NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

AFONSO FILIPE ESTEVES QUINAZ

Master in Computer Science

FACIAL EXPRESSION RECOGNITION IN PORTUGUESE SIGN LANGUAGE

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon

month, year

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF
COMPUTER SCIENCE

FACIAL EXPRESSION RECOGNITION IN PORTUGUESE SIGN LANGUAGE

AFONSO FILIPE ESTEVES QUINAZ

Master in Computer Science

Adviser: João Magalhães
Full Professor, NOVA University Lisbon

Co-adviser: Sofia Carmen Faria Maia Cavaco
Associate Professor, NOVA University Lisbon

Examination Committee

Chair: Name of the committee chairperson
Full Professor, FCT-NOVA

Rapporteur: Name of a rapporteur
Associate Professor, Another University

Members: Another member of the committee
Full Professor, Another University

Yet another member of the committee
Assistant Professor, Another University

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
month, year

Facial Expression Recognition in Portuguese Sign Language

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.1.5) [38].

To Luís Miguel Clemente. Your memory continues to inspire me every day, and I dedicate this work to you with all my love.

Acknowledgements

I would like to express my deepest gratitude to Professor João Magalhães for his invaluable guidance and support throughout this journey. His insights and encouragement have been crucial in helping me navigate both the challenges and successes of my academic path, and I am sincerely grateful for his wisdom and mentorship. I also extend my thanks to FCT for the academic experience they provided during my bachelor's and master's degrees. Although there were tough moments, these challenges have shaped me into the person I am today, and I am grateful for the opportunities and growth that FCT has facilitated.

To my university colleagues—António C., Ricardo L., Rafael G., Vasco C., and João E., thank you for the countless hours of work, late nights on projects, and weeks of study. Your support and determination made this journey easier, and I will always treasure the memories of what we've accomplished together.

To my dear friends and cousins—Américo A., Diogo C., Gonçalo D., Tomás S., Tiago C., Alberto Q., António Q., and Francisco Q.—your friendship has been a constant source of strength and joy. Whether through a quick call or a longer conversation, you always manage to lift me and improve my day. I will always cherish the memories we've created, and I am forever thankful for your support.

Finally, to my family: To my father, thank you for the love, for trusting me and opening every door possible, always encouraging me to strive for the best in everything. To my mother, who every day gives her heart to make me feel safe and loved, I am endlessly grateful. To my older brother my best friend, your determination, support, and strength have shaped me into who I am today. You are a role model, and I know I haven't thanked you enough for that. To my younger brother, my infinite source of joy, you make every day better. Thank you for your kindness, your company, and the happiness you bring into my life. And to Francisca, thank you for being with me through every kind of day—the good, bad, average, and everything in between. Your love and your constant push for me to do better mean the world. I love you all and am eternally grateful for everything.

Abstract

Millions of individuals rely on Sign Language as their primary mode of communication each day. Despite this, the community encounters substantial barriers in interacting with non-Sign Language users, contributing to their exclusion within society. This situation brings issues even in the more basic aspects of life, such as obtaining medical services or pursuing educational opportunities, highlighting the pressing challenges they face.

Historically, the majority of research in this field has predominantly concentrated on manual gestures, often overlooking the critical role of facial expressions in conveying grammatical nuances. Our work seeks to bridge this gap by evaluating the efficacy of various methodologies across the diverse spectrum of Sign Languages globally established to Portuguese Sign Language.

In this thesis, we implement the first facial recognition models (CNN and SqueezeNet) for Portuguese Sign Language, marking a significant advancement in the domain of LGP communication.

This framework holds the potential to motivate future investigations within this linguistic area, as well as to enhance the integration of facial expressions with manual signs. This integrated approach aspires to develop a comprehensive model capable of significantly enriching the communication landscape for the deaf and hard-of-hearing community, thereby making a substantial impact on their lives.

Keywords: Sign Language Recognition, Facial Expressions, Portuguese Sign Language,

Resumo

Milhões de pessoas dependem da Língua Gestual como principal meio de comunicação. Apesar disso, esta comunidade enfrenta barreiras significativas na interação com quem não usa a linguagem gestual, contribuindo para um aumento de exclusão destes indivíduos na sociedade. Esta situação cria adversidades em aspectos essenciais da vida, como a obtenção de serviços médicos ou a prossecução de oportunidades académicas, evidenciando os desafios que enfrentam.

Historicamente, a maioria das pesquisas neste campo concentrou-se predominantemente nos gestos manuais, muitas vezes subestimando o papel crítico das expressões faciais na transmissão de nuances gramaticais. O nosso trabalho procura colmatar esta lacuna, avaliando a eficácia de várias metodologias em todo o diverso espectro de Línguas Gestuais globalmente estabelecidas, desta vez para a Língua Gestual Portuguesa.

Nesta tese, implementamos os primeiros modelos (CNN and SqueezeNet) de reconhecimento de expressões faciais para a Língua Gestual Portuguesa, marcando um avanço significativo nesta mesma linguagem.

A estrutura que propomos detém o potencial de motivar futuras investigações nesta área linguística, bem como de aprimorar a integração de expressões faciais com sinais manuais. Esta abordagem integrada aspira a desenvolver um modelo abrangente capaz de enriquecer significativamente a paisagem comunicativa da comunidade surda e com deficiência auditiva, impactando substancialmente as suas vidas.

Palavras-chave: Reconhecimento de Linguagem Gestual, Expressões Facias, Linguagem Gestual Portuguesa,

Contents

List of Figures

List of Tables

Acronyms

ASL American Sign Language *(pp. 4, 6, 23, 24)*

AU Action Units *(pp. 6, 8, 21, 22)*

CLIP Contrastive Language–Image Pre-training *(pp. 18–21, 38)*

CLIPER Contrastive Language–Image Pre-training for Facial Expression Recognition *(pp. 19–21, 38)*

CNN Convolutional Neural Networks *(pp. 7, 11–16, 21–25, 35, 36)*

FER Facial Expression Recognition *(pp. 2–4, 8, 9, 11, 13–15, 34, 35, 37)*

GSL German Sign Language *(pp. 4, 25)*

LIBRAS Brazilian Sign Language *(pp. 2, 4, 6, 14, 23, 38)*

LSTM Long Short-Term Memory Networks *(pp. 11, 13, 14, 21, 23, 24)*

PSL Portuguese Sign Language *(pp. 2, 3, 5, 28, 38)*

RNN Recurrent Neural Networks *(pp. 13, 14)*

ROI Region of Interest *(pp. 7, 34)*

SLR Sign Language Recognition *(pp. 1, 3)*

1 Introduction

In the evolving field of Sign Language Recognition (SLR), many advanced frameworks often fail to effectively replicate the depth and authenticity of human expression [46, 29]. This limitations arises from few different aspects. Firstly, the existing Sign Language Corpus Annotations are very limited, making access to proper datasets very hard. Secondly the annotation of them is not a straight forward process, as the translation is not direct from Sign Language to written text, involving few more classes corresponding to different parts of the body or combinations of it. This limitation mainly arises from a primary focus on manual signs, neglecting the essential roles that facial expressions play in delivering emotion, grammar, and meaning.

1.1 Sign Language

Sign Language, serving as the primary mode of communication for over 70 million individuals across the globe [50], has received significant amounts of attention compared to recent years. Advancements in artificial intelligence and image recognition technologies have propelled the development of Sign Language recognition systems, making these efforts increasingly effective and promising. This progress opens new avenues for enhancing communication accessibility and the needed inclusivity for the deaf and hard-of-hearing communities.

However, contrary to common misconceptions, Sign Language is a multi-modal language. It encompasses more than just hand gestures and body movements; facial expressions play a crucial role in conveying the nuances and full meaning of the message.

1.1.1 Evolution of Sign Language Recognition

The computational study of SLR, which began in the late 80s [48], initially focused on estimating hand position and orientation through sensorial gloves [46]. This field has seen a significant impact in research and development with the advent of deep learning. Various

deep learning architectures such as Multilayer Perceptrons (MLP)[20, 23], Recurrent Neural Networks (RNN) [56, 53], Generative Networks [34] and more have been explored to enhance the recognition of sign language [46]. In the realm of deep learning, the predominant focus has been on hand detection, tracking, pose estimation, gesture recognition, and pose recovery [46]. This concentration on hand-related modalities inadvertently led to an undervaluation of facial expressions, which impacted the overall effectiveness of recognition systems.

It's only in more recent years that the importance of Emotional Facial Expression Recognition (EFER) in sign language has been recognized. The increase in research in this area underscores the vital role those facial expressions play in adding context and meaning to sentences. Studies now demonstrate that incorporating facial expression data significantly improves the performance of recognition models, leading to more accurate and meaningful interpretations of sign language [17].

1.1.2 Facial Expressions

Sign language is a multifaceted spatiotemporal language that extends beyond the commonly perceived hand movements and placements [50]. It incorporates a broad spectrum of elements, making it a richly expressive form of communication. These elements include the positioning, movement, and placement of hands, as well as nuanced facial features like muscle intensity, rotation, eye gaze, eyebrow movements, head orientation, and other bodily gestures [46].

Facial expressions play a pivotal role in sign language, often altering or enhancing the meanings conveyed by hand gestures. These expressions are integral in expressing a range of linguistic functions, such as indicating yes/no questions, expressing doubt, negation, affirmation, conditional clauses, WH-questions, and emphasizing focus or relative clauses, as outlined in [16, 18, 29, 20].

A notable example of the complexity and importance of facial expressions can be observed in Brazilian Sign Language (LIBRAS). In this language, facial expressions are categorized into three distinct grammatical effects: Grammatical Facial Expression for Sentence (GES), Grammatical Facial Expressions of Intensity (GEI), and Grammatical Facial Expressions of Distinction (GED). Each category plays a specific role in the grammatical structure and interpretation of the language, highlighting the depth and intricacy of facial expressions in sign language communication [29].

1.2 Objective

The primary goal of this thesis is to develop a Facial Expression Recognition (FER) model specifically for Portuguese Sign Language (PSL). This model aims to bridge a crucial gap in

sign language recognition by focusing on the often under looked aspect of facial expressions, which are integral for fully understanding and interpreting PSL.

The methodology involves a two-stage approach:

- The first stage is dedicated to data preparation, where facial expressions and landmarks will be extracted and annotated from a dataset of PSL videos. This step is essential for creating a robust dataset that accurately represents the range of facial expressions used in PSL.
- The second stage is the development of the FER model itself. With the pre-extracted and annotated data, the model will be trained to predict the grammatical meanings of facial expressions in PSL. The model's ability to accurately interpret these expressions will add significant value to the understanding of PSL, particularly in how facial nuances contribute to the language's grammar and overall meaning.

Key aspects of this project include data extraction and annotation from PSL videos, and the development of a model that effectively correlates facial expressions with their grammatical significance in PSL. This research aims to make a meaningful contribution to enhancing communication tools for the deaf and hard-of-hearing communities, particularly those using Portuguese Sign Language.

1.3 Applications and Impact

The development of this FER model for PSL holds significant potential for a variety of impactful applications. One of the primary applications is to provide a deeper understanding of the role and effectiveness of facial expressions in PSL. This insight is crucial, as it will pave the way for the enhancement of existing SLR models, encouraging them to incorporate facial expressions as a fundamental component of sign language interpretation.

Furthermore, this models can be integrated with other SLR systems that focus on different spectrums of sign language, like hand gestures or body postures. By combining these models, there's an opportunity to create a more comprehensive and accurate SLR system. Such a system would not only recognize the manual components of sign language but also the non-manual elements like facial expressions, offering a holistic understanding of the language.

1.4 Document Structure

- **Chapter 2**: Background and Related work is the topic we will explore in this chapter. Our project is multi-faceted, therefore we explore what is Portuguese Sign language and more importantly the effect and studies on facial expressions. Secondly, to support the

input for our models we dive deep into facial expression, AU, and landmark extraction models, which are imperative for the pre-processing stage of our proposed framework. Thirdly we explore some research done in the field of FER models, more precisely CNNs and Vision Transformers. Lastly, we explore case studies made to three of the most used sign languages (LIBRAS, American Sign Language (ASL), and German Sign Language (GSL)) and what the advances in the recognition of sign languages and more specifically the facial expressions impact and recognition.

- **Chapter 3**: This chapter explains the dataset and methodology used in the project. It describes the dataset collection, structure, and the steps involved in preparing the data. The chapter also discusses how facial features were extracted using techniques such as Hermite Curve Generation and Data Augmentation. Lastly, it covers the preliminary analysis of the results, where different models were tested and their initial performance evaluated, providing insights into the effectiveness of these approaches.

- **Chapter 4**: This chapter focuses on the implementation and evaluation of various models for facial expression recognition in PSL. It provides a detailed explanation of the models used, including CNN, SqueezeNet, and LSTM. The chapter discusses the architecture of each model, the training process, and the results achieved. Additionally, it offers a comparison of the models, highlighting their strengths and weaknesses in recognizing facial expressions for PSL.

- **Chapter 5**: This chapter summarizes the main findings of the study. It reflects on the performance of the facial expression recognition models and the challenges faced during the research, such as data limitations and model accuracy. The chapter also suggests directions for future research, including the need for larger datasets, the use of more advanced models, and exploring multimodal learning to improve PSL recognition systems.

2

Background and Related Work

2.1 Portuguese Sign Language

The PSL, is the primary way of communication for the deaf and hard-of-hearing Portuguese community. Formally originated in 1823 with the help of Casa Pia Lisboa[1] [2] but only considered official by the Constitution in [47], it consists like many others, of the use of a combination of human features primarily hands but also movements, facial expressions, and body postures. Despite many studies [8, 12, 11, 19, 40, 4] in PSL, there is not yet a strictly defined grammar[39].

Of the studies done, with hands' major impact on sign language, current Sign language recognition models do not explore and lose performance by underrating the impact that facial expressions can convey.

2.1.1 Importance of Facial Expressions in Sign Language

As Humans know, facial expressions are present to us in an extra form to convey emotions rather than express them solely from words. In fact, human facial expressions sometimes are even more true than the words being spoken. A study, [15], reinforces that the range of facial expressions covers a wide spectrum from universal messages, such as a feeling of surprise to a less global one as "Hello".

Despite cultural and regional Variations, facial expressions are crucial across many Sign Languages. They are a strong source of grammar, contextual clarity, emphasis intensity, demonstration of emotions, and regulation of interactions. PSL is not an exception and it is pointed to have a similar representation. On the one hand, it serves as morphologic additive such as giving a degree of size and intensity to substantives. On the other hand, it marks syntactically the construction of a sentence with negations and interrogations. [50, 21]

[1] http://https://casapia.pt/

In Figure 2.1 we can observe the word "Easter" being represented with the use of hands, however, due to different facial representations one of them means "almond".

Table 2.1 on the other hand gives us a brief understanding of the semantic functions and details that non-manual signs can affect the LIBRAS. It demonstrates the impact of creating WH, Y/S, or doubt questions, topics, Negations, Assertions, Conditional and relative clauses, and so on.

Furthermore, studies [50, 14, 45] on facial expression variation across regions point out that indeed there are universal facial expressions, for example, raising the eyebrow is considered to mean "I want to know more about this" and in sign languages, it usually serves to construct yes-no questions. However, for different cultures, there are a few changes. For example in ASL, exists the adverbial "th" (carelessly), which is represented by a facial expression (sticking the tongue between closed lips and tilting the head), while not all languages have labeled that complex set of Action Units (AU)s, which represent simple and specific facial muscle movements and is further explored in chapter 2.2.2.

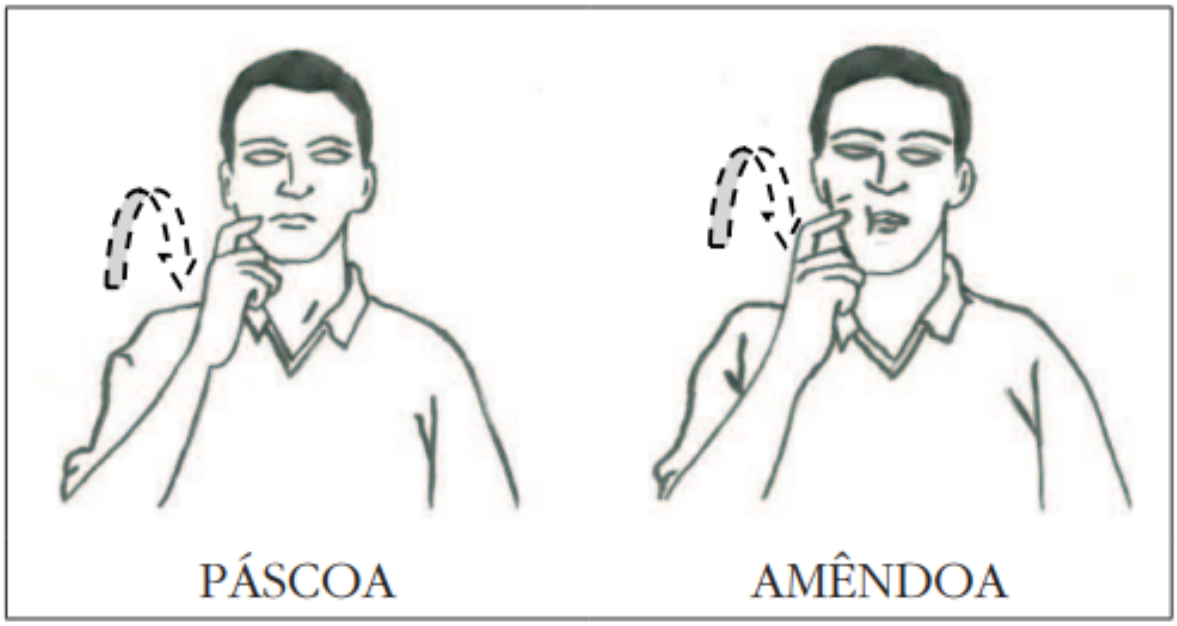


Figure 2.1: Impact of facial expressions [29].

2.2 Sign and Facial Recognition Models

2.2.1 Feature Extraction

In this thesis, we emphasize the significance of feature extraction in acquiring essential data from our primary sources, which are video datasets. Within the realm of Sign Language

AFE	Left / Right eyebrow raised; Raised eyebrows and wide-open eyes; Raised eyebrows, wide-open eyes and open mouth; Slightly closed eyes and crooked mouth up; Smile with apparent teeth; Smile with apparent teeth and open mouth; Lowered eyebrows and crooked mouth down; Frown and contraction of the upper lip; Crooked mouth up laterally.		
GFE	GES	WH	Brief and upward movement of the head and frown.
		YN	Brief and upward movement of the head and raised eyebrows.
		DQ	Frown, slightly closed eyes and contracted lips.
		T	Brief upward and forward movement of the head, raised eyebrows, open mouth, projected lips; Quick nod, brief upward movement and wide open eyes; Quick nod, brief upward movement, raised eyebrows and wide open eyes; Quick nod, brief upward movement, raised eyebrows, open mouth and projected lips.
		N	Crooked mouth down; Quick nod, frown and crooked mouth down; Head balancing sideways.
		A	Balance back and forth of the head.
		CC	Brief and upward movement of the head and raised eyebrows.
		F	Brief upward and forward movement of the head, raised eyebrows, open mouth, projected lips; Quick nod, brief upward movement and wide open eyes; Quick nod, brief upward movement, raised eyebrows and wide open eyes; Quick nod, brief upward movement, raised eyebrows, open mouth and projected lips.
		RC	Raised eyebrows.
	GEI		Frown < Frown and Slightly closed eyes; Inflated cheeks and semi-open mouth < inflated cheeks, semi-open mouth and frown; Contracted cheeks and frown < contracted cheeks; Contracted lips and frown < contracted lips; Projected lips and frown < projected lips; Open mouth and frown < open mouth; Crooked mouth up< Smile with apparent teeth; Quick nod < balance back and forth of the head;
	GED		Left eye closed; Inflated cheeks; Only right cheek inflated;

Semantic Functions: AFS- Affective Facial Expression; GFS- Grammatical Facial Expression; WH- WH-question; YN-Yes/No questions, DQ- Doubt question, T- Topic, N- Negation, A- Assertion, CC- Conditional Clause, F- Focus, RC- Relative Clause.

Table 2.1: Libras non-manual signs classification.

