PUCRS

ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

MARCELO MUSSI DELUCIS

ISOLATED SIGN LANGUAGE RECOGNITION IN LIBRAS

Porto Alegre
2025

PÓS-GRADUAÇÃO - *STRICTO SENSU*

AD VERVM DVCIT

Pontifícia Universidade Católica
do Rio Grande do Sul

PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
SCHOOL OF TECHNOLOGY
COMPUTER SCIENCE GRADUATE PROGRAM

ISOLATED SIGN LANGUAGE RECOGNITION IN LIBRAS

MARCELO MUSSI DELUCIS

Master Thesis submitted to the Pontifical Catholic University of Rio Grande do Sul in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Prof. Dr. Lucas Silveira Kupssinskü
Co-Advisor: Prof. Dr. Rodrigo Coelho Barros

Porto Alegre
2025

Ficha Catalográfica

M989r Mussi Delucis, Marcelo

Reconhecimento de sinais isolados de LIBRAS / Marcelo Mussi Delucis. – 2025.
79 p.
Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Lucas Silveira Kupssinskü.
Coorientador: Prof. Dr. Rodrigo Coelho Barros.

1. LIBRAS. 2. Reconhecimento isolado de línguas de sinais. 3. Vision transformers. I. Kupssinskü, Lucas Silveira. II. Barros, Rodrigo Coelho. III. , . IV. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).
Bibliotecária responsável: Clarissa Jesinska Selbach CRB-10/2051

MARCELO MUSSI DELUCIS

ISOLATED SIGN LANGUAGE RECOGNITION IN LIBRAS

This Master Thesis has been submitted in partial fulfillment of the requirements for the degree of Master in Computer Science of the Computer Science Graduate Program, School of Technology of the Pontifical Catholic University of Rio Grande do Sul

Sanctioned on 6th March, 2025.

COMMITTEE MEMBERS:

Prof. Dr. Dennis Giovani Balreira (INF/UFRGS)

Prof. Dr. Duncan Dubugras Alcoba Ruiz (PPGCC/PUCRS)

Prof. Dr. Rodrigo Coelho Barros (PPGCC/PUCRS- Co-Advisor)

Prof. Dr. Lucas Silveira Kupssinskü (PPGCC/PUCRS - Advisor)

I could not have reached this milestone, along with many others, without the never-ending love, encouragement, and steadfast presence of my beloved, Alessandra Schunski. I dedicate this work, first and foremost, to my pet, Nina, who may have left our lives but will never leave our memories and our heart. Also to my family and friends, whose unwavering support has guided me throughout this journey. I am also deeply grateful for the patience, guidance, and learning opportunities provided by my colleagues and advisor, Prof. Dr. Lucas S. Kupssinskü.

“There is no royal road to science, and only those who do not dread the fatiguing climb of its steep paths have a chance of gaining its luminous summits.”
(Karl Marx)

ACKNOWLEDGMENTS

First and foremost, I extend my sincere gratitude to my colleagues in the Machine Learning Theory and Applications Laboratory (MALTA-LAB), whose discussions significantly enhanced my research and academic knowledge. I also wish to thank every member who contributed to annotating the data for this work, an endeavor that required considerable commitment and patience. I deeply appreciate the funding provided by FAPERGS, CAPES, and PUCRS-PPGCC throughout this journey. Lastly, I am especially thankful to my advisor, Prof. Dr. Lucas S. Kupssinskü, for his guidance and for the invaluable knowledge he shared.

RECONHECIMENTO DE SINAIS ISOLADOS DE LIBRAS

RESUMO

O presente trabalho tem foco no reconhecimento isolado da língua de sinais na Língua Brasileira de Sinais, fundamental para promover acessibilidade digital à comunidade surda. Porém, a escassez de dados e a diversidade limitada de sinais disponíveis e atores dificultam o desenvolvimento de modelos capazes de generalização e avanço na área. Trabalhos anteriores, como o *dataset* MINDS, limitam-se a vocabulários reduzidos, ambientes controlados e baixa diversidade de sinalizadores, o que tende a resultar, em alguns casos, em modelos super-especializados e com baixa acurácia em cenários diferentes do que é visto no conjunto de treinamento. Com o intuito de abordar as atuais limitações, foi desenvolvido um conjunto de dados, MALTA-LIBRAS, construído pela coleção de vídeos de LIBRAS disponíveis publicamente, introduzindo variabilidade em sinalizadores, ambientes e condições de gravação. Três arquiteturas baseadas em *Transformers*, VideoMAE, TimeSformer e ViViT, são investigadas em três configurações experimentais: pré-treinamento em conjuntos de dados de reconhecimento de ações, aplicação de estratégias de aumento de dados e exploração de possível transferência de conhecimento entre línguas de sinais a partir de conjuntos de dados das línguas de sinais norte americana e russa. Resultados no dataset MALTA-LIBRAS indicam que os modelos pré-treinados em tarefas de reconhecimento de ações atingem 29% de acurácia, enquanto modelos sem pré-treino atingem o equivalente a predição aleatória. Técnicas de aumento de dados auxiliam na generalização do modelo, aumentando a acurácia de 29% para 33,6%. A transferência de conhecimento entre línguas para LIBRAS mostrou-se limitada, com ganhos de 2,7% em acurácia, reforçando a necessidade de adaptação específica por domínio. Conclui-se que a diversidade de dados (sinalizadores, ambientes) é tão crucial quanto o volume para aplicações reais, e é proposto um *framework* unificado para SLR em cenários de baixos recursos, combinando pré-treinamento em ações humanas, aumento de dados direcionado e *fine-tuning*.

Palavras-Chave: LIBRAS, reconhecimento isolado de línguas de sinais, Vision Transformers.

ISOLATED SIGN LANGUAGE RECOGNITION IN LIBRAS

ABSTRACT

The present work focuses on the isolated recognition of sign language in Brazilian Sign Language (LIBRAS), essential for promoting digital accessibility for the Deaf community. However, data scarcity and the limited diversity of available signs and actors hinder the development of models capable of generalization and advancement in the field. Previous works, such as the MINDS dataset, are limited to reduced vocabularies, controlled environments, and low signer diversity, which tends to result, in some cases, in super-specialized models with low accuracy in scenarios different from what is seen in the training set. To address current limitations, a dataset, MALTA-LIBRAS, was developed, constructed by collecting publicly available LIBRAS videos, introducing variability in signers, environments, and recording conditions. Three architectures based on Transformers, VideoMAE, TimeSformer, and ViViT, are investigated in three experimental configurations: pre-training on action recognition datasets, application of data augmentation strategies, and exploration of possible knowledge transfer between sign languages using datasets from North American and Russian sign languages. Results on the MALTA-LIBRAS dataset indicate that models pre-trained on action recognition tasks achieve 29% accuracy, while models without pre-training achieve the equivalent of random prediction. Data augmentation techniques aid model generalization, increasing accuracy from 29% to 33.6%. Knowledge transfer between languages to LIBRAS proved limited, with gains of 2.7% in accuracy, reinforcing the need for domain-specific adaptation. It is concluded that data diversity (signers, environments) is as crucial as volume for real-world applications, and a unified framework for SLR in low-resource scenarios is proposed, combining pre-training on human actions, targeted data augmentation, and fine-tuning.

Keywords: LIBRAS, isolated sign language recognition, Vision Transformers.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AI – Artificial Intelligence

ANN – Artificial Neural Network

ASL – American Sign Language

CNN – Convolutional Neural Network

CSL – Chinese Sign Language

CSLR – Continuous Sign Language Recognition

CV – Computer Vision

DL – Deep Learning

GPU – Graphics Processing Unit

GSL – German Sign Language

IBGE – Brazilian Institute o Geography and Statistics

ISLR – Isolated Sign Language Recognition

LIBRAS – Brazillian Sign Language

MAE – Masked Autoencoder

ML – Machine Learning

MLP – Multi-Layer Perceptron

MSE – Mean-Squared Error

NHS – National Health Survey

NLP – Natural Language Processing

NMT – Neural Machine Translation

NN – Neural Network

RELU – Rectified Linear Unit

RNN – Recurrent Neural Network

RSL – Russian Sign Language

SL – Sign Language

SLR – Sign Language Recognition

SOTA – State Of The Art

WHO – World Health Organization

CONTENTS

1. INTRODUCTION

In recent years Artificial Intelligence (AI) has gained increasing importance in a wide range of applications, becoming an important component in many aspects of modern life. Its integration into computational tools and services has streamlined processes and supported decision-making in a wide range of subjects, from healthcare and transportation to entertainment and communication [36]. This growth has been closely tied to progress in Machine Learning (ML), a branch of AI that is able to identify patterns in data and make predictions or decisions without explicit instructions [11]. In particular, Deep Learning (DL), a subfield of ML that leverages multi-layer Neural Networks (NNs) to learn hierarchical representations directly from raw data, have set in motion a new wave of breakthroughs in areas such as Computer Vision (CV) and Natural Language Processing (NLP). The recent development of large-scale datasets, coupled with improvements in computing resources, such as Graphics Processing Units (GPUs), has enabled DL models to handle complex tasks with a level of sophistication and scalability previously unattainable [52].

Within the domain of computer vision, DL models have transitioned focus on tasks from static image analysis to the more challenging realm of video understanding. This transition recognizes that dynamic scenes, captured as sequences of frames, contains valuable temporal information. By incorporating both spatial and temporal cues, models designed for video classification can recognize actions, detect events, and interpret interactions that unfold over time [50, 85]. These methods have become increasingly relevant over the years in diverse applications, such as security, sports analytics and human-machine interfaces [34, 84]. As research in CV has evolved, the field has grown more comprehensive, integrating specialized applications such as Sign Language Recognition (SLR). Rather than a narrow shift, frameworks now embrace SLR as one facet of a diversified research agenda, leveraging advances in spatio-temporal feature extraction, cross-modal alignment and real-time processing to address challenges in CV [102, 58].

According to the World Health Organization (WHO) [101], more than 1.5 billion people worldwide experience some degree of hearing loss, and around 466 million of them face serious hearing impairments. Although members of the Deaf community have their own linguistic and cultural traditions, these identities are frequently overlooked. Many hearing societies perceive deafness as a disability, rather than recognizing it as part of a distinct culture, which can lead to general marginalization [81]. SLs, as unique and natural languages expressed through manual and non-manual visual signals, present distinct challenges to computational modeling [46]. Unlike spoken languages, which heavily rely on linear sequences of phonemes and words, SLs convey meaning through complex spatial-temporal patterns of hand shapes, orientations, movements, facial expressions, and body postures.

This form of communication is not merely a transcription for spoken languages but, instead, a linguistic system with unique grammars and lexicons [13]. The inherent complexity of SLs, along with variations in signing style, dialects, signer appearance, and recording conditions, makes the creation of robust and adaptable recognition models non-trivial [46]. While the advent of wearable devices and assistive technologies, such as SLR gloves and gesture-based interfaces, initially promised significant improvements in accessibility, their adoption within Deaf communities has been limited due to practical and cultural reasons. Deaf individuals often express skepticism towards these devices because they fail to fully capture the nuances and linguistic richness of SL, particularly the crucial role of facial expressions and body language integral to meaning. Additionally, the requirement for specialized equipment or unnatural signing conditions further diminishes their practicality and acceptance. Moreover, research highlights that many Deaf users prefer interacting with technologies that respect their natural linguistic behaviors, rather than adapting their signing style to fit technological constraints. Consequently, effective SLR technologies must incorporate these considerations, focusing on naturalistic sign capture and recognition methods to genuinely enhance accessibility, inclusivity, and usability for Deaf communities [35].

SLR involves interpreting signs at varying levels of granularity, ranging from character-level recognition, commonly referred to as fingerspelling, to word-level, Isolated Sign Language Recognition (ISLR), and finally to Continuous Sign Language Recognition (CSLR), which captures entire phrases or sentences [8]. Fingerspelling typically involves recognizing handshapes that represent individual letters of an alphabet and is often considered a more straightforward task, as it comprises a limited and fixed set of gestures. However, challenges still arise from variations in speed, style, and signer proficiency [83]. ISLR, in contrast, deals with recognizing discrete signs corresponding to individual words, which introduces additional complexities, including significant variability in handshapes, motion trajectories, facial expressions, and contextual nuances [8]. CSLR further elevates the complexity by requiring recognition of a continuous stream of signs, where segmentation and temporal alignment become crucial issues, compounded by the lack of explicit boundaries between signs. Additionally, CSLR faces substantial challenges due to the complex grammar, fluid transitions, and extensive vocabulary present in natural SL communication [21]. The focus on ISLR in this dissertation was primarily driven by the already limited availability of datasets at the word level in LIBRAS. Pursuing CSLR would have significantly exacerbated the data scarcity issue, hindering the feasibility and robustness of developing effective recognition models. Thus, focusing on ISLR allowed for a more achievable scope while still addressing critical gaps in existing research and dataset availability.

Several Sign Languages (SL), such as American Sign Language (ASL), Russian Sign Language (RSL), German Sign Language (GSL), and Chinese Sign Language (CSL), are supported by extensive datasets that have enabled the training of DL models addressing the SLR task. For instance, the WLASL dataset [53] for ASL, the SLOVO dataset [49] for RSL, the RWTH-PHOENIX-Weather dataset [30] for GSL, and the CSL dataset [44] for CSL collectively serve as benchmarks for evaluating spatio-temporal DL models. In contrast, Brazilian Sign Language (LIBRAS) remains limited by a shortage of large, diverse datasets, resulting in a constrained training scenario for DL approaches. Models trained on currently available LIBRAS datasets often show poor generalization, primarily due to insufficient and less representative data.

Among the initiatives for LIBRAS, the MINDS [78] dataset stands out as an effort in the area of ISLR. However, the constrained vocabulary and lack of a dedicated test set limit the extent to which models trained solely on MINDS can generalize to unseen signers, broader contexts and novel signs. Models trained on MINDS data risk overfitting, as they do not encounter sufficient diversity in vocabulary, signing style or recording conditions to handle real-world scenarios effectively.

