Sign Language Recognition:

Integrating Prior Domain Knowledge into Deep Neural Networks

Pedro Miguel Martins Ferreira

Supervisors: Ana Maria Rebelo (PhD)
Jaime dos Santos Cardoso (PhD)

Programa Doutoral em Engenharia Eletrotécnica e de Computadores
Faculdade de Engenharia, Universidade do Porto

This dissertation is submitted for the degree of
Doctor of Philosophy

FEUP January 2020

Abstract

Sign language is an integral form of communication and, currently, considered the standard education method of deaf people worldwide. Although sign languages are often characterized as being purely manual-visual languages, it has long been recognized that sign language communication is performed by means of manual articulations along with facial expressions to convey meaning. Sign languages are full-fledged complex systems of communication with their own lexicon, syntax, and grammar. This is why most hearing people are unfamiliar with sign language, which obviously creates a serious communication barrier between deaf communities and the hearing majority.

As a key technology to help to bridge the gap between deaf and hearing people, Sign Language Recognition (SLR) has become one of the most active research topics in the human-computer interaction field. Its main purpose is to automatically translate the signs, from images or video, into the corresponding text or speech.

The main goal of this thesis is to develop novel machine learning and pattern recognition methodologies to be integrated into SLR systems, for a robust recognition. Taking full advantage of the representational power of deep learning techniques, we investigate novel deep neural network architectures, training frameworks, and regularization strategies, in order to overcome several challenges that exist in the SLR research field.

The evolution of sign acquisition systems, especially thanks to the introduction of low-cost depth sensors (e.g., Microsoft Kinect and Leap Motion), has made possible the integration of different data modalities, such as RGB and depth, for a more accurate sign segmentation and recognition. In this thesis, several deep multimodal learning strategies are investigated. The most relevant contribution is a novel deep model that explicitly learns the complementary aspects among different input data modalities while maintaining the specificities of the signs captured by each modality individually.

Although the appearance of manual signs is well-defined in sign language dictionaries, in practice, there exists a large inter-signer variability in the manual signing process. These variations may arise due to regional, social, or educational factors and pose challenging problems in the development of SLR systems robust to new and unseen test signers. In this regard, we propose truly signer-independent SLR models capable of learning signer-invariant representations that preserve as much as possible the relevant information about the signs, while discarding the signer-specific traits that may hamper the sign recognition task.

Since non-manual elements, especially facial expressions, play an essential role in sign language communication, we develop fundamental research work on facial expression recognition due to its potential application into a complete SLR system. Particularly, we propose a novel end-to-end deep neural network architecture along with a well-designed loss function that jointly learns the most relevant facial parts along with the expression recognition. The result is a model that can learn expression-specific features.

The contributions presented throughout this thesis are validated in several databases, in which a series of state-of-the-art results are achieved. Additionally, we also present a new Portuguese sign language database, suitably annotated, and with such a unique composition that may open new research paths in the SLR research field.

Keywords: Machine Learning, Deep Learning, Neural Networks, Sign Language Recognition, Gesture Recognition, Multimodal Learning, Facial Expression Recognition, Regularization.

Resumo

A língua gestual é uma forma integral de comunicação e, atualmente, considerada o método padrão de educação de pessoas surdas em todo o mundo. Embora as línguas gestuais sejam frequentemente caracterizadas como sendo meios de comunicação puramente manuais, há muito tempo que se reconhece que a língua gestual é realizada através de articulações manuais em conjunto com expressões faciais, para transmitir significado. As línguas gestuais são sistemas complexos de comunicação com o seu próprio léxico, sintaxe e gramática. Razão pela qual a maioria dos ouvintes não está familiarizada com a língua gestual, o que obviamente cria uma enorme barreira de comunicação entre as comunidades surdas e a maioria dos ouvintes.

Como uma tecnologia chave para ajudar a colmatar a lacuna comunicacional entre os surdos e os ouvintes, o reconhecimento de língua gestual (SLR) tornou-se um dos tópicos de investigação mais ativos na àrea da interação homem-máquina. O principal objetivo é traduzir automaticamente os gestos, a partir de imagens ou vídeo, para texto ou voz.

O principal objetivo desta tese é desenvolver novas metodologias de aprendizagem computacional e reconhecimento de padrões a serem integradas em sistemas de SLR, para um reconhecimento robusto. Aproveitando ao máximo o poder representacional das técnicas de aprendizagem profunda, investigamos novas arquiteturas de redes neuronais profundas, estruturas de aprendizagem e estratégias de regularização, de forma a superar diversos desafios que existem na àrea de investigação de SLR.

A evolução dos sistemas de aquisição de gestos, especialmente graças à introdução de sensores de profundidade de baixo custo (por exemplo, o Microsoft Kinect e o Leap Motion), tornou possível a integração de diferentes modalidades de dados, como RGB e profundidade, para uma segmentação e reconhecimento dos gestos mais precisos. Nesta tese, são investigadas várias estratégias de aprendizagem multimodal. A contribuição mais relevante é um novo modelo capaz de aprender de forma explícita os aspectos complementares das diferentes modalidades, enquanto mantém as especificidades dos gestos que apenas podem ser capturadas por cada modalidade individualmente.

Embora a aparência dos gestos esteja bem definida nos dicionários de língua gestual, na prática, verifica-se uma grande variabilidade nos gestos quando realizados por diferentes falantes de língua gestual. Estas variações podem surgir devido a fatores regionais, sociais ou educacionais e criam enormes desafios no desenvolvimento de sistemas de SLR que sejam robustos a novos falantes de língua gestual. Neste sentido, são propostos modelos de SLR independentes do falante de língua gestual capazes de aprender representações que preservem tanto quanto possível as informações relevantes dos gestos e, simultaneamente, descartam os traços específicos do falante de língua gestual, que dificultam a tarefa de reconhecimento gestual.

Como os elementos não-manuais, especialmente as expressões faciais, desempenham um papel importante na comunicação através da língua gestual, foi desenvolvido trabalho de investigação fundamental ao nível do reconhecimento de expressões faciais devido à sua potencial aplicação num sistema de SLR completo. Especificamente, é proposta uma nova rede neural profunda, juntamente com uma função de perda, capaz de aprender em simultâneo os componentes faciais relevantes e o reconhecimento das expressões. O resultado é um modelo capaz de aprender características altamente discriminativas das expressões faciais.

As contribuições apresentadas ao longo desta tese são validadas em diversas bases de dados, nas quais uma série de resultados ao nível do estado de arte são alcançados. Além disso, é apresentada uma nova base de dados de língua gestual portuguesa, devidamente anotada, e com uma composição particular que poderá abrir novas linhas de investigação na área de SLR.

Palavras-chave: Aprendizagem Computacional, Aprendizagem Profunda, Redes Neuronais, Reconhecimento de Língua Gestual, Reconhecimento de Gestos, Aprendizagem Multimodal, Reconhecimento de Expressões Faciais, Regularização.

Acknowledgements[1]

I could not start this without expressing my most sincere acknowledgments to my supervisor, Prof. Ana Rebelo. I will never forget the opportunity Ana gave me to pursue this Ph.D., for trusting and accepting to work with me since the beginning. Thank you so much for your permanent support, guidance, and availability that, during this entire journey, went beyond the role of a supervisor. I am very proud to call you supervisor, but I am even prouder to call you friend.

I would like to extend my acknowledgments to Prof. Jaime Cardoso. Although being "just" my co-supervisor on paper, also acted as a true supervisor. Thank you so much for all inspirational talks, brainstorming meetings, and insights. It is hard to believe how Prof. Jaime always has a solution for each research problem.

To all VCMI group members, our research group at INESC TEC, many thanks for the fantastic research environment and friendship. A special thanks for those who have collaborated with me: Eduardo, Ricardo, Filipa, Kelwin, and Diogo. Again, a special thanks to Diogo, who became one of my best friends.

Thanks to Ana Rio and all the staff of the Agrupamento de Escolas Eugénio de Andrade, Escola EB2/3 de Paranhos, who supported the acquisition and annotation of the database presented in this thesis.

Last but not least, I would like to thank my family, where I can include all my childhood friends from my hometown. Off course, a very special thanks to my parents, to my grandparents, and to my brother, who raised me and always believed in me, giving me the best conditions to live, study and follow my dreams. It would not be possible to accomplish this work without their unconditional understanding, encouragement, and love. A particular word to my grandfather, who unfortunately passed away during this journey, ..., this is for you!

[1] This work was supported by INESC TEC and Fundação para a Ciência e a Tecnologia (FCT) along with the Ph.D. grant "SFRH/BD/102177/2014".

“Much has been written about AI’s potential to reflect both the best and the worst of humanity. For example, we have seen AI providing conversation and comfort to the lonely; we have also seen AI engaging in racial discrimination. Yet the biggest harm that AI is likely to do to individuals in the short term is job displacement, as the amount of work we can automate with AI is vastly bigger than before. As leaders, it is incumbent on all of us to make sure we are building a world in which every individual has an opportunity to thrive. Understanding what AI can do and how it fits into your strategy is the beginning, not the end, of that process.” – Andrew Ng

Table of contents

List of figures

List of tables

Acronyms / Abbreviations

Acc	Classification Accuracy
ACER	Average Classification Error Rate
ANN	Artificial Neural Network
APCER	Attack Presentation Classification Error Rate
ASL	American Sign Language
BLSTM-NN	Long Short-Term Memory Neural Network
BPCER	Bona-fide Presentation Classification Error Rate
BSL	British Sign Language
CK+	Extended Cohn-Kanade
CNN	Convolutional Neural Network
CRF	Conditional Random Field
CVAE	Conditional Variational Autoencoder
DBN	Deep Belief Network
DeSIRe	Deep Signer-Invariant Representations
DGS	German Sign Language (orig. *Deutsche Gebärdensprache*)
DTW	Dynamic Time Wrapping
EENReg	End-to-End Network with Regularization
EmotiW	Emotion Recognition in the Wild

EV	Eigenvoice
FE	Facial Expression
FER	Facial Expression Recognition
FER-2013	Facial Expression Recognition 2013
GAN	Generative Adversarial Neural Network
HCI	Human-Computer Interaction
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
HSV	Hue, Saturation, Value
JAFFE	Japanese Female Facial Expressions
KL	Kullback-Leibler
LBP	Local Binary Patterns
LeakyReLU	Leaky Rectified Linear Unit
LFDA	Local Fisher Discriminant Analysis
LGP	orig. *Língua Gestual Portuguesa*
LIBRAS	orig. *Língua Brasileira de Sinais*
LSF	orig. *Langue des Signes Française*
LSTM	Long Short-Term Memory Network
MAP	Maximum a Posteriori
ME	Movement of Epenthesis
MKLM	Microsoft Kinect and Leap Motion
MLLR	Maximum Likelihood Linear Regression
ML	Machine Learning
MLP	Multilayer Perceptron

MMD	Maximum Mean Discrepancy
MSE	Mean-Squared Error
NGT	Dutch Sign Language (orig. *Nederlandse Gebarentaal*)
NN	Neural Network
OMG-Emotion	One-Minute Gradual-Emotional Behavior
PAD	Presentation Attack Detection
PAI	Presentation Attack Instrument
QP	Quadratic Programming
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RGB-D	Red, Green, Blue-Depth
RGB	Red, Green, Blue
RNN	Recurrent Neural Network
SDK	Software Development Kit
SFEW	Static Facial Expressions in the Wild
SGD	Stochastic Gradient Descent
SIFT	Scale Invariant Feature Transform
SLR	Sign Language Recognition
SOV	Subject-Object-Verb
SVM	Support Vector Machines
SVO	Subject-Verb-Object
SWLDA	Stepwise Linear Discriminant Analysis
t-SNE	t-Distributed Stochastic Neighbor Embedding
ULBP	Uniform Local Binary Patterns

VAE	Variational Autoencoder
VSIA	Visible Spectrum Iris Artefact
WFD	World Federation of the Deaf
wLBP	Weighted Local Binary Patterns
YCbCr	Luminance (Y) and Chrominance (Cb and Cr) color values

Chapter 1

Introduction

Sign languages are the naturally occurring linguistic systems that arise within a Deaf community and, currently, considered the standard education method of deaf people worldwide. Sign language communication is expressed trough manual signs (i.e., articulated hand gestures) in combination with non-manual elements (i.e., facial and body expressiveness). Deaf people have difficulty in speaking and learning spoken languages like hearing people. However, with sign language, they can communicate as efficiently and seamlessly.

According to the World Federation of the Deaf (WFD) [10], there are approximately 70 million deaf people worldwide. In Portugal, the deaf community is composed of about 30 000 members of deaf and severely hearing-impaired people that need to recur to sign language to communicate [2]. The population of sign language speakers may be extended by family and friends of the deaf, interpreters, and the curious, who learn the language by their own initiative.

Contrarily to popular belief, sign language is not universal and, just like spoken languages, it has its own lexicon, syntax, and grammar. As such, most hearing people are entirely unfamiliar with sign language, which creates a severe communication barrier between deaf communities and the hearing majority. The result is the isolation of deaf communities from the overall society. This communication problem becomes even more frightening if we think that there are deaf people that are not able to communicate with their own closest relatives.

1.1 Sign Language Recognition

Automatically analyzing and recognizing sign language has become one of the key problems in the human-computer interaction (HCI) field. Sign Language Recognition (SLR) systems are meant to translate the signs into the corresponding text or speech automatically. This is important not only to bridge the communication gap between deaf and hearing people but also to increase the amount of content the deaf can access, such as the creation of educational tools or games for deaf people and visual dictionaries of sign language.

The SLR problem has been addressed in the literature by using wearable devices, such as data gloves, or vision-based systems [59]. Vision-based systems, either those using RGB and/or depth camera systems, face the problem of the inherently noisy and ambiguous nature of the input data. Nevertheless, vision-based SLR is the most natural choice for real-world applications, since it is less invasive, and there is no need to wear cumbersome devices that may affect the natural signing movement.

A traditional vision-based SLR system is typically composed of three main steps: (i) hands and/or face detection, (ii) feature extraction, and (iii) sign recognition. Current SLR research is mostly facing the challenges associated with the extensive vocabulary size in the continuous SLR (i.e., recognition of sentences). This research trend may give a false impression that the recognition of isolated signs, either using video or static images, is already a solved problem. However, a brief literature review reveals that static SLR is still a very challenging task, especially under unconstrained scenarios. Current SLR systems often impose several restrictions either in the acquisition conditions or in the manual signing process of the signers. Moreover, several fundamental challenges, such as the combination of the complementary characteristics of different sources of input data, the signer-independent problem associated with the large inter-signer variability, and the analysis of the non-manual elements of sign languages, have been vaguely addressed by the SLR research community. This thesis proposes new contributions to each of these identified problems.

In terms of machine learning algorithms, SLR methodologies have gradually shifted from relying on hand-crafted feature extraction processes to deep learning-based approaches. Deep learning is currently considered a major breakthrough by both machine learning and computer vision communities thanks to the ability of jointly learning high-level feature representations along with the discriminative function providing the final classification. However, learning deep neural networks remains non-trivial, and the amount of training data required significantly increases along with the complexity of the models. Most of the available databases in the SLR context are relatively small, since collecting reliable labeled and annotated data is extremely costly. Incorporating prior domain knowledge in the neural network's architecture and learning framework is a principled way of regularizing the entire learning process and reducing the amount of required training data. It is, therefore, essential to devise novel regularization strategies of incorporating prior knowledge about the sign language domain when designing deep neural networks for SLR.

1.2 Objectives

This thesis aims at developing machine learning algorithms that can potentially be integrated into actual SLR systems. Following the progress that has been made in the SLR field, mainly thanks to the introduction of deep learning-based SLR techniques, we intend to further investigate novel deep network architectures, training frameworks, and regularization strategies, in order to achieve a more robust SLR. To accomplish this purpose, two major research lines were followed in the course of this thesis.

The first research line was directed towards the development of SLR methodologies based on the analysis of the manual component of sign languages (i.e., hand gestures). Here, two main problems were addressed, namely:

- **Multimodal SLR**: The introduction of low-cost consumer 3D sensors, such as the Microsoft Kinect and the Leap Motion, has made possible the development of hybrid SLR frameworks/models that integrate different input modalities of more than one device/sensor. During this thesis, we intend to investigate deep multimodal learning strategies to leverage the complementary aspects of different sources of input data. Rather than adopting a conventional multimodal learning structure that simply involves feature- or decision-level fusion strategies, our primary goal is to further explore the implicit dependence between different modalities. In this regard, we aim at developing a deep network model that explicitly models what is unique and shared between modalities. The underlying idea is that the desired multimodal features should comprise the shared properties between different modalities while retaining the modality-specific properties that can only be captured by each modality individually.

- **Signer-independent SLR**: Although current SLR systems demonstrate remarkable performances for signer-dependent settings, their recognition rates typically decrease significantly when the signer is new to the system. This performance drop is mainly due to the significant inter-signer variations that exist in the manual signing process of sign languages. To tackle this issue, our goal is to design a deep model capable of explicitly learning signer-invariant feature representations. These underlying feature representations should preserve as much information as possible about the signs while discarding the signer-specific traits that are irrelevant for sign recognition.

The second line of work is more related to the non-manual component of sign languages, in particular to the analysis of facial expressions. Given the critical role of facial expressions in the sign language communication, an ideal SLR system should integrate manual signs along with facial expressions. However, **Facial Expression Recognition** (FER) still remains

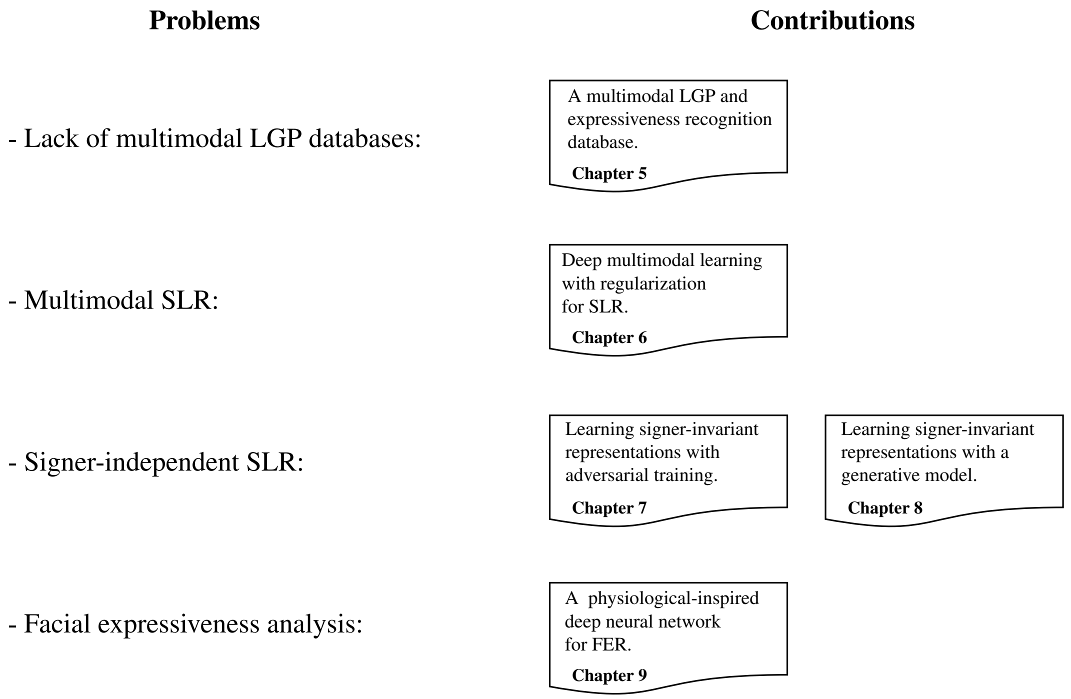

Problems
Contributions
- Lack of multimodal LGP databases:
A multimodal LGP and expressiveness recognition database.
Chapter 5
- Multimodal SLR:
Deep multimodal learning with regularization for SLR.
Chapter 6
- Signer-independent SLR:
Learning signer-invariant representations with adversarial training.
Chapter 7
Learning signer-invariant representations with a generative model.
Chapter 8
- Facial expressiveness analysis:
A physiological-inspired deep neural network for FER.
Chapter 9

Figure 1.1: Thesis problems and contributions.

an open problem by itself. Therefore, in the course of this thesis, we focus on the development of fundamental research work on FER, whose possible outcomes may have the potential of being integrated into an SLR system. Particularly, we intend to incorporate domain knowledge in the deep network architecture and learning process, based on the strong support from physiology and psychology that facial expressions are the result of the motions of facial muscles [58, 39]. The key idea is to explicitly drive the model towards the most relevant facial areas for the expression recognition, such as the facial components (i.e., eyes, eyebrows, nose, mouth) and expression wrinkles.

In parallel with these research activities, we intend to design a novel database in order to overcome some of the major shortcomings of the currently available SLR databases. On the one hand, there is a lack of **Portuguese Sign Language** (LGP - orig. *Língua Gestual Portuguesa*) **databases**. On the other hand, there is also a lack of SLR databases with multimodal data depicting both the manual and non-manual elements of sign language (i.e., hand gestures and facial/body expressiveness).

For a better comprehension of the reader, the four problems addressed in this thesis, along with the main contributions related to each of them are summarized in Figure 1.1.

1.3 Contributions

As refereed above, the work developed and presented throughout this thesis aimed at considering a wide array of topics of interest to the SLR research community. In this regard, the main contributions of this thesis can be summarized as follows:

- **Multimodal Sign Language and Expressiveness Recognition Database**, considers the acquisition and annotation of a novel SLR database of the LGP that comprises two major components: (i) an LGP dataset, and (ii) a duo-interaction dataset between deaf and/or hearing people. The designed database can be used for different purposes like SLR tasks or emotion/expressiveness recognition from body language. On the basis of this study, we foresee that the possibility of understanding the emotions and expressiveness behind the signs may open new research paths in SLR. The presented database along with the corresponding annotations are already made publicly available for research and benchmark purposes.

- **Deep Multimodal Learning for SLR**, considers the proposal of a novel end-to-end feature-level deep neural network that explicitly models private representations that are specific to each modality and shared feature representations that are similar between them. By imposing such constraints in the learning process, the model is able to jointly learn both modality-specific and modality-shared features and outperform the state-of-the-art multimodal approaches.

- **Learning signer-invariant representations with adversarial training**, considers the proposal of a deep neural network along with an adversarial training objective, which is able to learn feature representations that combine both sign discriminativeness and signer-invariance. We further demonstrate how to extend the proposed adversarial training objective for other applications (e.g., biometric liveness detection), in which it is desirable to learn feature representations invariant to some specific domain or aspect.

- **Learning signer-invariant representations with a generative model**, considers the development of a Conditional Variational Autoencoder (CVAE)-based model capable of learning latent representations whose conditional posterior distribution, given the image and its sign label, is independent of the signer identity. The result is a truly signer-independent model robust to new test signers.

- **Facial Expression Recognition**, considers the proposal of a novel deep neural network architecture along with a well-designed loss function that explicitly models both informative local facial regions and expression recognition. The result is a model that is able to jointly learn facial relevance maps and expression-specific features for a proper recognition.

1.4 List of Publications

The research work developed throughout this thesis resulted in several publications in journals, international conferences, and national conferences, as listed below.

Journal Papers

- **Ferreira, P. M.**, Marques, F., Cardoso, J. S., and Rebelo, A. (2018b). Physiological inspired deep neural networks for emotion recognition. *IEEE Access*, 6:53930–53943

- **Ferreira, P. M.**, Cardoso, J. S., and Rebelo, A. (2019a). On the role of multimodal learning in the recognition of sign language. *Multimedia Tools and Applications*, 78(8):10035–10056

- **Ferreira, P. M.**, Pernes, D., Rebelo, A., and Cardoso, J. S. (2019b). Desire: Deep signer-invariant representations for sign language recognition. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–16

- **Ferreira, P. M.**, Pernes, D., Rebelo, A., and Cardoso, J. S. (2019d). Signer-independent sign language recognition with adversarial neural networks. *International Journal of Machine Learning and Computing (IJMLC)*. (accepted)

International Conference Papers

- **Ferreira, P. M.**, Cardoso, J. S., and Rebelo, A. (2017a). Multimodal learning for sign language recognition. In Alexandre, L. A., Salvador Sánchez, J., and Rodrigues, J. M. F., editors, *Pattern Recognition and Image Analysis*, pages 313–321, Cham. Springer International Publishing

- **Ferreira, P. M.**, Pernes, D., Rebelo, A., and Cardoso, J. S. (2019c). Learning signer-invariant representations with adversarial training. In *The 12th International Conference on Machine Vision (ICMV 2019)*

- **Ferreira, P. M.**, Sequeira, A. F., Pernes, D., Rebelo, A., and Cardoso, J. S. (2019e). Adversarial learning for a robust iris presentation attack detection method against unseen attack presentations. In *2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*

National Conference Papers

- **Ferreira, P. M.**, Rodrigues, I. V., Rio, A., Sousa, R., Pereira, E. M., and Rebelo, A. (2014). Corsil: A novel dataset for portuguese sign language and expressiveness recognition. In *RecPad 2014: Conference on Pattern Recognition*, pages 1–2

- **Ferreira, P. M.**, Cardoso, J. S., and Rebelo, A. (2016). Facial key-points detection using a convolutional encoder-decoder model. In *RecPad 2016: Conference on Pattern Recognition*, pages 1–2

- **Ferreira, P. M.**, Cardoso, J. S., and Rebelo, A. (2017b). The potential of multimodal learning for sign language recognition. In *RecPad 2017: Conference on Pattern Recognition*, pages 1–2. **(best paper award)**

- **Ferreira, P. M.**, Marques, F., Cardoso, J. S., and Rebelo, A. (2018a). An expression-specific deep neural network for emotion recognition. In *RecPad 2018: Conference on Pattern Recognition*, pages 1–2

1.5 Document Structure

This thesis is organized into ten chapters, including the Introduction (Chapter 1), each one describing the work led during the last four years.

Chapter 2 presents the most fundamental aspects of sign languages, with a particular emphasis on the linguistics aspects of the LGP that served as the basis for the design and acquisition of the LGP database presented in this thesis. The chapter ends by highlighting the general role of the face in sign languages.

Chapter 3 focuses on the theoretical background behind the pattern recognition techniques used in the development of the present work.

In Chapter 4, a literature review of the most relevant SLR research work is provided. The major steps of a typical SLR system are fully described, and open-questions and near-future trends about SLR, in general, are discussed.

Chapter 5 introduces the acquired multimodal database for LGP and expressiveness recognition purposes.

Chapter 6 concerns the topic of multimodal SLR, in which several deep multimodal learning techniques for a robust SLR, making use of data provided by Kinect and Leap

Motion, are presented. Notably, a novel end-to-end multimodal deep neural network, along with a regularization scheme, that explicitly learns both modality-specific and complementary feature representations, is proposed.

Chapters 7 and 8 are both devoted to the signer-independent problem. In Chapter 7 a novel signer-independent model based on adversarial training is proposed, whereas Chapter 8 presents a novel signer-independent model based on a generative model.

Chapter 9 is more related to the non-manual component of sign languages. In particular, a new physiological-inspired deep neural network for FER is proposed.

Finally, conclusions and possible future lines of work, including improvements to the solutions proposed in the previous chapters, are presented in Chapter 10.

Additionally, Appendix A lists the entire content of the proposed database. In Appendix B, we further demonstrate how to extend the proposed signer-invariant adversarial training objective for other domains. Appendixes C and D provide auxiliary material to complement the FER research work presented in Chapter 9.

Chapter 2

Fundamentals of Sign Language

This chapter aims at presenting the fundamental aspects of sign languages. It starts with a general definition and the importance of sign languages, followed by a historical overview of sign language, with a particular focus on the linguistic aspects of the LGP to serve as the basis for the conception of the SLR database presented in chapter 5. The chapter ends by highlighting the role of the face in sign languages and describing the most common approaches to represent facial expressions.

2.1 Sign Language

Sign language is an integral form of communication, primarily used by hearing-impaired people within deaf communities. It is a visual means of communication that uses handshapes and gestures, instead of sound, to represent ideas or concepts. Sign language users combine articulated hand movements (i.e., manual signs) with facial expressions and head/body movements (i.e., non-manual signs) to convey feelings, intentions, humor, complex and abstract ideas, among others [135, 20].

Sign language is not universal but community-driven. Different sign languages are used in different countries or regions with their own lexicon and grammar. Sign languages borrow both from geographically proximal languages and from the culture and spoken language of the place. As it happens with spoken languages, not only different vocabulary is developed, but also different grammatical rules [135]. Furthermore, there are also regional dialects in sign language. Even when performing identical signs, the variations between different signers are considerable [226].

2.1.1 A Historical Overview

Sign language is currently considered an integral form of communication and the standard education method for deaf people worldwide. However, this was not always true, and only with the work of William Stokoe for the American Sign Language (ASL) in 1960 [191], sign language has been defined as a system of communication with its own lexicon and grammar. Since then, it has changed and evolved into the complete communication system that people see today.

Since the beginning of human communication, humans used basic sign language processes to express their own ideas and thoughts. Even when vocal communication became the mainstream form of interaction among human beings, people still execute unconscious hand gestures and facial expressions to reinforce ideas in communication. In ancient times, deaf people were often persecuted and mistreated and, hence, deaf people had no chance to work on creating a language system. This lasted until the middle of the sixteenth century, when Pedro Ponce de Leon, a Spanish monk, created his own form of sign language to overcome his "*vow of silence*". His teaching methods have been then successfully taught to deaf children in Spain. Later in 1620, inspired by Leon's methods, Juan Pablo Bonet wrote a sign language book with the first sign language alphabet recognized in deaf history [4, 5].

Until around 1750, there was no organized deaf education. This lasted until when Abbe Charles Michel de L'Epee, a French Catholic priest, founded the first public free deaf school in Paris. L'Epee standardized a sign language alphabet for French language and established symbolic gestures that conveyed concepts as opposed to just letters. These signs quickly became a standard signed language and spread across Europe as more schools were instituted. In this regard, Abbe de L'Epee is considered as the "Father of the Deaf". His work on sign language and deaf education were one of the most important contributions to the evolution of sign languages across the world.

Some years later, in the 1800s, Thomas Hopkins Gallaudet traveled from Connecticut, USA, to Europe to study and acquire knowledge about the teaching methods of sign languages. In Europe, he met Laurent Clerc, a deaf instructor of sign language, and both of them returned to America to found together the first school for hearing impaired people in the United States. From there, the ASL had been established mainly based on the signs from French Sign Language (LSF - orig. ***L**angue des **S**ignes **F**rançaise*). This is the reason why sign languages generally do not have any linguistic relation to the local spoken languages. For instance, ASL is much more similar to the LSF than with the British Sign Language (BSL). The same happens with the LGP and the Brazilian Sign Language (LIBRAS - orig. ***Lí**ngua **Bra**sileira de **S**inais*) [4, 135].

Although sign language became commonly used, supporters of the oralism method argued that deaf people must learn spoken language to be completely integrated into the hearing society. Oralism involved the utilization of speechreading and speech to teach deaf students instead of manual signs. Therefore, the discussion around the Sign Language existence and definition as a language was far from being over. In 1880, a lively discussion opposing sign language to the oralism method was carried to the Second International Congress on Education of the Deaf, in Milan. The supporters of the oralism method won the vote since the Congress declared "*that the oral method should be preferred to that of signs in the education and instruction of deaf-mutes*". While in Europe every country accepted and implemented the Congress resolutions, ASL still was primarily used out of the classroom environment in the United States. The outcome of the Milan Conference was devastating, and the deaf continued showing a retarded development when compared to the hearing people, contributing to their social exclusion [4, 5].

Only in 1962, with the work of William Stokoe [191], the ideas behind the Milan Conference were abolished. Stokoe demonstrated that the Sign Language is a genuine language with unique syntax and grammar. As a result of Stokoe's remarkable work, the Sign Language was henceforth adopted and recognized as the first language of deaf people worldwide.

2.1.2 The Portuguese Sign Language

The Portuguese Sign Language has its own grammar, lexicon, and syntax. The communication is performed by means of gestures, generally associated and organized in sentences. Gestures can be further divided in cheremes, the equivalent to the phonemes of spoken languages. These cheremes are the basic structural unit of sign language gestures and are constituted by the following five elements explained below in the text:

1. Hand configuration;
2. Orientation;
3. Place of articulation;
4. Movement;
5. Facial and/or body expressions.

Hand configuration Hand configuration refers to the configuration of the hand and fingers in a particular sign. The gestures in the LGP can be performed with both hands. However, the hands do not have the same importance. There is a dominant hand, which carries the most relevant information, and a supporting hand that can be used to complement some gestures. The dominant hand is defined according to the signer, depending if the signer is left- or right-handed.

Hands can adopt a wide range of configurations by just changing the fingers positions. There are several gestures which require specific hand configurations as the alphabet or the ordinal numbers. Through the hands' configurations, it is possible to give shape to words. Dactylology, also known as fingerspelling, is an essential tool to represent several Portuguese words that do not have a correspondent gesture in LGP, such as the names of people or cities [135].

Orientation The hand orientation describes the placement of the palm and is tightly coupled to the hand configuration and movement. In some cases, the inversion of the hand orientation indicates the opposite gesture, as occurs with the pair of gestures /COME/ and /GO/ [135, 20].

Place of articulation Since the sign language information is carried through the visual medium, the place of articulation, defined as the place where the gestures are performed, is extremely important. In LGP, there are necessarily two places of articulation levels: 1) the virtual rectangle in front of the signer's face and torso with no contact between the hand and the signer's body and 2) the body parts of the signer that can be used as reference or contact point to some signs (see Figure 2.1) [135, 20].

Movement In sign language, some gestures are static, as the cardinal numbers which do not use movement to transmit information, while others are dynamic, as verbal subjects that have the movement as the main component. In dynamic gestures, after hand configuration, the hand(s) can move through the space and/or the contact points of the signer's body to perform the gesture. A gesture can result from an isolated movement or the combination of two or more contiguous movements. The action of movement can be further differenced by long or short, slow or fast and smooth or tight movements [135, 20].

Facial and/or body expressions Facial and body expressions also have an essential role in sign language. This kind of information conveys information indicating feelings on a sign. Facial expressions can be used by themselves, especially to indicate negation and the sentence type (i.e., affirmative, interrogative, exclamatory, and imperative).

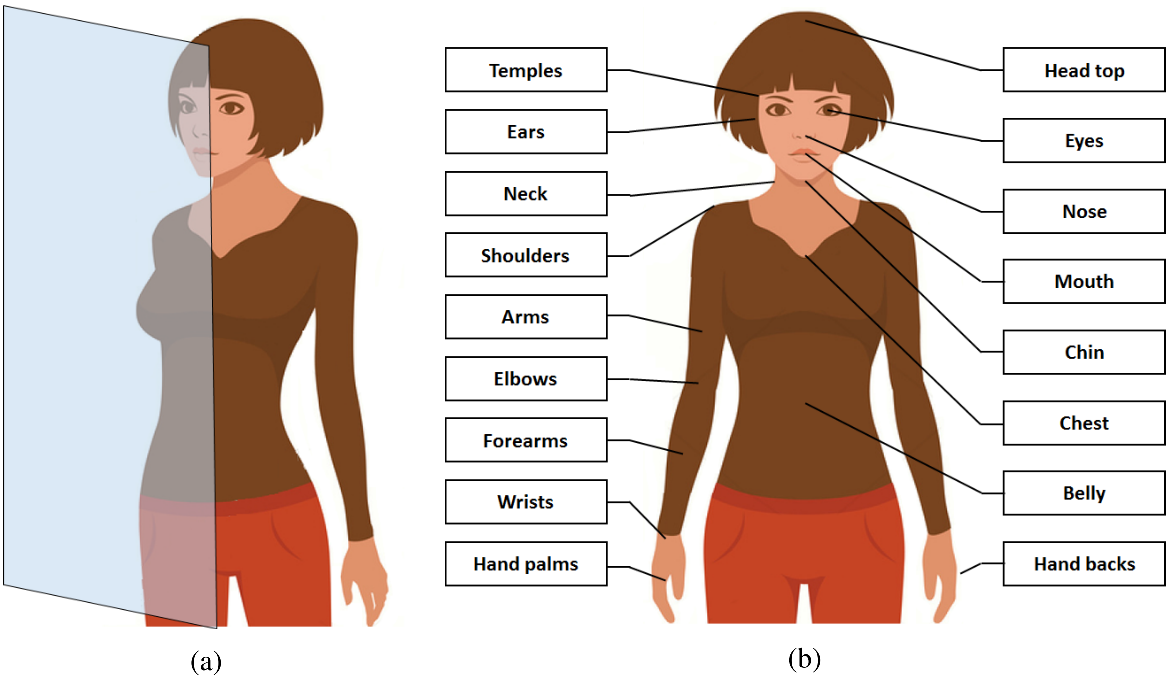


Figure 2.1: Places of articulation in LGP: (a) virtual rectangle in front of the signer's face and torso with no contact between the hand and the signer's body and (b) the body parts of the signer that can be used as reference or contact point to some signs (inspired by [135, 20]).

Body posture complements other gestures since the body can work as a reference point or be used to personify the subjects or objects to which the sign is referred [135, 20]. More details about the role of the face in sign languages can be found in Section 2.2.

2.1.3 Linguistic concepts

In this section we make a brief overview of some linguistic phenomena in LGP.

2.1.3.1 Syntax

Syntax in sign languages is made by spatial agreement of signs. In a syntactic point-of-view, the organization of sentences in LGP follows a Subject-Object-Verb (SOV) structure in opposition to what happens in the spoken languages that, typically, follows a Subject-Verb-Object (SVO) structure [135, 20]. Example 1 shows how the sentence - "*The cat eats fish*" - would be performed in the LGP. The sentence is in the gloss notation that is usually used to represent signs with text.

Example 1.

Table 2.1: General rules to write LGP in gloss notation [135].

1. It is always used capital letters.
2. The symbol **+** is used as separator between gestures.
3. The symbol **/** means a pause.
4. The symbol **//** means a long pause.
5. In Dactylology and cardinal numeration, the symbol **-** is used to separate letters and numbers, respectively.
6. The numbers that represent a quantity and the ordinal numbers are written out in full.
7. Verbs are always written in the infinitive form.

- EL[1]: The cat eats fish. (SVO)
- LGP: /CAT + FISH + EAT/. (SOV)

Nevertheless, as illustrated in Example 2, there are some situations in which the structure Object-Subject-Verb (OSV) can be used.

Example 2.

- EL: John bought a book. (SVO)
- LGP: /BOOK + J-O-H-N + BUY/. (OSV)

The negation in a sentence can be transmitted by just adding the gesture /NOT/ after the neutral form of the verb, as demonstrated in the Example 3.

Example 3.

- EL: The cat does not eat fish. (SVO)
- LGP: /CAT + FISH + EAT + NOT/. (SOV)

In Table 2.1 the most important rules to write the LGP in the gloss notation are presented.

2.1.3.2 Noun inflection

Regarding proper nouns, fingerspelling is often used since most of them do not have a known gestural representation. Therefore, the solution, in such cases, is to fingerspell the letters of the proper noun.

[1] EL - English Language

In Gender Typically, concepts in LGP do not have an associated gender, and, hence, they do not need inflection. For animated beings, including human beings and animals, gender can be specified with a prefix, by expressing the gender 'male' or 'female' before the noun (e.g., as what happens for 'female dog', for which the corresponding sign becomes /FEMALE+DOG/). In case of omission, the male gender is assumed. One possible exception is to have separate signs to denote the male and female gender, as in case of /LION/ and /LIONESS/.

In Number The number in LGP can be specified by three different processes, as illustrated in Figure 2.2. The first is the incorporation that allows specifying the quantity after the noun explicitly. Examples are /BOOK+FIVE/, or to use a determinative for amounts of difficult numerical quantification, for instance, /BOOK+MANY/. The second process is called repetition, which means that a sign is performed multiple times as it happens with the sign of /TREE(S)/. The last possibility is the reduplication process, in which the sign is performed with both hands as the example of /PERSON(S)/ [135, 20, 20].

2.1.3.3 Adjectives

As described in the following, the sign of an adjective depends on its origin [135, 20].

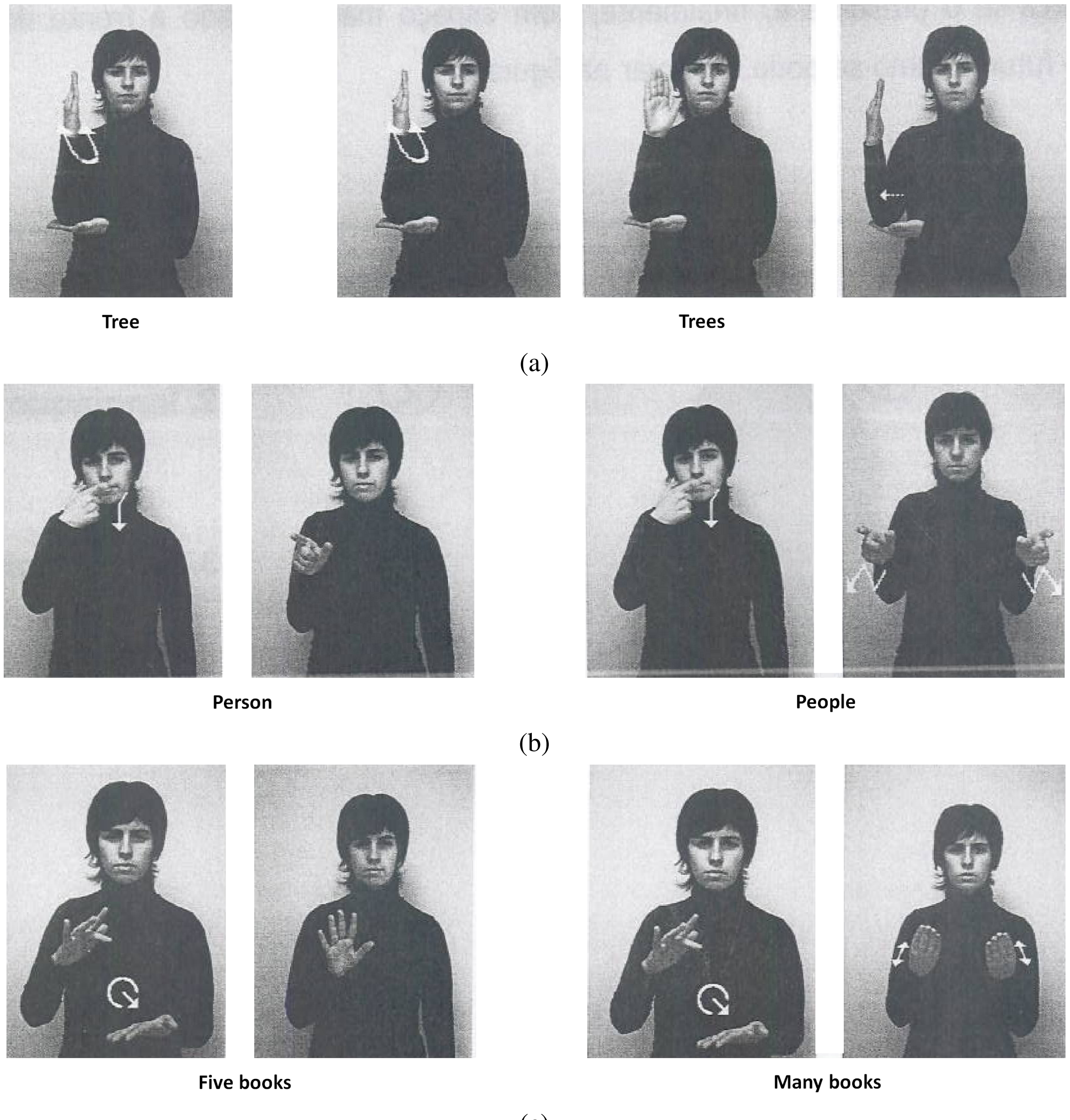
Nominal adjectives, which are adjectives derived from nouns, use the lexical root of the corresponding noun. However, the distinction is defined through the prolongation of the gesture along with facial expressions. For instance, the amplification of the sign /HUNGER/ along with a facial expression leads to the adjective /HUNGRY/.

Verbal adjectives, which are adjectives derived from verbs, use the lexical root of the corresponding verb. However, the distinction is defined through the gesture prolongation, facial expressions, repetition and/or reduplication processes. For example, the reduplication of the sign /SPEAK/ along with a facial expression leads to the adjective /CHATTY/.

Invariant adjectives keep their original signs independently of the context that they are performed. It is the case of /GIRL+BEAUTIFUL/ and /CAR+BEAUTIFUL/, in which the sign of the adjective 'beautiful' is performed in the same way.

2.1.3.4 Numbers

Numbers, in LGP, can be used as an isolated number (cardinal), ordinal number, composed number (e.g., 321), and quantitative qualifier. Compound numbers are signed using a similar process to words that do not have a correspondent gesture in LGP, in which each digit of the number is signed individually. For instance, '321' is signed as '3', followed by '2' and '1'



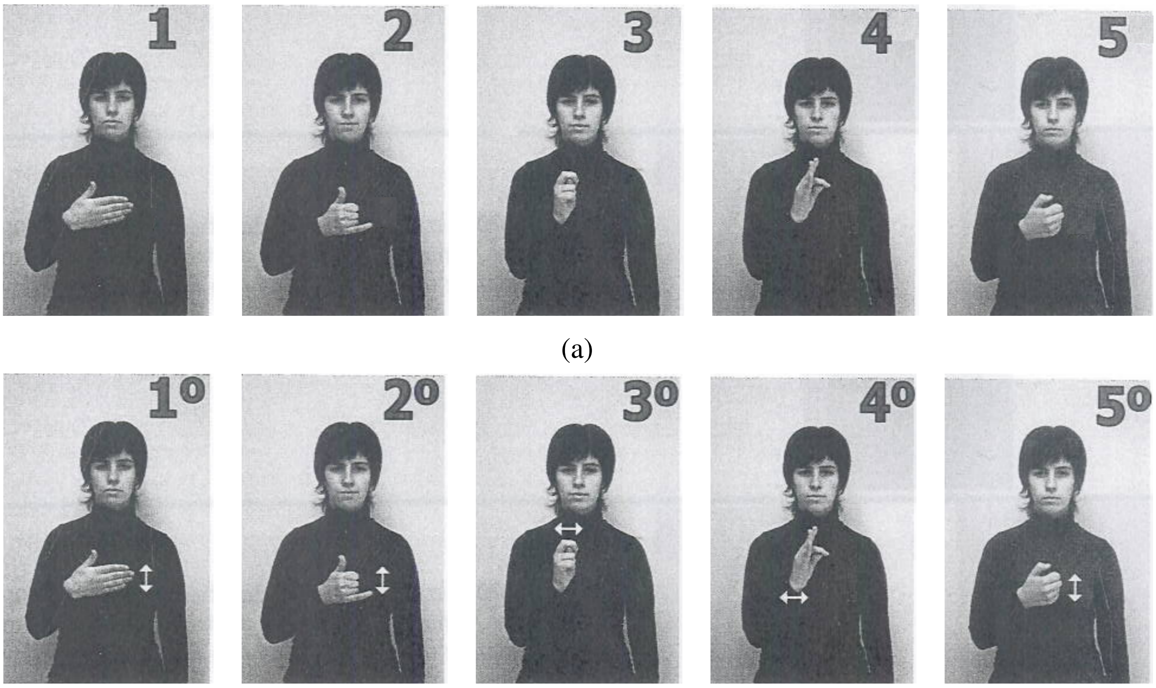
(a)

(b)

(c)

Figure 2.2: Noun inflection processes in the LGP: (a) repetition, (b) reduplication, and (c) incorporation (adapted from [135]).

with a slight offset in space as the number grows. Signs associated with ordinal numbers use the configuration of cardinal numbers along with a hand-shake movement (see Figure 2.3). In addition to these systems, some numbers have a different sign, such as ‘10’, ‘100’, and ‘1000’ which have their own sign. Moreover, signs associated with each number also vary their forms to express quantities, a repetition or a duration [135, 20].



(a)

(b)

Figure 2.3: Numeration in LGP: (a) cardinal numeration and (b) ordinal numeration (adapted from [135]).

2.1.3.5 Verb inflection

Most of the verbs in LGP are inflected according to the associated subjects. Thereby, their signs are affected by the action, the time, and the way the action is realized. Therefore, in some cases, verbs are signed recurring to different hand configurations and expressiveness, describing how the action is happening [135, 20].

In Time LGP grammar [135] refers to a temporal line in the gesturing space with which verbs should concord within past, present, and future tenses (see Figure 2.4). The verb inflection is made along this imaginary line with eye, eyebrow, and upper body movement. A common practice is to add a time adverb to the end of sentence, such as ‘past’, ‘now’, ‘tomorrow’ or ‘future’ [135].

The adverbial expression is also performed along the timeline with a possible emphasis on the distance in time. For instance, the word ‘now’ is always signed in front of the signer close to their torso, but it can be signed even closer as we want to express the immediateness of the action [135, 15].

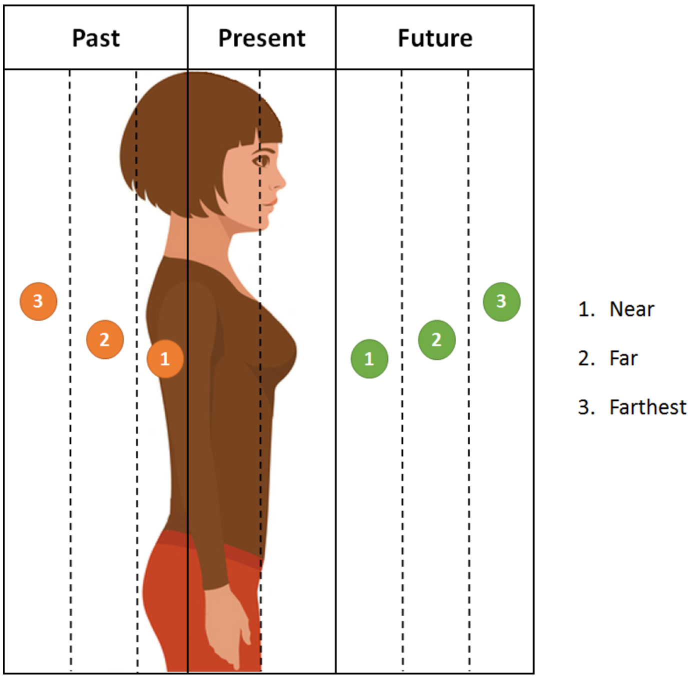


Figure 2.4: The imaginary temporal line for verb inflection in time (inspired by [135, 20]).

In Person Regarding verb agreement, there is no gender or number agreement in LGP. Typically, this kind of information is expressed by direct referencing to the subject (e.g., by indicating the personal pronoun before the verb).

In Aspect The grammatical aspect is crucial in LGP since it indicates the modality in which the verb is realized. Modality is realized throughout the imaginary temporal line, indicating duration and repetition through movement. For example, the verb ‘walk’ is signed with different movement modulation for ‘walk’, ‘walking’ and ‘walk hurriedly’. Another common practice is to add an adverb to the verb, as happens with time inflections, to express the aspect [15].

2.2 Facial Expressions

In psychology, emotion refers to the conscious and subjective experience that is characterized by mental states, biological reactions, and psychological or physiologic expressions (i.e., facial expressions). Facial expressions can be defined as the facial changes in response to a person’s internal emotional state, intentions, or social communication [114]. Together with voice, language, hands, and body posture, facial expressions form a fundamental communication system between humans in social contexts.

2.2.1 A Historical Overview

Humans perceive facial expressions as conveying meaning, but then arise the questions of where do they come from and what exactly do they mean?

In the 19th century, Duchenne de Boulogne conducted one of the first experiments on how facial expressions are produced, by electrically stimulating facial muscles (see Figure 2.5) [55]. This experiment leads him to believe that the human face worked as a map whose features could be codified into universal taxonomies of mental states. At the same time, based on observations of facial expressions typically associated with emotions, Charles Darwin hypothesized that they must have had some instrumental purpose in evolutionary history. For example, lifting the eyebrows might have helped our ancestors respond to unexpected environmental events by widening the visual field and therefore enabling them to see more. Also, constricting the nostrils in disgust served to reduce inhalation of noxious or harmful substances [39].

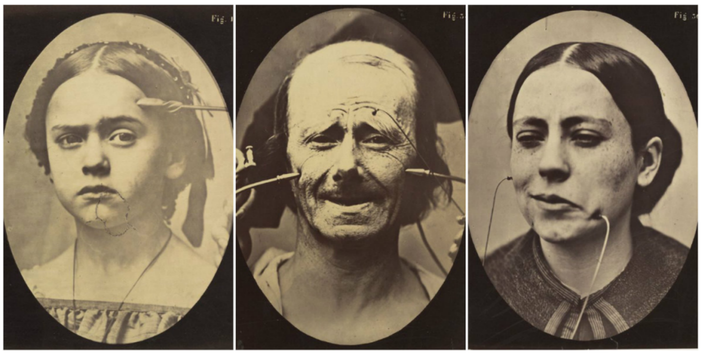


Figure 2.5: Study of facial expressions by electrically stimulating facial muscles [55].

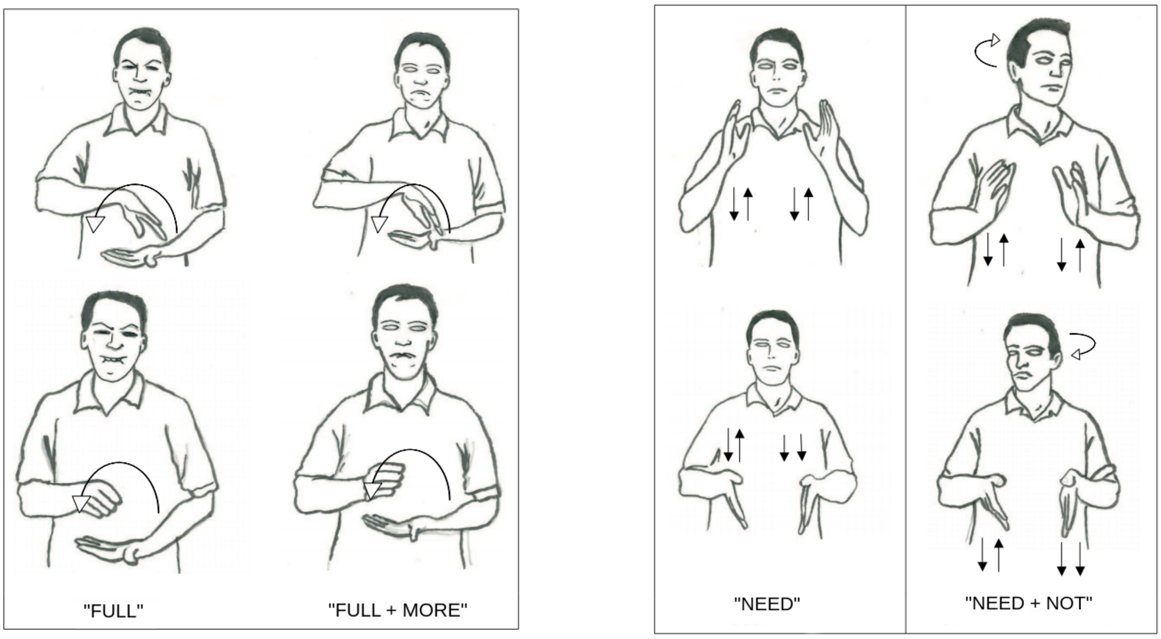
Following these ideas, Paul Ekman [60] claimed that there is a set of innate facial expressions, and they mean that the person making that face is experiencing an emotion, defending the universality of facial expression. He also claimed that there is a high degree of consistency in the facial musculature among peoples of the world. The muscles necessary to express primary emotions are found universally, and homologous muscles have been documented in non-human primates [61, 176].

Physiological specificity is also documented. Some studies support that both heart-rate and skin temperature vary with basic emotions. For instance, in anger, the blood flow of the hands increases to prepare for a fight. Left frontal asymmetry is greater during enjoyment while right frontal asymmetry is greater during disgust. These pieces of evidence support the argument that emotion expressions reliably signal action tendencies [112, 68].

The last decades of linguistic research on sign languages have revealed that there are facial expressions which are used in combination with manual signs and function as phonological features, morphemes, and syntactic/prosodic markers. For example, brow-raising marking conditional clauses [61].

2.2.2 Sign Languages and the Role of the Face

Facial expressions and head movements are used in sign languages at all levels of linguistic structure. Illustrative examples are depicted in Figure 2.6. At the phonological level, some signs have an obligatory facial component in their citation form. Facial expressions are used to mark grammatical forms, such as adverbial and adjectival modifiers (see Figure 2.6a). Facial expressions also include grammatical markings that extend over phrases to mark syntactic scope (e.g., to mark relative clauses, content questions, and conditionals, amongst others) [61]. Furthermore, as illustrated in Figure 2.6b, critical grammatical information such as negation is expressed through head gestures (e.g., periodic nods and shakes) [135, 20].



(a) (b)

Figure 2.6: Illustration of the role of the face in the sign language communication: (a) facial expressions as adjectival modifiers of quantity, and (b) head movements indicating negation (adapted from [140]).

2.2.3 Facial Expressions Description

Attempts to describe human emotion through the analysis of facial expressions mainly fall into two approaches: categorical and dimensional description.

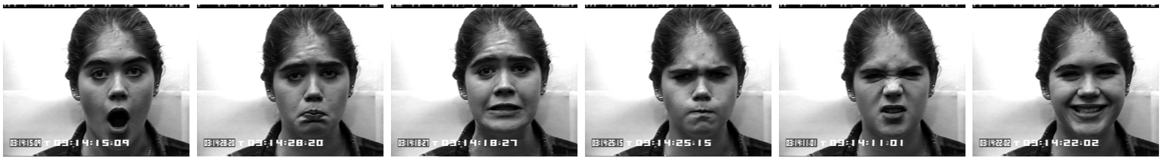


Figure 2.7: Primary emotions expressed on the face. From left to right: surprise, sadness, fear, anger, disgust, and happy (adapted from [127]).

2.2.3.1 Categorical description

Since at least the time of Darwin and, subsequently, greatly influenced by the research of Paul Ekman, it is common to classify emotion expressions into a set of distinct classes that can be easily recognized and described in the everyday language. The underlying assumption is that humans universally express a set of discrete primary emotions, which include happiness, surprise, fear, anger, sadness, and disgust (see Figure 2.7) [60]. Mainly due to its simplicity and its universality claim, this categorization scheme has been widely exploited in the development of automatic affective computing systems. More recently, some researchers have extended the set of primary emotions by considering a couple of additional emotion classes, such as relief and contempt [39].

Although these emotional categories are commonly inferred from facial expressions by most people, the way humans express themselves is more gradual and continuous and, hence, sometimes might be hard to categorize. As described below, this is somehow embedded in the dimensional emotion representation model [22].

2.2.3.2 Dimensional description

Another popular approach to describe emotion is the dimensional model, in which emotion expressions are represented in a two-dimensional space, usually arousal and valence (see Figure 2.8) [39, 22]. This dimensional space represents emotions based on their intensity and nature. For instance, high arousal is usually associated with expressions of high intensity (e.g., excitement) and low arousal with calm and relaxed expressions. High valence is commonly related to positive emotions and low valence to negative emotions. The higher dimensionality of such descriptions potentially allows describing more complex and subtle emotions. However, the richness of the representation space is more difficult to use in practice, because it can be challenging to link such described emotion to specific facial expressions. Automatic systems based on the dimensional representation model usually simplify the problem by dividing the space into a limited set of categories (e.g., positive vs. negative) [39].

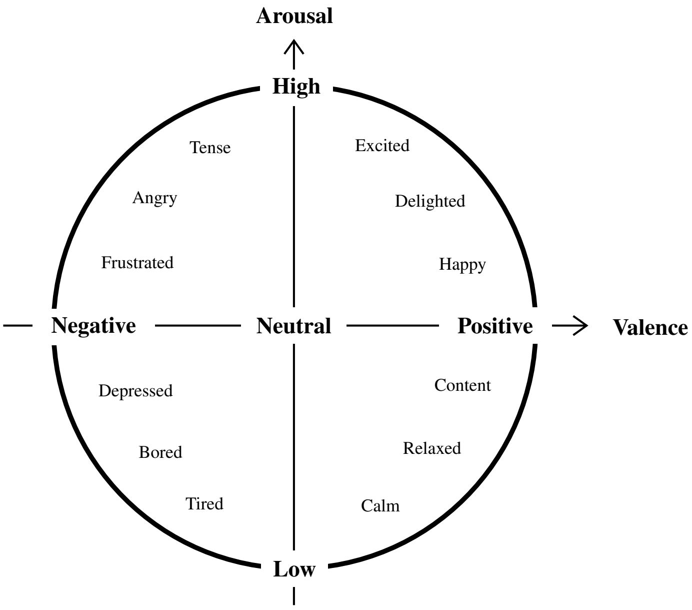


Figure 2.8: Dimensional emotion representation model.

2.3 Summary

Sign language is currently seen as a full-fledged form of communication and the standard education method of deaf people. Sign language communication is performed by means of gestures, instead of sound, to convey meaning. It involves not only hand shapes but also non-manual signs, such as facial expressions and body movements to express feelings in a sign.

The critical conclusions to grasp here are that sign languages are complex natural human languages with their vocabulary and grammatical rules. This form of communication is of paramount importance to deaf communities and helps to bridge the gap between deaf and hearing people.

Chapter 3

Background on Pattern Recognition

The automatic recognition of patterns in data is a fundamental problem with a long and successful history. Even though, nowadays, pattern recognition is still an exciting and thriving field with several practical applications and active research topics. Pattern recognition can be defined as the process of recognizing patterns and regularities in data through the utilization of machine learning (ML) algorithms [27]. These patterns are then used to take actions, such as classifying the data into different categories. Remarkable examples of pattern recognition applications include speech recognition, big data, and computer vision, in which the research topics addressed throughout this thesis, i.e., sign language recognition and facial expression recognition, can be included.

According to the learning (or training) procedure, pattern recognition can be broadly classified into two main problems, namely supervised learning and unsupervised learning. Supervised learning assumes that the training data, which is used to tune the parameters of a machine learning model, comprises a set of data observations along with their correct output labels (i.e., ground-truth labels). The ultimate goal is that the learned model generalizes as well as possible to new and unseen test data. Supervised learning problems are further categorized into: (i) classification tasks, in which the goal is to map input data into one of a finite number of discrete categories, and (ii) regression tasks, where the desired output consists of one or more continuous variables. In unsupervised learning problems, the training data consists of a set of data observations without any corresponding target values. Here, the goal may be either to discover groups (by clustering the data), or to determine the distribution of the data, or to project the data from a high-dimensional space to a low-dimensional space [27].

The performance of pattern recognition systems hugely depends on the representation of the data they are given [27, 74]. Traditional pattern recognition systems rely on feature engineering in order to design and extract a set of relevant features for each particular task.

However, for many tasks, it is difficult to know what kind of features should be extracted. This paradigm has been changed with the recent resurgence of deep learning in pattern recognition. Pattern recognition systems based on deep learning techniques can discover not only the mapping from the representation to the target output but also the representation itself [74].

This chapter aims at presenting the theoretical background behind the main pattern recognition techniques used throughout this thesis. Section 3.1 concerns the traditional pattern recognition pipeline, in which some of the most widely used feature extraction approaches and machine learning algorithms are described. Section 3.2 is devoted to deep learning in pattern recognition. Section 3.3 summarizes the most common regularization strategies that ensure the generalization of a machine learning model.

3.1 Traditional Pattern Recognition

After data acquisition, a traditional pattern recognition system has three major components: pre-processing, feature extraction and description, and a shallow machine learning algorithm. The traditional pattern recognition pipeline is depicted in Figure 3.1.

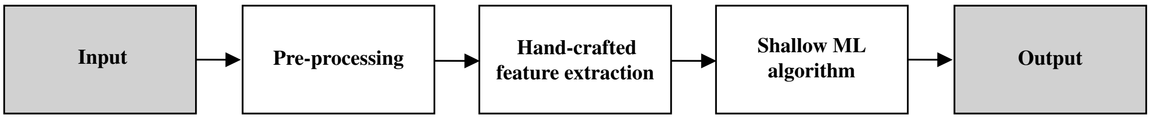


Figure 3.1: Traditional pattern recognition pipeline.

The first step usually entails a set of pre-processing techniques to normalize the raw data and remove noise, for instance, by using a low-pass filter. The next step concerns the extraction of a reliable data description. The goal of feature extraction is to extract a set of unique feature representations that best describe the data while reducing redundant information. In the final stage, the extracted features are used to train a shallow machine learning model for predicting the target-specific outputs.

3.1.1 Feature Extraction and Description

Given the importance of the representation in the overall performance of a pattern recognition system, several feature extraction, and description techniques have been proposed in the literature [195]. In this section, we outline Local Binary Patterns and Gabor filters, two of such techniques, as they will be used in this thesis.

