereas the data for dynamic FER was only 327 examples.

All in all, the highest accuracy of 86.5% was achieved by the proposed model. One important observation is that for all experiments, the *neutral* emotion classification always had the lowest accuracy, specifically, it was mostly mis-classified as *negative* emotion. This can be justified by two possible reasons: 1) the *neutral* class had the lowest number of examples and perhaps the weighted loss does not perform as well

as balancing the dataset would, or 2) *neutral* emotion is generally an obscure concept. To clarify, the dataset is labelled by humans who perceive emotions differently. More importantly, several psychologists argue that there is no *neutral emotion*, simply, if an emotion is not *positive*, it is *negative* [58]. In more words, some believe that the *neutral* emotion cannot exist as our emotions are negatively or positively influenced. This ambiguity of neutrality in emotion leads us to doubt the feasibility of quantifying *neutral emotion*.

CHAPTER 7: CONCLUSION AND FUTURE WORK

In this thesis, we tested several artificial intelligence approaches for emotion recognition from facial expression. All of the experiments followed a categorical emotion model. For the first part of the thesis, we experiment with static FER using SVM, transfer learning and a proposed CNN model. From the experiments, the highest accuracy achieved was by using dense blocks and residual connections in CNN. The latter helps reduce the size of the model significantly which is beneficial for deploying the model in hardware/computation-constrained devices.

The second part of the thesis aimed at FER from videos. With that goal, we tested the proposed CNN for video classification by testing it on the CK+ videos, and by running a real-life experiment. The second approach was using an LSTM with the features extracted from the proposed CNN. The first approach performed better than the LSTM. Some possible improvements are increasing the size of the data; also, increasing the size of the model.

To sum up, we found that CNNs indeed performed better for FER classification with an accuracy of 86.5%. Furthermore, the CNN trained on static data is also effective for video FER, especially because of its small size. Although we anticipated the hybrid CNN-LSTM approach to further improve the performance, the limitations mentioned previously limited the accuracy to 74.6% only.

For future work, we plan to further improve the model architecture to improve the classification performance. In the previous chapter we discussed the difficulty of classifying neutral emotion. It is possible that using dimensional model, instead of a categorical one, will be improve FER. Also, other approaches for dynamic FER can be tested such as 3D CNNs. Studying FER from different types of images is an interesting

task that might make a more robust FER system. Finally, this work can be extended to detect emotion by combining different modalities such as audio and facial expression.

REFERENCES

- [1] A. Stergiou and R. Poppe, “Analyzing Human-Human Interactions: A Survey,” *arXiv:1808.00022 [cs]*, Aug. 2019, arXiv: 1808.00022.
- [2] R. Beale and C. Peter, “The Role of Affect and Emotion in HCI,” en, in *Affect and Emotion in Human-Computer Interaction: From Theory to Applications*, ser. Lecture Notes in Computer Science, C. Peter and R. Beale, Eds., Berlin, Heidelberg: Springer, 2008, pp. 1–11.
- [3] T. Kundu and C. Saravanan, “Advancements and recent trends in emotion recognition using facial image analysis and machine learning models,” in *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, ISSN: null, Dec. 2017, pp. 1–6.
- [4] R. Kaiser and K. Oertel, “Emotions in HCI: An Affective e-Learning System,” in *Proceedings of the HCSNet Workshop on Use of Vision in Human-computer Interaction - Volume 56*, ser. VisHCI '06, event-place: Canberra, Australia, Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2006, pp. 105–106.
- [5] R. A. Calvo and S. D’Mello, “Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications,” *IEEE Transactions on Affective Computing*, vol. 1, no. 1, pp. 18–37, Jan. 2010, Conference Name: IEEE Transactions on Affective Computing.
- [6] P. Tzirakis, G. Trigeorgis, M. A. Nicolaou, B. Schuller, and S. Zafeiriou, “End-to-End Multimodal Emotion Recognition using Deep Neural Networks,” *IEEE*

- Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1301–1309, Dec. 2017, arXiv: 1704.08619.
- [7] *Facial Action Coding System*, en-US.
- [8] J. Hamm, C. G. Kohler, R. C. Gur, and R. Verma, “Automated Facial Action Coding System for Dynamic Analysis of Facial Expressions in Neuropsychiatric Disorders,” *Journal of Neuroscience Methods*, vol. 200, no. 2, pp. 237–256, Sep. 2011.
- [9] S. Li and W. Deng, “Deep Facial Expression Recognition: A Survey,” *arXiv:1804.08348 [cs]*, Apr. 2018, arXiv: 1804.08348.
- [10] C. Jacobé de Naurois, C. Bourdin, A. Stratulat, E. Diaz, and J.-L. Vercher, “Detection and prediction of driver drowsiness using artificial neural network models,” en, *Accident Analysis & Prevention*, 10th International Conference on Managing Fatigue: Managing Fatigue to Improve Safety, Wellness, and Effectiveness”. Vol. 126, pp. 95–104, May 2019.
- [11] N. Yitzhak, T. Gurevich, N. Inbar, M. Lecker, D. Atias, H. Avramovich, and H. Aviezer, “Recognition of emotion from subtle and non-stereotypical dynamic facial expressions in Huntington’s disease,” en, *Cortex*, Feb. 2020.
- [12] S. W. White, L. Abbott, A. T. Wieckowski, N. N. Capriola-Hall, S. Aly, and A. Youssef, “Feasibility of Automated Training for Facial Emotion Expression and Recognition in Autism,” en, *Behavior Therapy*, vol. 49, no. 6, pp. 881–888, Nov. 2018.
- [13] M. Mohammadpour, H. Khaliliardali, S. M. R. Hashemi, and M. M. Alyan-Nezhadi, “Facial emotion recognition using deep convolutional networks,” in

2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), ISSN: null, Dec. 2017, pp. 0017–0021.

- [14] S. P S and M. G S, “Emotion Models: A Review,” *International Journal of Control Theory and Applications*, vol. 10, pp. 651–657, 2017.
- [15] X. Yicheng and D. Kollias, “Interpretable Deep Neural Networks for Dimensional and Categorical Emotion Recognition in-the-wild,” *arXiv:1910.05784 [cs, stat]*, Dec. 2019, arXiv: 1910.05784.
- [16] R. Shoja Ghiass, O. Arandjelović, A. Bendada, and X. Maldague, “Infrared face recognition: A comprehensive review of methodologies and databases,” en, *Pattern Recognition*, vol. 47, no. 9, pp. 2807–2824, Sep. 2014.
- [17] N. Samadiani, G. Huang, W. Luo, Y. Shu, R. Wang, and T. Kocaturk, “A Novel Video Emotion Recognition System in the Wild Using a Random Forest Classifier,” en, in *Data Science*, ser. Communications in Computer and Information Science, Singapore: Springer, 2020, pp. 275–284.
- [18] P. Lucey, J. 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 *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, 2010, pp. 94–101.
- [19] M. Lyons, M. Kamachi, and J. Gyoba, *The Japanese Female Facial Expression (JAFFE) Database*, eng, type: dataset, Apr. 1998.
- [20] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R.

- Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio, “Challenges in Representation Learning: A report on three machine learning contests,” *arXiv:1307.0414 [cs, stat]*, Jul. 2013, arXiv: 1307.0414.
- [21] A. Dhall, A. Kaur, R. Goecke, and T. Gedeon, “EmotiW 2018: Audio-Video, Student Engagement and Group-Level Affect Prediction,” *arXiv:1808.07773 [cs]*, Aug. 2018, arXiv: 1808.07773.
- [22] T. Evgeniou and M. Pontil, “Support Vector Machines: Theory and Applications,” en, in *Machine Learning and Its Applications: Advanced Lectures*, ser. Lecture Notes in Computer Science, G. Paliouras, V. Karkaletsis, and C. D. Spyropoulos, Eds., Berlin, Heidelberg: Springer, 2001, pp. 249–257.
- [23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, en. MIT Press, Nov. 2016, Google-Books-ID: Np9SDQAAQBAJ.
- [24] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial Transformer Networks,” en, *arXiv:1506.02025 [cs]*, Feb. 2016, arXiv: 1506.02025.
- [25] François Chollet, *Deep Learning with Python*, 1st. Manning, 2017.
- [26] B. Fasel and J. Luetten, “Automatic facial expression analysis: A survey,” en, *Pattern Recognition*, vol. 36, no. 1, pp. 259–275, Jan. 2003.
- [27] C.-H. Hjortsjö, *Digitale Bibliothek / Man’s face and mimic language*.
- [28] M. Dahmane and J. Meunier, “Emotion recognition using dynamic grid-based HoG features,” in *Face and Gesture 2011*, Mar. 2011, pp. 884–888.
- [29] A. Dhall, A. Asthana, R. Goecke, and T. Gedeon, “Emotion recognition using PHOG and LPQ features,” in *Face and Gesture 2011*, Mar. 2011, pp. 878–883.