In view of these issues, this dissertation introduces a novel dataset, named MALTA-LIBRAS, which was compiled through web scraping from multiple open sources. Our dataset encompasses over 11,000 videos, with 60 unique actors performing signs and over 4,500 unique signs. MALTA-LIBRAS intersects with the MINDS vocabulary but extends coverage to more signers and less controlled video samples, aiming to capture some of the variability found in everyday signing. The dataset's complementary nature enables it to function as a test set for models originally trained and validated on MINDS, thereby shedding light on generalization capabilities of DL models.

This dissertation examines three State-Of-The-Art (SOTA) attention-based video classification architectures, VideoMAE [95], TimeSformer [10], and ViViT [6], in the context of ISLR for LIBRAS. These architectures exploit attention mechanisms to analyze both spatial and temporal dynamics, suitable to handle the gestural complexity of SLs. Our experiments consider multiple facets of model development, including the effect of pre-training on large-scale action recognition tasks, cross-datasets experiments using other SLs datasets, and the impact of different data augmentation techniques on model performance.

Ultimately, understanding and improving automated ISLR in LIBRAS bear practical significance. Effective SLR tools could aid in education, make public services more inclusive [90], support content creation in SL [93], and strengthen the connection between deaf and hearing communities [13]. By bridging the gap between the controlled conditions of existing datasets and the variability inherent in genuine everyday use, the research presented here aspires to contribute to the design and evaluation of models that are ready for deployment in diverse real-life signing contexts.

Our findings indicate that training these models exclusively on a small, controlled dataset like MINDS leads to an overfit scenario, whereas introducing additional data or pre-training techniques improves convergence speed and generalization. The MALTA-LIBRAS dataset, by presenting a broader range of signing conditions, highlights the limitations of models that have only been exposed to homogeneous training data. In sum, this dissertation aims to advance ISLR in LIBRAS beyond small or highly controlled datasets by demonstrating how data diversity, augmented training pipelines, and transfer learning strategies can help build DL models that more accurately reflect the complexities of real-world signing.

2. BACKGROUND

This chapter outlines the theoretical foundations for the research presented in this work. We begin by examining the nature and diversity of SLs, with particular emphasis on the distinctive features of LIBRAS. The discussion then turns to the key paradigms of machine learning, including supervised, unsupervised, semi-supervised, and reinforcement learning, followed by an overview of deep learning concepts—from early perceptrons to modern architectures designed to handle complex tasks at scale. The final sections address the domain of video understanding, focusing on attention-based models such as Transformers and Vision Transformers that capture spatiotemporal relationships and have proven effective in tasks like SLR.

2.1 Sign Languages

SLs are rich and complex forms of communication that naturally emerged to meet the communicative needs of Deaf individuals. Each SL has its own grammatical structure and syntactic rules that are distinct and are not mere gestural transcriptions of their corresponding spoken languages. For instance, ASL has its own grammatical system, which is significantly different from North American English [13].

Estimates indicate the existence of more than 200 SLs used by over 70 million Deaf people worldwide [100]. Within each SL, there are variations analogous to the dialects found in spoken and written languages. SLs incorporate manual elements, such as hand position, movement, shape, and orientation. They also rely on non-manual characteristics, including facial expressions, body posture, and the use of gaze to convey more subtle meanings [2, 78].

SLs have distinctive characteristics that differentiate them considerably from spoken languages, not only in grammatical aspects but also in terms of general structure, word sequence, and lexicon [89]. For example, when translating from a spoken language to its corresponding SL, translators face the challenge of mapping language concepts to equivalent signs, which may be represented by one or more specific gestures. This task requires a deep understanding of semantic nuances and the universal linguistic rules governing sign formation [47].

Moreover, interlinguistic translation between SLs and spoken languages is not a direct transcription but a transposition requiring adjustments to accommodate structural differences. While spoken languages often rely on syntax and word order for meaning, SLs frequently utilize physical space, movement and even near objects and individuals to express concepts, leading to unique sentence constructions and a spatial representation that is intrinsic to how signs are produced and understood [81].

2.1.1 LIBRAS

In 2002, LIBRAS was recognized as an official language of Brazil through Law No. 10,436 [14], establishing it as a legal means of communication and expression [32]. As part of the ongoing commitment to accessibility, Brazil's Ministry of Health developed a manual to improve access for individuals with disabilities to the Unified Health System (SUS), thus ensuring the availability of adequate health information and services [64, 19].

According to data from the 2019 National Health Survey (NHS), published by the Brazilian Institute of Geography and Statistics (IBGE) in 2021, approximately 1.1% of the Brazilian population, equivalent to about 2.3 million people, have some degree of hearing impairment. Among this group, 22.4% use LIBRAS as a form of communication [4].

As a language that emerged naturally, LIBRAS employs a grammatical system composed of unique syntax, morphology, and phonology. Its core word order typically follows Subject-Verb-Object (SVO), though alternative structures appear through topicalization and the use of non-manual markers such as facial expressions, gaze, and head movements. For example, the portuguese sentence "João deu o livro para Maria." (or "John gave the book to Maria" in english) is rendered in LIBRAS as "MARIA *EMPHASIZED GESTURE* JOÃO LIVRO DAR" (or "MARIA *EMPHASIZED GESTURE* JOHN BOOK GIVE" in english), moving "MARIA" to the front and using an emphasized gesture, such as an eyebrow movement or head tilt, to signal the shift in phrase structure. In LIBRAS, the verb "GIVE" is signed with a directional movement that visually connects the subject to the object, marking their relationship and emphasizing the object phrase by moving it to the beginning of the phrase. These features transmit grammatical relationships and emphasis, contributing to clear communication. Furthermore, the sign for boredom, gestured with different intensities and facial expressions, can semantically signify varying degrees of boredom.

In ISLR, capturing these syntactic and morphological elements is crucial for interpreting signs within their linguistic context. This includes attending to factors such as the distinction between inflecting and plain verbs, agreement processes, and the coordination of manual and non-manual signals [25, 47].

2.2 Machine Learning

ML can be broadly described as the study and development of models that learn patterns from data rather than relying on explicit programming rules. Unlike traditional software engineering, which encodes every logic path in a hard-coded manner, ML enables systems to adapt and generalize from a set of examples, then apply that knowledge when confronted with new, unseen inputs. This flexibility supports many applications, including computer vision, NLP, and recommender systems [65, 11].

A standard ML workflow begins with data collection and preparation, ensuring that the dataset encompasses a broad range of user groups and environmental conditions. This diversity reduces the likelihood that certain subsets of data are overrepresented or underrepresented, which could bias the model or limit its applicability. It is common practice to divide the dataset into training and validation sets for model training and hyperparameter tuning, while a separate test set is used to evaluate how well the model generalizes to examples it has not encountered before. During the training stage, the model iteratively adjusts its weights to optimize metrics such as accuracy or validation loss. Once training concludes, the model moves to an inference phase in which it employs learned patterns to handle novel input data [65, 36].

Broadly speaking, ML methods fall into four paradigms, each defined by the type of data involved and the nature of the learning objective: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Supervised algorithms rely on labeled examples for direct guidance, whereas unsupervised methods discover underlying patterns in unlabeled data. Semi-supervised techniques integrate both labeled and unlabeled samples to mitigate the costs of labeling, and reinforcement learning teaches an agent to select actions based on a reward. Recognizing these differences is essential when selecting solutions to address specific problems or data constraints [11, 69, 36].

Although ML models are able to learn from training data, they often presume that conditions at testing mirror those during training. If the real-world data shifts, model performance may decline unless additional adaptive measures are introduced. Moreover, advanced models and large networks can require substantial computational resources for training and real-time operation, sometimes needing specialized hardware. Addressing these factors, alongside considerations such as bias, data heterogeneity, and distribution changes, helps ensure that ML solutions remain relevant in different settings [65, 11].

2.2.1 Supervised Learning

Supervised learning uses labeled data as guidance. Each training sample has an associated label or target, and the objective is to learn a function that effectively maps inputs to the desired outputs. Common supervised tasks include classification (i.e., assigning discrete labels, such as the "cat" vs "dog" classification task) and regression (i.e., predicting continuous values, such as housing market prices) [11, 69]. In a typical supervised workflow, the dataset is split into training, validation, and test sets; the model iteratively learns from the training set, tunes hyperparameters using the validation set, and demonstrates its performance on the test set. In SLR, supervised methods require annotated videos of signs, each aligned with a word-level or sentence-level label, class or gloss. While these methods often yield high accuracy when many labeled examples are available, collecting comprehensive and diverse annotations is time-intensive and can limit dataset size, especially for SLs like LIBRAS with fewer available resources [65]. In video action recognition task, supervised models such as Inflated 3D ConvNets (I3D) are trained on labeled datasets (e.g., Kinetics-400 [51]), achieving SOTA accuracy on UCF101 [88] dataset by mapping RGB frames to action labels.

2.2.2 Unsupervised Learning

Unsupervised learning trains on unlabeled data by finding latent patterns or structures. Models such as clustering (e.g., K-means [70]) or dimensionality reduction (e.g., PCA [63]) seek to group similar inputs or capture essential features of the data without explicit labels [11]. This paradigm can be beneficial when labeled data is scarce. For instance, an unsupervised approach in SLR might cluster different hand configurations or gesture trajectories. Although unsupervised approaches can reveal insights into data organization and substructures, they often require domain expertise to interpret outputs or to select meaningful hyperparameters [11]. Unsupervised methods may also serve as a foundation for other paradigms by producing initial groupings or embeddings that guide subsequent supervised training. Techniques that learn spatiotemporal video representations without explicit labels, such autoencoder-based methods or deep clustering approaches applied to raw video data are examples of this paradigm which discovers latent video patterns and features solely from the video content.

2.2.3 Self-Supervised Learning

Self-supervised learning has emerged as a distinct paradigm where training patterns come solely from inherent data structures, rather than manual labels. The model predicts masked or transformed portions of data, leveraging patterns like temporal continuity or spatial coherence to learn representations [37]. For textual data, masked language modeling (e.g., BERT [26]) operates by obscuring some tokens and asking the model to reconstruct them. In image or video domains, related approaches hide patches or frames and task the model with reconstruction of missing data.

In SLR, such methods can learn spatiotemporal representations without exhaustive annotation. For example, VideoMAE [95] adopts a masked autoencoder strategy for video, masking a large fraction of input patches and reconstructing them. This method is referred to as self-supervised because it does not rely on external labels to define the objective; the data itself provides the learning signal. Once trained in this manner, the network can be fine-tuned on smaller labeled sets for tasks like SLR.

2.2.4 Semi-supervised Learning

Semi-supervised learning lies between supervised and unsupervised approaches, blending a limited quantity of labeled data with a larger pool of unlabeled data [105, 18]. The rationale is that labeled examples provide direct guidance, while unlabeled ones help the model uncover broader data structures or manifold properties. In practice, semi-supervised algorithms often attempt to ensure that data points close in the feature space share the same label or combine label consistency with unsupervised objectives. For SLR, semi-supervised strategies could leverage partly annotated SL datasets: a small set of gloss-labeled samples plus many unlabeled videos. These methods can diminish labeling costs by propagating labels or constraints from a few annotated samples to many unlabeled examples, potentially yielding higher coverage of different signers and scenarios.

2.2.5 Reinforcement Learning

Reinforcement learning is a learning paradigm wherein an agent interacts with an environment to maximize a cumulative reward signal, rather than relying on labeled examples. At each time step, the agent observes the current state, selects an action, and receives feedback in the form of a reward and a new state. Over time, it refines its policy, which is essentially a mapping from states to actions, to optimize long-term reward. This trial-and-error process allows the agent to adapt to uncertain or changing environments, with close to none prior knowledge of explicit objectives. While early methods often employed tabular representations for small-scale problems, more recent work combines principles with deep Neural Networks (NN) to handle high-dimensional inputs and complex tasks, such as robotics control and game-playing. In all cases, the core idea is that an agent can learn effective behaviors by iteratively observing and acting on feedback rather than depending on pre-defined training labels or direct instructions [66, 91].

2.2.6 Artificial Neural Networks

Artificial Neural Networks (ANNs) are models inspired by simplified views of how biological neurons process information in the human brain. ANNs are built upon interconnected layers of artificial neurons, or nodes, each performing weighted sums of inputs and passing the result through an activation function. This structure allows ANNs to approximate a wide variety of functions, making them useful for classification, regression, and numerous other learning tasks [65, 80, 11, 36].

The Perceptron represents one of the earliest forms of ANNs. It classifies an input vector **x** by computing a weighted sum of its features and passing the result through a step function. Mathematically, the Perceptron's output y_{pred} can be expressed as:

$$z = \mathbf{w}^\top \mathbf{x} + b, \quad y_{\text{pred}} = \begin{cases} +1 & \text{if } z \geq 0, \\ -1 & \text{otherwise}, \end{cases} \tag{2.1}$$

where **w** is the weight vector, and b is the bias term. During training, if a data point is misclassified, the Perceptron adjusts **w** and b by an amount proportional to the input and the prediction error [65]. Here, the step function (often named signal) saturates to 0 for negative values and 1 otherwise. A recognized drawback is the Perceptron's reliance on a linear decision boundary, making it unsuitable for data that require more complex, nonlinear separation. Figure 2.1 illustrates an example of this limitation, where a purely linear boundary prevents the solution of tasks like the XOR problem.

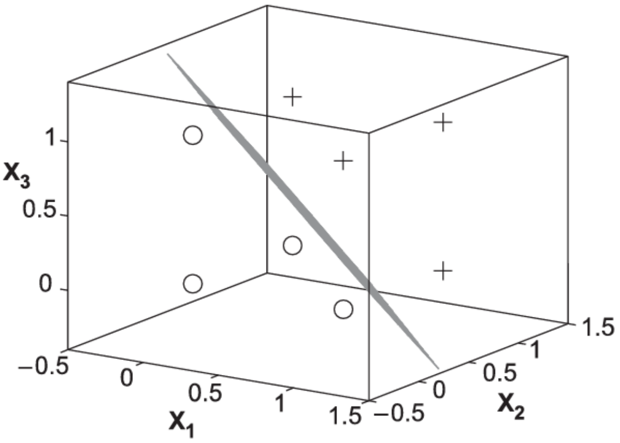

Figure 2.1 – Perceptron's linear decision boundary [92].