Local Binary Patterns

Local Binary Patterns (LBP) [146] is an effective texture descriptor which summarizes the local spacial structure and the grey-level contrast of an image. For every pixel (x_c, y_c) in a given image $I(x,y)$, the LBP operator is defined as an ordered set of binary comparisons of pixel intensities between the current pixel (x_c, y_c) and its P neighbouring pixels: $\{(x_i, y_i)\}_{i=0}^{P-1}$ (see Figure 3.2a). As illustrated in Figure 3.2b, the local neighbourhood is defined as a set of P pixels sampled uniformly on a circle with radius R centered at (x_c, y_c). The decimal representation of the resulting P-bit LBP code can be expressed as follows:

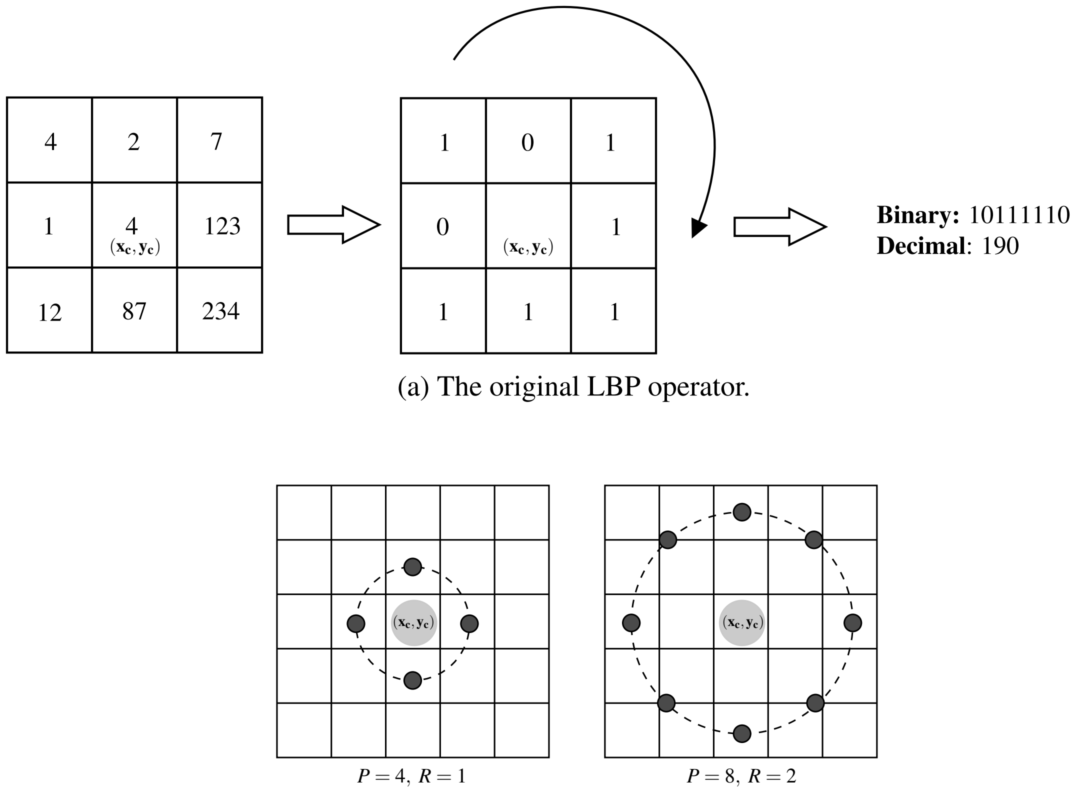
$$LBP(x_c, y_c) \;=\; \sum_{i=0}^{P-1} 2^i \cdot \; b(I(x_i, y_i) - I(x_c, y_c)), \tag{3.1}$$

where $I(x_c, y_c)$ is the intensity value of the current pixel (x_c, y_c), and $\{I(x_i, y_i)\}_{i=0}^{P-1}$ are the intensity values of its P surrounding pixels. The binarization function $b(\cdot)$ used for comparison is defined as:

$$b(x) \;=\; \begin{cases} 1, & \text{if } x \geq 0, \\ 0, & \text{if } x < 0. \end{cases} \tag{3.2}$$

The resulting LBP codes are then summarized into a histogram of 2^P bins, representing the frequency of occurrence of each one the 2^P possible binary codes. LBP histograms can be computed over the entire image or locally, over image sub-regions. In the latter case, local LBP histograms are concatenated to form a unified feature descriptor.

By definition, the original LBP operator is invariant to monotonic illumination transformations. Nevertheless, several extensions of the original LBP operator have been proposed, in order to attain other properties. One of the most useful LBP extensions is the so-called uniform local binary patterns (ULBP) [146]. Its main advantages are two-fold: (1) reduction of the feature vector size, and (ii) rotation invariance. The idea behind ULBP was motivated by the fact that some binary patterns (i.e., known as uniform patterns) occur more commonly in texture images than others. An LBP code is called uniform if the binary pattern contains at most two bitwise transitions from 0 to 1, or vice-versa. In the computation of the LBP histogram, all the non-uniform patterns are assigned to a single bin, while uniform patterns are assigned to individual bins. As there are a total of 58 uniform patterns, the length of the LBP feature vector is reduced 256 to 59.



(a) The original LBP operator.

(b) Circular neighbour-set for 2 different values of P and R.

Figure 3.2: Illustration of LBP computation.

Gabor Filters

Gabor filters are one of the most popular approaches for texture description. At the core of the Gabor filter-based feature extraction is the 2D Gabor filter function that, in the spatial domain, is defined as a Gaussian function modulated with a complex sinusoid [92]:

$$g(x,y) \;=\; \frac{f^2}{\pi\gamma\eta} e^{-\left(\frac{f^2}{\gamma^2}x'^2 \,+\, \frac{f^2}{\eta^2}y'^2\right)}\, e^{j2\pi f x'}, \tag{3.3}$$

where

$$x' \;=\; x\,\cos\theta \;+\; y\,\sin\theta, \tag{3.4}$$

$$y' \;=\; y\,\cos\theta \;-\; x\,\sin\theta. \tag{3.5}$$

Here, f denotes the central frequency of the filter, θ is the rotation angle, γ is the sharpness (or bandwidth) along the Gaussian major axis, and η corresponds to the sharpness along the minor axis that is perpendicular to the wave. In the given form, the aspect ratio of the Gaussian is η/γ.

The Gabor feature descriptor of a given image $I(x,y)$ is computed by passing the image through a bank of Gabor filters with different frequencies, orientations, and bandwidths. The idea is to make the feature descriptor invariant to illumination, rotation, scale, and translation. Particularly, the input image $I(x,y)$ is convolved with each Gabor filter g_i, $i = 1,...,N$, in the bank:

$$h_i(x,y) \;=\; g_i(x,y) * I(x,y), \tag{3.6}$$

where $h_i(x,y)$ is the i-th filter response (filtered image), and $*$ denotes a two-dimensional linear convolution. Then, under the assumption that image regions have homogeneous texture, means μ_i and standard deviations σ_i of the filter responses h_i are used to represent the texture of the region. The image is now characterized by a feature vector $[\mu_0, \sigma_0, ..., \mu_N, \sigma_N]^\top$.

3.1.2 Traditional Machine Learning Algorithms

Machine learning concerns the process of learning patterns within specific data representations, that can be used later to analyze and classify new samples [27]. In this section, one of the most widely used shallow classifiers, i.e., Support Vector Machines, is described.

Support Vector Machines

A Support Vector Machine (SVM), pioneered by Vapnik [218], is a discriminative classifier formally defined by a separating hyperplane (decision surface). The key idea is to find the hyperplane that separates data points from two distinct classes. Since an infinite number of such hyperplanes exist, the SVM algorithm finds the hyperplane that maximizes the margin (gap) between data points on the boundaries of each class (the so-called support vectors).

Formally, let $\mathbb{X} = \{(\boldsymbol{x}_i, y_i) | \boldsymbol{x}_i \in \mathbb{R}^m, y_i \in \{-1,1\}\}_{i=1}^N$ represent a binary dataset of N samples, where $\boldsymbol{x}_i$ corresponds to the i-th data point in the m-dimensional real space $\mathbb{R}^m$, and $y_i \in \{-1,1\}$ denotes its class label from one of two classes, $\mathbb{X}^+$ and $\mathbb{X}^-$. If the training data is linearly separable, there exist a pair of parallel bounding planes that separate the two classes of data $\mathbb{X}^+$ and $\mathbb{X}^-$:

$$\begin{aligned} \boldsymbol{w}^\top \boldsymbol{x} + b &\geq +1, \quad \text{for } \boldsymbol{x} \in \mathbb{X}^+, \\ \boldsymbol{w}^\top \boldsymbol{x} + b &\leq -1, \quad \text{for } \boldsymbol{x} \in \mathbb{X}^-, \end{aligned} \tag{3.7}$$

where $\boldsymbol{w}$ is the normal vector to these planes and b determines their location relative to the origin. The first plane of (3.7) bounds the data points of the class $\mathbb{X}^+$, whereas the second plane bounds the class $\mathbb{X}^-$.

According to the statistical learning theory [218], SVM achieves a better prediction ability via maximizing the margin between two bounding planes. The problem of maximizing the margin $\frac{2}{||\boldsymbol{w}||_2}$ is equivalent to the problem of minimizing $\frac{1}{2}||\boldsymbol{w}||_2^2$ subject to constraints that ensure class separability (i.e., all training samples $\boldsymbol{x}_i$ are correctly classified). This can be formulated as the following convex optimization problem:

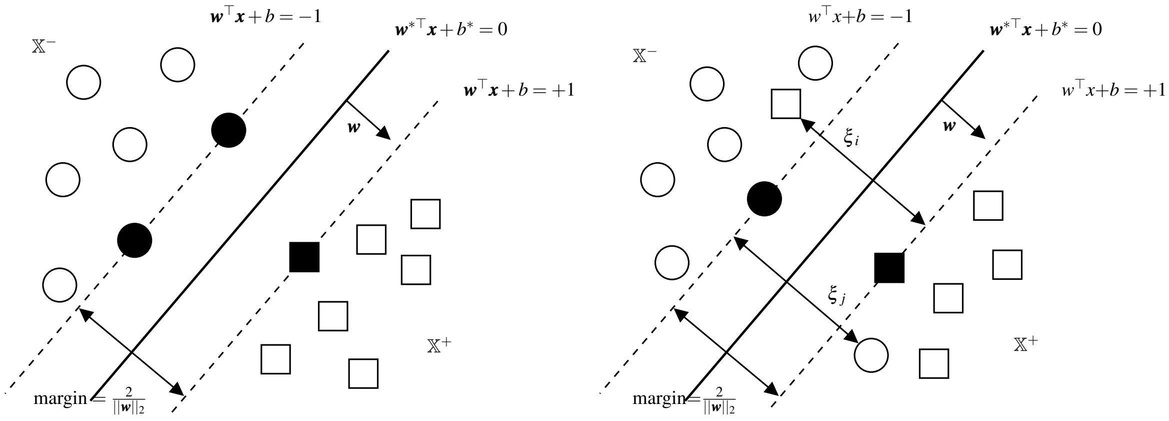
$$\begin{aligned} &\min_{\boldsymbol{w},b} && \frac{1}{2}||\boldsymbol{w}||_2^2 \\ &\text{subject to} && y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1,\ i = 1,...,N. \end{aligned} \tag{3.8}$$

Thus, the optimal solution parameters, $\boldsymbol{w}*$ and b^*, can be efficiently found by using any quadratic programming (QP) algorithm [110]. As illustrated in Figure 3.3, the optimal hyperplane is the plane midway between the bounding planes (3.7), defined by the optimal solution parameters as $\boldsymbol{w}^{*\top}\boldsymbol{x} + b^* = 0$. Once the optimal hyperplane has been found, the classifier (decision rule) is defined as follows:

$$g(\boldsymbol{x}) = \operatorname{sgn}(\boldsymbol{w}^{*\top}\boldsymbol{x} + b^*), \qquad \text{where } \operatorname{sgn}(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ -1, & \text{if } x < 0. \end{cases} \tag{3.9}$$

That is, new data points above or on the optimal hyperplane are classified as $+1$ (i.e., belonging to class $\mathbb{X}^+$), whereas data points below the hyperplane are classified as belonging to the class $\mathbb{X}^-$. The data points on the bounding planes, such that $\boldsymbol{w}^{*\top}\boldsymbol{x} + b^* = \pm 1$, are called the support vectors, which after the training process completely define the optimal separating hyperplane.

The above SVM formulation (3.8) is also known as hard-margin SVM. The problem with hard-margin SVMs is that it does not tolerate outliers. If the classes overlap, it is not possible to find a feasible solution $(\boldsymbol{w}, b)$ that satisfies the separation constraints in 3.8. To overcome this limitation, a variant called soft-margin SVM was introduced (see Figure 3.3b). The underlying idea is to introduce slack variables $\xi_i, i = 1,...,N$ into the constraints and penalize them in the objective function. This allows the violation of the separation constraints to a certain degree. The resulting soft-margin linear SVM formulation becomes:


$\boldsymbol{w}^\top\boldsymbol{x}+b=-1$
$\boldsymbol{w}^{*\top}\boldsymbol{x}+b^*=0$
$\boldsymbol{w}^\top\boldsymbol{x}+b=+1$
$\mathbb{X}^-$
$\boldsymbol{w}$
$\mathbb{X}^+$
margin $=\frac{2}{||\boldsymbol{w}||_2}$
$w^\top x+b=-1$
$\boldsymbol{w}^{*\top}\boldsymbol{x}+b^*=0$
$w^\top x+b=+1$
$\mathbb{X}^-$
$\boldsymbol{w}$
ξ_i
ξ_j
$\mathbb{X}^+$
margin $=\frac{2}{||\boldsymbol{w}||_2}$

(a) (b)

Figure 3.3: The illustration of (a) the hard-margin SVM in a linearly separable dataset, and (b) the soft-margin SVM in a nonlinearly separable dataset. Square and circle symbols represent data points from $\mathbb{X}^+$ and $\mathbb{X}^-$, respectively. Fulfilled symbols denote the support vectors (inspired by [110]).

$$\begin{aligned}
\min_{\boldsymbol{w},b,\xi} \quad & \frac{1}{2}||\boldsymbol{w}||_2^2 + C\sum_{i=1}^{N}\xi_i \\
\text{subject to} \quad & y_i(\boldsymbol{w}^\top\boldsymbol{x}_i+b) \geq 1-\xi_i \\
& \xi_i \geq 0, \ \text{ for } i=1,\dots,N.
\end{aligned} \tag{3.10}$$

where $C > 0$ is a hyparameter which balances the weights of the penalty term $\sum_{i=1}^{N}\xi_i$ versus the margin maximization term $\frac{1}{2}||\boldsymbol{w}||_2^2$. When C is very large, the soft-margin SVM is equivalent to the hard-margin SVM. Small values of C will result in a wider margin, at the expense of some misclassifications in the training data.

Although the soft-margin SVM can handle with nonlinearly separable data, especially caused by noisy data, its scope is limited since the classifier is still linear. In practice, data tend to have nonlinear hypersurfaces that better separate them. Soft-margin linear SVMs can be easily extended to nonlinear classifiers if solved using its dual formulation. Following the Lagrangian multipliers method, the dual problem of (3.10) can be formulated as follows:

$$\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j\langle\boldsymbol{x}_i,\boldsymbol{x}_j\rangle \\
\text{subject to} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0 \\
& 0 \leq \alpha_i \leq C, \ \text{ for } i=1,\dots,N,
\end{aligned} \tag{3.11}$$

where $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$ is the inner product of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Since the dual maximization problem is a convex quadratic objective function of the Lagrangian multipliers $\boldsymbol{\alpha}_i$ subject to linear constraints, it can be efficiently solvable by QP algorithms. Once the solution α^* is computed, both parameters $\boldsymbol{w}^*$ and b^*, which define the optimal hyperplane, can be determined. That is, the primal vector $\boldsymbol{w}^*$ is given by:

$$\boldsymbol{w}^* = \sum_{\{i|\alpha_i>0\}}^{N} \alpha_i y_i \boldsymbol{x}_i. \tag{3.12}$$

$\boldsymbol{w}^*$ only depends on the training samples $\boldsymbol{x}_i$, whose corresponding Lagrangian multipliers α_i^* are positive (i.e. the support vectors). Afterwards, the scalar b^* can be simply determined by taking any training point $\boldsymbol{x}_i$, such that $i \in \{k|0 \leq \alpha_k \leq C\}$, and solving $b^* = y_i - \boldsymbol{w}^{*\top}\boldsymbol{x}_i$ [110]. Finally, the classifier can be defined as follows:

$$g(\boldsymbol{x}) = \mathrm{sgn}(\boldsymbol{w}^{*\top}\boldsymbol{x} + b^*) = \mathrm{sgn}(\sum_{i|0<\alpha_i^*<C}^{N} \alpha_i^* y_i \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + b^*) \tag{3.13}$$

From the above dual SVM formulation, SVMs can be easily extended to nonlinear decision surfaces. The idea is to map the training data points from the original input space $\mathbb{R}^m$ to a higher dimensional feature space $\mathcal{F}$ by a nonlinear mapping Φ ($\Phi : \mathbb{R}^m \to \mathbb{R}^l$), and then fit a linear SVM in $\mathcal{F}$ that can separate the samples. Although the separating hyperplane is still linear in the transformed feature space $\mathcal{F}$, it will be nonlinear in the original input space $\mathbb{R}^m$. In practise, this is simply achieved by replacing the inner dot products $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$ in the dual optimization problem (3.11) with a nonlinear kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i)^\top \Phi(\boldsymbol{x}_j)$, $i, j = 1, ..., N$. An interesting property of the kernel function is that it allows computing the inner dot products in the transformed feature space $\mathcal{F}$ without actually transforming the data. The most popular kernel functions are:

- Linear kernel: $K(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x}^\top \boldsymbol{z}$.
- Polynomial kernel: $K(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^\top \boldsymbol{z} + 1)^d$, where d denotes the degree of the exponentiation.
- Gaussian kernel or Radial Basis Function (RBF): $K(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\gamma||\boldsymbol{x} - \boldsymbol{z}||^2)$, where γ is the width parameter of the Gaussian kernel.

Furthermore, it is worth to mention that several additional SVM extensions either for multi-class classification [54] or regression problems [53] were also proposed.

3.2 Deep Learning in Pattern Recognition

Deep learning techniques are currently encountered in many everyday pattern recognition applications, ranging from sign language recognition to a wide variety of image recognition and object detection tasks. Instead of relying on hand-crafted feature extractors, deep learning techniques, also considered as representation learning approaches, can learn useful features (or representations) from the raw input data directly (see Figure 3.4). Deep learning architectures consist of multiple layers, each one consisting of simple units, that in the course of the learning process, each layer yields a slightly more abstract and "useful" representation.

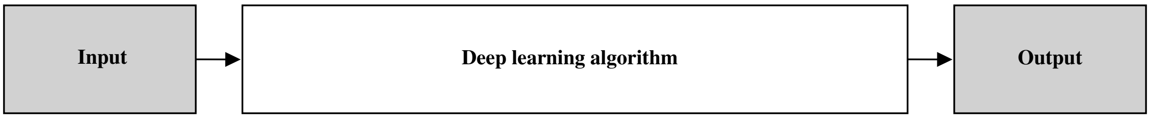


Figure 3.4: Deep learning-based pattern recognition pipeline.

In this section, we briefly review the deep learning approaches that provide the backbone of this thesis. Throughout the rest of this subsection, let $\mathbb{X} = \left\{(\boldsymbol{x}_i, y_i) | \boldsymbol{x}_i \in \mathbb{R}^D, y_i \in \mathbb{R}^C\right\}_{i=1}^N$ represent a labelled dataset of N samples, where $\boldsymbol{x}_i$ and y_i correspond to the i-th feature vector, in the D-dimensional real space $\mathbb{R}^D$, and the corresponding target output vector, respectively.

3.2.1 Fully Connected Neural Networks

Fully connected Neural Networks, also known as Multilayer Perceptrons (MLPs), represent the general foundation of deep learning architectures and methods [74]. As illustrated in Figure 3.5, an L-layer neural network consists of D input units, C output units, and several so-called hidden units. These units or neurons are arranged in layers, in such a way that an MLP comprises an input layer, an output layer, and $(L-1)$ hidden layers. The input layer $\boldsymbol{a}^0$ corresponds to the input feature vectors of the data, such that $\boldsymbol{a}^0 = \boldsymbol{x} = [x_1, \ldots, x_D] \in \mathbb{R}^D$, whereas the last layer represents the expected task-specific outputs, such that: $\boldsymbol{a}^L = [a_1^L, \ldots, a_C^L] \in \mathbb{R}^C$. The intermediate layers are referred to as hidden layers since their correct values are unknown and need to be found during the learning process. Specifically, the output of the i-th unit within layer l is given by a weighted sum of the neuron's activations (outputs) in the previous layer $(l-1)$, plus a constant bias, followed by a nonlinear activation function:

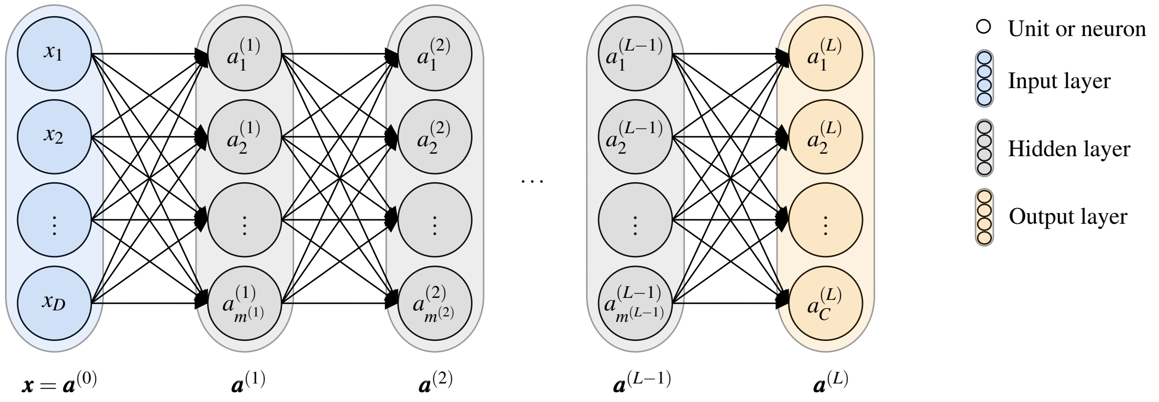


Figure 3.5: Illustrative representation of an L-layer fully connected neural network with D input units and C output units. The l-th layer contains $m^{(l)}$ hidden units.

$$a_i^{(l)} = \phi^{(l)}(z_i^{(l)}), \tag{3.14}$$

$$\text{with} \qquad z_i^{(l)} = \sum_{k=1}^{m^{(l-1)}} w_{i,k}^{(l)} a_k^{(l-1)} + b_i^{(l)}, \tag{3.15}$$

where $a_k^{(0)} = x_k$ corresponds to the k-th input feature, $w_{i,k}^{l}$ denotes the weighted connection from the k-th neuron in layer $(l-1)$ to the i-th neuron in layer l, and b_i^l can be regarded as an external input to the neuron and is referred to as bias. Here, $m^{(l)}$ denotes the number of units in layer l and, therefore, $m^{(0)} = D$, and $m^{(L)} = C$. For simplicity, the bias can be regarded as a weight, by introducing a dummy unit $a_0^{(l)} := 1$ in each layer, and Equations (3.14) may be rewritten in matrix notation for $m^{(l)}$ many neurons stacked horizontally as:

$$\boldsymbol{z}^{(l)} = \boldsymbol{W}^{(l)} \boldsymbol{a}^{(l-1)} + \boldsymbol{b}^{(l)} \tag{3.16}$$

$$\boldsymbol{a}^{(l)} = \phi^{(l)}(\boldsymbol{z}^{(l)}), \tag{3.17}$$

where $\boldsymbol{a}^{(l-1)} \in \mathbb{R}^{m^{(l-1)}}$, $\boldsymbol{W}^{(l)} \in \mathbb{R}^{m^{(l)} \times m^{(l-1)}}$, and $\boldsymbol{a}^{(l)} \in \mathbb{R}^{m^{(l)}}$. The nonlinear activation function $\phi(\cdot)$ is applied elementwise, such that:

$$\phi^{(l)}(\boldsymbol{z}^{(l)}) = \phi^{(l)}([z_1^{(l)}, \ldots, z_{m^l}^{(l)}]) = [\phi^{(l)}(z_1^{(l)}), \ldots, \phi^{(l)}(z_{m^l}^{(l)})] \tag{3.18}$$

Therefore, the output of a given neural network layer can be seen as a transformation of its input that captures various interactions of the original inputs. Typically, we speak of deep neural networks when there are more than three hidden layers present [23]. The capacity of the neural networks to approximate any functions, especially non-convex, is directly the

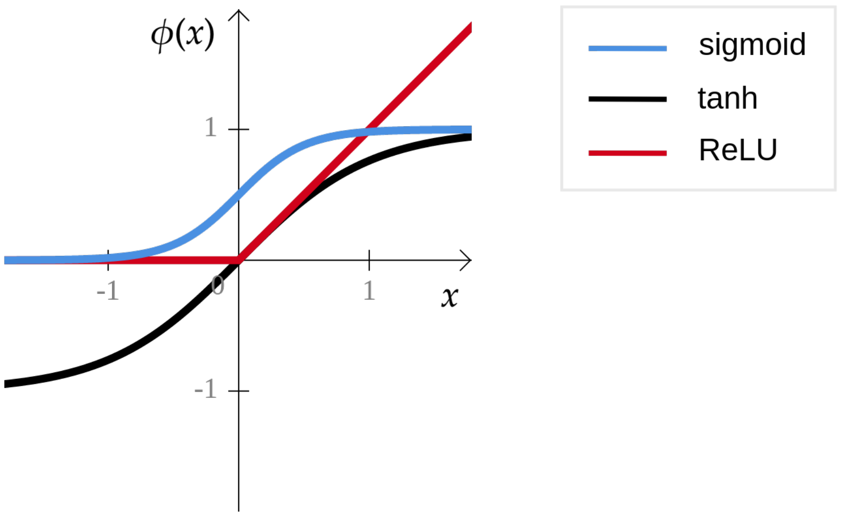


Figure 3.6: Nonlinear activation functions.

result of the nonlinear activation functions $\phi(\cdot)$. Sigmoidal (i.e., s-shaped) functions are historically the most common activation functions. Examples either include the sigmoid function:

$$\phi(z) \;=\; \sigma(z) \;=\; \frac{1}{1+e^{-z}} \tag{3.19}$$

or the hyperbolic tangent function:

$$\phi(z) \;=\; \tanh(z) \;=\; \frac{1-e^{-2z}}{1+e^{-2z}} \tag{3.20}$$

The sigmoid activation function maps any real number to the interval of $[0,1]$. Accordingly, the activation value can be interpreted as the probability of the neuron to be on. The hyperbolic tangent can be regarded as the linear transformation of the sigmoid function into the interval $[-1,1]$ (see Figure 3.6). Despite the loss of a probabilistic interpretation, the $\tanh(\cdot)$ function is often preferred in practice, due to better empirical performance [23].

However, in recent years, sigmoidal nonlinearities have been replaced to a large extent by Rectified Linear Units (ReLUs) [72]. The ReLU is defined as:

$$\phi(z) \;=\; \mathrm{ReLU}(z) \;=\; \max(0,z). \tag{3.21}$$

In fact, the adoption of ReLU may be considered one of the most important milestones of the recent deep learning revolution. It allows training MLPs with a larger number of hidden layers as it promotes sparse activation patterns and better gradient flow [72].

In classification problems, the neurons of the output layer should provide class posterior probabilities. For this reason, the activation function that is typically used in the last layer of a neural network for classification tasks is the softmax function [74]. It is a normalized exponential function, which is formally defined as:

$$\sigma(\boldsymbol{z})_i = \frac{\exp(z_i)}{\sum_{j=1}^{C}\exp(z_j)}, \tag{3.22}$$

where $\sigma(\boldsymbol{z})_i$ corresponds to the output of the i-th neuron in the output layer, which represents the probability of a given instance belonging to the class i, and C is the total number of classes.

3.2.2 Learning process

Overall, an MLP represents a function:

$$\boldsymbol{a}(\cdot,\theta) : \mathbb{R}^D \rightarrow \mathbb{R}^C, \boldsymbol{x} \mapsto \boldsymbol{a}(\boldsymbol{x};\theta), \tag{3.23}$$

where $\boldsymbol{a}(\cdot;\theta)$ represents the output vector of the neural network, and $\theta = \{\boldsymbol{W}^l \cup \boldsymbol{b}^l, l = 1,...,L\}$ summarizes all the trainable parameters of the network, which typically include the connection weights $\boldsymbol{W}$ and the bias $\boldsymbol{b}$.

The learning process (or training) of a neural network is accomplished by adjusting the parameters of the model θ to minimize a given loss (cost) function which can be interpreted as an error measure between the output of the neural network $\boldsymbol{a}(\cdot,\theta)$ and the desired target output $\boldsymbol{y} \in \mathbb{R}^C$. The most popular choice for classification problems is the categorical cross-entropy:

$$\mathcal{L}_{\text{classification}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{y}_i^\top \log \boldsymbol{a}(\boldsymbol{x}_i;\theta), \tag{3.24}$$

where N is the number of training examples. For regression tasks, the mean-squared error (MSE) is the typical choice:

$$\mathcal{L}_{\text{regression}}(\theta) = \frac{1}{N\,C}\sum_{i=1}^{N}(\boldsymbol{y}_i - \boldsymbol{a}(\boldsymbol{x}_i;\theta))^2. \tag{3.25}$$

The optimization of the neural network's trainable parameters is performed using a form of gradient descent. For brevity, the discussion will be limited to Stochastic Gradient Descent (SGD). However, in recent years several alternatives of gradient-based optimization techniques, such as ADAM [97] or ADAGRAD [56], have been proposed. SGD iteratively evaluates the partial derivatives of the loss function with respect to all trainable parameters in the network using backpropagation [171]. Partial derivatives are computed on mini-batches $\mathbb{X}_i$ randomly sampled from the overall training dataset $\mathbb{X}$. In the context of SGD, one pass through all mini-batches is called an epoch, and processing a single batch is referred to as an iteration. The weights are then updated as:

$$\Delta\theta^t = \alpha \frac{\partial J(\theta^t, \mathbb{X}_i)}{\partial \theta^t} \tag{3.26}$$

$$\theta^{t+1} \leftarrow \theta^t - \Delta\theta^t \tag{3.27}$$

where θ^t represents the parameters at epoch t, and α is a hyperparameter called learning rate, which defines how large update steps are performed in the optimization space.

3.2.3 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a particular type of deep neural network, specially designed to deal with grid-like data, such as time-series (1D-grid), images (2D-grid), and video (3D-grid) [109]. As noted earlier in section 3.2.1, a standard fully connected neural network is characterized by a hierarchy of fully connected layers, in which all neurons in a given hidden layer are connected to all neurons in the previous layer. As a result of this full connectivity, the number of trainable parameters substantially increases with the input dimensions. In addition, the local structure of the data is lost [74].

Taking images as an example, CNNs make use of the convolution operation in order to encode the spatial information between the neighboring pixels of an image, but also to reduce the overall complexity of the model. In practice, a neural network is referred to as a CNN when at least one of the fully connected layers is replaced by a convolutional layer. A major difference is that the neurons within such convolutional layers display a 3D arrangement. That is, neurons are organized into three dimensions, regarding the spatial dimensionality of the input (i.e., the height and the width) and the depth of the activation volume. In addition, neurons are only connected to a small region of the layer preceding it [109, 74, 227].

Overall CNN architecture

The architecture of a CNN typically comprises three different types of layers, namely (i) convolutional layers, (ii) pooling layers, and (iii) fully connected layers [227]. Although there are no strict rules on how to structure these layers, CNNs are commonly composed of two main blocks. The first block is commonly referred to as *feature extraction*, which typically consists of alternating convolutional and pooling layers. The activations after the

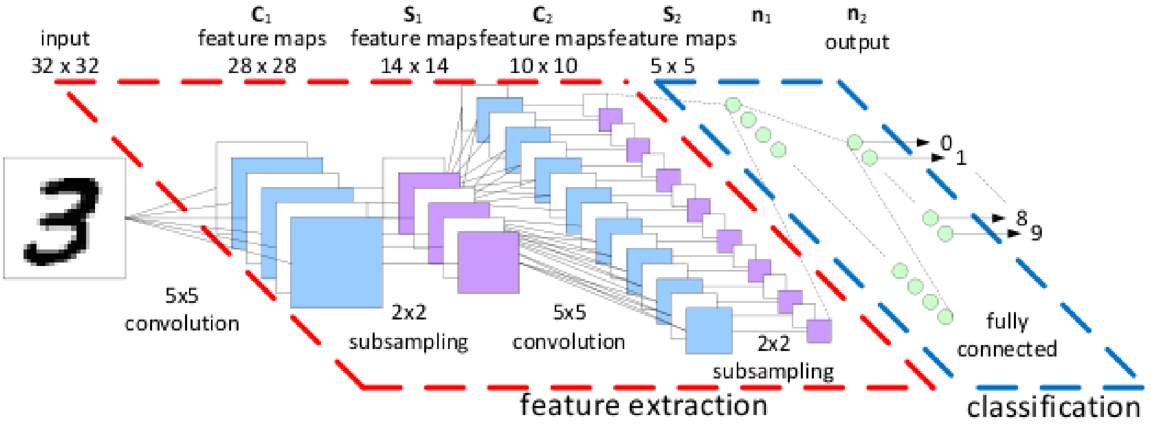


Figure 3.7: Typical architecture of a Convolutional Neural Network for classification tasks (adapted from [164]).

convolution (and pooling operations) are stored in feature maps. The second block, the so-called *classification* block, is composed of fully connected layers in order to provide a prediction based on the feature maps produced by the convolutional layers. This is illustrated in Figure 3.7.

Convolutional layer

As the name suggests, convolutional layers play a vital role in how CNNs operate. In a convolutional layer, the input data is convolved with a set of kernels, which can be seen as learnable filters [109]. Therefore, every convolution operation will produce a 2D activation map (feature map). Specifically, the activation at spatial location (i, j) in layer l is computed as:

$$a(i,j)^l = \phi\left(\sum_{m=0}^{K-1}\sum_{n=0}^{K-1} W_{m,n}a^{l-1}_{i+m,j+n} + b^l\right), \tag{3.28}$$

where $\sigma(\cdot)$ is the nonlinear activation function, b^l denotes the bias, $\boldsymbol{a}^{l-1}$ corresponds to the incoming activation map, and $\boldsymbol{W}$ is a trainable filter with a kernel size of $K \times K$ pixels. Therefore, each kernel will have a corresponding 2D activation map, which are stacked together along the depth dimension to form the complete output volume of the convolutional layer.

The convolution operation introduces a set of assumptions that allow a considerable reduction in weight parameters [109, 74, 227]:

- *Local connectivity*: every neuron in a convolutional layer is only connected to a small region of the input volume, the so-called receptive field, which is related to the filter size. This way, neurons are capable of combining local neighborhood information to extract elementary visual features, such as edges or corners. These features are then combined by the subsequent convolutional layers in order to detect high-level features.

- *Parameter sharing*: it works based on the assumption that if a feature detector (filter) is useful in some region of an image, then it is likely to be useful in another region. This means that each activation map within the output volume shares the same weights and bias, which results in a massive reduction in the total number of trainable parameters.

Furthermore, the convolution operation is not dependent on image size and introduces equivariance to translation (i.e., a translation in the input activations will result in the same translation in the output activations).

Pooling layer

Pooling or subsampling is another important operation in the context of CNNs [109, 227]. Pooing layers are in charge of computing summary statistics over local regions of the input volume (or feature maps), in order to make the feature representations and the subsequent prediction robust to small variations in the input space. One of the most commonly used pooling strategies is called max-pooling, where only the maximum activation value is kept for each local region in the feature maps. A common practice is to use a 2×2 local region, which reduces the spatial dimensions of the input volume in half.

3.2.4 Autoencoders

Autoencoders are a particular type of neural network that, in the course of the training process, aim to learn latent feature representations from the input data. Although their most traditional application was dimensionality reduction or feature learning, the autoencoder concept has recently become widely used for learning generative models.

Standard Autoencoder

An autoencoder consists of two major components, an encoder f, parameterized by θ_e, which maps the input data $\boldsymbol{x} \in \mathbb{R}^D$ to a latent representation (or code) $\boldsymbol{h} \in \mathbb{R}^M$, such that:

$$\boldsymbol{h} = f(\boldsymbol{x}; \theta_e), \tag{3.29}$$

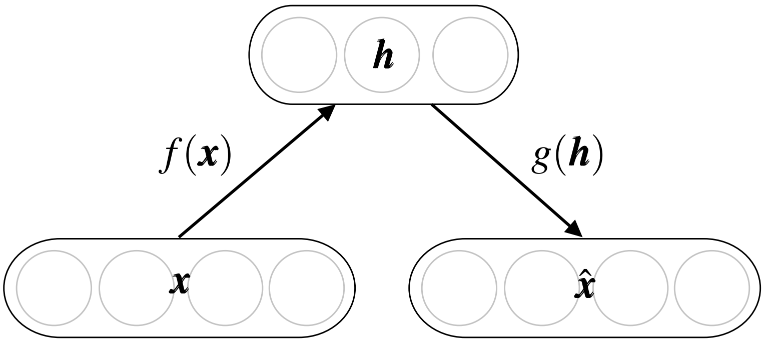


Figure 3.8: Standard Autoencoder.

and a decoder g, parameterized by θ_d, which maps the feature vector $\boldsymbol{h}$ back from the feature space $\mathbb{R}^M$ to the input space $\mathbb{R}^D$, such that:

$$\hat{\boldsymbol{x}} = g(\boldsymbol{h}; \theta_d). \tag{3.30}$$

Both encoder and decoder functions are implemented as neural networks, commonly represented by one or more fully connected or convolutional layers [23, 227]. Figure 3.8 illustrates the simplest form of an autoencoder, which simply consists of a single-layer encoder, a hidden layer, and a single-layer decoder.

The parameters θ_e and θ_d are learned by optimizing the entire network for the task of reconstruction, so that the output of an autoencoder resembles its input. Therefore, the MSE, as previously discussed in section 3.2.2, appears as a natural choice of such reconstruction loss function [23, 227]. However, in this case, the desired target values $\boldsymbol{y}$ used in (3.25) should be replaced by the input $\boldsymbol{x}$. This means that the learning process of autoencoders does not require labeled data. For this reason, autoencoders are considered unsupervised or self-supervised models.

Since the primary goal of an autoencoder is to learn useful representations of the data, the dimension of the latent representation M should not be larger than the input (and output) dimension D, unless regularization techniques are employed. The main reason is that, if $M \geq D$, the model could end up with just an identity mapping, achieving a perfect reconstruction, but would not produce good representations. The original way to overcome such trivial solution is to force the encoder to perform dimensionality reduction (i.e., $M < D$), so that the autoencoder needs to learn how to compress and decompress the data, thereby reducing redundancy. However, dimension reduction is sometimes undesirable, as it may be useful to have representations with a much larger dimensionality than the input data in order to disentangle many concepts [227].

To tackle the problem of ending up with trivial models, several extensions of the standard autoencoder have been proposed. Notable examples are the regularized autoencoders (e.g., sparse, denoising, and contractive autoencoders), which proved effective in learning representations for subsequent classification tasks [219], as well as the variational autoencoders, with their potential application as generative models [98]. The variational autoencoder, which was widely explored in this thesis, is fully described in the following.

Variational Autoencoder and Conditional Variational Autoencoder

The Variational Autoencoder (VAE) [98] is a generative model that directly estimates the probability density function $p(\mathbf{X})$ of the data $\mathbf{X}$. The VAE is a latent variable model which is trained to maximize a lower bound for the log-likelihood of the data. Specifically, introducing a random variable $\mathbf{z}$ with prior distribution $p(\mathbf{z})$ and writing the data probability density as $p(\mathbf{X}) = \int_{\mathbf{z}} p(\mathbf{X}|\mathbf{z})p(\mathbf{z})\,d\mathbf{z}$ it is easy to verify that the equality:

$$\begin{aligned} \log p(\mathbf{X}) - D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{X})||p(\mathbf{z}|\mathbf{X})) &= \\ = \mathbb{E}_{\mathbf{z}\sim q}[\log p(\mathbf{X}|\mathbf{z})] - D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{X})||p(\mathbf{z})) & \qquad (3.31) \end{aligned}$$

holds for any distribution $q(\mathbf{z}|\mathbf{X})$. Here, D_{KL} denotes the Kullback-Leibler (KL) divergence, which is minimum and equal to zero when the two distributions coincide. Thus, given the non-negativity of the KL divergence, the right-hand side of (3.31) is a lower bound for $\log p(\mathbf{X})$ and therefore it may be used as the training objective for the VAE:

$$\min_{\theta_e,\theta_d} -\mathbb{E}_{\mathbf{z}\sim q}[\log p(\mathbf{X}|\mathbf{z};\theta_e)] + D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{X};\theta_d)||p(\mathbf{z})), \qquad (3.32)$$

where $q(\mathbf{z}|\mathbf{X};\theta_e)$ is the probability distribution mapping data $\mathbf{X}$ to latent codes $\mathbf{z}$ (encoding) and $p(\mathbf{X}|\mathbf{z};\theta_d)$ is the posterior probability of data $\mathbf{X}$ given latent codes $\mathbf{z}$ (decoding) - see Figure 3.9. Here, θ_e and θ_d summarize the parameters of the encoder and decoder networks, respectively. A usual choice is to have these networks parameterizing Gaussian distributions and to set the prior $p(\mathbf{z})$ to a standard Gaussian. Specifically,

$$\begin{aligned} q(\mathbf{z}|\mathbf{X};\theta_e) &= \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_e(\mathbf{X};\theta_e), \mathrm{diag}(\boldsymbol{\sigma}_e^2(\mathbf{X};\theta_e))), & (3.33) \\ p(\mathbf{z}) &= \mathcal{N}(\mathbf{z}|0,\boldsymbol{I}), & (3.34) \\ p(\mathbf{X}|\mathbf{z};\theta_d) &= \mathcal{N}(\mathbf{X}|\boldsymbol{\mu}_d(\mathbf{z};\theta_d), \sigma^2\boldsymbol{I}). & (3.35) \end{aligned}$$

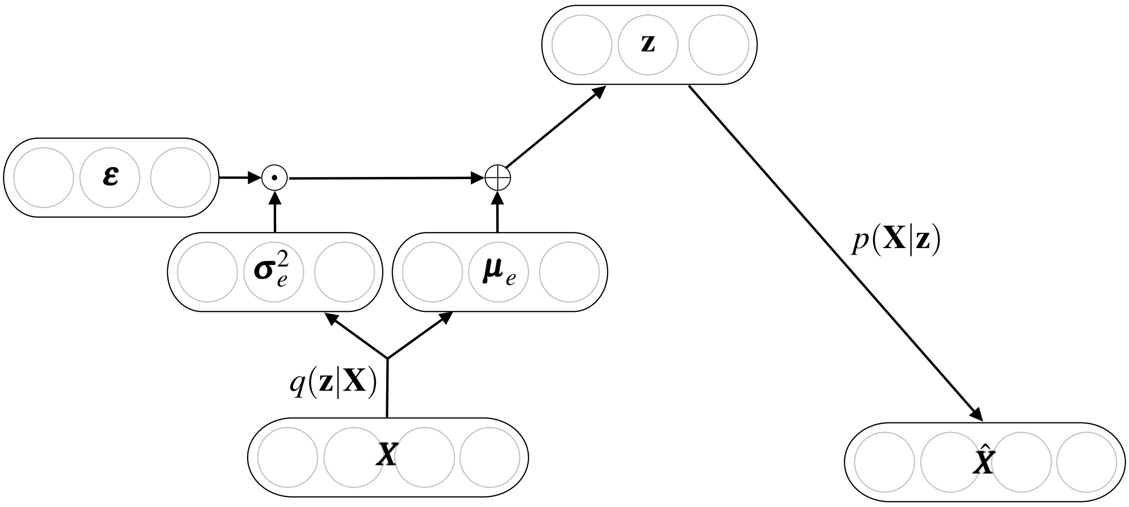


Figure 3.9: Variational autoencoder.

Under this setting and approximating the expectation in objective (3.32) with a Monte Carlo estimation with one sample, the objective may be rewritten as:

$$\min_{\theta_e,\theta_d} ||\mathbf{X} - \boldsymbol{\mu}_d(z_0;\theta_d)||^2 + \lambda D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{X};\theta_e)||p(\mathbf{z})), \tag{3.36}$$

where $\boldsymbol{z}_0$ is sampled from $q(\mathbf{z}|\mathbf{X};\theta_e)$, $||\cdot||$ denotes the ℓ^2-norm of a vector (or vectorized matrix) and $\lambda = 2\sigma^2$ is a hyperparameter. Here, the reconstruction error $||\mathbf{X} - \boldsymbol{\mu}_d(z_0;\theta_d)||^2$ resembles the loss of a standard autoencoder, promoting good reconstructions. On the other hand, the KL divergence term prevents the VAE from becoming deterministic. This KL divergence may be computed analytically, since both distributions are Gaussian, and therefore no Monte Carlo approximation is needed. Gradient backpropagation through the random sample $\boldsymbol{z}_0$ becomes trivial using a reparameterization trick [98], where a sample from a standard Gaussian is transformed into a sample from $q(\mathbf{z}|\mathbf{X};\theta_e)$ by a deterministic function:

$$\boldsymbol{z}_0 \;=\; \boldsymbol{\sigma}_e(\mathbf{X};\theta_e) \odot \boldsymbol{\varepsilon}_0 + \boldsymbol{\mu}_e(\mathbf{X};\theta_e), \quad \boldsymbol{\varepsilon}_0 \sim \mathcal{N}(\cdot|0,\boldsymbol{I}), \tag{3.37}$$

where $\odot$ denotes the elementwise product of two vectors.

The Conditional Variational Autoencoder (CVAE) [183] is an extension of the VAE which is trained to maximize a conditional log-likelihood. In its simplest form, a CVAE models the distribution $p(\mathbf{X}|\mathrm{y})$, where y represents some extra information about the data (e.g. image labels). This is attained by conditioning on y every probability in objective (3.32):

$$\min_{\theta_e,\theta_d} -\mathbb{E}_{\mathbf{z}\sim q}[\log p(\mathbf{X}|\mathbf{z},\mathrm{y};\theta_d)] + D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{X},\mathrm{y};\theta_e)||p(\mathbf{z}|\mathrm{y})), \tag{3.38}$$

In practice, usually y is fed as an extra input to both the encoder and decoder networks, while the prior on latent variables is assumed to be independent of y, that is $p(\mathbf{z}|\mathrm{y}) = p(\mathbf{z})$.

3.3 Regularization

In machine learning, particularly in deep learning, overfitting is a major issue that arises during the training process of the models. When the training error of a model keeps decreasing but, on the other hand, the test (or generalization) error starts increasing, the model is said to be "overfitted" [27]. At this point, the model is just learning the distribution of the training data and not generalizing to new and unseen data.

Regularization can be defined as any modification performed either to the learning algorithm or the model architecture, in order to reduce the generalization error, probably at the expense of an increased training error [27, 74]. As detailed in the following subsections, several regularization techniques have been proposed, some of which are based on constraining the parameter values, adding extra terms to the loss function, or enlarging the training set artificially.

In general, regularization techniques are designed to encode some sort of prior knowledge, with a preference towards simpler models to promote generalization.

3.3.1 Parameter Norm Penalties

The underlying idea is to add a parameter norm penalty, $\Omega(\theta)$, to the loss function, J, in order to limit the capacity of the model or, in other words, limit the space of all possible model families [27, 74]. That is, the loss function becomes:

$$\mathcal{L}(\theta) \ := \ \mathcal{L}(\theta) \ + \ \lambda \ \Omega(\theta), \tag{3.39}$$

where λ is a hyperparameter that balances the relative contribution of the norm penalty. Here, θ typically represents only the weights and not the biases. This is due to the fact that the biases require much less data to fit and do not add much variance.

l^1- and l^2-norms are the most common types of regularization. l^2-norm penalty is also commonly refereed to as weight decay. The idea of l^2-norm regularization is to penalize large weights as they tend to result in overfitting [26]. The regularization term is defined as:

$$\Omega(\theta) \ = \ ||\theta||_2^2. \tag{3.40}$$

l^1-norm regularization enforces sparsity of the weights by penalizing the absolute value of the weights:

$$\Omega(\theta) = ||\theta||_1. \tag{3.41}$$

The sparsity property of the l^1-norm is also very useful for model compression and feature selection [27, 74].

3.3.2 Data Augmentation

The simplest way to reduce overfitting and, thereby, increase the generalization capability of a model is to increase the amount of training data [27]. However, in practice, it is common to deal with limited sized datasets as the acquisition of labeled data is too costly.

Data augmentation is the process of artificially enlarging the training set using label-preserving transformations. Dataset augmentation is one of the most widely used regularization techniques in computer vision, since image data contain several sources of variably, many of which can be easily artificially generated. The common practice is to apply both geometric (e.g., translations, rotations, scalings, etc) and color transformations to the images and, then, train the model using both original and synthetic data [74].

3.3.3 Multi-task Learning

Multi-task learning attempts to improve generalization by solving multiple learning tasks simultaneously while exploiting commonalities and specificities across tasks. The underlying idea is that when part of a model is used for different but somehow related tasks, that part of the model will be constrained towards good values, often yielding better generalization [74].

In practice, multi-task learning is performed by learning multiple related tasks in parallel while using a shared representation. Therefore, the model parameters can be roughly divided into two main classes:

- *Task-specific* parameters, which are optimized for their particular task.

- *Generic* parameters shared across all tasks, which benefit from learning through different tasks.

3.3.4 Early Stopping

The idea behind early stopping is to simply stop the training process once the model begins to overfit the training set. Early stopping is one of the oldest forms of neural network regularization, and it is widely used, in general, when the models are trained with iterative methods, such as gradient descent [26].

In practice, early stopping works as follows. During training, the model is evaluated on a holdout validation set after each epoch. If the performance of the model on the validation dataset does not improve over a certain fixed number of training epochs, then the training process is stopped. This effectively reduces the capacity of the model by reducing the number of steps required to fit the model.

3.3.5 Parameter Tying and Parameter Sharing

In practice, there might be several situations in which it would be beneficial to incorporate some prior knowledge on the kind of dependencies that the model should encode. For instance, suppose that two distinct models were trained to perform related tasks with similar input and output distributions. In such a case, it would be expected that the parameters of both models would be identical. This kind of information can be leveraged through regularization, either by imposing a norm penalty on the distance between the parameters or forcing the parameters to be equal [74].

This is related to the transfer learning concept. Transfer learning aims to extract knowledge from one or multiple source tasks (or domains) and, then, use this prior knowledge when learning a model for a new target task [152]. A common approach is also to penalize the parameter differences between different tasks. More recently, the concept of "partial transfer" was introduced by Fernandes *et al.* [64]. The idea is to transfer high-level properties of the source model instead of the whole model structure. In particular, it encourages parameters to share the same contribution type (e.g., the sign) instead of the actual parameter values.

3.3.6 Sparse Representations

As previously discussed in section 3.3.1, weight decay acts by placing a norm penalty directly on the model parameters. Following this idea, another regularization strategy is to place a penalty on the activations of the units in a neural network, encouraging their activations to be sparse. This leads to representational sparsity, which can be obtained by the same sort of mechanisms used in the parameter regularization (i.e., l_1-norm). This procedure will indirectly impose a penalty on the model parameters [74].

3.3.7 Dropout

Dropout is a computationally inexpensive, yet powerful regularization technique for neural networks [189]. During training, individual neurons are either "dropped out" or kept according to a defined probability p, so that a reduced network is left. Note that, at each training iteration, only the reduced network is trained on the data. Then, the removed units are reinserted into the network with their original weights. This procedure forces individual neurons to learn features without co-adapting to each other.

At test time, all the units are used to compute the prediction. However, they have to be appropriately re-scaled according to their corresponding dropout rate. Therefore, dropout can be interpreted as a form of model averaging over all possible instantiations of the model.

3.4 Summary

This chapter presented an overview of the pattern recognition techniques used throughout this thesis. First, the standard pattern recognition pipeline was introduced, in which two traditional feature extraction approaches (i.e., LBP and Gabor filters) and a shallow machine learning algorithm (i.e., SVMs) were detailed. Second, the deep learning techniques that constitute the backbone of this thesis were thoroughly described. Finally, the most commonly used regularization strategies in machine learning, especially in deep learning, were presented.

Chapter 4

An Overview on Sign Language Recognition

SLR is an appealing topic in modern society because such systems can ideally be used to reduce the communication barriers that exist between deaf and hearing people. Hence, several works have been proposed for the development of SLR systems for different sign languages. However, as SLR is a multidisciplinary challenging task (i.e., for signs capturing methods, machine learning classifiers, sign language, and human action understanding) there are still many opportunities for research and improvement. This chapter aims to present an overview of some of the most relevant contributions in the SLR research field.

4.1 Data Acquisition Process

As illustrated in Figure 4.1, the first step to take into account in an SLR system is the data acquisition process. Regarding the data acquisition process, SLR systems can be roughly classified into two main groups: (i) SLR systems based on wearable hardware equipment's, and (ii) SLR systems based on computer vision [13].

The former category of SLR systems resorts to the utilization of data gloves or similar equipment's that store information of the hand and fingers position and their relative movement. It is the example of the work of Fang *et al.* [63] that proposes a system for the recognition of Chinese sign language, in which data gloves are used to collect information about the hand shape, orientation, position, and movement trajectory. Typically, these systems allow recording gesture information with high accuracy without any kind of pre-processing. However, SLR systems based on wearable hardware equipment's are not suitable for real-world scenarios, since the signer has to wear cumbersome devices which

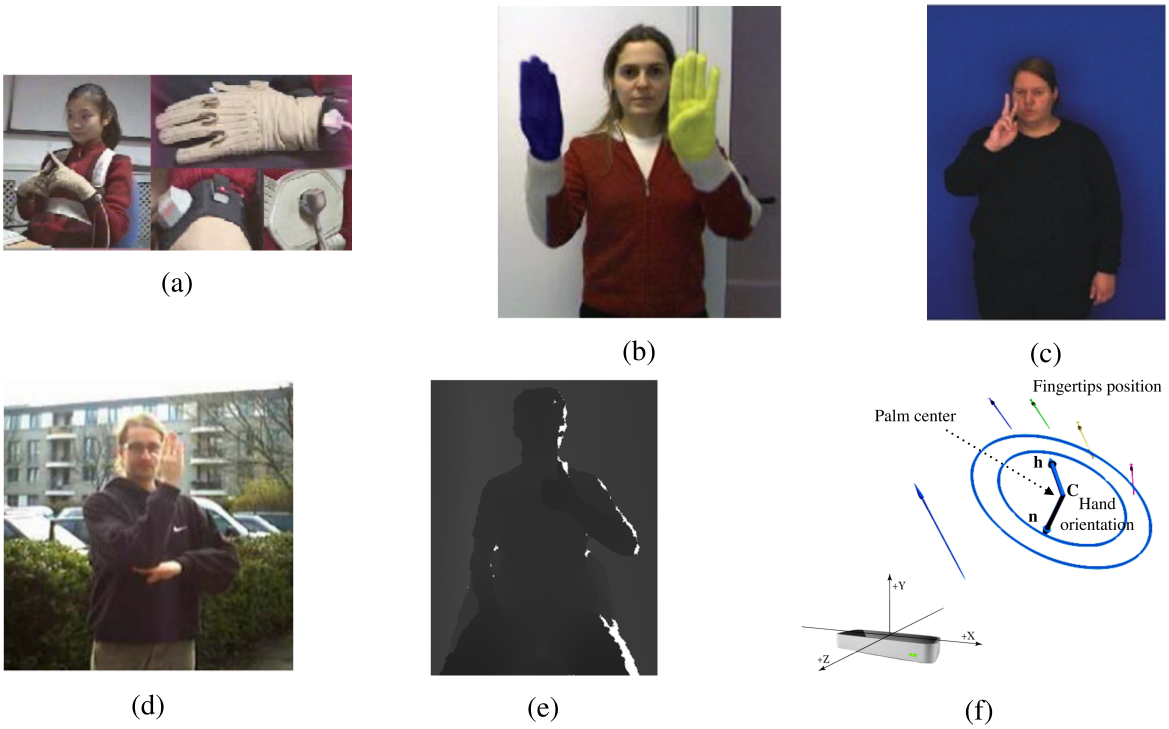


Figure 4.1: Data acquisition in SLR: (a) Data gloves, (b) Visual markers, (c) Controlled scenarios - dark background, (d) Uncontrolled environments, (e) Depth information and (f) Leap Motion.

might affect the natural signing movement. In addition, non-manual signals, that convey important meaning in sign language, are neglected [63, 13].

The second group of SLR systems, i.e., vision-based, rely on computer vision algorithms for detecting hands and, then, extracting meaningful information from them. This kind of systems eases the interaction with deaf people since there is no need to wear any extra hardware and, therefore, such systems are preferred for real-world applications. Nevertheless, vision-based data is harder to handle in terms of feature extraction because of the difficulties of correctly segmenting the hands and face in uncontrolled scenarios. Attempting to avoid those difficulties, most of the proposed works in the literature impose several constraints to the signers, use data recorded in controlled scenarios and some of them use visual markers (e.g., different colored gloves on each hand or colored markers on each finger). Illustrative examples of SLR systems that make use of visual markers can be found in [81, 141]. In [80, 38, 242, 150], the gestures are recorded in studios with a uniform and dark background and signers are wearing clothes with non-skin color and long sleeves. Fewer works attempt to address the problem of SLR in uncontrolled environments without background restrictions [226].

The recent introduction of Kinect-like sensors has put more emphasis on the development of multimodal SLR systems, especially those using RGB-D data. Several works have been using depth data in order to complement the color information [216, 48, 132]. Moreover, the recent introduction of the Leap Motion device has launched new research lines for SLR. The Leap Motion controller was specially designed for hand gesture recognition. When compared with Kinect-like depth cameras, it produces a far more limited amount of information within a smaller field of view (i.e., only a few key-points instead of the complete depth description). On the other side, Leap Motion directly provides the 3D spatial positions of the fingertips and the hand orientation with quite accuracy (according to [234], its accuracy is of about $200\mu m$) (see Figure 4.1f). In this regard, SLR researchers have been exploring the complementary characteristics between Kinect and Leap Motion data. It is the example of the works recently proposed by Marin *et al.* [131, 132] and Kumar *et al.* [107], in which the input data from both Kinect and Leap Motion sensors are combined for more accurate gesture recognition.

Since the focus of this thesis is vision-based SLR, in the following subsections, we will only discuss the most relevant state-of-the-art techniques used in this area. This field can be divided into three main areas, as presented in Figure 4.2: (i) spatial segmentation and tracking, (ii) feature extraction, and (iii) sign recognition. Section 4.2 presents an overview about segmentation and tracking in SLR. In Section 4.3, the most common features, used to represent gestures, are described. Section 4.4 presents the most widely used recognition techniques. Finally, a description of some benchmark datasets in SLR finishes this chapter.

4.2 Segmentation

In the SLR context, segmentation is the process of extracting the objects of interest from the images. The objects of interest that are typically considered are the hands, face, and body. For instance, if an SLR system considers both manual and non-manual signals for recognition, then hands and face have to be extracted from the images. The segmentation task can be divided into two distinct areas: (i) detection, and (ii) tracking. Detection is the task of localizing the objects of interest in single images independently. Tracking aims to predict the new location of the object in the next frame using previous or posterior information of the position of the objects under analysis.

Most of the proposed works in SLR use color information to detect the objects of interest in the images [38, 12, 89, 65, 162]. The color information is used to build a skin color model that allows the separation between skin color pixels (hands and face) and the background. It is the example of Adithya *et al.* [12] that proposes a hand segmentation method based on the YCbCr color space. Skin color pixels are detected by applying a

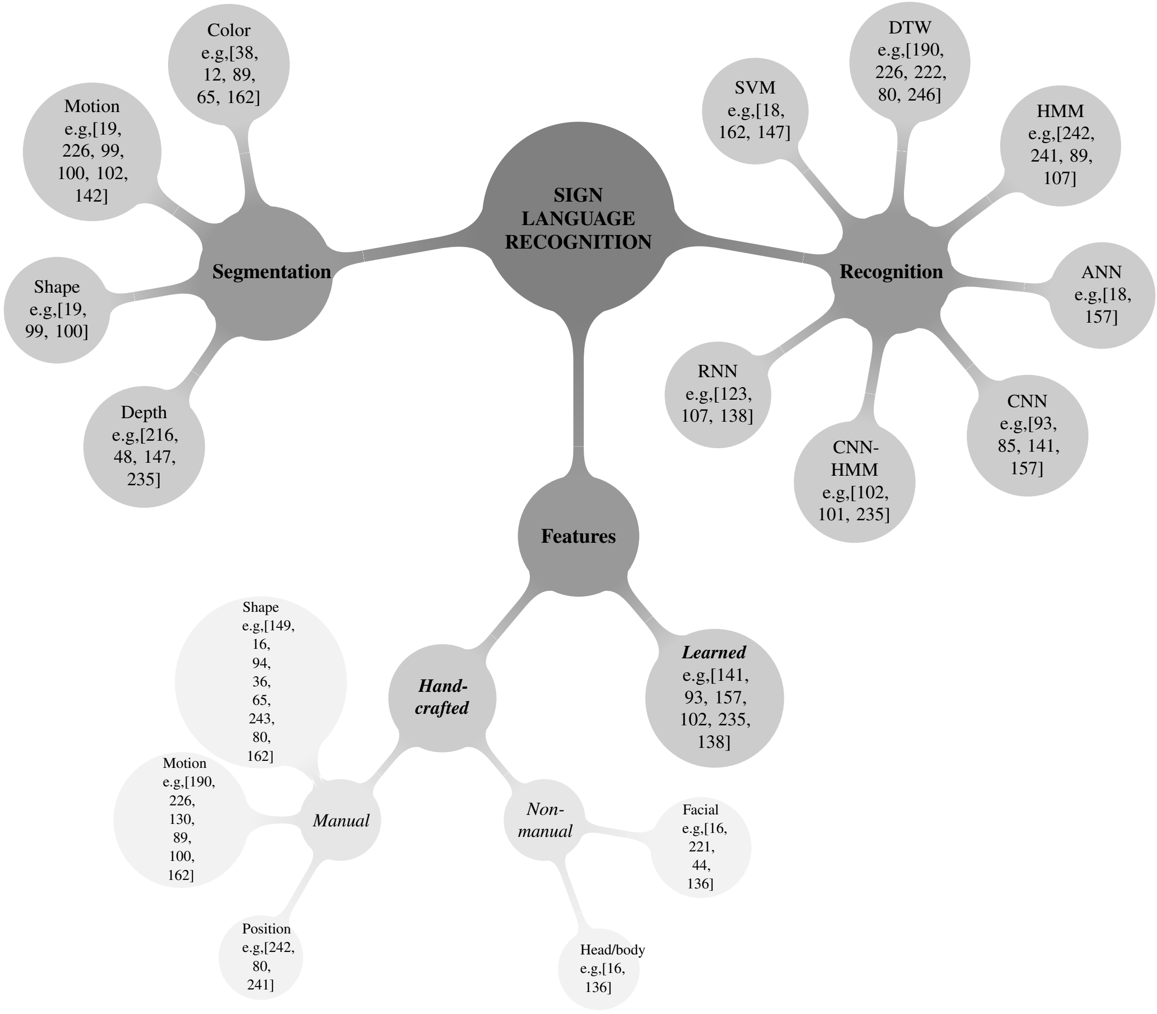

SIGN LANGUAGE RECOGNITION
Segmentation
Color e.g,[38, 12, 89, 65, 162]
Motion e.g,[19, 226, 99, 100, 102, 142]
Shape e.g,[19, 99, 100]
Depth e.g,[216, 48, 147, 235]
Recognition
SVM e.g,[18, 162, 147]
DTW e.g,[190, 226, 222, 80, 246]
HMM e.g,[242, 241, 89, 107]
ANN e.g,[18, 157]
CNN e.g,[93, 85, 141, 157]
CNN-HMM e.g,[102, 101, 235]
RNN e.g,[123, 107, 138]
Features
Hand-crafted
Learned e.g,[141, 93, 157, 102, 235, 138]
Manual
Shape e.g,[149, 16, 94, 36, 65, 243, 80, 162]
Motion e.g,[190, 226, 130, 89, 100, 162]
Position e.g,[242, 80, 241]
Non-manual
Facial e.g,[16, 221, 44, 136]
Head/body e.g,[16, 136]

Figure 4.2: Categorization of the related work.

thresholding technique based on the skin color distribution in YCbCr color space. The work of Cooper and Bowden [38] proposes a detection algorithm based on a skin color model obtained using the color information of the face. In a first step, the face is detected using the Viola-Jones algorithm [220] and, then, a Gaussian model of the signer skin is created. Recently, Fernando *et al.* [65] proposed a heuristic skin color segmentation method in both HSV and YCbCr color spaces. The advantages of color-based segmentation methods are mainly related to their computational simplicity and the invariance of color descriptors to geometric transformations (e.g., translation and rotation). However, in real-world scenarios, skin color models demonstrate several problems. These models may detect background objects as skin, especially when there are skin-colored objects in the background (e.g., other people). Another problem related to skin color models is the difficulty of dealing with the superposition, that often occurs in sign language, between both hands and/or face.