Recognition, numerous methodologies for data pre-processing have been explored, highlighting its critical importance. An illustrative study [16] focusing on the recognition of action units in Brazilian Sign Languages showcases the impact of different Convolutional Neural Networks (CNN) architectures and pre-processing techniques. Initially, the study utilized raw images as input, yielding an average accuracy and F1 score of 0.48 and 0.49, respectively. In a subsequent strategy, the models were trained using the Region of Interest (ROI), specifically the subject's face, rather than raw images. This approach alone led to an approximate 20 percent increase in both metrics, achieving average accuracy and F1 scores of 0.68 and 0.66, respectively. The research then introduced an innovative representation of action units and landmarks by employing a Hermite Curve. This mathematical technique, which generates

smooth curves through specified points, was applied to facial landmarks to enhance pattern recognition. By integrating a Heatmap filter to improve image contrast and combining it with the rest of the image, they successfully elevated the results to an average accuracy of 0.75 and an F1 score of 0.77.

Sign Language focuses on extracting two key types of data from videos of individuals using sign language: hand positioning and facial movements. Thus, exploring models developed for hand and pose feature extraction as well as FER becomes crucial. Object detection, a core aspect of computer vision, has been explored for over two decades and spans various applications, including face recognition, and pedestrian and vehicle detection, among others [55].

2.2.2 Facial Landmarks and Action Units

Facial landmarks refer to distinct and specific points on the human face, such as the corners of the lips, the tip of the nose, and the border of the eyebrows or eyes. These points play a pivotal role in tracking and analyzing facial movements, providing a foundation for various applications like emotion recognition and face morphing. Facial landmarks and AU detectors use fixed referenced points in the in the face as the landmarks. For example, in 2.2.a) we can observe the 68 landmarks registerd, and in this cased the 17th landmark will always be the furthest point to the right of the right eyebrow. Consequently AUs correspond always to the difference of the same landmarks, for example the AU represented by 21 and 17 in 2.2.b).

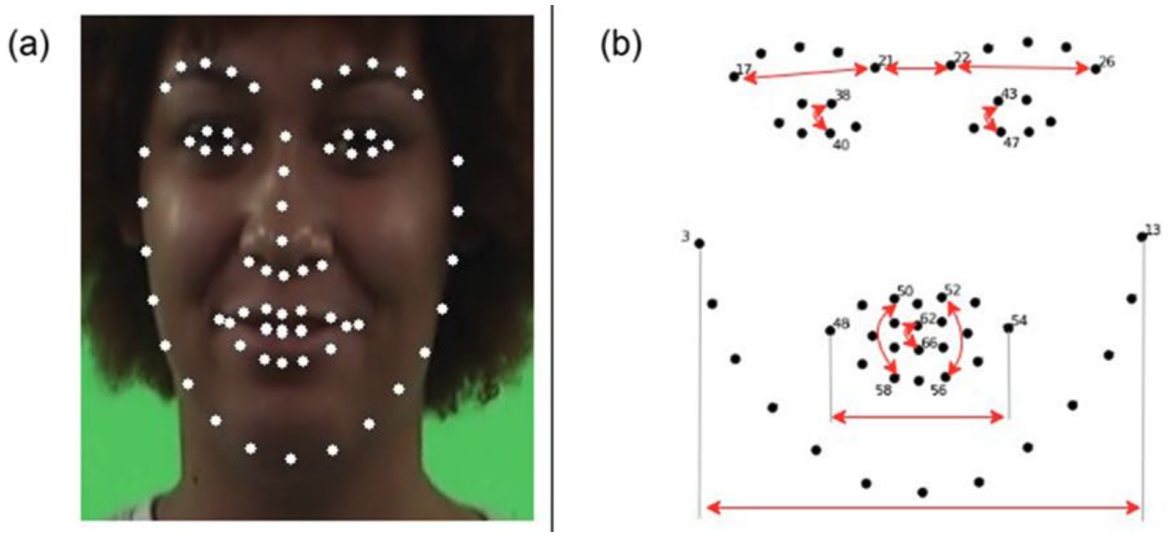


Figure 2.2: A) Facial Landmark representation in white dots B) Action Unit Representation in Red Arrows[29].

On the other hand, Action Units such as AU1 (inner brow raiser) or AU12 (Lip corner puller), as described by Friesen and Paul Ekman in their seminal 1978 research, are a systematic

way to represent the changes in intensity of facial muscle activities. By understanding these Action Units, we gain deeper insights into the intricate dynamics of facial expressions and the emotions they convey. Action Units are considered to be the most indicator of facial movements, which is derived from FACS, Facial Action Coding System, which describes facial movements in anatomical terms [41] while one AU is the representation of a single facial movement also known as a muscle movement. FACS is described to be the most useful, used, and also the most influential standardization for the recognition, judgment, and comprehension of facial behavior and emotions. [59].

2.2.3 Facial Feature Extraction Models

In recent years, the use of technological tools for understanding facial expressions through the analysis of facial features and movements has gained significant prominence. Tools such as Py-Feat [10], OpenFace, and others stand out in this domain. These tools facilitate the observation and study of subtle changes in facial expressions, offering invaluable insights into human emotions and reactions. Their primary function revolves around identifying facial landmarks and action units associated with specific facial movements.

OpenFace is often noted for its superior performance in this area. However, its effectiveness varies with different angles, a limitation not observed in tools like Py-Feat, which maintain consistent performance across various angles (as illustrated in figure 2.3). Despite advancements, there is still a scarcity of automated, open-source tools for FER, with notable examples including OpenFace[2], AFARtoolbox [3] , and Py-Feat[4].

FaceReader, another tool in this category, is limited to predicting 20 specific action units and was predominantly trained using data from Japanese participants, potentially limiting its cross-cultural applicability.

It is important to note that these tools differ not only in accuracy but also in the types of detection outputs they provide. For instance, OpenFace detects facial landmarks, head pose, 18 facial Action Units (AUs), and eye gaze. Conversely, Py-Feat employs a variety of state-of-the-art algorithms, including RetinaFace for face detection and MobileNets for landmark detection, and can recognize 20 AUs using a random forest approach.

The development of facial feature extraction models is a central focus in computer vision. These models capture various facial characteristics, such as boundary boxes, landmarks, and action units. RetinaFace, for example, is a comprehensive single-stage face detector that incorporates face detection, alignment, pixel-wise parsing, and 3D dense correspondence regression, achieving an impressive 91.4% average precision on the Wider Face hard subset. Its

[2] https://github.com/TadasBaltrusaitis/OpenFace
[3] https://github.com/AffectAnalysisGroup/AFARtoolbox
[4] https://py-feat.org/

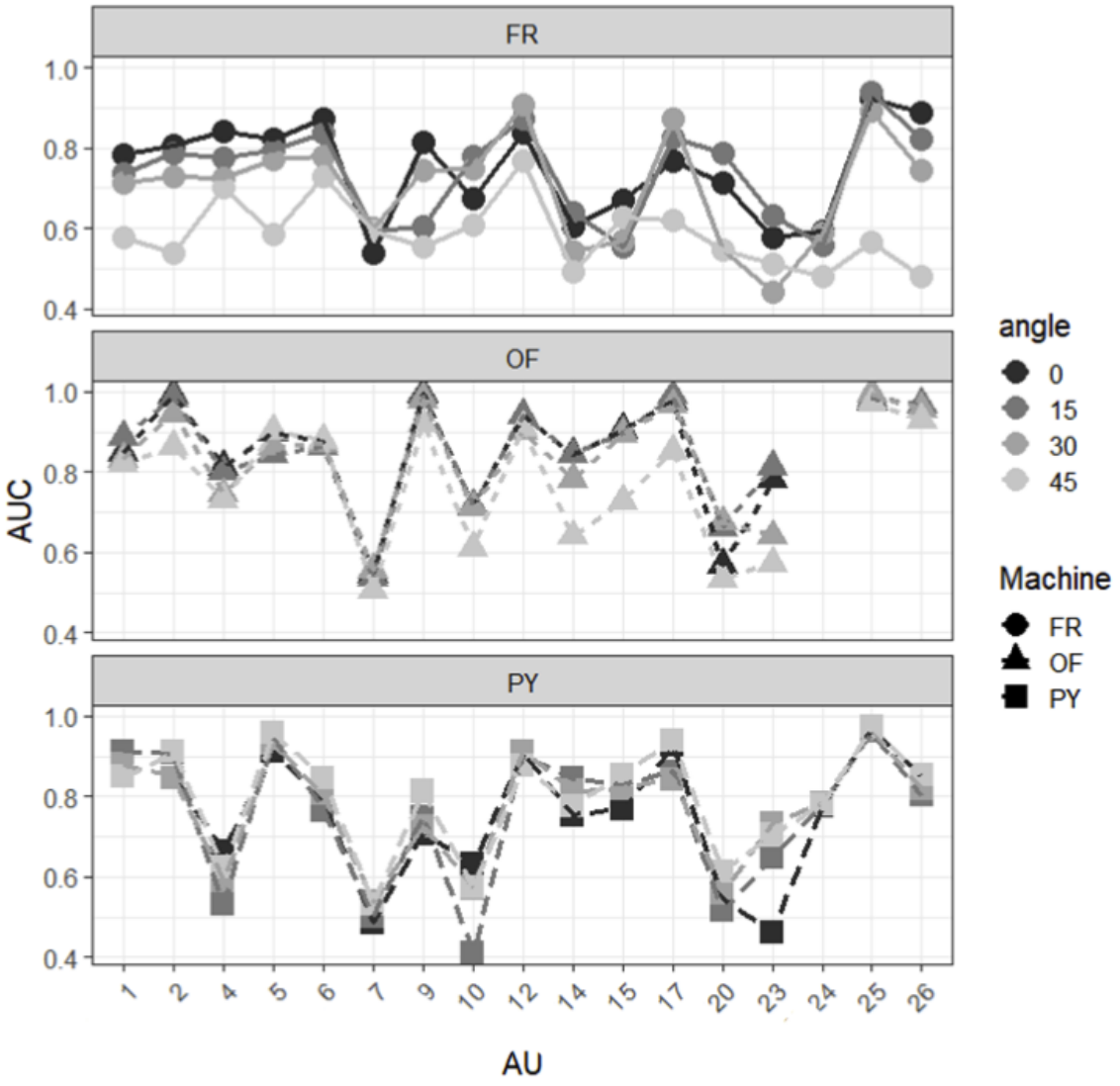


Figure 2.3: Average area under the curve (AUC) values for action units (AUs) predicted by FR: FaceReader; OF: OpenFace; and PY: Py-Feat[41].

multi-task learning strategy, which predicts facial landmarks, face score, and per-facial-pixel 3D position, allows for detailed and accurate facial analysis [31] .

Another significant model is the MTCNN (Multi-task Cascaded Convolutional Networks), which has recorded an 85.7% accuracy score on the same subset. MTCNN's efficient strategy involves smaller models generating candidate bounding boxes, followed by a three-layer network for precision tuning, making it quicker and more accurate in face detection [55] .

In the realm of landmark detection, the PFLD (Practical Facial Landmark Detector) is notable for tackling challenges like facial variations and data imbalance while maintaining

model efficiency. Its architecture combines MobileNet [9, 1] blocks with an auxiliary network for precise geometric estimation, enhancing its ability to detect facial landmarks crucial for interpreting facial expressions in sign language [25].

Studies [17, 16] have shown promising results when using facial feature extraction models to enrich input data for encoders, demonstrating improved performance compared to simple CNNs. This indicates the potential of these advanced models in enhancing the accuracy and effectiveness of tools like Py-Feat, which are already widely used and acclaimed for their capabilities.

2.3 Facial Expressions Recognition (FER)

FER is an essential and challenging component of understanding human interactions and emotions, particularly in the context of sign language communication. This field has been a subject of intensive study for several decades, and a lot of methodologies have been proposed to advance its effectiveness [17]. The primary focus of FER has been to identify and interpret basic emotional expressions, such as anger, disgust, fear, happiness, sadness, and surprise. Significant progress has been made in this area, particularly under controlled conditions with frontal faces and posed expressions, marking a considerable achievement in the field [42] with the use of machine learning algorithms.

Conventional approaches in FER have often relied on CNNs, which have been essential in the field for their accuracy in recognizing facial expressions. However, the primary limitations of CNNs, notably their struggles with occlusions and a tendency to prioritize static data over temporal dynamics, have led researchers to explore alternative solutions. To address these challenges, there has been a growing interest in Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTM) networks, which are adept at considering the temporal aspects of facial expressions. Additionally, the exploration of Vision Transformers represents a promising frontier in FER, offering innovative ways to tackle issues like occlusion, thus enhancing the accuracy and robustness of FER systems.

More recent approaches like Multimodal systems are becoming increasingly important in Facial Expression Recognition (FER) because they combine information from different sources like images and text to improve accuracy. Traditional FER methods often rely solely on images, which can miss small details. By incorporating additional data sources, multimodal systems provide a more comprehensive understanding of facial expressions.

One such system is CLIP (Contrastive Language-Image Pre-Training) [44], which uses both image and text encoders to learn from images and their descriptions. This method is usefull for model understand the context better, making it more effective at recognizing emotions.

Building on this, CLIPER (CLIP-based Facial Expression Recognition) [34] has been developed to enhance FER further, introducing multiple text descriptors for each expression,

capturing finer details and variations in emotions.

By combining visual and textual data, multimodal systems like CLIP and CLIPER improve the robustness and accuracy of FER, especially in complex real-world scenarios where facial expressions alone might not provide all the necessary information.

2.3.1 CNNs in FER

Artificial intelligence in contemporary times is significantly influenced by Deep Neural Networks (DNNs). Among these, CNNs, a subclass of DNNs, stand out due to their unique incorporation of convolutional and pooling layers, which are adept at processing invariant data like images [26]. CNNs have been recognized for their exceptional ability to learn from vast quantities of images, a feat achieved by adjusting the network's depth and breadth. When compared to traditional feedforward neural networks with layers of a similar size, CNNs demonstrate a superior capacity for learning from unstructured data such as images[32].

In recent developments, CNNs have solidified their status as formidable tools in the domain of object detection, with their methodologies being adeptly applied to Facial Expression Recognition (FER). It's noteworthy that image-based FER, despite being a subject of study for many years, gained significant traction following a CNN-based FER system's triumph in the FER2013 challenge, underscoring the potential of CNNs in this field.

Showcases of CNNs across the years are: In 2015 Zhiding Yu and Cha Zhang [54] proposed an Emotion Recognition in for the "Wild Challenge (EmotiW)" 2015 to classify images into the 7 basic emotions using five convolutional layers, three stochastic pooling layers and three fully connected layers, generating state-of-the-art results in the FER and SFEW datasets, achieving the first place in the challenge. The winner [26] of this same challenge won by using multiple CNN based on varying the number of properties of the individual networks. From another perspective, HoloNet, a CNN architecture where CReLU was used to increase the network depth without efficiency reduction, while in another hand, SSE, a supervised scoring ensemble, a different CNN Model, using three kinds of supervised blocks in the early hidden layers of the mainstream CNN for shallow, intermediate and deep supervision [35].

However, despite the strides made using CNNs in FER, certain limitations persist. Studies [26, 32] have highlighted that these models are often trained on datasets comprising well-defined images with minimal occlusions. This aspect of training can impact the models' effectiveness in real-world scenarios, where factors such as varying lighting and other environmental conditions play a crucial role. In the following sections, we will delve into an emerging solution designed to address these constraints.

2.3.2 RNNs and LSTMs in FER

Recurrent Neural Networks (RNN) and LSTM address a critical limitation in the design of CNNs. While CNNs are great at processing static inputs and extracting spatial features, they often overlook the dynamic, temporal aspects crucial for understanding facial expressions and emotions [46, 56, 53]. RNNs perform well in capturing long-range spatial dependencies and processing sequential data over time, making them more suited for analyzing time-variant data.

An RNN is a type of neural network where a temporal sequence is formed between the connection of the nodes, allowing it to exhibit temporal dynamic behavior. In contrast, an LSTM is a special kind of RNN, designed to avoid the long-term dependency problem. With LSTMs are capable of dependencies in sequence problems. The key difference between standard RNNs and LSTMs is the latter's ability to store information over extended time intervals through its unique gating mechanism.

The bi-directional temporal RNN (TRNN) is a significant variant, analyzing both forward and backward temporal sequences, thus demonstrating a remarkable capacity for grasping long-term temporal dependencies. This is pivotal in contexts where understanding the evolution of data patterns over time is crucial, such as facial expression and emotion recognition (FER).

The growing research in RNNs and LSTMs for processing sequential data in action recognition, speech recognition, and natural language processing has naturally extended to FER. These networks' ability to handle time-variant data is essential in accurately interpreting human emotions and expressions [56, 53].

To overcome the limitations of CNNs in handling the dynamic nature of facial expressions, the CNN-RNN (or CNN-LSTM) framework has gained attention. In this framework, the RNN (or LSTM) takes the appearance features extracted by CNN over individual frames and encodes the temporal dynamics. This combination enables the modeling of both appearance features and temporal dynamics simultaneously.

Further innovations [53] include the Nested LSTM (NLSTM) and the Spatio-Temporal Convolutional NLSTM (STC-NLSTM). The NLSTM, composed of Multi-Scale Spatial Pooling Normalization (MSSP-norm), Temporal LSTM (T-LSTM), and Convolutional LSTM (C-LSTM), aims to normalize spatio-temporal features and capture temporal dynamics and multi-level features encoded in individual convolutional layers. Similarly, the STC-NLSTM integrates a 3D CNN module for extracting spatio-temporal convolutional features, multiple T-LSTM modules for capturing temporal dynamics of facial muscle motions, and a C-LSTM module for seizing multi-level features.

In addition to the discussion on RNN, LSTM, and their application in FER, it's also beneficial to consider the role of Hidden Markov Models (HMMs) in this domain. HMMs

bring a unique approach to the recognition process, complementing the capabilities of RNNs and LSTMs[33].

Recent advancements have further highlighted the synergy between CNNs and LSTMs in FER. Huang and Chouvatut's 2024 study introduced a robust framework combining ResNet and LSTM for video-based sign language recognition. This model leverages the powerful spatial feature extraction capabilities of ResNet and the temporal sequence learning strengths of LSTM, effectively addressing the challenges of gradient explosion and vanishing gradients in deep networks. The ResNet component extracts detailed features from each frame, which are then processed by the LSTM to capture long-term dependencies in the video sequences. This integration has shown significant improvements in recognition accuracy, achieving an accuracy of 86.25%, an F1-score of 84.98%, and a precision of 87.77% on the Argentine Sign Language (LSA64) dataset [28]. This demonstrates the potential of combining CNN and LSTM architectures to enhance the performance of FER systems, particularly in handling complex temporal dynamics and spatial features.

These advancements show that the integration of RNNs and LSTMs with CNNs results in a more robust model, capable of capturing both spatial and temporal features necessary for comprehensive and accurate FER.

2.3.3 Multi Layer Perceptrons

In exploring alternatives for FER, Multilayer Perceptrons (MLPs) present a compelling approach distinct from image-based analysis. Previous research [20, 23] has delved into this domain, especially highlighting its significance in conjunction with the Facial Action Coding System (FACS) for recognizing facial expressions. Unlike Convolutional Neural Networks (CNNs), which excel in image processing, MLPs are particularly adept at handling tabular data, performing classification and regression tasks, and recognizing basic patterns. These capabilities have been applied in studies focusing on LIBRAS, where Grammatical Facial Expressions (GFEs) play a critical role. By defining and utilizing features such as distances, angles, and coordinates, MLPs were trained on data representations that encapsulate the nuances of GFEs. The introduction of windowed sequences to represent the temporal dynamics of facial expressions enabled MLPs to harness time-dependent information, thereby enhancing the accuracy of GFE recognition. This underscores the potential of MLPs to process sequential data efficiently. A notable survey [29] on the linguistic interpretation of facial expressions reported that two MLP models achieved accuracies of 91% and 89.4%, respectively, when trained on 255 video clips and 45 sentences. These findings underscore the feasibility of employing MLPs for FER and pave the way for future research in this area.

2.3.4 Vision Transformers in FER

Regarding Vision Transformers, their significance and application in FER deserve attention. Vision Transformers, a type of neural network, leverage the self-attention mechanism to extract essential features. Originally successful in Natural Language Processing (NLP), these transformers have recently increased research interest in their effectiveness in various visual tasks, including object detection, semantic segmentation, image processing, and video understanding[26].

In the context of FER, traditional methodologies often rely on recognizing facial expressions that are well-defined and constrained, a scenario that is not always reflective of real-life complexities. Factors such as varying lighting conditions, the presence of glasses, or other occlusions can lead to misinterpretation of facial features. In addressing these challenges, CNN-based architectures have shown limitations, particularly with unconstrained expressions. Vision Transformer (ViT) approaches have been introduced to mitigate this issue[36].