The Multi-Layer Perceptron (MLP) was introduced to overcome the Perceptron's limitation [65, 11]. An MLP consists of multiple Perceptrons interconnected and arranged in the fashion of an input layer, one or more hidden layers, and an output layer. Each layer typically performs an affine transformation of the preceding layer's outputs, followed by a nonlinear activation function instead of the typical step function. One commonly used activation in modern architectures is the Rectified Linear Unit (ReLU) [**?**], defined as

$$ReLU(x) = \max(0, x). \tag{2.2}$$

Thus emerged a novel feed-forward network by stacking multiple hidden layers and non-linear activations [36]. Training a Neural Network (NN) involves three key elements: a loss function, the forward pass, and the backward pass. The loss function quantifies how far a model's output deviates from the desired target. For regression tasks, the Mean-Squared Error (MSE) is a frequent choice:

$$J(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{2.3}$$

where y_i and $\hat{y}_i$ represent the ground-truth and predicted values, respectively. Additionally, for classification tasks, Cross-Entropy loss is the most usual:

$$J(\mathbf{w}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c}), \tag{2.4}$$

where N is the number of samples, C is the number of classes, $y_{i,c}$ is a binary indicator (0 or 1) of whether class c is the correct class for sample i, and $\hat{y}_{i,c}$ is the predicted probability for class c for sample i. During the forward pass, the network computes its predictions by applying the learned weights to the input data and then evaluates how well it performed via the loss function. In the backward pass, the network updates its weights using the backpropagation algorithm, which calculates the gradients of the loss with respect to each parameter and adjusts them based on an optimization method such as Gradient Descent. This iterative cycle minimizes the error over time, aligning predictions more closely with actual targets [36].

The introduction of the backpropagation algorithm enabled these networks to adjust internal parameters through gradient-based optimization, thus learning more detailed patterns than a single Perceptron could capture. Even though early MLPs could approximate a variety of functions, they sometimes suffered from issues such as vanishing gradients and limited computational power [65].

As hardware advanced and new techniques arose (improved weight initialization, momentum-based optimization, alternative activation functions), deeper networks with more hidden layers became feasible [36]. This evolution led to architectures like Convolutional Neural Networks (CNNs) for image analysis and Recurrent Neural Networks (RNNs) for sequence modeling. More recently, attention-based architectures have gained prominence for tasks involving language and vision. Nevertheless, the core principle remains that a network's capacity to approximate complex functions hinges on its layered com-position of linear transformations and nonlinear activations. Larger networks, equipped with more parameters, can learn complex representations, though they also require larger datasets and often greater computational resources [36].

2.3 Deep Learning

DL is a subfield of ML that employs NNs to learn complex representations from data. Instead of relying on feature engineering procedures designed by human experts, DL architectures are able to determine these features through hierarchical representations. Early layers may capture low-level features (such as edges or simple textures in images), while deeper layers combine those basic elements into increasingly abstract concepts. This end-to-end learning approach has shown effectiveness in tasks ranging from image recognition to natural language processing, demonstrating greater performance compared to shallow models in many benchmark evaluations [36, 9].

Several factors contributed to the rise of DL, such as the rapid growth of accessible training data, often referred to as "Big Data". Large, annotated datasets, such as ImageNet [52], enabled deep networks to learn intricate patterns that had previously been beyond reach by traditional ML approaches. Advances in parallel computing further accelerated progress, especially with Graphics Processing Units (GPUs), which handle the large-scale matrices computations required by DL models more efficiently than traditional CPUs [52, 80].

In contrast to early ANNs that employed only a handful of layers, DL architectures may include dozens or even hundreds of layers, increasing their representational ability. However, deeper networks also become more susceptible to problems like vanishing or exploding gradients if not designed or initialized carefully. Techniques such as weight initialization, batch normalization, and different activation functions (such as ReLU [**?**], PReLU [38], GeLU [40], etc) helped mitigate these difficulties, enabling the stable training of deeper and larger models [36].

Recent developments moved beyond purely feed-forward networks by integrating specialized approaches for different data types. Convolutional Neural Networks (CNNs) are commonly used as a mechanism for feature extraction in images, while Recurrent Neural Networks (RNNs) addresses temporal information in domains such as speech or text. Attention-based architectures are able to capture longer-range dependencies without explicit recurrences. Despite structural differences and distinct application areas, DL models embraces the idea of stacking multiple layers to process raw inputs and uncover relevant patterns for the given task [96].

2.3.1 Video Understanding

Video classification tasks encompass a range of applications that involve analyzing spatiotemporal patterns in sequences of video frames to categorize various types of content. For instance, models can be employed in tasks such as event detection, where the goal is to identify occurrences like sports highlights or public gatherings [45]; scene classification, which involves categorizing videos based on settings such as urban environments or natural landscapes [104]; gesture recognition, focusing on subtle hand or body movements [67]; and video sentiment analysis, which assesses the emotional tone conveyed by visual cues [1]. In all these cases, models must effectively capture both spatial features, such as body posture, object presence, and facial expressions, and temporal coherence that reflects movement, transitions, and the continuity of objects over time.

Among these, action recognition is a task that focuses on identifying specific human actions. In this process, models learn to extract spatial features, such as body posture and the appearance or continuity of objects, and combine them with temporal information that captures the movement and transitions over time. The challenge lies in discerning subtle cues that distinguish one action from another, even when variations in lighting, perspective, or background occur.

Moreover, action recognition systems must handle sequences of variable length and ensure that the extracted representations are robust to changes throughout the video. The unique challenges posed by these tasks have fostered the development of specialized architectures for video understating, such as 3D CNNs and transformer-based models, which are frequently evaluated on large-scale benchmark datasets like Kinetics-400 [51] and UCF101 [88].

Although SLR has its own set of unique requirements, such as capturing subtle hand configurations and facial expressions, it does share similarities with action recognition in terms of modeling spatiotemporal patterns and dealing with motion across video frames. Both tasks must manage variable video lengths, account for background variations, and encode body movements. Despite these parallels, how advances in action recognition could be directly transferred to SLR remains underexplored. As SLR often requires finer-grained analysis and linguistic contextualization, bridging the gap between these domains poses a valuable research direction, with the potential to enrich SLR approaches through knowledge gained from action recognition tasks.

2.3.2 Attention

Early neural architectures, such as RNNs, often found it challenging to maintain relevant information over extended sequences or to parallelize computation efficiently. Convolutional approaches are effective at extracting local features but rely on increasingly deeper architectures to capture long-range dependencies, which can lead to vanishing gradients in very deep networks. As gradients travel back through many layers, they become progressively smaller, complicating the training of early layers. These challenges motivated the design of a mechanism that is able to focus on relevant parts of the data at any point in a sequence without relying on explicit recurrence or a large number of hidden layers. This mechanism, known as attention, dynamically reassign weights so that each segment of the model is able to selectively prioritize the most informative features [43, 62, 96, 9].

Attention is used in Neural Machine Translation (NMT), where an encoder-decoder framework processes one language at the input and produces another language at the output. In a typical setup, the encoder reads the source sequence token by token, forming context representations, called embeddings, that summarizes each token's semantics. The decoder then generates the target sequence one step at a time, but instead of using all encoder states uniformly, it uses attention to decide which encoded tokens matter most for each output token. This selective approach often outperforms naive encoding and helps address the vanishing context problem by letting the decoder dynamically retrieve relevant information from the entire source sequence [7].

In practice, the data is split into tokens (e.g., words or "subwords"), and each token is associated with an embedding that represents basic features. Due to the embedding alone not conveying the position of a token, models usually also add positional information, which ensures it's discernment of the relative ordering of the sequence elements. Once the notion of tokens and embeddings is in place, the core of attention revolves around Queries, Keys, and Values: the model measures how similar a query vector is with a set of key vectors and produces a weighted combination of corresponding value vectors. The most common way which models measures this similarity is using the scaled dot-product attention:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\,\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \tag{2.5}$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are the Queries, Keys, and Values, d_k is the dimension of queries and keys, and the denominator $\sqrt{d_k}$ helps normalize the dot-product magnitudes [62].

To expand the model's ability to capture different relationship types in parallel, some architectures split the projection space into multiple “heads,” each performing the attention step independently. This idea, referred to as multi-head attention, aggregates the outputs of each head after they have processed Queries, Keys, and Values in their own subspace. By combining these multiple perspectives, attention-based models are able to learn a variety of contextual cues from the data in a single pass.

Because the encoder-decoder structure in NMT demonstrates how attention can dynamically link an input context to the tokens being generated, the concept has been adapted to other tasks where the model must retrieve relevant details from large or unstructured sources. In image analysis, for instance, tokens can represent patches, while in audio or video, tokens may represent segments of a waveform or frames of a sequence. Although these different data modalities require specific transformations, the primary benefit remains: attention directs the model's focus to the most pertinent features at any given time, rather than relying on uniform weighting [96, 62].

2.3.3 Transformer Architecture

From the desire to alleviate the drawbacks of earlier sequence-processing models, the Transformer architecture emerged. It's design was centered on the attention mechanism, much like previous encoder-decoder architectures, with the exception that it explicit relies on multi-head self-attention in both the encoder and decoder. At each layer, multi-head self-attention directs how tokens in the sequence attend to one another, without enforced sequential operations. By splitting the attention into the multiple heads, the model is able to capture representations in parallel. Formally, each head is expressed as:

$$\text{head}_i = \text{Attention}\Big(\mathbf{Q}\,\mathbf{W}_i^Q,\ \mathbf{K}\,\mathbf{W}_i^K,\ \mathbf{V}\,\mathbf{W}_i^V\Big), \tag{2.6}$$

where $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V$ are trainable projection matrices for head i. The final output of the multi-head attention is obtained by concatenating the outputs of all heads, then passing them through a linear transformation $\mathbf{W}^O$:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \big[\text{head}_1; \text{head}_2; \ldots; \text{head}_h\big]\,\mathbf{W}^O. \tag{2.7}$$

Residual connections and layer normalization help maintain gradient flow as the data propagate through deeper layers. This enables the Transformer to be able to learn contextual dependencies across an entire sequence in a single forward pass rather than through accumulation, reducing the time complexity per token and improving parallelization. Furthermore, in contrast to previous architectures, each token additionally receives a positional embedding that tracks sequence order. Consequently, it captures and propagates information across the entire architecture directly, while retaining the encoder-decoder paradigm [96].

While the encoder reads the entire input at once and produces a rich contextual embedding for each token, the decoder, as exemplified by Figure 2.2, relies on two types of attention: self-attention over the tokens already generated and cross-attention that aligns the decoder's current state with relevant parts of the encoder output. This encoder-decoder division makes the Transformer well suited for tasks such as NMT, which require continuous reference to earlier tokens or contexts in another language [62, 7].

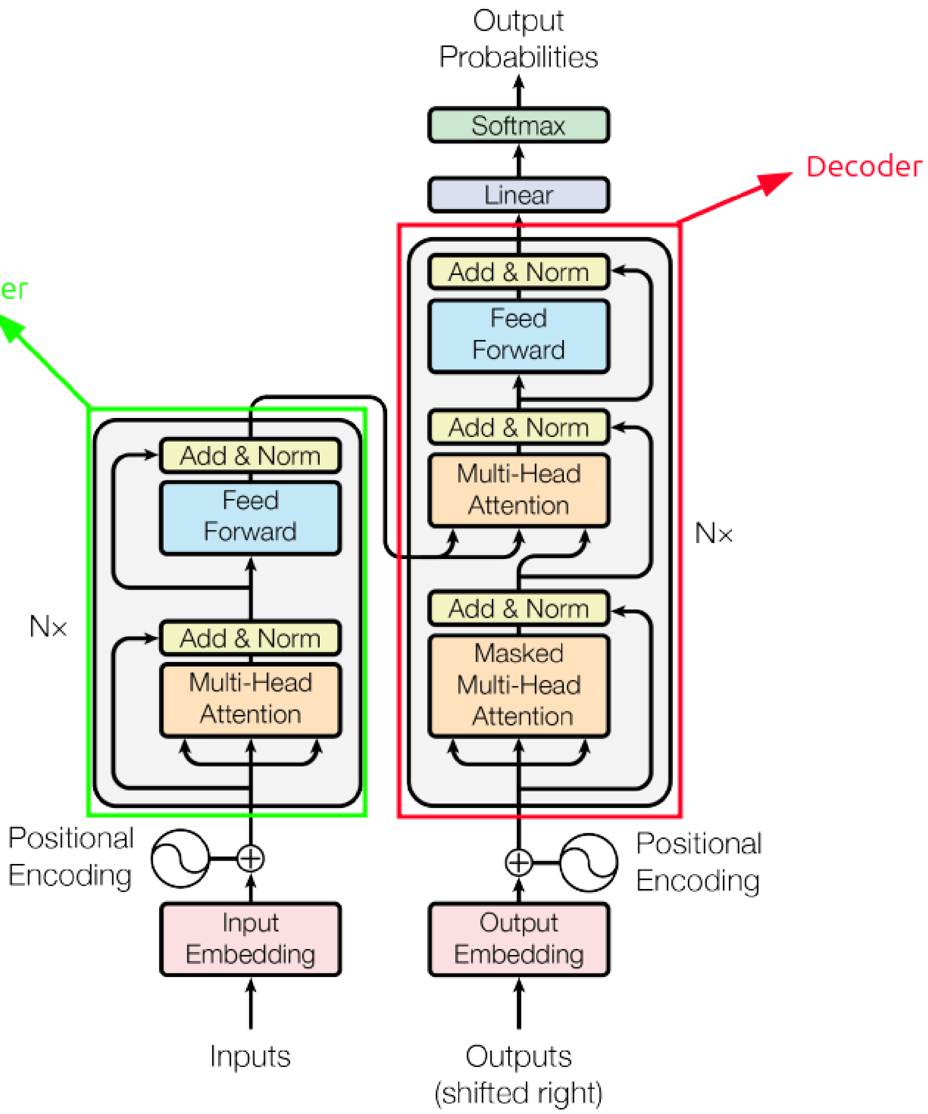

Figure 2.2 – The Transformer architecture [96].