- [30] D.-T. Lin, “Facial Expression Classification Using PCA and Hierarchical Radial Basis Function Network,” *J. Inf. Sci. Eng.*, 2006.
- [31] N. Lopes, A. Silva, S. R. Khanal, A. Reis, J. Barroso, V. Filipe, and J. Sampaio, “Facial emotion recognition in the elderly using a SVM classifier,” in *2018 2nd International Conference on Technology and Innovation in Sports, Health and Wellbeing (TISHW)*, Jun. 2018, pp. 1–5.
- [32] Y.-I. Tian, T. Kanade, and J. Cohn, “Recognizing action units for facial expression analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 97–115, Feb. 2001.
- [33] R. Melaugh, N. H. Siddique, S. A. Coleman, and P. Yogarajah, “Gabor and HOG approach to facial emotion recognition,” en, Maynooth, Ireland: NUI Maynooth, Jun. 2017.
- [34] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, ISSN: 1063-6919, vol. 1, Dec. 2001, pp. I–I.
- [35] Z. Yu and C. Zhang, “Image Based Static Facial Expression Recognition with Multiple Deep Network Learning,” in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ser. ICMI ’15, event-place: Seattle, Washington, USA, New York, NY, USA: ACM, 2015, pp. 435–442.
- [36] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning Transferable Architectures for Scalable Image Recognition,” *arXiv:1707.07012 [cs, stat]*, Jul. 2017, arXiv: 1707.07012.

- [37] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, “Deep Learning for Emotion Recognition on Small Datasets Using Transfer Learning,” in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ser. ICMI '15, event-place: Seattle, Washington, USA, New York, NY, USA: ACM, 2015, pp. 443–449.
- [38] O. Arriaga, M. Valdenegro-Toro, and P. Plöger, “Real-time Convolutional Neural Networks for Emotion and Gender Classification,” *CoRR*, vol. abs/1710.07557, 2017.
- [39] D. K. Jain, P. Shamsolmoali, and P. Sehdev, “Extended deep neural network for facial emotion recognition,” en, *Pattern Recognition Letters*, vol. 120, pp. 69–74, Apr. 2019.
- [40] S. Minaee and A. Abdolrashidi, “Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network,” *arXiv:1902.01019 [cs]*, Feb. 2019, arXiv: 1902.01019.
- [41] Z. Ming, J. Xia, M. M. Luqman, J.-C. Burie, and K. Zhao, “FaceLiveNet+: A Holistic Networks For Face Authentication Based On Dynamic Multi-task Convolutional Neural Networks,” *arXiv:1902.11179 [cs]*, Feb. 2019, arXiv: 1902.11179.
- [42] J. C. Hung, K.-C. Lin, and N.-X. Lai, “Recognizing learning emotion based on convolutional neural networks and transfer learning,” en, *Applied Soft Computing*, vol. 84, p. 105 724, Nov. 2019.
- [43] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for Activation Functions,” *arXiv:1710.05941 [cs]*, Oct. 2017, arXiv: 1710.05941 version: 1.