To tackle occlusions in FER, CNN-based models incorporating attention mechanisms have demonstrated remarkable results. Furthermore, integrating patch-level attention-based CNNs with ViTs has proven effective in learning both local and global facial features. An example of this is the PACVT model 2.4, which combines a pre-trained ResNet-18 backbone CNN for intermediate convolutional feature map extraction, a PAU for local feature extraction, and a ViT for global feature learning. PACVT's performance, assessed using datasets like AffectNet, FERPlus, and RAF-DB, has shown promising results, even outperforming state-of-the-art models in certain aspects[36].

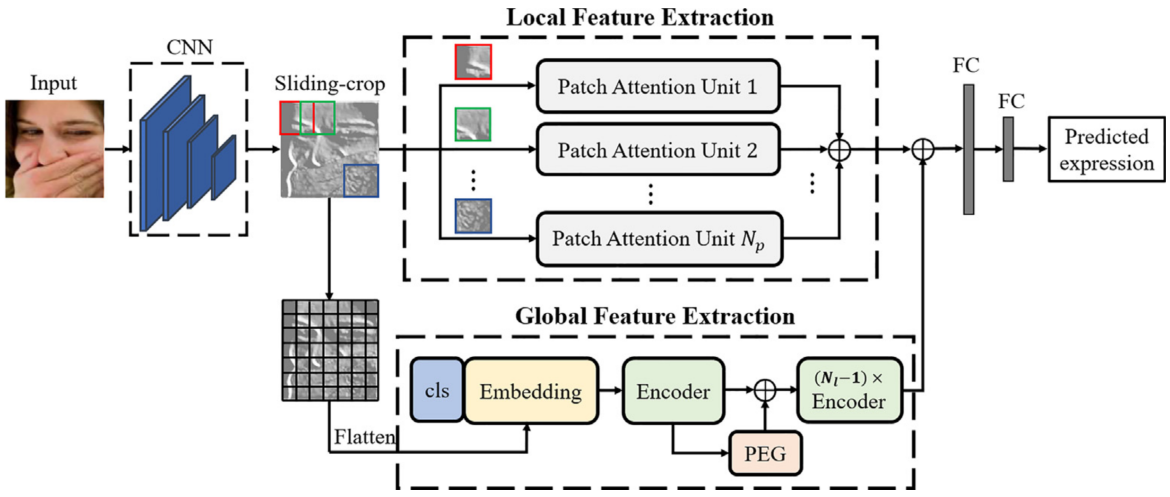


Figure 2.4: Overview of proposed PACVT[36].

Despite the advancements Vision Transformers have brought in recognizing long-range

dependencies in inputs, a gap remains in their ability to understand local features, especially when compared to the performance of shallow CNNs. A proposed solution involves the combination of transformers and convolutional networks, enhancing the transformer's capability to perceive local contexts[26].

2.3.5 Multi-Modal Learning

Multi-modal learning (MML) is an essential research area that focuses on the integration and processing of data from multiple sources, such as images, text, and speech. As [57] explains, multi-modal works similarly to human brains by accounting with multiple sources of information for decision-making. This approach is critical as real-world data often comes from various sources. Traditional methods often involve concatenating feature vectors from different sources into one long vector. However, this method is to simple, undersetimating the importance of different sources and does not account for the unique statistical properties of each modality, leading to suboptimal results. To address this, MML explicitly fuses complementary information from different modalities to enhance performance [22].

MML can be categorized into three primary groups: co-training, multiple kernel learning, and subspace learning. Co-training alternates training models to maximize mutual agreement on different modalities of unlabeled data. Multiple kernel learning combines kernels from different modalities to improve performance. Subspace learning assumes a latent subspace shared by multiple modalities, from which the input modalities are generated [22, 5].

MML is widely used in semi-supervised image classification, with approaches such as multi-modal semi-supervised learning (SSL), adaptive multimodal SSL, and multi-view vector-valued manifold regularization. For example, a multiple kernel classifier that fuses image content and descriptive keywords has shown improved performance. Another method integrates various visual features through graph fusion and employs label propagation to infer class labels of unlabeled images. Additionally, a multi-modal algorithm using vector-valued functions for multi-label image classification demonstrates the effectiveness of MML [22].

Multi-modal research focuses on three main aspects: modal alignment, modal fusion, and modal transformation. Modal alignment synchronizes data from different sources to ensure consistency. Modal fusion integrates data from various modalities to enhance the richness of the information. Modal transformation converts data from one modality to another to improve processing and understanding [37].

Recent studies show that multi-modal emotion recognition significantly outperforms single-modal methods. However, the application of multi-modal methods presents challenges such as handling missing data, synchronization issues, and the complexity of fusing different modalities [37]. Addressing these challenges, especially choosing the optimal modal fusion strategy, has been a focal point of recent research. The fusion of modalities leverages the

different expressions and perspectives inherent in each modality, enriching the available feature information. This approach utilizes both redundancy and complementarity among the modalities, enhancing the overall performance of multi-modal systems [37].

2.3.5.1 MML Applications

The efforts on research made in Multimodal Learning have achieved further more applications rather than the initial single modality ones. From them there are 5 application areas that can be highlighted where multimodals outperform the others: Speech recognition and synthesis (Audio-visual speech recognition and (visual) speech synthesis), event detection (Action Classification and Multimedia event detection), emotion and affect (Recognition and synthesis), media description (Image description, video description, visual question-answering and media summarization) and multimedia retrieval(Cross modal retrieval and cross modal hashing [5].

Multi-modal learning has also shown promising results in the field of grounded language learning. For instance, [30] demonstrated the feasibility of acquiring grounded language directly from raw speech inputs paired with visual percepts. This approach not only bypasses the need for intermediate textual representations but also enhances the inclusivity of language grounding systems. Their study showed that learning from raw speech can mitigate the performance disparities commonly seen in automatic speech recognition systems, particularly benefiting users with accented speech. By leveraging self-supervised speech representation models such as wav2vec 2.0, they were able to achieve higher accuracy and robustness in object retrieval tasks, underscoring the potential of multi-modal learning to improve human-robot communication and reduce demographic biases. This application highlights the significant advancements in integrating auditory and visual data, providing more natural and effective interactions in robotics and human-computer interaction systems.

Multi-modal learning has also shown significant advancements in the domain of sports analytics. The integration of Natural Language Processing (NLP) with multimodal models has opened new avenues for enhancing fan experiences, tactical analysis, and medical diagnostics in sports. Recent research has categorized sports datasets into language-based, multimodal, and convertible types, each supporting various applications from game summarization to player performance prediction. These developments leverage large language models (LLMs) to improve the depth and accuracy of sports analytics. For instance, the use of NLP in generating sports news from live commentaries and predicting game outcomes based on pre-game interviews has shown to enhance the analytical support for coaches and players. Additionally, the creation of multimodal datasets that combine text, video, and audio data enables the development of more sophisticated models capable of delivering real-time, personalized insights to both fans and professionals [51].

2.3.6 CLIP: Contrastive Language-Image Pre-Training

A novel approach in state-of-the-art (SOTA) computer vision systems is Contrastive Language–Image Pre-training (CLIP) [44], designed to understand and interpret images through natural language. CLIP introduces a multi-modal learning approach that consists of an architecture combining an image encoder and a text encoder, projecting images and text into the same embedding space. The image encoder can be based on Convolutional Neural Networks (CNNs) or Vision Transformers (ViT) for processing image input, while the text encoder utilizes a Transformer architecture.

The training process of CLIP involves two main objectives. The first objective is to maximize the cosine similarity for features of matched image-text pairs, ensuring that the representations of corresponding images and descriptions are closely aligned in the embedding space. Minimizing the cosine similarity for mismatched image-text pairs is the second objective, pushing apart the embeddings of non-corresponding images and texts. This contrastive learning approach allows CLIP to effectively learn a joint embedding space where images and their corresponding descriptions are closely aligned.

CLIP's ability to understand and interpret images through natural language enables it to perform a wide range of tasks, such as zero-shot classification, image retrieval, and text-to-image generation, without requiring task-specific fine-tuning. This versatility and robustness make CLIP a powerful tool in the field of computer vision and natural language processing, providing a unified framework for understanding visual and textual information.

2.3.6.1 CLIP Architecture: Image and Text Encoders and Training

The CLIP model is composed of an image encoder and a text encoder. For the image encoder, two main architectures were used. The first is based on ResNet-50 [27], but with modifications like ResNet-D, which applies downsampling to maintain spatial details. It replaces standard 1x1 convolutions with 2x2 average pooling followed by 1x1 convolutions. In [58], additional improvements like antialiased blur pooling were added to reduce artifacts from downsampling. Another key change was the replacement of the global average pooling layer with a multi-head attention layer, which enhances the representation of the image features. The second architecture is the Vision Transformer (ViT) [13], which includes an extra normalization layer for better performance.

The text encoder in CLIP uses a modified Transformer architecture [49], with 12 layers, 512 dimensions, and 8 attention heads, as described in [43]. Text is tokenized with byte pair encoding (BPE), which has a vocabulary size of 49,152 [58], and the model supports sequences up to 76 tokens. [SOS] and [EOS] tokens mark the start and end of text, and the output at [EOS] is used as the text representation. This representation is normalized and projected into the shared embedding space.

Training was done using several versions of ResNet (ResNet-50, ResNet-101, RN50x4, RN50x16, and RN50x64) and Vision Transformers (ViT-B/32, ViT-B/16, and ViT-L/14). Models were trained for 32 epochs using Adam optimizer, with weight decay and cosine learning rate decay. Memory-saving methods, such as gradient checkpointing and half-precision Adam, were used. The most successful model, ViT-L/14@336px, was pre-trained at a higher resolution, which improved performance.

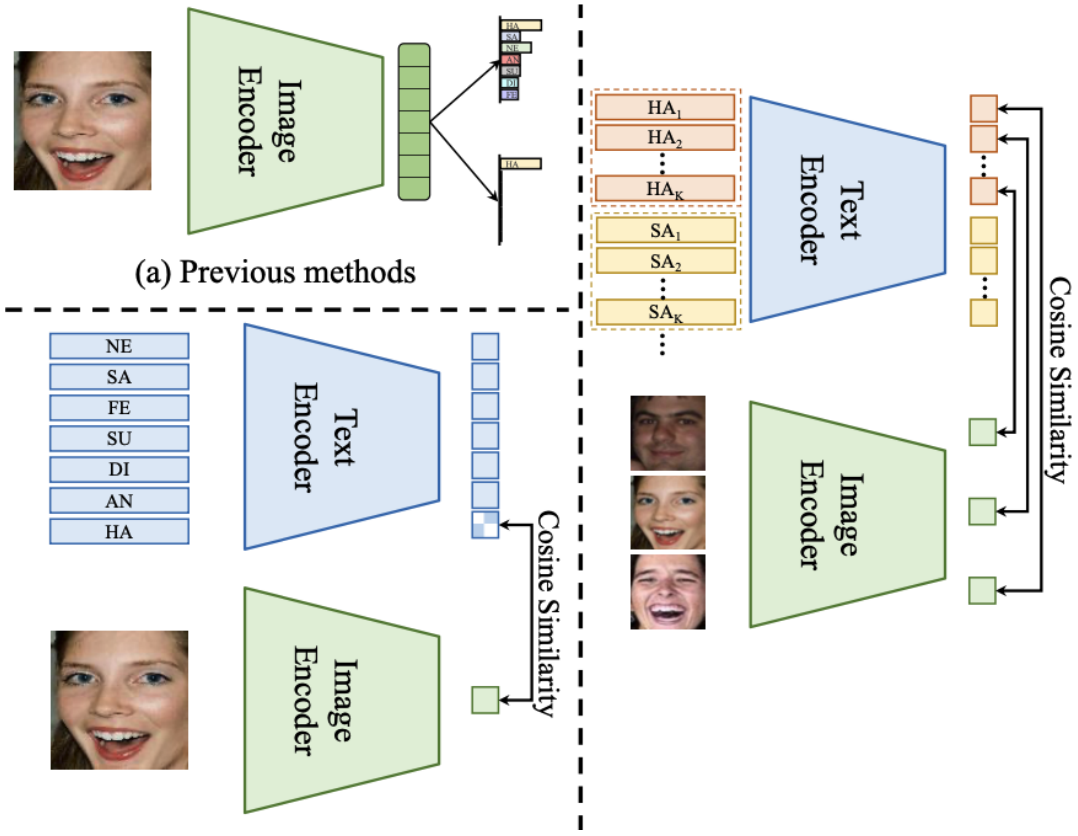


Figure 2.5: A) CLIP and B) CLIPER [44, 34].

2.3.7 CLIPER: CLIP-based Facial Expression Recognition

In 2023, a novel study [34] proposed Contrastive Language–Image Pre-training for Facial Expression Recognition (Contrastive Language–Image Pre-training for Facial Expression Recognition (CLIPER)), a tailored approach for facial expression recognition that achieved state-of-the-art (SOTA) performance in popular FER benchmarks. This research was motivated by the recognition that, much like language, facial expressions carry a wealth of information.

CLIPER builds upon the multi-modal learning framework of CLIP and introduces Multiple Expression Text Descriptors (METD). METD allows the model to learn a variety of text

descriptors for each facial expression, enabling it to capture fine-grained nuances in facial expressions. This capability is crucial for recognizing subtle differences in expressions that can convey different emotions or intentions.

2.3.7.1 Overview of CLIPER

CLIPER is designed to handle both Static Facial Expression Recognition (SFER) and Dynamic Facial Expression Recognition (DFER). It extends the CLIP framework by incorporating multiple text descriptors for each expression category, addressing the complexity and variability of human facial expressions. Unlike standard CLIP, which uses a single text prompt for each class, CLIPER generates a set of text descriptors, improving the model's ability to interpret and differentiate between various expressions.

2.3.7.2 Multiple Expression Text Descriptors (METD)

One of the key innovations in CLIPER is the introduction of Multiple Expression Text Descriptors (METD). METD allows the model to automatically learn a group of text descriptors for each facial expression. This approach provides a more nuanced and fine-grained understanding of expressions, capturing different forms and intensities of the same emotion. For example, METD can distinguish between a slight smile and a broad grin, both of which fall under the category of "happy".

2.3.7.3 Two-Stage Training Paradigm

The training process of CLIPER is divided into two stages:

Learning Expression Text Descriptors In the first stage, CLIPER learns the multiple expression text descriptors (METD) from the aligned vision-text embedding space. The text descriptors are designed to represent different forms of each facial expression. During this stage, the parameters of both the text and image encoders are frozen to maintain the alignment of the embedding space. A fine-grained cross-entropy loss is used to maximize the cosine similarity between the image embedding and the closest text embedding of the target expression, while minimizing the similarities with non-target expressions.

Fine-Tuning the Image Encoder In the second stage, the learned text descriptors are fixed, and only the image encoder is fine-tuned. This stage focuses on extracting more discriminative features related to facial expressions. The same loss function is used to guide the image encoder, ensuring that it learns to recognize fine-grained features according to the METD learned in the first stage.

2.3.7.4 Performance and Applications

CLIPER has been extensively tested on several popular in-the-wild FER datasets, achieving SOTA performance. The model's ability to interpret and differentiate between subtle variations in facial expressions makes it particularly suitable for applications in sign language recognition, where facial expressions play a crucial role in conveying meaning.

By leveraging the strengths of CLIP and enhancing them with features specifically designed for facial expression recognition, CLIPER provides a powerful tool for understanding and interpreting complex facial expressions in various contexts. The innovative design and capabilities of CLIP and CLIPER demonstrate the potential of multi-modal learning approaches in advancing the fields of computer vision and natural language processing, offering new possibilities for applications that require the integration of visual and textual information.

2.4 Case studies

2.4.1 Brazilian Sign Language - LIBRAS

Facial expressions in Libras are described [18], as in many other languages, to have a great impact and responsibility on a morphological and syntactic level. it can determine polarity, and conditionals, creating questions and relative phrases.

In 2022, a study on facial action unit detection in Libras [16], obtained 88% accuracy for 199 classes, with an AU recognition model architecture combining geometric-based features and SqueezeNet 2.7. It recognizes that CNNs are the most used option in AU detection, however, it highlights the recent effort to use a hybrid solution more specifically a combination of CNNs and temporal LSTMs. They recognized the challenge of the recognition of faces and, therefore used OpenCV and DLib to detect and resize the image to the face with an image of 96 by 96 pixels. Using Dlib with the Haar.cascade method they've extracted 68 landmarks denoted as

$$l_i = (x_i, y_i), \text{ where } i \in \{1, ..., 68\}. \tag{2.1}$$

and from then they created a set of data points L_i (for $i = 1, ..., 68$) with the Euclidean distances:

$$\begin{aligned} d^2(l_j, l_k), \text{ where } (j,k) \in \{&(3,13), (17,21), (21,22), (22,26), \\ &(38,40), (43,47), (48,54), (50,58), \\ &(52,56), (62,66)\}. \end{aligned} \tag{2.2}$$

For example, the 13th landmark minus the third would result if an eye would represent how open an eye would be. The AUs were then transformed into vectors to define a hermite curve which is highly used to obtain patterns, graphical curves, and trajectories in data points.

A remarkable note from this study was the approach to occlusion of the face, where they decided to drop the frame and mark it with the code AU 73. As said before a SqueezeNet architecture and a CNN, with one input layer, three convolution layers, two max-pooling layers, and two full connection layers, were compared and both outperformed models like AlexNet, VGG-16 and ResNet-50 as shown in figure 2.6.

Architecture	Preprocessing	Metrics			
		Precision	Recall	Accuracy	F1
AlexNet	Upper face	0.7199	0.6226	0.6620	0.6306
	Lower face	0.8027	0.5719	0.5402	0.5454
	Avg.	0.76	0.59	0.60	0.58
VGG-16	Upper face	0.6123	0.6098	0.5322	0.5249
	Lower face	0.4732	0.4698	0.4082	0.3949
	Avg.	0.54	0.53	0.47	0.45
ResNet-50	Upper face	0.5768	0.4316	0.51	0.5066
	Lower face	0.4612	0.3535	0.4898	0.4542
	Avg.	0.52	0.39	0.50	0.48
CNN	Upper face	0.8564	0.6697	0.7999	0.7516
	Lower face	0.8187	0.6670	0.7257	0.7351
	Avg.	0.83	0.66	0.76	0.74
SqueezeNet	Upper face	*0.8956*	*0.8134*	*0.8991*	*0.8525*
	Lower face	*0.8840*	*0.8092*	*0.8635*	*0.8450*
	Avg.	**0.85**	**0.71**	**0.88**	**0.84**

Italic values represent the best results for each network

Bold values represent the best results overall

Figure 2.6: Comparative analysis of Different Networks on the Libras Dataset [17]

A different study [17] focused on the recognition of affective and grammatical expressions in Brazilian Sign Language (LIBRAS) highlights a promising direction for our research in Portuguese Sign Language. The proposed methodology involves a two-stage approach: feature extraction followed by the learning process .

In the feature extraction stage, 68 facial landmarks, as represented in figure 2.2, are identified across the face using DLib, and the images are adjusted to focus on a 96 x 96 facial area. These landmarks are divided into two groups: the upper portion (including the forehead, eyebrows, and eyes) and the lower portion (comprising the chin, mouth, and nose). For Action Unit (AU) classification, geometric characteristics are derived using the positions of these landmarks, with specific distances calculated for further analysis.