Due to its ability to handle all positions in a sequence in parallel, the Transformer cuts down the training bottlenecks found in RNNs, allowing longer-range dependencies to be captured more effectively. In addition, the attention weights yield partial interpretability, as they indicate which tokens or encoded features the network emphasizes at each step. Although the Transformer was initially applied to text-focused tasks, recent adaptations involve images, audio, and even multimodal applications by dividing input data into token-like patches or segments. Thus emerging Vision Transformer models, where video frames are segmented into spatial or spatio-temporal patches and processed in a manner reminiscent of NLP tokens. This approach leverages the Transformer's original principles, parallel attention, positional embeddings, and layered processing, but repurposes them to model visual sequences rather than purely textual inputs [26, 15].

2.3.4 Vision Transformers

The success of Transformer-based architectures in NLP led researchers to explore whether similar attention mechanisms could benefit vision tasks. Instead of relying on convolutions or purely sequential operations, Vision Transformers (ViTs) restructure image or video data into token-like units, allowing multi-head self-attention to capture patterns across spatial or spatio-temporal dimensions [27]. By dividing inputs into different patches, these models leverage the scalability and parallelization advantages provided by the transformer architecture, adapting concepts such as positional embeddings to account for two or three-dimensional layouts.

ViTs divide an image of size $H \times W$ with three color channels into patches of size $P \times P$, producing $N = \frac{HW}{P^2}$ patches. Each patch is then flattened and linearly mapped into a d-dimensional embedding:

$$x_i^{\text{patch}} = \text{Flatten}\big(\text{Patch}_i\big)\,\mathbf{W}_e + \mathbf{b}_e, \quad i = 1, \dots, N, \tag{2.8}$$

where $\mathbf{W}_e \in \mathbb{R}^{(P^2 \cdot 3) \times d}$ and $\mathbf{b}_e \in \mathbb{R}^d$ are learnable parameters. A learnable class token, $\mathbf{x}_{\text{class}}$, is added to these embeddings, and positional embeddings $\mathbf{E}_{\text{pos}}$ are added to retain spatial information:

$$\mathbf{X}^0 = \big[\mathbf{x}_{\text{class}}; x_1^{\text{patch}}; \dots; x_N^{\text{patch}}\big] + \mathbf{E}_{\text{pos}}. \tag{2.9}$$

This sequence is processed by the Transformer encoder, where multi-head self-attention and feed-forward layers capture increasingly complex spatial and contextual patterns. The class token aggregates global context across patches, serving as a compact image representation for classification tasks. When extended with a decoder, ViTs enable applications such as image-to-text mapping and generative tasks [27]. Unlike CNNs, ViTs leverage global attention, improving performance on tasks requiring understanding, such as scene analysis and object interactions [48, 12]. A depiction of the ViT architecture is shown in Figure 2.3, from Dosovitskiy *et al.* [27].

By replacing localized processing with global self-attention, ViTs effectively capture both fine-grained details and high-level semantics, achieving SOTA results across various benchmarks, particularly when pre-trained on large-scale datasets such as Kinetics [22, 51]. Among various approaches, TimeSformer [10], ViViT [6], and VideoMAE [95] have emerged as efficient tools for video classification tasks.

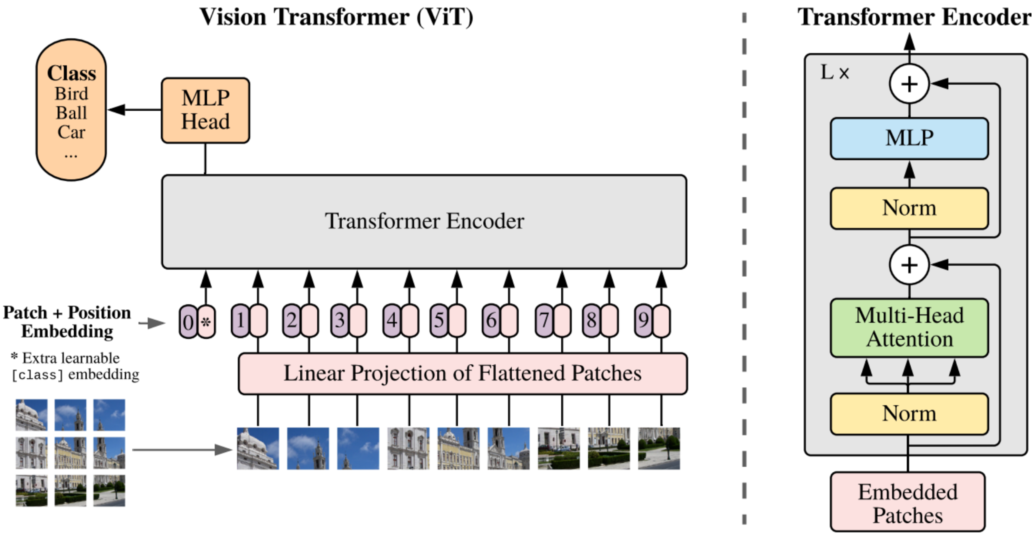

Figure 2.3 – Overview of the Vision Transformer architecture designed by [27].

TimeSformer

TimeSformer addresses the spatiotemporal nature of video data by extending the self-attention mechanism to operate across both spatial and temporal dimensions. It adopts a purely transformer-based architecture, leveraging its capacity for parallelization and global attention. This design enables the model to capture dependencies within individual frames as well as across sequences of frames.

The model processes video data by dividing each frame into non-overlapping patches, similar to the ViT architecture. These patches are flattened and embedded into a high-dimensional space, effectively transforming each patch into a token. To retain spatial and temporal information, positional embeddings are added to the tokenized input. The resulting sequence of tokens is then fed into an encoder, where self-attention mechanisms model relationships within and across frames.

TimeSformer employs a few variants of spatiotemporal attention, as can be seen by Figures 2.4 and 2.5. In the divided space-time attention mechanism, spatial attention is applied independently within each frame to capture intra-frame relationships, followed by temporal attention across frames to model dependencies over time. In contrast, joint space-time attention applies attention across spatial and temporal dimensions simultaneously, providing a comprehensive representation at the expense of higher computational demands. Other variants, such as space-only or time-only attention, restrict the model to focus on either spatial or temporal dimensions, simplifying computation but sacrificing performance. Sparse local-global attention combines local attention over subsets of tokens with global attention to retain overall context, further optimizing computational efficiency.

The model is trained using supervised learning, with cross-entropy loss serving as the primary objective for classification tasks. Pre-training on general video datasets enhances the model's ability to generalize to specific tasks, such as action recognition. During training, techniques like patch tokenization, positional embeddings, and layer normalization are employed to stabilize learning and improve convergence. Patch tokenization reduces input dimensionality by segmenting frames into smaller regions, while positional embeddings ensure the model captures the sequential ordering of frames and layer normalization mitigates gradient instability during the optimization process [10, 27].

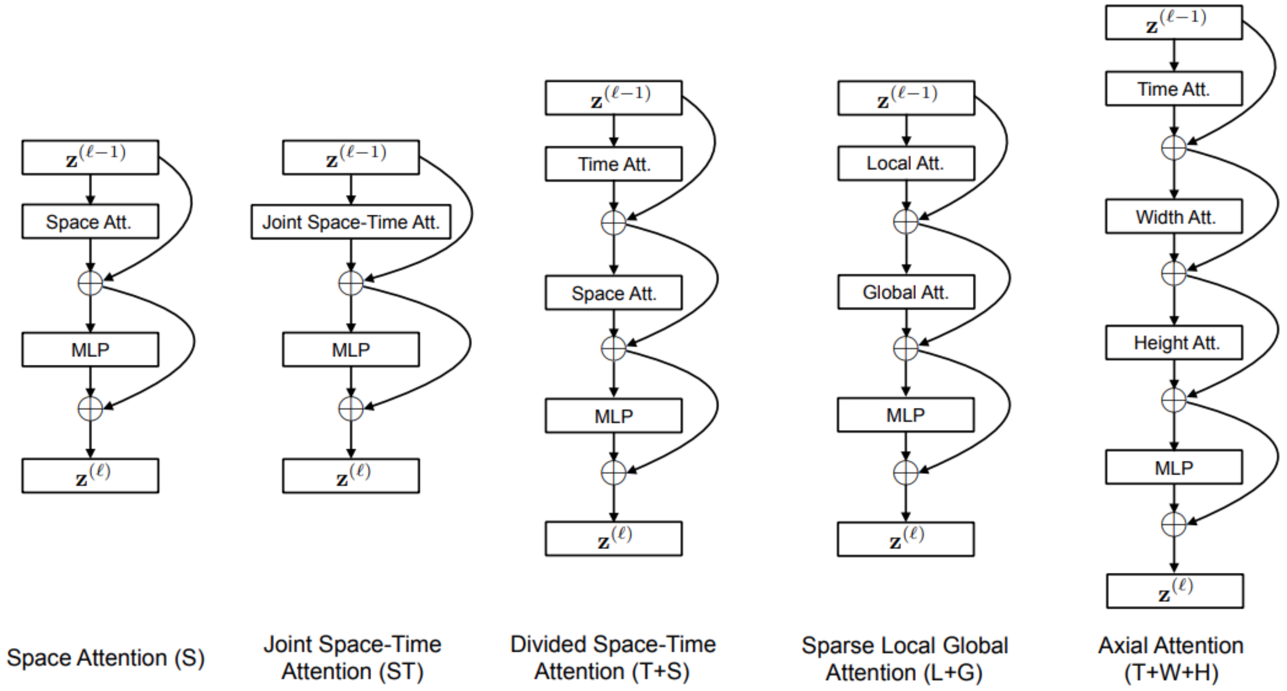

Figure 2.4 – Overview of the attention mechanisms explored in TimeSformer: illustrating five variations of attention strategies, including space-only attention, joint space-time attention, divided space-time attention (separate temporal and spatial), sparse local-global attention (combining localized and global attention), and axial attention (sequentially applied along time, width, and height dimensions), showcasing the flexibility in capturing spatiotemporal relationships [10].

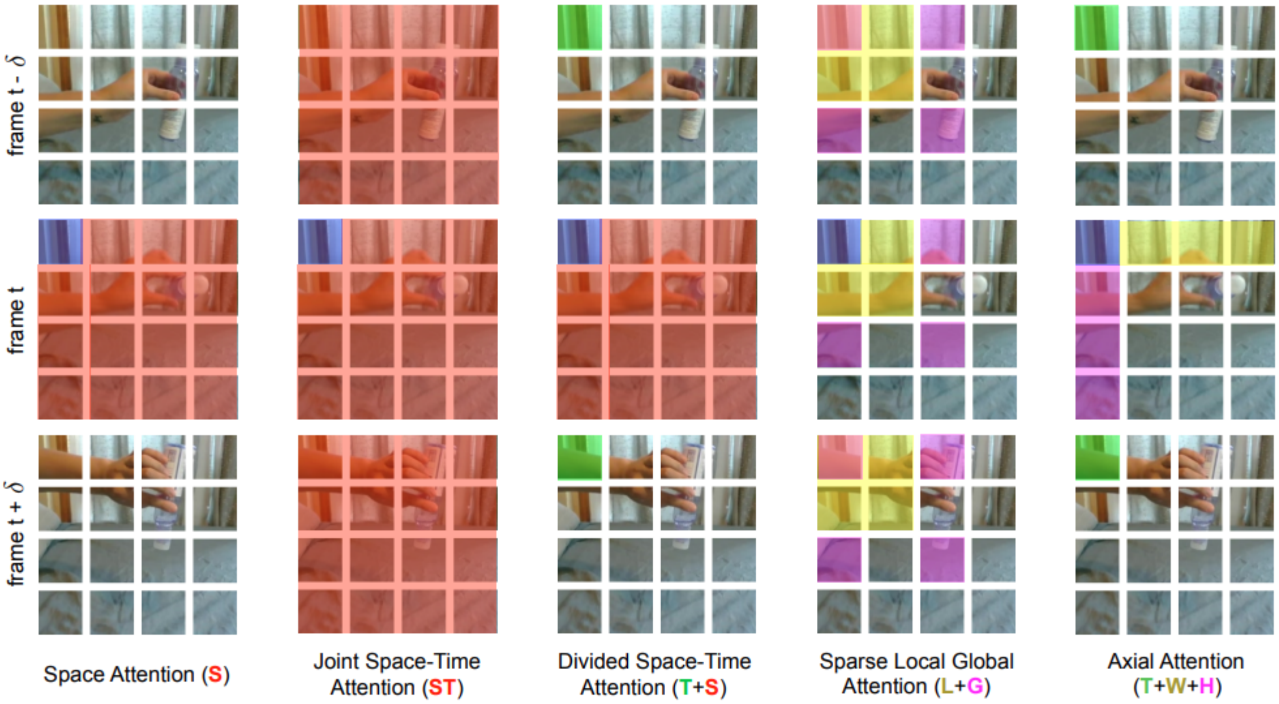
frame t - δ
frame t
frame t + δ
Space Attention (S)
Joint Space-Time Attention (ST)
Divided Space-Time Attention (T+S)
Sparse Local Global Attention (L+G)
Axial Attention (T+W+H)

Figure 2.5 – Visualization of different attention mechanisms in TimeSformer: demonstrating five approaches: space attention (S), joint space-time attention (ST), divided space-time attention (T+S), sparse local-global attention (L+G), and axial attention (T+W+H). Showcasing variations in the regions and frames emphasized during video analysis [10].

ViViT

The ViViT model is a transformer-based architecture specifically designed for video classification tasks. Building on the success of Vision Transformers (ViTs) in image recognition, ViViT extends the transformer framework to process spatiotemporal data. The model leverages self-attention mechanisms to capture both spatial features within individual frames and temporal relationships across sequences of frames. By using a purely transformer-based approach, ViViT eliminates the need for convolutional or recurrent modules, providing a unified framework for video understanding.

The ViViT model also relies on the success of ViT's in image recognition and, such as the TimeSformer model, extends the transformer framework to process spatiotemporal data. It also applies the self-attention mechanism to capture spatial features within frames and temporal features across sequences of frames. Much like the TimeSformer model, it creates non-overlapping patches, which are flattened, linearly embedded, tokenized and a positional embedding is also added to preserve sequential ordering of frames and spatial arrangement of patches.