Therefore, skin color models are often used by imposing several restrictions to the signer and the background environment. For instance, in [35] and [141], the signers have to wear long-sleeved clothing to cover other skin regions such as the arms and colored gloves in the hands, respectively.

Attempting to overcome those limitations, some works use motion and shape information along with the color [99, 19]. Other works [226] suggest the utilization of a background subtraction model to filter the background in the sequence. Awad *et al.* [19] propose a method to detect the skin that combines color, motion, and position in order to segment the two hands and the face. Afterwards, a regular Kalman filter is used to keep track of the skin blobs obtained in the segmentation. The work of Kishore *et al.* [99] addresses the problem of SLR under unconstrained environments (e.g., cluttered backgrounds, different lighting conditions, and occlusions). Both hands and the head of the signer are segmented and tracked using active contour models. The minimization energy of the active contour model is defined based on the signers' hands and head skin color, texture, boundary, and prior shape information. In the same research line, Kishore and Prasad [100] proposed a tracking-by-detection approach in which an active contour level set model is used for shape segmentation and, then, an optical flow algorithm is employed for hands tracking. Although motion and shape information can be used to increase segmentation accuracy, there are still unsolved issues [35]. Motion information is typically used under the assumptions that the hand is the only moving object among the skin-colored regions, and the movement of the hand has a constant velocity. The problem of using shape information comes from the fact that the hand is a non-rigid object with a very high degree of freedom. Therefore, in order to take shape information into account, several hand shapes restrictions are often used.

Another segmentation challenge is occlusions. In some sign language gestures, this may occur by the superposition between both hands or between hands and face. The works presented in [99, 80] explore the use of active contour models to distinguish overlapped objects. Bergh and Gool [216] show that depth information can be used together with the color information to increase the recognition accuracy, especially when there are superimposed gestures (see Figure 4.3). Dominio *et al.* [48] also presented a hand segmentation method that combines these two types of information.

Tracking the objects of interest in SLR is commonly carried out using a Kalman filter or a particle filter. Kalman filters are linear systems with Gaussian noise assumption in which the motion of each hand is approximated by a constant velocity or a constant acceleration motion model. As an alternative, particle filters can also be used since they work better under nonlinear and non-Gaussian conditions [35]. Apart from these algorithms, some works

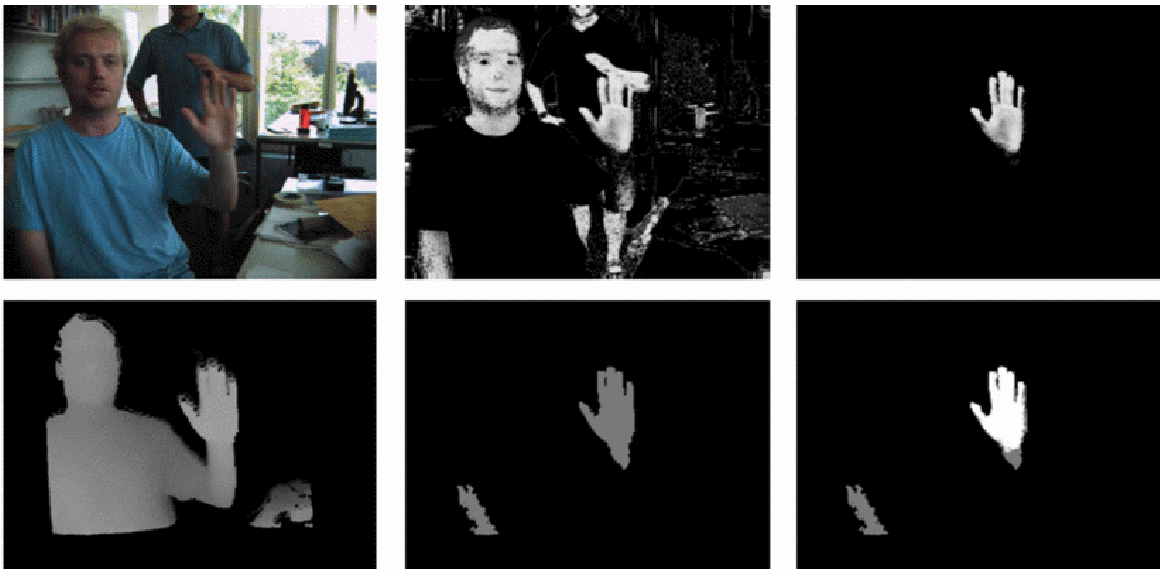


Figure 4.3: Usage of depth for segmentation [216]: RGB image (top left), depth image (bottom left), skin color probability (top middle), depth image after the threshold (bottom middle), skin color probability limited to foreground pixels (top right), and segmented hand (bottom right).

propose the usage of several dynamic programming approaches [50, 102, 142]. However, all of these methods need a dynamic model for the hand motion, which is hard to estimate.

Other works [148, 213, 186, 241, 235] addressed both detection and tracking tasks by using the skeletal joint information available on the Kinect software development kit (SDK), which directly provides the positions of twenty joints of the user's body.

Segmentation is one of the most critical steps in SLR since its accuracy determines the eventual success or failure of the sign recognition stage. In a single-modality scenario, segmentation is still a very challenging task, and most of the methods impose strong assumptions regarding background and signer's clothing. The introduction of low-cost depth sensors such as Kinect has made the segmentation task simpler. The Kinect sensor comes with its associated SDK that enables to acquire the depth map and the 3D skeleton of the whole body, which in combination with the color information promoted the development of more robust sign segmentation methods. Therefore, the possibility of using multimodal data is of paramount importance for the segmentation task.

4.3 Feature Extraction

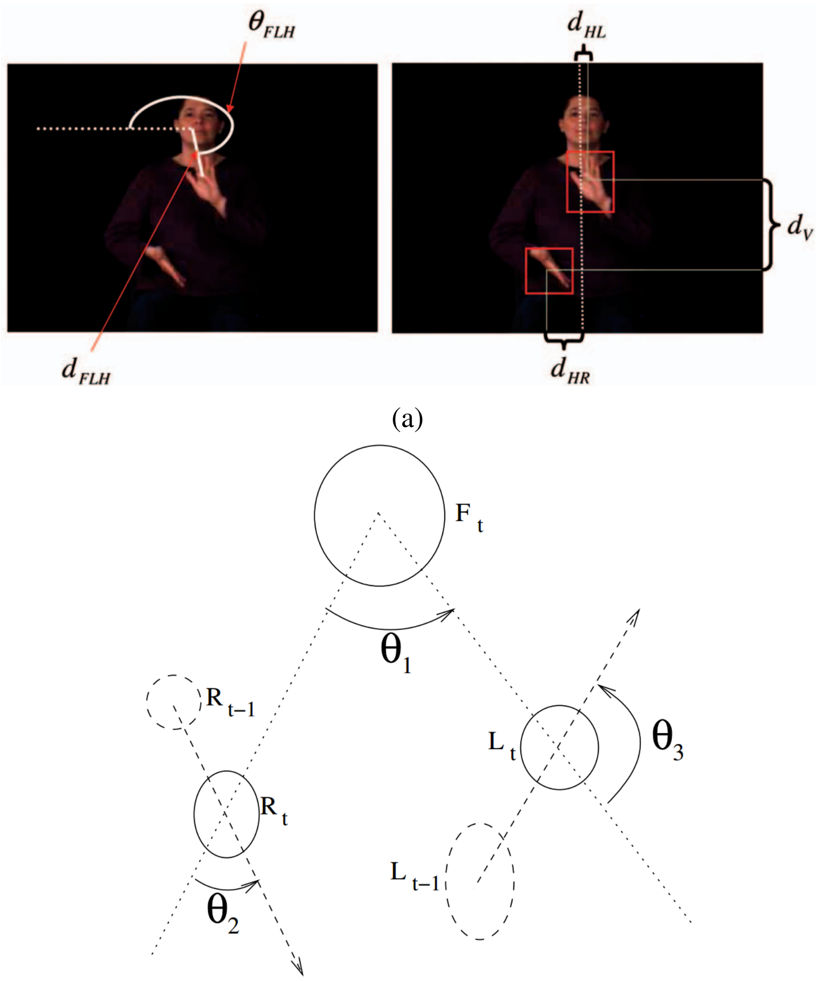
Sign language communication is performed by means of two main components: manual and non-manual signs. Manual signs are the basic components of sign language. These can be divided into hand shapes, motion, and position with respect to body parts. Non-manual signs include facial expressions as well as the movement of the head and body.

The vast majority of SLR works seek to manually extract a set of features describing either the manual or the non-manual components of sign language and, then, build a classifier on top of those features. However, recent advances in deep learning have been promoting the development of SLR methods based on deep neural networks. Deep neural networks can jointly learn high-level feature representations and the classifier directly from the data, avoiding designing hand-crafted features. The following subsections describe the most relevant hand-crafted feature extraction techniques to describe both manual and non-manual signs. The SLR methodologies based on deep learning will be detailed later on Section 4.4.

4.3.1 Manual features

The manual features used for sign language recognition can be classified into three main groups: (i) handshape, (ii) hand motion, and (ii) hand position with respect to the body [35]. The first studies in SLR literature started with the identification of static hand shapes. For the analysis of hand shapes, most of the works available in the literature use appearance-based methods by analyzing a 2D hand image. Typically, the hand is first segmented and, then, features are either extracted from the grey-scale segmented hand or the binary segmented hand. These features include the height, width, area, angle, hand contour, convex hull, image moments, and orientation histograms of the hand [149, 31, 16, 94, 36, 65]. It is the example of the work proposed by Chang *et al.* [36], in which Zernike Moments and Pseudo-Zernike Moments are extracted from the binary segmented hands for the identification of 6 hand gestures. More recently, Fernando *et al.* [65] proposed the utilization of Hu moments with height to width ratio filtration. As Hu moments are rotation invariant, signs that have similar shapes at different angles might be misclassified. When the ratio is considered, Hu moments remain to be scale-invariant and rotation invariant up to a certain level. Moreover, active contours models or 2D deformation templates can be used to find the hand contour and extract features related to the hand contours and edges [243, 80].

The majority of the gestures in sign language are performed dynamically by including motion with the handshape. For instance, the same handshape performed along with a different hand motion may represent a totally different sign. In this regard, several works stress the importance of extracting hand motion-related features [190, 226, 130, 89, 100]. In [89], an orientation feature is extracted from the trajectory of the centroid of the hand. Von Agris *et al.* [226] proposed the use of regional characteristics as features. After segmenting the hands, a feature vector of 11 features is created. These features correspond to the x and y position, x and y derivatives, hand blobs area, the orientation of the main axis, inertia



(a)

(b)

Figure 4.4: Relative position-based features: (a) features extracted in [242], and (b) features extracted in [80].

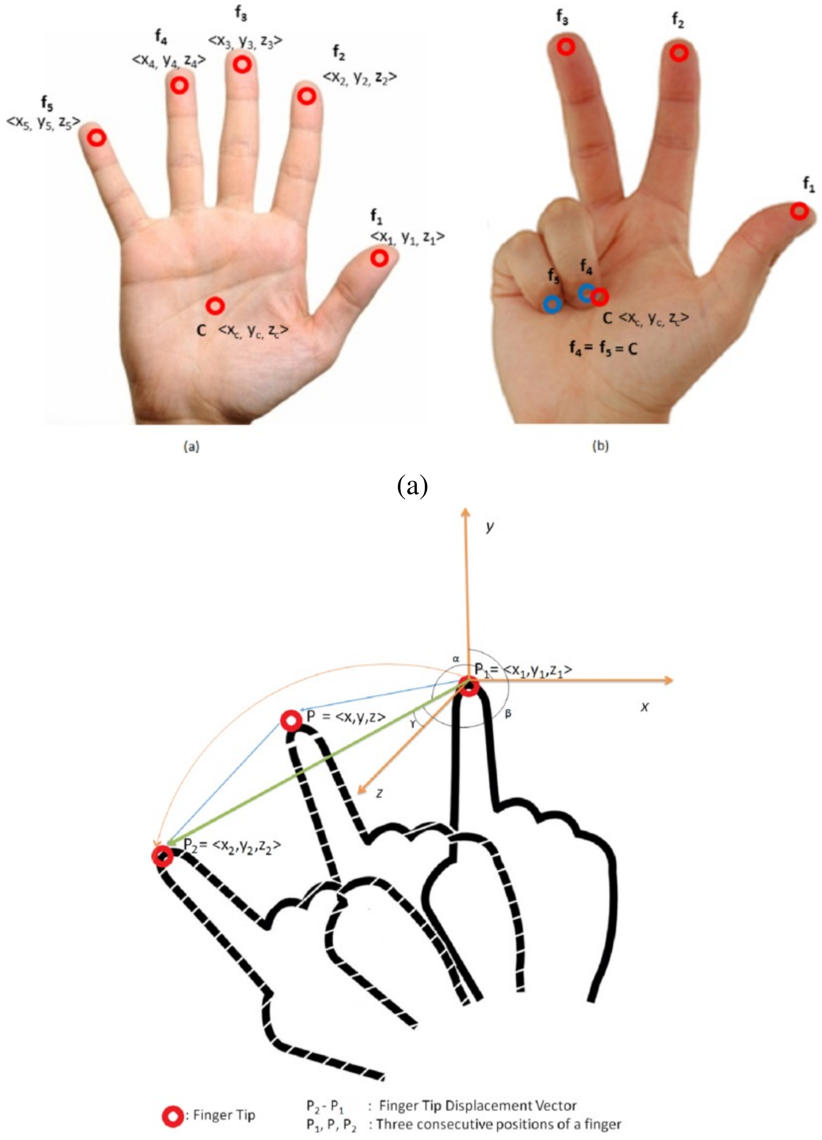
ratio, eccentricity, and compactness. In order to turn the model invariant to the signers' physiognomy, the feature vector is normalized according to the signers' head position and shoulders distance. Kishore and Prasad [100] proposed the utilization of a combined feature vector, comprising motion and shape features. Motion features are given by the velocity vectors of the optical flow of each hand, whereas shape features are extracted by using a level set model. Madani and Nahvi [130] proposed a different approach, in which the hand trajectories are extracted using the Camshift algorithm. Then, the Radon transform is applied in order to recognize the hand trajectories in the Radon space.

Other works also proposed the usage of features regarding the relative position of the hands with other body parts (e.g., head) [242, 80]. Yang *et al.* [242] proposed the combination of motion and relative position features. As illustrated in Figure 4.4a, these relative position-based features are: the angle between the face centre and the left hand centre θ_{FLH}; the distance between the face centre and the left hand centre d_{FLH}; the horizontal distances between the face centre and the centres of the left and right hands d_{HL} and d_{HR}, respectively; and the vertical distance between the centres of both hands d_V. Holden *et al.* [80] uses a similar approach but with a different set of features (see Figure 4.4b). The authors proposed a set of features based on the relative position of the hands with respect to each other and to the head.

With the emergence of low-cost consumer 3D sensors (e.g., Kinect and Leap Motion), some works proposed systems based on 3D information [48, 108]. This new layer of information could be helpful, especially when the position and angles of the fingers are needed with high precision. It is the case of Dominio *et al.* [48] that proposed the usage of several depth-based features for gesture recognition. In the first stage, hands are detected, using both color and depth information, and then segmented into three non-overlapping regions: palm, fingers, and wrist/arm. Afterwards, four subsets of features, which consider depth information are extracted, including distance, elevation, curvature, and palm area features. Recently, Yang *et al.* [241] proposed an extended version of his work presented in [242]. The idea is to still use a set of features regarding the relative position of the hands with respect to the face but in the 3D space.

Attempting to fully describe the input signs, recent SLR works have resorted to multimodal frameworks/models by combining the inputs of more than one device [131, 132, 107]. Marin *et al.* [131, 132] proposed a feature-level fusion approach using Kinect and Leap Motion. Their feature representation comprises a set of features extracted from depth data, such as curvature, correlation, and connected components features, and from Leap Motion mainly based on the position, distance and orientation of the fingertips. Kumar *et al* [107] proposed a similar multimodal framework, but for recognizing dynamic signs. As illustrated in Figure 4.5, they have extracted a set of features from the Leap Motion data, describing the 3D fingertip position and direction.

As stated throughout this subsection, the extraction of manual features is one of the main focus of the SLR researchers. A wide range of manual features has been proposed in the literature, typically involving a tremendous feature engineering work, in order to build feature descriptors robust to the large variations of manual signs. However, most of the available works in the SLR literature use their own set of features without making a comparison with other features previously proposed. In addition, each work typically uses different training and testing scenarios, which hampers the assessment of the quality and applicability of the features. As further detailed in section 4.4, the resurgence of deep neural networks has



(a)

(b)

Figure 4.5: Leap Motion-based features extracted in [107]: (a) 3D fingertip and palm center positions, and (b) fingertip direction.

changed this paradigm, since deep learning allows to develop feature representations and learn machine learning models in a fully integrated way.

4.3.2 Non-manual features

Non-manual signs have an important role in sign language since they are often used along with the gestures either to reinforce or to weaken or, sometimes, to entirely change the meaning of the manual sign. Non-manual signs, especially facial expressions, are also used to convey information indicating feelings on a sign. Moreover, non-manual signs can also be used by themselves, especially to convey negation in a sentence. For instance, the sign HERE in the ASL may mean NOT HERE (negation), HERE (affirmative) or IS HERE (interrogative), depending on the used non-manual sign [35].

Despite the importance of non-manual signs in sign language, there are just a few works in the literature addressing the non-manual component of the sign language. Most of these works attempt to recognize the non-manual signs independently from the manual signs. In [173, 137], facial expressions and in [239, 62], head movements are analysed. The integration of manual and non-manual signs for recognition is almost nonexistent in the SLR literature. Examples of such works can be found in [16, 221, 44, 136]. Most of these existing studies usually simplify the problem. In [16], head movements are analyzed as the non-manual component of the sign language. More precisely, three features derived from the head movement are extracted, namely the quantity of motion and the vertical and horizontal velocity. Then, sign recognition is applied via sequential belief-based fusion of manual and non-manual signs. Das *et al.* [44] attempted to recognize some letters from the ASL alphabet only based on the lip pattern. In particular, histograms of oriented gradients (HOG) are used to describe the lips texture and shape. Michael *et al.* [136] designed a framework based on the analysis of facial expressions for recognizing non-manual markings associated with wh-questions, negative expressions, and topics.

Despite the work already done, the integration of non-manual signs along with manual signs still involves several issues. A major problem is related to the scarceness of SLR databases describing the non-manual component of the signs. It is extremely difficult to capture the facial expressiveness involved in sign languages in a natural environment. Besides, such systems may require multiple cameras, one to record the body movement and another to record the face with high resolution for facial expressions analysis.

4.4 Recognition

Sign language recognition can be divided into two main fields: (i) isolated sign recognition and (ii) continuous sign recognition. The purpose of isolated sign recognition is to recognize individual signs represented by a single gesture. This can be achieved by using either static images or video. Typically, isolated signs represent an alphabet letter, a single word, or a thought. The main idea of continuous sign recognition is the identification of sign language sentences, which are composed of a sequence of several signs.

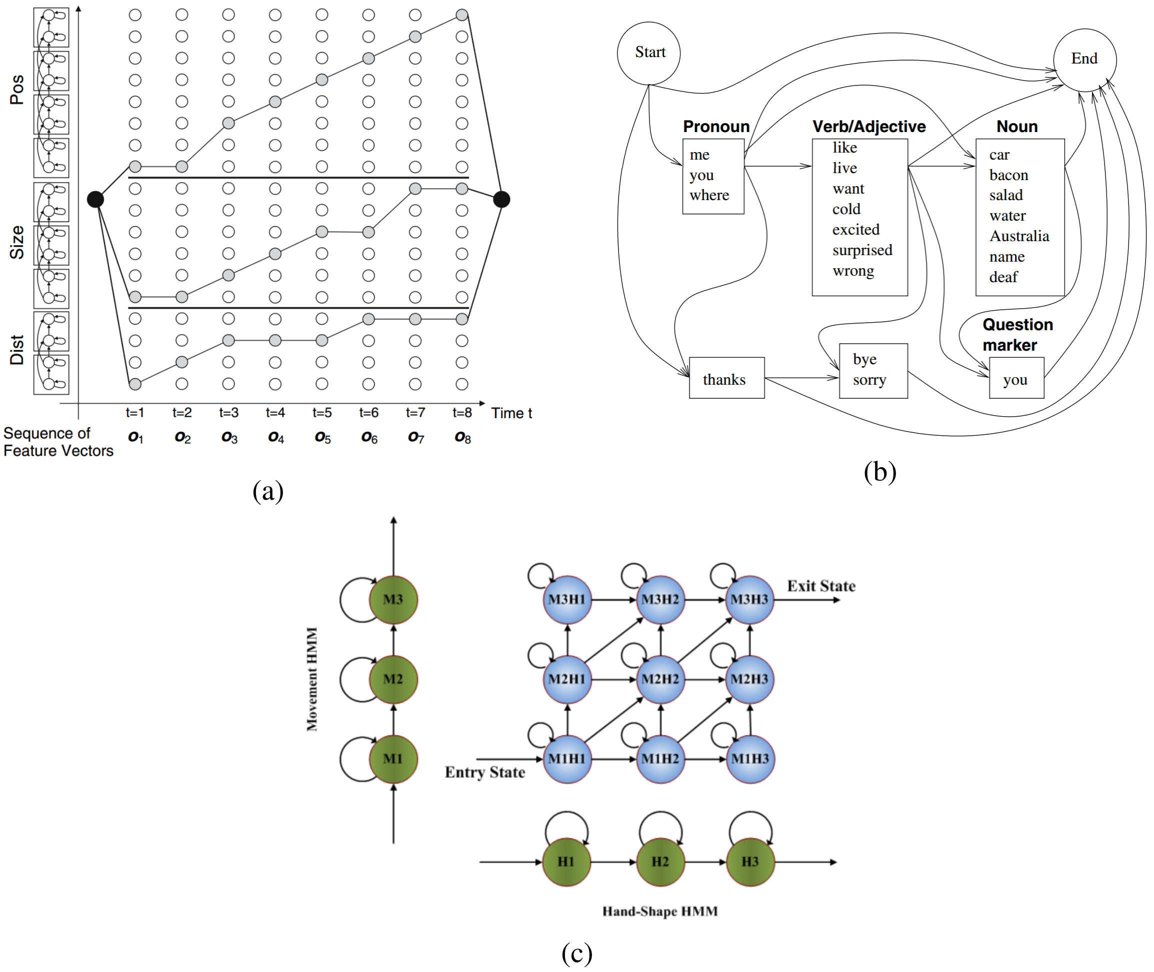
In conventional SLR systems, after the feature extraction process, the hand-crafted feature descriptors are then fed to a classifier for sign recognition. Oliveira *el al.* [147] reported an interesting comparison between different classifiers for SLR. Given the underlying nature of SLR data (i.e. sequential data), most of the proposed SLR systems use Hidden Markov Models (HMMs) or their variants for sign recognition [190, 226, 222, 80, 246]. Some works have also been using artificial neural networks (ANN) [99, 12, 100], Conditional Random Fields (CRFs) [242, 241, 89] or SVMs [130, 162]. In simpler approaches, Dynamic Time Wrapping (DTW) has also been applied. It is the case of Athitsos *et al.* [18] that uses DTW to recognize gestures by just using the hand motion (i.e., ignoring hands shape and position with respect to the body). Given a new test gesture, the DTW metric is computed and compared with the DTW of each sample in the training set and, then, the test gesture is classified according to the smallest value obtained.

One of the first studies in which HMM was used for sign recognition was the one proposed by Starner *et al.* [190]. The authors suggested an HMM with four states in their vision-based SLR system. Von Agris *et al.* [226] proposed a more complex approach, in which each sign is classified by means of subunit models using parallel HMMs. As illustrated in Figure 4.6a, there are subunit models from three different groups features: size, position, and distance.

The complexity of SLR arises with continuous sign recognition, especially as the vocabulary size increases. Phoneme modeling is one of the solutions to this problem. The underlying idea is to identify smaller units of the signs, called phonemes or cheremes (like the phonemes of speech). Vogler and Metaxas [222] proposed a system that follows this idea. The phonemes, representing the movement and handshape of gestures, are modeled in single channels using HMM. The main advantage of phoneme modeling in sign language is that the number of phonemes is much smaller than the overall number of signs. Despite this, the phonemes of sign language are not clearly defined. Typically, phonemes correspond to different hand shapes, motion types, orientation, or body location [35].

Continuous sentences in sign language, like in spoken language, are composed of several signs arranged according to the grammar structure of the corresponding sign language. Holden *et al.* [80] proposed an HMM model combined with grammar information for continuous sign recognition. These grammar constraints are used to prevent the model of identifying signs in wrong positions of the sentence (see Figure 4.6b).

Another difficulty with continuous sign recognition is related to the movement of epenthesis that appears as an extra movement between two sequential signs in a sentence. Some works have been proposed to handle this problem by modeling these movements of epenthesis explicitly as non-vocabulary signs appearing between two consecutive vocabulary signs. Following this idea, Yu *et al.* [246] proposed a continuous recognition system with two main steps. In the first stage, the continuous sign language is segmented into isolated sign segments. Then, each sign segment is classified as a vocabulary sign or non-vocabulary sign,



(a)

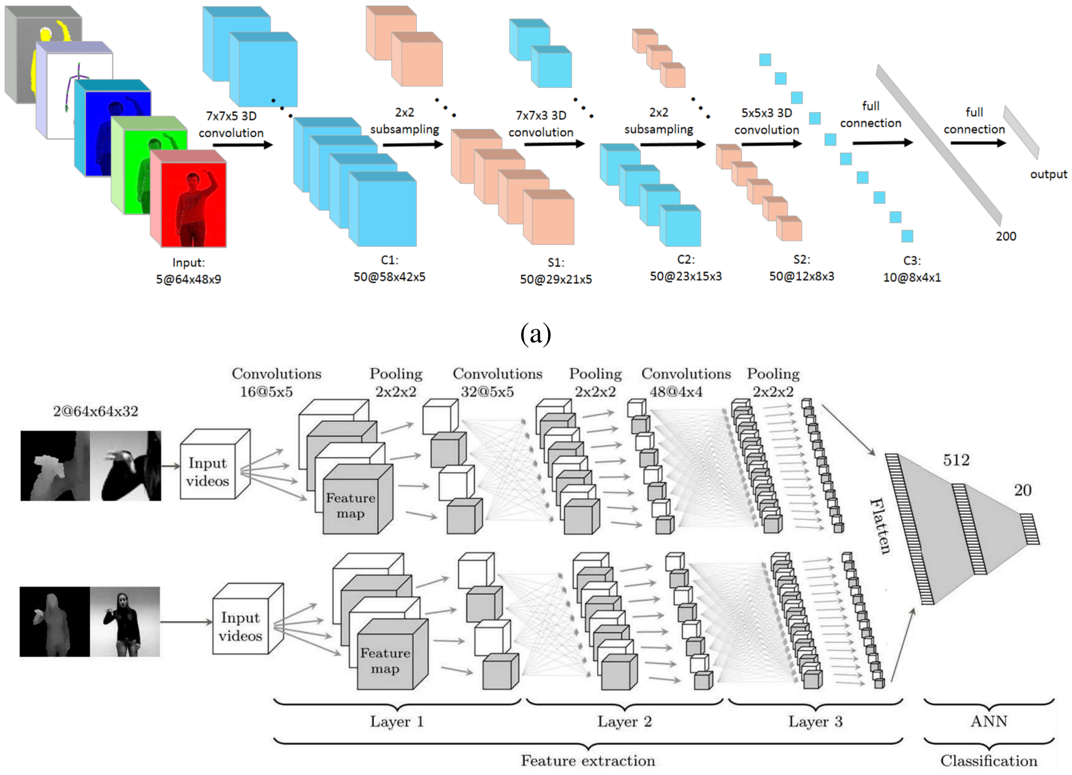
(b)

(c)

Figure 4.6: HMMs for SLR: (a) parallel HMMs proposed in [226], (b) HMM model with grammar constraints [80], and (3) product HMM model to deal with epenthesis [246].

using a product HMM model (see Figure 4.6c). Similar approaches are used in [242, 241]. Both works proposed CRF models to distinguish signs from non-sign patterns.

Although the aforementioned approaches have promoted a significant evolution in the SLR research field, the recent success of deep neural networks approaches, particularly those using CNNs, in tasks like object detection and recognition, has been extended to the SLR problem [93, 150, 85, 141, 157, 102, 101, 235, 123, 107, 138]. The underlying motivation is to avoid the extraction of hand-crafted features and the inherent difficulty of designing reliable features to the large variations of hand gestures. Unlike hand-crafted feature extraction approaches, deep neural networks can automatically learn multiple levels of representations from the data, with higher levels representing more abstract concepts. Deep learning techniques have been applied either for static, isolated, or continuous SLR. Notable examples of such works can be found in [93, 150, 85]. Kang *et al*. [93] proposed one the first



(a)

(b)

Figure 4.7: Examples of CNN architectures for sign language recognition: (a) 3D CNN for integrating colour, depth and trajectory information in the recognition - proposed in [85], and (b) a model of two CNNs, one for extracting hand features and another for extracting upper body features, that is followed by a classical ANN with one hidden layer for the final classification - proposed in [157].

CNN-based SLR recognition methods, which simply involved a straightforward application of a standard CNN architecture for fingerspelling recognition. Oyedotun and Khashman [150] also applied CNNs for recognizing static hand gestures of the ASL. However, they also explored the potential of stacked denoising autoencoders to learn high-level discriminative features in an unsupervised manner.

Regarding dynamic SLR, the spatial-temporal modeling capability of 3D CNN's has been widely explored [85, 141, 157]. In [85], a multi-channel video stream, including color and depth data, and the body joint positions, is used as input to a 3D CNN in order to integrate color, depth and trajectory information in the recognition task. The architecture of the CNN is illustrated in Figure 4.7a. Pigou *el al.* [157] proposed a slightly different approach. The architecture of the model consists of two 3D CNNs, one for extracting hand features and another for extracting upper body features. Then, a classical ANN with one hidden layer is used for classification, after concatenating the outcomes of both CNNs (see Figure 4.7b). However, in most aforementioned CNN-based approaches, the temporal domain is not elegantly taken into consideration. That is, 2D CNNs are usually trained on the frame-level, and even 3D CNNs approaches must use a sliding window scheme for obtaining the final sequence prediction. In this regard, some recent works [102, 101, 235] proposed an end-to-end embedding of deep CNNs into an HMM framework. This hybrid CNN-HMM framework combines the strong discriminative capabilities of deep CNNs along with the sequence modeling abilities of HMMs for simultaneous gesture segmentation and recognition. CNNs are known to possess much more powerful image modeling capabilities than generative models such as GMMs, which are traditionally used to model the observation probabilities within such a GMM-HMM framework. Other works [123, 107, 138] resorted to recurrent neural networks (RNNs), especially to long short-term memory networks (LSTMs), due to their ability to deal with sequential data.

Although the works mentioned above have tackled some interesting problems of the SLR research field, most of the researches focused on the signer-dependent scenario, which means that the test signers have been seen during the training process of the models. In real applications, the performance of such systems will decrease dramatically when the signer is new to the system. Some works [223, 245] attempted to mitigate this problem using signer adaption approaches, in which a previously trained model is adapted to a new test signer by using a small amount of signer specific data. However, collecting enough training data from each new test signer and, then, retrain the SLR model is not realistic. Other works [226, 94] tried to address the signer-independent problem implicitly, by simply normalizing the extracted features accordingly to the signers' heights and distance to the camera. All of these attempts do not cover the inter-signer variations that exist in the actual manual signing process (e.g., the variations on the handshapes between different signers). The development of truly signer-independent SLR models still remains an open problem in the SLR research field.

4.5 Databases

This section presents an overview of the most relevant benchmark datasets in SLR. The available databases can be grouped into two main groups, regarding the recognition task to which they can be used for: isolated sign recognition and continuous sign recognition. Table 4.1 summarizes the important features of these databases.

Table 4.1: Main features of the most relevant benchmark for SLR.

Database		Sign language	Video Resolution	Number of signers	Isolated recognition	Continuous recognition	Tracking
Jochen-Triesch		ASL	128x128 pixels	24	✓	x	✓
SIGNUM		DGS*	776x578 pixels at 30 fps	25	✓	✓	x
Boston Corpora	*RWTH-BOSTON-50*	ASL	312x242 pixels at 30 fps	3	✓	x	✓
	RWTH-BOSTON-104	ASL	312x242 pixels at,30 fps	3	x	✓	✓
	RWTH-BOSTON-400	ASL	312x242 pixels at,30 fps	4	x	✓	x
RWTH-PHOENIX-Weather		DGS*	210x260 pixels at 25 fps	7	x	✓	✓
Corpus-NGT		NGT†	1920x1080 pixels	92	✓	✓	✓
Purdue RVL-SLLL		ASL	640x480 pixels	14	✓	✓	✓
Arabic Sign Language	*Isolated recognition*	Arabic	-	3	✓	x	-
	Continuous recognition		720x528 pixels at 25 fps	1	x	✓	x
CopyCat		ASL	-	5	✓	✓	-

*DGS - orig. *Deutsche Gebärdensprache* (German Sign Language)

†NGT - orig. *Nederlandse Gebarentaal* (Netherlands Sign Language)

4.5.1 Jochen-Triesch Database

Jochen-Triesch [212] is a static hand posture database, which consists of 10 hand posture signs performed by a total of 24 subjects against three types of backgrounds: uniform light, uniform dark and complex. There exist three images for each subject and sign, one for each background type. The images of the Jochen-Triesch database are in grey-scale with a resolution of 128×128 pixels. Besides the class labels, the bounding box annotations of the hands are also available.

4.5.2 SIGNUM Corpus

The SIGNUM Corpus [224] is a German sign language (DGS - orig. *Deutsche Gebärdensprache*) database that contains videos of isolated signs as well as of continuous utterances performed by several signers. Therefore, it is suitable for signer independent continuous sign language recognition tasks. In the SIGNUM database, there are 450 basic signs, representing different word types (i.e., nouns, verbs, adjectives, and numbers), and 780 continuous sentences. The entire corpus was performed once by 25 native signers under some constraint conditions. For instance, the database was recorded in a controlled environment using diffuse lighting and uniform blue background. In addition, all signers wear dark clothes with long sleeves. Although the annotation of the recorded sign/sentence is available in this database, the signer's hands and head positions are absent. Therefore, this database is not suitable for the development and evaluation of tracking algorithms.

4.5.3 Boston Corpora

The National Center for sign language and Gesture Resources of the Boston University published a database of ASL sentences. Then, several subsets of the database were defined at the RWTH Aachen University in order to create benchmark databases for the evaluation of both isolated and continuous sign language. These databases are the RWTH-BOSTON-50, the RWTH-BOSTON-104, and the RWTH-BOSTON-400 [51, 49].

On the one hand, the RWTH-BOSTON-50 database was developed for the task of isolated sign language recognition. This database contains 50 sign language words that were performed by three signers (one male and two female signers). The data was recorded without any kind of clothing constraint since all the signers are dressed differently. Currently, the database contains about 1450 freely available annotated frames [51].

On the other hand, RWTH-BOSTON-104 is a sign language database for continuous sign language recognition. The database comprises 201 continuous sentences constructed from 104 signs. For the evaluation of hand tracking methods, the signer's hands position have been manually annotated in 15 videos (1119 frames) of the RWTH-BOSTON-104 database [51].

The RWTH-BOSTON-400 is the largest subset of the Boston University corpus. The database contains a total of 843 continuous sentences created from about 400 signs. For benchmark purposes, all data are divided into 633 training sentences, 106 development sentences, and 104 evaluation sentences. The database was collected from four different signers, two male, and two female signers, which are not equally represented in the data [51, 49].

4.5.4 RWTH-PHOENIX-Weather corpus

The RWTH-PHOENIX-Weather corpus [8] is a sign language database of weather forecasts recorded from German public TV. The weather forecasts videos have been transcribed, using gloss notation, by deaf native speakers of native DGS. The RWTH-PHOENIX-Weather corpus consists of 1 980 sentences in DGS, with an overall vocabulary of 911 different signs. The signing was performed by seven different signers using a stationary color camera for recording. Although the videos of the database have not been recorded under laboratory conditions, the signers wear dark clothes in front of a grey background and are placed in front of the camera.

Besides the annotation of the signs, the center point of the hands' palms and the nose tip have been annotated in a subset of 266 signs of the corpus. Moreover, 38 facial landmarks have been annotated for all seven interpreters in a total of 369 images [8, 67].

4.5.5 Corpus-NGT Database

The Corpus-NGT database is a large open-access corpus of the Dutch Sign Language (NGT - orig. *Nederlandse Gebarentaal*). The database currently contains 72 hours of video recordings collected from 92 different signers that were selected, taking into account the age and dialect variations that occur in the Dutch Deaf community. The signers performed several tasks, such as introducing themselves briefly, telling about an event, and debating specific topics. The signing data were recorded using four cameras. While two of the cameras are used to record the upper body of the signers (giving detailed recordings of the face and head), the other ones are used to record each signer from above. Currently, some of the video data are annotated, to enabling the evaluation of hand and head tracking algorithms [3, 52].

4.5.6 Purdue RVL-SLLL American Sign Language database

The Purdue RVL-SLLL ASL database was first presented in [133], and was created to be used on the development of automatic recognition systems for the ASL. The database provides a wide range of signed material, including the American fingerspelling alphabet, numbers, several isolated signs, and examples of short discourse narratives for testing continuous sign language recognition systems. The database was collected from fourteen signers under controlled lighting conditions. In addition, most of the data, except the short discourse narratives, were also recorded in less-controlled lighting conditions in order to provide more complex and real-life recognition situations. The entire database, together with a customized interface, is available under request [7].

4.5.7 Arabic Sign Language database

Assaleh *et al.* [17] presented two Arabic Sign Language databases, one for isolated gesture recognition and the other one for continuous sentence recognition. No restrictions on clothing or background were imposed in both databases. The database created for isolated gesture recognition contains 23 gestures selected from the greeting section of the Arabic sign language. All gestures were performed fifty times by three signers over different sessions. Altogether 150 repetitions of each gesture are available in the database. The second database contains a total of 40 continuous sentences created from a set of 19-word vocabulary. The sentences and words were selected so that those sentences could comprise the most common situations that Deaf people might find in their daily life. In this database, each sentence, as well as the individual gestures boundaries that make up that sentence, are labeled.

4.5.8 CopyCAT corpus

Zafrulla *et al.* [248] uses a sign language database, the CopyCAT corpus, created based on an educational game for children. The signing data were collected from five deaf students with ages varying from 6 to 9 years. The database comprises a total of 420 ASL sentences based on a set of 19-word vocabulary. All sentences were manually labeled in order to provide sign boundaries for training.

4.5.9 Databases with depth information

In the sign language literature, there are few publicly available databases with depth information. Two notable exceptions can be found in [11, 131]. The MSRGesture3D [11] database was acquired by a Kinect device and, currently, contains 12 dynamic ASL gestures, performed by 10 different people. Each person performed each gesture 2-3 times. There are 336 files in total, each corresponding to a depth sequence. This database also includes the manual segmentation of the hands. Marin *et al.* [131, 132] proposed a Microsoft Kinect and Leap Motion (MKLM) hand gesture recognition database. The database comprises 10 static gestures from the ASL, performed by 14 different people, and repeated 10 times. For each sign, data from both Leap Motion and Kinect were acquired together. The Kinect data include the color images along with the corresponding depth maps.

Although these databases include depth information, none of them contain sentences in their vocabulary and, hence, they are not suitable for continuous sign recognition.

4.6 Summary

Attempting to bridge the communication barriers between deaf and hearing people, SLR has increasingly become one of the most appealing research topics in modern societies. Several SLR methodologies have been proposed in the last few years, with increasing progress in the sign recognition performance.

The progress of SLR systems is due to many aspects, but there are two main factors: (i) the evolution of sign acquisition systems, allowing the integration of different data modalities for a more accurate sign segmentation and recognition, and (ii) the recent resurgence of deep learning techniques, which, when applied in the SLR context, avoid the laborious feature engineering work for designing reliable hand-crafted features to the large sign variability.

The ultimate goal of the SLR research field is to come out with a continuous SLR system that works in unconstrained environments. In fact, most of the SLR researchers are facing the main challenges related to the continuous SLR, such as the large vocabulary size, the grammatical processes in the manual signing, and the movement of epenthesis. Nonetheless, isolated/static SLR is far from being a solved problem and still remains a multidisciplinary challenging task, especially under uncontrolled settings. There are still several fundamental problems in the SLR field, in general, that should be tackled before start thinking in an unconstrained continuous SLR system. Some of the major shortcomings in the SLR area are related to:

- **Benchmark datasets and evaluation protocol**: Although several SLR databases have been proposed for benchmark purposes, each research group typically uses their own dataset along with their own specific evaluation protocol, which hampers a proper and reliable assessment of the SLR methodologies. In addition, there still exist some major flaws on the available SLR databases: (i) most of the available datasets were recorded with several constraints regarding the acquisition conditions, signer's clothing, and signing process; (ii) there are few sign language databases that gather RGB data with depth information; (iii) lack of databases depicting the non-manual component of the sign language; and (iv) there are no databases with videos depicting the interaction between deaf and hearing people.

- **Integration of non-manual signs in the recognition**: In practice, most of the SLR researchers stress the importance of the non-manual component in sign language (i.e., facial expressions and head movements); however, facial expressiveness is an often neglected aspect of current SLR methods. In addition, facial expression recognition is, by itself, a very challenging problem that requires further fundamental research.

- **Multimodal SLR**: The introduction of low-cost 3D sensors, such as Kinect and Leap Motion, has promoted the development of multimodal SLR systems that integrate RGB data and depth information. Although several multimodal SLR have been proposed in the literature, the combination of different modalities is often performed in a straightforward manner (e.g., by combining either the features extracted from each modality or the decisions of the modality-specific models). To take full advantage of the available data modalities, it would be desired to develop SLR models that are able to automatically learn the complementary and specifics aspects of different modalities during the learning stage.

- **Signer-independent SLR**: Although recent SLR methods have demonstrated remarkable performances, especially in signer-dependent scenarios, their recognition rates typically decrease significantly when the signer is new to the system. This performance drop is the result of the large inter-signer variations in the manual signing process. Most of the SLR works addressed the signer-independent problem implicitly, by merely building normalized feature descriptors robust to the physical variations of the signers (e.g., hand size and length of the arm) and different acquisition conditions (e.g., distance to the camera). It is, therefore, of crucial importance the development of SLR frameworks that explicitly learn signer-invariant feature representations that preserve the relevant part of the information about the signs while discarding the signer-specific traits that may hamper the sign classification task.

- **Regularizing deep learning-based SLR methods**: The recent success of deep learning techniques in many pattern recognition problems has also been extended to the SLR problem. Several deep learning-based SLR frameworks have been proposed, outperforming previous existing conventional SLR methods. However, training these complex deep models remains a non-trivial task, since the amount of training data required increases drastically with the complexity of the prediction model. The most common solution is to reduce the complexity of the model. This is generally done by applying standard regularization techniques (e.g., l^2, dropout, data augmentation, or batch-normalization), or by training the model in an unsupervised fashion such that more unlabelled data can be used, or by embedding domain/prior knowledge in the learning process. In the SLR context, embedding prior knowledge, either about the sign language process or the available input data modalities, in the deep model's architecture and training process remains quite unexplored.

Chapter 5

A Portuguese Sign Language and Expressiveness Recognition Database

The content presented in this Chapter was partially published in [209], and submitted to [206]:

- **Ferreira, P. M.**, Rodrigues, I. V., Rio, A., Sousa, R., Pereira, E. M., and Rebelo, A. (2014). Corsil: A novel dataset for portuguese sign language and expressiveness recognition. In *RecPad 2014: Conference on Pattern Recognition*, pages 1–2
- **Ferreira, P. M.**, Pernes, D., Rebelo, A., and Cardoso, J. S. (2019b). Desire: Deep signer-invariant representations for sign language recognition. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–16

One of the main challenges in the development of any automatic recognition system, especially in the sign language field, is the availability of suitable ground-truth data. In this chapter, a novel video-based database, called CorSiL, is presented. It comprises two major components: (i) an LGP dataset, and (ii) a duo-interaction dataset, between Deaf and/or hearing people. The database can be used for different purposes like Sign Language recognition tasks or emotion/expressiveness recognition from body language.

5.1 Introduction

The development and validation of automatic SLR recognition systems rely on the availability of benchmark databases along with reliable ground-truth data. Although several SLR databases have been proposed in the literature (see chapter 4), many issues remain unexplored:

- Until the recently introduced online LGP Dictionary[1], there were no LGP databases available;
- Most of the available SLR datasets are recorded under very constrained conditions;
- There are few multimodal SLR databases that gather RGB color data with depth information;
- There are no databases with videos depicting the interaction between deaf and hearing people.

To address the problems mentioned above, a novel video-based sign language and body expressiveness database, called CorSiL, is proposed. It can be used for the evaluation and validation of (i) SLR systems, and (ii) expressiveness/behavior recognition systems. In this regard, the CorSiL database is composed of two distinct datasets, one suitable for SLR tasks, called signLangDB, and another for expressiveness recognition from body behavior, called corpLangDB. To the best of our knowledge, it is the first database that gathers sign language videos along with videos depicting the duo-interaction between deaf and/or hearing people. This composition makes the CorSiL database so unique and valuable, since the possibility of understanding the emotions and expressiveness behind the signs may open new research paths in SLR. Both datasets have been already manually annotated. The entire CorSiL database is already freely available to the research community for benchmark purposes[2].

5.2 Database description

The CorSiL database has two major components (or subsets) each one with a specific purpose:

1. **signLangDB:** a Portuguese Sign Language video dataset.
2. **corpLangDB:** a duo-interaction video dataset between Deaf and/or hearing people.

Besides video content, both datasets are available along with technical annotations. It is important to stress that the contact with the signers and volunteers of the recordings was obtained with a partnership with the *Escola EB2/3 Eugénio de Andrade* and *Escola Artística de Soares dos Reis*, Porto, Portugal.

[1] https://www.infopedia.pt/dicionarios/lingua-gestual

[2] https://github.com/pmmf/CorSiL

5.2.1 signLangDB subset

The signLangDB dataset is an LGP database suitable for both isolated and continuous SLR tasks. The dataset contains 182 isolated signs, representing the alphabet and the numbers as well as nouns, pronouns, verbs or common expressions; some performed with one hand and others with both. These signs include not only the informative part of the sign but also the entire movement from the rest position to the return to it. It also contains 40 continuous sentences that were selected in an attempt to comprise the most common situations that Deaf people might find in their daily life. All sentences are grammatically well-constructed in which there are no constraints regarding a specific sentence structure. In addition, no intentional pauses are placed between signs within a sentence. The entire list of signs and sentences that constitute the proposed database is presented in Appendix A.

All gestures and sentences were performed once by 15 native signers, including 5 males and 10 females, in a free and natural expression environment, without any clothing restriction but with a slightly-controlled uniform background. Moreover, some of the signers performed their gestures from a standing position while others performed seated in a chair. The recording conditions were set with this minimal amount of constraints so that they could meet a real environment scenario (see Figure 5.1).

The signing data were acquired using the Microsoft Kinect camera, making this dataset one of the few with depth information associated with the RGB color data. All videos were recorded using an image resolution of 640×480 at 30 fps. This spatial information should ensure a reliable extraction of hand and facial features from the images. Each video clip was stored as a sequence of *.png* images in order to speed up access to individual frames. Figure 5.2 illustrates a pair of color and depth images.

The annotations of the signLangDB database include the segmentation of each sign and sentence for classification purposes as well as the bounding boxes of the hand's position.

5.2.2 corpLangDB subset

The corpLangDB dataset contains videos depicting the interaction that occurs between a pair of individuals during a dialogue. The purpose of such a dataset is to enable the possibility of performing studies that analyze dialogue relationships (from sociological, psychological and technical perspectives) between two individuals, from distinct populations: Deaf and hearing people, in a relaxed environment. The conversation scenarios and topics recorded in this dataset were defined by socio-psychologists from the *Faculdade de Psicologia e Ciências da Educação da Universidade do Porto*. In this regard, the following three conversation scenarios were defined:

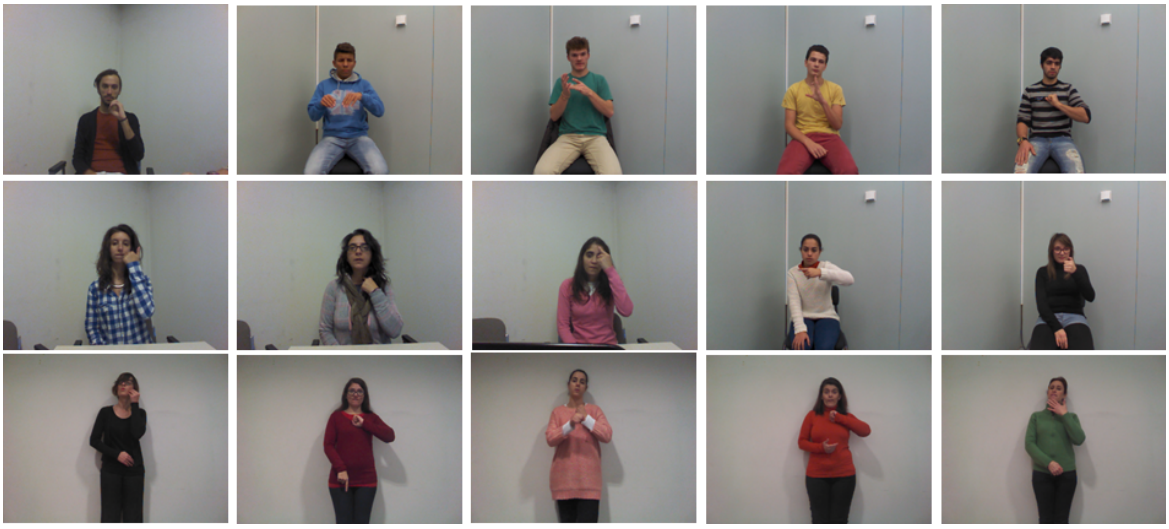


Figure 5.1: An overview of all 15 signers available in the signLangDB subset.

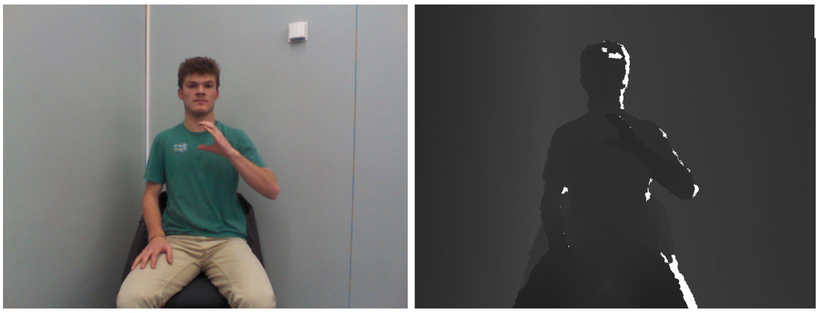
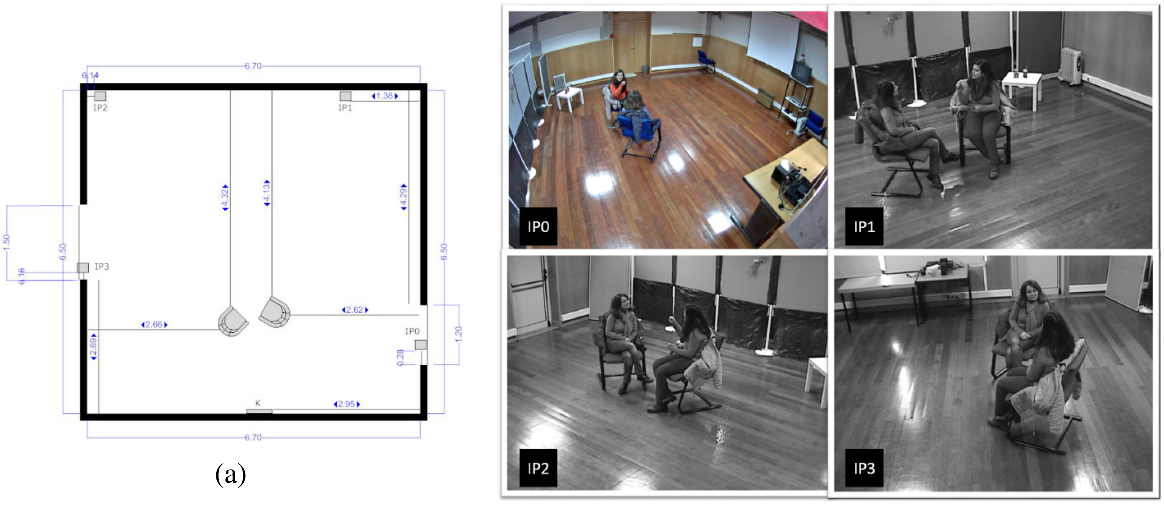


Figure 5.2: Color and depth pair of images from the signLangDB dataset.

1. Conversation between two deaf people;
2. Conversation between two hearing people;
3. Conversation between a deaf and a hearing person.

In order to execute these scenarios, two requirements were defined so that the interaction between each pair of individuals could occur in the most natural way possible. These requirements are: 1) the individuals should know each other and have some affinity, and 2) the acquisition should take place in a venue that was familiar to all subjects.

As the focus of the corpLangDB database is to enable the analysis of behavior and expressiveness, the set of conversation topics was defined in a staggered way, so that the discussion would generate emotions of increasing intensity in the actors of the conversation. To build a framework for the videos' acquisition, four different conversation topics were defined as belonging to two-fold moments: positive (1 and 2) and negative (3 and 4):



(a)

(b)

Figure 5.3: corpLangDB subset acquisition setup: (a) recording scenario and (b) field of view of cameras P0, P1, P2, and P3.

1. Talk about happy moments;
2. Talk about people with which the actor has a strong love or friendship bond;
3. Talk about sad moments;
4. Talk about situations that awaken anger/indignation/injustice.

The volunteer subjects, 13 in total, were coupled so that the conversation scenarios were covered. Each conversation between a pair of subjects was designated as a session. Accordingly, the database currently comprises a total of 9 sessions.

Figure 5.3 represents the entire scenario used to record the videos and also the field of view of each camera. IP0, IP1, IP2, and IP3 represent the cameras used and, K a Microsoft Kinect. These were all placed in strategic locations (at the height of 2.58 meters) for the best capture possible. Two chairs were centered in the room in a way that was propitious for the dialogue in terms of proximity and comfort and for the video acquisition. The annotations available in this dataset include the positions of the head, hands, trunk, elbows, eyes, mouth, and nose. These annotations were performed using the VIPER-GT tool. [3]

[3] http://viper-toolkit.sourceforge.net/

5.3 Summary

This chapter describes a novel SLR database, the so-called CorSiL, which was created during this thesis. CorSiL is currently composed of two distinct subsets one suitable for SLR tasks and another for expressiveness recognition from body behavior.

At this moment, the sign language component of the dataset contains 182 isolated signs and 40 continuous sentences, recorded from 15 signers in an environment without restrictions. Given the potential of depth information in SLR, this dataset has been recorded using a low-cost RGB-D camera (i.e., Microsoft Kinect), making this dataset one of the few with depth information associated to the RGB color data.

The expressiveness component of the dataset contains videos depicting the interaction that occurs between a pair of individuals, from distinct populations: Deaf and hearing people, during different conversation topics (e.g., happiness, love, hate, anger). Therefore, this dataset can be used to analyze the body expressiveness of Deaf people in the sign language.

To the best of our knowledge, the CorSiL database is the first multimodal database that gathers sign language videos along with videos depicting the duo-interaction between deaf and/or hearing people. With such a unique composition, we expect to open new research paths in the SLR research field. Until now, the proposed database was already used in one Ph.D. Thesis [156] and two Master's Thesis [134, 168].

Chapter 6

Multimodal Sign Language Recognition

The content presented in this Chapter was published in [201, 200**?**]:

- **Ferreira, P. M.**, Cardoso, J. S., and Rebelo, A. (2019a). On the role of multimodal learning in the recognition of sign language. *Multimedia Tools and Applications*, 78(8):10035–10056

- **Ferreira, P. M.**, Cardoso, J. S., and Rebelo, A. (2017a). Multimodal learning for sign language recognition. In Alexandre, L. A., Salvador Sánchez, J., and Rodrigues, J. M. F., editors, *Pattern Recognition and Image Analysis*, pages 313–321, Cham. Springer International Publishing

- **Ferreira, P. M.**, Cardoso, J. S., and Rebelo, A. (2017b). The potential of multimodal learning for sign language recognition. In *RecPad 2017: Conference on Pattern Recognition*, pages 1–2. **(best paper award)**

The introduction of low-cost consumer 3D sensors, such as the Microsoft Kinect and, more recently, the Leap Motion, has launched new research lines for SLR. Several multimodal SLR methodologies, which integrate either RGB, depth or Leap Motion data, have been proposed. This fact, along with the resurgence of deep learning techniques, promoted significant progress in the recognition performance of SLR systems. However, their recognition performance can be further improved. In practice, a multimodal deep neural network requires a lot of training data to generalize well. This is not the case of the SLR context where large multimodal datasets, with both Kinect and Leap Motion data, are scarce.

This chapter aims to exploit multimodal learning techniques for a robust SLR, making use of data provided by Kinect and Leap Motion. In particular, single-modality approaches, as well as different multimodal methods, mainly based on convolutional neural networks, are proposed. Our main contribution is a novel multimodal end-to-end neural network along with a regularization scheme that explicitly learns both modality-specific and complementary feature representations.

6.1 Introduction

Researchers have been addressed the SLR problem by means of wearable devices (e.g., data gloves or similar equipment's) or vision-based systems. Although data gloves yield more reliable and descriptive features, vision-based SLR systems are arguably the most natural choice for real-world applications. Vision-based SLR is less invasive since there is no need to wear cumbersome devices that may affect the natural signing movement.

The first vision-based SLR approaches were focused on the extraction of color information from 2D images or videos [38, 12]. In general, a set of relevant color-based features is extracted to be used in a traditional classification module that provides the sign recognition. When sign inputs are captured in 2D using a single RGB camera, automatic segmentation and recognition become difficult due to several environmental factors, such as self occlusions, background noise or illumination variability. In addition, as these representations contain a 2D description of the three-dimensional hand pose, 2D color-based approaches often demonstrate several limitations, especially when the signs to be recognized involve complex 3D poses and/or movements.

Thanks to the emergence of low-cost consumer depth cameras (e.g., Microsoft Kinect), there has been a great interest in the development of SLR systems based on RGB-D data (i.e., the combination of an RGB image and its corresponding depth map). Several works have explored the 3D information for an accurate gesture recognition [216, 108, 48, 241]. This new layer of information is particularly helpful since it allows describing the 3D hand pose of the signs. However, it is worth to mention that Kinect-like sensors are not able to localize all the small details associated with the pose of the fingers.

The recent introduction of the Leap Motion [169] controller has launched new research lines for SLR. Leap Motion can acquire 3D data with a millimeter level precision and has been specifically designed for hand gesture and finger recognition. The device is available along with a built-in interface and SDK that provides direct access to various features such as fingertip positions, palm center, and palm orientation. One of the first studies referring to the utilization of Leap Motion for SLR has been presented in [159]. The authors stated that, although Leap Motion may have a great potential for sign recognition, it is not always able to recognize all fingers in some hand configurations (e.g., when the hand rotates and is perpendicular to the controller). Moreover, the Leap Motion device has a small field of view as compared to Kinect-like sensors. Therefore, its usage is limited to acquire hand and finger movements.

Our observation, on top of the considerations mentioned above, is that a single-sensor SLR system may not be enough for robust sign recognition. It is not possible to accurately capture all articulations or movements of the signs using a single sensor. Since both Leap Motion and Kinect sensors have quite complementary characteristics, it seems promising to exploit both of them to develop a robust multimodal framework for SLR. While Leap Motion provides few accurate and relevant key-points, Kinect produces both a color image and a complete depth map with a large number of less accurate 3D points.

In this chapter, we propose a novel multimodal end-to-end neural network, called End-to-End Network with Regularization (EENReg), that explicitly models the complementary characteristics of the input modalities. Our novel architecture, along with a well-designed loss function, results in a model that jointly learns to extract representations that are specific to each modality as well as shared representations across modalities. The underlying idea is to increase the discriminative ability of the learned features by regularizing the entire learning process and, hence, improve the generalization capability of multimodal deep models. In particular, our main contributions are:

- A comparative study between single-modality and multimodal learning techniques, in order to demonstrate the effectiveness of multimodal learning in the overall sign recognition performance;

- The introduction of a robust hand gesture detection algorithm, which promotes an overall improvement in the sign recognition performance;

- The implementation of a complete randomized data augmentation scheme, which allows training deeper neural networks without overfitting;

- The proposal of a novel multimodal end-to-end neural network architecture, the so-called EENReg, along with a well-designed loss function that explicitly learns to extract deep features representations that are unique and shared between modalities. By inducing the model to jointly learn both modality-specific and modality-shared features, the proposed EENReg outperforms the state-of-the-art multimodal approaches.

The chapter is organized in eight sections including the Introduction (Section 6.1). The existing multimodal SLR methodologies are summarized in Section 6.2. Section 6.3 presents a pre-processing step for segmenting the hands from the noisy background, before sign recognition. The implemented single-modality and conventional multimodal SLR methodologies are fully described in Sections 6.4 and 6.5, respectively. Section 6.6 fully describes the proposed EENReg model along with the proposed regularization schemes. Section 6.7 reports the experimental evaluation of the proposed methodologies. Finally, the major achievements presented in this chapter are summarized in Section 6.8.

6.2 Related Work

SLR methodologies have gradually shifted from being single-modality, mainly based on 2D color images, to multimodal approaches thanks to the emergence to 3D depth sensors such as Kinect and Leap Motion. In the following subsections, we present some of the existing multimodal SLR methodologies and, then, the most interesting deep multimodal regularization techniques that have been employed in more generic pattern recognition problems.

6.2.1 Multimodal Sign Language Recognition

In order to capture complementary aspects of the input signs, several researchers have been proposed hybrid frameworks/models by combining different input modalities of more than one device/sensor. The first attempts to develop multimodal SLR systems involved a huge feature engineering effort in order to extract a set of hand-crafted features from each input modality (e.g., RGB, depth, or Leap Motion data). In [48], multiple depth-based descriptors are fed into an SVM classifier for gesture recognition. In the first stage, the hands are detected and segmented using both color and depth information. Afterwards, different subsets of depth-based features, such as distance, elevation, curvature, and palm area features, are extracted. In order to overcome the limitations of depth maps provided by Kinect-like sensors (i.e., the lack of fingertip detail), Marin *et al.* [131, 132] introduced one of the first attempts of combing the input data from Leap Motion with Kinect for a more robust SLR. Particularly, they proposed a feature-level fusion approach with hand-crafted features extracted from two data modalities (i.e., depth data from Kinect and Leap Motion data). The extracted features are based on the distances between the hand contour points and the hand's centroid, the curvature of the hand contour, and the convex hull of the hand shape. These feature descriptors are then fed into an SVM classifier for the recognition of the performed signs. More recently, Kumar *et al.* [107] also proposed a multimodal SLR framework using data acquired from Kinect and Leap Motion. A set of features is extracted from the raw data captured with both sensors. Then, sign recognition is performed by a combination of two sequential classifiers, i.e., an HMM and a Bidirectional Long Short-Term Memory Neural Network (BLSTM-NN).

Since the recent resurgence of deep neural networks, there is a trend in the SLR research community to learn features directly from the data, in contrast to engineering them [85, 157, 235]. Pigou *et al.* [157] proposed a multimodal 3D CNN which integrates RGB and depth data. The combination of these two modalities is performed at the input level, by simply concatenating the RGB channels with the depth map. Huang *et al.* [85] proposed a similar network architecture. However, the body skeleton is integrated as an extra input modality. Recently, Wu *et al.* [235] employed a slightly more complex multimodal framework, which also considers skeleton joint information, depth, and RGB images as the multimodal input observations. Different from the previous approaches, the architecture of the model comprises modality-specific neural networks. In addition, different conventional multimodal fusion strategies, such as feature-level and decision-level fusion schemes, were explored. However, direct and unconstrained training of these complex multimodal deep neural networks remains non-trivial, since the amount of training data required increases significantly with the complexity of the model. It is, therefore, common practice either reduce the complexity of the model or employ regularization in the training process.