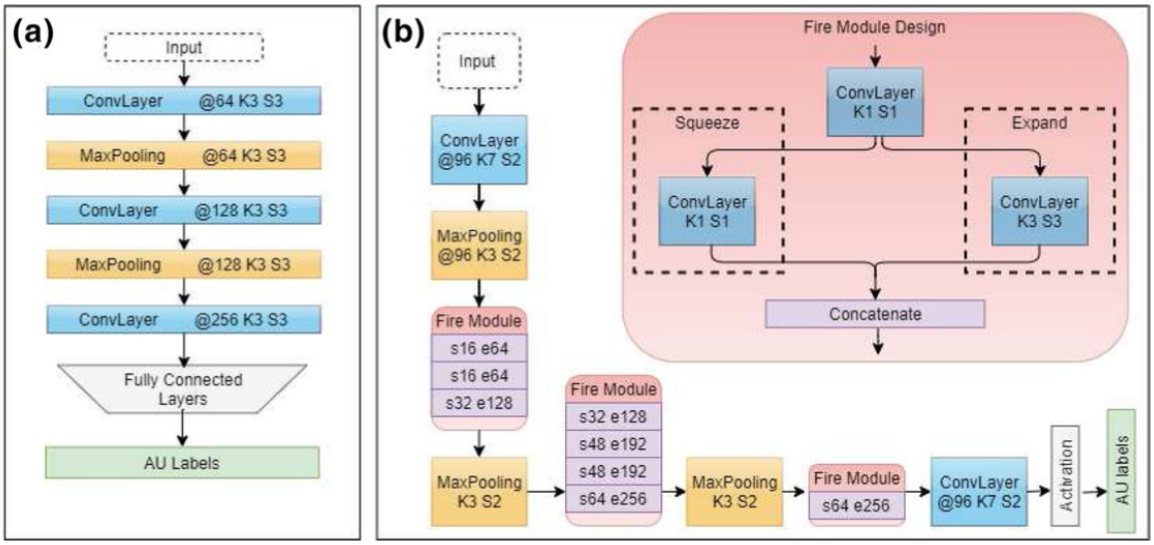


Figure 2.7: Input image, CNN architecture and SqueezeNet architecture. (a) Architecture of the CNN, consisting of one input layer, three convolution layers, two max pooling layers, and two full connection layers. (b) Structure of the SqueezeNet, with @, K, S, e, s denoting number of filters, kernel size, stride, expand filters, and squeeze filters, respectively. Detailed descriptions are given in the text[18].

The implementation utilizes a CNN-based classification strategy, drawing from previous studies and methodologies. Two architectures 2.9 are explored and compared. The first is a straightforward CNN design featuring an input layer, three convolutional layers, two max-pooling layers, and two fully connected layers. The second, more complex architecture, is a hybrid model in which the input layer feeds into a CNN, followed by a pooling layer that then connects to an LSTM network. With LSTMs stacks, are used to capture temporal dependency across frames, and the outputs of them will be the input to a final dense layer that performs multi-label learning.

2.4.2 American Sign Language - ASL

ASL is extensively used worldwide, predominantly by Americans and Canadians. It is estimated that over 500,000 individuals use ASL. Additionally, in relation to Portuguese Sign Language, the significance of facial expressions in conveying meaning is noteworthy.[6]

Similar to LIBRAS, research on the recognition of ASL has been underway for several years. A notable study[3], focused on this area. This research developed two models: the first utilized CNNs and achieved an accuracy rate of 80.59%, while the second employed YOLOv5, resulting in a more promising accuracy rate of 84.96%.

It is important to note that, despite achieving high accuracy, the dataset used in this

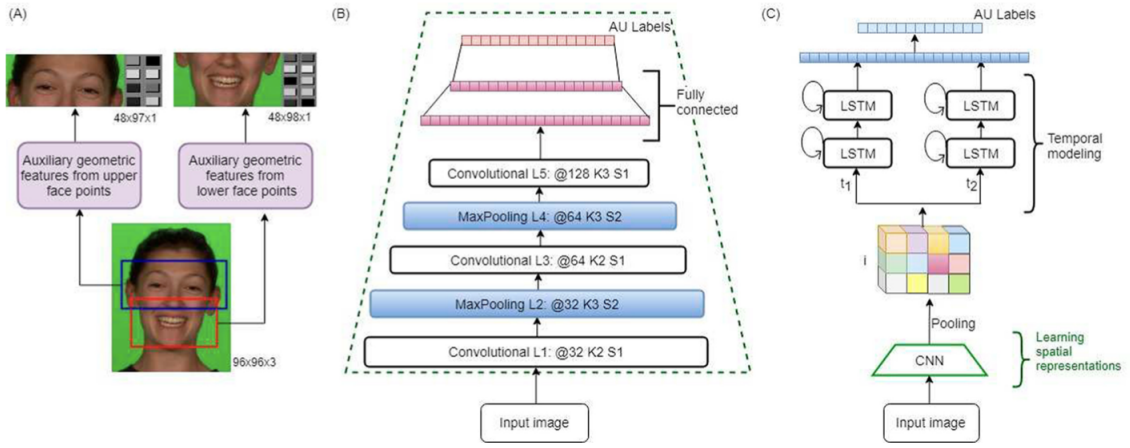


Figure 2.8: Input image, CNN architecture and CNN+LSTM architecture. In image (A), input for the networks. In image (B), CNN arquitectures. In image (C), Hybrid network CNN + LSTM [17].

study was the MNIST dataset, consisting on a large dataset of images of handwritten digits. Therefore, the model only predicts a limited number of classes, which contrasts with the broader model we aim to develop.

Another significant study [6] delved into the recognition of these handwritten digits. It employed a leap motion controller to gather data on palm flexion, hand movement, the relationship between palm and fingertips, and the distance between fingertips. The model used a pre-trained classifier and was trained with 2600 samples, 100 samples for each of the 26 ASL digits. It achieved an impressively high accuracy rate of 99.44% and a 91.82% accuracy rate in a 5-fold cross-validation. However, this study also focused solely on digits.

In the scope of our thesis, a 2020 study [24] on ASL focused on comparing a model using manual data alone and another incorporating both manual data and facial expression data. Electromyography sensors (EMG) were employed to collect data from facial expressions. The comparison of the two models revealed that the inclusion of facial expressions significantly impacted the results, improving both test error rates and accuracy. Two models were trained: the first being a Long Short-Term Memory (LSTM) model and the second, a Transformer model, with a CNN utilized to extract features from the input data. For the LSTM model, the sentence error rate without input from EMG sensors was 9.17%, which increased to 14.72% when facial expressions data were excluded. The Transformer model demonstrated even more notable results, with error rates of 4.72% and 8.89% for inputs with and without EMG data, respectively.

2.4.3 German Sign Language - GSL

Similar to the previously mentioned sign languages, GSL also has a significant global impact. It is used by over 200,000 individuals for everyday communication.

Like the other studied languages, GSL has been the subject of research for some time. One notable study [7] focuses on the application of Hidden Markov Models (HMMs). HMMs are particularly effective in automatically detecting the boundaries of signs, making them highly efficient for identifying signs that consist of a sequence of movements rather than a single pose. This particular study included a lexicon of 97 signs in German Sign Language and achieved an accuracy rate of 91.7%.

Furthermore, another research [20] highlights the necessity to treat sign language as spatio-temporal language rather than simply spatial. With the use of a famous German Continuous SLT dataset, RWTH-PHOENIX-Weather 2014T [5] , which provides video translations and gloss level annotations for GSL videos of weather broadcasts, they have proposed a state-of-art framework using a spatial and a word embedding to generate spoken language translations from sign language videos, however neglecting facial expressions. They have used a CNN spatial based spatial embedding, RNN-HMM hybrid tokenization methods, and attention-based encoder decoder networks.

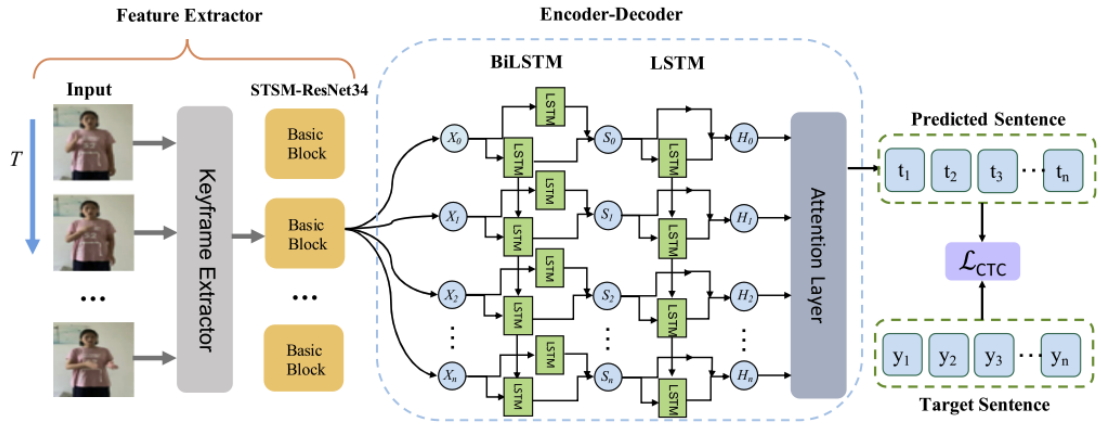


Figure 2.9: KSRB-NET model architecture[52].

In 2023, a significant advancement in continuous sign language recognition was introduced with the development of the KSRB-Net model [52]. This model consists of four main components that together enhance the accuracy and robustness of the recognition process. The first component is keyframe extraction, which efficiently reduces the redundancy in the data by selecting the most informative frames from the video stream. This process accelerates training and minimizes the risk of overfitting.

[5] https://www-i6.informatik.rwth-aachen.de/~koller/RWTH-PHOENIX-2014-T/

The second component is a visual encoder based on ResNet34, a deep convolutional neural network known for its strong feature extraction capabilities. ResNet34 captures detailed visual information from each frame, providing a solid foundation for recognizing complex sign language gestures.

The third component integrates a BiLSTM (Bidirectional Long Short-Term Memory) network, which processes the temporal dynamics of sign language. This is crucial for understanding the sequence of movements and ensuring that the temporal context of gestures is preserved.

Additionally, KSRB-Net includes a spatial-temporal and motion information perception mechanism. This innovative approach captures the intricate timing and spatial relationships between movements, enhancing the model's ability to interpret subtle variations in sign language gestures.

Finally, the model employs an LSTM network combined with a Connectionist Temporal Classification (CTC) mechanism for sequence alignment and decoding. This setup allows the model to handle continuous sequences effectively, ensuring accurate alignment between the video frames and the corresponding sign language annotations.

The effectiveness of KSRB-Net was demonstrated through extensive experiments on several datasets, including the TJUST-SLRT dataset. The results showed significant improvements in accuracy over previous methods, starting from simpler CNN+LSTM configurations and evolving to the more complex and effective Keyframe+STSM-ResNet34(pre)+BiLSTM (KSRB-Net) methodology.

2.4.4 Notable Studies in Sign Language Recognition

The field of Sign Language Recognition (SLR) has seen considerable advancements across various languages and datasets, with researchers exploring innovative methodologies to improve recognition accuracy. For instance, deep learning models such as CNNs and LSTMs have been pivotal in enhancing the recognition of facial expressions, gestures, and grammatical structures in LIBRAS [17, 16]. The incorporation of facial landmarks, Hermite curves, and Action Units into recognition models has significantly improved performance in both static and dynamic sign language data [16, 15]. Meanwhile, hybrid approaches combining CNNs with Recurrent Neural Networks (RNNs) have demonstrated marked success in continuous sign language recognition, as illustrated in studies on ASL [24] and GSL [20].

Recent frameworks such as CLIP-based models [44, 34] and PACVT [36] exemplify the application of multimodal learning and attention mechanisms, providing further insights into recognizing complex spatial-temporal patterns in sign language. Moreover, transformer-based architectures and multimodal systems have shown promise in overcoming occlusion and variability in video data, thereby pushing the boundaries of sign language recognition

across languages such as Brazilian Sign Language (LIBRAS) and German Sign Language (GSL) [52, 20].

These researches underscore the importance of integrating hand, facial, and body movement data to create robust SLR models. As SLR research continues to evolve, the exploration of multimodal systems, deep learning frameworks, and culturally adaptive models will remain key factors in improving the inclusivity and accuracy of sign language recognition systems.

3

PSL Dataset and Methodology

3.1 Dataset

The dataset used in this thesis consists of videos recorded between 1998 and 2021, where translations from Portuguese to Portuguese Sign Language (PSL) were made. This collection was gathered with the help of Universidade Católica Portuguesa. The university's language students contributed additional resources related to the video translations, which helped expand the dataset. As discussed earlier, translating into sign language comes with its own challenges, which is shown through the example of a JSON annotation from a single video in figure 3.2.

Figure 3.1: Frame Example from Datasets

The main focus of the project is to analyze facial landmarks and action units (AUs) to see how they relate to specific expressions or sentences in PSL. In this process, we extracted 68

facial landmarks (with x and y coordinates) from each video frame, resulting in a dataset with 136 columns. Additionally, the dataset includes six main emotions—*Anger*, *Disgust*, *Fear*, *Happiness*, *Sadness*, *Surprise*—and a neutral emotion, all measured from 0 to 1. The facial boundaries are captured by four coordinates (*FaceRectX*, *FaceRectY*, *FaceRectWidth*, *FaceRectHeight*) and a *FaceScore*, which gives the confidence of detecting a face. To gather these key facial metrics, we used the py-feat library, which connects facial features with the sign language translations. Additional data like the video path, unique ID, and the frame's timestamp were also recorded to ensure proper organization. Each frame is labeled with annotation values, representing the grammatical structure in PSL. There are 72 unique annotation values and 824 unique meanings attached to these annotations, giving us a broader context for each PSL gesture in the dataset.

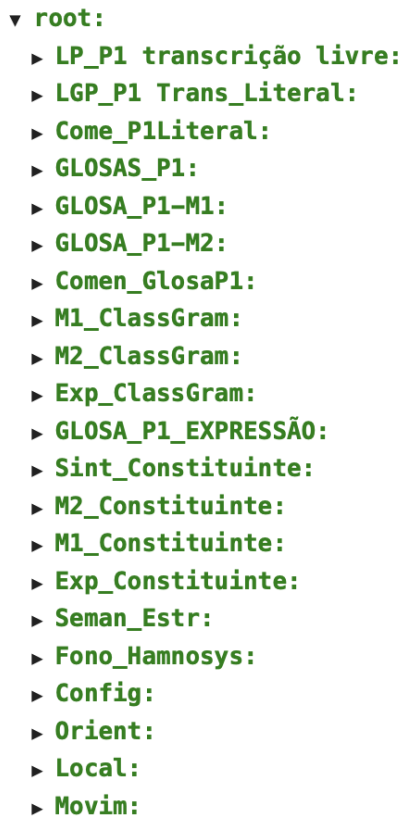


Figure 3.2: JSON showing the classes from a translated PSL video.

3.2 Data Preparation

3.2.1 Data Format

The dataset begins in a structured format, often stored as a JSON file that contains annotations corresponding to different time frames in each video. Each annotation provides various types of linguistic and grammatical information about the PSL gestures. For example, the field *LP_P1 transcrição livre* provides full-sentence transcriptions, such as "E o elefante olhou para cima para o pássaro," which is a natural language description. In contrast, the *LGP_P1 Trans_literal* field offers a more literal translation in PSL, such as "[ELEFANTE_TROMBA_PEQUENO_OLHAR_PARA_CIMA-PATA_ELEFANTE]."

Each annotation array is aligned with the time stamps of the video frames, ensuring synchronization between the video data and the corresponding linguistic or grammatical descriptions. Of particular interest for the study of grammatical classes is the *Exp_ClassGram* field, which categorizes the grammatical components of each PSL gesture.

Beyond the annotations, each frame of the video includes crucial facial data. Facial boundaries are represented by four coordinates: *FaceRectX*, *FaceRectY*, *FaceRectWidth*, and *FaceRectHeight*, which define the face's position and size within the frame. A confidence score, *FaceScore*, indicates the reliability of the facial detection.

In addition to the facial boundaries, each frame contains 68 facial landmarks (with x and y coordinates) that capture facial features crucial for interpreting PSL grammar. Emotional information is also incorporated, with scores for *Anger*, *Disgust*, *Fear*, *Happiness*, *Sadness*, *Surprise*, and a neutral state, each on a scale from 0 to 1, representing the intensity of the corresponding emotion.

Metadata is also included with each frame, such as the frame number, the timestamp within the video, and the file path for the frame image. Annotation values provide detailed grammatical structure and meaning of each gesture, including 72 unique annotation values and 824 different meanings, which offer rich linguistic insight into PSL.

This combination of facial, emotional, and grammatical data provides a comprehensive basis for analyzing PSL gestures and their grammatical components.

3.2.2 Data Processing

To enhance the model's ability to recognize and classify facial expressions within PSL gestures, several data processing steps were applied.

Firstly, the 68 facial landmarks were extracted from each frame to capture detailed facial movements. Action Units (AUs), which measure the intensity of specific facial muscle movements such as the raising of eyebrows or lip corners, were computed for a deeper analysis of facial expressions.

Based on the extracted AUs, Hermite curves were generated to model the temporal changes in facial expressions. These curves were overlaid onto the face images to visualize the dynamic aspects of facial movements. Different types of images were created to enrich the dataset: raw face images, face images with overlaid landmarks, and face images with Hermite curves drawn either directly on the face or adjacent to the image. This variety helps the model learn both the spatial and temporal aspects of facial expressions.

Emotional scores were incorporated to provide context about the emotional tone associated with each facial expression, which is significant for interpreting PSL gestures. By systematically processing the data in this manner, the dataset effectively captures both the static and dynamic features of facial expressions, facilitating more accurate recognition and classification of PSL gestures by the model.

3.2.3 Data Extraction

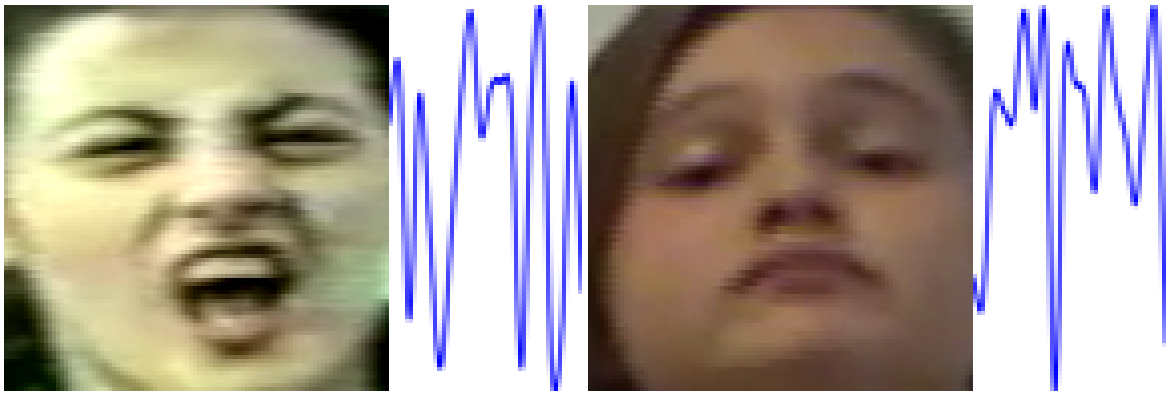
The raw data consists of video clips annotated with PSL gestures, specifically focusing on the grammatical expressions represented by facial movements. Using the Py-Feat library, we extracted the facial regions from these clips to create a dataset of images that capture the facial expressions corresponding to different PSL gestures. The following image types were derived:

- **Raw Face Image**: This image represents the bounding box around the subject's face, extracted directly from the video frames.
- **Face Image with Landmarks**: This image is similar to the raw face image but includes 68 facial landmarks overlaid as white dots. These landmarks correspond to key facial points, such as the eyes, nose, and mouth, and are crucial for identifying specific facial expressions.

3.2.4 Hermite Curve Generation

To further enhance the representation of facial expressions, we generated Hermite curves that model the temporal progression of Action Unit (AU) values across the face. In The Hermite curves were computed using a set of AU values extracted for each frame and were then superimposed on the corresponding face images. The process for generating these curves is as follows:

- **AU Extraction**: Each face image is associated with a set of AU values, which quantitatively describe the intensity of specific facial muscle movements. The AUs were extracted using Py-Feat, including values for actions such as brow raising (AU01), lip corner puller (AU12), and more.



(a) Person 1: No LM with Curve Right (b) Person 2: No LM with Curve Right

Figure 3.3: Comparison of Hermite Curves for Different Facial Expressions

- **Hermite Curve Computation**: The extracted AU values were used to compute a Hermite curve, which interpolates the AUs across the face. The interpolation was performed using the `CubicHermiteSpline` function, which ensures a smooth transition between AUs, capturing the nuanced changes in facial expressions.

- **Hermite Curve Overlay**: The Hermite curve was overlaid, on top or the right, onto both the raw face images and the images with facial landmarks. This overlay provides a visual representation of the AU dynamics, enhancing the image data with temporal information that reflects the underlying facial muscle movements.

As represented in figure 3.3 different facial expressions will generate different Hermite curves. With those, we aim to give the models a more comprehensive way to understand the nuances needed and impact effectively and better predict the grammatical facial expression.