It also offers architectural variants, seen on Figure 2.6. Joint spatiotemporal attention processes all tokens simultaneously, capturing global interactions across both spatial and temporal dimensions. While this method provides comprehensive feature modeling, it is computationally intensive, especially for long video sequences or high-resolution frames. The factorized spatiotemporal attention variant separates spatial and temporal attention into distinct stages. Spatial attention is first applied within individual frames to extract local features, followed by temporal attention to model dependencies across frames. This approach reduces the computational cost while maintaining performance.

Another variant, factorized encoder attention, uses separate transformer encoders for spatial and temporal processing. This design further simplifies the attention computation by isolating spatial and temporal interactions. The fourth variant, factorized dot-product attention, divides attention computations into spatial and temporal components across different attention heads, offering a balance between computational efficiency and representational capacity. These architectures leverage different tokenization methods and regularization techniques to handle the high computational demands of video data while achieving SOTA performance across multiple benchmarks [6].

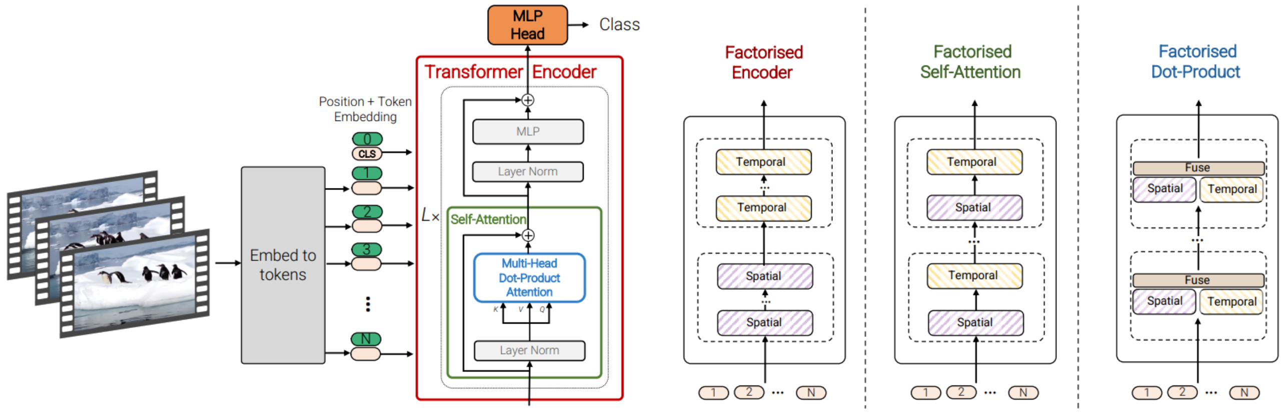

Figure 2.6 – Overview of the ViViT architecture: the model tokenizes video frames, applies positional embeddings, and processes them through a Transformer encoder with multiple layers of self-attention. Different factorization strategies are illustrated: (1) factorized encoder separates spatial and temporal encoding, (2) factorized self-attention splits attention operations across spatial and temporal dimensions, and (3) factorized dot-product fuses spatial and temporal features across different heads [6].

VideoMAE

VideoMAE draws it’s principles from masked autoencoders in NLP. It randomly masks video tokens (patches) at a high ratio (90–95%) and reconstructs the missing regions, using the temporal redundancy of videos to create a challenging reconstruction task. This process encourages the model to learn spatiotemporal representations. VideoMAE represents video tokens through temporal downsampling and a joint space-time embedding strategy, facilitating efficient handling of video data. It is built on a standard ViT backbone that employs joint space-time attention and achieves competitive results without depending on pre-training datasets or external labels.

VideoMAE extends the concept of Masked Autoencoders (MAEs) from NLP to the domain of spatiotemporal data, focusing on learning video representations without the need for extensive annotated datasets. The model processes video data by dividing each frame into patches, tokenizing these patches, and combining spatial patch extraction with temporal downsampling. A high proportion of tokens, typically ranging from 90% to 95%, is randomly masked during training, leaving only a small subset of visible patches. This masking strategy introduces a challenging reconstruction task for the model.

During training, VideoMAE reconstructs the masked patches using the information from the visible ones, driving the model to capture meaningful spatiotemporal features that account for both motion and appearance. This reconstruction task emphasizes learning temporal dynamics and spatial details simultaneously, which are critical for understanding video content. Figure 2.7 illustrates this tokenization and masking process. Various masking strategies were evaluated to determine their impact on model performance. As shown in Figure 2.8, tube masking, which masks tokens along both spatial and temporal dimensions, was found to achieve the best performance in experiments conducted in [95]. This approach ensures that the model learns coherent representations across frames, leveraging the inherent redundancy of video data to facilitate efficient self-supervised learning.

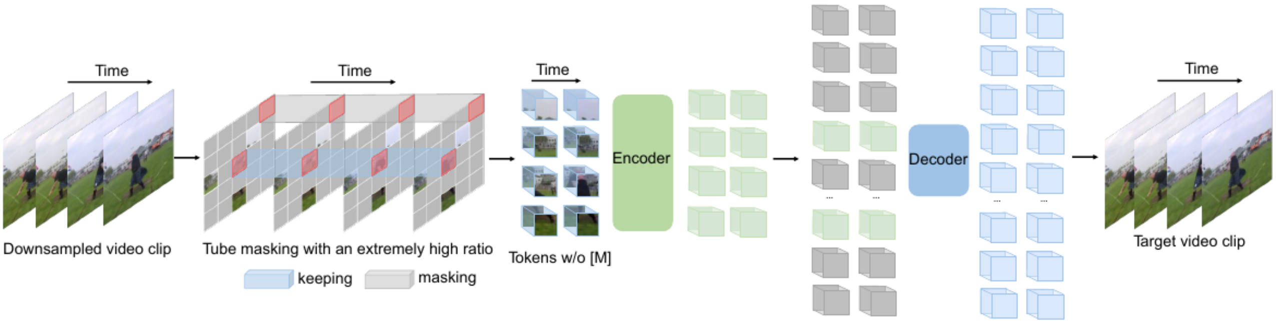

Figure 2.7 – "Illustration of the VideoMAE [95] training process, showcasing video downsampling, tube masking with a high ratio, encoding of visible tokens, and reconstruction of the masked content by the decoder to generate the target video clip."

Similar to masked language modeling, this setup exploits the redundant nature of videos to make reconstruction a challenging task, pressing the model to identify relevant structures over time rather than focusing solely on single-frame patterns. VideoMAE employs a standard ViT backbone, where tokens are processed by multi-head self-attention in space and time. The key difference from pure Transformer architecture lies in the reconstruction objective: the model predicts missing patches to minimize a reconstruction error, typically in the pixel or feature space. By handling videos in a tokenized, masked framework, VideoMAE avoids labeling requirements and can harness large-scale unannotated video data for representation learning.

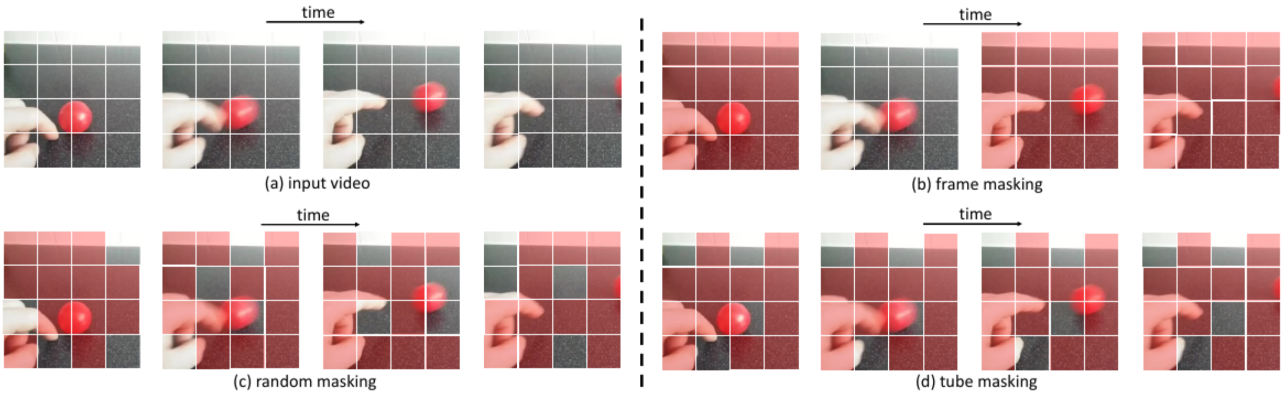

Figure 2.8 – "Comparison of masking strategies in VideoMAE [95]: (a) input video without masking, (b) frame masking applied across entire frames, (c) random masking applied to scattered patches, and (d) tube masking applied along spatial and temporal dimensions, capturing consistent patterns across time."

When fine-tuned on downstream tasks, such as action recognition, VideoMAE often achieves favorable results compared to supervised or other self-supervised methods, highlighting the effectiveness of learning from extensive temporal context. Since the model operates on a “vanilla” ViT backbone, it inherits the modular advantages of Transformers, including the capacity for parallel processing and flexible integration of spatiotemporal cues. Because masking leads to a reduced computation burden on the network, the approach also scales well, making it feasible to handle sizable video datasets without heavy external supervision [95].

2.4 Metrics

All experiments focus on word-level recognition in LIBRAS, where each video is labeled with exactly one sign class. In a multi-class classification scenario, each class is treated as the “positive” class in turn, while all other classes are grouped as “negative”. Metrics such as accuracy, precision, recall, and F1-score are then computed for each class i, and the final result is obtained by taking the macro-average across all classes. This approach ensures that each class, even if it has fewer samples, is weighed equally in the overall evaluation, an essential consideration when the class distribution is not perfectly balanced [11, 69].

When training each model, the MINDS dataset’s validation set guides early stopping and learning rate adjustments. The newly curated subset of the MALTA-LIBRAS dataset is used exclusively as an unseen test set to measure model generalization.

- **Accuracy:**

  Accuracy provides a straightforward ratio of the total number of correctly classified samples to the total number of samples [65, 11]. In binary form:

  $$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}.$$

  For multi-class classification, we extend this concept by counting correct predictions for each class i and then computing the overall fraction of correct predictions:

  $$\text{Accuracy} = \frac{\sum_{i=1}^{C} \text{Correct}_i}{\sum_{i=1}^{C} \text{Total}_i},$$

  where Correct_i is the number of correctly classified samples of class i, and Total_i is the total number of ground-truth samples of class i. Although accuracy is easy to interpret, it may not fully capture performance on minority classes in unbalanced datasets [69].

- **Precision:**

  Precision captures the proportion of samples predicted as belonging to a given class i that actually do belong to that class [11]. In the binary setting:

  $$\text{Precision} = \frac{TP}{TP + FP}.$$

  To handle multiple classes, each class i is considered "positive" in turn, yielding Precision_i. The macro-average precision aggregates these values equally:

  $$\text{Precision}_{\text{macro}} = \frac{1}{C} \sum_{i=1}^{C} \frac{TP_i}{TP_i + FP_i}.$$

  A high macro-average precision indicates that, across all classes, when the model predicts a certain sign, it is typically correct.

- **Recall:**

  Recall (also called sensitivity) quantifies the fraction of actual positives the model correctly identifies [69]. In the binary case:

  $$\text{Recall} = \frac{TP}{TP + FN}.$$

  For multiple classes, each class i again becomes the positive class, and Recall_i is computed. The macro-average recall is the unweighted mean over all classes:

  $$\text{Recall}_{\text{macro}} = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{TP_i + FN_i}.$$

  A high macro-average recall indicates that, on average, classes rarely go undetected.

- **F1-score:**

  The F1-score is the harmonic mean of precision and recall, striking a balance between them [69]. In binary form for class i:

  $$\text{F1}_i = 2 \times \frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}.$$

  Extending to a multi-class context, we compute F1_i for each class i and take the average across all classes:

  $$\text{F1-score}_{\text{macro}} = \frac{1}{C}\sum_{i=1}^{C}\text{F1}_i.$$

  This measure provides a single score that accounts for both false positives (through precision) and false negatives (through recall). It is often considered more informative than accuracy for unbalanced datasets, as it requires strong performance across both dimensions [11].

Throughout our experiments, top-1 accuracy remains the primary measure, while macro-averaged precision, recall, and F1-score offer additional insights into error patterns. Metrics are averaged over multiple runs with different random seeds to mitigate variability in training and allow a more reliable assessment of each model's ability to generalize.

3. RELATED WORK

The ISLR task is conceptualized as a multi-class classification task where each video serves as the input variable and the corresponding sign serves as the target variable. Consequently, every unique sign in the vocabulary constitutes a distinct class that the model must learn to discriminate. In this framework, the spatiotemporal features captured in consecutive frames, which encompass hand configurations, facial expressions, and body movements, are analyzed by the model to classify the intended sign. This task poses unique challenges, as the models must effectively integrate both spatial details and temporal coherence to recognize signs, a requirement that has driven the exploration of diverse DL architectures [23, 46].

Datasets	Language	Signers	Vocabulary	Vídeos	Task	Input
SIGNUM [98]	German	25	450	11,375	CSLR+ISLR	RGB
RWTH-PHOENIX14 [31]	German	9	1,117	4,667	CSLR+ISLR	RGB
CSL [44]	Chinese	50	500	125,000	ISLR	RGB+D
WLASL-1000 [53]	English	222	1,000	25,513	ISLR	RGB
SLOVO [49]	Russian	194	1,000	20,000	ISLR	RGB
AUTSL [86]	Turkish	43	226	38,336	ISLR	RGB+D
GrSL-Isol. [2]	Greek	7	310	40,785	ISLR	RGB+D
MINDS-LIBRAS [**?**]	Portuguese	12	20	1,200	ISLR	RGB+D
MALTA-LIBRAS (ours)	Portuguese	60	4,586	11,093	ISLR	RGB

Table 3.1 – Well-stablished datasets on SLR for both ISLR and CSLR tasks. Input differs on using solely RGB data and RGB with depth (D) information.