In contrast to existing solutions, in this thesis, we propose a novel multimodal end-to-end neural network that explicitly models private feature representations that are specific to each modality and shared feature representations that are similar between modalities. By imposing such regularization in the learning process, the underlying idea is to increase the discriminative ability of the learned features and, hence, improve the generalization capability of the model.

6.2.2 Deep Multimodal Regularization

In the deep multimodal learning context, an important design consideration is the formulation of well-designed loss functions along with regularization terms that enforce inter-modality and intra-modality relationships. Although the relationship between different modalities has not been thoroughly investigated in the SLR task, several deep multimodal regularization techniques have been proposed in the scope of more generic problems, such as RGB-D object recognition [184, 231, 111, 230, 229], transfer learning [30], and deep feature embeddings [84, 177].

In order to learn relationships between modalities, Sohn *et al.* [184] proposed a loss function that minimizes the variation of information between modalities. The underlying idea is that learning to maximize the amount of information that one data modality has about the others would allow multimodal generative models to reason about the missing data modality given partial observations. Wu *et al.* [237] explored both inter-modality and intra-class relationships, for video semantic classification, by imposing trace-norm based regularizations on the shared and output layers of the neural network. Loss functions that enforce inter- and intra-modality correlations have also been proposed in [230, 229]. In particular, Wang *et al.* [229] proposed a multimodal fusion layer that uses matrix transformations to enforce a common part to be shared by features of different modalities while retaining modality-specific properties. Lenz *et al.* [111] introduced a structured regularization term in the loss function, in order to regularize the number of modalities used per feature (node). In this regard, the model is able to learn correlated features between multiple input modalities, while discarding weak correlations between them. The formulation of well-designed loss functions, along with additional regularization terms, have also been explored in many other domains, such as transfer learning [30, 192], deep feature embeddings [84, 177], and image retrieval [249] as well as to maximize domain-specific performance metrics [249, 71, 116]. A very comprehensive and recent survey on deep multimodal learning and regularization can be found in [165].

Our work is inspired by the recent works on transfer learning [30] and local similarity-aware deep feature embeddings [84], which explore the complementary properties between the source and target domains. However, we extend their ideas for supervised deep multimodal learning, in particular, for the SLR task. We also pay special attention to incorporating the complementarity and specifics of multimodality in the training procedure, which implied an entire refinement of the neural network architecture, loss function, and regularization terms.

6.3 Pre-processing

Both Kinect modalities, color and depth, require a pre-processing step in order to segment the hands, from the noisy background of the image, before feature extraction and sign recognition. As illustrated in Figure 6.1, the developed hand segmentation method exploits both color and depth information of Kinect.

In a first step, a skin color segmentation, in the YCbCr color space, is performed to roughly distinguish skin pixels from background pixels. The YCbCr color space was adopted since it is perceptually uniform and separates luminance and chrominance, which makes this color space suitable for skin color detection [77]. The YCbCr color space comprises three channels, representing the luminance component (Y) and the chrominance components (Cb and Cr). The conversion from RGB to YCbCr is defined as follows:

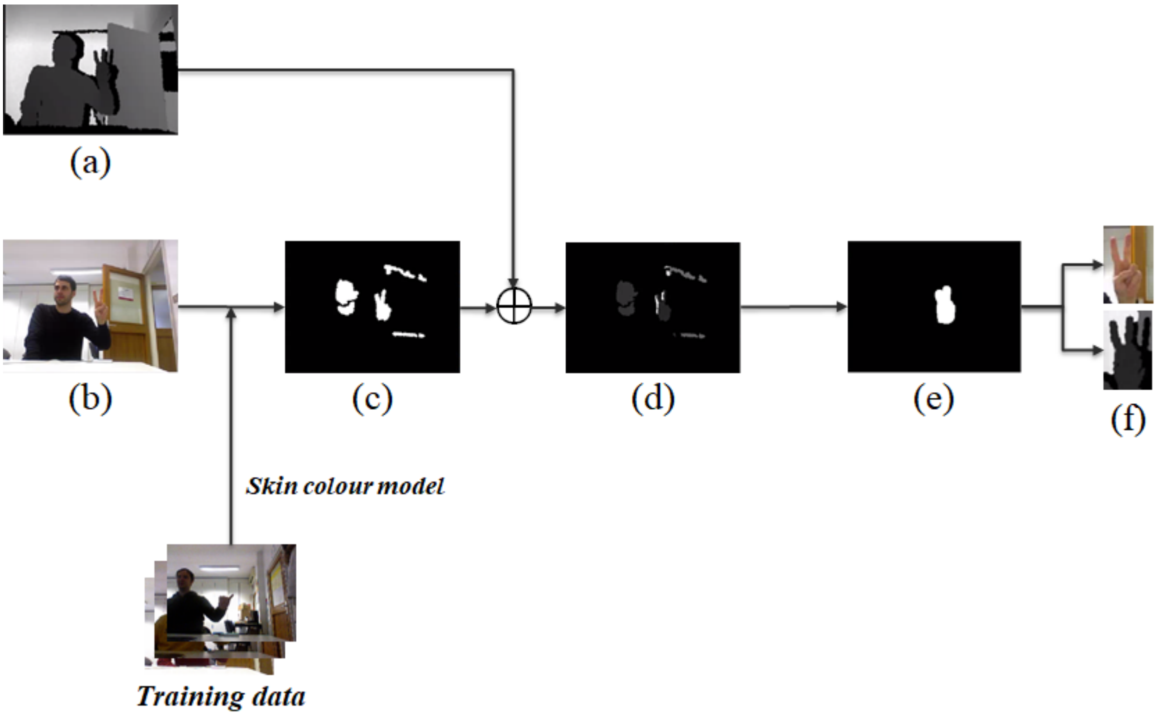


Figure 6.1: Hand detection methodology: input depth image (a), input colour image (b), skin colour segmentation (c), filtered depth map (d), hand segmentation result (e) and the cropped colour and depth images (f).

$$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \tag{6.1}$$

$$C_b = (B - Y) \cdot 0.564 + 128 \tag{6.2}$$

$$C_r = (R - Y) \cdot 0.713 + 128 \tag{6.3}$$

For illumination-invariance, the implemented skin color segmentation method just makes use of both chrominance components (CbCr). In the CbCr subspace, the distribution of skin and background colors is modeled each one by a multivariate Gaussian mixture model $\mathcal{S}$ and $\mathcal{B}$, respectively. Therefore, the probability that a pixel j with the color value of X_j belongs to the skin color model $\mathcal{S}$ is defined as:

$$\begin{aligned} p(X_j \mid \mathcal{S}) &= \sum_{i=1}^{k} \gamma_i p(X_j \mid \mathcal{S}_i) \\ &= \sum_{i=1}^{k} \frac{\gamma_i}{(2\pi)^{l/2} |\Sigma_{\mathcal{S}_i}|^{1/2}} \exp\left\{ -\frac{1}{2} (X_j - \mu_{\mathcal{S}_i})^T \Sigma_{\mathcal{S}_i}^{-1} (X_j - \mu_{\mathcal{S}_i}) \right\}, \end{aligned} \tag{6.4}$$

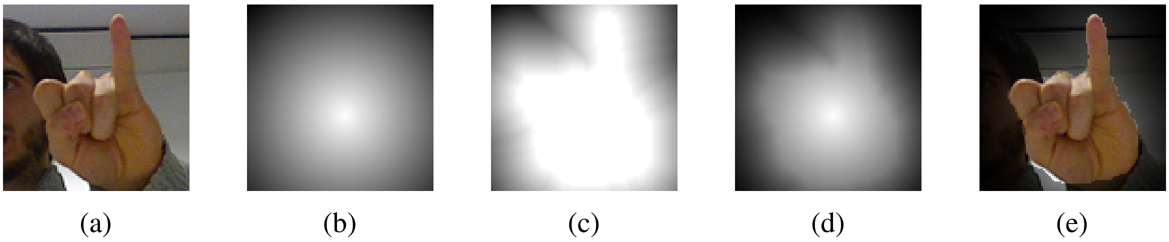


Figure 6.2: Illustration of the background suppression methodology for a given colour image: original cropped colour image (a), Euclidean distance map of each pixel to the segmentation mask centroid (b), distance transform of the segmentation mask (c), linear combination of the two distance maps (d) and its application on the cropped colour image (e).

where l denotes the feature space dimension, k represents the number of Gaussian components of $\mathcal{S}$, each one characterized by its mean vector $\mu_{\mathcal{S}_i}$, covariance matrix $\Sigma_{\mathcal{S}_i}$ and proportions γ_i. Likewise, the probability of a pixel belonging to the background color model $\mathcal{B}$ is modeled in a similar manner.

After obtaining the skin model $\mathcal{S}$ and the background model $\mathcal{B}$, the skin color segmentation is performed by maximum likelihood classification of pixels within a test image. That is, a pixel with color value X is classified as skin pixel if the following condition is verified:

$$p(X \mid \mathcal{S}) > p(X \mid \mathcal{B}) \tag{6.5}$$

As illustrated in Figure 6.1c, the skin color segmentation process results in a binary mask of the skin colored objects present in the image (i.e., hand, face or other uncovered body parts). This binary mask is then used to filter the depth map, in order to only retain depth samples associated with skin-colored objects (see Figure 6.1d). The underlying assumption is that the closest skin-colored object of the image corresponds to the hand, as the signer is typically the nearest object to the camera.

After this stage, hand segmentation is performed on the filtered depth map using a region growing technique. First, a search for the region with the minimum depth value D_{min} on the filtered depth map is performed. The corresponding region R_{min} is chosen as the seed region for the hand detection process if its area is greater than a threshold T_{area}; otherwise, the next closest region is selected. The area criterion is used so that the selected R_{min} does not correspond to an isolated artifact due to measurement noise. In the next step, the neighboring pixels are examined and added to the seed region R_{min} based on a homogeneity criterion (i.e., if the depth value difference between those pixels and R_{min} does not exceed a threshold T_{depth}). This process is applied iteratively until no more pixels satisfy the homogeneity criterion. As illustrated in Figure 6.1e, the segmented hand is then represented by all pixels that have been merged during this iterative procedure.

Once the segmentation process is completed, the original color and depth images are both cropped by the bounding box of the segmented sign and, then, these resulting cropped images are resized to the average sign size of the training set (see Figure 6.1f).

To further reduce the influence of the background in the recognition task, a background suppression methodology is applied to the cropped images (see Figure 6.2). First, a Euclidean distance map of each pixel to the segmentation mask centroid as well as the distance transform of the segmentation mask are computed (Figures 6.2b and 6.2c, respectively). These maps are linear combined and, then, multiplied with the cropped image. As illustrated in Figure 6.2e, the final result is the fading out of the background pixels according to their distance to the segmentation centroid, while it keeps the foreground pixels unchanged.

Finally, the image inputs are normalized to ensure that each pixel (i.e., input parameter) has a similar data distribution and, hence, make converge faster while training the models. Data normalization is done by subtracting the mean from each pixel and then dividing the result by the standard deviation.

6.4 Single-modality Sign Recognition

In this section, the implemented single-modality methodologies for SLR are presented. For both Kinect modalities (color and depth), we resorted to a deep learning strategy based on CNNs; whereas for Leap Motion, we implemented a traditional machine learning pipeline with hand-crafted feature extraction. This choice was motivated by the different nature of the data of these modalities. As the leap motion data is already at a high semantic level (i.e., well-structured features), a shallow classifier is suitable for making predictions.

6.4.1 Kinect modalities (colour and depth)

CNN architecture

The implemented neural network follows the traditional CNN architecture for classification [187]. It starts from several sequences of convolution-convolution-pooling layers to fully connected layers. More specifically, the implemented CNN is composed of six convolutional layers, three fully connected layers (or dense layers) and two max-pooling layers. The number of filters is doubled after each pooling operation. Finally, the last layer of the CNN is a softmax output layer, which contains the output probabilities for each class label. The

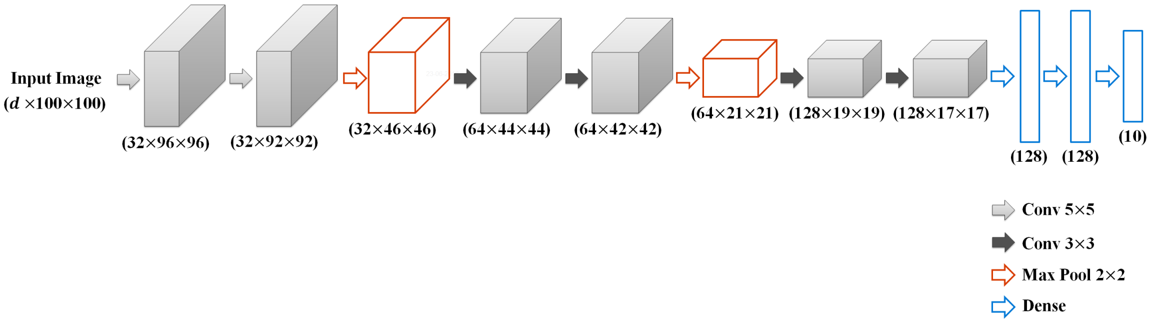


Figure 6.3: The architecture of the implemented CNN model for single-modality sign recognition, using colour $(d = 3)$ or depth $(d = 1)$.

output node that produces the largest probability is chosen as the overall classification. The architecture of the implemented CNN is illustrated in Figure 6.3.

For training the model, the goal is to minimize the categorical cross-entropy, a commonly used loss function for classification tasks, which is given by:

$$\mathcal{L} = -\sum_{i=0}^{N} \mathbf{y}_i^\top \log \hat{\mathbf{y}}_i, \tag{6.6}$$

where $\mathbf{y}_i$ is a column vector denoting the one-hot encoding of the class label for input i and $\hat{\mathbf{y}}_i$ are the softmax predictions of the model. The Nesterov's Accelerated Gradient Descent with momentum was used for optimization.

Regularization

During the training stage, several regularization techniques were applied to prevent overfitting (i.e., dropout, ℓ^2-norm, and data augmentation). Fundamental details about these commonly used regularizations techniques can be found in section 3.3.

In practice, dropout was applied to the fully connected layers of the implemented CNN. Data augmentation is the process of increasing, artificially, the number of training samples, by means of different image transformations and noise addition. In here, a randomized data augmentation scheme based on both geometric and color transformations is applied during the training step. The underlying idea is to increase the robustness of the CNN model to the wide range of hand gestures positions, poses, viewing angles as well as to different illumination conditions and contrasts. The data augmentation process is applied in an online-fashion, within every iteration, to a random half of the images of each mini-batch.

Specifically, the considered geometric transformations are obtained through the following randomized affine image warping:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & k_1 \\ k_2 & 1 \end{bmatrix} \begin{bmatrix} x - t_1 \\ y - t_2 \end{bmatrix}, \tag{6.7}$$

where θ is the rotation angle, k_1 and k_2 are the skew parameters along the x and y directions. t_1 and t_2 denote both translation parameters and s is the scale factor. It is import to note that the values of these parameters are randomly selected from predefined sets (those sets are listed in Section 6.7). Pixels mapped outside the original image are assigned with the pixel values of their mirrored position.

The other type of image augmentation focuses on randomly normalizing the contrast of each channel in the training images. Formally, let S_c be the c-th channel of the input image, the new intensity value at each pixel in channel c is given by:

$$S_c' = \begin{cases} 0 & , \text{if } S_c < S_c(p_L) \\ \dfrac{S_c - S_c(p_L)}{S_c(p_H) - S_c(p_L)} & , \text{if } S_c(p_L) \leq S_c \leq S_c(p_H), \\ 1 & , \text{if } S_c > S_c(p_H) \end{cases} \tag{6.8}$$

where p_L and p_H represent the lower and higher histogram percentiles that are randomly selected for the color transformation, respectively. This scheme simulates the scenario that the input images are acquired with different intensities, contrasts, and illumination conditions.

Figure 6.4 illustrates the application of the implemented data augmentation procedure. Although the resulting augmented images may be highly correlated between them, this randomized augmentation scheme significantly increases the size of the training set, which allows the utilization of deep CNN architectures without overfitting.

6.4.2 Leap Motion

Unlike Kinect, Leap Motion does not provide a complete depth map. Instead, it directly provides a set of relevant features of hand and fingertips. The raw data of Leap Motion include the number of detected fingers, the position of the fingertips, the palm center, the hand orientation, and the hand radius [132]. From these data, 3 different types of features were computed:

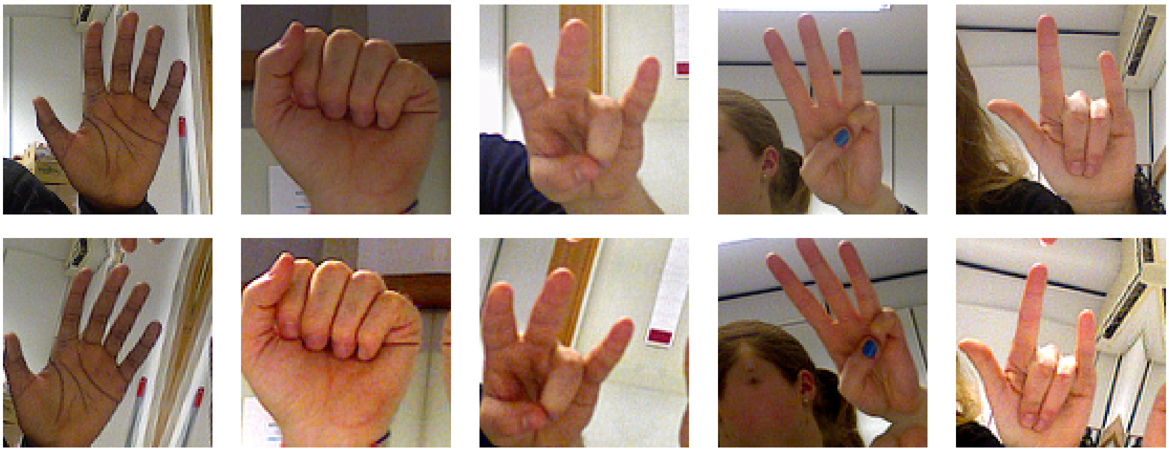


Figure 6.4: Illustration of the implemented randomized data augmentation process: original colour images (top row) along with the corresponding augmented images (bottom row).

1. **Fingertip distances** $D_i = \|F_i - C\|, i = 1, ..., N$; where N denotes the number of detected fingers and D_i represents the 3D distances between each fingertip F_i and the hand centre C;

2. **Fingertip inter-distances** $I_i = \|F_i - F_{i+1}\|, i = 1, ..., N-1$; represent the 3D distances between consecutive fingertips;

3. **Hand direction** O: represents the direction from the palm position toward the fingers. The direction is expressed as a unit vector pointing in the same direction as the directed line from the palm position to the fingers;

where $\|\cdot\|$ denotes the ℓ^2-norm, corresponding to the geometric distance between the fingertips. Both distance features are normalized by the signer (user), according to the maximum fingertip distance and fingertip inter-distance of each user. This normalization is performed to make those features robust to people with different hand's size. Then, these 3 sets of features are used as input into a multi-class SVM classifier for sign recognition. The block diagram of the implemented Leap Motion-based sign recognition approach is illustrated in Figure 6.5.

6.5 Conventional Multimodal Sign Recognition

The data provided by Kinect and Leap Motion have quite complementary characteristics. In this Section, we exploit them together for SLR purposes.

According to the level of fusion, multimodal fusion techniques can be roughly grouped into two main categories: (i) decision-level and (ii) feature-level fusion techniques [145]. As

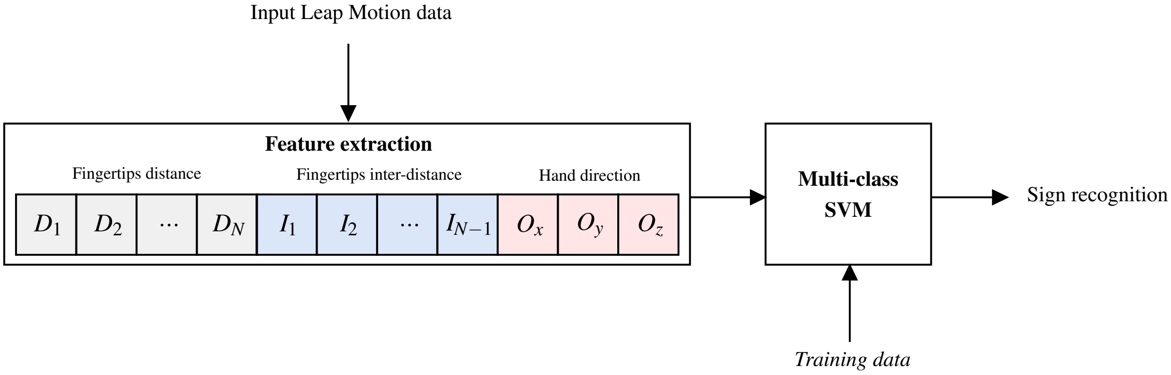


Figure 6.5: Single-modality sign recognition methodology of Leap Motion data.

described in the following, we propose multimodal approaches of each fusion category for the SLR task, making use of 3 modalities (i.e., color, depth, and Leap Motion data).

Throughout the rest of the chapter, let $\mathbb{X} = \{(x_i^c, x_i^d, x_i^l, y_i)\}_{i=1}^N$ denote the labelled multimodal dataset of N samples, where x_i^c, x_i^d and x_i^l represent the i-th colour, depth and leap motion sample, respectively, and y_i denotes the ground-truth class labels.

6.5.1 Decision-level fusion

The purpose of decision-level fusion is to learn a specific classifier for each modality and, then, to find a decision rule between them. In this paper, we apply this concept making use of the output class probabilities of the models designed individually for each modality under analysis. Then, two main kinds of decision rules, to combine these class probabilities, were implemented: 1) pre-defined decisions rules, and 2) decision rules learned from the data. Let $\hat{\mathbf{y}}^c$, $\hat{\mathbf{y}}^d$ and $\hat{\mathbf{y}}^l$ be the predictions of colour, depth and leap motion modalities, respectively; then, the decision-level fusion schemes is illustrated in Figure 6.6.

Pre-defined decision rules

Herein, two different pre-defined decision rules were implemented. In the first approach, the final prediction is given by the argument that maximizes the averaged class probabilities. In the second approach, the final prediction is given by the model with maximum confidence. The confidence of a model in making a prediction is measured by its highest class probability.

Learned decision rule

The underlying idea of this approach is to learn a decision rule from the data. Therefore, a descriptor that concatenates the class probabilities, extracted from the individual models of

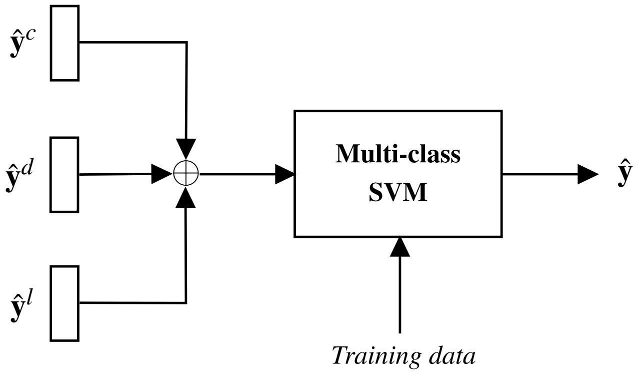


Figure 6.6: Decision-level fusion, in which the decision rule is learned from the data. $\oplus$ is an aggregate operator representing the concatenation of the modality-specific class probabilities.

each modality, is created and, then, used as input into a multiclass SVM classifier for sign recognition.

6.5.2 Feature-level fusion

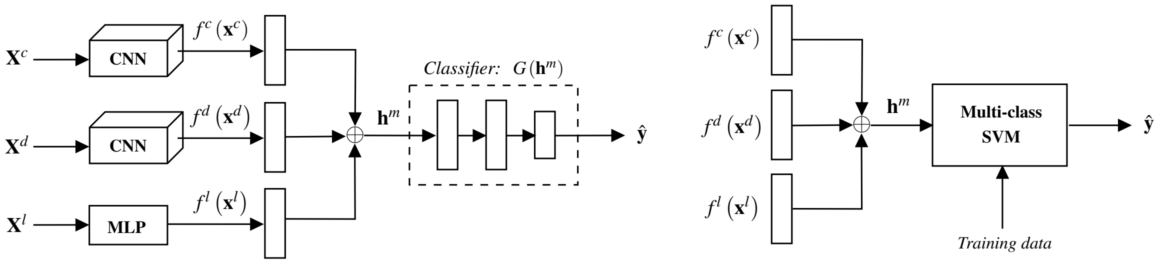
In general, feature-level fusion is characterized by three phases: (i) learning a feature representation/embedding, (ii) supervised training, and (iii) testing [145]. According to the order in which phases (i) and (ii) are made, feature-level fusion techniques can be roughly divided into two main groups: 1) End-to-end fusion, where the representation and the classifier are jointly learned; and 2) Multi-step fusion, where the representation is first learned, and then the classifier is learned from it.

End-to-end fusion

The underlying idea of this approach is to jointly learn a multimodal deep feature representation $\mathbf{h}^m$ and a classifier $G(\mathbf{h}^m)$ that maps from the multimodal representation $\mathbf{h}^m$ to the task-specific predictions $\hat{\mathbf{y}}$. In our scenario, the neural network has three input-specific pipes, one for each data type: (i) colour $\mathbf{x}^c$, (ii) depth $\mathbf{x}^d$ and (iii) leap motion $\mathbf{x}^l$. Therefore, the multimodal feature embedding is simply given by the concatenation of the embeddings of each modality, such that:

$$\mathbf{h}^m = \big(f^c(\mathbf{x}^c) \oplus f^d(\mathbf{x}^d) \oplus f^l(\mathbf{x}^l)\big), \tag{6.9}$$

where $f^c(\mathbf{x}^c)$, $f^d(\mathbf{x}^d)$ and $f^l(\mathbf{x}^l)$ denote the deep feature representations of colour, depth and leap motion modalities, respectively, and $\oplus$ represents the concatenation operation. While the embeddings of colour $f^c(\mathbf{x}^c)$ and depth $f^d(\mathbf{x}^d)$ are both learned by a CNN, the leap



(a) (b)

Figure 6.7: Feature-level fusion schemes: end-to-end feature fusion (a) and multi-step feature fusion (b). $\oplus$ represents a concatenation operator.

motion embedding $f^l(\mathbf{x}^l)$ is learned by a classical MLP with two hidden layers (each one with 128 neurons). All the layers are trained together end-to-end. The architecture of the implemented end-to-end multimodal neural network is represented in Figure 6.7a.

Multi-step fusion

As in the end-to-end approach, a multimodal representation $\mathbf{h}^m$ is created, by concatenating the modality-specific representations $f^c(\mathbf{x}^c)$, $f^d(\mathbf{x}^d)$ and $f^l(\mathbf{x}^l)$. However, in this case, these representations are first learned individually. In particular, the representations $f^c(\mathbf{x}^c)$ and $f^d(\mathbf{x}^d)$ correspond to the activations extracted from the penultimate dense layer of each modality-specific CNN, and $f^l(\mathbf{x}^l)$ corresponds to the features extracted from the leap motion data (see Section 6.4.2). Then, for sign recognition, the multimodal representation vector $\mathbf{h}^m$ is fed into an additional classifier (i.e., a multi-class SVM). The multi-step feature-level fusion scheme is depicted in Figure 6.7b.

6.6 Multimodal End-to-end Fusion with Regularization

Ideally, the end-to-end network, as previously described in Section 6.5.2, should be able to encode the most relevant aspects of the input modalities for the classification task. However, in practice, training an end-to-end multimodal network with multiple input-specific pipes without overfitting is very difficult, mainly due to its huge number of parameters and, especially, if we have to deal with small datasets.

Rather than adopting a conventional multimodal learning structure that involves simple feature- or decision-level fusions, our goal is to further explore the implicit dependence between different modalities. In this regard, we propose a novel multimodal end-to-end architecture, the so-called EENReg, that explicitly models what is unique and shared between modalities. The underlying idea is that the desired multimodal features should comprise the agreement or shared properties between different modalities, while retaining the modality-specific properties that can only be captured by each modality individually. By imposing such regularization in the learning process, the model's ability to extract meaningful features for the classification should improve.

To induce the model to extract both modality-specific and modality-shared features, the EENReg network is composed of three private streams that are specific to each modality and three shared streams between modalities. In addition, the loss function is defined in such a manner that encourages independence between these private and shared representations. The result is a model that produces shared representations that are similar for all modalities and private representations that are modality-specific. The classifier is then trained on these private and shared representations to enhance the discriminative capability of the model.

6.6.1 Architecture

As depicted in Figure 6.8, the architecture of the EENReg comprises three private streams that are specific to each modality, three shared streams between modalities and a classifier.

While the purpose of each private stream is to transform the data of each modality into a new modality-specific feature representation, the purpose of each shared stream is to perform a mapping from each input modality to a shared representation between modalities. Therefore, the architecture of each stream consists of several sequences of convolution-convolution-pooling layers, for a typical CNN feature extraction, with a dense layer on top of that. In particular, each multimodal stream has the same architecture of the implemented CNN model for single-modality sign recognition (see Figure 6.3 for more details). By concatenating the shared and modality-specific feature representations, a multimodal feature representation is, then, created.

Finally, a classifier that simply comprises three fully connected layers is fed with the multimodal feature representation. The last layer is a softmax output layer, which contains the output probabilities for each class label.

6.6.2 Learning

Let $f_s^m(\mathbf{x})$ be an embedding function that maps from an input sample $\mathbf{x}$ to a shared feature representation of modality m. Also, let $f_p^m(\mathbf{x})$ be an embedding function that maps from a sample $\mathbf{x}$ to a private feature representation that is specific to its modality. In order to maintain feature comparability, the representations $f_s^m(\mathbf{x})$ and $f_p^m(\mathbf{x})$ are first normalized onto

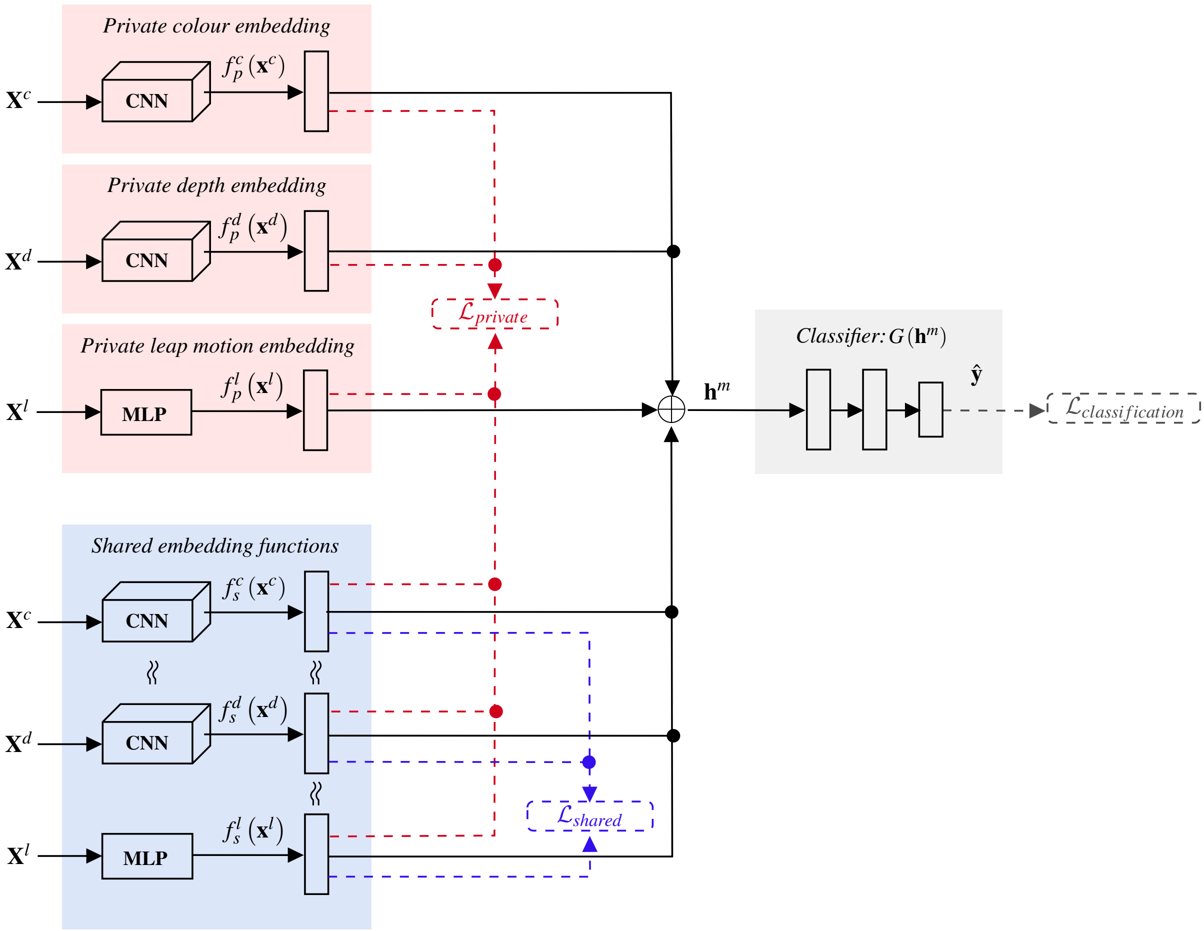


Figure 6.8: The architecture of the EENReg model that explicitly learns to extract deep feature representations that are unique and shared between modalities.

the unit hypersphere, i.e., $\|f(\mathbf{x})\|_2 = 1$. Then, the EENReg model is trained by minimizing the following loss function:

$$\mathcal{L} = \mathcal{L}_{classification} + \alpha\ \mathcal{L}_{private} + \beta\ \mathcal{L}_{shared}, \tag{6.10}$$

where α, β are the weights that control the interaction of the loss terms. The classification loss, $L_{classification}$, trains the model to predict the output labels and corresponds to the categorical cross-entropy as defined in Eq.6.6.

The purpose of the private loss $\mathcal{L}_{private}$ is to encourage the shared and private representations of each modality to encode different aspects of the inputs. Therefore, $\mathcal{L}_{private}$ is defined by imposing orthogonality between the shared and the private representations of each modality, such that:

$$\begin{aligned}\mathcal{L}_{private} = \;& \alpha_c \sum_{i=1}^{N} \langle f_p^c(x_i^c), f_s^c(x_i^c) \rangle \; + \; \alpha_d \sum_{i=1}^{N} \langle f_p^d(x_i^d), f_s^d(x_i^d) \rangle \\ &+ \; \alpha_l \sum_{i=1}^{N} \langle f_p^l(x_i^l), f_s^l(x_i^l) \rangle,\end{aligned} \tag{6.11}$$

where $\langle \cdot, \cdot \rangle$ is the dot product. α_c, α_d and α_l are the weights that control the orthogonality between each modality representations.

The shared loss $\mathcal{L}_{shared}$ encourages the shared representations of all modalities, $f_s^c(\mathbf{x}^c)$, $f_s^d(\mathbf{x}^d)$ and $f_s^l(\mathbf{x}^l)$, to be as similar as possible. Then, the shared loss is simply defined to minimize the pair-wise differences between the shared representations $f_s^c(\mathbf{x}^c)$, $f_s^d(\mathbf{x}^d)$ and $f_s^l(\mathbf{x}^l)$, such that:

$$\begin{aligned}\mathcal{L}_{shared} = \;& \beta_{cd} \sum_{i=1}^{N} \| f_s^c(x_i^c) - f_s^d(x_i^d) \|_2^2 \; + \; \beta_{cl} \sum_{i=1}^{N} \| f_s^c(x_i^c) - f_s^l(x_i^l) \|_2^2 \\ &+ \; \beta_{dl} \sum_{i=1}^{N} \| f_s^d(x_i^d) - f_s^l(x_i^l) \|_2^2,\end{aligned} \tag{6.12}$$

where $\|\cdot\|_2^2$ is the squared l^2-norm. β_{cd}, β_{cl} and β_{dl} are the weights of each pair-wise difference.

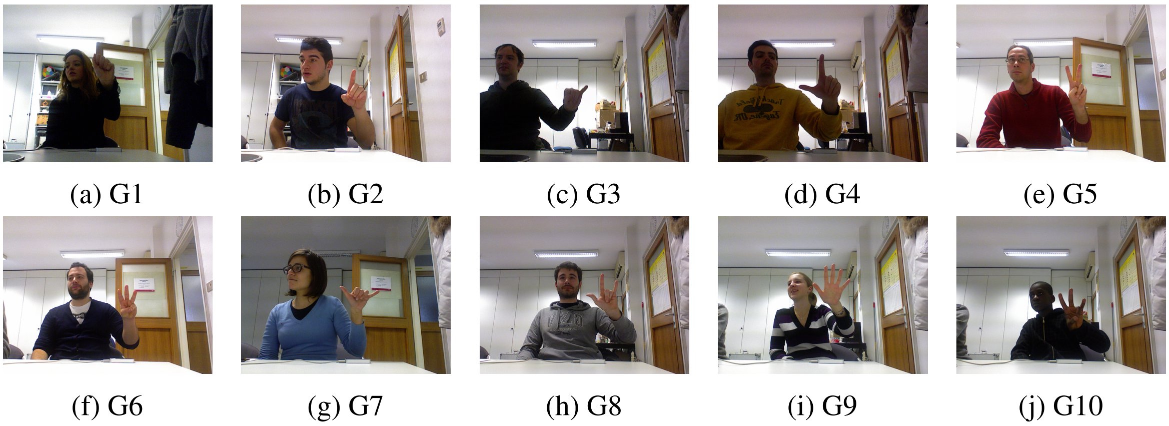
Finally, inference in an EENReg model is given by $\hat{y} = G(\mathbf{h}^m)$, where $\mathbf{h}^m$ represents a multimodal feature embedding given by merging (either by concatenation or sum) all private and shared feature representations, such that:

$$\mathbf{h}^m = \; \big(f_p^c(\mathbf{x}^c) \oplus f_s^c(\mathbf{x}^c) \oplus f_p^d(\mathbf{x}^d) \oplus f_s^d(\mathbf{x}^d) \oplus f_p^l(\mathbf{x}^l) \oplus f_s^l(\mathbf{x}^l) \big) \tag{6.13}$$

6.7 Experimental Evaluation

6.7.1 Dataset and Evaluation Protocol

The experimental evaluation of the proposed methodologies was performed in a public Microsoft Kinect and Leap Motion (MKLM) hand gesture recognition database [131, 132]. This is a balanced dataset of 10 classes, representing 10 static gestures from the American Sign Language (see Figure 6.9). Each sign was performed by 14 different people and repeated 10 times, which results in a total of 1400 gestures. For each sign, data from both Leap Motion



(a) G1 (b) G2 (c) G3 (d) G4 (e) G5

(f) G6 (g) G7 (h) G8 (i) G9 (j) G10

Figure 6.9: Illustrative samples of 10 signs from the MKLM database [131, 132].

and Kinect were acquired together. The Kinect data include the color images along with the corresponding depth maps.

To maximize the usage of the data in the evaluation process, the performance of the models was assessed using a k-fold cross-validation scheme with signer independence, where $k = 5$. Therefore, all performance measures reported throughout this section are the average of their values computed in each split. This evaluation scheme, with $k = 5$, yields at each split a training set of 1100 images from 11 signers and test set of 300 images from the other 3 signers. The training set is further divided, also with signer independence, in 80% for training and 20% for validation.

6.7.2 Implementation Details

The parameters of the hand segmentation algorithm were empirically defined based on the available dataset and remained the same in all the experiments. That is, the number of Gaussian components of the skin and background colour models was set to 2 and 4, respectively. In addition, $T_{area} = 75$ and $T_{depth} = 5$.

All deep models were implemented in Theano [211] and trained with the Nesterov's Accelerated Gradient Descent with momentum using a batch size of 50 samples. We used a learning rate with step decay, in which the initial learning rate was multiplied by 0.99 at each training epoch. The hyperparameters that are common to all the implemented models (i.e., the learning rate and the l_2 coefficient) as well as the specific hyperparameters of the EENReg model (i.e., both $\mathcal{L}_{private}$ and $\mathcal{L}_{shared}$ coefficients) were optimized by means of a grid search approach and cross-validation on the training set. The dropout rate was empirically set as 0.4 for all the experiments. The range of values of the adopted hyperparameters' grid search is

Table 6.1: Hyperparameters sets.

Hyperparameters	Acronym	Set
Leaning rate	-	$\{1e^{-03}, 1e^{-04}\}$
l_2-norm coefficient	-	$\{1e^{-04}, 1e^{-05}\}$
$\mathcal{L}_{private}$ coefficients	α_c, α_d, α_l	$\{1e^{-03}, 5e^{-03}, 1e^{-04}\}$
$\mathcal{L}_{shared}$ coefficients	β_{cd}, β_{cl}, β_{dl}	$\{1e^{-03}, 5e^{-03}, 1e^{-04}\}$

presented in Table 6.1. For a fair comparison, it is important to note that the CNNs streams of all multimodal networks have the same architecture of the CNN model employed for single-modality classification.

Regarding the parameters of the data augmentation scheme, the rotation angle θ was randomly sampled from $\{-\pi/18, -\pi/36, 0, \pi/36, \pi/18\}$. The skew parameters, k_1 and k_2, were both randomly sampled from $\{-0.1, 0, 0.1\}$. The scale parameter s was randomly sampled from five different resize factors $\{0.9, 0.95, 1, 1.05, 1.1\}$. Finally, the translation parameters t_1 and t_2 are randomly sampled integers from the interval $[0, 5]$. Note that these sets of values were selected carefully, so that the meaning of the sign is not changed after the transformation.

The adopted SVM classifier consists of a multi-class SVM classifier based on the one-against-one approach, in which a nonlinear Gaussian Radial Basis Function (RBF) kernel is used. The parameters (C, γ) of the RBF kernel are estimated using a grid search and cross-validation on the training set.

6.7.3 The potential of multimodal learning

In order to assess the potential of multimodal learning in the SLR context, we computed the rate of test signs for which each single-modality method made a correct prediction while the others were wrong. As presented in Table 6.2, these results clearly demonstrate that there is a relative big potential to tackle the SLR problem via multi-modality. In particular, there is a higher complementarity between each Kinect modality (i.e., color or depth) with the Leap Motion rather than between both Kinect modalities. For instance, there are 4.88% and 5.00% of test instances for which Leap Motion made correct predictions while color and depth made incorrect ones, respectively.

6.7.4 Discussion

The experimental results of the proposed single-modality and multimodal sign recognition methodologies are presented in Tables 6.3 and 6.4, respectively. The results are reported in

Table 6.2: The potential of multimodal learning, expressed by the rate of test instances for which modality B made correct predictions while modality A made incorrect ones.

Modality A	Modality B	Multi-modality potential (%)
Colour	Depth	3.88
Colour	Leap Motion	4.88
Depth	Colour	4.25
Depth	Leap Motion	5.00
Leap Motion	Colour	15.50
Leap Motion	Depth	15.25

Table 6.3: Experimental results of the single-modality approaches with and without data augmentation and background suppression. The results are presented in terms of classification accuracy (%).

Modality	Acc (%)		
	w/o background suppression	w/o augmentation	full
Colour	90.12	82.61	**93.17**
Depth	91.22	88.22	**92.61**
Leap Motion	-	-	**82.83**

terms of classification accuracy (Acc), which is given by the ratio between the number of correctly classified signs t and the total number of test signs n: $Acc\% = \frac{t}{n} \times 100$.

A first observation, regarding single-modality approaches, is that both color and depth outperform Leap Motion, with classification accuracies of 93.17%, 92.61%, and 82.83%, respectively. However, it should be noticed that Leap Motion sign recognition does not require any kind of preprocessing in order to segment the hand from the background for feature extraction.

To validate the impact of the proposed background suppression method and data augmentation scheme, both color and depth CNN models were trained without them. As presented in Table 6.3, both color and depth single-modality models performed consistently worse without background suppression and data augmentation, which clearly demonstrate their importance in the overall sign recognition performance.

The most interesting observation is that multimodal fusion often promotes an overall improvement in the sign recognition accuracy - see Table 6.4. These results clearly demonstrate the complementarity between the three modalities. Typically, the classification accuracy increases as each modality is added to the recognition scheme. In particular, the novel end-to-end feature fusion model (EENReg), provides the best overall classification accuracy $(Acc = 97.66\%)$. The EENReg clearly outperforms the other two implemented feature-level

Table 6.4: Experimental results of the multimodal fusion methodologies. C, D and L denote colour, depth and leap motion modalities, respectively. The results are presented in terms of classification accuracy (%).

(a) Proposed multimodal fusion methods.

Fusion Level	Method	Involved modalities	Acc (%)
Feature	End-to-end	C + D	92.80
		C + D + L	94.20
	Multi-step	C + D	96.78
		C + D + L	97.11
	EENReg	C + D	96.17
		C + D + L	**97.66**
Decision	Average rule	C + D	95.78
		C + D + L	97.33
	Confidence rule	C + D	95.78
		C + D + L	96.44
	Learned rule	C + D	95.83
		C + D + L	97.44

(b) State-of-the-art methodologies.

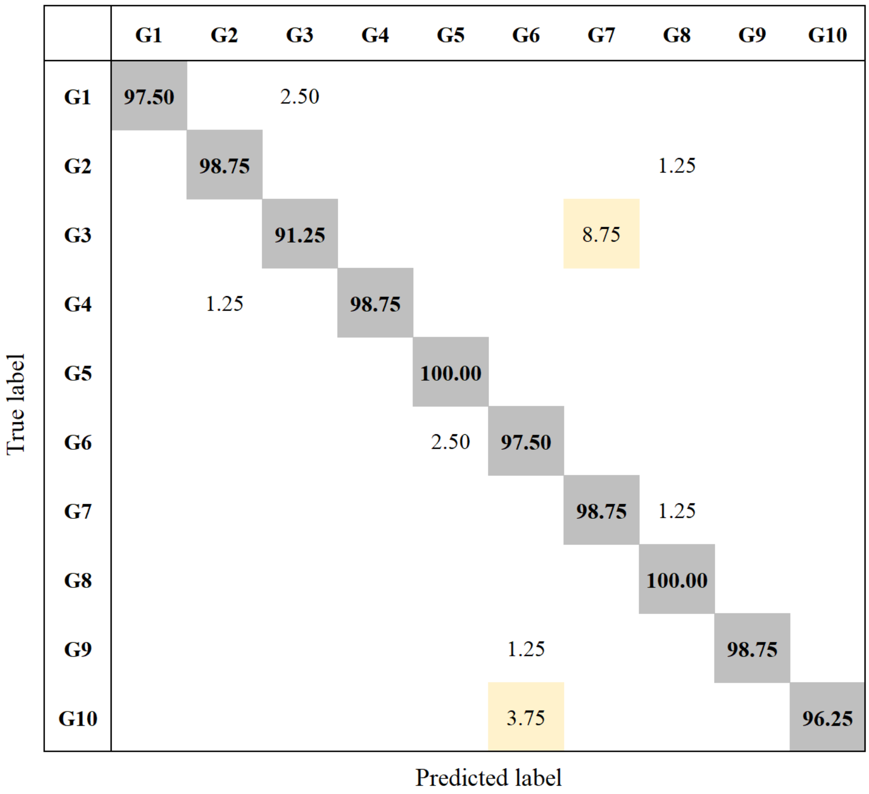
Method	Acc (%)
Marin *et al.* 2014 [131]	91.28
Marin *et al.* 2016 [132]	96.50

Table 6.5: The effect of each loss term in the EENReg model. In the first column, the $\mathcal{L}_{private}$ term was removed from the loss. In the second column, the $\mathcal{L}_{shared}$ term was removed from the loss. The third column is replicated from Table 6.4a as it includes all loss terms. The results are presented in terms of classification accuracy (%).

Method (modalities)	Acc (%)		
	w/o $\mathcal{L}_{private}$	w/o $\mathcal{L}_{shared}$	All loss terms
EENReg (C + D + L)	97.06	96.88	**97.66**

approaches, especially if compared with the traditional end-to-end feature fusion model. These results demonstrate that explicitly modeling what is unique and shared between modalities can improve the model's ability to extract highly discriminative features for the sign classification.

In order to assess the impact of the loss terms in the EENReg model, both $\mathcal{L}_{private}$ and $\mathcal{L}_{shared}$ constraints were removed from the loss, during the training, one at a time. These



True label
Predicted label

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
G1	**97.50**		2.50							
G2		**98.75**						1.25		
G3			**91.25**				8.75			
G4		1.25		**98.75**						
G5					**100.00**					
G6					2.50	**97.50**				
G7							**98.75**	1.25		
G8								**100.00**		
G9						1.25			**98.75**	
G10						3.75				**96.25**

Figure 6.10: Confusion matrix of the best implemented methodology, i.e., the EENReg model. Gray cells represent the true positives, while yellow cells correspond to the false positive rates greater than 2.5%.

results are reported in Table 6.5 and, clearly, suggest that each loss term contributes to a better generalization of the model as its performance was consistently worse without them.

Figure 6.10 shows the confusion matrix obtained for the best methodology, which is the proposed EENReg model. The classification accuracy is larger than 97% for all signs, with the exceptions of signs G3 and G10. While G3 is sometimes misclassified as G7, G10 is a few times misclassified as G6. This happens because these two pairs of signs have a very similar shape between each other. For instance, G10 and G6 just differ from each other in one finger position - see Figure 6.9.

Finally, it is important to stress that the best implemented multimodal fusion approach (i.e., EENReg) outperformed both state-of-art methods [131] and [132], with an Acc of 97.66% against 91.28% and 96.50%, respectively.

6.8 Summary

This chapter addresses the topic of static SLR, by exploring multimodal learning techniques, using of data from 3 distinct modalities: (i) color; (ii) depth, both from Kinect; and (iii) Leap Motion data. In this regard, single-modality approaches, as well as different multimodal methods, to fuse them at different levels, are proposed. Multimodal techniques include feature-level and decision-level fusion techniques.

Experimental results suggest that both Kinect modalities are more discriminative than the Leap Motion data. However, the most interesting observation is that, in general, multimodal learning techniques outperform single-modality methods.

Our main contribution is a novel end-to-end feature-level deep neural network that explicitly models private representations that are specific to each modality and shared feature representations that are similar between them. By imposing such constraints in the learning process, the model is able to jointly learn both modality-specific and modality-shared features and outperform the state-of-the-art multimodal approaches.

Chapter 7

Signer-Independent Sign Language Recognition: Part I

The content presented in this Chapter was published in [207, 210]:

- **Ferreira, P. M.**, Pernes, D., Rebelo, A., and Cardoso, J. S. (2019c). Learning signer-invariant representations with adversarial training. In *The 12th International Conference on Machine Vision (ICMV 2019)*
- **Ferreira, P. M.**, Sequeira, A. F., Pernes, D., Rebelo, A., and Cardoso, J. S. (2019e). Adversarial learning for a robust iris presentation attack detection method against unseen attack presentations. In *2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*
- **Ferreira, P. M.**, Pernes, D., Rebelo, A., and Cardoso, J. S. (2019d). Signer-independent sign language recognition with adversarial neural networks. *International Journal of Machine Learning and Computing (IJMLC)*. (accepted)

Although important steps have been made towards the development of real-world SLR systems, signer-independent SLR is still one of the bottleneck problems of this research field. In this chapter, we propose a deep neural network along with an adversarial training objective, specifically designed to address the signer-independent problem. Concretely speaking, the proposed model consists of an *encoder*, mapping from input images to latent representations, and two classifiers operating on these underlying representations: (i) the *sign-classifier*, for predicting the class/sign labels, and (ii) the *signer-classifier*, for predicting their signer identities. During the learning stage, the *encoder* is simultaneously trained to help the *sign-classifier* as much as possible while trying to fool the *signer-classifier*. This adversarial training procedure allows learning signer-invariant latent representations that are, in fact, highly discriminative for sign recognition.

7.1 Introduction

A practical SLR system must operate in a signer-independent scenario. That is, the signer of the probe must not be seen during the training process of the models. Although current SLR systems demonstrate excellent performances for signer-dependent settings, their recognition rates typically decrease significantly when the signer is new to the system. This performance drop is the result of the large inter-signer variability in the manual signing process of sign languages. Although the appearance of the manual signs is well-defined in sign language dictionaries, in practice, variations may arise due to regional and social factors, and also from age, gender, education and family background. This can lead to significant variations in manual signs performed by different signers, and pose challenging problems for developing robust signer-independent SLR systems. Figure 7.1 illustrates the gesture “eight" performed by six different signers and clearly reveals this inter-signer variability. It is possible to observe not only phonological variations (i.e., different handshapes, palm orientations, and gesture locations) but also a large physical variability (i.e., different hand sizes, body sizes, and arm lengths).

Borrowing from recent works on adversarial neural networks [76, 66] and domain transfer [70], we introduce a deep neural network along with a novel adversarial training objective to specially tackle the signer-independent SLR problem. The underlying idea is to preserve as much information as possible about the signs, while discarding the signer-specific information that is implicitly present in the manual signing process. For this purpose, the proposed deep model is composed by an *encoder* network, which maps from the input images to latent representations, as well as two discriminative classifiers operating on top of these underlying representations, namely the *sign-classifier* network and the *signer-classifier* network. While the *sign-classifier* is trained to predict the sign labels, the *signer-classifier* is trained to discriminate their signer identities. In addition, the parameters of the *encoder* network are optimized to minimize the loss of the *sign-classifier* while trying to fool the *signer-classifier* network. This adversarial and competitive training scheme encourages the learned representations to be signer-invariant and highly discriminative for the sign classification task. To further constrain the latent representations to be signer-invariant, we introduce an additional training objective that operates on the hidden representations of the *encoder* network in order to enforce the latent distributions of different signers to be as similar as possible.

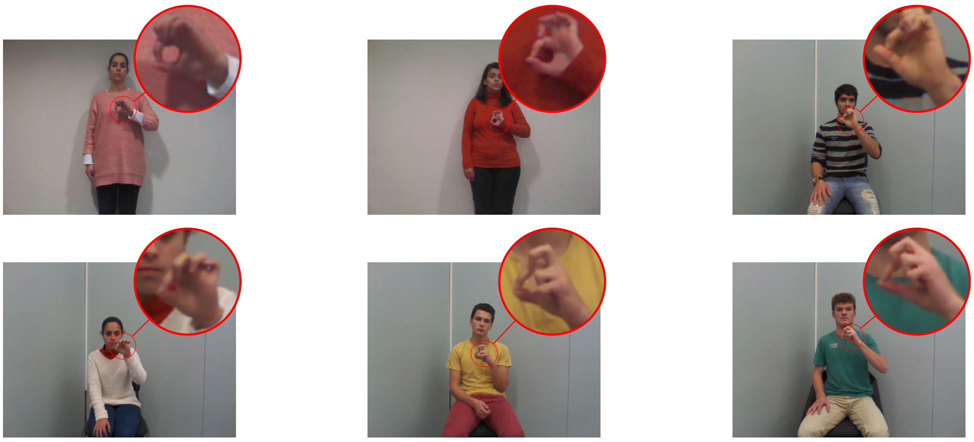


Figure 7.1: Illustration of the inter-signer variability using some samples of the presented CorSiL-signLangDB database. The six signers are performing the sign “eight” of the LGP.

Although this adversarial training framework is similar to those initially introduced by Ganin *et al.* [70], in the context of domain adaptation, and then by Feutry *et al.* [66] to learn anonymized representations, our main contributions on top of these works are three-fold:

- The application for the first time of the adversarial training concept to the signer-independent SLR problem.
- A novel adversarial training objective that differs from the ones of Ganin *et al.* [70] and Feutry *et al.* [66] in two ways. First, our training objective is minimum if and only if the adversarial classifier, which in our case corresponds to the *signer-classifier*, produces a uniform distribution over the signer identities, meaning that our model is completely invariant to the signer identity of the training data. Second, we introduce an additional term to the adversarial training objective that further discourages the learned representations of retaining any signer-specific information, by explicitly imposing similarity in the latent distributions of different signers.
- The extension of the proposed adversarial training objective for other applications (e.g., biometric liveness detection), in which it is desirable to learn feature representations invariant to some specific domain or aspect (see Appendix B).

The remainder of the chapter is organized as follows. Section 7.2 presents the related work. The proposed model along with its adversarial training scheme are fully described in Section 7.3. Experimental results are reported in Section 7.4. Finally, Section 7.5 summarizes the entire chapter.

7.2 Related Work

SLR has become an appealing topic in modern societies because such systems can ideally be used to reduce the communication barriers that exist between deaf and hearing people. Although several works have been proposed towards the development of SLR systems for different sign languages, SLR is still a multidisciplinary challenging task. One of the biggest challenges is related to the large inter-signer variability.

We will start this section by presenting the most important methods directly designed for the signer-independent SLR problem (see subsection 7.2.1). Afterwards, as the signer-independent SLR problem may be addressed as a domain adaptation task, in which the goal is to reduce the distribution difference (i.e., domain shift) between different signers, we will briefly discuss some of the most relevant works on deep domain adaption (see subsection 7.2.2)

7.2.1 Signer-independent SLR

According to the amount of data required from the test signers, previously signer-independent SLR works can be broadly divided into two main groups: (i) signer adaptation approaches, where a previously trained model is adapted to a new signer by using a small amount of signer specific data, and (ii) truly signer independent methodologies, in which a generic model robust for new test signers is built without using data of those test signers.

The former signer adaptation approaches were greatly inspired by speaker adaptation methods from the speech recognition research [225, 223, 96]. Von Agris *et al.* [225] used maximum likelihood linear regression (MLLR) and maximum a posteriori (MAP) estimation for signer adaptation. Later, in [223], they extended their work by combining the eigenvoice (EV) approach [106] with MLLR and MAP to adapt trained HMMs to new signers. MLLR and MAP were the basic adaptation strategies, and the EV approach provided constraints to reduce the number of free parameters to be adapted. More recently, Kim *et al.* [96] investigated the potential of several signer normalization techniques (e.g., speed normalization) and different deep neural network adaptation strategies for the signer-independence problem. They found that while signer normalization is ineffective, a simple neural network adaptation strategy, such as fine-tuning the signer-specific neural networks on the adaptation data, is very effective.

The aforementioned methods are all supervised adaptation approaches, in the sense that the adaptation data from the new signer must be labeled. However, in practice, collecting labeled data may be a cumbersome and time-consuming task. To overcome this issue, a few works have resorted to unsupervised adaptation strategies [254, 245]. Yin *et al.*

[245] proposed a two-step weakly supervised metric learning framework to perform signer adaptation with some unlabeled sign data of the new signer. In the first step, a generic metric is learnt from the available labeled data of several different signers. In the second step, the generic metric is adapted to the new signer by considering clustering and manifold constraints along with the collected unlabeled data.

Although signer adaptation is a reasonable approach, there is still the need to collect either labeled or unlabeled data to retrain and adapt the model for a new signer. Therefore, a truly signer independent approach, which does not require any data from the new signers, would be the ideal solution for a practical SLR system. Examples of such works can be found in [257, 182, 226, 103, 94, 42, 244]. Most of them involved a huge feature engineering effort in order to build normalized feature descriptors robust to the physical variations of the signers (e.g., height, hand size, and length of the arm) and different acquisition conditions (e.g., distance to the camera). Afterwards, most of these works use HMMs or their variants for sign recognition. It is the example of the work proposed by Von Agris *et al.* [226], in which a set of 11 regional features are extracted (e.g., x and y positions, hand blobs area, the orientation of the main axis, inertia ratio, eccentricity, and compactness) and, then, normalized according to the head position and shoulders distance of the signer. Kelly *et al.* [94] introduced a novel signer-independent hand posture feature descriptor, along with an eigenspace size function which represents both qualitative and quantitative properties of a visual shape. Kong and Ranganath [103] gave particular importance to the movement of epenthesis (ME), which appears as the transition movement that connects successive signs. Concretely speaking, they removed the ME by using a segment and merge approach to decrease the inter-signer variations in ME and used a two-layer CRF classifier for sign recognition. More recently, Yin *et al.* [244] proposed an interesting and alternative approach that relies on distance metric learning. In particular, the metric is learnt by constraining the distances between the training samples and generic references of the sign classes. The references are constructed by signer invariant representations of each sign class (i.e., the average of all samples within the specific class). Afterwards, a two-step iterative optimization strategy is employed to obtain more appropriate references and update the corresponding distance metric alternately.

Although the methods mentioned above have promoted a significant evolution in the signer-independent research, there are still many opportunities for improvement. A major weakness across all the methods is related to the fact that representation and metric learning is not jointly performed. It is well known that the recent success of deep learning approaches, particularly those using CNNs, in tasks like object detection and recognition, has been extended to the SLR problem. The underlying motivation is to automatically learn multiple levels of representations directly from the data. Examples of such works can be found in

[158, 101, 235, 142, 107]. However, none of these works explicitly constrains the learned representations to be signer invariant.

7.2.2 Deep domain adaptation

Domain adaptation aims at learning from a *source* domain a well-performing model on a different (but related) *target* domain. The ultimate goal is then to solve the domain shift (distribution difference) between *source* and *target* domains. Deep domain adaptation leverages the representation learning power of deep learning techniques to the domain adaptation task [40]. The underlying idea is to embed domain adaptation into the representation learning process and, hence, learn a deep feature representation that is domain-invariant as well as semantically meaningful. In a broad sense, deep domain adaptation methods can be categorized into *discrepancy-based*, *reconstruction-based*, and *adversarial-based* approaches [232].

The idea behind *discrepancy-based* approaches is to fine-tune a deep neural network model, previously trained in the *source* domain, with labeled or unlabeled data of the *target* domain. Several fine-tuning mechanisms have been employed. Most of them use the information about the *target* class labels in order to guide the entire transferring knowledge process [96]. Other works attempt to align the statistical distribution shift between the *source* and *target* domains using well-known distance-measures, such as Maximum Mean Discrepancy (MMD) [125, 240, 124, 82] or KL divergence [256]. Embedding metric learning in deep neural networks is another mechanism that can be used to further reduce the distance between samples from different domains with the same class, while increasing the separability of those samples with different class labels [255, 236, 90].

Reconstruction-based domain adaptation approaches typically follow an encoder-decoder framework to jointly learn a domain-invariant representation by a shared encoder, while maintaining the domain-specific characteristics by a reconstruction loss in the *source* and *target* domains. Notable examples of such works can be found in [82] and [256]. Hu *et al.* [82] proposed a deep transfer metric learning method that brings together some of the ideas of *discrepancy-based* and *reconstruction-based* approaches. Specifically, the marginal Fisher analysis criterion is applied to ensure both intra-class compactness and inter-class separability, while the MMD criterion is used to minimize the distribution difference between domains. In addition, to further preserve the local manifold of input data samples in the representation space, an autoencoder regularization of the *source* and *target* domains is performed. The work proposed by Zhuang *et al.* [256] follows a similar encoder-decoder architecture. However, the latent distribution difference between *source* and *target* domains is minimized in terms of KL divergence.

Adversarial-based domain adaptation approaches attempt to either learn domain-invariant feature representations [70] or map representations between domains [29], using an adversarial training framework. Ganin *et al.* [70] proposed a deep neural network composed by an encoder, which maps from the input data to a latent representation, and two classifiers operating on top, namely: (i) a task-specific classifier, for predicting the task-specific labels, and (ii) a domain-classifier, for predicting the domain label. During the training stage, the parameters of the encoder network are optimized in order to minimize the loss of the task-specific classifier as well as to maximize the loss of the domain-classifier. In the course of the adversarial training procedure, the latent representations are encouraged to be both domain-invariant and predictive for the target task. Bousmalis *et al.* [29] proposed a generative adversarial neural network (GAN) in order to generate synthetic *target* samples from the *source* domain samples, while preserving the annotation information of the *source* domain. Once adapted, any off-the-shelf classifier can be trained for the *target*-specific task.

The problem of signer-independent SLR has some particularities that are not found in the standard domain adaptation setting. In the latter, one generally learns a model for a *target* domain by training it with a set of (labeled or unlabeled) data sampled from this domain and from the original *source* domain. In the former, we aim to learn a model that ideally performs equally well on infinitely many *target* domains (new signers), by training it with data sampled from a diverse set of source domains (known signers).

In this chapter, we propose a novel deep neural network specifically designed to tackle the signer-independent SLR problem. Different from the methodologies mentioned above, our proposed model jointly learns the representation and the classifier from the data, while explicitly imposing signer-independence in the high-level representations for a robust and truly signer-invariant sign recognition.