3.2.5 Data Augmentation

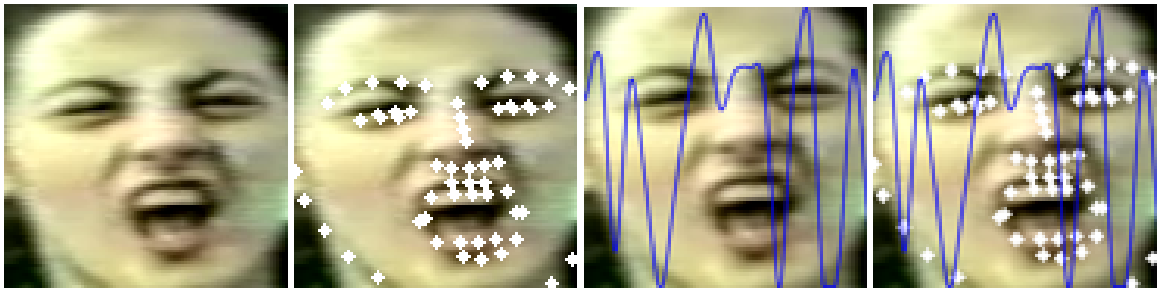
To further enrich the dataset, we introduced variations by creating multiple versions of the face images:

- **Face Image with Hermite Curve**: This image includes the Hermite curve overlaid on the raw face image, providing a direct visual correlation between facial features and the curve.

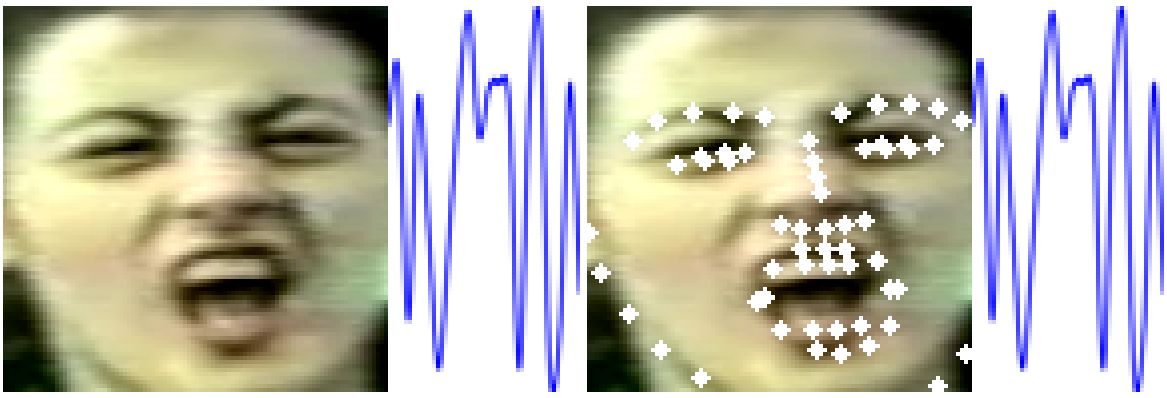
- **Face Image with Landmarks and Hermite Curve**: This version combines the facial landmarks with the Hermite curve overlay, offering a comprehensive representation of both the spatial structure and the temporal dynamics of the facial expressions.

- **Face Image with Hermite Curve on the Right**: In this variation, the Hermite curve is placed on the right side of the face image. This layout is inspired by related works that utilize the Hermite curve as a complementary visual tool next to the image.

- **Face Image with Landmarks and Hermite Curve on the Right**: Similar to the previous version, this image includes both the facial landmarks and the Hermite curve, but with the curve positioned to the right of the face image. This format aims to enhance the visual separation between the facial features and the Hermite curve, following the design of established methodologies in the field.



(a) No LM (b) Face Image (c) No LM Hermite (d) Hermite Image



(e) No LM with Curve Right (f) Face with Curve Right

Figure 3.4: Comparison of Different Face Image Processing Techniques

These augmented images aim to provide the model with a richer set of features, potentially improving its ability to recognize subtle variations in facial expressions associated with PSL gestures. The addition of the Hermite curve on the right side is specifically designed to align with methodologies from the literature, where such complementary visual cues have been shown to aid in gesture recognition tasks.

3.3 Evaluation methodology and Dataset

In a preliminary stage of the research, we've tested the performance of different architectures, across different types of datasets.

Using a small part of the full dataset, we extracted data from a set of 12 videos, with 9 different subjects speaking PSL making a total of 9179 different facial expressions, 46 different grammatical classes and 347 unique meanings for the grammatical classes, using the pre-processing stage shown in Figure 3.5. To train the different models, we have used the same parameters to have a fair evaluation, we set the learning rate to 0.01, the batch size to 32, and the number of epochs to 25. Despite not using AUs in this stage of the research we already extract them in the first stage of the pre-processing of a single frame in the video. Furthermore we have splitted the data in 3: 70% Training (60% training and 10% validation) and 30% Test.

Figure 3.5: Pre-Processing Stages

Given that we are pioneering the development of a model for recognizing facial expressions in Portuguese Sign Language, we face constraints in evaluating our models. To address this, we will employ the F1 Score as our primary metric across all networks, recognizing its effectiveness in assessing models with imbalanced datasets. Additionally, we will incorporate Accuracy, Precision, and Recall in our testing phase to provide a comprehensive evaluation of our models' performance.

Our preliminary analysis will not only compare the results across different architectural frameworks but will also explore the effects of various data preprocessing techniques. This includes the use of raw images, images with a defined Region of Interest (ROI), and images with both a defined ROI and facial landmarks depicted. Essential for our understanding on what to expect in the final implementation of the models.

In the absence of direct comparisons with models designed for PSL (Portuguese Sign Language) recognition, we will compare our findings with those from Facial Expression Recognition (FER) models for other languages. This comparative approach will help contextualize our results within the broader field of FER.

Firstly, following the most standard research [17, 16] in FER for SL, we have trained a shallow Convolutional Neural Networks (CNN), achieving an F1 Score of 0.6 over 46 classes and 0.648 over 10 classes. The accuracy, precision, and recall metrics are 0.67, 0.72, and 0.553 respectively for the 46 classes model and 0.69, 0.71, and 0.62 for 10 classes. The increase on the value can be attributed to excluding the classes with less samples in the dataset.

In the subsequent model for our investigation, we experimented with the AlexNet architecture, renowned for its importance in advancing deep learning within the visual recognition domain. This decision was motivated by AlexNet's historical significance and potential applicability to FER in sign language (SL). The results of employing AlexNet were notably superior to those obtained with a shallow CNN model. Specifically, for the 46-class scenario, the AlexNet model achieved an F1 Score of 0.742, Accuracy of 0.752, Precision of 0.767, and Recall of 0.741. Meanwhile, within the more constrained 10-class framework, the model maintained its robust performance, recording an F1 Score of 0.741, Accuracy of 0.740, Precision of 0.713, and Recall of 0.620. These outcomes underscore the efficacy of deeper, more complex architectures like AlexNet in handling the intricacies of FER across diverse class specifications in sign language.

3.3.1 Qualitative and Comparative Analysis of Preliminary Results

The findings from various architectural models not only encourage further refinement in data processing to enhance model performance but also stimulate additional research within this field. To have a good and standardized measure of comparison among the models tested, we trained them with 25 epochs and a batch size of 32. A noteworthy aspect of our analysis is the comparison between models trained on datasets comprising 46 and 10 classes, respectively. Contrary to initial expectations of significant improvements in performance metrics with fewer classes, the increase in F1 scores was relatively modest, with the most notable improvement observed in our shallow CNN model, from 0.60 to 0.648. This outcome may be attributed to the considerable imbalance in our dataset, as depicted in 3.6, which likely influenced the models' performance.

An in-depth examination of the F1 scores for each class 3.7 within the models revealed that the five most represented classes underscored in performance, despite the assumption that a higher volume of data would yield better results. This was particularly evident for four classes (Adjectives, Verbs of Transportation, Adverbs, and Nouns), which under-performed against expectations. In contrast, the class representing interrogative sentences demonstrated superior performance among the top five classes. This success can be attributed to its direct semantic alignment, as opposed to the broader and more varied sub-meanings encompassed by the grammatical categories of Adjectives and Nouns in Portuguese.

We can further observe that deeper architectures like AlexNet add overall better results

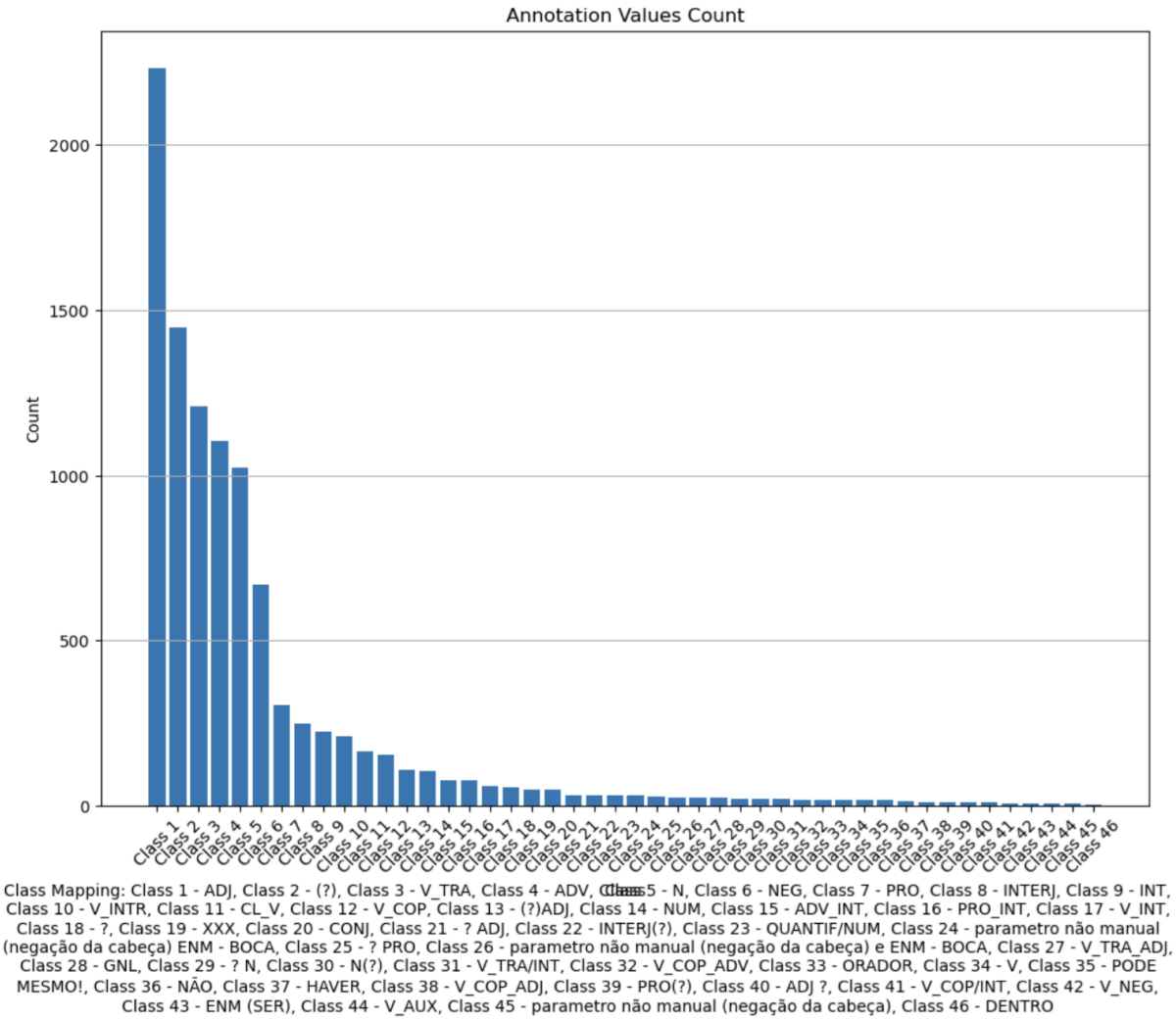


Figure 3.6: Number of Occurrences of frames per class

than simpler ones like the proposed CNN by [17, 16], although we have not yet adjusted the pre-processing stage to take into account the action units either by concatenating to the image, grayscaled squares or Hermite Curves to represent the Action Units in each analyzed face.

In this initial study, we also tested ResNet-50, a network known for its pre-training on extensive datasets. Interestingly, this model showed the lowest performance metrics compared to the others we tested. ResNet-50 did not converge within 25 epochs, which was a different outcome from what we observed with models like AlexNet and our initial CNN.

The performance of ResNet-50 might point towards the need for adjusting our approach, especially in how we prepare our data and set up the network for training. This could involve looking more closely at how many epochs we use and possibly changing how we preprocess our data to better suit the characteristics unique to facial expressions in sign language.

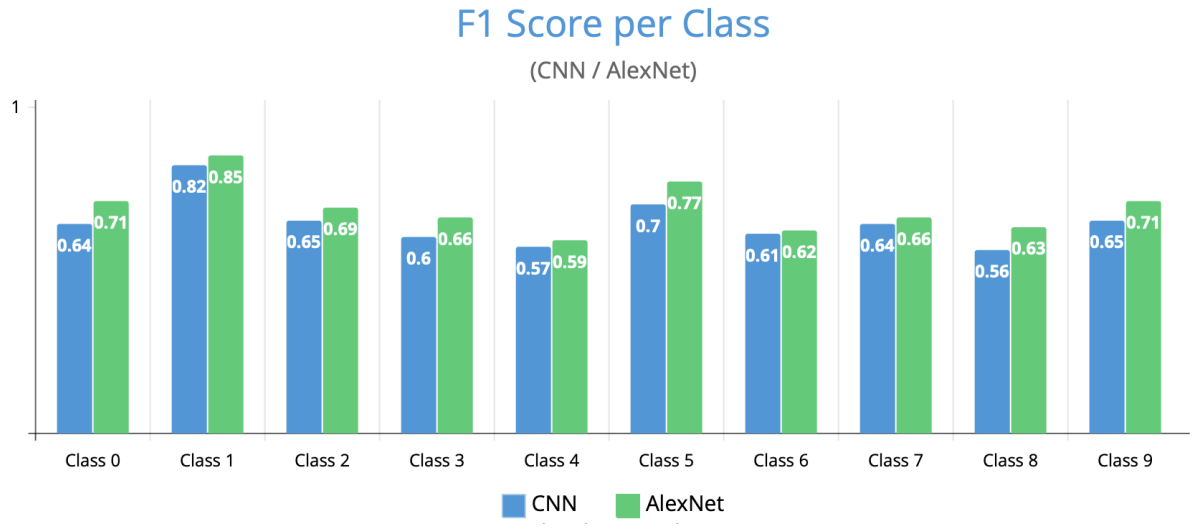


Figure 3.7: F1 Score per Class (CNN / AlexNet)

This analysis underscores the challenges of FER in sign language and points to the need for methodological improvements and further investigative efforts.

4

Facial Expression Recognition

4.1 Implementation

For the purpose of this thesis, we address the under-researched area of facial expression recognition within the context of PSL. Utilizing the PSL dataset provided by Católica University, we aim to develop an innovative model for recognizing action units in PSL. The proposed idea is described in figure 4.1 and figure 4.2.

The implementation was done in three main steps. Initially, using py-feat, we will extract the necessary data from the 5-hour video dataset. The second step involves the preprocessing of this data. Here, facial landmarks, initially represented as x and y coordinates, will be transformed into more meaningful metrics, such as the degree of eye closure or the extent of a smile. This transformation aims to yield values that are more indicative of facial expressions for subsequent analysis. In the final step, we will develop and compare three models. The first two models will draw inspiration from existing research on facial expression recognition in Brazilian Sign Language (LIBRAS), as documented in studies [17, 18], the third and novel approach is to adapt the research done in Contrastive Language–Image Pre-training (CLIP) and Contrastive Language–Image Pre-training for Facial Expression Recognition (CLIPER) but to sign language recognition, changing the descriptor input of the text embedding to the context given by manual markers. This comparative approach aims to evaluate the efficacy of these models in the context of PSL.

4.2 CNN Implementation

In this section, we thoroughly present our implementation of the Convolutional Neural Network (CNN) model, which is inspired by techniques and methodologies from *(CITE relevant sources)*. We provide a comprehensive overview of the architecture, detailing each layer's functionality and contribution to the overall model performance. Additionally, we

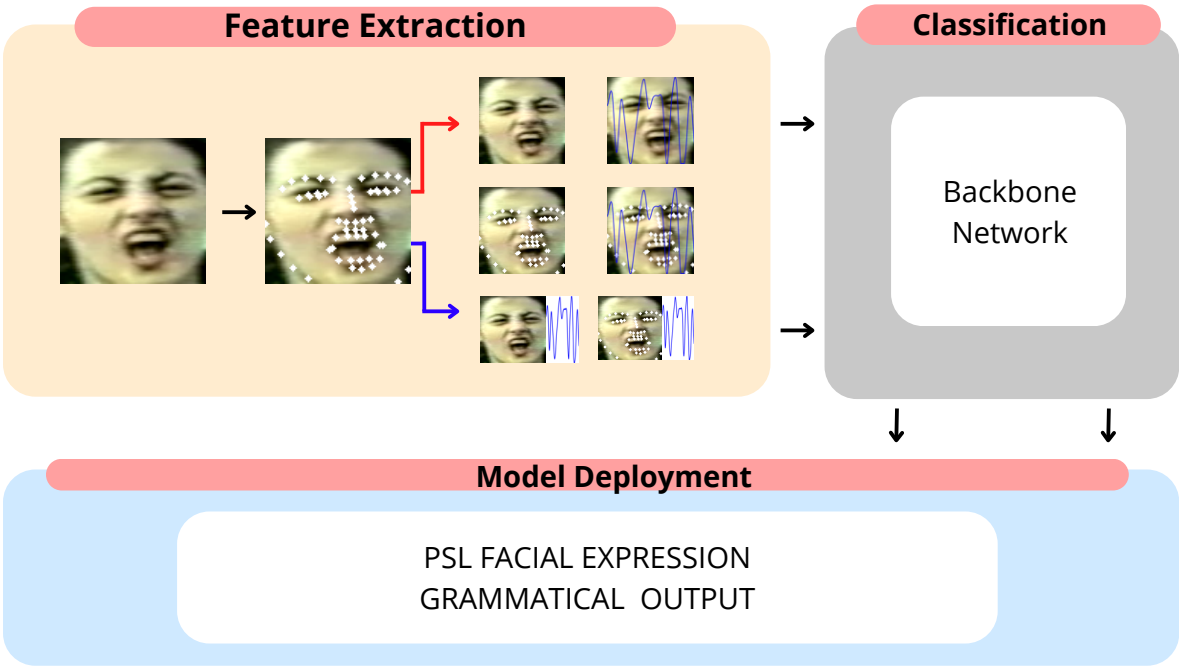


Figure 4.1: Proposed Framework

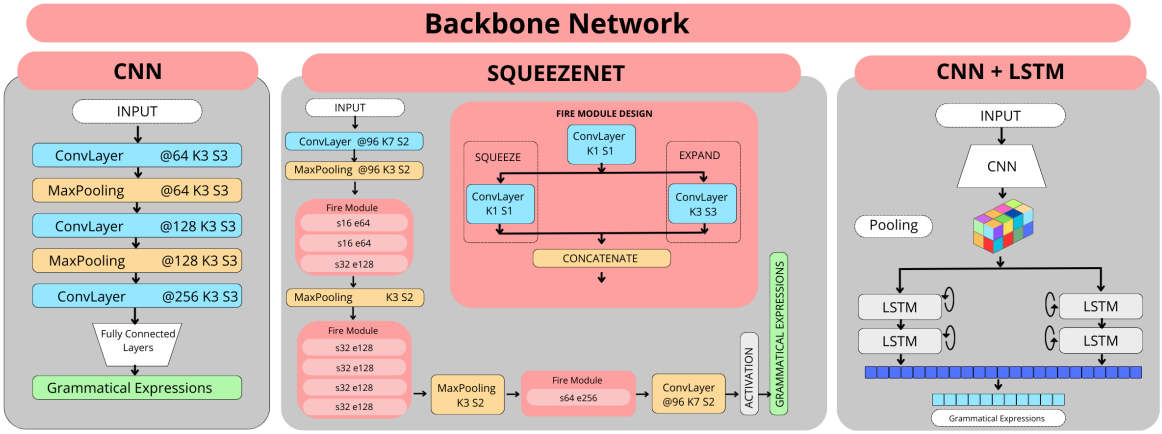


Figure 4.2: Backbone Networks

discuss the regularization and optimization strategies employed to enhance the model's generalization ability and prevent overfitting.