Existing datasets for SLR vary in annotation granularity and video capture modalities, with some limited to RGB frames and others incorporating depth data [8, 23]. Key characteristics of these datasets, including the novel MALTA-LIBRAS corpus under development, are detailed in Table 3.1. While differing in scope, all these datasets exhibit large signer diversity, vocabulary size and total number of videos, all critical factors for enhancing the robustness of DL models for real-world applicability [2, 3].

Recent advancements leverages weights from models pre-trained on action recognition tasks, capitalizing on large-scale datasets [51, 82, 76]. Alternatively, a common approach involves keypoint extraction, where joint positions of the body, hands, or face are tracked in videos using tools like OpenPose [16] and MediaPipe [61], enabling pose-driven gesture modeling [68, 46, 72].

Specialized capture methods, such as depth-sensing cameras and sensory gloves, have also been investigated [29, 41]. While these devices improve motion tracking accuracy, their reliance on hardware constraints, such as user discomfort and research-specific setups, limits practical adoption. Moreover, such methods often overlook complex and critical aspects of SL communication, such as facial expressions, which are essential to semantic interpretation [46, 99].

3.1 Models

ISLR targets word-centric gestures in video and still presents itself as a challenge in various SLs. These challenges mainly stem from the restricted size and diversity of annotated samples, as well as the simultaneous integration of hand movements, orientations, and non-manual cues (e.g., facial expressions, head poses) required to convey meaning [87, 53, 8, 39].

Common ISLR approaches explore DL models that account for spatial-temporal patterns in signing. Usual pipelines rely on pose estimations (e.g., Mediapipe or OpenPose) to isolate hand coordinates before classification via RNNs [61, 16]. Recent work, however, focuses on end-to-end models [71, 23].

Hybrid models, utilizing combinations of CNNs and RNNs, exploit the spatial feature extractions from convolutional layers and temporal modeling from recurrent networks. Earlier ISLR work was heavily dependent on these hybrid networks due to data scarcity, where utilizing augmentations such as optical flow and pose estimators were the go-to alternative [54].

Building upon classic 2D CNNs, inflated 3D CNNs emerged with the intention to be able to also handle the temporal domain, Inflated 3D ConvNets (I3D) is an example of these networks. However, due to the number of convolutional layers they can be computationally expensive. Additionally, their receptive fields still needs to be careful initialized to capture subtle body or hand movements [51].

Inspired by breakthroughs in NLP, attention-based models have increasingly been tested on SLR. Early uses of Transformers in SL have shown that the global context captured by attention can improve generalization, especially when balancing hand shape, facial expression, and body posture [71, 24].

Within the Transformer-based model alternatives, models such as TimeSformer [10], ViViT [6], and VideoMAE [95] have begun to shape spatiotemporal modeling in videos. These adaptations replace or supplement convolutions with tokenized patches and multi-head self-attention. By allowing any patch to directly attend to others in a video, Transformers sidestep limitations posed by purely sequential or local operations. However, researchers also grapple with high computational demands and data scarcity, especially when SLs are low-resourced.

3.2 Datasets

Within the dataset realm, most existing SL datasets were originally developed for educational or instructional purposes, not specifically for training DL models. Researchers address this gap by collecting specialized resources that facilitate training and benchmarking. RWTH-PHOENIX-Weather 2014 is an example, aimed at the CSLR task in GSL. It encompasses weather forecast recordings paired with word and sentence level annotations, serving as a reference for modeling complexities such as signer variation and infrequent vocabulary. An n-gram language model was trained and tested on a single-signer subset and attained a Word-Error Rate (WER) of 49.2%, highlighting the need for robust data design and models that adapt to linguistic variety [31].

A dedicated dataset for CSL was compiled to explore the CSLR task. This collection surpasses 25,000 labeled video sequences from 50 signers, captured in RGB, depth, and body-joint modalities using a Microsoft Kinect sensor. The recorded data averaged seven words per sentence and was allocated to training a total of 17,000 clips, validation of 2,000 clips, and testing of 6,000 clips. A Hierarchical Attention Network with Latent Space (LS-HAN) showed that an extensive dataset combined with attention-based modeling, can capture spatiotemporal properties effectively. LS-HAN achieved 82.7% accuracy and performed strongly when evaluated on RWTH-PHOENIX-Weather as well, illustrating the benefits of a self-attention approach without explicit segmentation [44].

In ASL, the Word-Level American Sign Language (WLASL) dataset is comprised of of videos focusing on the ISLR task. It includes over 20,000 clips of more than 2,000 glosses, each gloss displayed by multiple signers to accommodate inter-signer diversity. Subsets (e.g., WLASL-100, WLASL-300, WLASL-1000, WLASL-2000) vary the training and testing conditions. Strategies such as the Inflated 3D CNN (I3D) and a pose-driven Temporal Graph Convolutional Network (Pose-TGCN) illustrate the effectiveness of spatiotemporal modeling approaches. On WLASL-2000, I3D achieved a top-1 accuracy of 32% and a top-10 accuracy of 66%, while Pose-TGCN reached 23.65% top-1 accuracy and 62% top-10. The discrepancy between top-1 and top-10 highlights the complexities arising in large-vocabulary, word-level sign recognition [53].

Researchers also introduced the SLOVO dataset for RSL, featuring 20,000 videos of 1,000 distinct words recorded in varied settings. In their experiments, each video was resized and augmented before training. Three models were experimented, leveraging pre-training weights from the action recognition dataset Kinetics-400 [51]: MViTv2-small [55], Swin-large [59], and ResNet3D-50 [51]. Different frame sampling strategies and frame interval settings from 1 to 4 were tested. MViTv2-small achieved an accuracy of 64% when using 32 frames with interval of 2 frames, whereas ResNet3D-50 achieved an accuracy of 44% when using 48 frames with interval of 1 frame, suggesting that Transformer-based designs outperform 3D CNN approaches in modeling spatiotemporal features. By capturing inter-signer and environmental diversity, SLOVO provides a testbed for ISLR that seeks to handle large vocabularies and unconstrained real-world scenarios [49]. This direction parallels work in other SLs, where greater data coverage supports more flexible and robust models, an approach similarly sought for LIBRAS, where data is limited.

For LIBRAS, data is scarce. The MINDS dataset was introduced to support ISLR in LIBRAS using a standardized protocol. It is comprised of 1,158 annotated videos featuring 20 signs, with each sign performed 5 times by 12 signers, and each video summarized into 10 frames. In initial experiments, a 3D CNN architecture was explored, testing both RGB and grayscale inputs, the inclusion of optical flow, and limited data augmentation techniques such as temporal frame displacement, horizontal flips, and zooming. Results indicated that normalizing test samples in accordance with training data, as well as increasing the training sample count through augmentation techniques, led to improved accuracy.

The best-performing method, which incorporated grayscale frames and the data augmentation pipeline, achieved an average accuracy of 93%. One constraint of the experiments, however, was the absence of a designated test set, leaving training-validation splits to serve as the sole basis for assessing performance [78].

Despite this limitation, the MINDS dataset demonstrates how targeted data augmentation techniques can enhance ISLR in low-data conditions and lays groundwork for further LIBRAS research. This configuration makes it challenging to assess how well models generalize to broader, real-world scenarios with more signers, different lighting conditions, and varying signing styles.

Building on insights from larger corpora in other SLs and the MINDS dataset, we introduce the MALTA-LIBRAS dataset, which intersects with the 20 classes from MINDS. MALTA-LIBRAS was assembled from diverse, open-source materials, aiming for a broader distribution of signers and recording settings. This extension of 131 novel videos parallels efforts in other SLs, where expanding vocabulary coverage and environmental complexity supports evaluations of different DL architectures, including those based on Transformers.

The present work in LIBRAS exploration utilizes these insights by focusing on spatiotemporal Transformer-based video models, targeted data augmentation pipeline, and cross-dataset experiments. The aim is to address the scarce representation in LIBRAS, bridging the gap between smaller, controlled datasets such as MINDS and real-world conditions. The combination of carefully curated data and strategies that capture both spatial details and temporal cues is key to designing systems capable of accomplish the ISLR task in LIBRAS. By unifying these strategies, this study seek to extend achievements from existing SL corpora and architectures to the specific needs of LIBRAS ISLR.

4. RESEARCH GOALS

SLR presents unique challenges that extend beyond the analysis of generic human actions. A single sign may blend distinct handshapes, orientations, and motion trajectories with facial expressions or head poses, all of which guide how the sign is perceived and interpreted [53, 87]. To unravel these features it is necessary robust datasets and models that are able to capture both spatial and temporal features, as well as inter-signer variability.

Recently, languages such as ASL and CSL have been widely explored, leading to progress in the ISLR task due to large and diverse datasets. However, LIBRAS remains under-resourced, limiting systematic development of data-driven approaches. MINDS [78] offers a structured basis for LIBRAS ISLR yet contains a small vocabulary and insufficient samples to form a standalone test set. These constraints make it difficult to assess how well models can generalize beyond the dataset's restricted vocabulary.

We outline two objectives in this dissertation:

- **I.** Expand current LIBRAS data by gathering novel videos that may serve as a designated test set;
- **II.** Explore how different approaches using transformer-based video classification architectures can enhance ISLR in LIBRAS.

By leveraging advanced spatiotemporal attention mechanisms and previous knowledge attained from action recognition datasets, the aim is to address the scarcity and narrow coverage of the only current available LIBRAS data. Investigations encompass developing a suitable data augmentation pipeline and cross dataset transfer learning. These steps intend to improve the reliability of LIBRAS ISLR while minimizing the need for extensive manual annotation.

4.1 Research Questions

To address our objectives we've devised three Research Questions (RQ) we answer in this dissertation followed by the methods used to achieve those answers.

- **RQ1:** How does pre-training in action recognition task affect LIBRAS ISLR?
- **RQ2:** How data augmentation impacts LIBRAS ISLR?
- **RQ3:** How cross-sign-language transfer learning, from ASL to LIBRAS; RSL to LIBRAS; and combined ASL and RSL to LIBRAS, affect LIBRAS ISLR?

To answer each of our RQs, we will follow a specific methodology. The first step involves gathering and curating LIBRAS videos from open online sources to increase available data for experimentation. After curating these videos, the resulting samples will serve as a separate test set, complementing MINDS data, thereby enabling the evaluation of model generalization. These curated samples comprehend what we call MALTA-LIBRAS data set. A series of experiments will then be conducted to address each RQ, ensuring robustness and mitigating the impact of random fluctuations. All experiments will be performed with ten preset seeds and performance metrics will then be averaged across these runs.

To answer RQ1, three attention-based video classification models will be evaluated. Each model will be tested both with pre-trained weights, initially trained on action recognition tasks, and with randomly initialized weights. Models with pre-trained weights will be fine-tuned on MINDS and then tested on the newly curated MALTA-LIBRAS set, while models with randomly initialized weights will be trained solely on MINDS and tested on the same novel set.

To address RQ2, a video data augmentation pipeline will be explored, incorporating random rotation, random perspective, color jitter, and Mixup [103]. Each technique will be applied separately at various intensities during training, validated on the MINDS dataset, and ultimately tested on MALTA-LIBRAS. The goal is to identify which augmentation approach best improves generalization.

For RQ3, we investigate cross-lingual transfer learning by fine-tuning the VideoMAE model (pretrained on Kinetics-400) end-to-end on the WLASL and SLOVO datasets. The model is then fine-tuned on MINDS without freezing any layers, ensuring full parameter updates, and finally evaluated on the MALTA-LIBRAS subset to assess generalization to unseen LIBRAS signs. WLASL, focused on ISLR in ASL, contains videos in more controlled settings, whereas SLOVO addresses ISLR in RSL with more varied scenarios. Models will be trained on these two datasets, then fine-tuned on MINDS and tested on MALTA-LIBRAS to determine whether knowledge from other SLs can transfer effectively to LIBRAS. The outcomes of these experiments may prompt further expansions in LIBRAS data collection, cross-dataset transfer learning studies, and potential exploration of alternative approaches for ISLR in LIBRAS.

5. METHODOLOGY

This chapter presents the steps and rationale behind the experiments conducted in this dissertation. The goal is to evaluate Isolated Sign Language Recognition (ISLR) in LIBRAS using a newly introduced MALTA-LIBRAS dataset as a test bed. Section 5.1 formalizes ISLR as a classification task. Section 5.2 describes how we quantify model performance of all our experiments. Section 5.3 details data collection and curation for MALTA-LIBRAS, and all datasets used in our experiments. Section 5.4 outlines the selected transformer-based video classification models and their configurations. Section 5.5 describes in detail our experiments and rationale to answer each of our RQs.

5.1 ISLR as a Classification Problem

ISLR is, in it's core, a supervised multi-class classification task where a model learns to map input videos to discrete sign labels. Formally, given a dataset $\mathcal{D} = \{(\mathbf{X}_i, y_i)\}_{i=1}^{N}$ of N labeled videos, the goal is to learn a function $\varphi : \mathcal{X} \to \mathcal{Y}$ that minimizes prediction errors on unseen test data. Here, $\mathcal{X}$ denotes the input space of videos, and $\mathcal{Y} = \{1, \dots, K\}$ represents the set of K possible sign classes (e.g., LIBRAS glosses) [5, 20, 36, 94].

As an input $\mathbf{X} \in \mathcal{X}$ we have a video sequence of T frames:

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T], \quad \mathbf{x}_t \in \mathbb{R}^{H \times W \times C}, \tag{5.1}$$

where H, W, and C are the height, width, and channels (e.g., RGB), respectively, of each frame. Commonly, for computational efficiency, videos are downsampled to fixed dimensions (e.g., 224×224 pixels) and truncated to a uniform number of frames T [51].