7.3 Learning Sign-Invariant Representations with Adversarial Training

Motivated by the inherent difficulty of designing reliable hand-crafted features to the large inter-signer variability, recent SLR systems are mostly based on deep neural networks [158, 101, 235, 142, 107]. Deep neural networks are remarkably good at figuring out reliable high-level feature representations from data. However, in previous deep SLR methodologies, the neural networks are typically trained just to predict the sign labels given the ground-truth, for instance, by minimizing the standard loss function for classification tasks (i.e., the categorical cross-entropy). Therefore, there is nothing to prevent the learned representations

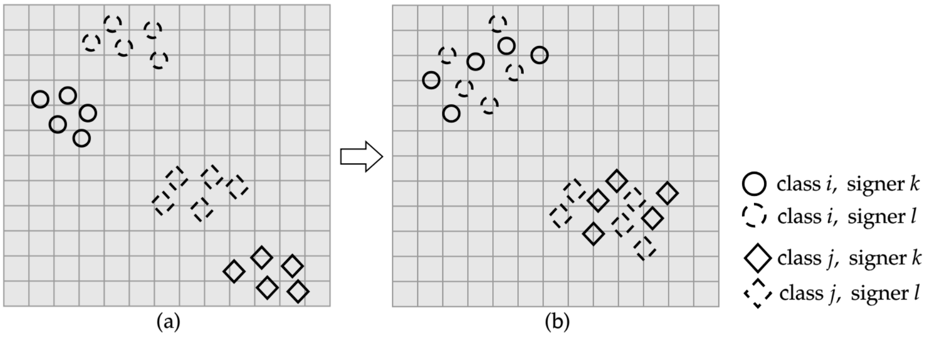


Figure 7.2: Illustration of the desired signer-independent representation space: (a) a signer-dependent representation space, in which the latent representations from the same class and different signers are far apart from each other; and (b) a signer-independent representation space, in which the latent representations from the same class and different signers are mixed.

of different signers and the same class from being far apart in the representation space and, hence, signer invariance is not ensured. A possible signer-dependent representation space is illustrated in Figure 7.2a.

Rather than adopting a conventional deep neural network topology and training strategy that simply involves training a CNN with the categorical cross-entropy, our goal is to design a deep model capable of explicitly learning signer-invariant feature representations (see Figure 7.2b). To accomplish this purpose, we introduce a deep neural network along with an adversarial training scheme that is able to learn feature representations that combine both sign discriminativeness and signer-invariance.

More specifically, let $\mathbb{X} = \{\boldsymbol{X}_i, y_i, s_i\}_{i=1}^{N}$ denote a labeled dataset of N samples, where $\boldsymbol{X}_i$ represents the i-th colour image, and y_i and s_i denote the corresponding class (sign) label and signer identity, respectively. To induce the model to learn signer-invariant representations, the proposed model comprises three distinct sub-networks:

- an *encoder* network, which aims at learning an encoding function $h(\mathbf{X};\theta_h)$, parameterized by θ_h, that maps from an input image $\mathbf{X}$ to a latent representation $\boldsymbol{h}$;

- a *sign-classifier* network, which operates on top of this underlying latent representation $\boldsymbol{h}$ to learn our task-specific function $f(\boldsymbol{h};\theta_f)$, parameterized by θ_f, that maps from $\boldsymbol{h}$ to the predicted probabilities $p(\mathrm{y}|\boldsymbol{h};\theta_f)$ of each sign class.

- a *signer-classifier* network, with the purpose of learning a signer-specific function $g(\boldsymbol{h};\theta_g)$, parameterized by θ_g, that maps the same hidden representation $\boldsymbol{h}$ to the predicted probabilities $p(\mathrm{s}|\boldsymbol{h};\theta_g)$ of each signer identity.

During the learning stage, the parameters of both classifiers are optimized in order to minimize their errors on their specific tasks on the training set. In addition, the parameters of the *encoder* network are optimized in order to minimize the loss of the *sign-classifier* network while forcing the *signer-classifier* of being a random guessing predictor. In the course of this adversarial training procedure, the learned latent representations $\boldsymbol{h}$ are encouraged to be signer-invariant and highly discriminative for sign classification. To further discourage the latent representations of retaining any signer-specific traits, we introduce an additional training objective that enforces the latent distributions of different signers to be as similar as possible. The result is a truly signer-independent model robust to new test signers.

7.3.1 Architecture

As illustrated in Figure 7.3, the architecture of the proposed model is composed by three main sub-networks or blocks, i.e. an *encoder*, a *sign-classifier* and a *signer-classifier*.

The *encoder* network attempts to learn a mapping from an input image $\mathbf{X}$ to a latent representation $\boldsymbol{h}$. It simply consists of a sequence of L_e pairs of consecutive 3×3 convolutional layers with ReLUs as nonlinearities. For downsampling, the last convolutional layer of each pair has a stride of 2. On top of that, there is a fully connected layer, also with a ReLU, representing the desired signer-invariant latent representations $\boldsymbol{h}$.

Taking the latent representations $\boldsymbol{h}$ as input, the *sign-classifier* block is composed by a sequence of L_s fully connected layers, with ReLUs as the nonlinear functions, for predicting the sign class $\hat{y} = \arg\max f(\boldsymbol{h}; \theta_f)$. Therefore, the last fully connected layer has a softmax activation function which outputs the probabilities for each sign class.

The *signer-classifier* network has exactly the same topology as the *sign-classifier* net. However, it maps the latent representations $\boldsymbol{h}$ to the predicted signer identity $\hat{s} = \arg\max g(\boldsymbol{h}; \theta_g)$. Therefore, the number of nodes of the output layer is defined accordingly to the number of signers in the training set.

7.3.2 Adversarial training

By definition, signer-invariant representations discard all signer-specific information and, as such, no function (i.e., classifier) exists that maps such representations into the correct signer identity. This naturally leads to an adversarial problem, in which: (i) a *signer-classifier* network $g(\cdot; \theta_g)$ receives latent representations $\boldsymbol{h} = h(\mathbf{X}; \theta_h)$ from an *encoder* network $h(\cdot; \theta_h)$ and tries to predict the signer identity s corresponding to image $\mathbf{X}$ and (ii) the *encoder* network tries to fool the *signer-classifier* network while still providing good representations for the

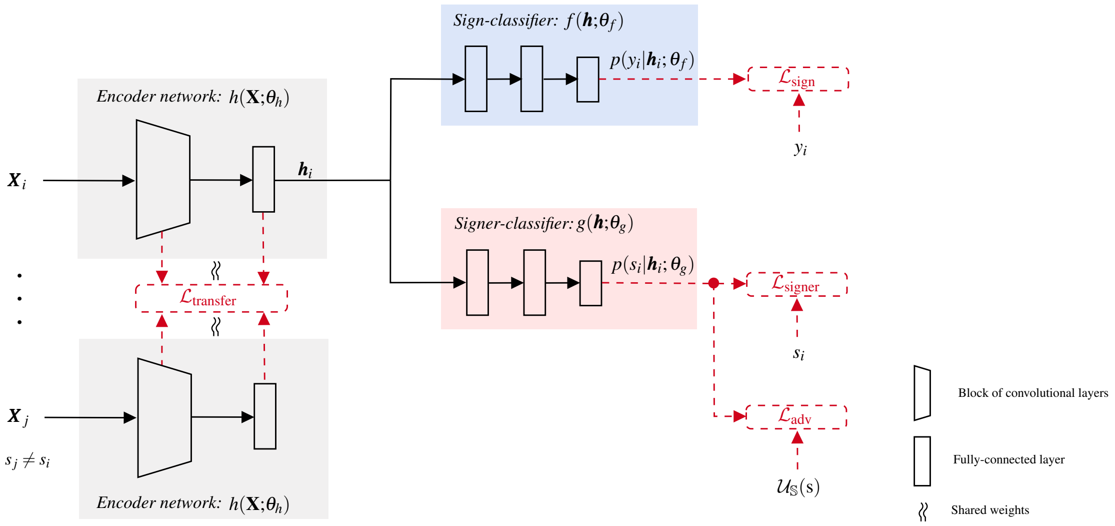

Sign-classifier: $f(\boldsymbol{h};\theta_f)$
$p(y_i|\boldsymbol{h}_i;\theta_f)$
$\mathcal{L}_{\text{sign}}$
y_i
Encoder network: $h(\mathbf{X};\theta_h)$
$\boldsymbol{X}_i$
$\boldsymbol{h}_i$
$\mathcal{L}_{\text{transfer}}$
Signer-classifier: $g(\boldsymbol{h};\theta_g)$
$p(s_i|\boldsymbol{h}_i;\theta_g)$
$\mathcal{L}_{\text{signer}}$
s_i
$\mathcal{L}_{\text{adv}}$
$\mathcal{U}_{\mathbb{S}}(\mathrm{s})$
$\boldsymbol{X}_j$
$s_j \neq s_i$
Encoder network: $h(\mathbf{X};\theta_h)$
Block of convolutional layers
Fully-connected layer
Shared weights

Figure 7.3: The architecture of the proposed signer-invariant neural network. It comprises three main sub-networks or blocks, i.e. an *encoder*, a *sign-classifier* and a *signer-classifier*.

sign-classifier network $f(\cdot;\theta_f)$, which in turn receives the same representations $\boldsymbol{h}$ and aims to predict the sign label y corresponding to image $\mathbf{X}$.

Therefore, the *signer-classifier* network shall be trained to minimize the negative log-likelihood of correct signer predictions:

$$\min_{\theta_g} \mathcal{L}_{\text{signer}}(\theta_h,\theta_g) = -\frac{1}{N}\sum_{i=1}^{N} \log p(s_i|h(\boldsymbol{X}_i;\theta_h);\theta_g) \tag{7.1}$$

In the perspective of the *encoder*, the predictions of the *sign-classifier* should be as accurate as possible and the predictions of the *signer-classifier* should be kept close to uniform, meaning that this latter classifier is not capable of doing better than random guessing the signer identity. Formally, this may be translated into the following constrained objective:

$$\min_{\theta_h,\theta_f} \mathcal{L}_{\text{sign}}(\theta_h,\theta_f) = -\frac{1}{N}\sum_{i=1}^{N} \log p(y_i|h(\boldsymbol{X}_i;\theta_h);\theta_f), \tag{7.2}$$

$$\text{subject to } \frac{1}{N}\sum_{i=1}^{N} D_{\text{KL}}(\mathcal{U}_{\mathbb{S}}(\mathrm{s})||p(\mathrm{s}|h(\boldsymbol{X}_i;\theta_h);\theta_g) \leq \varepsilon, \tag{7.3}$$

where D_{KL} is the KL divergence and $\mathcal{U}_{\mathbb{S}}(\mathrm{s})$ denotes the discrete uniform distribution on the random variable s, defined over the set of identities $\mathbb{S}$ in the training set. Here, $\varepsilon \geq 0$ determines how far from uniform the *signer-classifier* predictions are allowed to be (as measured by the KL divergence). The choice of the uniform distribution implies the underlying assumption that the training set is balanced relatively to the number of examples per signer (which should be true for most practical datasets). When this is not the case, the empirical distribution of signer identities in the training set may be used instead.

The constraint inequality (7.3) may be rewritten as:

$$\mathcal{L}_{\text{adv}}(\theta_h, \theta_g) = -\frac{1}{N|\mathbb{S}|}\sum_{i=1}^{N}\sum_{\text{s}\in\mathbb{S}} \log p(\text{s}|h(\boldsymbol{X}_i;\theta_h);\theta_g) \leq \varepsilon + \log|\mathbb{S}|, \tag{7.4}$$

and the constrained optimization problem may be equivalently formulated as:

$$\min_{\theta_h,\theta_f} \mathcal{L}(\theta_h, \theta_f, \theta_g) = \mathcal{L}_{\text{sign}}(\theta_h, \theta_f) + \lambda \mathcal{L}_{\text{adv}}(\theta_h, \theta_g), \tag{7.5}$$

where $\lambda \geq 0$ depends on ε and $\mathcal{L}_{\text{adv}}$ plays the role of an adversarial loss with respect to the signer classification loss $\mathcal{L}_{\text{signer}}$.

This objective and the structure of our model are similar to those used in [70], in the context of domain adaptation, and in [66], to learn anonymized representations for privacy purposes. However, the former uses the negative signer classification loss as the adversarial term (i.e., $\mathcal{L}_{\text{adv}} \leftarrow -\mathcal{L}_{\text{signer}}$), which is not lower bounded, leading to high gradients and difficult optimization. The latter addresses this problem by replacing this term with the absolute difference between the adversarial loss as defined in equation (7.4) and the signer classification loss (i.e., $\mathcal{L}_{\text{adv}} \leftarrow |\mathcal{L}_{\text{adv}} - \mathcal{L}_{\text{signer}}|$). This option has a nice information theoretic interpretation as being an empirical upper-bound for the mutual information between the distribution of signer identities and the distribution of latent representations. Nonetheless, there exist infinitely many (non-uniform) distributions for which this loss vanishes. Our choice, besides being clearly lower bounded by the entropy of the uniform distribution, $\log|\mathbb{S}|$, is minimum if and only if $p(\text{s}|h(\boldsymbol{X}_i;\theta_h);\theta_g) \equiv \mathcal{U}_{\mathbb{S}}(\text{s}), \forall i$, meaning that the *signer-classifier* block is completely agnostic relatively to the signer identity of the training data.

7.3.3 Signer-transfer training objective

To further encourage the latent representations $\boldsymbol{h}$ to be signer-invariant, we introduce an additional term in objective (7.5), the so-called signer-transfer loss $\mathcal{L}_{\text{transfer}}$. The core idea of $\mathcal{L}_{\text{transfer}}$ is to enforce the latent distributions of different signers to be as similar as possible. In practise, this is achieved by minimizing the difference between the hidden representations of different signers, at each layer of the *encoder* network. To measure the signer's distribution difference at the m-th layer, $m = 1,...,M$, we compute a distance $\mathcal{D}^{(m)}$ between the hidden representations $h^{(m)}(\cdot;\theta_h)$ of two signers s and t at the output of that layer, such that:

$$\mathcal{D}^{(m)}(s,t;\theta_h) = \Big|\Big|\frac{1}{N_s}\sum_{i:\ s_i=s} h^{(m)}(\boldsymbol{X}_i;\theta_h) - \frac{1}{N_t}\sum_{j:\ s_j=t} h^{(m)}(\boldsymbol{X}_j;\theta_h)\Big|\Big|_2^2, \tag{7.6}$$

where $||\cdot||_2$ is the ℓ^2-norm, and N_s and N_t denote the number of training examples of signers s and t, respectively. Accordingly, the signer-transfer loss at the m-th layer is the sum of the pairwise distances between all signers, i.e.:

$$\mathcal{L}^{(m)}_{\text{transfer}}(\theta_h) = \sum_{s\in\mathbb{S}} \sum_{\substack{t\in\mathbb{S},\\ t\neq s}} \mathcal{D}^{(m)}(s,t;\theta_h) \tag{7.7}$$

The overall signer-transfer loss $\mathcal{L}_{\text{transfer}}$ is then a weighted sum of the losses computed at each layer of the *encoder* network, such that:

$$\mathcal{L}_{\text{transfer}}(\theta_h) = \sum_{m=1}^{M} \beta^{(m)}\ \mathcal{L}^{(m)}_{\text{transfer}}(\theta_h), \tag{7.8}$$

where $\beta^{(m)} \geq 0$ is a hyperparameter that controls the relative importance of the loss obtained at the m-th layer. By combining (7.5) and (7.8), the *encoder* and *sign-classifier* networks are trained to minimize the following loss function:

$$\min_{\theta_h,\theta_f}\ \mathcal{L}(\theta_h,\theta_f,\theta_g) = \mathcal{L}_{\text{sign}}(\theta_h,\theta_f) + \lambda\mathcal{L}_{\text{adv}}(\theta_h,\theta_g) + \gamma\mathcal{L}_{\text{transfer}}(\theta_h), \tag{7.9}$$

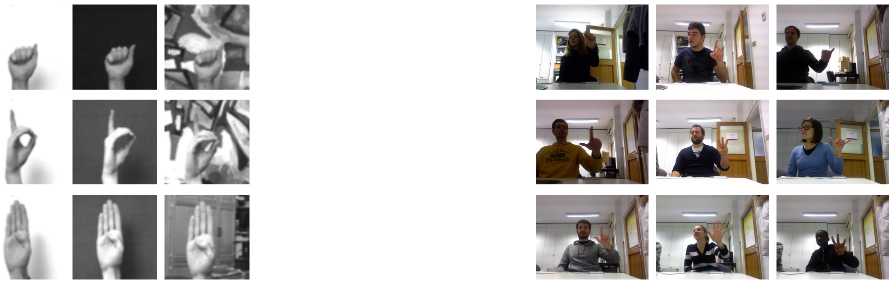
where $\gamma \geq 0$ is the weight that controls the relative importance of the signer-transfer term.

7.3.4 Training strategies

Summing up, the adversarial training procedure is organized by alternatively either training both the *encoder* and the *sign-classifier* in order to minimize objective (7.9) or training the *signer-classifier* in order to minimize objective (7.1).

7.3.5 Inference

Inference in the proposed adversarial deep neural network model simply consists of a forward pass through the *encoder* and *sign-classifier* networks, such that the sign prediction is given by $\hat{y} = \arg\max f(\boldsymbol{h};\theta_f)$, where $\boldsymbol{h} = h(\mathbf{X};\theta_h)$.



(a) Jochen-Triesch [212].

(b) MKLM [131, 132].

Figure 7.4: Illustrative samples of the datasets used in the experiments.

7.4 Experimental Evaluation

The experimental evaluation of the proposed model was performed using two publicly available SLR databases: the Jochen-Triesch database [212], and the MKLM database [131, 132]. Jochen-Triesch [212] is a dataset of 10 hand signs performed by 24 signers against three different types of backgrounds: uniform light, uniform dark and complex.

Experiments on Jochen-Triesch were conducted using the standard evaluation protocol of this dataset [91], in which 8 signers are used for the training and the remaining 16 signers are used for testing. MKLM [131, 132] contains a total of 10 signs, each one repeated 10 times by 14 different signers. In this dataset, the performance of the models is assessed using 5 random splits, created with signer-independence, yielding at each split a training set of 10 signers, a validation set of 2 signers and a test set of 2 signers.

7.4.1 Implementation details

7.4.1.1 Pre-processing

As a pre-processing step, the manual signs must be extracted from the noisy background of the images. For the images of Jochen-Triesch database, this task is performed by using the available bounding box annotations of the hands. As this kind of annotation is not available for the images of the MKLM dataset, the automatic hand detection algorithm, previously introduced on Chapter 6, was used. The images are then cropped and resized to the average sign size of the training set. Finally, the image pixel values are normalized to be in the range $[-1, 1]$. This normalization procedure ensures that each pixel (i.e., input parameter) has a similar distribution, providing faster training convergence.

7.4.1.2 Baselines

Throughout this section, the proposed model is compared with state-of-the-art methods for each dataset. Nevertheless, to further attest the robustness of the proposed model, two different baselines are also implemented:

- (Baseline 1) A CNN trained from scratch with ℓ^2 regularization. For a fair comparison, the architecture of the baseline CNN corresponds to the architecture of the *encoder* network followed by the *sign-classifier* network of the proposed model.
- (Baseline 2) A CNN with the Baseline 1 topology, but trained with the triplet loss [178].

The triplet loss was originally proposed in the face recognition context [178], in order to minimize the distance between an *anchor* and a *positive* sample, both of them with the same person identity, while maximizing the distance between the *anchor* and a *negative* sample of a different identity. Although the triplet loss concept has been explored in many other biometric domains, to the best of our knowledge, it is the first time that the triplet loss is being applied for signer-independent SLR purposes. Therefore, the implementation of Baseline 2 could be considered by itself a contribution of this thesis. Here, the underlying idea is to use the triplet loss to impose signer-independence in the representation space. Specifically, let $\boldsymbol{h}_{y_i,s_i}$ be an *anchor* latent representation, and $\boldsymbol{h}_{y_p,s_p}$ and $\boldsymbol{h}_{y_n,s_n}$ represent *positive* and *negative* latent representations, respectively. The triplet loss $\mathcal{L}_{\text{triplet}}$ used to train the implemented Baseline 2 is defined as follows:

$$\mathcal{L}_{\text{triplet}} = \frac{1}{N}\sum_{i=1}^{N}\Big[||\boldsymbol{h}_{y_i,s_i} - \boldsymbol{h}_{y_p,s_p}||^2 - ||\boldsymbol{h}_{y_i,s_i} - \boldsymbol{h}_{y_n,s_n}||^2 + \alpha\Big], \tag{7.10}$$

where $y_p = y_i$, and $y_n \neq y_i$. This means that while *anchor* and *positive* latent representations have to be from the same sign class, their signer identity may or may not change. On the other hand, *anchor* and *negative* representations are from different sign classes, whereas their signer identity may also change. In our experiments, the margin enforced between *positive* and *negative* pairs was fixed at $\alpha = 1$. In addition, following [183], we have adopted an *online* triplet generation strategy, by selecting the hardest *positive*/*negative* samples within every mini-batch. In order to train Baseline 2 in an end-to-end fashion for sign classification, the overall loss function to be minimized also contains a classification loss term, such that:

$$\mathcal{L} = \rho\,\mathcal{L}_{\text{triplet}} + \mathcal{L}_{\text{sign}}, \tag{7.11}$$

Table 7.1: Hyperparameters sets.

Hyperparameters	Acronym	Set
Leaning rate	-	$\{1e^{-04}, 1e^{-03}\}$
ℓ^2-norm coefficient	-	$\{1e^{-05}, 1e^{-04}\}$
$\mathcal{L}_{\text{triplet}}$ weight	ρ	$\{0.1, 0.5, 1, 5, 10\}$
$\mathcal{L}_{\text{adv}}$ weight	λ	$\{0.1, 0.5, 0.8, 1, 3\}$
$\mathcal{L}_{\text{transfer}}$ weight	γ	$\{1.5e^{-04}, 2e^{-04}, 4e^{-04}, 1e^{-03}\}$

where $\rho \geq 0$ is the weight that controls the relative importance of the triplet loss, and $\mathcal{L}_{\text{sign}}$ corresponds to the categorical cross-entropy as defined in equation (7.2).

7.4.1.3 Training, architecture, and hyperparameters details

All deep models were implemented in PyTorch and trained with the Adam optimization algorithm using a batch size of 32 samples. For reproducibility purposes, the source code as well as the weights of the trained models are publicly available online[1]. The hyperparameters that are common to all the implemented models (i.e., learning rate and ℓ^2 regularization weight) as well as some hyperparameters that are specific to the proposed model (i.e., λ and γ) and to the implemented Baseline 2 (i.e., ρ) were optimized by means of a grid search approach and cross-validation on the training set (see Table 7.1 for more details). The signer-transfer penalty $\mathcal{L}_{\text{transfer}}$ is applied to the last two layers of the *encoder* network with a relative weight of 1. Regarding the model's architecture, the number of consecutive convolutional layers pairs L_e was set to 3, which results in a total of 6 convolutional layers. The number of filters starts as 32, which is then doubled after each convolutional pair. The dense layer on top of the *encoder* network has 128 neurons. The number of dense layers of both classifiers L_s was set to 3, and the number of nodes of each hidden layer was set as 128.

During the training stage of all the implemented models, besides ℓ^2 regularization and dropout, a randomized data augmentation scheme was also employed. As previously introduced on section 6.4.1, the adopted data augmentation procedure is based on both geometric and color transformations. The underlying idea is to further increase the robustness of the models to the wide range of hand gestures positions, poses, viewing angles as well as to different illumination conditions and contrasts.

7.4.2 Results and discussion

Experiments on the Jochen-Triesch and MKLM databases are summarized in Tables 7.2 and 7.3, respectively. The results on the Jochen-Triesch database are presented in terms

[1] https://github.com/pmmf/SI-SLR

Table 7.2: Jochen-Triesch experimental results. The results are reported in terms of average classification accuracy. The first block of the table presents the results of state-of-the-art SLR methods. The second block depicts the results of the proposed model and of both implemented baselines.

Method	Classification accuracy (%)		
	Background		
	Uniform	Complex	Both
Just *et al.* [91]	92.79	81.25	87.92
Kelly *et al.* [94]	91.80	-	-
Dahmani *et al.* [42]	93.10	-	-
CNN (Baseline 1)	97.50	74.38	89.79
CNN with Triplet loss (Baseline 2)	98.13	75.63	90.63
Proposed method	**98.75**	**91.25**	**96.25**

of average classification accuracy in the overall test set as well as against each specific background type (i.e., uniform and complex). For the MKLM database, Table 7.3 depicts the average classification accuracy computed across all the 5 test splits, as well as the minimum and maximum accuracy value achieved by each method.

The most interesting observation is the superior performance of the proposed model. Specifically, the proposed model provides the best overall classification accuracy on both SLR databases, clearly outperforming both implemented baselines and all the previous state-of-the-art models. In complex scenarios, as reported in Table 7.2, the proposed model surpasses all the other methods by a large margin (i.e., 91.25% against 81.25%, 74.38% and 75.63%). In addition, by analyzing the standard deviation as well as the minimum and maximum accuracy values, it possible to observe that the proposed model is the method with the lowest variability, yielding consistently high accuracy rates across all test splits of the MKLM dataset (see Table 7.3). These results attest the robustness of the proposed model and its capability of better dealing with the large inter-signer variability that exists in the manual signing process of sign languages. Interestingly, the obtained results also reveal that the implemented baselines are in fact fairly strong models, both of them outperforming most of the state-of-the-art methods on both datasets.

Table 7.4 illustrates the effect of each proposed training scheme by itself. For this purpose, the proposed model was trained either (i) with just the adversarial procedure, without the signer-transfer $\mathcal{L}_{\text{transfer}}$ loss, or (ii) with just the $\mathcal{L}_{\text{transfer}}$ penalty on the *encoder* network without adversarial training. The results clearly demonstrate the complementary effect between the two training procedures, as their combination provides the best overall classification accuracy. Interestingly, each training scheme outperforms on its own both baselines and state-of-the-art methods.

Table 7.3: MKLM experimental results. The results are reported in terms of average and standard deviation (std), minimum (min) and maximum (max) classification accuracy across all the test splits. The first block of the table presents the results of state-of-the-art SLR methods. The second block depicts the results of the proposed model and of both implemented baselines.

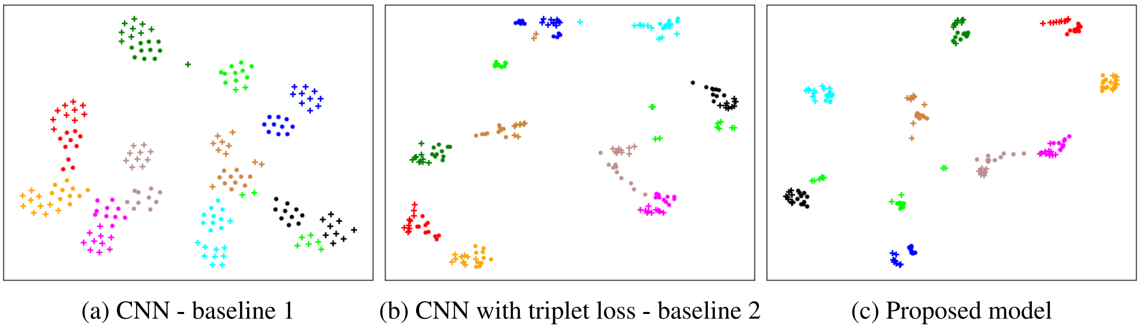
Method	Classification accuracy (%)		
	average (std)	min	max
Marin *et al.* [131]	89.71 (-)	-	-
CNN (Baseline 1)	89.90 (8.81)	73.00	98.00
CNN with Triplet loss (Baseline 2)	91.40 (3.93)	86.50	96.50
Proposed method	**94.80 (3.53)**	**89.50**	**100.00**

Table 7.4: The effect of each training procedure in the proposed model. The results in the last column are replicated from Tables 7.2 and 7.3 as they include both training procedures.

Dataset	Classification accuracy (%)		
	Only adversarial training	Only $\mathcal{L}_{\text{transfer}}$ penalty	Both
Jochen-Triesch	95.21	94.38	**96.25**
MKLM	94.00	94.10	**94.80**

7.4.3 Latent space visualization

To further demonstrate the effectiveness of the proposed model in promoting signer-invariant latent representation spaces, we have performed a visual inspection of the latent representations through the t-distributed stochastic neighbor embedding (t-SNE) [217] (see Figure 7.5). These plots clearly demonstrate the better capability of the proposed model of imposing signer-independence in the latent representations. The proposed model yields a latent representation space in which representations of the same signer and different classes are close to each other and well mixed, while it keeps latent representations of different classes far apart. By analyzing the t-SNE plot of Baseline 1, it is possible to observe that the latent representations of different signers and the same class tend to be far apart in the latent space. In addition, there is some overlapping between clusters of different classes. Although Baseline 2 (CNN with the triplet loss) promoted slightly improvements over the standard baseline CNN, the proposed model achieved by far the best signer-invariance and class separability.



(a) CNN - baseline 1 (b) CNN with triplet loss - baseline 2 (c) Proposed model

Figure 7.5: Two-dimensional projection of the latent representation space using the t-SNE [217]. Markers ● and + represent 2 different test signers, while the different colors denote the 10 sign classes.

7.5 Summary

In this chapter, we present a novel adversarial training objective, based on representation learning and deep neural networks, specifically designed to tackle the signer-independent SLR problem.

The underlying idea is to learn signer-invariant latent representations that preserve as much information as possible about the signs, while discarding the signer-specific traits that are irrelevant for sign recognition. For this purpose, we introduce an adversarial training procedure for simultaneously training an *encoder* and a *sign-classifier* over the target sign variables, while preventing the latent representations of the *encoder* to be predictive of the signer identities. To further discourage the underlying representations of retaining any signer-specific information, we propose an additional training objective that enforces the latent distributions of different signers to be as similar as possible. Experimental results demonstrate the effectiveness of the proposed model in several SLR databases.

In Appendix B, we further demonstrate how to extend the proposed adversarial training objective for other applications (e.g., biometric liveness detection), in which it is desirable to learn feature representations invariant to some specific domain or aspect.

Chapter 8

Signer-Independent Sign Language Recognition: Part II

The content presented in this Chapter was submitted for publication in [206]:

- **Ferreira, P. M.**, Pernes, D., Rebelo, A., and Cardoso, J. S. (2019b). Desire: Deep signer-invariant representations for sign language recognition. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–16

The current chapter expands our ideas previously presented in chapter 7, in which we proposed an adversarial training objective for the signer-independent SLR problem. Although the previous proposed adversarial SLR model represents a major step forward, there is always an inherent training instability regarding any adversarial framework. Herein, we address the signer-independent problem by exploiting another type of machine learning algorithm, i.e., a CVAE. The key idea of our CVAE-based model is to explicitly learn a distribution over latent representations, conditionally independent of the signer identity. Accordingly, the learned latent representations will preserve as much information as possible about the signs, and discard signer-specific traits that are irrelevant for recognition. Experimental results demonstrate further signer-independent SLR improvements.

8.1 Introduction

The previous chapter of this thesis represented a breakthrough in the signer-independent research field. To recap: in chapter 7, we proposed a truly signer-independent SLR model that, based on adversarial neural networks, demonstrated a better generalization capability to unseen test signers than current state-of-the-art SLR methods. However, given some concerns related to the underlying adversarial training process of the proposed signer-invariant model, we consider that further improvements can be made in this area.

This chapter extends our previous work on signer-independent SLR in two different ways. First, the signer-independent SLR problem is addressed using a different type of machine learning algorithm, namely a generative model, in particular, a CVAE. The underlying idea of using a CVAE-base model relies on its ability to learn latent representations whose conditional posterior distribution, given the image and its sign label, is independent of the signer identity. Second, the experimental evaluation of the models is extended to another SLR database, and additional state-of-the-art deep domain adaptation methods are also considered for comparison. Summing up, the reason for our choice of a CVAE-based model is 4-fold:

- The greater stability of the CVAE training process when compared to adversarial training. Training adversarial neural networks requires finding a Nash equilibrium of a minimax two-player game. Sometimes it is possible to find the Nash equilibrium using gradient descent, but sometimes it is not. So, adversarial training is unstable compared to CVAE training.

- CVAEs are a probabilistic graphical model whose explicit goal is latent modelling. CVAEs provide a latent representation space very naturally, at the bottleneck of the encoder-decoder, where meaningful constraints can be added in order to promote signer-invariance.

- The possibility of exploiting the regularizing effect of the stochastic noise introduced during the CVAE training process, due to Monte Carlo sampling. This noise may be combined with other constraints to introduce further variability in the training process, preventing overfitting and promoting the desired signer-invariance property.

- As claimed by several research works [143, 28, 214], the performance of generative models is usually much better than the discriminative one for limited amounts of training data (as it happens in most SLR datasets).

The rest of the chapter is organized as follows: The proposed CVAE-based signer-independent model, along with the proposed loss function and regularization schemes, are fully described in Section 8.2. Section 8.3 reports the experimental evaluation of the proposed methodology, in which a comparison with state-of-the-art and baseline methods is performed. Finally, conclusions are presented in Section 8.4.

8.2 Learning Sign-Invariant Representations with a Generative Model

To specifically tackle the signer-independent SLR problem, we propose a novel deep neural network that aims to learn **De**ep **S**igner-**I**nvariant **Re**presentations, the so-called DeSIRe.

More specifically, let $\mathbb{X} = \{\boldsymbol{X}^{(r_i)}_{y_i,s_i}, y_i, s_i\}_{i=1}^{N}$ denote a labeled dataset of N samples, where $\boldsymbol{X}^{(r_i)}_{y_i,s_i}$ represents the i-th colour image, and y_i, s_i and r_i denote the corresponding class label, signer identity and sign/gesture repetition, respectively. In this context, a sign repetition corresponds to an image of the same signer and class (sign), acquired in a different time instant.

To induce the model to learn signer-invariant representations, the proposed neural network is composed by two main modules or components: (i) a CVAE, and (ii) a classifier. The high-level block diagram of the proposed DeSIRe model is depicted in Figure 8.1. The underlying idea of the CVAE is to learn signer-invariant latent representations $\mathbf{z}$ of the input data $\mathbf{X}$. The CVAE can be thought as a teacher model for the classifier, as the distribution over latent representations $\mathbf{z}$ is used to regularize the hidden representations $\boldsymbol{h}$ of the classifier. These hidden representations $\boldsymbol{h}$ are then fed into an MLP for a robust signer-independent SLR.

Specifically, the CVAE consists of an encoder and a decoder network, parameterized by θ_e and θ_d, respectively. The purpose of the encoder network is to learn a distribution $q(\mathbf{z}|\mathbf{X}, \mathrm{y}, \mathrm{s}; \theta_e)$ which approximates the true posterior distribution of the latent code $\mathbf{z}$ given the image $\mathbf{X}$, the class label y and the signer identity s. By conditioning the posterior distribution on s and y, we intend to facilitate the task of the encoder (i.e., to preserve the relevant sign information, while discarding signer-specific properties). Here, the key idea is to learn latent codes whose conditional posterior distribution is independent of the signer identity, that is $q(\mathbf{z}|\mathbf{X}, \mathrm{y}, \mathrm{s}; \theta_e) = q(\mathbf{z}|\mathbf{X}, \mathrm{y}; \theta_e)$. Equivalently, latent codes are conditionally independent of the signer identity given the image and its class if and only if:

$$q(\mathbf{z}|\mathbf{X}, \mathrm{y}, \mathrm{s} = s_i; \theta_e) \ = \ q(\mathbf{z}|\mathbf{X}, \mathrm{y}, \mathrm{s} = s_k; \theta_e), \tag{8.1}$$

for any two distinct signers s_i and s_k. In order to promote this signer-independence property, the loss function includes a term that penalizes deviations from this equality. However, if no additional care is taken, this condition would compete with the reconstruction objective, since reconstructing an image implies preserving as much information about the image as possible, including signer-specific information. Therefore, the signer identity is fed as an additional input to the decoder network. Moreover, it does not always coincide with the signer identity used in the encoding step. By doing this, the burden of learning signer-specific information is left to the decoder network and, consequently, latent codes need not preserve the signer identity.

Intuitively, as the latent vector $\mathbf{z}$ is sampled from $q(\mathbf{z}|\mathbf{X}, \mathrm{y}, \mathrm{s}; \theta_e)$, the latent representations $\mathbf{z}$ will preserve as much information as possible about the class (sign), and discard the irrelevant parts that are characteristic of each signer. The loss function is defined in such a manner that it encourages similarity between the latent codes $\mathbf{z}$ and the hidden representations $\boldsymbol{h}$ of the classifier module. The classifier is then trained on these signer-invariant representations for a robust signer-independent SLR. Formally, $g(\boldsymbol{h}; \theta_g)$ represents our task-specific function, parameterized by θ_g, that maps from the hidden representation $\boldsymbol{h}$ to the predicted sign class $\hat{y}$, and $f(\mathbf{X}; \theta_f)$ denotes an encoding function, parameterized by θ_f, that maps the input image $\mathbf{X}$ to its hidden representation $\boldsymbol{h}$.

8.2.1 Architecture

As shown in Figure 8.1, the architecture of the DeSIRe model comprises a CVAE and a classifier.

8.2.1.1 CVAE

The CVAE consists of an encoder and a decoder. The encoder network attempts to learn a mapping from an input image $\mathbf{X}$, its class label y, and signer identity s to a latent representation $\mathbf{z}$. These additional conditional variables, y and s, are incorporated in the encoder network by simply concatenating them as extra channels with the input image $\mathbf{X}$. In this case, both y and s are represented categorically using one-hot encoding. Therefore, the encoder network simply consists of a sequence of several 3×3 convolutional layers with batch-normalization and Leaky Rectified Linear Units (LeakyReLUs) as nonlinearities. For downsampling, the stride length of every convolution is set to 2. On top of that, there are two output fully connected layers, with linear activation functions, describing the mean $\boldsymbol{\mu}_e(\mathbf{X}, \mathrm{y}, \mathrm{s}; \theta_e)$ and the log-variance $\log \boldsymbol{\sigma}_e^2(\mathbf{X}, \mathrm{y}, \mathrm{s}; \theta_e)$ of the latent space distribution $q(\mathbf{z}|\mathbf{X}, \mathrm{y}, \mathrm{s}; \theta_e)$.

The decoder module will then generate a latent code $\mathbf{z}$ by sampling from $q(\mathbf{z}|\mathbf{X}, \mathrm{y}, \mathrm{s}; \theta_e)$ and proceed for the reconstruction of the original input $\mathbf{X}$. In practice, the latent code $\mathbf{z}$, which is represented by a fully connected layer with batch normalization and a LeakyReLU as nonlinearity, is concatenated with a one-hot representation of the signer identity s to be fed to the decoder network. The decoder network, in its turn, comprises several 2D transposed convolutions for up-sampling and densifying the incoming activations. Every transposed convolutional layer is followed by batch-normalization and a LeakyReLU. The output layer also consists of a transposed convolutional layer but with a hyperbolic tangent activation function in order to output the reconstruction $\boldsymbol{\mu}_d(\mathbf{z}, \mathrm{s}; \theta_d)$ of the normalized input $\mathbf{X}$.

8.2.1.2 Classifier

The implemented classifier module follows a typical CNN architecture for classification tasks. It starts with a block of convolutional layers for feature extraction purposes, implementing the function $h = f(\mathbf{X}; \theta_f)$. This is followed by a block of fully connected layers for sign classification, which predicts the sign class $\hat{y} = g(\boldsymbol{h}; \theta_g)$. In particular, the convolutional block comprises a sequence of several pairs of consecutive 3×3 convolutional layers with ReLUs as nonlinearities. For downsampling, the last convolutional layer of each pair has a stride of 2.

The fully connected block consists of a sequence of fully connected layers with ReLUs as the nonlinear functions. The last fully connected layer has a softmax activation function which outputs the probabilities for each sign class.

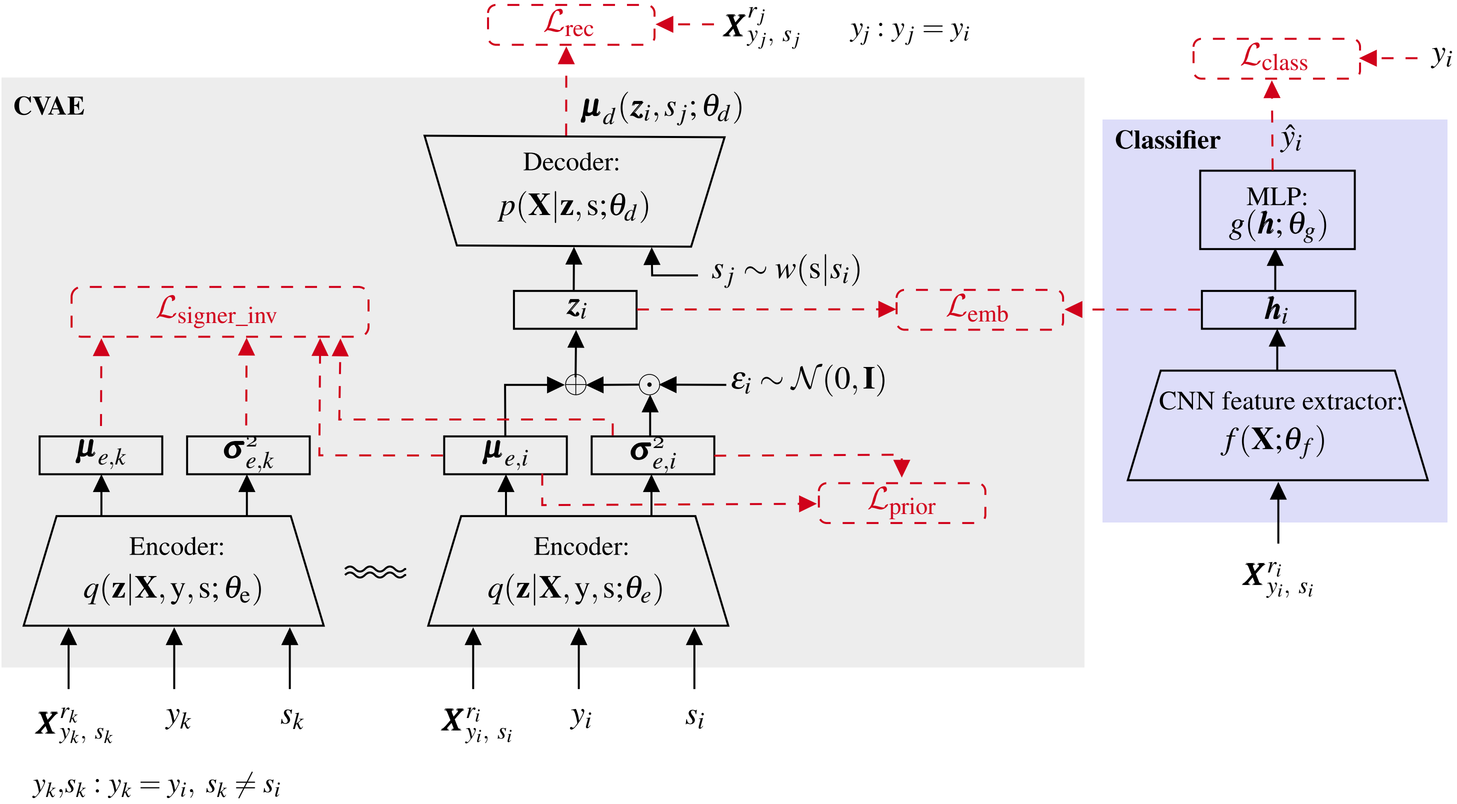

$\mathcal{L}_{\text{rec}}$
$\boldsymbol{X}^{r_j}_{y_j,\,s_j}$
$y_j : y_j = y_i$
$\mathcal{L}_{\text{class}}$
y_i
CVAE
$\boldsymbol{\mu}_d(\boldsymbol{z}_i, s_j; \theta_d)$
Classifier
$\hat{y}_i$
Decoder:
$p(\mathbf{X}|\mathbf{z}, \mathrm{s}; \theta_d)$
MLP:
$g(\boldsymbol{h}; \theta_g)$
$s_j \sim w(\mathrm{s}|s_i)$
$\mathcal{L}_{\text{signer\_inv}}$
$\boldsymbol{z}_i$
$\mathcal{L}_{\text{emb}}$
$\boldsymbol{h}_i$
$\varepsilon_i \sim \mathcal{N}(0, \mathbf{I})$
CNN feature extractor:
$f(\mathbf{X}; \theta_f)$
$\boldsymbol{\mu}_{e,k}$
$\boldsymbol{\sigma}^2_{e,k}$
$\boldsymbol{\mu}_{e,i}$
$\boldsymbol{\sigma}^2_{e,i}$
$\mathcal{L}_{\text{prior}}$
Encoder:
$q(\mathbf{z}|\mathbf{X}, \mathrm{y}, \mathrm{s}; \theta_{\mathrm{e}})$
Encoder:
$q(\mathbf{z}|\mathbf{X}, \mathrm{y}, \mathrm{s}; \theta_e)$
$\boldsymbol{X}^{r_i}_{y_i,\,s_i}$
$\boldsymbol{X}^{r_k}_{y_k,\,s_k}$
y_k
s_k
$\boldsymbol{X}^{r_i}_{y_i,\,s_i}$
y_i
s_i
$y_k, s_k : y_k = y_i,\ s_k \neq s_i$

Figure 8.1: The architecture of the proposed DeSIRe deep neural network for signer-independent SLR. It comprises two main modules or components, i.e. a Conditional Variational Autoencoder (CVAE) and a Classifier.

8.2.2 Loss function

Training the proposed DeSIRe model is achieved by minimizing the following loss function with respect to parameters $\Theta = \{\theta_e, \theta_d, \theta_f, \theta_g\}$:

$$\mathcal{L}(\Theta) \;=\; \mathcal{L}_{\text{CVAE}}(\theta_d, \theta_e) \;+\; \lambda_1\, \mathcal{L}_{\text{emb}}(\theta_e, \theta_f) \;+\; \lambda_2\, \mathcal{L}_{\text{class}}(\theta_f, \theta_g), \tag{8.2}$$

where $\lambda_1, \lambda_2 \geq 0$ are the weights that control the interaction between the loss terms.

The ultimate goal of the CVAE loss, $\mathcal{L}_{\text{CVAE}}$, is to explicitly impose signer independence on the learned latent representations $\mathbf{z}$. This is achieved by encouraging the CVAE to learn latent representations $\mathbf{z}$ whose conditional posterior distribution is independent of the signer identity s. In this regard, $\mathcal{L}_{\text{CVAE}}$ is defined by:

$$\mathcal{L}_{\text{CVAE}}(\theta_d, \theta_e) \;=\; \mathcal{L}_{\text{rec}}(\theta_d) \;+\; \alpha_1\, \mathcal{L}_{\text{prior}}(\theta_e) \;+\; \alpha_2\, \mathcal{L}_{\text{signer\_inv}}(\theta_e), \tag{8.3}$$

where $\alpha_1, \alpha_2 \geq 0$ are hyperparameters that control the relative importance of each loss term. The first two terms, $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{prior}}$, resemble the loss function of a standard CVAE, containing some special modifications for promoting signer-independence in the latent space. The reconstruction loss $\mathcal{L}_{\text{rec}}$ encourages the decoder to learn how to reconstruct the input data $\mathbf{X}$. For the decoder, we assume that the conditional likelihood of the data $\mathbf{X}$ given the latent code $\mathbf{z}$ and the signer identity s follows a Gaussian distribution. Accordingly, as previously explained in chapter 3.2.4, the reconstruction loss corresponds to the mean-squared error between a training image and a generated image. Here, however, instead of working with pairs of ground-truth images together with their respective reconstructions, we make a slight modification that further promotes signer-invariant encodings. Specifically, we compute the mean-squared error between the j-th D-dimensional training image $\boldsymbol{X}_{y_j,s_j}^{(r_j)}$ and the generated D-dimensional image $\boldsymbol{\mu}_d(\boldsymbol{z}_i, s_j; \theta_d)$ which is produced by the decoder when fed with the encoding $\boldsymbol{z}_i$ of the i-th training image and with the signer identity s_j of the j-th training image:

$$\mathcal{L}_{\text{rec}}(\theta_d) \;=\; \frac{1}{ND} \sum_{i=1}^{N} ||\boldsymbol{X}_{y_j,s_j}^{(r_j)} - \boldsymbol{\mu}_d(\boldsymbol{z}_i, s_j; \theta_d)||^2, \tag{8.4}$$

where $y_j = y_i$, $\boldsymbol{z}_i$ is sampled from $q(\mathbf{z}_i|\boldsymbol{X}_{y_i,s_i}^{(r_i)}, y_i, s_i; \theta_e)$ using the reparameterization trick (3.37), s_j is sampled from a distribution $w(\mathrm{s}|s_i)$, defined below, and r_j is sampled uniformly from the set of available sign repetitions:

$$w(\mathrm{s}|s_i) \;=\; \begin{cases} 1-\rho, & \text{if } \mathrm{s} = s_i, \\ \\ \frac{\rho}{|\mathbb{S}|-1}, & \text{if } \mathrm{s} \in \mathbb{S} \setminus \{s_i\}. \end{cases} \tag{8.5}$$

Here, $\mathbb{S}$ is the set of signer identities in the training data, $|\mathbb{S}|$ denotes its cardinality and $\rho \in (0,1)$ is a hyperparameter. By sampling the identity s_j of the ground-truth image from $w(\mathrm{s}|s_i)$, in a proportion ρ of the cases the decoder will be trained to reconstruct an image of a different subject (but same gesture class) than the one that was used to produce the encoding. This procedure further discourages the latent codes to preserve signer-specific information and therefore aims to reduce inter-signer variability. On the other hand, by sampling the gesture repetition r_j, the decoder will also be trained to reconstruct a distinct image of the same person and gesture class as the image that produced the encoding. Here, the purpose is to gain robustness to intra-signer variability. Although less problematic than the former, this type of variability is also relevant since the same signer does not always repeat the same gesture in exactly the same way. Moreover, different image acquisition conditions (e.g. background, illumination, distance to the camera, etc.) from one repetition to another also result in intra-signer variability.

The $\mathcal{L}_{\text{prior}}$ corresponds to the KL divergence between the posterior and the prior as commonly used in a standard CVAE:

$$\begin{aligned} \mathcal{L}_{\text{prior}}(\theta_e) &= \\ &= \frac{1}{NL} \sum_{i=1}^{N} D_{\text{KL}}(q(\mathbf{z}_i|\boldsymbol{X}^{(r_i)}_{y_i,s_i}, y_i, s_i; \theta_e)||\mathcal{N}(\mathbf{z}_i|0,\boldsymbol{I})) \\ &= \frac{1}{2NL} \sum_{i=1}^{N} \sum_{l=1}^{L} \mu^2_{e,i,l} + \sigma^2_{e,i,l} - 1 - \log \sigma^2_{e,i,l}, \end{aligned} \tag{8.6}$$

where L is the dimension of the latent space and $\mu_{e,i,l}$ and $\sigma_{e,i,l}$ denote the l-th elements of the vectors $\boldsymbol{\mu}_e(\boldsymbol{X}_i, y_i, s_i; \theta_e)$ and $\boldsymbol{\sigma}_e(\boldsymbol{X}_i, y_i, s_i; \theta_e)$, respectively.

To further discourage the latent codes $\mathbf{z}$ of retaining signer-specific information, we introduced a signer-invariance term, i.e. $\mathcal{L}_{\text{signer\_inv}}$, in the CVAE loss function. $\mathcal{L}_{\text{signer\_inv}}$ encourages the conditional posterior distribution of latent codes $\mathbf{z}$, given the image $\mathbf{X}$ and its class y, to be independent of the signer identity s. In this regard, the $\mathcal{L}_{\text{signer\_inv}}$ loss is defined as the KL divergence between conditional posterior distributions of $\mathbf{z}$, conditioned on the same class but also on different signer identities:

$$\begin{aligned}
\mathcal{L}_{\text{signer\_inv}}(\theta_e) \;&=\\
&= \frac{1}{NL}\sum_{i=1}^{N} D_{\text{KL}}\left(q(\mathbf{z}_i|\boldsymbol{X}^{(r_i)}_{y_i,s_i}, y_i, s_i; \theta_e) \big|\big| q(\mathbf{z}_k|\boldsymbol{X}^{(r_k)}_{y_k,s_k}, y_k, s_k; \theta_e) \right)\\
&= \frac{1}{2NL}\sum_{i=1}^{N}\sum_{l=1}^{L}\left(\frac{(\mu_{e,i,l}-\mu_{e,k,l})^2}{\sigma^2_{e,k,l}} + \frac{\sigma^2_{e,i,l}}{\sigma^2_{e,k,l}} - 1 + \log\sigma^2_{e,k,l} - \log\sigma^2_{e,i,l} \right), \qquad (8.7)
\end{aligned}$$

where $y_k = y_i$ and s_k is sampled uniformly from $\mathbb{S} \setminus \{s_i\}$. The second equality follows from the fact that both distributions are Gaussian and so their KL divergence may be computed analytically, like in equation (8.6).

The signer-invariant latent representations $\mathbf{z}$ learned by the CVAE are then used to regularize the hidden representations $\boldsymbol{h}$ of the classifier. Such regularization is promoted by the $\mathcal{L}_{\text{emb}}$ loss term, which encourages the latent representations of the CVAE and the classifier to be as similar as possible. Following this idea, the embedding loss $\mathcal{L}_{\text{emb}}$ is defined to minimize the expected mean-squared error between $\mathbf{z}$ and $\boldsymbol{h}$, that is:

$$\mathcal{L}_{\text{emb}}(\theta_e, \theta_f) \;=\; \frac{1}{NL}\sum_{i=1}^{N} \mathbb{E}_{\mathbf{z}_i \sim q(\mathbf{z}_i|\boldsymbol{X}^{(r_i)}_{y_i,s_i}, y_i, s_i; \theta_e)} ||\mathbf{z}_i - \boldsymbol{h}_i||^2. \qquad (8.8)$$

In practice, we replace equation (8.8) by its Monte Carlo approximation with one sample, which yields:

$$\mathcal{L}_{\text{emb}}(\theta_e, \theta_f) \;=\; \frac{1}{NL}\sum_{i=1}^{N} ||\boldsymbol{z}_i - \boldsymbol{h}_i||^2, \qquad (8.9)$$

where $\boldsymbol{z}_i$ is sampled from $q(\mathbf{z}_i|\boldsymbol{X}^{(r_i)}_{y_i,s_i}, y_i, s_i; \theta_e)$, again using the reparameterization trick (3.37). This approximation has an extra regularizing effect on the classifier network, by introducing some stochastic noise in its training routine.

Finally, the classification loss, $\mathcal{L}_{class}$, trains the model to predict the output sign labels and corresponds to the categorical cross-entropy, defined by:

$$\mathcal{L}_{\text{class}}(\theta_f, \theta_g) \;=\; -\frac{1}{N}\sum_{i=1}^{N} \log p(y_i|\boldsymbol{X}^{(r_i)}_{y_i,s_i}; \theta_f, \theta_g), \qquad (8.10)$$

where $p(y|\boldsymbol{X}; \theta_f, \theta_g)$ is the predicted probability that a given image $\boldsymbol{X}$ belongs to its ground-truth class y, according to the current classifier parameters θ_f and θ_g.

8.2.3 Training strategies

Our model aims to learn signer-invariant latent representations to be used for a robust signer-independent sign classification. To accomplish this goal, our DeSIRe model is composed by two main components, namely a CVAE and a classifier. Specifically, the task of the CVAE module is to learn signer-invariant representations $\mathbf{z}$ from the input data $\mathbf{X}$, which are further used to impose a signer-independence property on the latent representations of the classifier module $\boldsymbol{h}$. However, we have experimentally observed that the classification task is much easier than the reconstruction task of the CVAE. That is, the classifier tends to overfit the data with fewer training epochs than the CVAE, learning embeddings that are essentially not signer invariant. In order to avoid this behavior, we have adopted an annealing strategy to define the classification weight λ_2. Specifically, at the start of training, this weight is set to zero, so that the CVAE learns to produce signer-invariant latent representations. At this stage, the CVAE behaves as a pure teacher model for the classifier network and, therefore, the $\mathcal{L}_{\text{emb}}$ error is backpropagated only through the classifier. After a few epochs, the weight λ_2 starts increasing according to a sigmoid annealing schedule and the $\mathcal{L}_{\text{emb}}$ loss starts to be backpropagated through the CVAE encoder as well. This procedure will endow the CVAE with a better sense of the classification task. As a result, the model will be able to learn signer-invariant representations that are, in fact, highly discriminative for the sign recognition task.

Following [32] and in order to stabilize the training of the CVAE, we have employed a similar annealing strategy to define the KL divergence weights of the prior and signer-invariant loss terms, α_1 and α_2, respectively.

8.2.4 Inference

During the training stage, the CVAE module plays the role of a teacher model for the classifier. Accordingly, the CVAE can be discarded at inference time. Therefore, inference in the DeSIRe simply consists of a forward pass through the classifier network, such that $\hat{y} = g(\boldsymbol{h}; \theta_g)$ and $\boldsymbol{h} = f(\mathbf{X}; \theta_f)$.

8.3 Experimental Evaluation

Similarly to the experimental evaluation of the previous proposed adversarial signer-independent SLR model (see section 7.4), the performance assessment of the proposed DeSIRe model was performed using both Jochen-Triesch and MKLM databases, along with the same evaluation protocols. This allows a direct comparison between both proposed signer-independent SLR models. In this chapter, the experimental evaluation of the models was further extended to the proposed signLangDB database (see chapter 5, for more details).

While the sign gestures available on the Jochen-Triesch dataset were acquired under lab-controlled conditions, the gestures of both MKLM and signLangDB datasets were performed in a more spontaneous and natural signing process, under non-controlled scenarios. Therefore, the inter-signer variability becomes more noticeable on both MKLM and signLangDB datasets rather than on Jochen-Triesch.

8.3.1 Implementation details

Baselines, and compared methods

During this section, the DeSIRe model will be directly compared with the previous proposed adversarial signer-independent SLR model and both implemented baselines, previously presented in section 7.4, as well as all with state-of-the-art SLR methods that followed the same evaluation protocol. Nevertheless, to further demonstrate the effectiveness of the proposed model, two state-of-the-art deep domain adaptation methods were also implemented:

- (Baseline 1) A CNN trained from scratch with ℓ^2 regularization. It has the same network architecture as the classifier component of the proposed DeSIRe model.

- (Baseline 2) A CNN with a same topology of Baseline 1, but trained with the triplet loss [178]. The adaptation of the triplet loss concept for the signer-independent problem is fully explained in section 7.4.

- (Ganin *et al.* [70]) The adversarial-based domain adaptation method proposed in [70]. The application of this method to our problem implied two main changes in the original method: 1 - the binary domain-classifier (source vs. target domain) was extended to $|\mathbb{S}|$ classes (number of training signers); 2 - since our data is fully annotated (sign classes and signer identities are always available), training was performed in a fully supervised fashion.

- (Hu *et al.* [82]) The reconstruction-based domain adaption method proposed in [82]. The implementation of this methodology for our particular task also implied the generalization of the original model from a *source* domain to $|\mathbb{S}|$ *source* domains. Therefore, the MMD criterion is applied in a pairwise fashion between all training signers. In addition, given the nature of our input data (i.e., images), the topology of the layers in the original encoder-decoder network was changed from fully connected to convolutional.

Table 8.1: Hyperparameters sets.

Hyperparameters	**Acronym**	**Set**
$\mathcal{L}_{\text{emb}}$ weight	λ_1	$\{0.1,0.5\}$
$\mathcal{L}_{\text{class}}$ weight	λ_2	$\{1,5,10\}$
$\mathcal{L}_{\text{prior}}$ weight	α_1	$\{5\text{e}^{-03}, 8\text{e}^{-02}\}$
$\mathcal{L}_{\text{signer\_inv}}$ weight	α_2	$\{8\text{e}^{-02}, 4\text{e}^{-01}, 8\text{e}^{-01}\}$

Training, architecture, and hyperparameters details

The implementation of the deep neural networks was performed using the PyTorch [154] framework. As illustrated in Table 8.1, some of the most important hyperparameters were optimized by means of a grid search approach and cross-validation on the training set. The hyperparameter of the $|\mathbb{S}|$-nomial distribution $w(\text{s}|s_i)$, defined for the proposed sampling scheme of the signer identity, was set as $\rho = 0.5$. The dropout rate was empirically set as 0.5 for all the experiments.

A detailed description of the architecture of the proposed DeSIRe model is presented in Table 8.2. For illustrative purposes, the presented DeSIRe architecture considers input colour images with a resolution of 100×100 pixels, 10 signer identities in the training set and a total of 10 sign classes. It is important to stress out that, for a fair comparison, the topology of both implemented baselines follows the same architecture of the classifier component of the proposed DeSIRe model.

It is worth mentioning that the pre-processing step and the randomized data augmentation scheme, previously introduced in sections 7.4.1.1 and 7.4.1.3,respectively, are also employed to the proposed DeSIRe model and to the additionally implemented deep domain adaption methods.

8.3.2 Results on the Jochen-Triesch database

Jochen-Triesch [212] comprises a total of 10 hand posture signs from the ASL, each one performed by 24 signers and repeated against three types of backgrounds, i.e. uniform light, uniform dark and complex. There exist three images for each subject and sign, one for each background type, which we treat as the different sign repetitions, as defined in Section 8.2.

Table 8.3 compares the performance of the proposed DeSIRe model against the previous proposed adversarial signer-independent SLR model, all the implemented methods (i.e., both baselines and the domain adaptation methods of Ganin *et al.* [70] and Hu *et al.* [82]), and the state-of-the-art SLR methods that followed the same evaluation protocol. The results are

Table 8.2: A detailed description of the architecture of the proposed DeSIRe model. The output shape is described as (#filters, rows, columns).

Layer #	DeSIRe module	Layer (type)	Non-linearity	Output shape	Connected to
-	Inputs	input_x	-	(3,100,100)	-
-		input_y_2d	-	(10,100,100)	-
-		input_s_2d	-	(10,100,100)	-
-		input_s_1d	-	(10,)	-
1	$q(\mathbf{z} \mid \mathbf{X},y,s;\boldsymbol{\theta}_e)$	Concat2d-1	-	(23,100,100)	[input_x; input_y_2d; input_s_2d]
2		Conv2d-1	LeakyReLU	(64,50,50)	Concat2d-1
3		Conv2d-2	LeakyReLU	(64,25,25)	Conv2d-1
4		Conv2d-3	LeakyReLU	(128,13,13)	Conv2d-2
5		Conv2d-4	LeakyReLU	(256,7,7)	Conv2d-3
6		Conv2d-5	LeakyReLU	(512,4,4)	Conv2d-4
7		Dense-1	Linear	(128,)	Conv2d-5
8		Dense-2	Linear	(128,)	Conv2d-5
8		Dense-3	LeakyReLU	(128,)	[Dense-1; Dense-2]
10	$p(\mathbf{X} \mid \mathbf{z},s;\boldsymbol{\theta}_d)$	Concat1d-1	-	(138,)	[Dense-3; input_s_1d]
11		Reshape-1	-	(512,4,4)	Concat1d-1
12		ConvTr2d-1	LeakyReLU	(512,7,7)	Reshape-1
13		ConvTr2d-2	LeakyReLU	(256,13,13)	ConvTr2d-1
14		ConvTr2d-3	LeakyReLU	(128,25,25)	ConvTr2d-2
15		ConvTr2d-4	LeakyReLU	(64,50,50)	ConvTr2d-3
16		ConvTr2d-5	Tanh	(3,100,100)	ConvTr2d-4
17	$f(\mathbf{X};\boldsymbol{\theta}_f)$	Conv2d-6	ReLU	(32,100,100)	input_x
18		Conv2d-7	ReLU	(32,50,50)	Conv2d-6
19		Conv2d-8	ReLU	(64,50,50)	Conv2d-7
20		Conv2d-9	ReLU	(64,25,25)	Conv2d-8
21		Conv2d-19	ReLU	(128,13,13)	Conv2d-9
22		Conv2d-11	ReLU	(128,13,13)	Conv2d-10
23	$g(\boldsymbol{h};\boldsymbol{\theta}_g)$	Dense-4	ReLU	(128,)	Conv2d-11
24		Dropout-1	-	(128,)	Dense-4
25		Dense-5	ReLU	(128,)	Dropout-1
26		Dropout-2	-	(128,)	Dense-5
27		Dense-6	Softmax	(10,)	Dropout-2

presented in terms of classification accuracy in the overall test set as well as against each specific background type (i.e., uniform and complex).

A first observation is the superior performance of both proposed signer-independent SLR models. Interestingly, the proposed DeSIRe model promoted substantial gains even over the

Table 8.3: Jochen-Triesch experimental results. The results are reported in terms of average classification accuracy. The first block of the table presents the results of state-of-the-art SLR methods and the domain adaptation methods proposed by Ganin *et al.* [70] and Hu *et al.* [82]. The second block depicts the results of the proposed DeSIRe model, the previous proposed adversarial signer-independent SLR model, and of both implemented baselines.