4.2.1 Model Architecture

Our CNN model is designed to effectively capture the features from the input images and perform the classification task. The architecture is structured to progressively extract complex patterns through multiple layers of convolution, pooling, and dense operations.

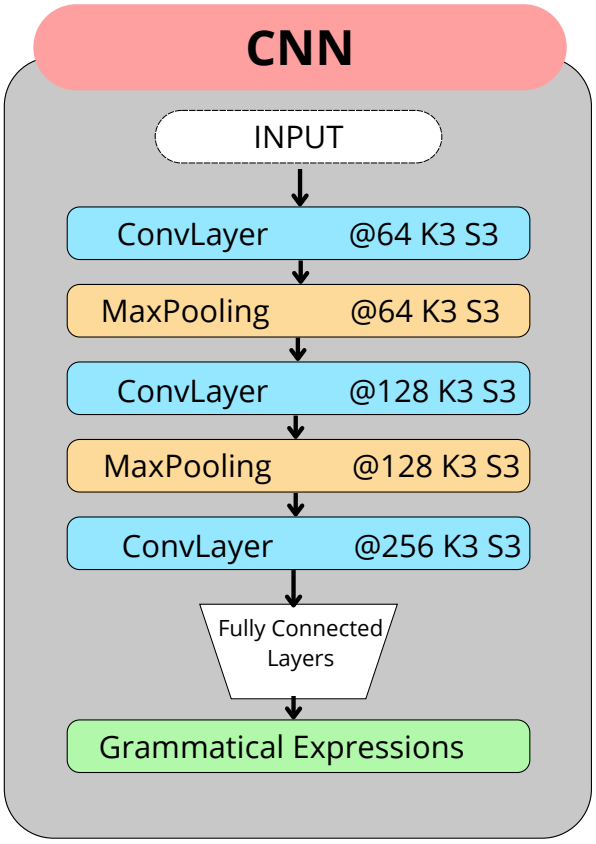


Figure 4.3: CNN Arquitecture

4.2.1.1 Layer Description

The architecture of the Convolutional Neural Network begins with an input layer that takes images of size 96×96 pixels with 3 color channels (RGB). The first convolutional layer follows, using 32 filters of size 3×3, with the ReLU activation function. Padding is set to 'same', ensuring the output dimensions match the input size. After this, a max-pooling layer reduces the feature map dimensions using a 2×2 pooling window, selecting the maximum value within each region to down-sample the data.

A dropout layer with a rate of 0.25 is added next, which randomly drops a fraction of input units during training to help prevent overfitting. The second convolutional layer comes after, this time with 64 filters of size 3×3, again using ReLU activation and 'same' padding. Another max-pooling layer with the same 2×2 pooling size follows, further reducing the spatial dimensions.

Another dropout layer with a rate of 0.25 is applied before the network reaches the flatten layer, which reshapes the 2D feature maps into a 1D vector so that it can be passed into the dense layers. The dense layer contains 128 units and also uses ReLU activation. L2

regularization, with a factor of 0.001, is applied here to limit the size of the weights and prevent overfitting.

Following this, another dropout layer is included with a higher rate of 0.5, increasing the regularization. Finally, the output layer is configured with a number of units equal to the number of classes, using a softmax activation function to generate probability distributions across the different classes.

This architecture is designed to progressively extract complex patterns through multiple layers of convolution, pooling, and dense operations, leading to a final classification decision.

4.2.2 Results and Evaluation

4.2.2.1 Model Performance

Accuracy and Loss Curves The training and validation accuracy and loss curves for the CNN models across different data configurations are presented in the figures below. These plots provide insights into the model's learning process and its ability to generalize from the training data to unseen validation data.

4.2.2.2 Classification Metrics

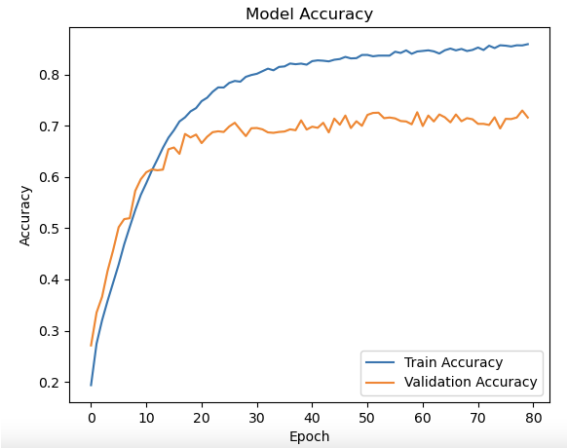
Precision, Recall, and F1 Score The performance of the CNN models was further evaluated using precision, recall, and F1-score for each class. These metrics provide a detailed view of how well the models differentiate between the various classes, particularly in handling imbalanced data.

Average Metrics Weighted averages for precision, recall, and F1-score were also calculated to provide an overall assessment of the models' performance. The table below summarizes these metrics for different configurations of the CNN models:

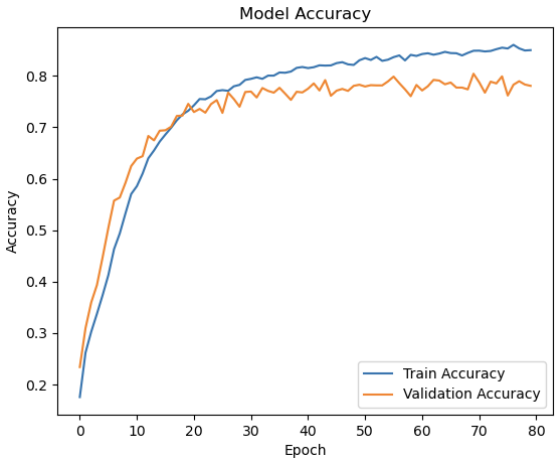
4.2.2.3 Comparative Analysis

Comparison with Baseline Models The classification performance of the CNN models was evaluated using F1 Score, accuracy, precision, and recall, which are critical in assessing models working with imbalanced datasets, such as those found in PSL recognition. Among the tested architectures, the CNN ROI model demonstrated superior performance, particularly after the fine-tuning process, achieving an F1 Score of 0.8408, accuracy of 0.8419, precision of 0.8442, and recall of 0.8419 on the full dataset. This fine-tuning, referred to as FT CNN ROI, consistently outperformed the baseline versions.

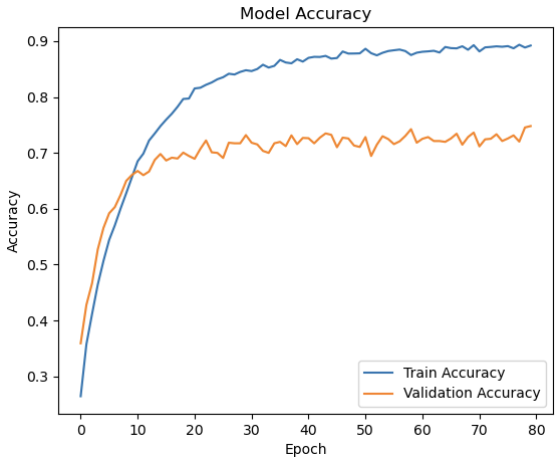
A key observation was that images without landmarks yielded the best results, reinforcing the idea that simplifying the input by removing potentially noisy landmarks helps the model



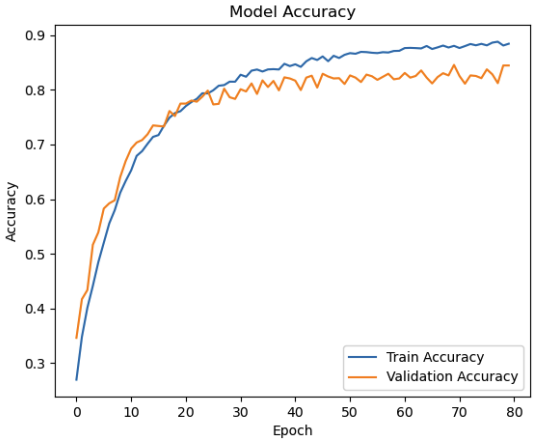
(a) Accuracy for Full Dataset with LM



(b) Accuracy for Full Dataset with ROI



(c) Accuracy for Top 10 Classes with LM



(d) Accuracy for Top 10 Classes with ROI

Figure 4.4: Training and Validation Accuracy for Different CNN Models

focus on the essential features for PSL recognition. However, an exception was seen when Hermite curves were included on the right side of the images. In those cases, the geometrical data provided by the Hermite curve added significant value by capturing the temporal progression of facial expressions, thus improving model performance.

As expected, training the models on the dataset containing only the 10 most predominant classes resulted in better overall performance compared to the full dataset. Reducing the number of classes allowed the models to focus on the most frequent and distinct PSL expressions, thus enhancing F1 scores and accuracy. This result aligns with our hypothesis that reducing class imbalance can have a positive impact on model generalization.

In contrast, adding geometrical data such as landmarks and action units generally did not

Table 4.1: Performance of different CNNs for PSL Recognition

Architecture	Dataset	F1 Score	Accuracy	Precision	Recall
CNN LM	Full Dataset	0.7122	0.7157	0.7494	0.7157
	Top 10 Classes	0.7425	0.7436	0.7562	0.7436
CNN ROI	Full Dataset	**0.7896**	0.8144	0.7553	0.7899
	Top 10 Classes	**0.8376**	0.8371	0.8457	0.8371
CNN LM Hermite	Full Dataset	0.7019	0.7035	0.7361	0.7035
	Top 10 Classes	0.7621	0.7625	0.7729	0.7625
CNN ROI Hermite	Full Dataset	0.7503	0.7517	0.7854	0.7517
	Top 10 Classes	0.7859	0.8069	0.7854	0.7845
CNN LM Hermite on Right	Full Dataset	0.7055	0.7092	0.7302	0.7092
	Top 10 Classes	0.7728	0.7721	0.7786	0.7721
CNN ROI Hermite on Right	Full Dataset	0.6838	0.6886	0.7520	0.6886
	Top 10 Classes	0.7858	0.7855	0.7887	0.7858
CNN [18]	Upper face	0.7516	0.7999	0.8564	0.6687
	Lower face	0.7351	0.7257	0.8187	0.6670
	Avg.	0.74	0.76	0.83	0.66
CNN [17]	Upper face	0.8900	0.9018	0.8972	0.8194
	Lower face	0.8522	0.8585	0.8892	0.8091
	Avg.	0.8711	0.8805	0.8932	0.8142
FT CNN ROI	Full Dataset	**0.8408**	0.8419	0.8442	0.8419

provide substantial improvements. This was evident across most architectures, except for the CNN ROI Hermite model, which incorporated Hermite curves. Beyond this model, adding geometrical information often introduced unnecessary complexity and did not significantly enhance the ability to recognize PSL gestures.

A crucial aspect of our work is comparing these results with those obtained from models referenced in the literature, such as the CNN models from [17] and [18]. These reference models, which focus on facial expression recognition in sign language tasks, achieved solid performance in their respective domains. For instance, the CNN model from [18] recorded an F1 Score of 0.8525 for the upper face dataset, with an accuracy of 0.8991 and precision of 0.8956. Similarly, the model from [17] reached an F1 Score of 0.8900 for the upper face dataset with an accuracy of 0.9018. While these models outperformed our initial architectures in certain areas, it is important to note that they relied on more complex preprocessing steps, such as extracting facial regions and combining features from both the upper and lower face.

When comparing our FT CNN ROI model to these reference models, it becomes clear that our fine-tuned architecture, which focuses on pure appearance-based features, performed

competitively. Despite not incorporating as many geometrical or region-based features as the models in [17] and [18], our architecture achieved comparable results in terms of F1 Score and accuracy, particularly for the full dataset. For example, FT CNN ROI attained an F1 Score of 0.8408, which is very close to the performance of the referenced models, demonstrating that our simpler architecture is still highly effective in recognizing PSL gestures.

The fine-tuning process of the FT CNN ROI model involved optimizing several key hyperparameters, which contributed to the improved performance. Specifically, the best model used 112 filters for both the first and second convolutional layers, with dropout rates of 0.1 and 0.3, respectively. The dense layer contained 256 units with a dropout rate of 0.3, and the learning rate was fine-tuned to 0.0002. These hyperparameters allowed the model to generalize better without overfitting, especially when tested on the full dataset.

In contrast, the other CNN models were trained with consistent hyperparameters to ensure fair comparisons. For the CNN LM model, 32 filters were used in both convolutional layers, with a dropout rate of 0.25, and the dense layer contained 128 units with a dropout rate of 0.5. The CNN ROI Hermite model followed a similar architecture but included Hermite curves, which slightly improved the results in certain cases, particularly for the most predominant classes.

4.3 SqueezeNet Implementation

In this section, we present our implementation of the SqueezeNet model, a convolutional neural network architecture designed for efficiency with a reduced number of parameters compared to traditional CNNs. SqueezeNet is particularly well-suited for our task of facial expression recognition in Portuguese Sign Language (PSL) due to its lightweight design and the use of innovative "fire modules," which allow the model to achieve high performance without large computational overhead.

4.3.1 Model Architecture

Our SqueezeNet model is designed to efficiently perform facial expression recognition by using a compact architecture with fewer parameters than traditional CNNs. SqueezeNet achieves this efficiency through the use of "fire modules," which balance between model size and performance, making it particularly suitable for tasks like Portuguese Sign Language (PSL) recognition.

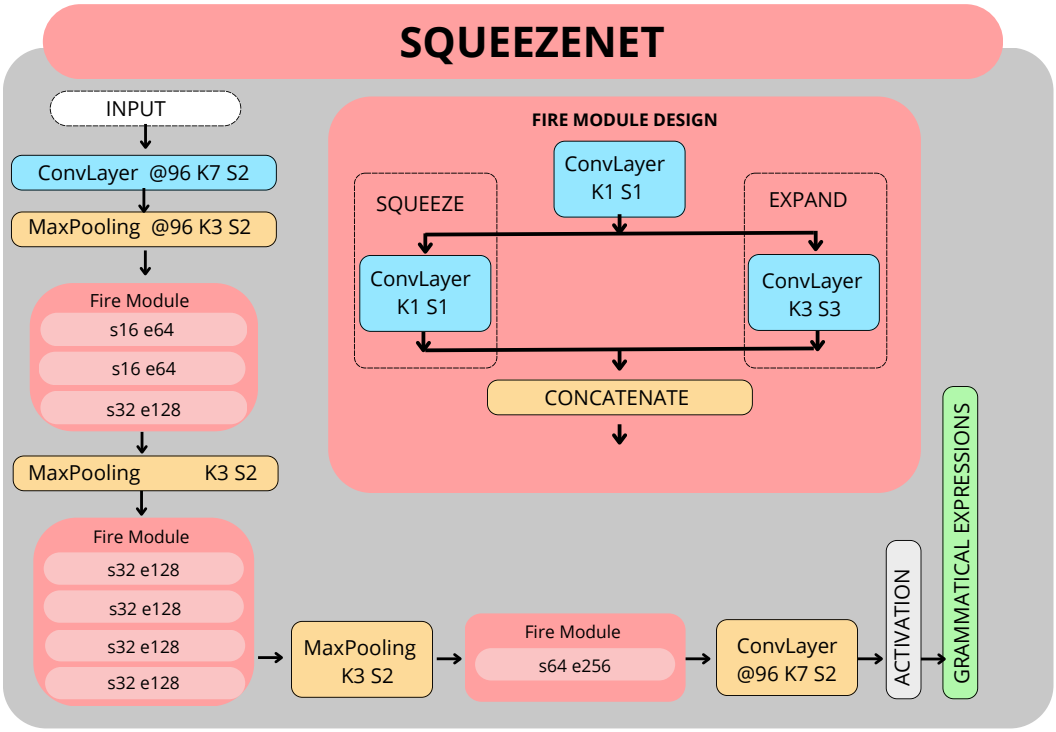


Figure 4.5: SqueezeNet Arquitecture

4.3.1.1 Layer Description

The architecture of SqueezeNet is structured in several distinct layers. The input layer accepts images sized 96×96 pixels, each with 3 color channels (RGB). The initial convolutional layer applies 96 filters with a size of 7×7, using strides of (2, 2), and a ReLU activation function. Padding is set to 'same' to preserve the spatial dimensions. Following this, a max-pooling layer is used, employing a 3×3 pooling window with strides of (2, 2) to reduce the spatial size of the feature maps.

The core of the SqueezeNet model consists of several fire modules. Each fire module starts with a squeeze layer, which is a 1×1 convolutional layer that reduces the number of input channels using ReLU activation. This is followed by two expand layers: one is a 1×1 convolution and the other is a 3×3 convolution, both using ReLU. The outputs from these two expand layers are then concatenated to form the final output of the fire module. As the network progresses, the complexity of these modules increases, with configurations like (16, 64), (32, 128), and (64, 256) for the squeeze and expand layers, respectively.

Throughout the network, intermediate max-pooling layers are placed after certain fire modules to further reduce the spatial dimensions. The final convolutional layer in the network has a number of filters equal to the number of classes being predicted, using a 1×1 filter size and ReLU activation.

To prepare the features for classification, a global average pooling layer is applied, which reduces each feature map to a single value. This simplifies the output to a vector that can be passed into the final softmax layer, which provides a probability distribution across the classes.

This architecture, particularly the use of fire modules, achieves a good balance between model efficiency and performance, keeping the number of parameters relatively low while still maintaining competitive accuracy.

In our SqueezeNet implementation, we apply regularization and optimization strategies similar to those used in our previous CNN model, with adjustments to leverage SqueezeNet's unique architecture:

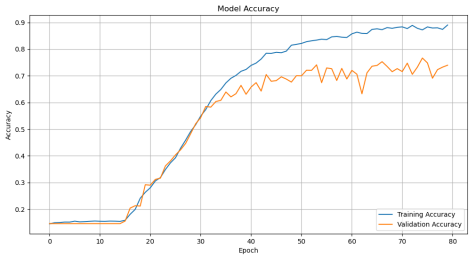
- **Dropout Layers**:
  - While not explicitly included in the fire modules, dropout can be added after certain layers to reduce overfitting, similar to the previous CNN implementation. Dropout rates of 0.25 or 0.5 can be used based on dataset size and complexity.
- **Optimization with SGD**:
  - We employ the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01 and a momentum of 0.9. This choice mirrors the previous CNN setup, where SGD effectively guided model convergence.
- **Regularization Techniques**:
  - Given SqueezeNet's reduced parameter count, the need for aggressive regularization is mitigated. However, L2 regularization can still be applied to certain dense layers to encourage model simplicity.

Overall, the integration of fire modules and effective optimization strategies allows SqueezeNet to maintain robust performance, highlighting the advantages of parameter-efficient architectures in modern machine learning tasks.

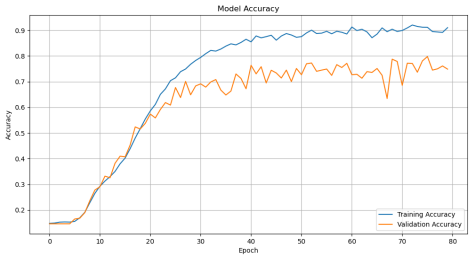
4.3.2 Results and Evaluation

4.3.2.1 Model Performance

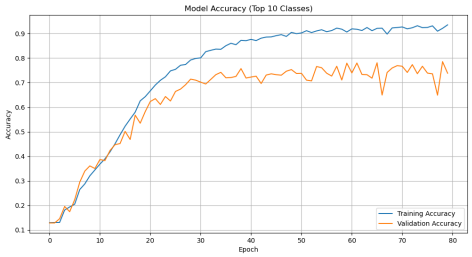
Accuracy and Loss Curves The training and validation accuracy and loss curves for each of the four SqueezeNet models are presented below.



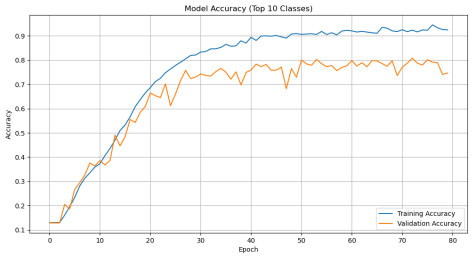
(a) Accuracy for Full Dataset with LM



(b) Accuracy for Full Dataset with ROI



(c) Accuracy for Top 10 Classes with LM



(d) Accuracy for Top 10 Classes with ROI

Figure 4.6: Training and Validation Accuracy for Different SqueezeNet Models

4.3.2.2 Classification Metrics

Precision, Recall, and F1 Score The performance of the SqueezeNet models was evaluated using precision, recall, and F1-score for each class. These metrics provide detailed insights into how well the models differentiate between various facial expressions.