The output $y \in \mathcal{Y}$ is a categorical label representing the lexical sign in $\mathbf{X}$. For instance, in LIBRAS, $y = 1$ might denote the sign for “book” and $y = 2$ for “help”. The model outputs a probability distribution over $\mathcal{Y}$:

$$\mathbf{p} = [p_1, p_2, \dots, p_K], \quad \sum_{k=1}^{K} p_k = 1, \tag{5.2}$$

where $p_k = P(y = k \mid \mathbf{X})$ is the estimated probability of class k [11]. The function φ is parameterized by a network that extracts spatiotemporal features from $\mathbf{X}$ and computes class probabilities. For transformer-based models (discussed in detail in Section 5.4), this involves a few steps:

- **Tokenization**: First, each frame is splitted into $P \times P$ patches, yielding tokens $\mathbf{z}_t \in \mathbb{R}^{N \times d}$ for frame t, where $N = \frac{HW}{P^2}$ and d is the embedding dimension [27].
- **Positional Encoding**: Then, it's injected spatial and temporal positional information into each token [96].
- **Self-Attention**: It's then modelled dependencies between tokens across space and time dimensions [10].
- **Classification Head**: The aggregated features is then mapped into class probabilities through two steps:
  - **Linear Projection**: First, transforming the hidden state $\mathbf{h}$ into class scores (logits):

    $$z_k = \mathbf{w}_k^\top \mathbf{h} + b_k, \tag{5.3}$$

    where $\mathbf{w}_k \in \mathbb{R}^d$ and $b_k \in \mathbb{R}$ are learnable parameters for class k.
  - **Softmax Normalization**: Then, logits are converted $\{z_1, ..., z_K\}$ into probabilities by a softmax function:

    $$p_k = \frac{\exp(z_k)}{\sum_{j=1}^K \exp(z_j)} = \text{softmax}(z_k) \tag{5.4}$$

    This ensures $0 \leq p_k \leq 1$ and $\sum_{k=1}^K p_k = 1$.

  The final hidden state $\mathbf{h}$ (e.g., from the [CLS] token) contains compressed spatiotemporal information from all patches. The softmax operation emphasizes large logit values while suppressing smaller ones, producing interpretable probabilities [36].

5.1.1 Loss Function

In general, classification models are optimized to minimize their loss function during training. For multi-class scenarios, the categorical cross-entropy loss function (CCE) is commonly employed to measure the discrepancy between the predicted probability distribution and the true distribution, which is typically represented using one-hot encoding [36].

Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, where each $\mathbf{y}_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,C}]$ is a one-hot vector indicating the correct class among C possible classes, and $\hat{\mathbf{y}}_i = [\hat{y}_{i,1}, \hat{y}_{i,2}, \ldots, \hat{y}_{i,C}]$ denotes the model's predicted probabilities for each class, the categorical cross-entropy loss is defined as [36]:

$$\mathcal{L}_{\text{CCE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c}) \tag{5.5}$$

This loss function penalizes deviations between the predicted probabilities $\hat{y}_{i,c}$ and the ground-truth indicators $y_{i,c}$ for each class:

- When $y_{i,c} = 1$, the term $\log(\hat{y}_{i,c})$ encourages $\hat{y}_{i,c} \rightarrow 1$.
- Since $y_{i,c} = 0$ for all other classes, only the correct class contributes to the summation, ensuring that the model focuses on assigning high probability to the true class.

Minimizing $\mathcal{L}_{\text{CCE}}$ over the entire dataset aligns the model's predicted probability distribution with the actual labels, ensuring that if the model assigns a low probability to the correct class, $\log(\hat{y}_{i,c})$ becomes large and negative, increasing the overall loss. Alternatively, if the model consistently assigns high probability to the correct class across all samples, the loss decreases, indicating better performance [11].

As a result, CCE serves as a direct measure of how confident the model is about its predictions. By driving $\hat{y}_{i,c}$ towards 1 for the correct class, this loss effectively optimizes model precision while preserving a proper probability interpretation of the outputs [69]. This makes CCE a standard choice in multi-class settings such as image classification, speech recognition, and, in the case of this dissertation, isolated sign language recognition.

5.2 Metrics

In this dissertation, the evaluation of classification performance relies on top-1 accuracy, along with macro-averaged precision, recall, and F1-score to account for potential class imbalances and provide a balanced view of model performance. These metrics are discussed in greater detail in Chapter 2, where their definitions and extensions to multi-class scenarios are thoroughly explained.

5.3 Datasets

To achieve the objectives of this study and answer our RQ, we utilized SLOVO [49], WLASL [53], MINDS [78], and our novel MALTA-LIBRAS dataset. SLOVO and WLASL are used as pre-training datasets to assess how transfer learning between distinct SL affect LIBRAS ISLR.

These datasets were selected based on their size and distinct focus: WLASL emphasizes maintaining control over instances, while SLOVO prioritizes covering various scenarios. This combination enables us to answer our RQ3, thus evaluating the effectiveness of transfer learning for LIBRAS and how the differences between datasets influence the models. On the other hand, MINDS and MALTA-LIBRAS are used to fine-tuning, validation and testing the models.

All video samples were resized to a standardized resolution of 224×224 pixels. Due to variations in recording protocols (e.g., frame rates, video durations) across datasets, samples exhibited significant disparities in total frame counts. To ensure uniformity, we extracted 32 temporally equidistant frames from each video. During training, a subset of 16 frames was randomly selected from these 32 preprocessed frames to serve as input to the model, balancing computational efficiency with temporal representation and matching the required number of frames for all models used. The following subsections provide detailed specifications for each dataset.

5.3.1 MALTA-LIBRAS

To address the limitation of the MINDS dataset, namely the absence of a dedicated test set and its relatively controlled recording conditions, we collected and curated additional videos to perform testing in models fine-tuned to MINDS. Initially, using web scraping techniques, we compiled a total of 21,000 LIBRAS videos from INES Dictionary 2.0 [56], INES Dictionary 3.0 [57], V-LIBRASIL [79], Libras-Corpus [75], SignBank [74], UFV LIBRAS Dictionary [33], and Spread the Sign [42] platforms. These dictionaries alone, with statistics shown in Table 5.1, did not contain enough samples per sign to serve as a standalone dataset, so combining them was a necessary step. For our experiments, we used a curated subset of our dataset, with samples aligned with the same class labels as the MINDS dataset. The resulting subset introduces variability in terms of signers, backgrounds, and recording conditions, while encompassing the same 20 classes as the MINDS dataset but featuring different signers and environments. Due to data usage policies, we provide instructions on how to download all MALTA-LIBRAS data in our repository, along with all annotations and code used in this dissertation[1].

After assembling this video collection, we undertook manual analysis of all collected data. Each video was examined independently at least by two different researchers to label the corresponding word or phrase and to identify the signer performing it. Video resolutions vary between 240×180 and 1280×720 pixels, and frame rates range from 24 to 30 frames per second. The MALTA-LIBRAS dataset was exclusively used for testing to

[1] https://github.com/Malta-Lab/ISLR_LIBRAS.git

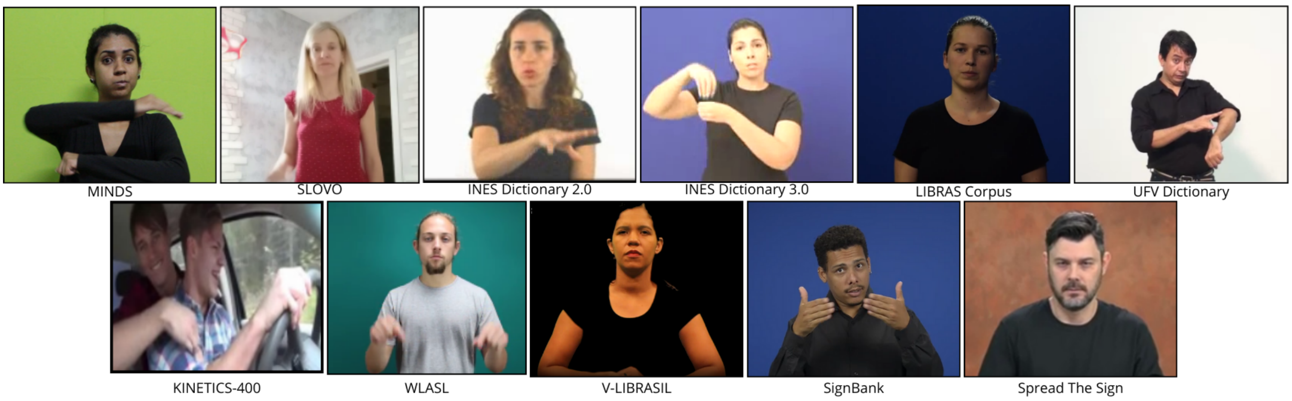

Figure 5.1 – An illustration showcasing a single frame from all datasets used and from each collected source from MALTA-LIBRAS dataset, emphasizing the variability of data in the creation of the MALTA-LIBRAS dataset.

evaluate the model's ability to generalize to new signers and environments not encountered during training.

Since we design MALTA-LIBRAS to be a test set for models trained in MINDS, we are interested only in a subset of signals that appear on both datasets. The first data source collected was from the National Institute of Deaf Education (INES) online LIBRAS dictionaries. Data collection yielded two collections of videos: one from LIBRAS Dictionary 2.0 [56] and another from LIBRAS Dictionary 3.0 [57]. From these, we obtained 24 and 23 video instances that matched 20 and 19 signs in the MINDS dataset, respectively. We also explored the V-LIBRASIL [79] dataset, yielding 26 video instances across 9 signs that align with the MINDS dataset. The LIBRAS Corpus [75] provided 18 instances across 19 signs. UFV LIBRAS Dictionary [33] provided 7 instances for 7 different signs. Signbank [74] contributed 19 instances covering 17 signs featured in the MINDS dataset. Additionally, we utilized data from Spread The Sign [42], an initiative by the European Sign Language Center that features over 400,000 video signs across 44 languages. From this dataset, we were able to match only 12 instances for 12 signs within the MINDS dataset.

The resulting collection, our MALTA-LIBRAS dataset, includes 129 unique samples covering the 20 signs that intersect with the MINDS dataset. We preprocessed each video to be resized to 224×224 pixels. To lower the compute demand we extract 32 frames

Table 5.1 – Specification of each open data source collected through web scraping. Last column shows the intersection size between our dataset and MINDS dataset.

	Signers	Videos	Length [h]	Resolution	|Vocabulary|	$\lvert\bigcap$ MINDS$\rvert$
INES Dictionary 2.0 [56]	1	3,287	1.19	240x180	3,066	20
INES Dictionary 3.0 [57]	1	3,208	3.34	240x180	2,993	19
V-LIBRASIL [79]	3	4,089	6.49	1920x1080	1,363	7
Libras-Corpus [75]	1	1,011	0.77	360x300	929	19
SignBank [74]	9	1,000	0.66	1280x720	916	20
UFV LIBRAS Dictionary [33]	6	242	0.27	831x467	242	6
Spread the Sign [42]	40	1,788	1.79	320x240	1,788	16

uniformly spaced throughout the video, and as a part of our data augmentation strategy, among these 32 frames we randomly sample 16 frames that serve as model input. This approach ensured a manageable and consistent input size while capturing essential temporal information.

Figure 5.1 illustrates that each dataset was recorded under predefined protocols, including standardized clothing, backgrounds, and resting poses, in order to reduce variability and assist students in focusing on learning sign features. While such consistency may help individuals understand signs more easily, it limits the model's exposure to diverse conditions and thus restricts its ability to adapt to more realistic scenarios.

5.3.2 MINDS

The MINDS-Libras dataset was developed to serve as a standardized and reproducible resource for research on ISLR in LIBRAS [78, 78]. The creation process followed a strict protocol to ensure consistency, diversity, and the ability to replicate the methodology. The dataset includes 20 signs carefully selected by a LIBRAS expert to capture a wide range of phonological parameters. These parameters encompass aspects such as movements, palm orientations, and non-manual expressions, ensuring the dataset represents the linguistic complexity of LIBRAS.

To compile the dataset, twelve signers participated in the recording sessions, varying in gender, age, hearing status, and fluency in LIBRAS, allowing the dataset to account for some of the natural variability present in real-world scenarios. Participants were not required to follow strict clothing guidelines, although a significant number chose to wear black. The recording protocol resulted in videos with a resolution of 1920x1080 pixels, all conducted in a controlled studio with fixed lighting and a chroma key background, which allows for potential manipulation of the background. Additionally, the distance between the participant and the recording device was standardized to achieve uniformity in the framing of gestures.

During the recording process, each of the 12 participants performed the 20 selected signs five times, resulting in an initial total of 1200 samples. Participants were instructed to maintain a neutral facial expression and start each gesture from a fixed position with hands at their sides. Each recording lasted five seconds, capturing 150 frames at a rate of 30 frames per second.

The dataset encompasses multiple types of data, including RGB videos, depth videos, and textual files containing body and face point coordinates. This multimodal approach ensures comprehensive coverage of the gestural elements critical for SLR. The data underwent preprocessing to separate individual gestures, ensuring usability across different research applications. Despite careful planning, some data were lost due to equipment malfunctions, reducing the final dataset to 1158 samples.

One of the limitations of the MINDS dataset is the absence of a designated testing set, making it difficult to assess model generalization to data not seen during training. Without this dedicated set and without a larger dataset focused on LIBRAS, it is not straightforward to determine how well models trained on MINDS would perform in other scenarios.

5.3.3 WLASL

The WLASL [53] dataset was created to provide a large-scale resource for ISLR in ASL. The dataset was developed to address the limitations of existing resources, such as their small vocabulary sizes and limited signer diversity, which hinder scalability and practical applications.

To build the dataset, the authors used two main sources. The first source comprised educational SL websites, including ASLU [97] and ASL-LEX [17], which provided gloss-to-sign mappings verified by experts. The second source involved ASL tutorial videos available on YouTube. Videos were selected based on clear titles that described the gloss being signed. In total, 68,129 videos covering 20,863 unique ASL glosses were gathered from 20 websites. These videos featured signers performing a single sign in a near-frontal view with various backgrounds.

The dataset underwent a thorough filtering process. Videos with gloss annotations consisting of more than two English words were excluded to ensure consistency with single-word glosses. Glosses with fewer than seven videos were also removed, as such small sample sizes would compromise the dataset's utility for training DL models. This step reduced the dataset to 34,404 video samples covering 3,126 glosses.