Method	Classification accuracy (%)		
	Background		
	Uniform	Complex	Both
Just *et al.* [91]	92.79	81.25	87.92
Kelly *et al.* [94]	91.80	-	-
Dahmani *et al.* [42]	93.10	-	-
Ganin *et al.* [70]	98.13	83.75	93.33
Hu *et al.* [82]	98.75	85.63	94.38
CNN (Baseline 1)	97.50	74.38	89.79
CNN with Triplet loss (Baseline 2)	98.13	75.63	90.63
Proposed adversarial model in Chapter 7	98.75	91.25	96.25
Proposed DeSIRe model	**99.69**	**92.50**	**97.29**

previous proposed adversarial signer-independent SLR model. These results clearly represent another step forward towards the development of robust signer-independent SLR systems. Specifically, the DeSIRe model achieved an overall classification accuracy of 97.29% against 89.79%, 90.63% and 87.92% achieved by Baseline 1, Baseline 2 and the SLR method of Just *et al.* [91], respectively. However, it is worth mentioning that both domain adaption methods outperformed the implemented baselines and the previous state-of-the-art SLR methods. These results clearly attest the importance of learning signer-invariant representations for a robust SLR. Nevertheless, both proposed signer-independent models, especially the DeSIRe model, achieved by far the best overall classification accuracy.

Another interesting observation is the performance of the proposed DeSIRe model against complex backgrounds. In complex scenarios, the DeSIRe model clearly outperforms all the other methods by a large margin (92.50% against 81.25%, 83.75%, 85.63%, 74.38% and 75.63%). These results demonstrate the robustness of the proposed model to inter-signer variability as well as its capability of dealing with the large intra-signer variability of this dataset. As previously explained in Section 8.2.2, the robustness to intra-signer variability is mostly due to the proposed sampling scheme of the sign repetition introduced to the decoder network, which enforces the learned latent representations to discard this type of variability.

8.3.3 Results on the MKLM database

Experiments on the MKLM database are summarized in Table 8.4, which depicts the average classification accuracy computed across all the 5 test splits, as well as the minimum and

Table 8.4: MKLM experimental results. The results are reported in terms of average and standard deviation (std), minimum (min) and maximum (max) classification accuracy across all the test splits. The first block of the table presents the results of state-of-the-art SLR methods and the domain adaptation methods proposed by Ganin *et al.* [70] and Hu *et al.* [82]. The second block depicts the results of the proposed DeSIRe model, the previous proposed adversarial signer-independent SLR model, and of both implemented baselines.

Method	Classification accuracy (%)		
	average (std)	min	max
Marin *et al.* [131]	89.71 (-)	-	-
Ganin *et al.* [70]	94.30 (2.49)	91.50	96.50
Hu *et al.* [82]	94.10 (3.84)	87.00	97.50
CNN (Baseline 1)	89.90 (8.81)	73.00	98.00
CNN with Triplet loss (Baseline 2)	91.40 (3.93)	86.50	96.50
Proposed adversarial model in Chapter 7	94.80 (3.53)	89.50	**100.00**
Proposed DeSIRe model	**96.80 (2.38)**	**93.00**	99.00

maximum accuracy value achieved by each method. Once again, the proposed model clearly outperforms both implemented baselines, with an overall classification accuracy of 96.80% against 89.90% and 91.40% of Baseline 1 and Baseline 2, respectively. In addition, our method provides substantial improvements over both domain adaptation methods and the previous proposed adversarial SLR model. The analysis of the standard deviation also indicates that the proposed DeSIRe model yields consistently the highest accuracy rates across all test splits.

8.3.4 Results on the CorSiL-signLangDB database

As previously presented in chapter 5, the signLangDB database comprises a total of 182 isolated signs and 40 continuous sentences. For this particular signer-independent SLR experiment, we selected a subset of 31 isolated signs from 11 signers, representing the alphabet and the cardinal numbers 0 to 9 of the Portuguese sign language. All the signs were performed in a free and natural signing environment, without any clothing restriction. This variability, together with the large number of sign classes, makes this dataset a rather challenging one. The signLangDB database has a well-defined standard evaluation protocol, which consists of 6 signers for training, 1 signer for validation and the remaining 4 signers are used for testing. For reproducibility purposes, this particular signLangDB subset is also publicly available[1].

The experimental results obtained on signLangDB are presented in Table 8.5. As the signLangDB database contains a large number of sign classes (i.e. 31), the results are

[1] github.com/pmmf

Table 8.5: CorSiL-signLangDB experimental results. The results are reported in terms of top-1, top-3 and top-5 classification accuracy. The first block of the table presents the results of the domain adaptation methods proposed by Ganin *et al.* [70] and Hu *et al.* [82]. The second block depicts the results of the proposed DeSIRe model, the previous proposed adversarial signer-independent SLR model, and of both implemented baselines.

Method	Classification accuracy (%)		
	Top-1	Top-3	Top-5
Ganin *et al.* [70]	48.66	75.54	83.33
Hu *et al.* [82]	39.25	68.01	79.84
CNN (Baseline 1)	45.97	74.73	85.75
CNN with Triplet loss (Baseline 2)	42.74	72.31	81.99
Proposed adversarial model in Chapter 7	49.13	76.01	85.19
Proposed DeSIRe model	**51.88**	**76.61**	**87.90**

presented in terms of top-1, top-3 and top-5 classification accuracy. As shown in Table 8.5, the proposed DeSIRe model outperformed both the implemented baselines and the state-of-the-art domain adaption methods in all the three classification metrics. Once again, the DeSIRe model also promoted a consistent improvement with respect to the previous proposed adversarial signer-independent SLR model. However, it should be noticed that, regardless of the employed methodology, the overall performance in this database is significantly below that obtained in the other two databases. These results attest to the difficulty of the classification task on the presented database.

With the obtained results, we intend to establish the first state-of-the-art methods for the introduced signLangDB database and, hence, further encourage signer-independent SLR research.

8.3.5 Visualization of the latent space

To further demonstrate the effectiveness of the proposed model in promoting signer-independent latent representation spaces, we have performed a visual inspection of the latent representations through the t-SNE [217].

Figure 8.2 depicts the t-SNE provided by the DeSIRe model and both implemented baselines in two test splits, of the MKLM dataset, with different degrees of inter-signer variability. Figure 8.3 illustrates the t-SNE plots obtained by the domain adaptation methods for the same exact test splits. For a better visual comparability, the t-SNE plots of the proposed model are replicated from Figure 8.2 to this figure.

As it is possible to observe in Figures 8.2 and 8.3, all the implemented models achieved high classification accuracies on the test split 1 (see the top row of Figures 8.2 and 8.3).

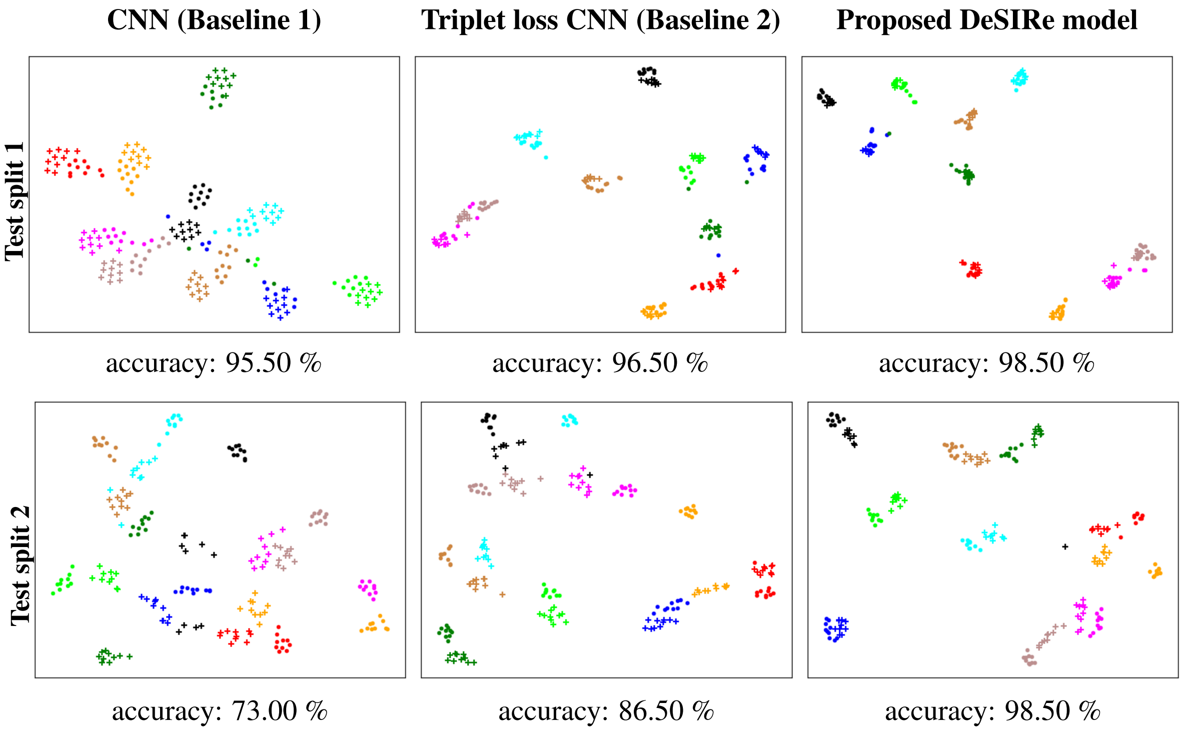


Figure 8.2: Two-dimensional projection of the latent representation space provided by the DeSIRe model and both baselines, using the t-SNE [217]. Markers ● and + represent the 2 different test signers, while the different colors denote the 10 sign classes.

Nevertheless, the t-SNE plots clearly demonstrate the better capability of the DeSIRe model of imposing signer-independence in the latent representations. The DeSIRe model yields a latent representation space in which latent representations of the same signer and different classes are close to each other and well mixed, while it keeps latent representations of different classes far apart.

The test split 2, depicted in the bottom row of Figures 8.2 and 8.3, is characterized by a larger inter-signer variability. Consequently, for this particular test split, the gains of the DeSIRe model are much more noticeable. Specifically, the DeSIRe model achieved 98.50% classification accuracy against 73.00%, 86.50%, 91.50% and 87.00% of Baseline 1, Baseline 2, and the domain adaptations methods of Ganin *et al.* [70] and Hu *et al.* [82], respectively. The t-SNE plots support these classification results (see the bottom row of Figures 8.2 and 8.3). Here, it is possible to observe that Baseline 1 completely fails in the arrangement of the latent space. Specifically, the latent representations of different signers and the same class are too far apart. In addition, there is a clear overlap between clusters of different classes. Although the CNN with the triplet loss (i.e. Baseline 2) and both domain

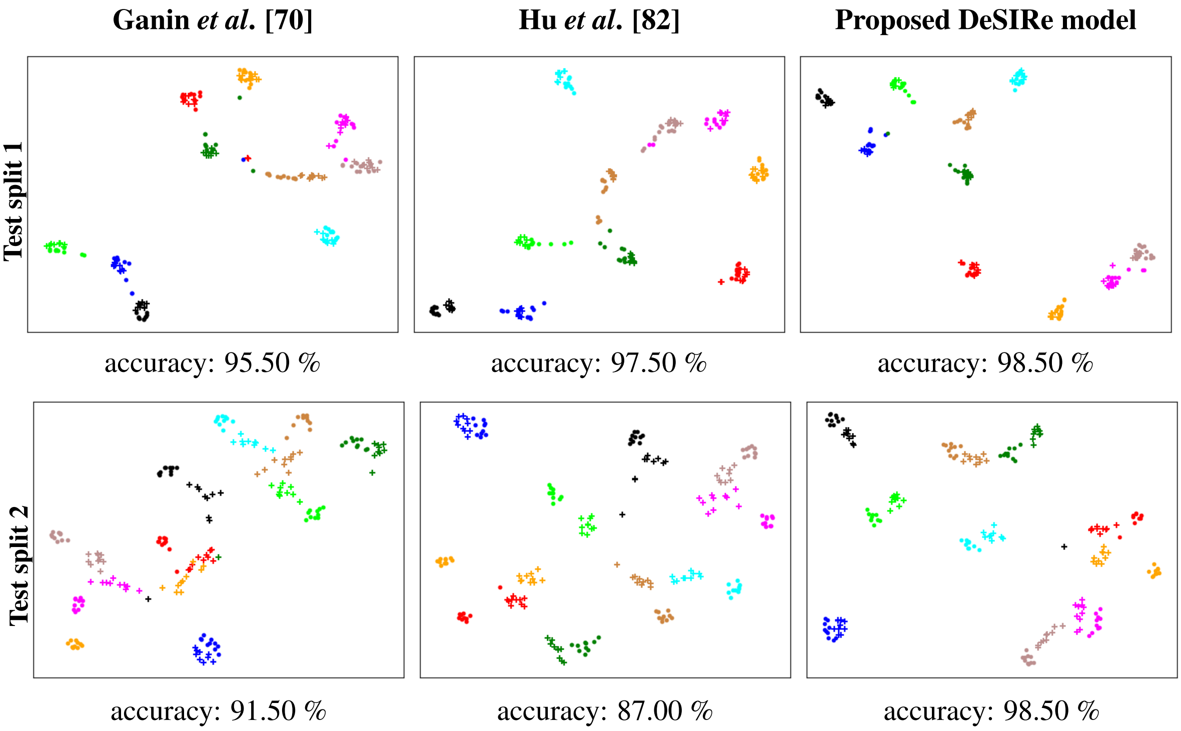


Figure 8.3: Two-dimensional projection of the latent representation space provided by the DeSIRe model and the implemented state-of-the-art domain adaptation methods proposed by Ganin *et al.* [70] and Hu *et al.* [82], using the t-SNE [217]. Markers • and + represent the 2 different test signers, while the different colors denote the 10 sign classes.

adaptation methods promoted slight improvements over the standard baseline CNN, the proposed DeSIRe model achieved by far the best inter-class separability.

8.3.6 Cluster analysis in the latent space

In order to obtain an objective quality assessment of the produced latent representations, we have evaluated how well the model is able to cluster the different sign classes (and thus ignore the signer identity) in the latent space. For this purpose, we use two cluster validation metrics: the average Silhouette coefficient [170] per cluster and the Dunn's index [57] per cluster.

The Silhouette coefficient for an observation i is computed as follows. Let be C_i the cluster (sign class) associated with the observation i. The average intra-cluster distance a_i and the minimum average inter-cluster distance b_i for the observation i are obtained as follows:

$$a_i = \frac{1}{|C_i|-1}\sum_{j\in C_i} d(i,j), \tag{8.11}$$

$$b_i = \min_{\overline{C}\neq C_i}\frac{1}{|\overline{C}|}\sum_{j\in\overline{C}} d(i,j), \tag{8.12}$$

where $|C_i|$ denotes the number of observations in the cluster $|C_i|$ and $d(i,j)$ is the Euclidean distance between the observations i and j. Then, the Silhouette index S_i for the observation i is defined as:

$$S_i = \frac{b_i - a_i}{\max(a_i,b_i)}, \tag{8.13}$$

whrere $-1 \leq S_i \leq 1$. Intuitively, clusters are desirably compact (small a_i) and well separated (large b_i), so a larger value of S_i indicates better clustering. However, this metric is defined per observation. Hence, in order to have a global measure of clustering quality, we compute the average Silhouette coefficient for each cluster.

Dunn's index follows a similar idea of measuring cluster compactness versus separation, but uses minimum and maximum distances instead of average distances, and is more sensitive to extreme and occasional errors. Specifically, the Dunn's index D_C for a cluster C is defined as the ratio between the minimum inter-cluster distance δ_C from C to all other clusters (which measures cluster separation) and the maximum intra-cluster distance Δ_C for the cluster C (which measures cluster compactness):

$$\delta_C = \min_{i\in C, j\notin C} d(i,j),\ \Delta_C = \max_{i,j\in C} d(i,j),\ D_C = \frac{\delta_C}{\Delta_C}. \tag{8.14}$$

Again, according to this metric, larger values indicate better clustering. Results are shown in Table 8.6. As anticipated by the analysis of the two-dimensional t-SNE projection in Figure 8.2, the results confirm that the DeSIRe model produces the most compact and separated sign clusters, when compared with the remaining models. This observation supports the signer-invariance property of the representations produced by the DeSIRe model: when exposed to images obtained from new signers, our model does a better job of grouping them according to the respective sign class only, ignoring the signer identity. The Baseline 2 and the domain adaptation model by Hu *et al.* [82] are also capable of producing fairly good sign clusters. This is not a surprising fact since both approaches include explicit penalties in the respective training objectives that favor compactness in the latent space among samples of the same sign class. The absence of such a compactness constraint in the adversarial approach by Ganin *et al.* [70] allows its latent representations to be more widely spread over

Table 8.6: Dunn's index and Silhouette coefficient for the sign class clusters in the latent space for the test data. These metrics were computed per cluster and the average and worst results are reported for each model and dataset.

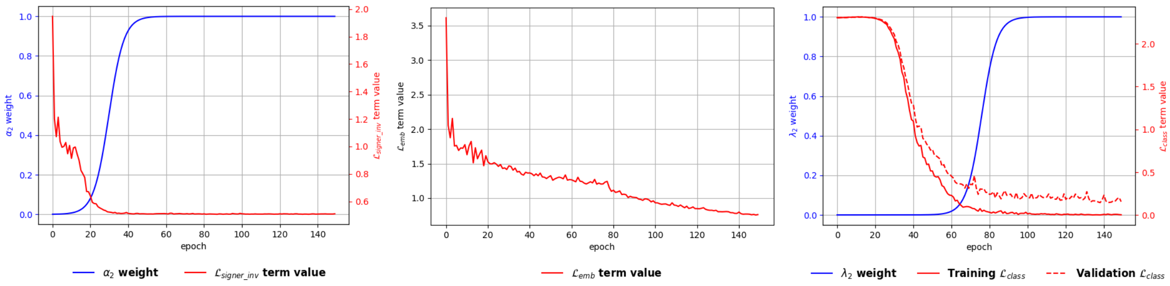
Method	Jochen-Triesch				MKLM				CorSiL-signLangDB			
	Dunn's index		Silhouette		Dunn's index		Silhouette		Dunn's index		Silhouette	
	Average	Worst	Average	Worst	Average	Worst	Average	Worst	Average	Worst	Average	Worst
Ganin *et al.* [70]	0.165	0.105	0.457	0.342	0.380	0.102	0.531	0.253	0.310	0.205	0.281	0.109
Hu *et al.* [82]	0.218	0.170	0.557	0.493	0.693	0.236	**0.653**	0.342	0.298	0.200	0.312	0.107
CNN (Baseline 1)	0.171	0.132	0.405	0.326	0.378	0.159	0.537	0.295	0.346	0.184	0.316	0.112
CNN with Triplet loss (Baseline 2)	0.249	0.179	0.509	0.453	0.559	0.208	0.623	0.171	0.359	**0,210**	0.313	0.186
Proposed DeSIRe model	**0.316**	**0.184**	**0.582**	**0.541**	**0.695**	**0.240**	0.646	**0.377**	**0.374**	0.186	**0.320**	**0.197**

the latent space. As such, according to the adopted metrics, the obtained sign clusters are comparable to those obtained using a simple CNN (Baseline 1), although the resulting sign classification accuracy is undoubtedly superior.

8.3.7 Unveiling the training behavior of the DeSIRe model

In this subsection, we further analyze the training process of the proposed DeSIRe model. Figure 8.4 shows the behavior of different loss terms, during 150 epochs of training on the MKLM dataset, with the sigmoid annealing schedules in place. Specifically, we have plotted the curves of the key loss terms, $\mathcal{L}_{\text{signer\_inv}}$ and $\mathcal{L}_{\text{emb}}$, responsible for promoting signer-invariant latent representations (see Figures 8.4a and 8.4b, respectively). In addition, we have also plotted the curve of the classification loss term $\mathcal{L}_{\text{class}}$, which trains the model to predict the output sign labels.

Figure 8.4c depicts the observed behavior for the classification loss term $\mathcal{L}_{\text{class}}$. As previously explained in Section 8.2.3, at the start of training, the classification weight λ_2 is set to zero and the $\mathcal{L}_{\text{emb}}$ error is backpropagated only through the convolutional block of the classifier module. Therefore, during the first training epochs, $\mathcal{L}_{\text{class}}$ remains at a high value, as the classifier predicts random guesses (see Figure 8.4c). Then, $\mathcal{L}_{\text{class}}$ drops quickly once the classification weight λ_2 starts increasing. This shows that the feature representations learned in the previous phase are highly discriminative for the classification task. On the other hand, the CVAE module starts to be trained on the reconstruction task as soon as the training process begins. At this stage, the sampling scheme associated with the reconstruction loss $\mathcal{L}_{\text{rec}}$ is the only mechanism promoting signer-invariance. After a few epochs, the KL weights α_1 and α_2 start increasing and the CVAE is further enforced to produce signer-invariant latent representations. In particular, Figure 8.4a shows the evolution of the signer-invariance loss $\mathcal{L}_{\text{signer\_inv}}$ together with the respective weight α_2 and attests that, at the end of training, the encoder produces signer-independent embeddings in the training data. Finally, Figure 8.4b



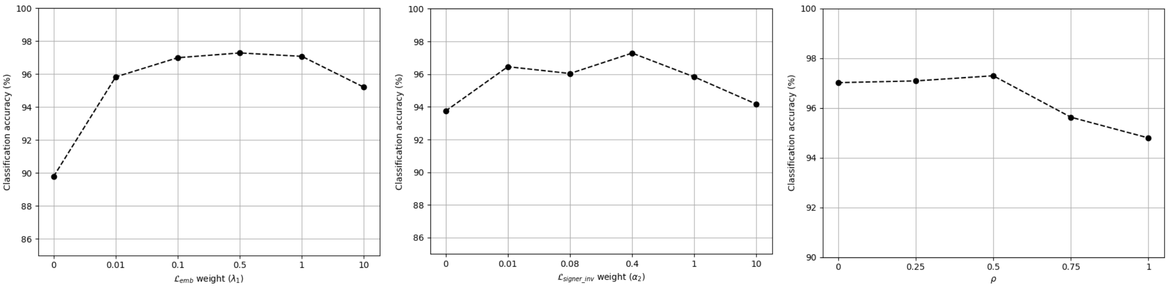
(a) (b) (c)

Figure 8.4: Training behavior of the proposed DeSIRe model: (a) the evolution of the $\mathcal{L}_{\text{signer\_inv}}$ loss term alongside the corresponding weight α_2 according to a sigmoid annealing schedule; (b) the evolution of the $\mathcal{L}_{\text{emb}}$ term value; and (c) the evolution of training and validation $\mathcal{L}_{\text{class}}$ curves alongside the corresponding weight λ_2 according to a sigmoid annealing schedule.

depicts a consistent decrease of the embedding loss $\mathcal{L}_{\text{emb}}$ throughout the entire training routine.

8.3.8 Hyperparameter sensitivity analysis

This subsection presents a sensitivity analysis of three key hyperparameters of the proposed DeSIRe model, namely the $\mathcal{L}_{emb}$ weight λ_1, the signer-invariance weight α_2 and the probability of changing the signer identity fed to the decoder network ρ. For this purpose, we plotted the curves of the average test accuracy of the proposed model with varying values of $\lambda_1 \in [0,10]$, $\alpha_2 \in [0,10]$, and $\rho \in [0,1]$ (see Figures 8.5a, 8.5b, and 8.5c, respectively). Some interesting conclusions can be drawn from these plots. Particularly, when $\lambda_1 = 0$, $\mathcal{L}_{class}$ is the only loss term still active during training. Accordingly, the proposed DeSIRe model has exactly the same behavior as Baseline 1, which results in a significant drop in the test accuracy (i.e., from 97.29% to 89.79%). When $\lambda_1 \neq 0$ and $\alpha_2 = 0$, the loss terms $\mathcal{L}_{\text{emb}}$, $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{prior}}$ become active and only $\mathcal{L}_{\text{signer\_inv}}$ is inactive. Under this setting, the test accuracy increases to 93.75% (see Figure 8.5b). Here, the classifier is trained to follow the latent representations produced by the CVAE. Although the term $\mathcal{L}_{\text{signer\_inv}}$ is not present, signer-invariance is still promoted by (i) the $\mathcal{L}_{prior}$ loss term; and (ii) by conditioning the decoder on the signer identity, which is drawn from a random distribution. Finally, when λ_1 and α_2 are both set to their optimal values, all loss terms are active and a maximum test accuracy of 97.29% is achieved. The observed accuracy gain clearly supports the beneficial regularizing effect of the $\mathcal{L}_{\text{signer\_inv}}$ loss term, which explicitly promotes signer-invariant representations.



(a) (b) (c)

Figure 8.5: Hyperparameter sensitivity analysis: (a) The average accuracy of the DeSIRe model with varying values of $\lambda_1 \in [0, 10]$ and $\alpha_2 = 0.4$ and $\rho = 0.5$ on the Triesch dataset; (b) The average accuracy of the DeSIRe model with varying values of $\alpha_2 \in [0, 10]$ and $\lambda_1 = 0.5$ and $\rho = 0.5$ on the Triesch dataset; and (c) The average accuracy of the DeSIRe model with varying values of $\rho \in [0, 1]$ and $\lambda_1 = 0.5$ and $\alpha_2 = 0.4$ on the Triesch dataset.

In addition, it is worth mentioning that the proposed DeSIRe model is quite robust to these hyperparameters as the accuracy curves remain quite stable over a large range of values (i.e., $\lambda_1, \alpha_2 \in [0.01, 1]$). The impact of ρ, which controls the proposed sampling scheme of the signer identity in the learning process, is depicted in Figure 8.5c. From this figure, it is possible to observe that ρ should be set around 0.5. The test accuracy progressively decreases when ρ falls into the interval $[0.75, 1]$. In these cases, the decoder will be trained most of the time to reconstruct an image of a different signer than the one that was used to produce the encoding. This naturally makes the reconstruction task and the overall CVAE training process too difficult, explaining the significant performance drop.

8.4 Summary

This chapter extends our previous work on the signer-independent SLR problem, improving its results. Rather than adopting a discriminative adversarial training framework, we tackled the signer-independent problem by exploiting generative models, in particular, CVAEs.

Specifically, the proposed model is composed by two main modules, namely a CVAE and a classifier. The purpose of the CVAE module is to learn latent representations of the input data, whose conditional posterior distribution, given the image and its sign label, is independent of the signer identity. During the training stage, the CVAE plays the role of a teacher model for the classifier, since the conditional posterior distribution over latent representations is used to regularize the hidden representations of the classifier. These signer-invariant hidden representations are then used for a robust signer-independent SLR recognition.

Experimental results demonstrate the robustness of the proposed model to new test signers. The proposed model provides quite promising results, outperforming the implemented baseline methods, the state-of-the-art SLR and domain adaptation methods, and the previous proposed adversarial signer-independent SLR model. Therefore, the current chapter constitutes another step forward towards the development of robust signer-independent SLR systems.

Chapter 9

Facial Expressions Recognition: a categorical approach

The content presented in this Chapter was published in [204, 203, 199]:

- **Ferreira, P. M.**, Marques, F., Cardoso, J. S., and Rebelo, A. (2018b). Physiological inspired deep neural networks for emotion recognition. *IEEE Access*, 6:53930–53943
- **Ferreira, P. M.**, Marques, F., Cardoso, J. S., and Rebelo, A. (2018a). An expression-specific deep neural network for emotion recognition. In *RecPad 2018: Conference on Pattern Recognition*, pages 1–2
- **Ferreira, P. M.**, Cardoso, J. S., and Rebelo, A. (2016). Facial key-points detection using a convolutional encoder-decoder model. In *RecPad 2016: Conference on Pattern Recognition*, pages 1–2

Automated FER has been one of the key problems in human-computer interaction, with growing application areas including SLR [39]. Facial expressions play a significant role in sign language communication. Facial expressions and head movements are used in sign languages at all levels of linguistic structure. At the phonological level, some signs have an obligatory facial component in their citation form. Facial actions also mark relative clauses, content questions, and conditionals, amongst others [61]. Therefore, automated FER shall be an integral part of an overall SLR system. This chapter is devoted to more fundamental research work on FER, whose outcomes will have the potential to be further integrated into an SLR system.

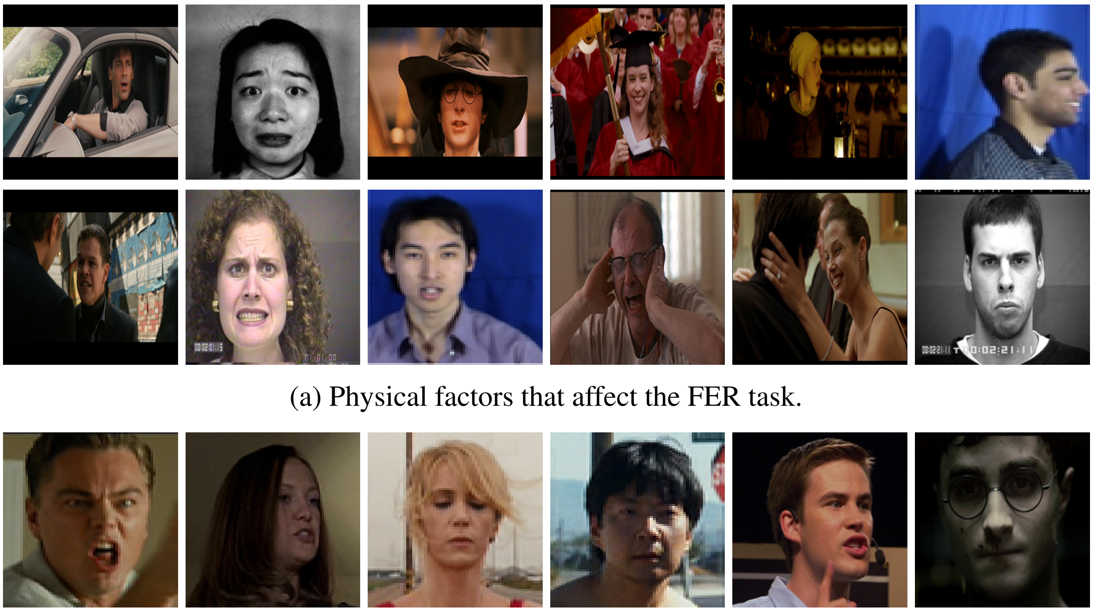
Current FER methodologies are mostly based on deep learning approaches. However, training deep neural networks for FER is still a very challenging task, since most of the available FER datasets are relatively small. Although *transfer learning* can partially alleviate the issue, the performance of deep models is still below of its full potential as deep features may contain redundant information from the pre-trained domain. Instead, we propose a novel end-to-end neural network architecture along with a well-designed loss function based on the strong prior knowledge that facial expressions are the result of the motions of some facial muscles and components. The loss function is defined to regularize the entire learning process, so that the proposed neural network can learn expression-specific features explicitly.

9.1 Introduction

Facial expressions (FEs) can be defined as the facial changes in response to a person's internal emotional state, intentions, or social communication [115]. Together with voice, language, hand gestures, and body posture, they form a fundamental communication system between humans in social contexts. FEs were introduced as a research field by Charles Darwin in his book "*The Expression of the Emotions in Man and Animals*" [43]. Since then, FEs were established as one of the most important features of human emotion recognition. In the last few years, automatic FER has attracted much attention due to its wide range of applications, such as human-robot interaction, data-driven animation, interactive games, crowd analytics, biometrics, clinical monitoring, and SLR systems due to the importance of facial expressiveness in sign languages [115, 39].

Expression recognition is a task that human beings perform daily and effortlessly, but it is not yet easily performed by computers. Although recent methods have demonstrated remarkable performances in highly controlled environments (i.e., high-resolution frontal faces with uniform backgrounds), the automatic FER in real-world scenarios is still a very challenging task [39]. Those challenges are mainly related to different acquisition conditions and to the inter-individual's facial expressiveness variability (see Figure 9.1a and Figure 9.1b, respectively). Figure 9.1b shows six subjects with the angry expression. As illustrated in the figure, the images vary a lot from each other not only in the way that the subjects show their expression, but also in lighting, brightness, viewing angle, pose, position, occlusions, and background.

The majority of existing FER systems focus on classifying 6 basic (prototypical) expressions, which have been found to be universal across cultures and subgroups, namely: happy, surprise, fear, anger, sadness, and disgust (see Figure 9.2); some systems also recognize the neutral and the contempt expressions [39]. Fewer works follow the dimensional approach, in which the FER is treated as a regression problem in a continuous two-dimensional space, usually arousal and valence [252, 104]. The higher dimensionality of the arousal/valence



(a) Physical factors that affect the FER task.

(b) Inter-individuals facial expressivenesses variability.

Figure 9.1: Illustration of the main challenges of FER. Those challenges are mainly related to: (a) several physical factors such as pose, viewing angle, occlusions and illumination; and (b) psychological factors such as the inter-individuals facial expressiveness variability.

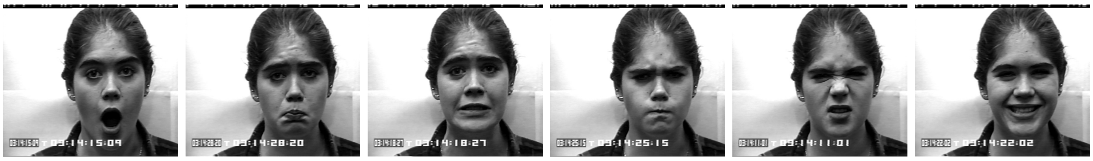


Figure 9.2: The six basic facial expressions. From left to right: surprise, sadness, fear, anger, disgust and happy.

space potentially allows describing more complex and subtle emotions. However, this richer representation of the expressions is more difficult to use in practice, since the linkage of such dimensional representation to a specific emotion is not straightforward [39]. In fact, during this thesis, we have also developed a dimensional-based FER methodology in the scope of the One-Minute Gradual-Emotional Behavior (OMG-Emotion) challenge. All details about our participation in the OMG-Emotion competition can be found in Appendix C.

An automatic facial analysis system is typically composed by three main steps: (i) face detection and/or alignment, (ii) feature extraction, and (iii) expression recognition (see Figure 9.3). Face detection and face alignment are important pre-processing steps for background removal and, then, to rotate or frontalize the face. Effective expression analysis is tightly coupled with the feature extraction step. According to the adopted feature representation, previous FER approaches can be roughly categorized into two main

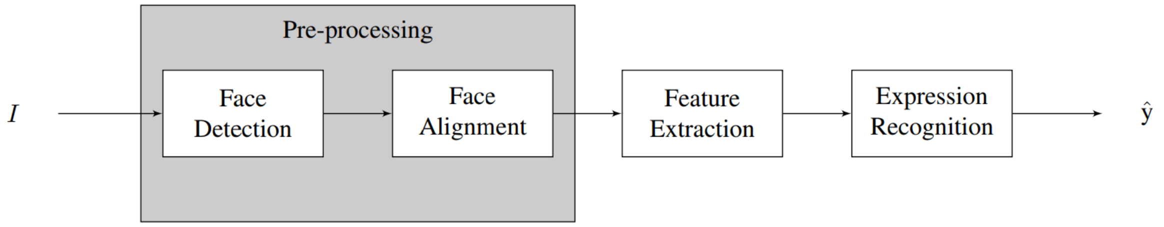


Figure 9.3: Diagram of blocks of a typical FER system, where I denotes the input image and $\hat{y}$ represents the predicted FE.

groups: geometric-based methods [253, 105, 196, 129, 228] and appearance-based methods [180, 119, 118, 193, 14, 233, 45, 24, 198, 45]. Geometric-based methods involve, in a first stage, the location of facial landmarks and/or some facial components (e.g., mouth, eyes, nose and eyebrows) and, then, the extraction of geometric features from these fiducial points. Geometric features attempt to measure distances, deformations, curvatures, and other geometric properties to represent the face geometry. Appearance-based methods rely on the principle that facial expressions involve change in local texture. Typically, a bank of filters, such as *Local Binary Patterns* [180], *Gabor filters* [119, 118], *Local Gabor Binary Patterns* [193, 14], *Local Phase Quantization* [233, 45], *Scale Invariant Feature Transform* (SIFT) [24, 198], and *Pyramids of Histograms of Gradients* [45], are applied to either the whole face or specific face regions to encode the texture. However, the performance of these hand-crafted feature extraction methods decreases in illumination changes, noise variability, changes in pose, and expression conditions [167]. Another commonly used local feature extraction method for FER is the Local Fisher Discriminant Analysis (LFDA) [163]. In a related work, Kosti *et al.* [104] recently employed the Stepwise Linear Discriminant Analysis (SWLDA) for a robust FER. However, LFDA and SWLDA fail to determine the essential assorted structure when face image space is highly nonlinear.

The recent success of deep learning approaches, particularly those using CNNs, in tasks like object detection and recognition, has been extended to the FER problem. The underlying motivation is to avoid the extraction of hand-crafted features, either geometric- or appearance-based, and the inherent difficulty of designing reliable features to the large inter-individual's facial expressiveness variability. Unlike hand-crafted feature extraction approaches, CNNs are able to automatically learn multiple levels of representations from the data, with higher levels representing more abstract concepts. In general, deep learning approaches became feasible due to two main reasons: (i) the larger amount of data that is currently available in most of the applications, and (ii) the recent advances in GPU technology. The former is crucial for training neural networks with deep architectures without overfitting, whereas the latter is crucial for performing the numerical computations required for the training procedure. However, this is not the case of the FER field, where the availability of large datasets is scarce.

To work around the problem of training high-capacity classifiers on small datasets, previous FER works have mainly resorted to (i) *transfer learning*, where a CNN is typically pre-trained in some domain-related dataset before being fine-tuned to the target dataset; and (ii) *classifier ensembles*, in which an ensemble of CNNs is created in order to combine their decisions and, hence, reduce the model's variance. However, the gains of *transfer learning* hugely depend on the source-target domain similarity and the availability of an auxiliary large dataset. In addition, the success of *classifier ensembles* requires a wide range of diverse single CNN models.

Inspired by the strong support from physiology and psychology that FEs are the result of the motions of facial muscles [58, 39, 120], a novel end-to-end deep neural network along with a well-designed loss function for FER are proposed. The loss function is defined in such a manner to regularize the entire learning process, so that the proposed model is able to automatically learn expression-specific features. The neural network is composed by three well-designed modules or components, i.e. the *facial-parts component*, the *representation component* and the *classification component*. The purpose of the *facial-parts component* is to regress a relevance map, representing the most important facial regions for the recognition. The relevance map is then used in the *representation component* in order to increase the discriminative ability of the learned features. The result is a model able to explicitly encode expression-specific features by capturing local appearance variations caused by the motion of facial muscles (e.g., frown, grin, and glare) and facial components (e.g., eyes, nose, mouth, and eyebrows). In addition, according to the level of the available data annotations, different regularization schemes, the so-called fully supervised and weakly supervised regularization schemes, are proposed. The fully supervised regularization scheme is suitable for datasets in which both facial landmarks and expressions are annotated, whereas the weakly supervised strategy just requires the annotation of the facial expressions. In order to combine the strengths of both fully supervised and weakly supervised regularization strategies, a hybrid formulation of them is also proposed.

The chapter is organized in five sections including the Introduction (Section 9.1). Section 9.2 presents the state-of-the-art methods for FER. The proposed neural network model along with the proposed regularization schemes are fully described in Section 9.3. Section 9.4 reports the experimental evaluation of the proposed methodology, in which a comparison with state-of-the-art methods is performed. Finally, Section 9.5 summarizes the Chapter.

9.2 Related Work

In the last decade, automatic facial expression recognition has been an active research topic in the artificial intelligence community due to its wide range of applications in the HCI field. Several facial expression recognition methodologies have been proposed, with an increasing progress in the recognition performance. An important part of this recent progress was achieved thanks to the emergence of deep learning approaches and more specifically with CNNs. Comprehensive surveys on automatic facial expression recognition can be found in [174, 41, 175, 39].

Different deep architectures have been proposed for FER. Song *et al.* [185], developed a very simple FER system that uses a traditional CNN architecture composed of five layers. Some conventional training strategies, such as data augmentation and dropout, were applied in order to prevent overfitting. Similar approaches are proposed in [34, 215, 181]. In [34], a slightly more complex CNN architecture is presented. Inspired by the success of GoogleNet [194], the key structure of their architecture is a parallel feature extraction block that consists of convolutional, pooling, and ReLU layers. Tang *et al.* [197] reported a small but consistent advantage of replacing the softmax layer of the CNN with a linear support vector machine. The goal was to minimize a margin-based loss instead of the conventional cross-entropy loss function.

In most cases, a deep neural network model requires a lot of training data to generalize well, a condition that is not entirely fulfilled in the FER context, where the amount of data is limited. Attempting to overcome this issue, several works have been using conventional deep learning regularization techniques (e.g., dropout, data augmentation, l_2-norm) along with *transfer learning* [144, 247], *classifier ensembles* [247, 95, 96], and *unsupervised learning* [122], which typically involves an unsupervised layer-wise training step that allows the usage of larger and unlabeled datasets.

Ng *et al.* [144] followed a transfer learning approach for deep CNN architectures, by utilizing a two-stage supervised fine-tuning process. More concretely, starting from a generic pre-training of two different CNN architectures based on the ImageNet dataset [172], a cascade fine-tuning approach is, then, applied using two different facial expression datasets. Yu *et al.* [247] propose a classification module that consists of an ensemble of multiple deep CNNs. Each CNN model is randomly initialized and pre-trained in a larger dataset before being fine-tuned on the target dataset. To combine multiple CNN models, they propose two constrained optimization frameworks to automatically learn the ensemble weights of the network responses. Similar approaches are proposed in [95, 96]. The authors propose a hierarchical architecture of a committee of deep CNNs with an exponentially-weighted decision fusion. The individual CNN models were trained varying the network architecture, input normalization, and weight initialization, in order to obtain diverse decisions boundaries.

More recently, Connie *et al.* [37] proposed a hybrid approach, in which SIFT features are merged with one of the later CNN layers. The underlying idea is to combine the strengths of hand-crafted and deep learning approaches. Experimental results suggest that the fusion approach yields an overall improvement in the FER performance.

Other deep learning techniques, such as deep belief networks (DBNs), have also been used for FER [122, 121]. It is the example of the work of Liu *et al.* [122], in which a two-step iterative learning process is used to train boosted DBNs. First, each DBN learns a nonlinear feature representation from a facial patch in an unsupervised manner. Second, these DBNs are connected through a boosted classifier and fine-tuned jointly driven by a single objective function. In this regard, the features extracted at different locations are selected and strengthened jointly according to their relative importance to the facial expression recognition. Liu *et al.* [121] propose the so-called AU-aware deep networks, in which a fixed convolutional step (i.e., application of a predefined set of hand-crafted filters) followed by a pooling step is applied to extract a feature representation. Then, the representation is grouped into a set of relevant receptive fields for each expression. Each receptive field is fed to a DBN to obtain a nonlinear feature representation, using an SVM to detect each expression independently.

In terms of motivation, the work of Liu *et al.* [121] is probably the most related to our proposed methodology, as they also explore the psychological theory that FEs can be decomposed into multiple action units. However, it should be noticed that the proposed neural network architecture, objective function, as well as the entire learning strategy, are completely different. First, the neural network proposed in [121] is not trained end-to-end. Second, they do not explore the potential of CNNs to extract expression-specific representations. As they use a set of hand-crafted filters, the modeling capacity of their model is limited by the fixed transformations (filters).

9.3 A Physiological Inspired Deep Neural Network for Emotion Recognition

While *transfer learning* across tasks has been widely applied to work around the challenge of training deep models in small datasets, such as those available for FER, the benefits of *transfer learning* are tightly coupled with the source-target domain similarity. Instead, our goal is to design a deep model by imposing domain knowledge based on the strong support from physiology and psychology that FEs are the result of the motions of facial muscles [58, 39]. The underlying idea is to explicitly drive the model towards the most relevant facial areas for the expression recognition, such as the facial components (i.e., eyes, eyebrows, nose, mouth) and expression wrinkles.

In this regard, we propose a novel deep neural network architecture along with a well-designed loss function that explicitly models both informative local facial regions and expression recognition. The result is a model that is able to jointly learn facial relevance maps and expression-specific features for a proper recognition.

To induce the model to jointly learn the most relevant facial parts along with the FER, the proposed neural network is composed by three main components, namely (i) the *facial-parts component*, (ii) the *representation component*, and (iii) the *classification component*. The purpose of the *facial-parts component* is to learn an encoding-decoding function $E(\mathrm{x};\theta_E)$, parameterized by θ_E, that maps from an input image x to a relevance map $\hat{\mathrm{x}}$ representing the probability of each pixel being relevant for recognition. The loss function is defined in a such manner that enforces sparsity and spatial contiguity on the activations of $\hat{\mathrm{x}}$. This definition is supported by the physiological fact that just small and disjoint facial regions are relevant for recognition [58]. The *representation component* aims to learn an embedding function $F(\mathrm{x},\hat{\mathrm{x}};\theta_F)$, parameterized by θ_F, that maps from an input image x and its relevance map $\hat{\mathrm{x}}$ to an hidden representation h. The relevance map $\hat{\mathrm{x}}$ that is being learned in the *facial-parts component* is then used to filter the learned representations h, enforcing them to only respond strongly to the most relevant facial parts as possible. The result is a model that produces highly discriminative representations for FER. The *classification component* is then trained on these highly discriminative representations. Formally, $G(\mathrm{h};\theta_G)$ represents a task-specific function, parameterized by θ_G, that maps from hidden representations h to the task-specific predictions $\hat{\mathrm{y}}$.

9.3.1 Architecture

As shown in Figure 9.4, the architecture of the proposed neural network comprises three main modules, i.e. the *facial-parts component*, the *representation component*, and the *classification component*.

9.3.1.1 Facial-parts component

The architecture of the *facial-parts component* consists of a convolutional path followed by a deconvolutional path, in such way that it is possible to learn a mapping between an input image x to a relevance map $\hat{\mathrm{x}}$, with the same resolution of the input.

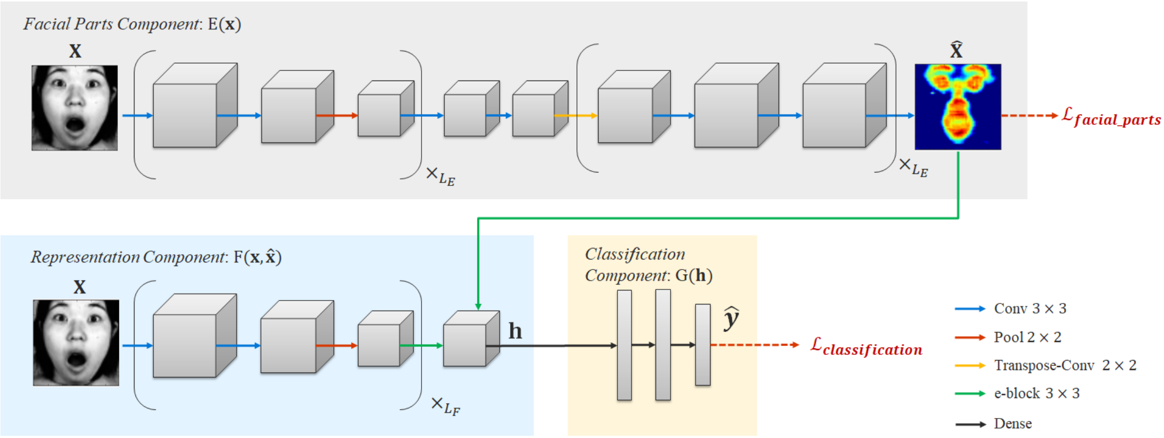

Facial Parts Component: E(x)
x
×L_E
×L_E
$\hat{x}$
$\mathcal{L}_{facial\_parts}$
Representation Component: F(x, $\hat{x}$)
x
×L_F
h
Classification Component: G(h)
$\hat{y}$
$\mathcal{L}_{classification}$
Conv 3 × 3
Pool 2 × 2
Transpose-Conv 2 × 2
e-block 3 × 3
Dense

Figure 9.4: The architecture of the proposed neural network for FER. It comprises three modules or components, i.e. the *facial-parts component*, the *representation component*, and the *classification component*.

The convolutional path follows the typical architecture of a fully convolutional network [188]. It comprises several sequences of two consecutive 3×3 convolutional layers, with ReLUs as nonlinearities, followed by a 2×2 max-pooling operation for downsampling. The number of convolutional filters is doubled at each max-pooling operation.

Every step in the deconvolutional path comprises a 2×2 transpose convolution and two 3×3 convolutions, each one followed by a ReLU. The transpose convolution is applied for up-sampling and densify the incoming features maps. At the final layer, a 3×3 convolution with a linear activation function is used to map the activations into a probability relevance map.

9.3.1.2 Representation component

The purpose of the *representation component* is to extract highly discriminative features for FER. Therefore, it starts with several sequences of convolution-convolution-pooling layers for a typical CNN feature extraction. Then, we introduce a novel building block in the network, the so-called expression block (e-block), in order to increase the discriminative ability of the learned features. As illustrated in Figure 9.5, an e-block comprises a convolutional layer and a elementwise multiplication. It takes as input the activations of the previous layer (the learned features) and the relevance map $\hat{\mathrm{x}}$. Formally, the e-block is defined as:

$$a^l = \sigma(W * a^{l-1}) \odot \hat{\mathrm{x}}, \tag{9.1}$$

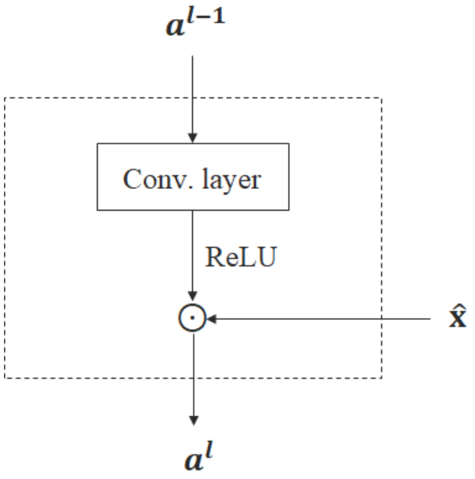


Figure 9.5: The expression block (e-block).

where $*$ and $\odot$ denote a convolution operation and an elementwise multiplication, respectively. a^{l-1} and a^l represent the input and output activations of the e-block, respectively. W represents the weights of the convolutional layer to be learned, and σ is the nonlinearity (i.e., ReLU). The biases are omitted for notation simplification. The elementwise multiplication with $\hat{\mathrm{x}}$ is performed to enforce the output activations a^l to just respond strongly to the most relevant facial parts. It should be noticed that $\hat{\mathrm{x}}$ has to be resized and cropped accordingly to the actual feature map size for a proper elementwise multiplication.

9.3.1.3 Classification component

The architecture of the *classification component* consists of a sequence of fully connected layers (or dense layers). The last layer of the CNN is a softmax output layer, which contains the output probabilities for each class label. The output node that produces the largest probability is chosen as the overall classification.

9.3.2 Learning

Inference in the proposed model is given by $\hat{\mathrm{x}} = E(\mathrm{x})$ and $\hat{\mathrm{y}} = G(\mathrm{h})$ where $\hat{\mathrm{x}}$ is the relevance map of the facial parts, $\hat{\mathrm{y}}$ is the task-specific prediction and $\mathrm{h} = F(\mathrm{x}, \hat{\mathrm{x}})$. Therefore, the goal of training is to minimize the following loss function with respect to parameters $\Theta = \{\theta_E, \theta_F, \theta_G\}$:

$$\mathcal{L} = \mathcal{L}_{\text{classification}} + \lambda\ \mathcal{L}_{\text{facial\_parts}}, \tag{9.2}$$

where $\lambda \geq 0$ is the weight that controls the interaction of the loss terms. The classification loss, $\mathcal{L}_{classification}$, trains the model to predict the output labels and corresponds to the categorical cross-entropy defined by:

$$\mathcal{L}_{\text{classification}} = -\sum_{i=1}^{N} \mathbf{y}_i^{\top} \log \hat{\mathbf{y}}_i, \tag{9.3}$$

where $\mathbf{y}_i$ is a column vector denoting the one-hot encoding of the class label for input i and $\hat{\mathbf{y}}_i$ are the softmax predictions of the model: $\hat{\mathbf{y}}_i = G(\mathbf{h}_i)$.

The purpose of the facial-parts loss, $\mathcal{L}_{facial\_parts}$, is to enforce the relevance map x̂ to encode the relative importance of each pixel to the facial expression classification. Based on the physiological support that FEs can be decomposed into several action units of facial muscles, the underlying assumption is that the relevance map x̂ should be sparse and spatially localized. It means that x̂ should take high values just in the neighborhood of important facial components (e.g., eyes, eyebrows, nose, mouth, and expression wrinkles).

To accomplish this purpose, we propose three different regularization strategies for regression of x̂, accordingly to the level of the available data annotations:

9.3.2.1 Fully Supervised Regularization

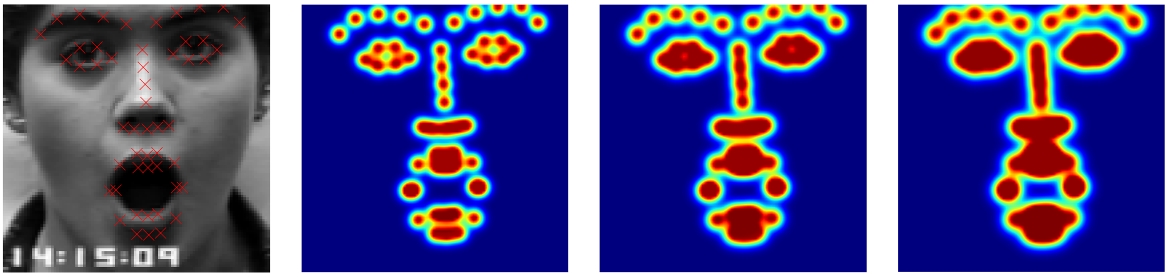
The proposed fully supervised regularization scheme requires not only the availability of the ground-truth class labels but also the annotation of the true coordinates of some facial landmarks (or key-points) located over important facial components, such as the eyes, nose, mouth, and eyebrows (see Figure 9.6a).

In this scenario, a target relevance map $\mathbf{x}_i^{target}$ for each training image i is created, $i = 1,...,N$. Let K $= \{(r,c)^j\}_{i=1}^{N}$, $j = 1,...,k$, represent the set all k annotated key-points coordinates. As illustrated in Figure 9.6, for a given training image, each facial landmark j is represented by a Gaussian, with mean at the key-point coordinates, i.e., $\mu = (r,c)^j$, and a predefined standard deviation σ. Then, the target relevance map $\mathbf{x}_i^{target}$ is simply formed by the mixture of the Gaussians of each facial landmark. The standard deviation σ should be set to control the neighborhood size around the facial landmarks (see Figures 9.6b-9.6d).

The facial-parts loss, $\mathcal{L}_{facial\_parts}$, is then defined to minimize the mean squared error between the target and the predicted relevance maps, such that:

$$\mathcal{L}_{\text{facial\_parts}} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i^{target} - \hat{\mathbf{x}}_i)^2 \tag{9.4}$$

Therefore, this loss term encourages the relevance map x̂ to take high values in the neighborhood of the most important facial components.



(a) (b) (c) (d)

Figure 9.6: Fully supervised learning scheme: (a) a training image with the true key-points coordinates superimposed (red crosses), and (b-d) examples of target relevance maps x^{target}, obtained by a superposition of Gaussians at the location of each facial landmark, with an increasing σ value.

9.3.2.2 Weakly Supervised Regularization

The weakly supervised regularization strategy does not require the annotation of the facial key-points coordinates. In this scenario, the facial-parts loss, $\mathcal{L}_{facial\_parts}$, is defined to regularize the activations of the relevance map $\hat{\mathbf{x}}$ by imposing sparsity and spatial contiguity as follows:

$$\mathcal{L}_{\text{facial\_parts}} = \sum_{i=1}^{N} \mathcal{L}_{\text{sparsity}}(\hat{\mathbf{x}}_i) + \gamma \sum_{i=1}^{N} \mathcal{L}_{\text{contiguity}}(\hat{\mathbf{x}}_i), \tag{9.5}$$

where $\gamma \geq 0$ is the weight that controls the interaction of the loss terms. The intuition is that just small and disjoint facial regions are relevant for the recognition task. In this regard, the sparsity term is defined by:

$$\mathcal{L}_{\text{sparsity}}(\hat{\mathbf{x}}) = \frac{1}{m \times n} \sum_{i,j} |\hat{\mathbf{x}}_{i,j}|, \tag{9.6}$$

where m, n denote the resolution of the relevance map $\hat{\mathbf{x}}$.

The spatial contiguity term $\mathcal{L}_{\text{contiguity}}$ encourages the activations of $\hat{\mathbf{x}}$ to be smooth and spatially localized. Then, the spatial contiguity loss is simply defined to minimize the local spatial transitions of the relevance map $\hat{\mathbf{x}}$, as follows:

$$\mathcal{L}_{\text{contiguity}}(\hat{\mathbf{x}}) = \frac{1}{m \times n} \sum_{i,j} |\hat{\mathbf{x}}_{i+1,j} - \hat{\mathbf{x}}_{i,j}| + |\hat{\mathbf{x}}_{i,j+1} - \hat{\mathbf{x}}_{i,j}| \tag{9.7}$$

It should be noticed that, as defined, $\mathcal{L}_{\text{sparsity}}$ and $\mathcal{L}_{\text{contiguity}}$ correspond to the l_1 regularization and the total variation regularization on the activations of $\hat{\mathbf{x}}$, respectively. In fact, the $\mathcal{L}_{\text{sparsity}}$ term could have been defined as the l_0-norm, since the l_0-optimization has also the property of producing sparse solutions [238, 166]. However, the corresponding l_0-optimization problem is non-convex and, hence, difficult to solve. It is known to be NP-hard. In this regard, $\mathcal{L}_{\text{sparsity}}$ was defined as the l_1-norm, since l_1 is indeed a good differentiable approximation to l_0 [166].

9.3.2.3 Hybrid Fully and Weakly Supervised Regularization

In a completely annotated scenario, i.e., when both expression labels and facial landmarks annotations are available, the regression task of the relevance map $\hat{\mathbf{x}}$ can be performed by combining both fully and weakly supervised regularization schemes. In this case, the facial-parts loss to be minimized is simply defined as the weighted summation of the loss terms defined in Equations 9.4 and 9.5. The underlying idea is to combine the strengths of both proposed regularization schemes while mitigating their potential weaknesses when used individually.

The proposed fully supervised regularization scheme encourages the predicted relevance maps $\hat{\mathrm{x}}$ to be as similar as possible to the target ones x^{target}. In this regard, the resulting relevance maps $\hat{\mathrm{x}}$ will "just" encode the local appearance information around the facial landmarks. Although the facial landmarks, along with the facial components in which they lay on, could represent some of the most relevant facial areas for expression recognition, other important facial clues, such as the expression wrinkles or dimples, may be neglected. As the weakly supervised regularization scheme relies on sparsity and contiguity impositions, the resulting relevance maps will have the potential to capture expressions wrinkles and dimples. However, as the relevance maps are learned with no supervision, the optimization process is more difficult and highly sensitive to the hyperparameters choice (λ and γ).

By combining both regularization schemes, the predicted relevance maps $\hat{\mathrm{x}}$ will encode local appearance information around facial landmarks with the freedom to capture additional sparse and contiguity facial features, such as expression wrinkles and dimples.

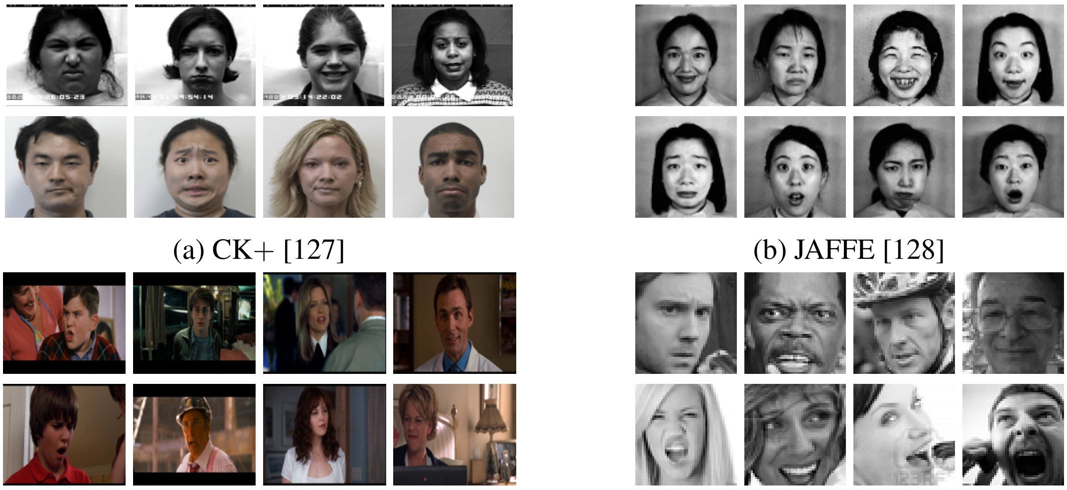
9.4 Experimental Evaluation

The experimental evaluation of the proposed deep neural network was performed using publicly available databases in the FER research field: the Extended Cohn-Kanade (CK+) database [127], the Japanese Female Facial Expressions (JAFFE) database [128], the Static Facial Expressions in the Wild (SFEW) database [46] and the Facial Expression Recognition 2013 (FER-2013) database [75].

While both CK+ and JAFFE datasets contain images acquired under lab-controlled conditions, SFEW 2.0 and FER contain images with spontaneous expressions acquired under

Table 9.1: Summary of the datasets used in the experimental evaluation.

Dataset	Neutral	Anger	Contempt	Disgust	Fear	Happy	Sadness	Surprise	Total
CK+ [127]	327	135	54	177	75	147	84	249	1308
JAFFE [128]	-	30	-	41	8	54	39	41	213
SFEW [46]	228	255	-	75	124	256	234	150	1322
FER-2013 [75]	6198	4953	-	547	5121	8989	6077	4002	35887



(a) CK+ [127]

(b) JAFFE [128]

(c) SFEW 2.0 [46]

(d) FER-2013 [75]

Figure 9.7: Illustrative samples of the datasets used in the experiments. While both CK+ (a) and JAFFE (b) datasets contain images acquired under controlled environments, the SFEW 2.0 (c) and FER-2013 (d) datasets contain images with spontaneous expressions acquired under non-controlled scenarios.

wild (non-controlled) scenarios. Table 9.1 depicts the total number of images of each dataset as well as the class distribution. Some representative samples of each dataset are shown in Figure 9.7.

9.4.1 Implementation Details

As a pre-processing step, the multi-task CNN face detector [251] is used for face detection. The faces are then normalized, cropped, and resized to 120×120 pixels.

The proposed fully supervised regularization scheme as well as the proposed hybrid regularization strategy require the facial landmarks annotation for creating the target relevance maps x^{target}. Although the CK+ dataset contains the annotations of the facial landmarks, these manual annotations are not available on both JAFFE and SFEW. Therefore, a robust facial landmarks detector [33] was applied on both JAFFE and SFEW datasets and, then, the automatically generated facial landmarks were used to build the target relevance maps.

Table 9.2: A detailed description of the architecture of the proposed model. The output shape is described as (#filters, rows, columns).