- [44] H.-J. Lee and K.-S. Hong, “A study on emotion recognition method and its application using face image,” en, in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju: IEEE, Oct. 2017, pp. 370–372.
- [45] A. Sepas-Moghaddam, A. Etemad, P. L. Correia, and F. Pereira, “A Deep Framework for Facial Emotion Recognition using Light Field Images,” in *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, ISSN: 2156-8103, Sep. 2019, pp. 1–7.
- [46] L. Chao, J. Tao, M. Yang, Y. Li, and Z. Wen, “Long short term memory recurrent neural network based encoding method for emotion recognition in video,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, ISSN: 2379-190X, Mar. 2016, pp. 2752–2756.
- [47] N. Jain, S. Kumar, A. Kumar, P. Shamsolmoali, and M. Zareapoor, “Hybrid deep neural networks for face emotion recognition,” en, *Pattern Recognition Letters, Multimodal Fusion for Pattern Recognition*, vol. 115, pp. 101–106, Nov. 2018.
- [48] Y. Fan, X. Lu, D. Li, and Y. Liu, “Video-based emotion recognition using CNN-RNN and C3D hybrid networks,” in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ser. ICMI ’16, Tokyo, Japan: Association for Computing Machinery, Oct. 2016, pp. 445–450.
- [49] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A Survey on Deep Transfer Learning,” *arXiv:1808.01974 [cs, stat]*, Aug. 2018, arXiv: 1808.01974.

- [50] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” en, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 1800–1807.
- [51] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *CoRR*, vol. abs/1512.03385, 2015.
- [52] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” *arXiv:1608.06993 [cs]*, Aug. 2016, arXiv: 1608.06993.
- [53] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv:1409.1556 [cs]*, Apr. 2015, arXiv: 1409.1556.
- [54] A. Geitgey, *Facial Recognition API for Python*, original-date: 2017-03-03T21:52:39Z, Jan. 2020.
- [55] K. S. Sahla and T. Senthil Kumar, “Classroom Teaching Assessment Based on Student Emotions,” en, in *Intelligent Systems Technologies and Applications 2016*, J. M. Corchado Rodriguez, S. Mitra, S. M. Thampi, and E.-S. El-Alfy, Eds., ser. *Advances in Intelligent Systems and Computing*, Cham: Springer International Publishing, 2016, pp. 475–486.
- [56] S. Sharmila and D. A. Kalaivani, “Automatic Facial Emotion Analysis System For Students In Classroom Environment,” en, p. 8, 2018.
- [57] P. Torres-Carrion, C. Gonzalez-Gonzalez, and A. M. Carreño, “Facial Emotion Analysis in Down’s syndrome children in classroom,” in *Proceedings of the XVI International Conference on Human Computer Interaction*, Vilanova i la Geltru, Spain: Association for Computing Machinery, Sep. 2015, pp. 1–2.

- [58] K. Gasper, L. A. Spencer, and D. Hu, “Does Neutral Affect Exist? How Challenging Three Beliefs About Neutral Affect Can Advance Affective Research,” English, *Frontiers in Psychology*, vol. 10, 2019, Publisher: Frontiers.

APPENDIX A: CNN ARCHITECTURES

Table A.1. VGG-16 and ResNet-50 for 3-Class on FER2013

Layer	VGG-16 (Unit)	ResNet-50 (Unit)
FC 1	512	512
FC 2	512	256
FC 3	3	3
Trainable Parameters	526,851	4,326,915

Table A.2. VGG-16 and ResNet-50 for 7-Class on FER2013

Layer	VGG-16 (Unit)	ResNet-50 (Unit)
FC 1	512	512
FC 2	512	256
FC 3	7	7
Trainable Parameters	528,903	4,327,943

Table A.3. VGG-16 and ResNet-50 for 3-Class on CK+

Layer	VGG-16 (Unit)	ResNet-50 (Unit)
FC 1	128	512
FC 2	64	512
FC 3	3	512
Trainable Parameters	74,115	4,459,011