The models performed particularly well on the top 10 classes, achieving higher precision and recall values, which translated into better F1-scores. This indicates a strong capability of the models to accurately identify and classify the most predominant facial expressions in the dataset.

Average Metrics The macro and weighted averages for precision, recall, and F1-score were calculated to provide an overall evaluation of the models:

- Macro Average: This metric averages the performance across all classes equally. The macro average F1-score was slightly lower, reflecting challenges in classifying less frequent classes.

- Weighted Average: This metric accounts for the distribution of the dataset by giving more weight to classes with more samples. The weighted average F1-score was higher, demonstrating robust performance, particularly on more common classes.

The table below summarizes the performance of the SqueezeNet models across different datasets and class distributions:

Table 4.2: Performance of different SqueezeNets for PSL Recognition

Architecture	Dataset	F1 Score	Accuracy	Precision	Recall
SqueezeNet LM	Full Dataset	0.7488	0.7489	0.7533	0.7489
	Top 10 Classes	0.7602	0.7594	0.7682	0.7594
SqueezeNet ROI	Full Dataset	0.7411	0.7423	0.7553	0.7423
	Top 10 Classes	0.7592	0.7593	0.7621	0.7593
SqueezeNet LM Hermite	Full Dataset	0.7440	0.7432	0.7558	0.7432
	Top 10 Classes	0.7710	0.7718	0.7775	0.7718
SqueezeNet ROI Hermite	Full Dataset	0.7474	0.7470	0.7582	0.7470
	Top 10 Classes	**0.7954**	0.7943	0.8029	0.7943
SqueezeNet LM Hermite on Right	Full Dataset	0.7310	0.7277	0.7470	0.7277
	Top 10 Classes	0.6723	0.6751	0.6896	0.6751
SqueezeNet ROI Hermite on Right	Full Dataset	**0.7682**	0.7680	0.7719	0.7680
	Top 10 Classes	0.7662	0.7660	0.7785	0.7660
SqueezeNet [18]	Upper face	0.8525	0.8991	0.8956	0.8134
	Lower face	0.8450	0.8635	0.8840	0.8092
	Avg.	**0.84**	0.88	0.85	0.71

4.3.2.3 Comparative Analysis

Comparison with Baseline Models The performance of the SqueezeNet models was evaluated against the baseline models referenced in the literature, particularly focusing on their efficiency in handling PSL recognition tasks. While the reference models, such as those from [[18]], demonstrated solid results in action unit detection, they incorporated more complex preprocessing techniques and dealt with broader facial expression tasks, which differ from the more specific PSL-related facial expression recognition that our models target.

Despite this difference in focus, the SqueezeNet architectures we implemented performed competitively. In particular, SqueezeNet ROI Hermite emerged as a standout, especially on the top 10 classes, where it achieved an F1 Score of 0.7954. This result highlights the model's strength in identifying the most predominant PSL expressions, even when geometrical data like Hermite curves is integrated. This finding underscores that the inclusion of Hermite curves can be valuable in certain configurations, specifically for models that focus on regions of interest, without overly complicating the input.

Generally, the SqueezeNet LM and SqueezeNet ROI models performed well, although their results were slightly lower than those of the reference models in terms of F1 score. This

positions the SqueezeNet architecture as a viable alternative for applications where model efficiency and speed are crucial factors.

A notable observation was the consistent improvement in results when focusing on the top 10 most frequent classes, a trend seen across all SqueezeNet models. The simpler class distribution allowed for better model generalization and accuracy, as indicated by higher precision and recall metrics. This is in line with our expectations and mirrors the improvements seen in the CNN models as well. The F1 scores for the full dataset were generally lower, reflecting the challenge of dealing with imbalanced datasets that contain underrepresented classes.

In contrast, adding geometrical information such as landmarks did not consistently improve performance, which aligns with the findings in the CNN models. SqueezeNet LM Hermite models, which incorporated Hermite curves overlaid on landmarks, showed moderate gains in certain cases, but these were not as substantial as expected. The SqueezeNet ROI Hermite on Right model, however, exhibited a more marked improvement, where it achieved an F1 Score of 0.7682 on the full dataset. This suggests that the inclusion of Hermite curves, when appropriately positioned, can provide useful supplementary information to enhance model accuracy.

The models that did not use landmarks, particularly the SqueezeNet ROI variants, generally performed better, reinforcing the idea that simplifying the input by focusing on key regions of interest without introducing additional complexity can lead to better results.

4.4 LSTM Implementation

The LSTM model's performance did not meet our expectations, primarily due to the limited dataset available for training and the inherent challenges of modeling temporal dynamics with insufficient data. The dataset used for training the LSTM model was derived from video sequences that captured facial expressions. Each video sequence was split into frames, which were then grouped into sequences of 5 frames each. This approach aimed to retain the temporal dynamics inherent in facial expressions. However, this method significantly reduced the overall size of the dataset, as only sequences with at least 5 frames could be utilized.

4.4.1 LSTM Model Architecture

The LSTM model was specifically designed to capture both the spatial and temporal dynamics of facial expressions across video sequences. The architecture includes three convolutional layers, each with 32, 64, and 128 filters, respectively. After each convolutional layer, a max-pooling layer is applied to reduce the spatial dimensions while retaining the important features

from the individual frames of the video sequences. These convolutional layers are used to extract spatial features from the images, which are then passed on to a 64-unit LSTM layer. Data preparation involved several key steps to ensure the model could accurately interpret the facial features and temporal aspects. First, the raw image data, stored as matrix-like strings, was cleaned and converted into numpy arrays representing the images. These arrays were reshaped into 96x96x3 dimensions to be suitable for input into the model. The facial expression labels were then encoded into numerical values using the LabelEncoder. Following this, the labels were further transformed into categorical format through one-hot encoding, which is essential for multi-class classification tasks. To preserve the temporal dynamics of the video sequences, the frames were grouped into sequences of five. This approach allowed the model to consider how facial expressions evolved over time within a short segment of the video. Lastly, the dataset was split into training, validation, and test sets using a 70-30 split ratio. The training set was also split further, creating a validation set to be used for tuning the model's hyperparameters during training.

4.4.2 Training and Evaluation

The LSTM model was trained using Stochastic Gradient Descent (SGD) as the optimizer, with a learning rate of 0.01 and momentum of 0.9. The categorical cross-entropy loss function was used, appropriate for multi-class classification tasks. The model was trained with a batch size of 64 over 50 epochs.

During training, the model's performance on the validation set was continuously monitored. However, the model's performance plateaued early, demonstrating limited improvement over the subsequent epochs. The final training accuracy was 36.84%, while the validation and test accuracies were 27.27% and 30.43%, respectively. The macro average F1 score across classes was approximately 0.27, with significant variance in F1 scores between different classes.

The results indicate that the LSTM model struggled to effectively capture the nuances of facial expressions in PSL videos. The primary challenges include:

- **Limited Data**: The small size of the dataset, particularly after splitting into sequences as it excluded all the annotations that did not have at least 5 frames and the own dataset after that would be at least 5 times smaller, reduced the model's ability to generalize.
- **Complexity of the Task**: Modeling temporal dynamics in facial expressions is inherently complex and requires large, diverse datasets to achieve high accuracy.

Future work could focus on expanding the dataset and exploring alternative architectures or data augmentation techniques to improve model performance.

4.4.3 Training

This section provides a comprehensive overview of the data preparation, training configuration, hyperparameter selection, and training procedures that were consistently applied across both the CNN and SqueezeNet models. These processes form the backbone of our approach, ensuring that both models were trained under similar conditions to allow for a fair comparison of their performance.

4.4.3.1 Data Preprocessing

The raw image data, stored as matrix-like strings within the dataset, was first converted into a usable format. This conversion process involved several steps:

- **String Cleaning**: The matrix-like strings were cleaned by removing unnecessary characters such as newlines and brackets. This step was essential to ensure that the data could be accurately parsed into numerical values.
- **String to Numeric Conversion**: The cleaned strings were then split into individual pixel values. These values represent the intensity of each pixel in the RGB color space and were converted into a 1D list of numbers.
- **Reshaping**: The 1D list of pixel values was reshaped into a 96x96x3 numpy array, representing the image in a format suitable for input into the neural network. This array format corresponds to the height, width, and color channels (RGB) of the image.
- **Normalization**: The pixel values were normalized by scaling them to the range [0, 1] by dividing by 255. This step is crucial for ensuring that the network can learn efficiently, as it stabilizes the gradients during backpropagation and speeds up the convergence of the model.

4.4.3.2 Data Splitting

After preprocessing, the dataset was split into three subsets: training, validation, and testing. The split was done in a manner that preserved the distribution of classes across each subset, ensuring that no subset was disproportionately represented.

- **Training Set (70%)**: This subset, comprising 70% of the total data, was used to train the models. It provided a diverse set of examples for the models to learn from, covering the full range of variations in the data.
- **Validation Set (15%)**: The validation set, consisting of 15% of the data, was used to tune hyperparameters and monitor the models' performance during training. It served as a

check against overfitting, allowing us to adjust the models before they were evaluated on the test set.

- **Test Set (15%)**: The remaining 15% of the data was reserved for testing. This set was not used during training and provided an unbiased evaluation of the models' performance on unseen data.

Class distribution was carefully monitored to ensure that all subsets were representative of the overall dataset. This was especially important in our case, where some classes were more prevalent than others, which could potentially bias the models if not handled correctly.

4.4.3.3 Training Configuration

The configuration of the training process was carefully designed to optimize the performance of both the CNN and SqueezeNet models. This section details the key decisions made regarding hyperparameters, training procedures, and the strategies employed to mitigate issues such as overfitting.

4.4.3.4 Hyperparameter Selection

Selecting the appropriate hyperparameters is critical for the successful training of deep learning models. The following hyperparameters were chosen based on initial experiments and best practices from the literature:

- **Batch Size**: A batch size of 32 was used for both models. This size was chosen as a compromise between training speed and the stability of the gradient updates. Smaller batches can lead to noisy updates, while larger batches require more memory and can slow down training.
- **Number of Epochs**: The models were trained for 80 epochs. This number was determined based on early stopping criteria observed during initial tests. Training for too few epochs could lead to underfitting, while too many epochs could result in overfitting.
- **Learning Rate**: The learning rate for the Stochastic Gradient Descent (SGD) optimizer was set to 0.01. This rate was selected to ensure a balance between the speed of convergence and the stability of the training process. A momentum of 0.9 was also employed to help the optimizer avoid local minima and smooth out the updates.
- **Dropout Rates**: Dropout layers were included in both models to prevent overfitting. For the CNN, dropout rates of 0.25 and 0.5 were used after the convolutional and dense layers, respectively. These rates were chosen to maintain a balance between model complexity and the need to prevent overfitting.

- **L2 Regularization**: L2 regularization was applied to the dense layers in both models. This regularization helps prevent the model from learning overly complex functions that may not generalize well to new data by penalizing large weights.

4.4.3.5 Training Procedure

The training procedure was standardized across both models to ensure a fair comparison of their performance. The key steps in the training process included:

- **Initialization**: Both models were initialized with weights using the Glorot uniform initializer, which is particularly effective for deep networks as it maintains the variance of the weights across layers.
- **Early Stopping**: Early stopping was employed based on the performance on the validation set. If the validation loss did not improve for a predefined number of epochs (patience), training was stopped to prevent overfitting.
- **Learning Rate Scheduling**: A learning rate scheduler was used to reduce the learning rate by a factor of 0.1 if the validation loss plateaued. This strategy helped in refining the models during the later stages of training, allowing them to converge to a better solution.
- **Time Management**: To manage the training process efficiently, the `TimeHistory` callback was used to estimate the remaining time for each epoch. This allowed for better planning and allocation of computational resources.
- **Data Augmentation (for CNN only)**: Data augmentation techniques such as random rotations, shifts, and flips were applied to the training data for the CNN model. These techniques help increase the diversity of the training data and reduce overfitting by making the model robust to variations in the input data.

Both models were trained using the categorical cross-entropy loss function, which is appropriate for multi-class classification tasks. The softmax activation function in the output layer ensured that the models produced a probability distribution across all classes, which the loss function used to compute the error.

We incorporated regularization and optimization techniques to enhance the model's generalization capabilities and avoid overfitting. The model architecture is as follows:

- **Dropout Layers**: Dropout is employed after both the convolutional and dense layers to prevent overfitting by randomly setting a fraction of input units to zero during training. We applied dropout rates of 0.25 and 0.5, allowing the model to learn more robust features by discouraging reliance on any particular neurons.

- **L2 Regularization**: L2 regularization is applied to the dense layer to penalize large weights. This technique helps in constraining the complexity of the model, promoting simpler models that generalize better on unseen data.
- **Optimization with SGD**: The model is optimized using the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01 and a momentum of 0.9.x

While initial experiments suggested that regularization might not be necessary with small amounts of data, subsequent tests demonstrated its importance as the dataset size increased. For example, training with a larger dataset of 5,000 samples without regularization led to overfitting, as evidenced by the discrepancy between training and validation accuracy. The model without regularization quickly reached high training accuracy, yet its validation accuracy plateaued at a lower value, indicating poor generalization to new data.

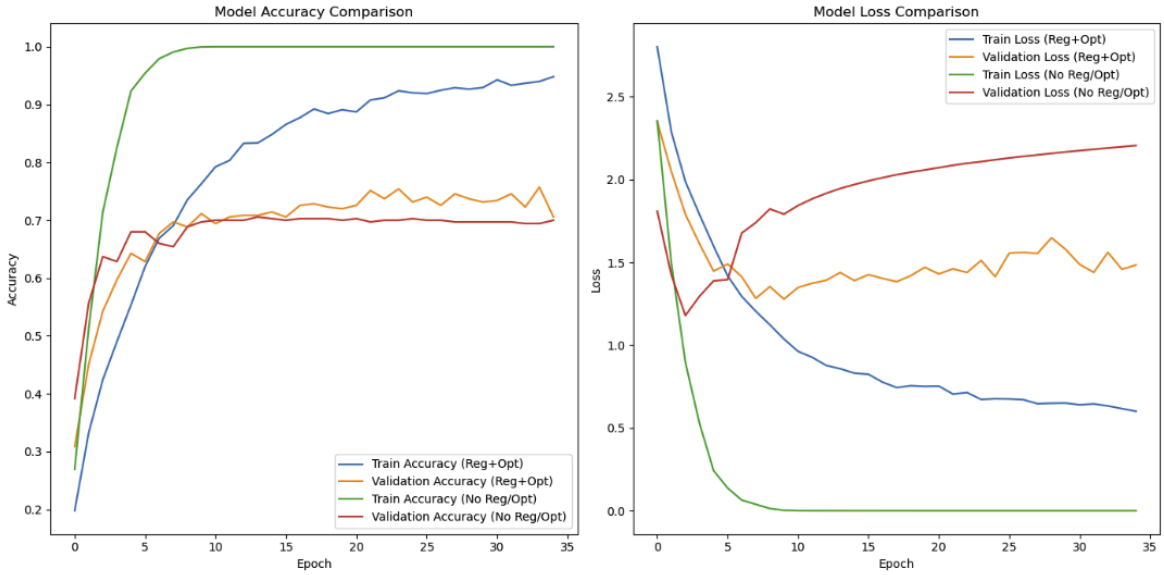


Figure 4.7: Comparison of Training and Validation Accuracy with and without Regularization

As shown in Figure 4.7, the model with regularization maintained a more consistent performance across training and validation datasets, highlighting its effectiveness in preventing overfitting. This confirms that regularization techniques are crucial for enhancing model robustness and performance, particularly when dealing with larger datasets. Our choice of regularization and optimization techniques was informed by the need to balance model complexity with generalization capability. These strategies ensured that the model learned meaningful patterns without memorizing the noise inherent in the training data. Thus, the integration of dropout and L2 regularization, coupled with a careful selection of optimizer settings, played a pivotal role in achieving optimal performance. By applying these standardized training configurations, we ensured that both the CNN and SqueezeNet models were

trained under optimal and comparable conditions. This consistency is crucial for drawing meaningful conclusions from the comparison of their performance.

4.4.3.6 Fine-Tuning and Performance Enhancement

In this research, we fine-tuned several aspects of our models to better suit the specific characteristics of our dataset. Initially, we utilized Keras Tuner to systematically explore and identify optimal settings for several hyperparameters, including convolutional filters, dropout rates, and dense layer sizes. This methodical approach was essential in determining configurations that offered the best performance for our data.

To address challenges with convergence and local minima, we implemented an adaptive learning rate strategy during training. This strategy adjusted the learning rate based on the validation performance, enhancing the model's ability to improve incrementally without plateauing.

Furthermore, to mitigate the risk of overfitting, we intensified our regularization strategies. By fine-tuning the L2 regularization parameters and experimenting with varying dropout rates at critical points in the model architecture, we ensured that our models generalized well to new data, avoiding the pitfall of memorizing the training set.

We also expanded our data augmentation techniques. By incorporating more advanced methods such as elastic deformations and noise injection, we prepared our models to handle a broader range of variations in input data, thus improving their robustness and ability to generalize from the training scenarios to real-world applications.

These targeted adjustments not only tailored the training process to our specific needs but also significantly enhanced the models' accuracy and generalization capabilities across various tests.

4.4.3.7 Evaluation Metrics

During training, the primary metric used to evaluate the models was accuracy, which measures the proportion of correct predictions. However, given the imbalance in the dataset, precision, recall, and F1-score were also calculated during the final evaluation to provide a more nuanced understanding of the models' performance across different classes. These metrics are particularly important for assessing how well the models handle the minority classes, which are often more challenging to predict accurately.

The results obtained from these evaluations are discussed in the respective sections for CNN and SqueezeNet, where we compare their performances based on these metrics and analyze the implications of the differences observed.

4.5 Discussion of Results

The results presented in this thesis offer valuable insights into the effectiveness of different model architectures in recognizing facial expressions in Portuguese Sign Language (PSL). Across all models, several trends emerged, shedding light on the strengths and limitations of the tested configurations.

One of the most consistent findings was the strong performance of models using images without facial landmarks (LM). These models, particularly the CNN and SqueezeNet variants, consistently outperformed their counterparts that included landmark data. This suggests that in the context of PSL facial expression recognition, the raw appearance-based features may provide more salient information than explicitly defined facial landmarks. This trend was expected, as facial landmarks, while useful in some contexts, can introduce noise or unnecessary complexity when the model's task is to focus on the overall facial expression.

An interesting exception occurred with the inclusion of Hermite curves in the images. The models using Hermite curves, particularly when overlaid on the right side of the face, showed some performance improvement. This suggests that while general facial landmarks might not be useful, geometrical features like Hermite curves can enhance the model's understanding of facial expression dynamics by capturing temporal and spatial changes in facial muscles. This finding is particularly relevant for PSL, where subtle changes in expression often carry significant meaning.

The performance of models on the top 10 most frequent classes was, as expected, significantly better than on the full dataset. This improvement highlights the importance of reducing class imbalance when training on diverse datasets like PSL. By focusing on the most common classes, the models were able to generalize more effectively, achieving higher F1 scores, precision, and recall. The CNN ROI model, for instance, achieved an F1 score of 0.8376 on the top 10 classes, demonstrating its strong capacity for recognizing the most frequent facial expressions.

However, the addition of geometrical data, aside from Hermite curves, did not consistently yield better results. Models incorporating both facial landmarks and geometrical data, such as the CNN LM Hermite model, generally performed worse than those relying on raw image data or images with Hermite curves only. This suggests that while some geometrical features can be helpful, the complexity added by combining multiple sources of geometrical data may hinder rather than improve the model's performance in this specific task.

In comparison to the reference models from the literature, such as the CNN models from [18] and [17], the results of this study show that while these reference models achieved slightly higher performance metrics, the architectures presented here performed competitively, particularly given the streamlined preprocessing steps and the use of fewer parameters. The fine-tuned CNN ROI model, for example, achieved an F1 score of 0.8408 on the full

dataset, coming close to the results obtained by more complex models in the literature. This demonstrates that with appropriate fine-tuning, even simpler architectures can perform effectively in the task of PSL recognition.

The fine-tuning process played a crucial role in the success of the CNN ROI model. By optimizing key hyperparameters, such as the number of filters, dropout rates, and learning rate, the model was able to generalize better and avoid overfitting, particularly when tested on the full dataset. This highlights the importance of careful hyperparameter selection and fine-tuning in achieving optimal performance in deep learning tasks.