Layer #	Network module	Layer (type)	Output shape	Connected to
1	$E(\mathrm{x})$	input_1 (InputLayer)	(3, 120, 120)	-
2		conv2d_1 (Conv2D)	(16, 120, 120)	input_1
3		conv2d_2 (Conv2D)	(16, 120, 120)	conv2d_1
4		max_pool2d_1 (MaxPooling2D)	(16, 60, 60)	conv2d_2
5		conv2d_3 (Conv2D)	(32, 60, 60)	max_pool2d_1
6		conv2d_4 (Conv2D)	(32, 60, 60)	conv2d_3
7		max_pool2d_2 (MaxPooling2D)	(32, 30, 30)	conv2d_4
8		conv2d_5 (Conv2D)	(64, 30, 30)	max_pool2d_2
9		conv2d_6 (Conv2D)	(64, 30, 30)	conv2d_5
10		max_pool2d_3 (MaxPooling2D)	(64, 15, 15)	conv2d_6
11		conv2d_7 (Conv2D)	(128, 15, 15)	max_pool2d_3
12		conv2d_8 (Conv2D)	(128, 15, 15)	conv2d_7
13		conv2d_tr_1 (Conv2DTranspose)	(64, 30, 30)	conv2d_8
14		concat_1 (Concatenate)	(128, 30, 30)	[conv2d_tr_1; conv2d_6]
15		conv2d_9 (Conv2D)	(64, 30, 30)	concat_1
16		conv2d_10 (Conv2D)	(64, 30, 30)	conv2d_9
17		conv2d_tr_2 (Conv2DTranspose)	(32, 60, 60)	conv2d_10
18		concat_2 (Concatenate)	(64, 60, 60)	[conv2d_tr_2; conv2d_4]
19		conv2d_11 (Conv2D)	(32, 60, 60)	concat_2
20		conv2d_12(Conv2D)	(32, 60, 60)	conv2d_11
21		conv2d_tr_3 (Conv2DTranspose)	(16, 120, 120)	conv2d_12
22		concat_3 (Concatenate)	(32, 120, 120)	[conv2d_tr_3; conv2d_2]
23		conv2d_13 (Conv2D)	(16, 120, 120)	concat_3
24		conv2d_14 (Conv2D)	(1, 120, 120)	conv2d_13
25	$F(\mathrm{x},\hat{\mathrm{x}})$	conv2d_15 (Conv2D)	(16, 120, 120)	input_1
26		conv2d_16 (Conv2D)	(16, 120, 120)	conv2d_15
27		max_pool2d_4 (MaxPooling2D)	(16, 60, 60)	conv2d_16
28		conv2d_17 (Conv2D)	(32, 60, 60)	max_pool2d_4
29		conv2d_18 (Conv2D)	(32, 60, 60)	conv2d_17
30		max_pool2d_5 (MaxPooling2D)	(32, 30, 30)	conv2d_18
31		conv2d_19 (Conv2D)	(64, 30, 30)	max_pool2d_5
32		conv2d_20 (Conv2D)	(64, 30, 30)	conv2d_19
33		max_pool2d_6 (MaxPooling2D)	(64, 15, 15)	conv2d_20
34		eblock_1 (e-block)	(64, 15, 15)	[max_pool2d_6; conv2d_14]
35	$G(\mathrm{h})$	dense_1 (Dense)	(512)	eblock_1
36		dropout_1 (Dropout)	(512)	dense_1
37		dense_2 (Dense)	(512)	dropout_1
38		dropout_2 (Dropout)	(512)	dense_2
39		dense_3 (Dense)	(8)	dropout_2

All deep models are implemented in Theano [211] and trained with the Adam optimization algorithm using a batch size of 50 samples. We used a learning rate with step decay, in which the initial learning rate was multiplied by 0.99 at each training epoch.

The hyperparameters of the models are optimized by means of grid search and cross-validation on the training set. These parameters include the weights of all loss terms (λ and γ), the learning rate α, the l_2 coefficient, and the number of convolution-convolution-pooling

Table 9.3: Hyperparameters sets.

Hyperparameters	Acronym	Set
Architecture	L_E	$\{3{,}4\}$
	L_F	$\{3{,}4\}$
Leaning rate	α	$\{1\mathrm{e}^{-03},1\mathrm{e}^{-04}\}$
l_2-norm coefficient	-	$\{1\mathrm{e}^{-04},1\mathrm{e}^{-05}\}$
Facial parts loss†	λ	$\{1,5,10,15\}$
Facial parts loss‡	λ	$\{1\mathrm{e}^{-03},1\mathrm{e}^{-04},1\mathrm{e}^{-05},1\mathrm{e}^{-06}\}$
	γ	$\left\{\frac{1\mathrm{e}^{-03}}{\lambda},\frac{1\mathrm{e}^{-04}}{\lambda},\frac{1\mathrm{e}^{-05}}{\lambda},\frac{1\mathrm{e}^{-06}}{\lambda}\right\}$

† fully supervised regularization scheme, ‡ weakly supervised regularization scheme.

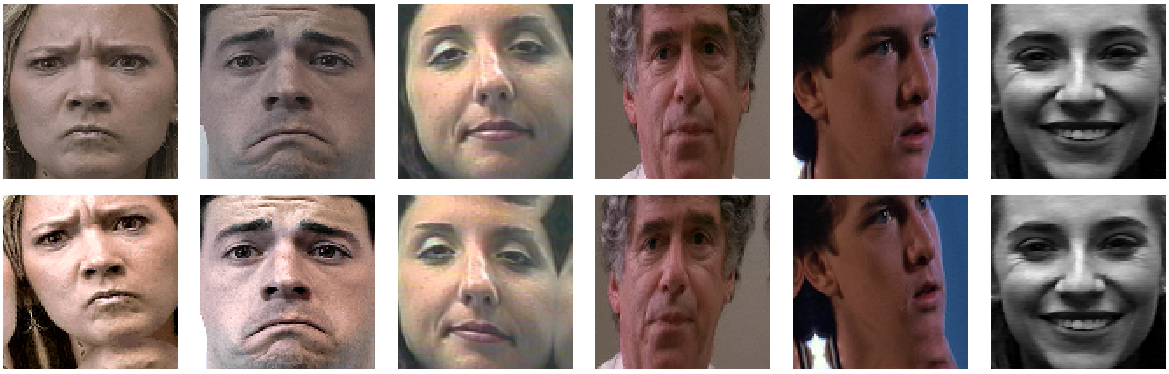


Figure 9.8: Illustration of the implemented data augmentation process: original colour images (top row) along with the corresponding augmented images (bottom row).

blocks of both *facial-parts* and *representation components* (L_E and L_F, respectively). The number of dense layers of the *classification component* was set to 3 in all the experiments. In particular, while the number of neurons of the last dense layer (i.e., the output layer) corresponds to the number of classes, the first two dense layers contain 512 neurons. A detailed description of the architecture of the proposed model is presented in Table 9.2. For a fair comparison, the hyperparameters (i.e., architecture, learning rate and l_2 coefficient) of the CNN trained from scratch as baseline were also optimized. The range of values of the adopted hyperparameters' grid search is presented in Table 9.3.

For an extra regularising effect, the randomized data augmentation scheme based on both geometric and colour transformations, previously introduced in section 6.4.1, is also applied. Regarding the parameters of the data augmentation scheme, the rotation angle θ is randomly sampled from $\{-\pi/18,-\pi/36,0,\pi/36,\pi/18\}$. The skew parameters, k_1 and k_2, are both randomly sampled from $\{-0.1,0,0.1\}$. The scale parameter s is randomly sampled from five different resize factors $\{0.9,0.95,1,1.05,1.1\}$. Finally, the translation parameters t_1 and

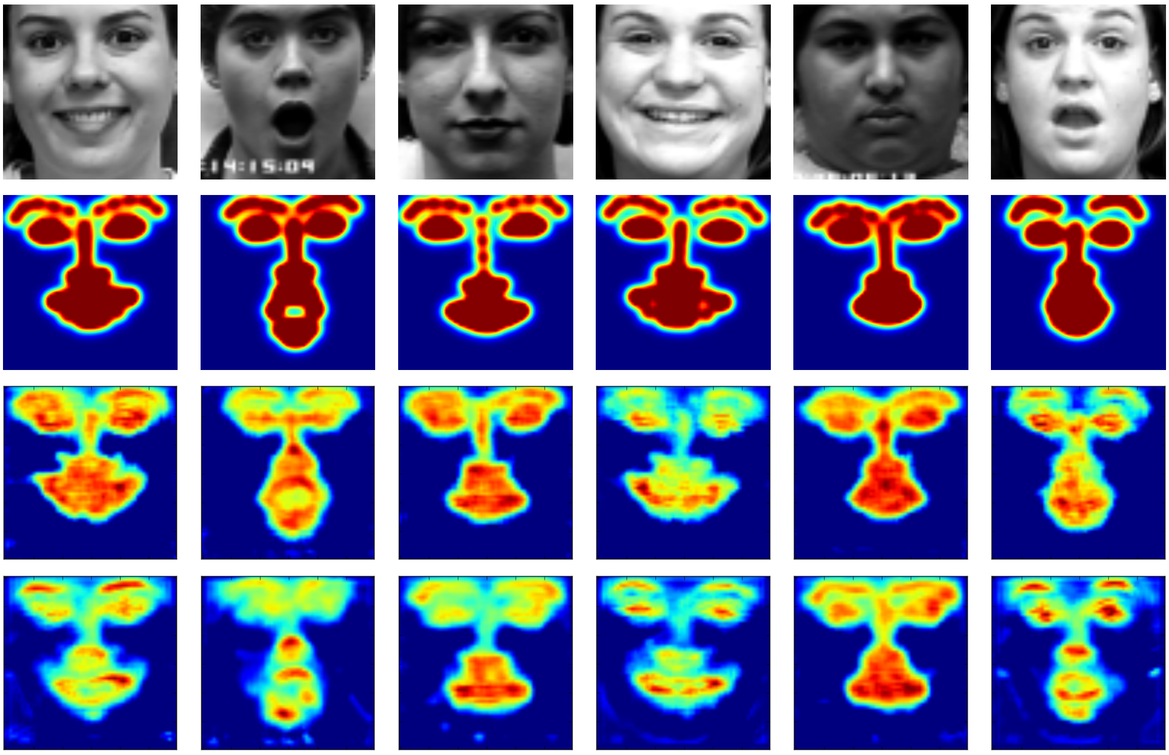


Figure 9.9: Illustrative examples of the predicted relevance maps $\hat{x}$ using the proposed fully supervised regularization scheme and the effect of varying the facial-parts loss $\mathcal{L}_{facial\_parts}$ coefficient: the input images (first row) and the corresponding target relevance maps x^{target} (second row), predicted relevance maps $\hat{x}$ with $\lambda = 10$ (third row), and predicted relevance maps $\hat{x}$ with $\lambda = 5$ (bottom row).

t_2 are randomly sampled integers from the interval $[0,5]$. Figure 9.8 depicts the application of the implemented data augmentation procedure.

9.4.2 Relevance Maps Visualization

In order to demonstrate the effectiveness of the proposed deep model in capturing high-level semantic concepts related to facial expressions, we have performed a visual inspection of the relevance maps $\hat{x}$ that are learned by the *facial-parts component* of our model. Figures 9.9 and 9.10 depict the learned relevance maps $\hat{x}$ for some test samples using the proposed fully supervised and weakly supervised regularization schemes, respectively. As expected, the activations of the predicted relevance maps using both training schemes are strong just in the neighborhood of important facial components. This demonstrates that the relevance maps are suitable to enforce the model to learn highly discriminative representations for FER.

The fully supervised regularization scheme minimizes the mean squared error between the predicted relevance maps $\hat{x}$ and the targets x^{target}, which are created based on the location

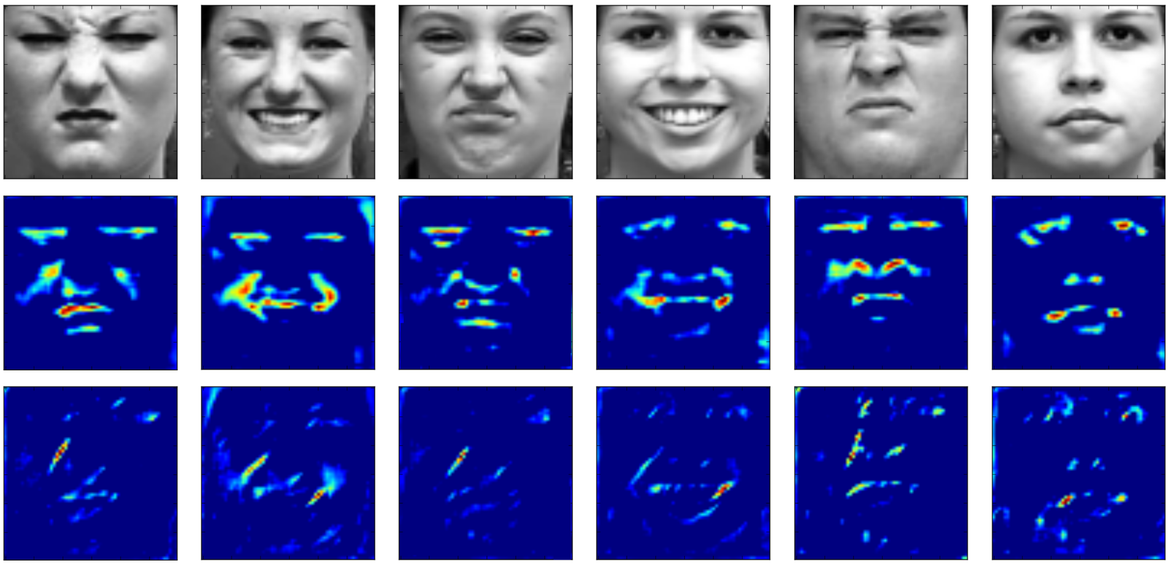


Figure 9.10: Illustrative examples of the predicted relevance maps $\hat{\mathrm{x}}$ using the proposed weakly supervised regularization scheme and the effect of varying the coefficients of $\mathcal{L}_{\text{sparsity}}$ and $\mathcal{L}_{\text{contiguity}}$: the input images (first row) and the corresponding predicted relevance maps $\hat{\mathrm{x}}$ with $\lambda = 1\mathrm{e}^{-04}$ and $\gamma = 1$ (middle row), and predicted relevance maps with $\lambda = 1\mathrm{e}^{-02}$ and $\gamma = 1$ (bottom row).

of the facial landmarks. Therefore, the predicted relevance maps encode the local appearance information around the facial landmarks. The weakly supervised regularization scheme does not rely on the facial landmarks location. Instead, the activations of the predicted relevance maps are regularized to be sparse and spatially localized. Interestingly, the resulting relevance maps are able to capture not only the local information around the facial landmarks but also the local information related to expression wrinkles (see the middle row of Figure 9.10). This clearly demonstrates the importance of the expression wrinkles to the recognition process.

Figures 9.9 and 9.10 also demonstrate the effect of varying the coefficients of the facial-parts loss $\mathcal{L}_{facial\_parts}$. Regarding the fully supervised version of the proposed model, as we decrease the λ coefficient, the predicted relevance maps $\hat{\mathrm{x}}$ are allowed to be more distant from the targets x^{target} (see the bottom row of Figure 9.9). In the weakly supervised setting, as we increase the coefficients of the sparsity and the spatial contiguity terms (λ and γ), the activations of the predicted relevance maps $\hat{\mathrm{x}}$ are forced to be sparser and smoother (i.e., with less transitions), respectively. The middle row of Figure 9.10 illustrates the effect of setting a good parameterization to the coefficients of $\mathcal{L}_{\text{sparsity}}$ and $\mathcal{L}_{\text{contiguity}}$ (i.e., $\lambda = 1\mathrm{e}^{-04}$ and $\gamma = 1$), which results in well defined relevance maps around the facial components (e.g., mouth, eyes, nose and expression wrinkles). The effect of an over-regularization is depicted in the bottom row of Figure 9.10 (i.e., $\lambda = 1\mathrm{e}^{-02}$ and $\gamma = 1$). The resulting relevance maps are then too sparse and not so well defined around the facial components.

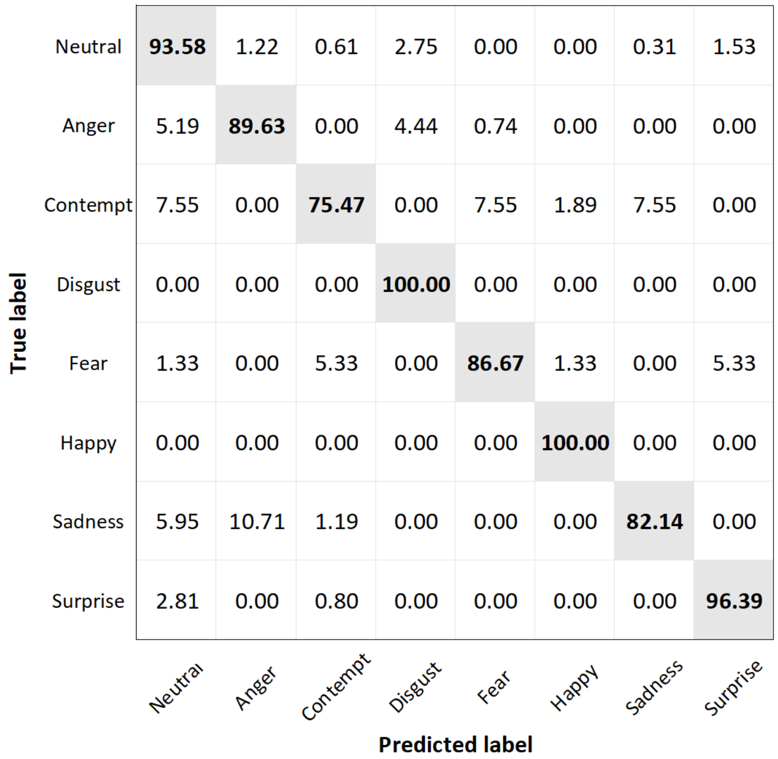
9.4.3 Results on CK+

CK+ consists of 593 videos from 123 subjects acquired in a controlled environment, 327 of them annotated with 8 expression labels (i.e., the 6 universal expressions plus the neutral and contempt ones). Each video starts with a neutral expression and reaches the peak in the last frame. As in other works [122], the first frame and the last three frames of each video were extracted, in order to construct our image-based CK+ dataset. The result is a subset of 1308 images. For model selection and evaluation, a stratified k-fold cross-validation scheme with subject independence was adopted (i.e., $k = 10$). In each split, the training set is further divided, also with subject independence, in 80% for training and 20% for validation.

Experiments on CK+ database are presented in Table 9.4, in which a comparison between the proposed model and state-of-the-art methods, including both traditional and deep learning-based approaches, is performed. The results are presented in terms of average classification accuracy. It is important to note that we just considered state-of-the-art methods that followed the same evaluation protocol (i.e., 1308 images with 8 expressions). To further demonstrate the effectiveness of the proposed method, a CNN trained from scratch with l_2 regularization was considered as baseline. The baseline CNN has the same network architecture as the *representation component* of the proposed model. As shown in Table 9.4, the implemented CNN is a fairly strong baseline, with an overall classification accuracy of 90.48%. We have

Table 9.4: CK+ experimental results. The results are reported in terms of average classification accuracy. The first block of the table presents the results of state-of-the-art methods. The second block depicts the results of all versions of the proposed model and the baseline CNN. Bold number indicates the best method with the highest average classification accuracy.

Method	Average Accuracy (%)
Liu *et al.* (2013) [120]	92.10
Ding *et al.* (2017) [47]	88.70
Ding *et al.* (2017) [47]	89.90
Ng *et al.* (2015) [144]	93.20
CNN from Scratch with Reg (baseline)	90.48
Fully Supervised	92.54
Weakly Supervised ($\mathcal{L}_{sparsity}$)	93.26
Weakly Supervised ($\mathcal{L}_{contiguity}$)	91.70
Weakly Supervised ($\mathcal{L}_{sparsity} + \mathcal{L}_{contiguity}$)	93.37
Hybrid Fully and Weakly Supervised	**93.64**



True label \ Predicted label	Neutral	Anger	Contempt	Disgust	Fear	Happy	Sadness	Surprise
Neutral	**93.58**	1.22	0.61	2.75	0.00	0.00	0.31	1.53
Anger	5.19	**89.63**	0.00	4.44	0.74	0.00	0.00	0.00
Contempt	7.55	0.00	**75.47**	0.00	7.55	1.89	7.55	0.00
Disgust	0.00	0.00	0.00	**100.00**	0.00	0.00	0.00	0.00
Fear	1.33	0.00	5.33	0.00	**86.67**	1.33	0.00	5.33
Happy	0.00	0.00	0.00	0.00	0.00	**100.00**	0.00	0.00
Sadness	5.95	10.71	1.19	0.00	0.00	0.00	**82.14**	0.00
Surprise	2.81	0.00	0.80	0.00	0.00	0.00	0.00	**96.39**

Figure 9.11: Confusion Matrix of CK+ dataset. Gray cells represent the true positives.

also implemented several hand-crafted FER methodologies, either geometric- or appearance-based, for comparison. As their performance is significantly below than the one achieved by any deep learning-based approach, all details about the implemented hand-crafted FER methods were referred to Appendix D.

Table 9.4 also depicts the performance of both versions of the proposed model (i.e., the fully and weakly supervised models) as well as their hybrid formulation (fully + weakly supervised). In order to assess the impact of the loss terms ($\mathcal{L}_{\text{sparsity}}$ and $\mathcal{L}_{\text{contiguity}}$) in the weakly supervised model, we report the results using each loss term independently and combined. Regardless of the training strategy, the proposed model always outperforms the baseline CNN. In particular, the proposed hybrid model, which combines both fully supervised and weakly supervised regularization schemes, provides the best classification accuracy (93.64%), outperforming all the state-of-the-art methods.

One of the most interesting observations is the superior performance of the weakly supervised model when compared with the fully supervised model, despite the fact the weakly supervised model does not require the availability of the facial landmarks annotations. These results can be explained by the capability of the weakly supervised model in capturing local information around the expression wrinkles (see the middle row of Figure 9.10).

Another interesting observation, as also reported in Table 9.4, is that the proposed hybrid model, which combines the ideas of both regularization schemes, yields a slight overall improvement in the classification accuracy.

The confusion matrix, as illustrated in Figure 9.11, shows the consistent performance of the proposed hybrid method. Both happy and disgust expressions are perfectly classified, while contempt is the most difficult to classify. This happens because the contempt expression is the class with the least number of training images and is typically performed in a subtle way.

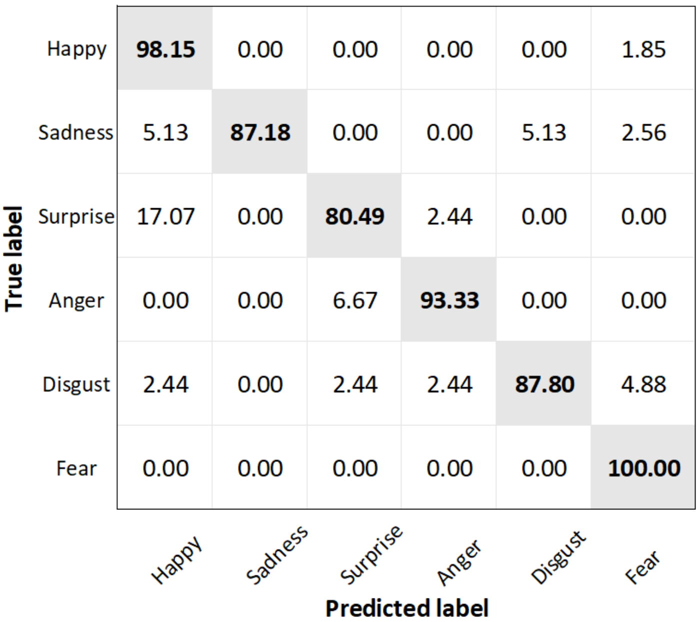
9.4.4 Results on JAFFE

The database contains 213 images of 6 facial expressions posed by 10 Japanese female models. Illustrative examples of the JAFFE dataset are shown in Figure 9.7b. For model selection and evaluation, a stratified 3-fold cross-validation with subject independence was performed. In each split, the training set is further divided, also with subject independence, in 80% for training and 20% for validation.

Table 9.5 compares the performance of the proposed approach with the baseline CNN and state-of-the-art methods. As observed from Table 9.5, the proposed fully supervised and weakly supervised models along with their hybrid version clearly outperform the implemented baseline CNN, with classification accuracies of 84.88%, 87.84%, 89.01% and 79.06%, respectively. In addition, our method provides substantial improvements over the previous best state-of-the-art performance achieved by Happy *et al.* [78], with a gain of 3.95%. These results clearly demonstrate the potential of the proposed approach to deal with the problem

Table 9.5: JAFFE experimental results. The results are reported in terms of average classification accuracy. The first block of the table presents the results of state-of-the-art methods. The second block depicts the results of all versions of the proposed model and the baseline CNN. Bold number indicates the best method with the highest average classification accuracy.

Method	Average Accuracy (%)
Shan *et al.* (2009) [180]	81.00
Lopes *et al.* (2017) [126]	84.48
Happy *et al.* (2015) [78]	85.06
CNN from Scratch with Reg (baseline)	79.06
Fully Supervised	84.88
Weakly Supervised ($\mathcal{L}_{sparsity}$)	87.21
Weakly Supervised ($\mathcal{L}_{contiguity}$)	80.81
Weakly Supervised ($\mathcal{L}_{sparsity} + \mathcal{L}_{contiguity}$)	87.84
Hybrid Fully and Weakly Supervised	**89.01**



True label \ Predicted label	Happy	Sadness	Surprise	Anger	Disgust	Fear
Happy	**98.15**	0.00	0.00	0.00	0.00	1.85
Sadness	5.13	**87.18**	0.00	0.00	5.13	2.56
Surprise	17.07	0.00	**80.49**	2.44	0.00	0.00
Anger	0.00	0.00	6.67	**93.33**	0.00	0.00
Disgust	2.44	0.00	2.44	2.44	**87.80**	4.88
Fear	0.00	0.00	0.00	0.00	0.00	**100.00**

Figure 9.12: Confusion Matrix of JAFFE dataset. Gray cells represent the true positives.

of training high-capacity classifiers in small datasets (e.g., JAFFE database is composed of only 213 images).

Figure 9.12 shows the confusion matrix obtained for the best model on JAFFE, which is the proposed hybrid model. As it is possible to observe, the fear expression is perfectly classified. The proposed model performed worst for the surprise expression as it tends to be misclassified as happy.

9.4.5 Results on SFEW

Different from both CK+ and JAFFE datasets, SFEW is targeted for unconstrained FER. It is the first database that depicts real-world or simulated real-world conditions for expression recognition. The images are all extracted from movies (see Figure 9.7c), and labeled with seven expressions. Therefore, there is a wide range of poses, viewing angles, occlusions, illumination conditions and, hence, the recognition is much more challenging. As SFEW was created as part of the Emotion Recognition in the Wild (EmotiW) 2015 Grand Challenge [1], it has a strict evaluation protocol with predefined training, validation, and test sets. In particular, the training set comprises a total of 891 images. Since we do not have access to the test set labels, the results are reported on the validation data that contains 431 images.

As SFEW is clearly one of the most challenging FER datasets, the top state-of-the-art methods on SFEW usually use other databases as additional training data. Typically, the current state-of-the-art models are pre-trained on FER-2013 before being fine-tuned to the

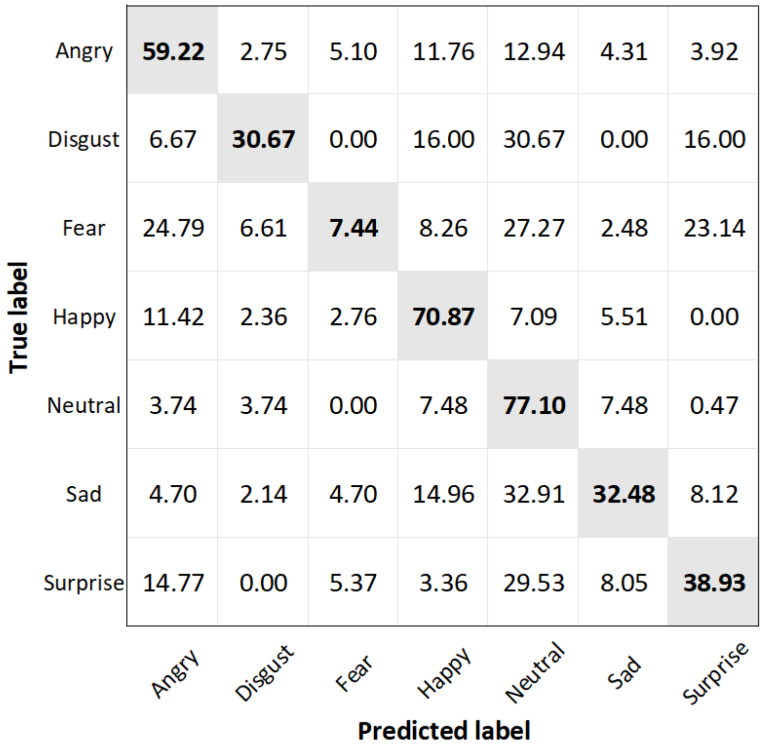
Table 9.6: SFEW experimental results. The results are reported in terms of average classification accuracy. The first block of the table presents the results of state-of-the-art methods that do not use transfer learning. The second block of the table presents the results of state-of-the-art methods that use FER-2013 for pre-training the models. The third block depicts the results of all versions of the proposed model and the baseline CNN. Bold number indicates the best method with the highest average classification accuracy.

Method	Average Accuracy (%)	Transfer Learning
Liu *et al.* (2013) [120]	26.14	None
Liu *et al.* (2014) [122]	31.73	
Levi *et al.* (2015) [113]	41.92	
Mollahosseini *et al.* (2016) [139]	47.70	
Ng *et al.* (2015) [144]	48.50	FER-2013
Yu *et al.* (2015) [247]	**52.29**	
Yu *et al.* (2015) [247] without ensemble †	44.37	
CNN from Scratch with Reg (baseline)	42.07	FER-2013
Fully Supervised	47.56	
Weakly Supervised ($\mathcal{L}_{sparsity}$)	48.72	
Weakly Supervised ($\mathcal{L}_{contiguity}$)	47.56	
Weakly Supervised ($\mathcal{L}_{sparsity} + \mathcal{L}_{contiguity}$)	47.80	
Hybrid Fully and Weakly Supervised	50.12	

† Implemented version of Yu *et al.* [247] method without ensemble.

SFEW dataset. The FER-2013 dataset comprises a total of 35887 grayscale images, labeled with seven facial expressions. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image (see Figure 9.7d). In this regard, for a fair comparison, the proposed model as well as the implemented baseline CNN are first pre-trained on the FER-2013 dataset and, then, fine-tuned to the target dataset (i.e., the SFEW). The fine-tunning process ends when the validation loss stops decreasing (~ 25 epochs).

The experimental results obtained on SFEW are presented in Table 9.6, in which the state-of-the-art methods are grouped into those that do not perform any kind of transfer learning and those that use the FER-2013 dataset for pre-training the models. Once again, the proposed network clearly outperforms the implemented baseline CNN, with an overall accuracy of 50.12% against 42.07%. Moreover, the proposed method achieves better recognition rates than all the other state-of-the-art methods with the exception of the method proposed by Yu *et al.* [247]. However, we argue that this could not be a fair comparison as the method proposed in [247] uses an ensemble of multiple networks to boost their performance. In order to mitigate the gains of their ensemble strategy, a version of the Yu *et al.* [247] method without ensemble was implemented. As reported in Table 9.6, our method clearly outperforms the method of Yu *et al.* [247] without ensemble (i.e, 50.12% against 44.37%).



True label \ Predicted label	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
Angry	**59.22**	2.75	5.10	11.76	12.94	4.31	3.92
Disgust	6.67	**30.67**	0.00	16.00	30.67	0.00	16.00
Fear	24.79	6.61	**7.44**	8.26	27.27	2.48	23.14
Happy	11.42	2.36	2.76	**70.87**	7.09	5.51	0.00
Neutral	3.74	3.74	0.00	7.48	**77.10**	7.48	0.47
Sad	4.70	2.14	4.70	14.96	32.91	**32.48**	8.12
Surprise	14.77	0.00	5.37	3.36	29.53	8.05	**38.93**

Figure 9.13: Confusion Matrix of SFEW dataset. Gray cells represent the true positives.

Figure 9.13 shows the confusion matrix obtained for the proposed model with the best performance on SFEW (i.e., the hybrid fully and weakly supervised model). The recognition accuracy for fear is much lower than other expressions. This is also observed in other works [144].

9.5 Summary

This chapter addresses the topic of facial expression recognition on static images due to its potential application in a complete SLR system. In particular, we propose a novel end-to-end deep neural network architecture along with a well-designed loss function that jointly learns the most relevant facial parts along with the expression recognition. The result is a model that is able to learn expression-specific features.

The proposed neural network is composed by three main components: (i) the *facial-parts component*, (ii) the *representation component* and (iii) the *classification component*. The *facial-parts component* aims to regress a relevance map, representing the most important facial regions for the expression recognition. The relevance map is then used in the *representation component* in order to increase the discriminative ability of the learned features.

Then, the *classification component* is trained on these highly discriminative representations for FER.

Experimental results on three well-known facial expression databases CK+, JAFFE, and SFEW demonstrate the potential of the proposed model in both lab-controlled and wild scenarios. The proposed model provides quite promising results, outperforming in most datasets the current state-of-the-art methods.

Chapter 10

Conclusions and Future Work

This final chapter aims at providing both a summary of the main conclusions related to the scientific contributions presented throughout this thesis, as well as a setup for future lines of research work.

10.1 Conclusions

This thesis addressed the topic of SLR. In this context, several fundamental problems were tackled, whose main contributions and achievements can be summarized as follows:

- **Multimodal SLR**: To fully exploit the complementary properties of the currently available data modalities of the signs, such as RGB, depth, and Leap Motion data, several multimodal learning strategies were investigated. These multimodal techniques, mainly based on deep neural networks, include feature-level and decision-level fusion techniques. In addition, a comparison between single-modality and multimodal learning techniques was conducted, in order to attest the potential of multimodal learning in the overall sign recognition performance.

  Experimental results suggest that, in a single-modality scenario, both RGB and depth modalities are more discriminative than Leap Motion data. Even though, it is worth to mention that in contrast to RGB and depth data, in which hand gestures must be segmented for the subsequent tasks, Leap Motion data does not require any kind of preprocessing. Nonetheless, the most interesting observation is that multimodal fusion often promotes an overall improvement in the sign recognition accuracy, clearly demonstrating the complementarity between the three modalities.

  In this context, our main contribution is a novel end-to-end feature-level deep neural network that jointly learns both modality-specific and modality-shared features. To accomplish this purpose, the proposed model comprises private streams that are specific to each modality and shared streams between modalities. Furthermore, the loss function is defined in such a manner that encourages independence between these private and shared representations. A classifier is then trained on top of these private and shared representations to enhance the discriminative capability of the model. By imposing such regularization constraints in the learning process, the proposed model outperformed the state-of-the-art multimodal approaches.

- **Signer-independent SLR**: One of the major challenges in the SLR field is related to the large inter-signer variations that exist in the manual signing process of sign languages. We addressed the signer-independent SLR problem as a domain adaptation task, in which the goal is to reduce the distribution difference between different signers (domains). For this purpose, we propose a deep neural network, along with an adversarial learning objective, for simultaneously training an encoder and a sign-classifier over the target sign variables, while preventing the latent representations of the encoder from being predictive of the signer identities. In the course of this adversarial training procedure, the learned latent representations are encouraged to be both signer-invariant and highly discriminative of the signs. Furthermore, we introduce an additional constraint to the adversarial training objective that further discourages the learned representations of retaining any signer-specific information, by explicitly promoting similarity in the latent distributions of different signers. Experimental results demonstrate the robustness of the proposed model to unseen test signers in several SLR databases. The proposed adversarial training objective was also successfully extended to another application, where it can be beneficial to learn feature representations invariant to some specific domain or aspect.

  Although the proposed adversarial SLR model represents a major step forward towards the development of a truly signer-independent model, there is always an inherent training instability regarding any adversarial framework. In regard, we further addressed the signer-independent problem by exploring a generative model, i.e., a CVAE. Besides the better training stability when compared with adversarial training, the underlying key idea of using a CVAE-based model relies on its ability to provide a latent representation space very naturally, at the bottleneck of the encoder-decoder, where meaningful constraints can be added in order to promote the desired signer-invariance property. Specifically, the proposed model is composed by a CVAE and a classifier. The purpose of the CVAE module is to learn latent representations of the input data, whose conditional posterior distribution, given the image and its sign label, is independent of the signer identity. During the learning stage, the CVAE can be seen as a teacher model for the classifier, since the conditional posterior distribution over latent representations is used to regularize the latent representations of the classifier. These signer-invariant hidden representations are then used for a robust signer-independent SLR recognition. The proposed CVAE-based model provides quite promising results, outperforming the implemented baseline methods, the state-of-the-art SLR and domain adaptation methods, and the proposed adversarial signer-independent SLR model. Therefore, it constitutes another step forward towards the development of robust signer-independent SLR models.

- **Facial expressiveness analysis**: Given the recognized importance of facial expressions in sign language communication, we develop fundamental research on FER. Based on the strong support from physiology and psychology that FEs are the result of the motions of facial muscles [58, 39], a novel end-to-end deep neural network along with a well-designed loss function for FER were proposed.

  In particular, the proposed model consists of three main components, namely a facial-parts component, a representation component, and a classification component. The main goal of the facial-parts component is to regress a relevance map, representing the most important facial regions for FER. The loss function was defined in order to encourage sparsity and spatial contiguity on the activations of the relevance map. This definition was supported by the physiological fact that just small and disjoint facial regions should be relevant for recognition [58]. The relevance map is then used in the representation component in order to increase the discriminative ability of the learned features. Finally, the classification component is trained on these highly discriminative representations for FER.

  Experimental results on several SLR databases demonstrate the potential of the proposed model in both lab-controlled and wild scenarios. The proposed model provided quite promising results, outperforming in most datasets the current state-of-the-art FER methods.

- **Lack of multimodal LGP databases**: In order to overcome some of the major flaws of the currently available SLR databases, a novel multimodal LGP database was acquired and annotated. It comprises two major components: (i) an LGP dataset, and (ii) a duo-interaction dataset, between deaf and/or hearing people. Therefore, the proposed database can be used for different purposes, such as SLR tasks or emotion/expressiveness recognition from body language.

It is worth to mention that to resemble a real environment scenario, all gestures and sentences were performed in a free and natural expression environment, without any recording and clothing restrictions.

To the best of our knowledge, the proposed database is the first multimodal database that comprises sign language videos along with videos depicting the duo-interaction between deaf and/or hearing people. With such expressiveness richness, it is expected that this database may open new research paths in SLR. The proposed database, along with the annotations, are already made publicly available for benchmark purposes.

10.2 Future Work

SLR has been an on-going research field mainly driven by its potential application in supporting the integration of deaf people into the hearing society. As referred above, the main contributions of this thesis provide solutions for several fundamental SLR problems, with proven performance either regarding multimodal SLR, signer-independent SLR, or facial expressiveness analysis. These contributions have great potential of being integrated into practical SLR systems as well as to represent the grounds for further developments and future lines of work in the field.

It is worth mentioning that for the sake of conciseness and objectivity, most of the proposed models were formulated and developed for the static setting. Off course, a real-world SLR has to deal with continuous SLR. Therefore, as the first line of future work, we intend to generalize all the proposed models for the continuous scenario. In addition, we would like to integrate all the ideas behind all the proposed models into a unified SLR framework.

Although facial expressions represent an important component in sign languages, they are still left out by most of the researchers. This may happen because of several factors. A major problem is related to the scarceness of SLR databases describing the non-manual component of the signs. For instance, during the acquisition of the presented database, we observed that it is extremely difficult to capture the facial expressiveness involved in sign languages naturally. There were several cases in which the signers did not express the non-manual component of the signs. Another problem is the lack of linguistic studies establishing the correlation between facial expressions and manual signs. For the LGP, the work of Gonçalves *et al.* [73] is one of the few exceptions. However, it is still a limited study, since it just comprises morphological grammatical facial expressions. In order to overcome these limitations, in our opinion, an SLR system should integrate the analysis of facial expressions, without relying on ground-truth labels about the facial expressions. Accordingly, as future work, we would like to further explore the main ideas behind the representation component of the proposed FER model. In particular, we intend to investigate the possibility of driving an SLR model towards the most relevant facial regions by just using the sign labels and, thereby, combine both manual and non-manual features.

Last but not least, as future work, we would like to address the problem of the large vocabulary size of sign languages, in particular, the problem of unseen sign words that naturally arise in real-world SLR applications. For that purpose, we would like to use some of the ideas of the domain adaptation neural machine translation models, from the natural language processing field [83]. In the SLR research field, it is possible to find different datasets of several sign languages, with a variable degree of annotation, that cover different sign word sets (i.e., different sign word domains). To mitigate the problem of unseen sign words, it could be interesting to perform lexicon induction by exploring information between different domains (datasets).

References

[1] (2015). *The Third Emotion Recognition in the Wild Challenge (EmotiW 2015).*

[2] (Accessed on 23/08/2019). Associação de surdos do porto. algumas definições úteis sobre a surdez. http://www.asurdosporto.org.pt/artigo.asp?idartigo=77.

[3] (Accessed on 23/08/2019). Corpus ngt. http://www.ru.nl/corpusngt/.

[4] (Accessed on 23/08/2019a). The history of sign language. http://www.deafwebsites.com/sign-language/history-sign-language.html.

[5] (Accessed on 23/08/2019b). History of sign language - deaf history - start asl. https://www.start-american-sign-language.com/history-of-sign-language_html.

[6] (Accessed on 23/08/2019). Omg emotion challenge. https://github.com/knowledgetechnologyuhh/OMGEmotionChallenge.

[7] (Accessed on 23/08/2019). RVL-SLLL American Sign Language Database. https://engineering.purdue.edu/RVL/Database/ASL/asl-database-front.htm.

[8] (Accessed on 23/08/2019). RWTH-PHOENIX-Weather. http://www-i6.informatik.rwth-aachen.de/~forster/database-rwth-phoenix.php.

[9] (Accessed on 23/08/2019). Textblob: Simplified text processing. "http://textblob.readthedocs.io/en/dev/".

[10] (Accessed on 23/08/2019). World federation of the deaf - world federation of the deaf, wfd, human rights, deaf, deaf people. http://wfdeaf.org/.

[11] (Accessed on 23/08/2019). Zicheng liu. https://documents.uow.edu.au/~wanqing/#Datasets.

[12] Adithya, V., Vinod, P., and Gopalakrishnan, U. (2013). Artificial neural network based method for indian sign language recognition. In *Information Communication Technologies (ICT), 2013 IEEE Conference on*, pages 1080–1085.

[13] Al-Ahdal, M. and Tahir, N. (2012). Review in sign language recognition systems. In *Computers Informatics (ISCI), 2012 IEEE Symposium on*, pages 52–57.

[14] Almaev, T. R. and Valstar, M. F. (2013). Local gabor binary patterns from three orthogonal planes for automatic facial expression recognition. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 356–361.

[15] Almeida, I. R. (2014). Exploring challenges in avatar-based translation from european portuguese to portuguese sign language. Master's thesis, Técnico Lisboa.

[16] Aran, O., Ari, I., Benoit, A., and et al. (2006). Signtutor: an interactive sign language tutoring tool. In *in Proceedings of the SIMILAR NoE Summer Workshop on Multimodal Interfaces*.

[17] Assaleh, K., Shanableh, T., Fanaswala, M., Amin, F., and Bajaj, H. (2010). Continuous arabic sign language recognition in user dependent mode. In *Journal of Intelligent Learning Systems and Applications*, volume 2, pages 19–27.

[18] Athitsos, V., Neidle, C., Sclaroff, S., Nash, J., Stefan, R., Thangali, A., Wang, H., and Yuan, Q. (2010). Large lexicon project: American sign language video corpus and sign language indexing/retrieval algorithms.

[19] Awad, G., Han, J., and Sutherland, A. (2006). A unified system for segmentation and tracking of face and hands in sign language recognition. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 239–242.

[20] Baltazar, A. B. (2010). *Dicionário de Língua Gestual Portuguesa*. Porto Editora.

[21] Barros, P., Churamani, N., Lakomkin, E., Siqueira, H., Sutherland, A., and Wermter, S. (2018). The omg-emotion behavior dataset. *CoRR*, abs/1608.06019.

[22] Barros, P. and Wermter, S. (2016). Developing crossmodal expression recognition based on a deep neural model. *Adapt Behav - Animals, Animats, Software Agents, Robots, Adaptive Systems*, 24(5):373–396.

[23] Bengio, Y. (2009). Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127.

[24] Berretti, S., Bimbo, A. D., Pala, P., Amor, B. B., and Daoudi, M. (2010). A set of selected sift features for 3d facial expression recognition. In *2010 20th International Conference on Pattern Recognition*, pages 4125–4128.

[25] Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

[26] Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA.

[27] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

[28] Bouchard, G. and Triggs, B. (2004). The tradeoff between generative and discriminative classifiers. *IASC International Symposium on Computational Statistics (COMPSTAT)*.

[29] Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., and Krishnan, D. (2016a). Unsupervised pixel-level domain adaptation with generative adversarial networks. *CoRR*, abs/1612.05424.

[30] Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., and Erhan, D. (2016b). Domain separation networks. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 343–351.

[31] Bowden, R., Windridge, D., Kadir, T., Zisserman, A., and Brady, M. (2004). *Computer Vision - ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I*, chapter A Linguistic Feature Vector for the Visual Interpretation of Sign Language, pages 390–401. Springer Berlin Heidelberg, Berlin, Heidelberg.

[32] Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Józefowicz, R., and Bengio, S. (2015). Generating sentences from a continuous space. *CoRR*, abs/1511.06349.

[33] Bulat, A. and Tzimiropoulos, G. (2017). Super-fan: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with gans. *CoRR*, abs/1712.02765.

[34] Burkert, P., Trier, F., Afzal, M. Z., Dengel, A., and Liwicki, M. (2015). Dexpression: Deep convolutional neural network for expression recognition. *CoRR*, abs/1509.05371.

[35] Caplier, A., Stillittano, S., Aran, O., Akarun, L., Bailly, G., Beautemps, D., Aboutabit, N., and Burger, T. (2007). Image and video for hearing impaired people. *J. Image Video Process.*, 2007(5):2:1–2:14.

[36] Chang, C.-C., Chen, J.-J., Tai, W.-K., and Han, C.-C. (2006). New approach for static gesture. *Journal of information science and engineering*, 22:1047–1057.

[37] Connie, T., Al-Shabi, M., Cheah, W. P., and Goh, M. (2017). *Facial Expression Recognition Using a Hybrid CNN–SIFT Aggregator*, pages 139–149. Springer International Publishing, Cham.

[38] Cooper, H. and Bowden, R. (2007). *Human–Computer Interaction: IEEE International Workshop, HCI 2007 Rio de Janeiro, Brazil, October 20, 2007 Proceedings*, chapter Large Lexicon Detection of Sign Language, pages 88–97. Springer Berlin Heidelberg, Berlin, Heidelberg.

[39] Corneanu, C. A., Simón, M. O., Cohn, J. F., and Guerrero, S. E. (2016). Survey on rgb, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1548–1568.

[40] Csurka, G. (2017). Domain adaptation for visual applications: A comprehensive survey. *CoRR*, abs/1702.05374.

[41] Căleanu, C. D. (2013). Face expression recognition: A brief overview of the last decade. In *2013 IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pages 157–161.

[42] Dahmani, D. and Larabi, S. (2014). User-independent system for sign language finger spelling recognition. *Journal of Visual Communication and Image Representation*, 25(5):1240 – 1250.

[43] Darwin, C. (1982). *The expression of the emotions in man and animals*. London: John Murray.

[44] Das, S. P., Talukdar, A. K., and Sarma, K. K. (2015). Sign language recognition using facial expression. *Procedia Computer Science*, 58:210 – 216. Second International Symposium on Computer Vision and the Internet (VisionNet'15).

[45] Dhall, A., Asthana, A., Goecke, R., and Gedeon, T. (2011a). Emotion recognition using phog and lpq features. In *Face and Gesture 2011*, pages 878–883.

[46] Dhall, A., Goecke, R., Lucey, S., and Gedeon, T. (2011b). Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 2106–2112.

[47] Ding, H., Zhou, S. K., and Chellappa, R. (2017). Facenet2expnet: Regularizing a deep face recognition net for expression recognition. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pages 118–126.

[48] Dominio, F., Donadeo, M., and Zanuttigh, P. (2014). Combining multiple depth-based descriptors for hand gesture recognition. *Pattern Recognition Letters*, 50:101 – 111. Depth Image Analysis.

[49] Dreuw, P. (Accessed on 23/08/2019). Automatic Sign Language Recognition (ASLR). http://www-i6.informatik.rwth-aachen.de/aslr/index.php.

[50] Dreuw, P., Deselaers, T., Rybach, D., Keysers, D., and Ney, H. (2006). Tracking using dynamic programming for appearance-based sign language recognition. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 293–298.

[51] Dreuw, P., Forster, J., and Ney, H. (2012). *Trends and Topics in Computer Vision: ECCV 2010 Workshops, Heraklion, Crete, Greece, September 10-11, 2010, Revised Selected Papers, Part I*, chapter Tracking Benchmark Databases for Video-Based Sign Language Recognition, pages 286–297. Springer Berlin Heidelberg, Berlin, Heidelberg.

[52] Dreuw, P., Neidle, C., Athitsos, V., Sclaroff, S., and Ney, H. (2008). Benchmark databases for video-based automatic sign language recognition. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*.

[53] Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J., and Vapnik, V. (1997). Support vector regression machines. In Mozer, M. C., Jordan, M. I., and Petsche, T., editors, *Advances in Neural Information Processing Systems 9*, pages 155–161. MIT Press.

[54] Duan, K.-B. and Keerthi, S. S. (2005). Which is the best multiclass svm method? an empirical study. In Oza, N. C., Polikar, R., Kittler, J., and Roli, F., editors, *Multiple Classifier Systems*, pages 278–285, Berlin, Heidelberg. Springer Berlin Heidelberg.

[55] Duchenne, G.-B. (1990). *The mechanism of human facial expression*. Cambridge university press.

[56] Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159.

[57] Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters.

[58] E., F. and P., E. (1978). Facial action coding system: A technique for the measurement of facial movement. In *Palo Alto, CA: Consulting Psychologists Press.*

[59] Ebrahim Al-Ahdal, M. and Nooritawati, M. T. (2012). Review in sign language recognition systems. In *2012 IEEE Symposium on Computers Informatics (ISCI)*, pages 52–57.

[60] Ekman, P. (1992). An argument for basic emotions. cognition & emotion. *CoRR*, 6(3-4):169–200.

[61] Elliott, E. A. and Jacobs, A. M. (2013). Facial expressions, emotions, and sign languages. *American journal of physical anthropology*, 4(115).

[62] Erdem, U. and Sclaroff, S. (2002). Automatic detection of relevant head gestures in american sign language communication. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 460–463 vol.1.

[63] Fang, G., Gao, W., and Zhao, D. (2007). Large-vocabulary continuous sign language recognition based on transition-movement models. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 37(1):1–9.

[64] Fernandes, K., Cardoso, J. S., and Fernandes, J. (2017). Transfer learning with partial observability applied to cervical cancer screening. In Alexandre, L. A., Salvador Sánchez, J., and Rodrigues, J. M. F., editors, *Pattern Recognition and Image Analysis*, pages 243–250, Cham. Springer International Publishing.

[65] Fernando, M. and Wijjayanayake, J. I. (2015). Novel approach to use hu moments with image processing techniques for real time sign language communication. *International Journal of Image Processing (IJIP)*, 9(6):335 – 345.

[66] Feutry, C., Piantanida, P., Bengio, Y., and Duhamel, P. (2018). Learning anonymized representations with adversarial neural networks. *arXiv preprint arXiv:1802.09386.*

[67] Forster, J., Schmidt, C., Hoyoux, T., Koller, O., Zelle, U., Piater, J., and Ney, H. (2012). Rwth-phoenix-weather: A large vocabulary sign language recognition and translation corpus. In Chair), N. C. C., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

[68] Frijda, N. H., Tcherkassof, A., and Mandler, G. (1997). *Facial expressions as modes of action readiness*, page 78–102. Studies in Emotion and Social Interaction. Cambridge University Press.

[69] Galbally, J., Ortiz-Lopez, J., Fierrez, J., and Ortega-Garcia, J. (2012). Iris liveness detection based on quality related features. In *2012 5th IAPR International Conference on Biometrics (ICB)*, pages 271–276.

[70] Ganin, Y. and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France. PMLR.

[71] Geng, Y., Zhang, G., Li, W., Gu, Y., Liang, R.-Z., Liang, G., Wang, J., Wu, Y., Patil, N., and Wang, J.-Y. (2017). A novel image tag completion method based on convolutional neural transformation. In Lintas, A., Rovetta, S., Verschure, P. F., and Villa, A. E., editors, *Artificial Neural Networks and Machine Learning – ICANN 2017*, pages 539–546, Cham. Springer International Publishing.

[72] Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In Gordon, G., Dunson, D., and Dudík, M., editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA. PMLR.

[73] Gonçalves, E. and Raposo, M. J. C. (2014). Facial expressions in grammatical morphology of portuguese sign language – expressions of degrees of augmentative and diminutive size in lgp. *Cadernos de Saúde*, 6:78–83.

[74] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. The MIT Press.

[75] Goodfellow, I., Erhan, D., Carrier, P.-L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.-H., Zhou, Y., Ramaiah, C., Feng, F., Li, R., Wang, X., Athanasakis, D., Shawe-Taylor, J., Milakov, M., Park, J., Ionescu, R., Popescu, M., Grozea, C., Bergstra, J., Xie, J., Romaszko, L., Xu, B., Chuang, Z., and Bengio, Y. (2013). Challenges in representation learning: A report on three machine learning contests.

[76] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

[77] Hamid, A.-T. Z., Wirza, R. R., Iqbal, S. M., and Suhaiza, S. P. (2014). Skin segmentation using yuv and rgb color spaces. *Journal of Information Processing Systems*, 10(2):283.

[78] Happy, S. L. and Routray, A. (2015). Automatic facial expression recognition using features of salient facial patches. *IEEE Transactions on Affective Computing*, 6(1):1–12.

[79] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

[80] Holden, E.-J., Lee, G., and Owens, R. (2005). Automatic recognition of colloquial australian sign language. In *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*, volume 2, pages 183–188.

[81] Holden, E.-J. and Owens, R. (2001). *Multi-Image Analysis: 10th International Workshop on Theoretical Foundations of Computer Vision Dagstuhl Castle, Germany, March 12–17, 2000 Revised Papers*, chapter Visual Sign Language Recognition, pages 270–287. Springer Berlin Heidelberg, Berlin, Heidelberg.

[82] Hu, J., Lu, J., Tan, Y., and Zhou, J. (2016). Deep transfer metric learning. *IEEE Transactions on Image Processing*, 25(12):5576–5588.

[83] Hu, J., Xia, M., Neubig, G., and Carbonell, J. G. (2019). Domain adaptation of neural machine translation by lexicon induction. *CoRR*, abs/1906.00376.

[84] Huang, C., Loy, C. C., and Tang, X. (2016). Local similarity-aware deep feature embedding. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 1262–1270.

[85] Huang, J., Zhou, W., Li, H., and Li, W. (2015). Sign language recognition using 3d convolutional neural networks. In *2015 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

[86] Ifrim, G. and Weikum, G. (2006). Transductive learning for text classification using explicit knowledge models. In *European Conf on Principles of Data Mining and Knowledge Discovery*, pages 223–234.

[87] ISO/IEC JTC1 SC37 (2017). Information Technology - Biometrics - Presentation attack detection Part 3: Testing and Reporting. *ISO Int. Organization for Standardization*.

[88] Jain, A. K., Flynn, P., and Ross, A. A. (2007). *Handbook of Biometrics*. Springer-Verlag, Berlin, Heidelberg.

[89] James, A. P., Al-Jumeily, D., Thampi, S. M., Laskar, M. A., Das, A. J., Talukdar, A. K., and Sarma, K. K. (2015). Second international symposium on computer vision and the internet (visionnet'15) stereo vision-based hand gesture recognition under 3d environment. *Procedia Computer Science*, 58:194 – 201.

[90] Jing, X., Zhu, X., Wu, F., Hu, R., You, X., Wang, Y., Feng, H., and Yang, J. (2017). Super-resolution person re-identification with semi-coupled low-rank discriminant dictionary learning. *IEEE Transactions on Image Processing*, 26(3):1363–1378.

[91] Just, A., Rodriguez, Y., and Marcel, S. (2006). Hand posture classification and recognition using the modified census transform. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 351–356.

[92] Kamarainen, J. (2012). Gabor features in image analysis. In *2012 3rd International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 13–14.

[93] Kang, B., Tripathi, S., and Nguyen, T. Q. (2015). Real-time sign language fingerspelling recognition using convolutional neural networks from depth map. *CoRR*, abs/1509.03001.

[94] Kelly, D., McDonald, J., and Markham, C. (2010). A person independent system for recognition of hand postures used in sign language. *Pattern Recognition Letters*, 31(11):1359 – 1368.

[95] Kim, B.-K., Lee, H., Roh, J., and Lee, S.-Y. (2015). Hierarchical committee of deep cnns with exponentially-weighted decision fusion for static facial expression recognition. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, pages 427–434, New York, NY, USA. ACM.

[96] Kim, B.-K., Roh, J., Dong, S.-Y., and Lee, S.-Y. (2016). Hierarchical committee of deep convolutional neural networks for robust facial expression recognition. *Journal on Multimodal User Interfaces*, 10(2):173–189.

[97] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

[98] Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. cite arxiv:1312.6114.

[99] Kishore, P. V. V. and Kumar, P. R. (2012). Segment, track, extract, recognize and convert sign language videos to voice/text. *International Journal of Advanced Computer Science and Applications*, 3(6):35–47.

[100] Kishore, P. V. V. and Prasad, M. V. D. (2015). Optical flow hand tracking and active contour hand shape features for continuous sign language recognition with artificial neural networks. *International Journal of Software Engineering and Its Applications*, 9(12):231 – 250.

[101] Koller, O., Ney, H., and Bowden, R. (2016). Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3793–3802.

[102] Koller, O., Zargaran, S., Ney, H., and Bowden, R. (2016). Deep sign: Hybrid cnn-hmm for continuous sign language recognition. In *BMVC*.

[103] Kong, W. and Ranganath, S. (2014). Towards subject independent continuous sign language recognition: A segment and merge approach. *Pattern Recognition*, 47(3):1294 – 1308. Handwriting Recognition and other PR Applications.

[104] Kosti, R., Alvarez, J. M., Recasens, A., and Lapedriza, A. (2017). Emotion recognition in context. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1960–1968.

[105] Kotsia, I. and Pitas, I. (2007). Facial expression recognition in image sequences using geometric deformation features and support vector machines. *IEEE Transactions on Image Processing*, 16(1):172–187.

[106] Kuhn, R., Junqua, J. ., Nguyen, P., and Niedzielski, N. (2000). Rapid speaker adaptation in eigenvoice space. *IEEE Transactions on Speech and Audio Processing*, 8(6):695–707.

[107] Kumar, P., Gauba, H., Roy, P. P., and Dogra, D. P. (2017). A multimodal framework for sensor based sign language recognition. *Neurocomputing*, 259:21 – 38. Multimodal Media Data Understanding and Analytics.

[108] Kurakin, A., Zhang, Z., and Liu, Z. (2012). A real time system for dynamic hand gesture recognition with a depth sensor. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 1975–1979.

[109] LeCun, Y., Kavukcuoglu, K., and Farabet, C. (2010). Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 253–256.

[110] Lee, Y.-J., Yeh, Y.-R., and Pao, H.-K. (NA). An introduction to support vector machines. pages 1–18. Department of Computer Science and Information Engineering National Taiwan University of Science and Technology Taipei, Taiwan.

[111] Lenz, I., Lee, H., and Saxena, A. (2015). Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724.

[112] Levenson RW, Ekman P, F. W. (1990). Voluntary facial action generates emotion-specific autonomic nervous system activity. *Psychophysiology*, 27(4).

[113] Levi, G. and Hassner, T. (2015). Emotion recognition in the wild via convolutional neural networks and mapped binary patterns. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, pages 503–510, New York, NY, USA. ACM.

[114] Li, S. Z. and Jain, A. K. (2011a). *Handbook of Face Recognition*. Springer Publishing Company, Incorporated, 2nd edition.

[115] Li, S. Z. and Jain, A. K. (2011b). *Handbook of Face Recognition*. Springer Publishing Company, Incorporated, 2nd edition.

[116] Liang, R., Liang, G., Li, W., Li, Q., and Wang, J. J. (2016). Learning convolutional neural network to maximize pos@top performance measure. *CoRR*, abs/1609.08417.

[117] Liao, S., Zhu, X., Lei, Z., Zhang, L., and Li, S. Z. (2007). Learning multi-scale block local binary patterns for face recognition. In Lee, S.-W. and Li, S. Z., editors, *Advances in Biometrics*, pages 828–837, Berlin, Heidelberg. Springer Berlin Heidelberg.

[118] Littlewort, G., Whitehill, J., Wu, T., Fasel, I., Frank, M., Movellan, J., and Bartlett, M. (2011). The computer expression recognition toolbox (cert). In *Face and Gesture 2011*, pages 298–305.

[119] Liu, C. and Wechsler, H. (2002). Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image Processing*, 11(4):467–476.

[120] Liu, M., Li, S., Shan, S., and Chen, X. (2013). Au-aware deep networks for facial expression recognition. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–6.

[121] Liu, M., Li, S., Shan, S., and Chen, X. (2015). Au-inspired deep networks for facial expression feature learning. *Neurocomput.*, 159(C):126–136.

[122] Liu, P., Han, S., Meng, Z., and Tong, Y. (2014). Facial expression recognition via a boosted deep belief network. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1805–1812.

[123] Liu, T., Zhou, W., and Li, H. (2016). Sign language recognition with long short-term memory. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2871–2875.

[124] Long, M., Wang, J., and Jordan, M. I. (2016a). Deep transfer learning with joint adaptation networks. *CoRR*, abs/1605.06636.

[125] Long, M., Zhu, H., Wang, J., and Jordan, M. I. (2016b). Unsupervised domain adaptation with residual transfer networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 136–144, USA. Curran Associates Inc.

[126] Lopes, A. T., de Aguiar, E., Souza, A. F. D., and Oliveira-Santos, T. (2017). Facial expression recognition with convolutional neural networks: Coping with few data and the training sample order. *Pattern Recognition*, 61(Supplement C):610 – 628.

[127] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., and Matthews, I. (2010). 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*, pages 94–101.

[128] Lyons, M., Akamatsu, S., Kamachi, M., and Gyoba, J. (1998). Coding facial expressions with gabor wavelets. In *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 200–205.

[129] Maalej, A., Amor, B. B., Daoudi, M., Srivastava, A., and Berretti, S. (2011). Shape analysis of local facial patches for 3d facial expression recognition. *Pattern Recognition*, 44(8):1581 – 1589.

[130] Madani, H. and Nahvi, M. (2013). Isolated dynamic persian sign language recognition based on camshift algorithm and radon transform. In *Pattern Recognition and Image Analysis (PRIA), 2013 First Iranian Conference on*, pages 1–5.

[131] Marin, G., Dominio, F., and Zanuttigh, P. (2014). Hand gesture recognition with leap motion and kinect devices. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 1565–1569.

[132] Marin, G., Dominio, F., and Zanuttigh, P. (2016). Hand gesture recognition with jointly calibrated leap motion and depth sensor. *Multimedia Tools and Applications*, 75(22):14991–15015.

[133] Martinez, A., Wilbur, R., Shay, R., and Kak, A. (2002). Purdue rvl-slll asl database for automatic recognition of american sign language. In *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pages 167–172.

[134] Martins, C. (2014). Contributions to the automatic recognition of portuguese sign language. Master's thesis, FEUP.

[135] Mesquita, I. and Silva, S. (2007). *Guia prático de Língua Gestual Portuguesa: Ouvir o silêncio*. Nova Educação.

[136] Michael, N., Neidle, C., and Metaxas, D. (2019). Computer-based recognition of facial expressions in asl: From face tracking to linguistic interpretation.

[137] Ming, K. W. and Ranganath, S. (2002). Representations for facial expressions. In *in Proceedings of the 7th International Conference on Control Automation, Robotics and Vision*, volume 2, pages 716–721.

[138] Mittal, A., Kumar, P., Roy, P. P., Balasubramanian, R., and Chaudhuri, B. B. (2019). A modified-lstm model for continuous sign language recognition using leap motion. *IEEE Sensors Journal*, pages 1–1.

[139] Mollahosseini, A., Chan, D., and Mahoor, M. H. (2016). Going deeper in facial expression recognition using deep neural networks. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10.

[140] Morais, A., Jardim, J. C., and e Ana Mineiro, A. S. (2011). Beyond hands: elements for the study of facial expression in portuguese sign language. *Cadernos de Saúde*, 4(1):37–42.

[141] Neto, G. M. R., Junior, G. B., de Almeida, J. D. S., and de Paiva, A. C. (2018). Sign language recognition based on 3d convolutional neural networks. In Campilho, A., Karray, F., and ter Haar Romeny, B., editors, *Image Analysis and Recognition*, pages 399–407, Cham. Springer International Publishing.

[142] Neverova, N., Wolf, C., Taylor, G., and Nebout, F. (2016). Moddrop: Adaptive multi-modal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1692–1706.

[143] Ng, A. Y. and Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, pages 841–848. MIT Press.

[144] Ng, H.-W., Nguyen, V. D., Vonikakis, V., and Winkler, S. (2015). Deep learning for emotion recognition on small datasets using transfer learning. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, pages 443–449, New York, NY, USA. ACM.

[145] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011). Multimodal deep learning. *In International Conference on Machine Learning (ICML)*, 6.

[146] Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987.

[147] Oliveira, M., Chatbri, H., Little, S., O'Connor, N. E., and Sutherland, A. (2017). A comparison between end-to-end approaches and feature extraction based approaches for sign language recognition. In *2017 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 1–6.

[148] Oliveira, O. (2013). Tradutor da lingua gestual portuguesa - modelo de tradução bidireccional. Master's thesis, ISEP.

[149] Ong, E.-J. and Bowden, R. (2004). A boosted classifier tree for hand shape detection. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 889–894.

[150] Oyedotun, O. K. and Khashman, A. (2017). Deep learning in vision-based static hand gesture recognition. *Neural Computing and Applications*, 28(12):3941–3951.