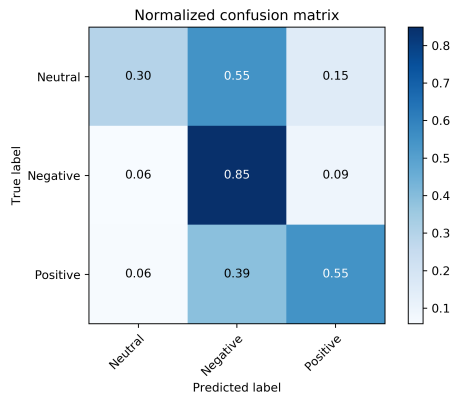
Table A.4. VGG-16 and ResNet-50 for 7-Class on CK+

Layer	VGG-16 (Unit)	ResNet-50 (Unit)
FC 1	512	512
FC 2	512	512
FC 3	7	7
Trainable Parameters	528,903	4,461,063

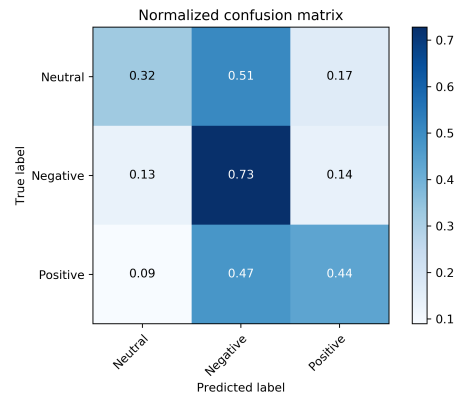
Table A.5. VGG-16 and ResNet-50 for Cross-dataset

Layer	VGG-16 (Unit)	ResNet-50 (Unit)
FC 1	512	512
FC 2	512	512
FC 3	3	3
Trainable Parameters	526,851	4,459,011

APPENDIX B: EXTENDED RESULTS

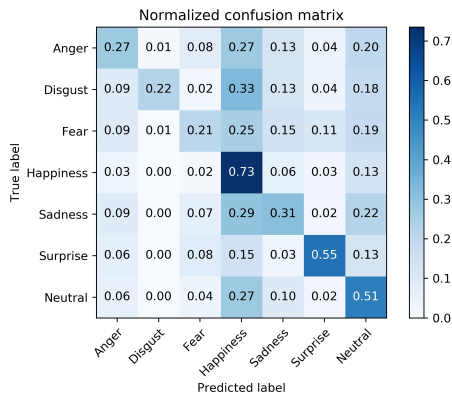


(a) VGG-16

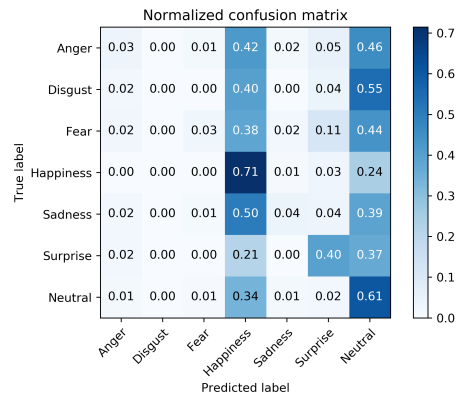


(b) ResNet-50

Figure B.1. Confusion matrix for 3-class FER2013.

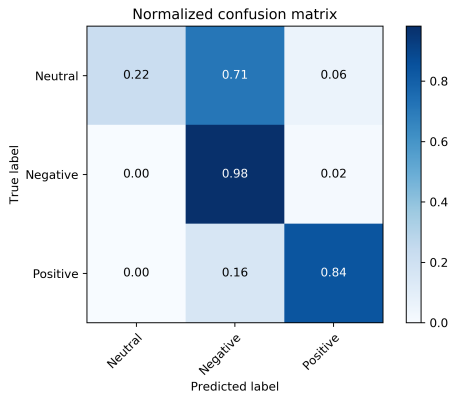


(a) VGG-16

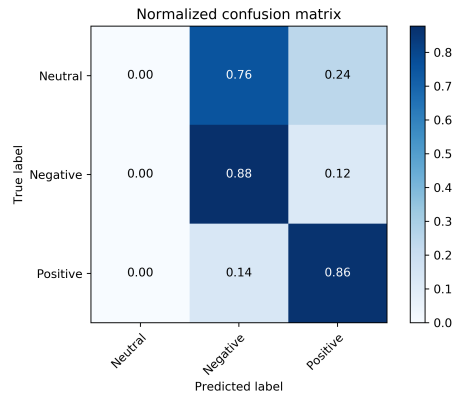


(b) ResNet-50

Figure B.2. Confusion matrix for 7-class FER2013.