Another key takeaway from the results is the role of dataset composition in model performance. The challenges posed by class imbalance were evident, especially when the models were trained on the full dataset, which contained many underrepresented classes. While the models were able to achieve reasonable accuracy and F1 scores on the full dataset, their performance was significantly better when focusing on the top 10 classes. This underscores the importance of addressing dataset imbalance through techniques such as oversampling, data augmentation, or even the development of more targeted datasets that balance class representation.

Finally, the inclusion of temporal dynamics through Hermite curves was one of the more successful enhancements in this study. By overlaying the Hermite curve on the right side of the face, the models were able to capture subtle facial movements over time, which proved beneficial for the task at hand. This suggests that for future work, further exploration of temporal modeling techniques, such as integrating recurrent layers or attention mechanisms, could yield even better results in recognizing the dynamic facial expressions associated with PSL.

In summary, the results of this thesis show that while complex preprocessing steps and landmark-based models can sometimes introduce unnecessary complexity, simplified models focusing on appearance-based features and the selective use of geometrical enhancements, like Hermite curves, can lead to strong performance in PSL recognition. The fine-tuning of hyperparameters and careful consideration of dataset composition were also key factors in achieving competitive results compared to existing models in the literature. This work lays the foundation for further exploration of more advanced architectures and techniques, such as transformers or recurrent neural networks, which could build upon the successes observed here and push the boundaries of facial expression recognition in PSL.

Table 4.3: Performance of different SqueezeNets, CNNs and based models for SL Recognition

Architecture	Dataset	F1 Score	Accuracy	Precision	Recall
SqueezeNet LM	Full Dataset	0.7488	0.7489	0.7533	0.7489
	Top 10 Classes	0.7602	0.7594	0.7682	0.7594
SqueezeNet ROI	Full Dataset	0.7411	0.7423	0.7553	0.7423
	Top 10 Classes	0.7592	0.7593	0.7621	0.7593
SqueezeNet Hermite	Full Dataset	0.7440	0.7432	0.7558	0.7432
	Top 10 Classes	0.7710	0.7718	0.7775	0.7718
SqueezeNet ROI Hermite	Full Dataset	0.7474	0.7470	0.7582	0.7470
	Top 10 Classes	0.7954	0.7943	0.8029	0.7943
SqueezeNet LM Hermite on Right	Full Dataset	0.7310	0.7277	0.7470	0.7277
	Top 10 Classes	0.6723	0.6751	0.6896	0.6751
SqueezeNet ROI Hermite on Right	Full Dataset	0.7682	0.7680	0.7719	0.7680
	Top 10 Classes	0.7662	0.7660	0.7785	0.7660
CNN LM	Full Dataset	0.7122	0.7157	0.7494	0.7157
	Top 10 Classes	0.7425	0.7436	0.7562	0.7436
CNN ROI	Full Dataset	0.7896	0.8144	0.7553	0.7899
	Top 10 Classes	0.8376	0.8371	0.8457	0.8371
CNN LM Hermite	Full Dataset	0.7019	0.7035	0.7361	0.7035
	Top 10 Classes	0.7621	0.7625	0.7729	0.7625
CNN ROI Hermite	Full Dataset	0.7503	0.7517	0.7854	0.7517
	Top 10 Classes	0.7859	0.8069	0.7854	0.7845
CNN LM Hermite on Right	Full Dataset	0.7055	0.7092	0.7302	0.7092
	Top 10 Classes	0.7728	0.7721	0.7786	0.7721
CNN ROI Hermite on Right	Full Dataset	0.6838	0.6886	0.7520	0.6886
	Top 10 Classes	0.7858	0.7855	0.7887	0.7858
SqueezeNet [18]	Upper face	0.8525	0.8991	0.8956	0.8134
	Lower face	0.8450	0.8635	0.8840	0.8092
	Avg.	0.84	0.88	0.85	0.71
CNN [18]	Upper face	0.7516	0.7999	0.8564	0.6697
	Lower face	0.7351	0.7257	0.8187	0.6670
	Avg.	0.74	0.76	0.83	0.66
CNN [17]	Upper face	0.8900	0.9018	0.8972	0.8194
	Lower face	0.8522	0.8585	0.8892	0.8091
	Avg.	0.8711	0.8805	0.8932	0.8142

5

Conclusions and Future Work

5.1 Conclusion

This thesis represents a pioneering step in the recognition of facial expressions for grammatical purposes in Portuguese Sign Language (PSL), a domain that has not been extensively explored until now. By focusing on the facial dynamics tied to PSL, we have contributed to the growing field of non-manual markers in sign language recognition. Unlike other studies that primarily focus on manual gestures, our work sheds light on the critical role facial expressions play in PSL grammar, highlighting the complexity and richness of non-verbal communication in sign language.

Our exploration of convolutional neural networks (CNNs) and SqueezeNet architectures has demonstrated the feasibility of using relatively lightweight models to recognize and classify PSL facial expressions with promising accuracy. While deeper architectures like AlexNet performed well, our optimized models, especially the CNN ROI and SqueezeNet variants, showed competitive results, proving that efficient models can still capture the subtleties of PSL facial expressions. The inclusion of Hermite curves, designed to enhance the temporal understanding of facial dynamics, further underscored the importance of integrating both spatial and temporal data for more accurate recognition.

This research stands as the first attempt to specifically address facial expression recognition within the context of PSL grammar. By leveraging facial landmarks, emotion detection, and action units (AUs), we created a dataset and methodology that lays the groundwork for future studies in this field. Moreover, our approach to addressing class imbalance through targeted data augmentation and focusing on top-performing classes proved effective, showcasing how small but deliberate adjustments can improve model performance.

As this thesis demonstrates, facial expression recognition in PSL is both a viable and crucial area for further research. The results achieved here not only validate the importance of facial dynamics in PSL but also open the door for more sophisticated future studies that

can build on this foundation. Larger datasets, advanced architectures like transformers, and the integration of multimodal data (manual and non-manual markers) are promising next steps that could significantly enhance the capabilities of PSL recognition models.

Ultimately, this work aims to contribute to giving the first steps of improving accessibility and communication tools for the Deaf community, especially in Portugal, a region where there exists a lack of research in the field. By advancing the recognition of PSL's grammatical facial expressions, we move closer to developing practical, real-time applications that can aid in education, public services, and communication. This thesis stands as a crucial first step in a growing field, with significant potential for impact in both academic and practical applications of sign language recognition.

5.2 Future Work

This work is a good starting point for Portuguese Sign Language (PSL) recognition, but several areas need further research. One key aspect is the dataset. The current dataset, while helpful, is limited in size and variation. For better generalization, a larger dataset covering more subjects, lighting conditions, and various expressions is needed. Extending the dataset to include more facial action units and different non-manual signals would improve model performance significantly.

Another promising area for future work is the exploration of more advanced models, particularly transformer-based architectures. Transformers, which have shown great results in other fields, could be adapted for PSL due to their ability to capture temporal dependencies in sequential data. Models like Vision Transformers (ViT) or hybrids that combine convolutional layers with attention mechanisms could help capture both the spatial and temporal features more effectively. Additionally, utilizing generative models for data augmentation would enrich the dataset without needing to gather more real-world data.

We also suggest integrating pre trained models like CLIP or modified versions such as CLIPER, adapted to PSL. These models could be retrained on a PSL-specific dataset, generating more detailed embeddings of sign language gestures and facial expressions. Future research should focus on evaluating these models in real-time applications, ensuring they are ready for practical use cases in areas where PSL is commonly needed.

Bibliography

[1] A. G. H. M. Z. B. C. D. K. W. W. T. W. M. A. H. Adam. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". In: (2017) (cit. on p. 11).

[2] T. Adão et al. "Empowering Deaf-Hearing Communication: Exploring Synergies between Predictive and Generative AI-Based Strategies towards (Portuguese) Sign Language Interpretation". In: *Journal of Imaging* 9.11 (2023). ISSN: 2313-433X. DOI: 10.3390/jimaging9110235. URL: https://www.mdpi.com/2313-433X/9/11/235 (cit. on p. 5).

[3] S. N. Anusha Puchakayala and P. K. "American Sign language Recognition using Deep Learning". In: (2023) (cit. on p. 23).

[4] "Baltazar, 2010". In: (2010) (cit. on p. 5).

[5] T. Baltrusaitis, C. Ahuja, and L. Morency. "Multimodal Machine Learning: A Survey and Taxonomy". In: *CoRR* abs/1705.09406 (2017). arXiv: 1705.09406. URL: http://arxiv.org/abs/1705.09406 (cit. on pp. 16, 17).

[6] K. Bantupalli and Y. Xie. "American Sign Language Recognition using Deep Learning and Computer Vision". In: *2018 IEEE International Conference on Big Data (Big Data)*. 2018, pp. 4896–4899. DOI: 10.1109/BigData.2018.8622141 (cit. on pp. 23, 24).

[7] B. Bauer and H. Hienz. "Relevant Features for Video-Based Continuous Sign Language Recognition". In: (2021) (cit. on p. 25).

[8] "Bettencourt, 2015;" in: (2015) (cit. on p. 5).

[9] S. Chen et al. *MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices*. 2018. arXiv: 1804.07573 [cs.CV] (cit. on p. 11).

[10] J. H. Cheong et al. "Py-Feat: Python Facial Expression Analysis Toolbox". In: *CoRR* abs/2104.03509 (2021). arXiv: 2104.03509. URL: https://arxiv.org/abs/2104.03509 (cit. on p. 9).

[11] "Choupina et al., 2017". In: (2017) (cit. on p. 5).

[12] "Choupina, 2017;" in: (2017) (cit. on p. 5).

[13] A. Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *CoRR* abs/2010.11929 (2020). arXiv: 2010.11929. URL: https://arxiv.org/abs/2010.11929 (cit. on p. 18).

[14] E. A. Elliott and A. M. Jacobs. "Facial expressions, emotions, and sign languages". In: (2013) (cit. on p. 6).

[15] E. A. Elliott and A. M. Jacobs. "Facial expressions, emotions, and sign languages". In: *FRONTIERS IN PSYCHOLOGY* 4 (2013-03). ISSN: 1664-1078. DOI: 10.3389/fpsyg.2013.00115 (cit. on pp. 5, 26).

[16] K. K. Emely Pujólli da Silva Paula Costa and J. Martino. "Facial action unit detection methodology with application in Brazilian sign language recognition". In: (2021) (cit. on pp. 2, 7, 11, 21, 26, 35, 36).

[17] K. M. O. K. Emely Pujolli da Silva Paula Dornhofer Paro Costa, J. M. D. Martino, and G. A. Florentino. "Recognition of Affective and Grammatical Facial Expressions: A Study for Brazilian Sign Language". In: (2020) (cit. on pp. 2, 11, 22, 24, 26, 35, 36, 38, 43, 44, 56, 58).

[18] S. P. Fernando Freitas and C. A. Lima. "Grammatical facial expression recognition in sign language discourse: a study at the syntax level". In: (2017) (cit. on pp. 2, 21, 23, 38, 43, 44, 48, 56, 58).

[19] "Ferreira,1997". In: (1997) (cit. on p. 5).

[20] F. Freitas et al. "Grammatical facial expressions recognition with machine learning". In: (2014-08), pp. 180–185 (cit. on pp. 2, 14, 25–27).

[21] E. Goncalves and M. J. C. Raposo. "Facial expressions in grammatical morphology of Portuguese Sign Language – Expressions of degrees of augmentative and diminutive size in LGP". In: (2013) (cit. on p. 5).

[22] C. Gong et al. "Multi-Modal Curriculum Learning for Semi-Supervised Image Classification". In: *IEEE TRANSACTIONS ON IMAGE PROCESSING* 25.7 (2016-07), pp. 3249–3260. ISSN: 1057-7149. DOI: 10.1109/TIP.2016.2563981 (cit. on p. 16).

[23] "Grammatical facial expression recognition in sign language discourse a study at the syntax level". In: (2017) (cit. on pp. 2, 14).

[24] Y. Gu et al. "American Sign Language Translation Using Wearable Inertial and Electromyography Sensors for Tracking Hand Movements and Facial Expressions". In: *Frontiers in Neuroscience* 16 (2022-07), p. 962141. DOI: 10.3389/fnins.2022.962141 (cit. on pp. 24, 26).

[25] X. Guo et al. *PFLD: A Practical Facial Landmark Detector*. 2019. arXiv: 1902.10859 [cs.CV] (cit. on p. 11).

[26] K. Han et al. "A Survey on Vision Transformer". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.1 (2023), pp. 87–110. DOI: 10.1109/TPAMI.2022.3152247 (cit. on pp. 12, 15, 16).

[27] K. He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90 (cit. on p. 18).

[28] J. Huang and V. Chouvatut. "Video-Based Sign Language Recognition via ResNet and LSTM Network". In: *JOURNAL OF IMAGING* 10.6 (2024-06). DOI: 10.3390/jimaging10060149 (cit. on p. 14).

[29] H. A. Ismail, I. A. Hashim, and B. H. Abd. "A Survey on Linguistic Interpretation of Facial Expressions and Technologies". In: *2019 2nd International Conference on Engineering Technology and its Applications (IICETA)*. 2019, pp. 161–166. DOI: 10.1109/IICETA47481.2019.9012983 (cit. on pp. 1, 2, 6, 8, 14).

[30] G. Y. Kebe et al. "Bridging the Gap: Using Deep Acoustic Representations to Learn Grounded Language from Percepts and Raw Speech". In: *CoRR* abs/2112.13758 (2021). arXiv: 2112.13758. URL: https://arxiv.org/abs/2112.13758 (cit. on p. 17).

[31] J. D. J. G. Y. Z. J. Y. I. Kotsia and S. Zafeiriou. "RetinaFace: Single-stage Dense Face Localisation in the Wild". In: (2019) (cit. on p. 10).

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet classification with deep convolutional neural networks". In: *Commun. ACM* 60.6 (2017-05), pp. 84–90. ISSN: 0001-0782. DOI: 10.1145/3065386. URL: https://doi.org/10.1145/3065386 (cit. on p. 12).

[33] P. Kumar, P. P. Roy, and D. P. Dogra. "Independent Bayesian classifier combination based sign language recognition using facial expression". In: *Information Sciences* 428 (2018), pp. 30–48. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2017.10.046. URL: https://www.sciencedirect.com/science/article/pii/S0020025516307897 (cit. on p. 14).

[34] H. Li et al. "CLIPER: A Unified Vision-Language Framework for In-the-Wild Facial Expression Recognition". In: (2023) (cit. on pp. 2, 11, 19, 26).

[35] S. Li and W. Deng. "Deep Facial Expression Recognition: A Survey". In: *CoRR* abs/1804.08348 (2018). arXiv: 1804.08348. URL: http://arxiv.org/abs/1804.08348 (cit. on p. 12).

[36] C. Liu, K. Hirota, and Y. Dai. "Patch attention convolutional vision transformer for facial expression recognition with occlusion". In: *Information Sciences* 619 (2023), pp. 781–794. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2022.11.068. URL: https://www.sciencedirect.com/science/article/pii/S0020025522013573 (cit. on pp. 15, 26).

[37] S. Liu et al. "Multi-modal fusion network with complementarity and importance for emotion recognition". In: *INFORMATION SCIENCES* (2023) (cit. on pp. 16, 17).

[38] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/main/template.pdf (cit. on p. i).

[39] L. C. Matilde Goncalves and H. Nicolau. "PE2LGP: tradutor de portugu^es europeu para l ıngua gestual portuguesa em glosas". In: (2021) (cit. on p. 5).

[40] "Mesquita Silva, 2009". In: (2009) (cit. on p. 5).

[41] S. Namba, W. Sato, and S. Yoshikawa. "Viewpoint Robustness of Automated Facial Action Unit Detection Systems". In: *Applied Sciences* 11.23 (2021). ISSN: 2076-3417. DOI: 10.3390/app112311171. URL: https://www.mdpi.com/2076-3417/11/23/11171 (cit. on pp. 9, 10).

[42] C. Pramerdorfer and M. Kampel. "Facial Expression Recognition using Convolutional Neural Networks: State of the Art". In: *CoRR* abs/1612.02903 (2016). arXiv: 1612.02903. URL: http://arxiv.org/abs/1612.02903 (cit. on p. 11).

[43] A. Radford et al. "Language Models are Unsupervised Multitask Learners". In: 2019. URL: https://api.semanticscholar.org/CorpusID:160025533 (cit. on p. 18).

[44] A. Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV] (cit. on pp. 11, 18, 19, 26).

[45] V. G. Rakesh Kumar Attar and L. Goyal. "State of the Art of Automation in Sign Language: A Systematic Review". In: (2023) (cit. on p. 6).

[46] R. Rastgoo, K. Kiani, and S. Escalera. "Sign Language Recognition: A Deep Survey". In: *Expert Systems with Applications* 164 (2021), p. 113794. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2020.113794. URL: https://www.sciencedirect.com/science/article/pii/S095741742030614X (cit. on pp. 1, 2, 13).

[47] J. M. dos Santos Oliveira. "A Língua Gestual Portuguesa como primeira língua da criança surda". In: (2021) (cit. on p. 5).

[48] S. Tamura and S. Kawasaki. "Recognition of sign language motion images". In: *Pattern Recognition* 21.4 (1988), pp. 343–353. ISSN: 0031-3203. DOI: https://doi.org/10.1016/0031-3203(88)90048-9. URL: https://www.sciencedirect.com/science/article/pii/0031320388900489 (cit. on p. 1).

[49] A. Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762 (cit. on p. 18).

[50] C. Viegas et al. *Including Facial Expressions in Contextual Embeddings for Sign Language Generation*. 2022. arXiv: 2202.05383 [cs.CL] (cit. on pp. 1, 2, 5, 6).

[51] H. Xia et al. *Language and Multimodal Models in Sports: A Survey of Datasets and Applications*. 2024. arXiv: 2406.12252 [cs.CL]. URL: https://arxiv.org/abs/2406.12252 (cit. on p. 17).

[52] F. Xiao et al. "KSRB-Net: a continuous sign language recognition deep learning strategy based on motion perception mechanism". In: *VISUAL COMPUTER* (2023-2023 DEC 26). ISSN: 0178-2789. DOI: 10.1007/s00371-023-03211-3 (cit. on pp. 25, 27).

[53] Z. Yu et al. "Spatio-temporal convolutional features with nested LSTM for facial expression recognition". In: *Neurocomputing* 317 (2018), pp. 50–57. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2018.07.028. URL: https://www.sciencedirect.com/science/article/pii/S0925231218308634 (cit. on pp. 2, 13).

[54] Z. Yu and C. Zhang. "Image based Static Facial Expression Recognition with Multiple Deep Network Learning". In: (2015) (cit. on p. 12).

[55] N. Zhang, J. Luo, and W. Gao. "Research on Face Detection Technology Based on MTCNN". In: *2020 International Conference on Computer Network, Electronic and Automation (ICCNEA)*. 2020, pp. 154–158. DOI: 10.1109/ICCNEA50255.2020.00040 (cit. on pp. 8, 10).

[56] T. Zhang et al. "Spatial–Temporal Recurrent Neural Network for Emotion Recognition". In: *IEEE Transactions on Cybernetics* 49.3 (2019), pp. 839–847. DOI: 10.1109/TCYB.2017.2788081 (cit. on pp. 2, 13).

[57] Y. Zhang et al. "Deep multimodal fusion for semantic image segmentation: A survey". In: *Image and Vision Computing* 105 (2021), p. 104042. ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2020.104042. URL: https://www.sciencedirect.com/science/article/pii/S0262885620301748 (cit. on p. 16).

[58] Y. Zhang et al. "Contrastive Learning of Medical Visual Representations from Paired Images and Text". In: *CoRR* abs/2010.00747 (2020). arXiv: 2010.00747. URL: https://arxiv.org/abs/2010.00747 (cit. on p. 18).

[59] R. Zhi, M. Liu, and D. Zhang. "Facial Representation for Automatic Facial Action Unit Analysis System". In: *PROCEEDINGS OF 2019 IEEE 8TH JOINT INTERNATIONAL INFORMATION TECHNOLOGY AND ARTIFICIAL INTELLIGENCE CONFERENCE (ITAIC 2019)*. Ed. by B. Xu. IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, PEOPLES R CHINA, MAY 24-26, 2019. IEEE; IEEE Beijing Sect; Chongqing Global Union Acad Sci & Technol; Chongqing Univ Technol; Chengdu Global Union Acad Sci & Technol; Chongqing Geeks Educ Technol Co Ltd. 2019, pp. 1368–1372. ISBN: 978-1-5386-8178-7. DOI: 10.1109/itaic.2019.8785870 (cit. on p. 9).