[151] P., E. (2007). Emotions revealed: Recognizing faces and feelings to improve communication and emotional life. *Macmillan*.

[152] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

[153] Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition. In *British Machine Vision Conference*.

[154] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*.

[155] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

[156] Pereira, E. M. (2016). *Humans in Action at Different Levels: the group, the whole, and the parts*. PhD thesis, FEUP, UT Austin|Portugal.

[157] Pigou, L., Dieleman, S., Kindermans, P.-J., and Schrauwen, B. (2015a). *Sign Language Recognition Using Convolutional Neural Networks*, pages 572–578. Springer International Publishing, Cham.

[158] Pigou, L., Dieleman, S., Kindermans, P.-J., and Schrauwen, B. (2015b). Sign language recognition using convolutional neural networks. In Agapito, L., Bronstein, M. M., and Rother, C., editors, *Computer Vision - ECCV 2014 Workshops*, pages 572–578, Cham. Springer International Publishing.

[159] Potter, L. E., Araullo, J., and Carter, L. (2013). The leap motion controller: A view on sign language. In *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, OzCHI '13, pages 175–178, New York, NY, USA. ACM.

[160] Prabhakar, S., Pankanti, S., and Jain, A. K. (2003). Biometric recognition: Security and privacy concerns. *IEEE Security and Privacy*, 1(2):33–42.

[161] Raghavendra, R. and Busch, C. (2015). Robust scheme for iris pres. attack det. using multiscale binarized statistical image features. *IEEE TIFS*, 10(4):703–715.

[162] Raheja, J. L., Mishra, A., and Chaudhary, A. (2016). Indian sign language recognition using svm. *Pattern Recognition and Image Analysis*, 26(2):434–441.

[163] Rahulamathavan, Y., Phan, R. C. W., Chambers, J. A., and Parish, D. J. (2013). Facial expression recognition in the encrypted domain based on local fisher discriminant analysis. *IEEE Transactions on Affective Computing*, 4(1):83–92.

[164] Rajendra, P., Sudheer, K., and Boadh, R. (2017). Design of a recognition system automatic vehicle license plate through a convolution neural network. *International Journal of Computer Applications*, 177:47–54.

[165] Ramachandram, D. and Taylor, G. W. (2017). Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108.

[166] Ramirez, C., Kreinovich, V., and Argaez1, M. (2013). Why l_1 is a good approximation to l_0: A geometric explanation. *Journal of Uncertain Systems*, 7.

[167] Rivera, A. R., Castillo, J. R., and Chae, O. O. (2013). Local directional number pattern for face analysis: Face and expression recognition. *IEEE Transactions on Image Processing*, 22(5):1740–1752.

[168] Rodrigues, I. V. (2014). Analysis of expressiveness of portuguese sign language speakers. Master's thesis, FEUP.

[169] Rossol, N., Cheng, I., and Basu, A. (2016). A multisensor technique for gesture recognition through intelligent skeletal pose analysis. *IEEE Transactions on Human-Machine Systems*, 46(3):350–359.

[170] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

[171] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA.

[172] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

[173] Sako, H. and Smith, A. (1996). Real-time facial expression recognition based on features' positions and dimensions. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 3, pages 643–648 vol.3.

[174] Sandbach, G., Zafeiriou, S., Pantic, M., and Yin, L. (2012). Static and dynamic 3d facial expression recognition: A comprehensive survey. *Image and Vision Computing*, 30(10):683 – 697. 3D Facial Behaviour Analysis and Understanding.

[175] Sariyanidi, E., Gunes, H., and Cavallaro, A. (2015). Automatic analysis of facial affect: A survey of registration, representation, and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(6):1113–1133.

[176] Schmidt, K. L. and Cohn, J. F. (2001). Human facial expressions as adaptations:evolutionary questions in facial expression research. *Psychophysiology*, 33(3.24).

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

[178] Schroff, F., Kalenichenko, D., and Philbin, J. (2015b). Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832.

[179] Sequeira, A. F., Thavalengal, S., Ferryman, J., Corcoran, P., and Cardoso, J. S. (2016). A realistic evaluation of iris presentation attack detection. In *39th International Conference on Telecommunications and Signal Processing (TSP)*, pages 660–664.

[180] Shan, C., Gong, S., and McOwan, P. W. (2009). Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803 – 816.

[181] Shan, K., Guo, J., You, W., Lu, D., and Bie, R. (2017). Automatic facial expression recognition based on a deep convolutional-neural-network structure. In *2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA)*, pages 123–128.

[182] Shanableh, T. and Assaleh, K. (2011). User-independent recognition of arabic sign language for facilitating communication with the deaf community. *Digital Signal Processing*, 21(4):535 – 542.

[183] Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 3483–3491. Curran Associates, Inc.

[184] Sohn, K., Shang, W., and Lee, H. (2014). Improved multimodal deep learning with variation of information. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2141–2149. Curran Associates, Inc.

[185] Song, I., Kim, H. J., and Jeon, P. B. (2014). Deep learning for real-time robust facial expression recognition on a smartphone. In *2014 IEEE International Conference on Consumer Electronics (ICCE)*, pages 564–567.

[186] Sotelo, C. (2014). Portuguese sign language recognition from depth sensing human gesture and motion capture. Master's thesis, Universidade do Minho.

[187] Srinivas, S., Sarvadevabhatla, R. K., Mopuri, K. R., Prabhu, N., Kruthiventi, S., and Radhakrishnan, V. B. (2016a). A taxonomy of deep convolutional neural nets for computer vision. *Frontiers in Robotics and AI*, 2(36).

[188] Srinivas, S., Sarvadevabhatla, R. K., Mopuri, K. R., Prabhu, N., Kruthiventi, S. S. S., and Babu, R. V. (2016b). A taxonomy of deep convolutional neural nets for computer vision. *Frontiers in Robotics and AI*, 2:36.

[189] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

[190] Starner, T., Weaver, J., and Pentland, A. (1998). Real-time american sign language recognition using desk and wearable computer based video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12):1371–1375.

[191] Stokoe, W. C. (2005). Sign language spotting with a threshold model based on conditional random fields. *Journal of deaf studies and deaf education*, (10):3–37.

[192] Su, F. and Wang, J. (2018). Domain transfer convolutional attribute embedding. *CoRR*, abs/1803.09733.

[193] Sun, X., Xu, H., Zhao, C., and Yang, J. (2008). Facial expression recognition based on histogram sequence of local gabor binary patterns. In *2008 IEEE Conference on Cybernetics and Intelligent Systems*, pages 158–163.

[194] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9.

[195] Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer-Verlag, Berlin, Heidelberg, 1st edition.

[196] Tang, H. and Huang, T. S. (2008). 3d facial expression recognition based on properties of line segments connecting facial feature points. In *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–6.

[197] Tang, Y. (2013). Deep learning using support vector machines. *CoRR*, abs/1306.0239.

[198] Tariq, U., Lin, K. H., Li, Z., Zhou, X., Wang, Z., Le, V., Huang, T. S., Lv, X., and Han, T. X. (2011). Emotion recognition from an ensemble of features. In *Face and Gesture 2011*, pages 872–877.

[199] **Ferreira, P. M.**, Cardoso, J. S., and Rebelo, A. (2016). Facial key-points detection using a convolutional encoder-decoder model. In *RecPad 2016: Conference on Pattern Recognition*, pages 1–2.

[200] **Ferreira, P. M.**, Cardoso, J. S., and Rebelo, A. (2017a). Multimodal learning for sign language recognition. In Alexandre, L. A., Salvador Sánchez, J., and Rodrigues, J. M. F., editors, *Pattern Recognition and Image Analysis*, pages 313–321, Cham. Springer International Publishing.

[201] **Ferreira, P. M.**, Cardoso, J. S., and Rebelo, A. (2017b). The potential of multimodal learning for sign language recognition. In *RecPad 2017: Conference on Pattern Recognition*, pages 1–2. **(best paper award)**.

[202] **Ferreira, P. M.**, Cardoso, J. S., and Rebelo, A. (2019a). On the role of multimodal learning in the recognition of sign language. *Multimedia Tools and Applications*, 78(8):10035–10056.

[203] **Ferreira, P. M.**, Marques, F., Cardoso, J. S., and Rebelo, A. (2018a). An expression-specific deep neural network for emotion recognition. In *RecPad 2018: Conference on Pattern Recognition*, pages 1–2.

[204] **Ferreira, P. M.**, Marques, F., Cardoso, J. S., and Rebelo, A. (2018b). Physiological inspired deep neural networks for emotion recognition. *IEEE Access*, 6:53930–53943.

[205] **Ferreira, P. M.**, Pernes, D., Fernandes, K., Rebelo, A., and Cardoso, J. S. (2018c). Dimensional emotion recognition using visual and textual cues. *CoRR*, abs/1805.01416.

[206] **Ferreira, P. M.**, Pernes, D., Rebelo, A., and Cardoso, J. S. (2019b). Desire: Deep signer-invariant representations for sign language recognition. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–16.

[207] **Ferreira, P. M.**, Pernes, D., Rebelo, A., and Cardoso, J. S. (2019c). Learning signer-invariant representations with adversarial training. In *The 12th International Conference on Machine Vision (ICMV 2019)*.

[208] **Ferreira, P. M.**, Pernes, D., Rebelo, A., and Cardoso, J. S. (2019d). Signer-independent sign language recognition with adversarial neural networks. *International Journal of Machine Learning and Computing (IJMLC)*. (accepted).

[209] **Ferreira, P. M.**, Rodrigues, I. V., Rio, A., Sousa, R., Pereira, E. M., and Rebelo, A. (2014). Corsil: A novel dataset for portuguese sign language and expressiveness recognition. In *RecPad 2014: Conference on Pattern Recognition*, pages 1–2.

[210] **Ferreira, P. M.**, Sequeira, A. F., Pernes, D., Rebelo, A., and Cardoso, J. S. (2019e). Adversarial learning for a robust iris presentation attack detection method against unseen attack presentations. In *2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*.

[211] Theano Development Team (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

[212] Triesch, J. and von der Malsburg, C. (2001). A system for person-independent hand posture recognition against complex backgrounds. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(12):1449–1453.

[213] Trigueiros, P., Ribeiro, F., and Reis, L. P. (2014). *Vision-Based Portuguese Sign Language Recognition System*, pages 605–617. Springer International Publishing, Cham.

[214] Ulusoy, I. and Bishop, C. M. (2006). *Comparison of Generative and Discriminative Techniques for Object Detection and Classification*, pages 173–195. Springer Berlin Heidelberg, Berlin, Heidelberg.

[215] Uçar, A. (2017). Deep convolutional neural networks for facial expression recognition. In *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 371–375.

[216] Van den Bergh, M. and Van Gool, L. (2011). Combining rgb and tof cameras for real-time 3d hand gesture interaction. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 66–72.

[217] van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

[218] Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg.

[219] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408.

[220] Viola, P. and Jones, M. (2001). Robust real-time object detection. In *International Journal of Computer Vision*.

[221] Vogler, C. and Goldenstein, S. (2008). Facial movement analysis in asl. *Universal Access in the Information Society*, 6(4):363–374.

[222] Vogler, C. and Metaxas, D. (2004). Handshapes and movements: Multiple-channel asl recognition. In *Lecture Notes in Computer Science*, pages 247–258. Springer.

[223] von Agris, U., Blomer, C., and Kraiss, K. (2008). Rapid signer adaptation for continuous sign language recognition using a combined approach of eigenvoices, mllr, and map. In *2008 19th International Conference on Pattern Recognition*, pages 1–4.

[224] von Agris, U. and Kraiss, K.-F. (2010). Signum database: Video corpus for signer-independent continuous sign language recognition. In *Proceedings of the 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies (Language Resources and Evaluation Conference LREC 2010)*, pages 243–246, Valletta, Malta.

[225] von Agris, U., Schneider, D., Zieren, J., and Kraiss, K. . (2006). Rapid signer adaptation for isolated sign language recognition. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 159–159.

[226] von Agris, U., Zieren, J., Canzler, U., Bauer, B., and Kraiss, K.-F. (2008). Recent developments in visual sign language recognition. *Universal Access in the Information Society*, 6(4):323–362.

[227] Voulodimos, A., Doulamis, N. D., Doulamis, A., and Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. In *Comp. Int. and Neurosc.*

[228] Walecki, R., Rudovic, O., Pavlovic, V., and Pantic, M. (2015). Variable-state latent conditional random fields for facial expression recognition and action unit detection. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 1, pages 1–8.

[229] Wang, A., Cai, J., Lu, J., and Cham, T. J. (2015a). Mmss: Multi-modal sharable and specific feature learning for rgb-d object recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1125–1133.

[230] Wang, A., Lu, J., Cai, J., Cham, T. J., and Wang, G. (2015b). Large-margin multi-modal deep learning for rgb-d object recognition. *IEEE Transactions on Multimedia*, 17(11):1887–1898.

[231] Wang, J. J.-Y., Wang, Y., Zhao, S., and Gao, X. (2015c). Maximum mutual information regularized classification. *Engineering Applications of Artificial Intelligence*, 37:1 – 8.

[232] Wang, M. and Deng, W. (2018). Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135 – 153.

[233] Wang, Z. and Ying, Z. (2012). Facial expression recognition based on local phase quantization and sparse representation. In *2012 8th International Conference on Natural Computation*, pages 222–225.

[234] Weichert, F., Bachmann, D., Rudak, B., and Fisseler, D. (2013). Analysis of the accuracy and robustness of the leap motion controller. *Sensors (Basel, Switzerland)*, 13:6380–6393.

[235] Wu, D., Pigou, L., Kindermans, P., Le, N. D., Shao, L., Dambre, J., and Odobez, J. (2016). Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1583–1597.

[236] Wu, F., Jing, X., Dong, X., Hu, R., Yue, D., Wang, L., Ji, Y., Wang, R., and Chen, G. (2018). Intraspectrum discrimination and interspectrum correlation analysis deep network for multispectral face recognition. *IEEE Transactions on Cybernetics*, pages 1–14.

[237] Wu, Z., Jiang, Y.-G., Wang, J., Pu, J., and Xue, X. (2014). Exploring inter-feature and inter-class relationships with deep neural networks for video classification. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 167–176, New York, NY, USA. ACM.

[238] Xu, L., Lu, C., Xu, Y., and Jia, J. (2011). Image smoothing via l0 gradient minimization. *ACM Trans. Graph.*, 30(6):174:1–174:12.

[239] Xu, M., Raytchev, B., Sakaue, K., Hasegawa, O., Koizumi, A., Takeuchi, M., and Sagawa, H. (2000). *Advances in Multimodal Interfaces — ICMI 2000: Third International Conference Beijing, China, October 14–16, 2000 Proceedings*, chapter A Vision-Based Method for Recognizing Non-manual Information in Japanese Sign Language, pages 572–581. Springer Berlin Heidelberg, Berlin, Heidelberg.

[240] Yan, H., Ding, Y., Li, P., Wang, Q., Xu, Y., and Zuo, W. (2017). Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. *CoRR*, abs/1705.00609.

[241] Yang, H.-D. (2015). Sign language recognition with the kinect sensor based on conditional random fields. *Sensors*, 15(1):135 – 147.

[242] Yang, H.-D., Sclaroff, S., and Lee, S.-W. (2009). Sign language spotting with a threshold model based on conditional random fields. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(7):1264–1277.

[243] Yang, M. H., Ahuja, N., and Tabb, M. (2002). Extraction of 2d motion trajectories and its application to hand gesture recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(8):1061–1074.

[244] Yin, F., Chai, X., and Chen, X. (2016). Iterative reference driven metric learning for signer independent isolated sign language recognition. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 434–450, Cham. Springer International Publishing.

[245] Yin, F., Chai, X., Zhou, Y., and Chen, X. (2015). Weakly supervised metric learning towards signer adaptation for sign language recognition. In Xianghua Xie, M. W. J. and Tam, G. K. L., editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 35.1–35.12. BMVA Press.

[246] Yu, S.-H., Huang, C.-L., Hsu, S.-C., Lin, H.-W., and Wang, H.-W. (2011). Vision-based continuous sign language recognition using product hmm. In *Pattern Recognition (ACPR), 2011 First Asian Conference on*, pages 510–514.

[247] Yu, Z. and Zhang, C. (2015). Image based static facial expression recognition with multiple deep network learning. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, pages 435–442, New York, NY, USA. ACM.

[248] Zafrulla, Z., Brashear, H., Hamilton, H., and Starner, T. (2010). A novel approach to american sign language (asl) phrase verification using reversed signing. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 48–55.

[249] Zhang, G., Liang, G., Li, W., Fang, J., Wang, J., Geng, Y., and Wang, J.-Y. (2017). Learning convolutional ranking-score function by query preference regularization. In Yin, H., Gao, Y., Chen, S., Wen, Y., Cai, G., Gu, T., Du, J., Tallón-Ballesteros, A. J., and Zhang, M., editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2017*, pages 1–8, Cham. Springer International Publishing.

[250] Zhang, H., Sun, Z., and Tan, T. (23 - 26 August 2010). Contact lens detection based on weighted LBP. In *20th ICPR*, pages 4279–4282.

[251] Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503.

[252] Zhang, L., Tjondronegoro, D., and Chandran, V. (2014). Representation of facial expression categories in continuous arousal–valence space: Feature and correlation. *Image and Vision Computing*, 32(12):1067 – 1079.

[253] Zhang, Z., Lyons, M., Schuster, M., and Akamatsu, S. (1998). Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 454–459.

[254] Zhou, Y., Yang, X., Zhang, Y., Xu, X., Wang, Y., Chai, X., and Lin, W. (2015). Unsupervised adaptive sign language recognition based on hypothesis comparison guided cross validation and linguistic prior filtering. *Neurocomputing*, 149:1604 – 1612.

[255] Zhu, X., Jing, X., You, X., Zhang, X., and Zhang, T. (2018). Video-based person re-identification by simultaneously learning intra-video and inter-video distance metrics. *IEEE Transactions on Image Processing*, 27(11):5683–5695.

[256] Zhuang, F., Cheng, X., Luo, P., Pan, S. J., and He, Q. (2015). Supervised representation learning: Transfer learning with deep autoencoders. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 4119–4125. AAAI Press.

[257] Zieren, J. and Kraiss, K.-F. (2005). Robust person-independent visual sign language recognition. In Marques, J. S., Pérez de la Blanca, N., and Pina, P., editors, *Pattern Recognition and Image Analysis*, pages 520–528, Berlin, Heidelberg. Springer Berlin Heidelberg.

Appendix A

CorSiL content

In this appendix, the isolated signs and sentences that constitute the proposed CorSiL database are listed.

Table A.1: CorSiL Database isolated signs (Portuguese version of the signs).

Category	Sign
Alphabet	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q R, S, T, U, V, W, X, Y, Z
Cardinal numbers	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Pronouns	Eu, Tu, Ele, Nós, Vós, Eles, Meu, Teu, Nosso, Vosso, Como, Onde, Porquê, Qual, Quando, Quem
Verbs	Andar, Aprender, Beber, Cair, Comer, Comprar, Condizir, Correr, Ensinar, Escrever, Estudar, Falar, Gostar, Ir, Jogar, Ler, Ouvir, Partir, Perder, Trazer, Vender, Ver, Vir
Adverbs	Bom, Bonito, Feio, Grande, Mau, Novo, Pequeno, Sujo, Velho
Basic Expressions	Adeus, Ajudar, Com licença, Desculpe, Não, Obrigado, Olá, Por favor, Sim
Feelings / Emotions	Aborrecido, Amor, Cansado, Doente, Feliz, Triste, Zangado
Colors	Amarelo, Azul, Branco, Castanho, Cinzento, Laranja, Preto, Rosa, Roxo, Verde, Vermelho
Family	Avó-Avô, Bebé, Casado, Divorciado, Filho, Irmão-Irmã, Mãe, Pai, Rapaz-Rapariga, Solteiro
Professions	Advogado, Arquiteto, Bombeiro, Cientista, Enfermeiro, Engenheiro, Médico, Músico, Policia, Professor
Places	Casa, Casa de Banho, Cidade, Cozinha, Escola, Hospital, Hotel, Igreja, Loja, País, Praia, Quarto, Restaurante
Food	Água, Café, Carne, Copo, Maça, Peixe, Prato, Queijo, Talheres
Animals	Cão, Cavalo, Gato, Inseto, Ovelha, Pássaro, Porco, Vaca
Time	Amanhã, Ano, Dia, Hoje, Mês, Noite, Ontem, Tarde
Weather	Calor, Chuva, Frio, Neve, Nevoeiro, Sol

Table A.2: CorSiL Database sentences (Portuguese version of the sentences).

#	Sentences
1	Tudo bem?
2	A tua família como está?
3	Como te chamas?
4	Qual é o nome da tua mãe?
5	Que idade tens?
6	Onde vives?
7	Qual é o teu número de telefone?
8	Onde trabalhas?
9	Qual é a tua Profissão?
10	És surdo ou ouvinte?
11	Tens irmãos?
12	Tens animais de estimação?
13	Queres ir ao cinema?
14	Onde fica a estação de metro mais próxima?
15	Qual é o melhor hotel da cidade?
16	Eu sou médico.
17	Eu tenho 30 anos.
18	O meu irmão quer ser jogador de futebol.
19	Eu tenho dois irmãos e uma irmã.
20	O meu irmão está à procura de trabalho.
21	Os meus avós vivem em nossa casa.
22	Eu vivo em Portugal mas nasci em Itália.
23	Eu vou de autocarro para a escola.
24	Hoje de manha fui de carro para o trabalho.
25	A minha irmã joga basquetebol.
26	Ontem fui ver um jogo de futebol.
27	Eu costumo ler antes de adormecer.
28	O meu pai comprou um livro novo.
29	A minha irmã adora fazer compras.
30	O meu primo vai casar no próximo ano.
31	No próximo fim-de-semana vou ao cinema com os meus amigos.
32	O meu amigo tem uma casa de praia.
33	Eu conheci uma rapariga muito bonita.
34	Ontem senti-me doente e fui ao hospital.
35	Eu gosto de tomar o pequeno-almoço no café.
36	Hoje de manha comi leite com cereais.
37	Eu prefiro comer carne do que peixe.
38	Amanha vou jantar a um restaurante indiano com a minha família.
39	Amanhã vai chover.
40	Hoje está frio.

Appendix B

Extension of the Proposed Adversarial Training Objective

Besides the signer-independent SLR problem, there are many other applications in which it is desirable to learn feature representations invariant to some domain or aspect. One of such potential applications is the biometric liveness detection.

Biometrics uses the individual's unique physical or behavior traits (e.g., fingerprint, face, iris, etc.) for personal identification purposes [88]. Biometric recognition systems are currently considered reliable enough to be deployed in several applications, ranging from government to civilian applications. However, the shift from a controlled acquisition process to a more autonomous acquisition scenario increased the vulnerabilities of biometric systems to different types of presentation attacks [160]. A presentation attack is any attempt to interfere with the intended purpose of a biometric system, and a presentation attack instrument (PAI) is a biometric characteristic or object used in a presentation attack. Methods to determine if a biometric sample is altered or fake constitute the denominated presentation attack detection (PAD) methods, also know as liveness detection methods.

The problem of liveness detection of a biometric trait can be seen as a binary classification problem where an input trait sample has to be assigned to one of two classes: real (aka. bona-fide) or fake. The key point of the process is to find a set of discriminant features which permits to build an appropriate classifier to predict the probability of the sample vitality given the extracted set of features [69]. The adaptation of the proposed adversarial framework, previously presented in chapter 7, for the liveness detection problem is depicted in the following section.

B.1 The proposed adversarial framework for liveness detection

We intend to apply the proposed adversarial framework, previously presented in chapter 7, to enforce the feature representations to be invariant to the PAI species/types, and, hence, increase the generalization capability of the biometric system to unseen presentation attacks.

Formally, let $\mathbb{X} = \{\boldsymbol{X}_i, y_i, s_i\}_{i=1}^{N}$ denote a labeled dataset of N samples, where $\boldsymbol{X}_i$ represents the i-th input feature vector, and y_i and s_i denote the corresponding class label (i.e. real or fake) and PAI specie, respectively. $\mathbb{X}$ comprises feature vectors extracted from real and fake biometric samples. Let $\mathbb{X}^{bf}$ and $\mathbb{X}^{a}$ denote these partitions and N^{bf} and N^{a} their cardinality, respectively. The adaptation of the proposed adversarial framework for the liveness detection problem is depicted in Figure B.1. Although it follows the original adversarial framework, there some major differences:

- It uses as input feature vectors extracted with the state-of-the-art liveness detection method proposed by Sequeira *et al.* [179] (as further detailed in Section B.2.2). Therefore, the topology of the layers in the original *encoder* network was changed from convolutional to fully connected.

- To better match the biometrics terminology, both classifiers of the original framework were renamed from *sign-classifier* and *signer-classifier* to *task-classifier* and *species-classifier*, respectively. In this context, while the *task-classifier* predicts the class labels (i.e., real or fake samples), the *species-classifier* predicts the PAI species/types. Accordingly, the loss terms of both classifiers were also renamed from $\mathcal{L}_{\text{sign}}$ and $\mathcal{L}_{\text{signer}}$ to $\mathcal{L}_{\text{task}}$ and $\mathcal{L}_{\text{species}}$, respectively.

- During training, we aim to learn PAI species-invariant representations. Therefore, $\mathcal{L}_{\text{transfer}}$, $\mathcal{L}_{\text{species}}$ and $\mathcal{L}_{\text{adv}}$ are just computed over the fake training samples within every mini-batch. In its turn, the *task-classifier* receives all the samples, i.e. fake and real, within a mini-batch.

B.2 Experimental Evaluation

The experimental evaluation of the adaptation of the proposed adversarial framework for biometric liveness detection was performed using the Visible Spectrum Iris Artefact (VSIA) database [161]. It comprises five different PAI species, including print and electronic screen

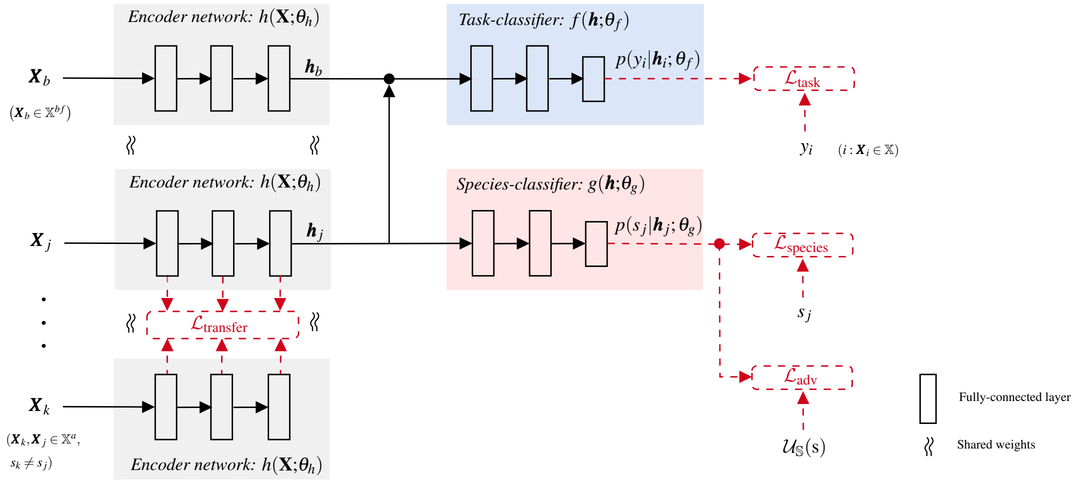

Encoder network: $h(\mathbf{X};\theta_h)$
$\boldsymbol{X}_b$
$(\boldsymbol{X}_b \in \mathbb{X}^{bf})$
$\boldsymbol{h}_b$
Task-classifier: $f(\boldsymbol{h};\theta_f)$
$p(y_i|\boldsymbol{h}_i;\theta_f)$
$\mathcal{L}_{\text{task}}$
y_i
$(i:\boldsymbol{X}_i \in \mathbb{X})$
Encoder network: $h(\mathbf{X};\theta_h)$
$\boldsymbol{X}_j$
$\boldsymbol{h}_j$
Species-classifier: $g(\boldsymbol{h};\theta_g)$
$p(s_j|\boldsymbol{h}_j;\theta_g)$
$\mathcal{L}_{\text{species}}$
s_j
$\mathcal{L}_{\text{transfer}}$
$\boldsymbol{X}_k$
$(\boldsymbol{X}_k, \boldsymbol{X}_j \in \mathbb{X}^a,$
$s_k \neq s_j)$
Encoder network: $h(\mathbf{X};\theta_h)$
$\mathcal{L}_{\text{adv}}$
$\mathcal{U}_{\mathbb{S}}(\text{s})$
Fully-connected layer
Shared weights

Figure B.1: Adaptation of the proposed adversarial framework for the liveness detection problem.

attacks. The methods are evaluated by leaving out one PAI species for testing. The training set is further divided into one species for validation, and the remaining are used for training.

B.2.1 PAD Performance Evaluation Metrics

The experimental results are reported in terms of three standard PAD metrics, as defined in the ISO/IEC 30107-3 [87]: the *attack presentation classification error rate* (*APCER*), the *bona-fide presentation classification error rate* (*BPCER*), and the *average classification error rate* (*ACER*).

The *APCER* is defined as the proportion of attack presentations using the same PAI species incorrectly classified as bona-fide presentations in a specific scenario, such that:

$$APCER = 1 - \left(\frac{1}{N^{PAIS}}\right) \sum_{i=1}^{N^{PAIS}} RES_i, \tag{B.1}$$

where N^{PAIS} is the number of attack presentations for the given *PAI* species, and RES_i takes value 1 if the i-th presentation is classified as an attack presentation, and value 0 if classified as a bona-fide presentation.

The *BPCER* is given by the proportion of bona-fide presentations incorrectly classified as attack presentations in a specific scenario:

$$BPCER = \frac{\sum_{i=1}^{N^{bf}} RES_i}{N^{bf}}, \tag{B.2}$$

where N^{bf} is the number of bona-fide presentations, and RES_i takes value 1 if the i-th presentation is classified as an attack presentation, and value 0 if classified as a bona-fide presentation.

The *ACER* simply consists in the average of both *APCER* and *BPCER* metrics. Although *ACER* is being deprecated, it is still used in this work since it allows a comparison with the state-of-the-art methods.

B.2.2 Baselines and compared methods

In order to assess the effectiveness of the proposed adversarial liveness detection framework, it will be compared with a state-of-the-art liveness detection algorithm and an implemented baseline:

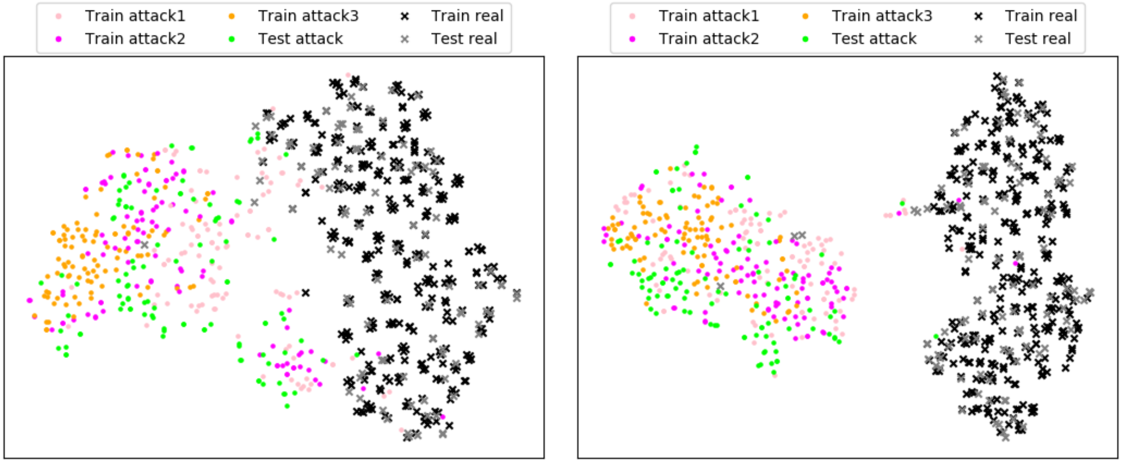
- (Sequeira *et al.* [179]) The state-of-the-art liveness detection method proposed in [179]. It consists in a hand-crafted feature extraction process, based on Weighted Local Binary Patterns (wLBP) [250], followed by a SVM classifier for liveness classification. According to Sequeira *et al.* [179], wLBP demonstrated the better generalization capability to unseen PAI species, among several feature extraction methods.
- (Baseline) It follows exactly the same methodology as the one proposed by Sequeira *et al.* [179]; however, the SVM classifier was replaced by a MLP. It is worth to mentioning that, for a fair comparison, the MLP in the baseline method has exactly the same architecture as the *task-classifier* module of the proposed adversarial liveness detection model.

B.2.3 Implementation details

The implemented models were also implemented in PyTorch and optimized by using the Adam optimization strategy with a batch size of 64. As depicted in Table B.1, the hyperparameters were fine-tuned using a grid-search strategy and cross-validation on the training set. Regarding the architecture of the proposed adversarial liveness detection model, the *encoder* simply consists of a sequence of L_e fully connected layers with 128 neurons, followed by a ReLU activation function. As depicted in Table B.1, L_e was also optimized by means of a grid search approach and cross-validation on the training set. Both classifiers, i.e. the *task-classifier* and the *species-classifier*, follow the same network topology. In particular, it comprises a total of 3 hidden layers with 256 neurons, also with a ReLU, along with a softmax output layer. The number of nodes of the output layer of the *species-classifier* is defined accordingly to the number of species in the training set.

Table B.1: Hyperparameters sets.

Hyperparameters	Acronym	Set
Leaning rate	-	$\{1\text{e}^{-04}, 1\text{e}^{-03}\}$
ℓ^2-norm coefficient	-	$\{1\text{e}^{-05}, 1\text{e}^{-04}\}$
encoder dense layers	L_e	$\{3,4\}$
$\mathcal{L}_{\text{adv}}$ weight	λ	10 values n $\in \{n : n = log_{10}C \wedge n \in [1\text{e}^{-03}, 1]\}$
$\mathcal{L}_{\text{transfer}}$ weight	γ	10 values n $\in \{n : n = log_{10}C \wedge n \in [1\text{e}^{-03}, 1]\}$



(a) Baseline

(b) Proposed model

Figure B.2: Two-dimensional projection of the latent representation space with t-distributed stochastic neighbor embedding (t-SNE) (colored • denote different PAI species; × are bona-fide presentations).

B.2.4 Results and discussion

The experimental results obtained on the VSIA dataset are presented in Tables B.2 and B.3, in which the proposed adversarial liveness detection model is compared against both the implemented baseline and the state-of-the-art method proposed by Sequeira *et al.* [179]. The results are reported in terms of the standard PAD metrics as well as in terms of classification loss and accuracy.

A comparison between the state-of-the-art method of Sequeira *et al.* [179] and the implemented baseline seems to indicate that the replacement of the SVM classifier by an MLP does not translate in improvements. This fact may be explained by the small size of the VSIA dataset. The MLP classifier tends to easily overfit due to the lack of training samples. These results attest the need for novel regularization strategies for deep neural networks, which is, in fact, the main purpose of the proposed adversarial training objective. Interestingly, the proposed adversarial liveness detection model achieved the best average *ACER* (i.e., 7.00% against 7.98% and 9.80%) as well as the best *APCER* and *BPCER* values in most cases. These results clearly reinforce the importance of learning feature representations invariant to the PAI species.

The visualization of the latent representations through the t-SNE confirms the superior performance of the proposed adversarial liveness detection model (see Figure B.2). The proposed model yields a latent representation space in which latent representation of different PAI species are well mixed. At the same time, it provides a better inter-class separability (i.e., latent representations of real and fake samples are kept far apart).

Table B.2: Experimental results achieved by the state-of-the-art liveness detection method of Sequeira *et al*. [179], the implemented baseline and the proposed adversarial liveness detection model in terms of PAD metrics.

Method	Attack 1			Attack 2			Attack 3			Attack 4			Attack 5			Average		
	APCER	BPCER	ACER	APCER	BPCER	ACER	APCER	BPCER	ACER	APCER	BPCER	ACER	APCER	BPCER	ACER	APCER	BPCER	ACER
Sequeira *et al*. [179]	-	-	21.15	-	-	9.61	-	-	1.92	-	-	4.32	-	-	2.88	-	-	7.98
Baseline	39.00	5.00	22.00	9.00	**5.00**	**7.00**	**0.00**	11.00	5.50	12.00	8.00	10.00	3.00	6.00	4.50	12.60	7.00	9.80
Proposed	**33.00**	**3.00**	**18.00**	**6.00**	8.00	**7.00**	**0.00**	**4.00**	2.00	**6.00**	**5.00**	5.50	**0.00**	**5.00**	**2.50**	**9.00**	**5.00**	**7.00**

Table B.3: Experimental results achieved by the implemented baseline and proposed adversarial liveness detection model in terms of loss ($\mathcal{L}_{\text{task}}$) and classification accuracy (Acc).

Method	Attack 1		Attack 2		Attack 3		Attack 4		Attack 5		Average	
	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc
Baseline	0.50	78.00	**0.27**	**93.00**	0.10	94.50	0.25	90.00	0.12	95.50	0.25	90.20
Proposed	**0.42**	**82.00**	**0.27**	**93.00**	**0.07**	**98.00**	**0.18**	**94.50**	**0.10**	**97.50**	**0.21**	**93.00**

Appendix C

Facial Expressions Recognition: a dimensional approach

The content presented in this Chapter was published in [205]:

- **Ferreira, P. M.**, Pernes, D., Fernandes, K., Rebelo, A., and Cardoso, J. S. (2018c). Dimensional emotion recognition using visual and textual cues. *CoRR*, abs/1805.01416

This Appendix summarizes our participation in the One-Minute Gradual-Emotional Behavior challenge (OMG-Emotion challenge) [6]. The underlying objective of the challenge is the automatic estimation of emotion expressions in the two-dimensional emotion representation space (i.e., arousal and valence). The adopted methodology is a weighted ensemble of several models from both video and text modalities. For video-based recognition, two different types of visual cues (i.e., face and facial landmarks) were considered to feed a multi-input deep neural network. Regarding the text modality, a sequential model based on a simple recurrent architecture was implemented. In addition, a model based on high-level features, in order to embed domain knowledge in the learning process, was also introduced. Experimental results on the OMG-Emotion validation set demonstrate the effectiveness of the implemented ensemble model as it clearly outperforms the current baseline methods.

C.1 Introduction

Some recent research trends within emotion recognition have resorted to the dimensional description of facial expressions. It is the example of the One-Minute Gradual-Emotional Behavior challenge (OMG-Emotion challenge) [6]. The OMG-Emotion competition focuses on long-term emotion recognition in the arousal/valence space. The OMG-Emotion Dataset [21] is composed of 420 relatively long emotion videos with an average length of 1 minute. The videos of the dataset are divided into clips based on utterances, and each utterance is annotated by at least five independent subjects. Each annotator could take into consideration not only the vision and audio information but also the context of each video. That is, each annotator watched the clips of a video in sequence and had to annotate each video using an arousal/valence scale and a categorical emotion based on the universal emotions from Ekman.

In this Appendix, an emotion recognition methodology for the OMG-Emotion challenge is presented. The goal is to predict one value of arousal and valence for each video utterance. The implemented methodology is an ensemble of several models from two distinct modalities, namely video and text. More concretely, four different types of models were implemented for the ensemble:

- *Face model*: a multi-input deep neural network fed with the extracted faces of the input sequence frames;
- *Facial landmarks model*: a multi-input deep neural network fed with the facial landmarks of each frame;
- *Sequential deep text model*: a recurrent deep neural network with an embedding layer initialized with the weights of GloVe [155];
- *Feature-engineering text model*: a two-stream multi-layer perceptron fed with *tf-idf* and high-level features.

C.2 Video-based Emotion Recognition

For video-based emotion recognition, two different models according to their input nature were designed, namely: 1) the *Face model* that takes directly the face images as input, and 2) the *Facial Landmarks model*, which takes as input 68 key-points located around important facial components (i.e., eyes, nose, and mouth). The purpose of the *face model* is to learn and extract appearance information about facial expressions, which comprises the contour, shape, and texture of a face. The *facial landmarks* model explicitly encodes the geometric information about facial expressions.

C.2.1 Pre-processing

To feed our video-based emotion recognition models, a pre-processing step for face detection and facial landmarks localization is required. To do so, the multi-task CNN face detector [251] is first used for face detection and, then, the FAN's state-of-the-art deep learning based face alignment method is used for facial landmarks location [33]. The faces are then normalized, cropped, and resized to 96×96 pixels and the facial landmarks coordinates are also normalized by the face image size. According to Ekman [151], an expression lasts for 300 ms to 2 s. To keep the model simplicity, face and facial landmarks were both extracted from a sequence of frames corresponding to 300 ms. The video sequences of the OMG-Emotion corpus have an average frame rate of approximately 30 f/s, which results in a total of 9 frames as input. Video sequences with higher and lower frame rates are downsampled and upsampled, respectively.

C.2.2 Face model

The implemented face model is an end-to-end multi-input deep neural network. An overview of the network architecture is shown in Figure C.1. In particular, the neural network has an input-specific pipe for each frame of the input sequence. Each input-specific pipe is responsible for extracting a feature representation of each frame. These feature representations are then merged, followed by a sequence of fully connected layers (or dense layers) with ReLUs as nonlinearities. The output layer consists of a dense layer with 9 nodes: one for valence, another for arousal, and the remaining for the classification of the 7 categorical emotions (i.e., the 6 universal emotions from Ekman plus the neutral one). While the arousal distribution ranges between $[0,1]$, the distribution of valence varies between $[-1,1]$. Therefore, a sigmoid activation function is used in the arousal node, whereas a hyperbolic tangent is used in the valence one. The neurons of the categorical emotions have a softmax activation function.

For training the model, the goal is to minimize the following loss function:

$$\begin{aligned}\mathcal{L} = \quad & -ccc(\mathbf{y}_{arousal}, \hat{\mathbf{y}}_{arousal}) \; - \; \lambda \; ccc(\mathbf{y}_{valence}, \hat{\mathbf{y}}_{valence}) \\ & + \; \beta \; \mathcal{L}_{\text{categorical}}(\mathbf{y}_{emotion}, \hat{\mathbf{y}}_{emotion}),\end{aligned} \tag{C.1}$$

where $\lambda, \beta \geq 0$ are the weights that control the interaction of the loss terms. The first two terms of the loss function are defined to maximize the Concordance Correlation Coefficient (CCC) between the model arousal and valence predictions ($\hat{\mathbf{y}}_{arousal}$ and $\hat{\mathbf{y}}_{valence}$) and their corresponding ground-truth values ($\mathbf{y}_{arousal}$ and $\mathbf{y}_{valence}$), respectively. The CCC is defined as:

$$ccc(\mathbf{y},\hat{\mathbf{y}}) \;=\; \frac{2\,\rho(\mathbf{y},\hat{\mathbf{y}})\;\sigma_{\mathbf{y}}\;\sigma_{\hat{\mathbf{y}}}}{\sigma_{\mathbf{y}}^2\;+\;\sigma_{\hat{\mathbf{y}}}^2\;+\;(\mu_{\mathbf{y}}-\mu_{\hat{\mathbf{y}}})^2}, \tag{C.2}$$

where $\rho(\mathbf{y},\hat{\mathbf{y}})$ is the Pearson's Correlation Coefficient between the ground-truth labels and the model response, $\mu_{\mathbf{y}}$ and $\mu_{\hat{\mathbf{y}}}$ denote the mean of the ground-truth labels and the model predictions, respectively. $\sigma_{\mathbf{y}}^2$ and $\sigma_{\hat{\mathbf{y}}}^2$ are the corresponding variances.

The choice of the CCC as a loss term is motivated by its capability of explicitly demonstrating the model's ability to describe the expressions in a video as a whole, taking into consideration the contextual information [22].

The last loss term, $\mathcal{L}_{\text{categorical}}$, trains the model to predict the categorical emotions ($\hat{\mathbf{y}}_{emotion}$) given the ground-truth ($\mathbf{y}_{emotion}$) and corresponds to the categorical cross-entropy. Although the purpose of the OMG-Emotion challenge is not the prediction of the 7 categorical emotions, we use them as an extra supervision layer to regularize the entire learning process.

To work around the problem of training high capacity classifiers in small datasets, such as the one of the OMG-Emotion challenge, the weights of each input-specific stream of the network are shared and initialized with the weights of the VGG-Face network (see Figure C.1). The VGG-Face network [153] is based on the VGG16 architecture and trained on a very large-scale dataset (2.6M images, 2.6k people) for the task of face recognition. Since the VGG-Face was trained in a similar domain but on a much larger dataset, only the top fully connected layers of our model are fine-tuned during the first training epochs (50 epochs). Afterwards, the whole network is trained, with a smaller learning rate, a few more epochs (15 epochs).

The hyperparameters of the face model, including the weights of the loss function, the l_2 regularization coefficient, the number of dense layers and neurons per layer, were optimized by means of grid search and cross-validation. The best models on the arousal and valence prediction tasks were kept and ensembled by averaging their outputs. Details about the adopted ensemble procedure can be found below in section C.4.

C.2.3 Facial landmarks model

The facial landmarks model is topologically identical to the adopted face model, which is also trained to minimize the loss function defined in equation (C.1). The facial landmarks model consists of a multi-input neural net with input-specific streams for the facial landmarks coordinates of each frame. In particular, each input-specific pipe consists of a classical neural network with two hidden layers with shared parameters. On top of that, there is also a sequence of dense layers, followed by a final output layer topologically identical to the output layer of the face model (i.e., 9 output nodes with appropriate activation functions). The facial

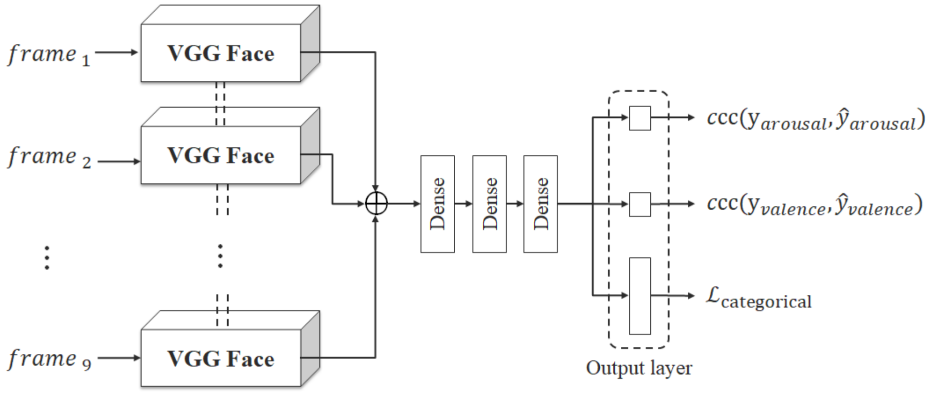

$frame_1$
$frame_2$
$frame_9$
VGG Face
VGG Face
VGG Face
Dense
Dense
Dense
$ccc(y_{arousal}, \hat{y}_{arousal})$
$ccc(y_{valence}, \hat{y}_{valence})$
$\mathcal{L}_{categorical}$
Output layer

Figure C.1: Face model architecture.

landmarks model is fed with the normalized facial landmarks coordinates (68 key-points × x and y coordinates) along with a set of temporal and geometric features computed from them. The temporal features attempt to encode how the input facial features changed over time. These features, computed between consecutive frames, include:

- The velocity of change, computed as the discrete 1st order derivate of the facial landmarks. It measures the rate of change of the per-frame facial features from one frame to the next.
- The acceleration of change of the per-frame facial landmarks. It is computed as the derivative of the corresponding velocities.

The geometric features are computed from the facial landmarks of each frame, individually. The extracted geometric features include:

- Relative x and y distances between each key-point and the center point of the face;
- Euclidean distance between each key-point and the center point;
- Relative angle between each key-point and the center point. The computed angles are corrected by the nose angle offset.

These features are then concatenated to form a feature descriptor of each input frame. The hyperparameters of the facial landmarks model were also optimized by means of grid search and cross-validation.

C.3 Text-based Emotion Recognition

C.3.1 Sequential deep model

The adopted sequential model is based on a simple deep recurrent architecture which is also trained to minimize the loss (C.1). The first layer is a 50-dimensional embedding layer, whose weights were initialized with pre-trained GloVe [155] word vectors and kept constant during training. The word embeddings are then fed through two cascaded LSTMs [79] of size 16. The final output of the recurrent part is applied to a fully connected layer which is structurally identical to the output layer of the face model (9 output nodes with approriate nonlinearities applied to each of them). The model was trained using Adam [97], with a learning rate of 10^{-3}, and an l^2 regularization coefficient of 10^{-4}. The relative weights λ and β of the loss function were cross-validated and the best models on the arousal and valence prediction tasks were kept and ensembled as described in section C.4.

C.3.2 Feature-engineering model

While end-to-end deep learning strategies are able to achieve state-of-the-art results on large corpus of data, the reduced size of the target dataset difficult the learning of robust models for these tasks. Therefore, a model based on high-level features that allow to embed domain knowledge was introduced.

In this sense, a Term-Frequency Inverse Document Frequency (*tf-idf*) descriptor from the text and the Part-of-Speech tags of the text was extracted. The vocabulary construction includes uni-, bi-, and trigrams.

Moreover, high-level features such as the followinf were extracted:

- Sentiment and Subjectivity scores: aggregated polarity, positive/neutral/negative words. We used the standard models from NLTK [25] and TextBlob [9] to extract these features.
- Number of tokens in the utterance transcripts.
- Number of stop-words in the utterance transcripts.
- Number of swear-words, masked in the dataset using asterisks (*).
- Number of negations (e.g. don't, not, wouldn't).

Both types of features (i.e. *tf-idf* and *high-level* features) were aggregated using a two-stream multi-layer perceptron following the architecture illustrated in Figure C.2. As done in

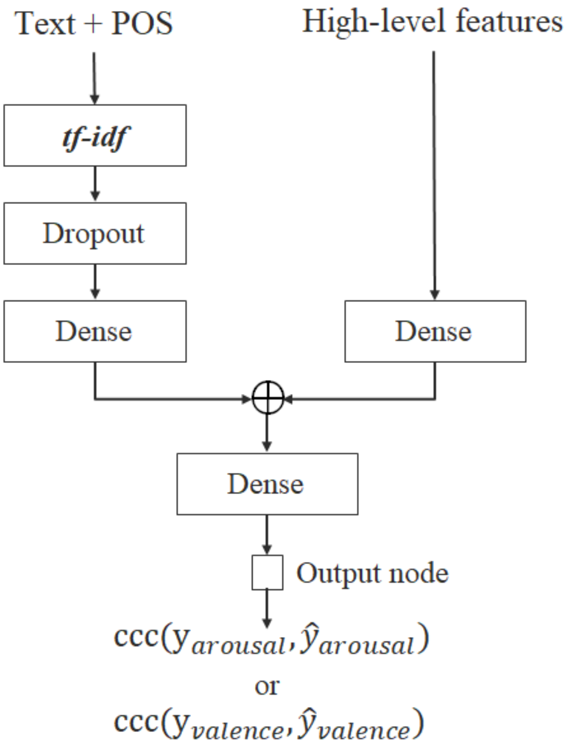

Text + POS
High-level features
tf-idf
Dropout
Dense
Dense
Dense
Output node
$ccc(y_{arousal}, \hat{y}_{arousal})$
or
$ccc(y_{valence}, \hat{y}_{valence})$

Figure C.2: Feature-engineering text model.

previous cases, the output activations are the hyperbolic tangent and sigmoid functions to project the network outcome to the target domain. The models were trained independently, using the CCC objective, defined in equation (C.2). In order to regularize the learning process of the *tf-idf* stream, which parameters grow linearly with the vocabulary size, dropout was used to simulate the stochastic absence of words in the input text. Furthermore, the test set distribution was considered in the computation of the inverse document frequency terms. This process is known as transductive learning [86].

C.4 Ensemble

Learning from multi-modal data is a challenging and compelling task, which is usually addressed using at least one of the following strategies: early modality fusion, in which the different modalities are merged in their original space and then fed through the classifier; intermediate fusion, where the modalities are projected and merged in a semantic space and this embedding is then used for classification; late fusion, where independent classifiers for each modality are designed and their predictions are combined via some form of model ensembling. For the given dataset, it is a bit impractical to implement an early fusion strategy, given the absence of text transcript for some of the videos in the training set. Moreover, the semantic level of both modalities is also quite different. Thus, if we opted for intermediate fusion, the classifier would be likely to rely mostly on the most represented modality (image data), wasting the useful information provided by the other one (text data). For these reasons, we decided to implement a late fusion procedure, where we compute a weighted average of the predictions of each classifier for each of the two target variables (arousal and valence). The weights for each prediction are given by the CCC score in the validation set of each model and for each variable. This averaging procedure reduces variance in the ensemble classifier, while preserving the relative importance of each individual model.

C.5 Experimental Evaluation

The experimental evaluation of the adopted emotion recognition methodologies was performed using the OMG-Emotion Dataset [21]. This dataset is composed of 420 relatively long emotion videos, collected from a variety of Youtube channels. The videos are divided into clips based on utterances, each of them annotated with arousal and valence values and a categorical label. The dataset as part of the OMG-Emotion competition has a strict evaluation protocol with predefined training, validation, and test sets. In particular, the training, validation and test sets comprise a total of 2442, 617, and 2229 video utterances, respectively. Since at the development phase of the models we do not have access to the test set labels, the results are reported on the validation set.

Table C.1 compares the performance of the implemented models with the baseline methods of the OMG-Emotion challenge. The results are reported in terms of CCC and MSE for both arousal and valence target variables.

A first observation, regarding the implemented approaches, is that the best arousal and valence results are achieved by the facial landmarks model and the feature-engineering text model, respectively. However, the most interesting observation is that the adopted multimodal ensemble strategy promotes a significant overall improvement in both arousal and valence results. These results clearly demonstrate the complementarity of both modalities. Finally, it is important to stress that our ensemble model clearly outperforms the four baselines on the validation set of the OMG-Emotion challenge.

C.6 Summary

This Appendix reports our emotion recognition methodology for the OMG-Emotion challenge. The implemented methodology is an ensemble of different models from two distinct

Table C.1: Results on the OMG-Emotion validation set: (first block) baseline methods, and (second block) implemented methods.

Method	Arousal		Valence	
	CCC	MSE	CCC	MSE
Vision - Face Channel [22]	0.12	0.053	0.23	0.12
Audio - Audio Channel [22]	0.08	0.048	0.10	0.12
Audio - OpenSmile Features [6]	0.15	0.045	0.21	0.10
Text [6]	0.05	0.062	0.20	0.12
Face model	0.18	0.067	0.32	0.16
Facial landmarks model	0.22	0.057	0.27	0.18
Feature-engineering text model	0.14	0.064	0.33	0.128
Sequential text model	0.11	0.066	0.32	0.18
Ensemble	**0.23**	**0.050**	**0.38**	**0.12**

modalities, namely video and text. Experiments results demonstrate that our ensemble model clearly outperforms the current baseline of the OMG-Emotion competition.

The final results of the 2018 OMG-Emotion Recognition Challenge are reported at: https://www2.informatik.uni-hamburg.de/wtm/OMG-EmotionChallenge/. Unfortunately, our team, the so-called EMO-INESC, was not able to reach the top-3 awarded teams. These results may be explained by the absence of the audio information in our methodology. Audio information proved to be crucial especially for the prediction of the arousal dimension. For instance, all the top-3 teams integrated audio information, along with video and text, into their methodologies.

Appendix D

Hand-crafted Facial Expression Recognition

This Appendix depicts all the implemented hand-crafted FER methodologies and their corresponding results on the CK+ database. In particular, several appearance-based methods as well as a geometric-based approach were implemented.

D.1 Geometric-based FER

As illustrated in Figure D.1, the implemented geometric approach is based on features computed from the facial key-points[1], such as:

- Relative *x* and *y* distances of each key-point to the center point;
- Euclidean distance of each key-point to the center point;
- Relative angle of each key-point to the center point corrected by nose angle offset (i.e., the angle between the nosebridge and the vertical image axis).

The center point is obtained by computing the average *x* and *y* coordinates across all detected facial key-points. The extracted features are then concatenated to form a geometric feature descriptor. Finally, this feature descriptor is fed into a multi-class SVM for recognition purposes.

[1] The coordinates of the facial landmarks are automatically obtained using the robust facial landmark detection approach proposed by Bulat *et al.* [33].

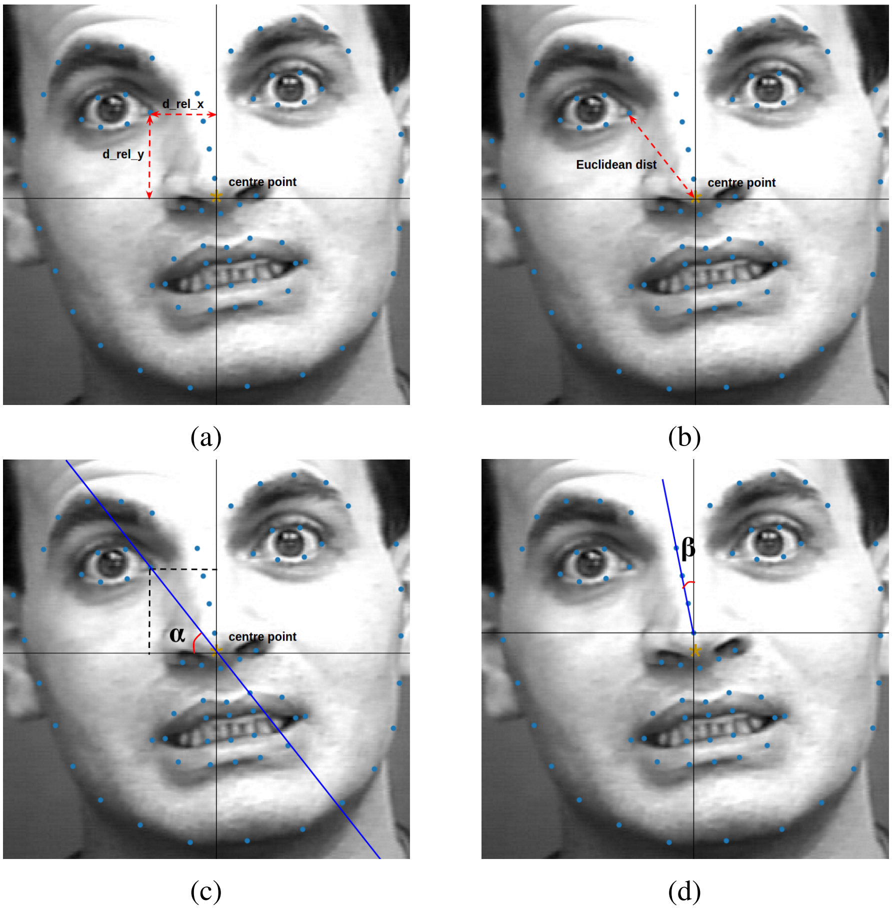


Figure D.1: Illustration of the geometric-based feature extraction process: (a) x and y relative distances between each facial key-point and the center point, (b) Euclidean distance between each key-point and the center point, (c) relative angle between each key-point and the center point corrected by nose angle offset (d).

D.2 Appearance-based FER

The implemented appearance-based FER methods rely on two commonly used techniques for texture classification, namely Gabor filter banks and LBP. Regarding the Gabor filters approach, a bank of Gabor filters with different orientations θ, frequencies f and standard deviations σ was first created. Afterwards, the input images are convolved (or filtered) with the different Gabor filter kernels, resulting in several image representations of the original image. The mean and variance of the filtered images (image representation) are then used as descriptors for classification. In particular, different Gabor-descriptors were extracted, according to the degree of local information:

- **[Gabor-global]**: This feature descriptor consists in the concatenation of global mean and variance values of each Gabor representation.

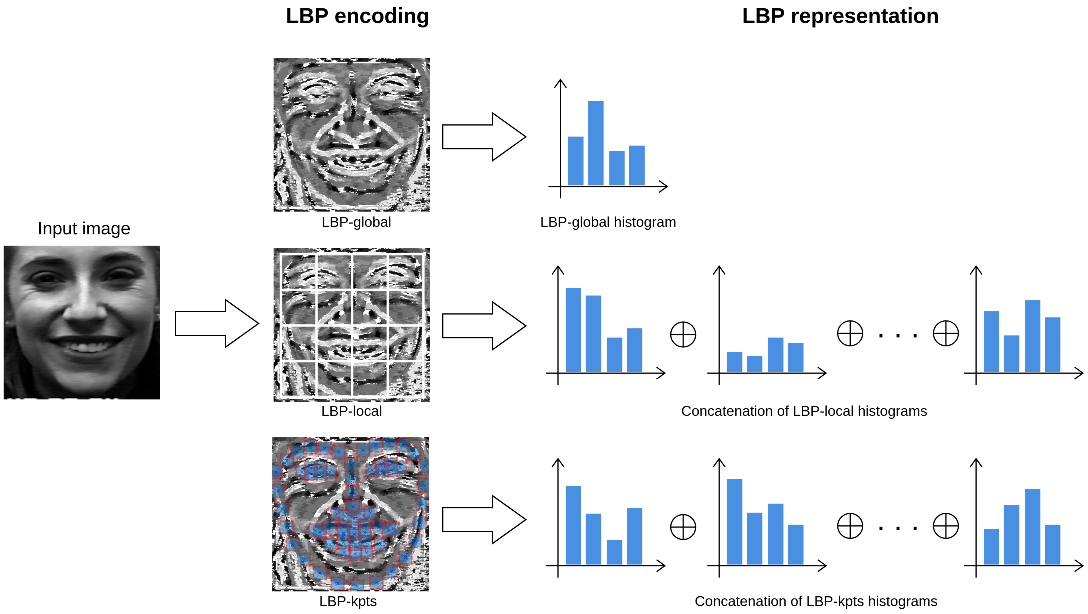


Figure D.2: Illustration of the implemented LBP-based FER methodologies.

- **[Gabor-local]**: The Gabor representations are divided into a grid of cells. The mean and variance of each cell are computed and, then, concatenated to form the feature vector.

- **[Gabor-kpts]**: It requires information about the facial key-points coordinates. In particular, the mean and variance of the Gabor representations are computed locally in a neighborhood around each facial key-point.

Then, these feature descriptors are fed into an SVM for expression classification.

Regarding the implemented LBP-based approach, the LBP representation of the input image is first computed and then used to build a histogram of the LBP patterns. For an extra level of rotation and luminance invariance, only the uniform LBP patterns [117] were extracted. Similarly to the Gabor-based approach, different LBP feature descriptors were computed:

- **[LBP-global]**: This feature vector consists in the global histograms of the LBP patterns.

- **[LBP-local]**: The LBP representations are divided into a grid of cells. Then, the histograms of the LBP representations of each cell are computed and, then, concatenated to form the feature vector.

- **[LBP-kpts]**: The histograms of the LBP representations of the region around each facial key-point are concatenated to form the feature vector.

For recognition purposes, the LBP-based features descriptors are also used as input into a multi-class SVM.

Moreover, different combinations of these methods were also performed as well. That is, LBPs were applied to Gabor representations and geometric features were concatenated with LBPs features. It is important to stress that a Principal Component Analysis (PCA) is performed on the extracted features to reduce the dimensionality of the feature space.

D.3 Experimental Evaluation

The experimental results of the implemented hand-crafted FER methodologies on the CK+ database is depicted in Table D.1. The results are reported in terms of average classification accuracy and were obtained using the same evaluation protocol as previously described in Chapter 9. Regarding the implementation details of the methods, a grid cell size of 10×10 pixels, for both Gabor- and LBP-local approaches, was used. The local window size around the facial key-points of the Gabor-kpts and LBP-kpts methods was set to 16×16 pixels. Regarding the LBP-based approaches, a neighborhood and a radius of 8 was chosen. The Gabor filter bank comprises a total of 16 filters with different values of standard deviation $\sigma : \{1,3\}$, orientation $\theta : \{0, \frac{\pi}{2}, \frac{\pi}{4}, \frac{3\pi}{4}\}$, and frequency $f : \{0.05; 0.25\}$. The number of components kept in the PCA transformation was chosen to retain 95% of the explained variance.

Table D.1: Average classification accuracy of the implemented hand-crafted FER methods on the CK+ database. Bold number indicates the best method with the highest average classification accuracy.

Method		Average Accuracy (%)
Geometric-based		79.76
Appearance-based	LBP-global	46.35
	Gabor-global	43.11
	LBP-global + Gabor-global	42.25
	LBP-local	67.82
	Gabor-local	65.53
	LBP-local+ Gabor-local	69.12
	LBP-kpts	72.41
	Gabor-kpts	70.13
	LBP-ktps + Gabor-kpts	77.26
Geometric- + Appearance-based	Geometric-based + LBP-ktps + Gabor-kpts	**79.76**