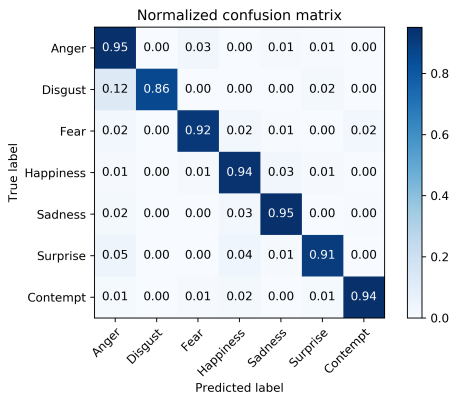


(a) VGG-16

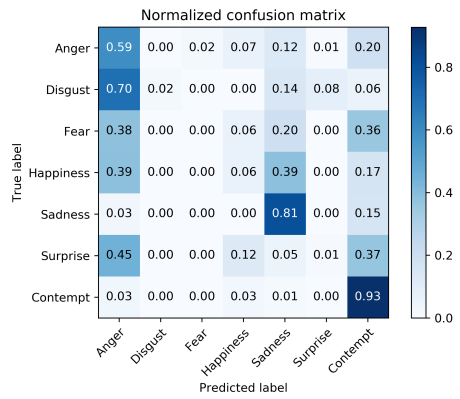


(b) ResNet-50

Figure B.3. Confusion matrix for 3-class CK+.

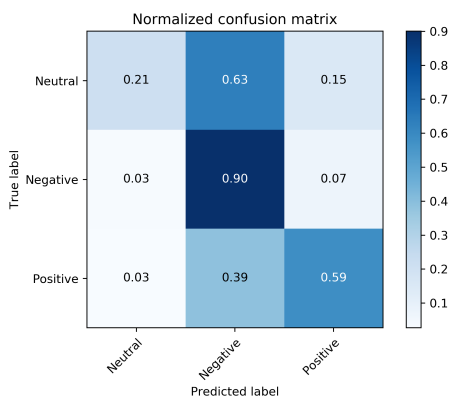


(a) VGG-16

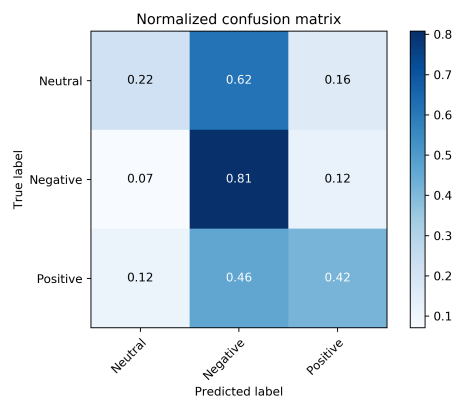


(b) ResNet-50

Figure B.4. Confusion matrix for 7-class CK+.



(a) VGG-16



(b) ResNet-50

Figure B.5. Confusion matrix for cross-dataset.

APPENDIX C: TRAINING SESSION EVALUATION EXPERIMENT

C.1. Pre-training Questionnaire

1. Have you used/do you know PowerBI?
2. Have you used/do you know pivot tables?
3. Do you use/know MS excel?
4. If yes, how many years of experience do you have in using MS Excel?
5. Why do you want to use PowerBI?

C.2. Post-training Questionnaire

1. Have you learned PowerBI?
2. Did you learn something new?
3. From 0-10, how much have you learned from this session?
4. Did you like the trainer? (what do you think about the trainer?)
5. What are the two main steps in PowerBI?
6. what is DAX?
7. Do you think the software is useful for you?
8. Do you know pivot tables?
9. Will this tool make your life easier?
10. Can you experience your data?