Annotations were then added to each video to enhance its utility for ML applications. Each video was labeled with its corresponding gloss, along with metadata such as temporal boundaries, body bounding boxes, signer identification, and dialect variations. Temporal boundaries were defined to indicate the start and end frames of a sign. In cases where videos included repeated signs, only one instance was retained to avoid redundancy during training. Body bounding boxes were identified using the YOLOv3 [77] detection algorithm to minimize background interference. Signer identities were determined using FaceNet embeddings, ensuring diversity in signer appearances and reducing potential biases in model training.

Dialect variations were also annotated to reflect linguistic diversity within ASL. Annotators, trained to understand ASL nuances, labeled these variations by comparing signs from different videos. Variations with fewer than five examples were discarded to maintain sufficient data for training, validation, and testing splits.

The final dataset was structured into four subsets based on vocabulary size: WLASL-100, WLASL-300, WLASL-1000, and WLASL-2000, containing 100, 300, 1,000, and 2,000 glosses, respectively. These subsets facilitated the evaluation of models under varying levels of complexity. The largest subset, WLASL2000, comprised 21,083 videos with an average duration of 2.41 seconds and 10.5 samples per gloss. In our study we chose to employ the 1,000-Class subset. This choice of subset provides us with an average of 8 training instances and 4 validation instances per class for pre-training to evaluate the effectiveness of transfer learning from ASL to LIBRAS, aiming to enhance our model's ability to recognize LIBRAS signs by leveraging knowledge learned from ASL.

5.3.4 SLOVO

The creation of the SLOVO [49] dataset, a novel large-scale dataset for RSL, followed a structured three-stage process: video collection, validation, and gesture annotation. The dataset was designed to address the scarcity of diverse, high-quality RSL data and involved a total of 194 signers producing 20,000 videos across 1,000 classes of isolated RSL gestures.

Initially the authors selected 1,000 glosses to represent the most frequent words used in daily life. These glosses covered topics such as food, emotions, colors, and animals. Crowdworkers were recruited through an online platform to record videos of themselves signing these glosses. Each participant was provided with templates sourced from the Spread The Sign [42] website to standardize the signing process.

Subsequently, a video validation process was conducted to assess if the quality of video was in conformity to what was expected. This step was also executed by crowdworkers and each video underwent verification by at least three different validators. In cases of disagreement, additional validators were brought in, with up to five reviewers per video. Videos failing to meet quality standards were excluded from the dataset. To maintain technical consistency, accepted videos were resized to a minimum edge of 720 pixels and converted to a frame rate of 30 frames per second.

Finally, gesture annotation addressed the presence of uninformative frames at the start and end of the recordings, often where participants prepared for or concluded their signing. Crowdworkers also annotated the temporal boundaries of each gesture, marking the start and end times. In situations where a video contained multiple gestures, these were annotated as a single gloss. Each video was annotated by three different workers, and the final boundaries were calculated using an aggregation algorithm. This process identified average start and end points while excluding annotations with high variance.

During post-processing, it was observed that some signers produced gestures significantly slower than others, leading to inconsistencies in video duration within the same gloss. To address this, videos that were slower than the class average by more than 30 frames were sped up, resulting in more homogeneous data. This adjustment improved the dataset's usability for ML applications.

The final dataset comprises of HD and FullHD videos with an average length of 1.67 seconds. It includes a "no event" class derived from frames with no signing activity, contributing 400 additional samples. The dataset was split into training and validation sets, with a 75%-25% split ratio. Efforts were made to minimize signer overlap between these sets to mitigate a overfitting scenario. The SLOVO dataset's design and methodology prioritize diversity and quality, making it a valuable resource for ISLR in RSL, even though it has the same limitation as the MINDS dataset, where both lack a designated test set.

5.4 Models

We selected three attention-based video classification architectures ViViT [6], TimeSformer [10] and VideoMAE [95], to investigate the task of ISLR in LIBRAS. The selection was guided by three key considerations: (i) robustness in spatiotemporal modeling, (ii) availability of open-source implementations, and (iii) proven performance in related video classification benchmarks such as Kinetics-400 [51].

TimeSformer was included because it represents a straightforward yet effective adaptation of the Transformer paradigm to video data, splitting attention into spatial and temporal components while avoiding the need for 3D convolutions. ViViT similarly employs a pure-Transformer approach but offers multiple factorization strategies for combining space and time, making it a flexible candidate for extracting fine-grained gestural cues critical to SLR. Finally, VideoMAE leverages masked autoencoders to learn video representations with minimal supervision, a valuable property in data-constrained scenarios like ISLR for LIBRAS.

Table 5.2 presents a side-by-side comparison of these models, including their backbone configurations, attention variants, pre-training datasets, and approximate parameter counts. They each tokenize, in their own way, video frames into patches, apply multi-head self-attention mechanisms to capture spatiotemporal dependencies, and use a classification head for final predictions.

Table 5.2 – Comparison of Transformer-based models employed for video classification.

Model Name	Variant Employed	Attention Scheme	Total Parameters
TimeSformer	Divided Space-Time	Spatial, then Temporal (Divided)	∼121M
ViViT	Joint Spatiotemporal	Global Space-Time	∼86M
VideoMAE	Tube Masking (MAE)	Global Space-Time	∼86M

All models accept an input of T frames (sampled from the full video) at a resolution of 224×224 pixels. The number of frames T is set to 16 unless otherwise stated. Each run uses a batch size of 16, the AdamW optimizer [60] with initial learning rate of 1×10^{-5}, and a plateau scheduler is set, lowering the learning rate if the validation loss stabilizes within five consecutive epochs. Early stopping is triggered if validation accuracy does not improve for 30 consecutive epochs, with a maximum of 200 epochs to maintain computational constraints.

For our implementation, we relied on the PyTorch framework [73], which served as the backbone for our code. We also leveraged PyTorch Lightning [28] to train and evaluate all of our models, which streamlined our workflow and improved efficiency. To conduct our experiments, we utilized as hardware six NVIDIA A6000 graphics cards, an AMD Threadripper with 32 cores, and 110GB of RAM. In the sections that follow we summarize the logic behind each different architecture.

5.4.1 TimeSformer

TimeSformer extends the self-attention mechanism of Transformers to both spatial and temporal dimensions. Instead of relying solely on 3D convolutions or recurrent modules, it divides a video into patches along frame and spatial axes. Each patch becomes a token, allowing the self-attention mechanism to link spatial and temporal cues in a parallel manner, as opposed to step-by-step processing. Several variants of TimeSformer have been explored in prior work [10], including factorized space-time attention and full space-time attention. In all approaches, TimeSformer receives a sequence of tokens of shape ($T \times$ patches) in an embedding layer, employs multi-head attention over space and time, and concludes with a classification head that produces an output over the possible classes. For our work, we employ the divided space-time attention variant, which outperformed all other variants, according to the authors of [10].

5.4.2 ViViT

ViViT also utilizes a tokenization strategy for video frames, creating patches that represent spatial regions of each frame. In contrast to approaches that combine convolutions with attention, ViViT is a pure-Transformer architecture [6]. Several versions of the model explore ways to factorize or jointly apply attention across space and time, aiming to balance computational requirements and performance. The model version chosen in our work applies attention jointly across all spatio-temporal tokens from the input video. Despite differences in the variations of ViViT, all variations follows a common scheme of flattening video frames into patches, augmenting them with positional embeddings, and learning dependencies in a multi-head self-attention block. A classification layer then outputs a probability distribution over the sign classes.

5.4.3 VideoMAE

VideoMAE adapts masked autoencoding from the NLP domain to the video domain [95]. It masks a high proportion of patches (often 90%-95%) and learns to reconstruct the missing patches during training, assuming that videos contain redundant temporal information. Once trained in this fashion, VideoMAE can be fine-tuned for classification or other downstream tasks. For classification, the pre-processing step involves sampling and tokenizing frames. Next, a joint space-time embedding is created, and the model is tasked with reconstructing masked patches to learn meaningful spatiotemporal features. During fine-tuning for ISLR, the final layer classifies each input as one of the vocabulary signs. Since VideoMAE can perform well without external pre-training labels, it is suitable for settings with constrained or partially labeled data.

5.5 Experiments

Our experiments aimed to evaluate the performance of SOTA video classification models in the task of ISLR for LIBRAS at different settings. For each of our RQ, we devise a experiment.

5.5.1 Pre-training Experiment

Our first experiment aimed to determine whether models pre-trained on large-scale action recognition tasks could provide advantages when fine-tuned for ISLR in LIBRAS. Specifically, we sought to measure how effectively knowledge acquired from the Kinetics-400 [51] dataset, which encompasses thousands of videos capturing a broad range of human actions, could be adapted to the domain of SLR.

To this end, we evaluated each of the three models in two configurations: one pre-trained on Kinetics-400, and another initialized from scratch with randomly assigned weights. We then fine-tuned the models on the MINDS dataset, allowing them to learn the nuances of LIBRAS signs. Finally, we tested the resulting models on the MALTA-LIBRAS dataset to assess their ability to generalize to new signers and varied environments not encountered during training.

Results were measured using accuracy, precision, recall, and F1-score. By comparing the pre-trained models against those trained solely on the MINDS dataset, we aimed to determine if leveraging prior knowledge from general action recognition tasks would improve ISLR performance for LIBRAS, thus providing evidence of effective transfer learning and answering our RQ1.

5.5.2 Data Augmentation Experiment

Our second experiment focused on evaluating how different data augmentation techniques influence the model's performance throughout the training process. Other studies, such as the development of the MINDS [78] dataset, employ limited data augmentation techniques in LIBRAS, such as temporal displacement of frames, horizontal mirroring and zoom. Since this experiment does not aim to find the best model among the three tested, we chose to only perform data augmentation experiment on the less compute intensive model, VideoMAE [95].

We experimented with four augmentation techniques: (i) color jitter, which varies brightness, contrast, saturation and hue in order to emulate varying background scenarios common in video augmentation experiments; (ii) Mixup [103] as a way to blend different samples and introduce controlled noise; (iii) random rotation to account for minor shifts in sign execution, and (iv) random perspective to simulate different camera angles. As part of the data augmentation process, a random sampling strategy was also implemented. From the total set of 32 frames in each video, a subset of 16 frames was randomly selected for further processing.

Subsequently, each technique's intensity was incrementally adjusted from 0.05 to 0.5 in steps of 0.05. To ensure robustness and account for stochastic variations, training was repeated with ten different preset seeds for each intensity level. Averaging the performance metrics across these runs allowed us to mitigate the impact of random fluctuations.

After identifying the best intensity for each individual augmentation technique based on performance on the MALTA-LIBRAS dataset, we conducted a new training session on the MINDS dataset, using all augmentation techniques simultaneously at their respective optimal intensity levels. We then evaluated the model on MALTA-LIBRAS once again, comparing the results to those from experiments without data augmentation. Utilizing the MALTA-LIBRAS dataset to determine the optimal intensity levels for each augmentation technique deviates from standard protocol, as it effectively redefines MALTA-LIBRAS as a validation set rather than a test set. However, this approach was adopted due to resource constraints and the near-perfect validation metrics observed on all experiments. This comparative assessment, designed to answer our RQ2, enabled us to measure the cumulative benefits of the optimized augmentation pipeline and determine whether it further enhanced the model's generalization capabilities.

5.5.3 Transfer learning Experiment

Our final experiment was designed to assess the potential of transfer learning between different SLs. VideoMAE was the model selected for this experiment following the same rationale of the previous one.

Initially, we trained the VideoMAE model on the SLOVO dataset, the RSL dataset. After training, we fine-tuned the model on the MINDS dataset and then tested it on our MALTA-LIBRAS dataset to evaluate its ability to generalize to new signers and environments not seen during training.

Subsequently, we repeated the same procedure using the WLASL dataset. The model was trained on WLASL, fine-tuned on MINDS, and tested on MALTA-LIBRAS. Furthermore, we explored a combined pre-training approach by pre-training the model on both the SLOVO and WLASL datasets before fine-tuning on MINDS and testing on MALTA-LIBRAS. This approach allowed us to thoroughly evaluate the transfer learning capabilities among different SLs and to assess whether pre-training on multiple SLs enhances performance on LIBRAS recognition tasks.

The initial training on the SLOVO and WLASL datasets was performed without data augmentation to isolate the effects of augmentation during fine-tuning. During the fine-tuning step on the MINDS dataset, we used the optimal set of augmentations obtained from the previous experiment. This training was conducted with a single preset seed to reduce computational time and resources. Results of this experiment provides a thorough assessment of the model's classification capabilities and its ability to generalize across different SLs and datasets, enabling us to answer our RQ3.

The experiments in this chapter seek to help understand how each step, pre-training, data augmentation, and cross-dataset transfer learning, contributes to model performance in LIBRAS ISLR. The first experiment compares random initialization against weights learned from action recognition tasks, answering our RQ1. The second experiment searches for suitable augmentation approaches to address the scarcity and uniform conditions in the MINDS dataset, answering our RQ2. The third experiment tests whether SL data from other languages helps address domain gaps in LIBRAS ISLR, answering our RQ3. Collectively, these experiments give a comprehensive view of how attention-based video classifiers can be adapted to LIBRAS ISLR.

6. RESULTS

Dennis: Sugeriu seguir a estrutura: descrição do experimento -> link com a pergunta de pesquisa -> resultados; e na conclusão trazer de volta o link com a pergunta de pesquisa e a 'análise' final.

his chapter provides results from the experiments described in Chapter 5, focusing on the research questions (RQ1, RQ2, RQ3) described in Chapter 4. All experiments use the same evaluation metrics (accuracy, precision, recall, and F1-score) and the same data splits introduced earlier.

6.1 Pre-Training Experiment

RQ1 explores whether models pre-trained on the Kinetics-400 dataset [51] achieve better performance on LIBRAS ISLR than models initialized from scratch. Three architectures, TimeSformer [10], ViViT [6], and VideoMAE [95], were tested under two conditions: (i) one with pre-trained weights and (ii) another with randomly initialized weights (NPT). For the NPT models, He [38] initialization was employed to maintain stable variance by scaling weights according to the number of input connections in each layer, mitigating vanishing and exploding gradients situations.

Each model configuration (pre-trained and NPT) was fine-tuned on the MINDS dataset, following a 75-25% training-validation split for each sign. The MALTA-LIBRAS dataset served as a separate test set to evaluate model generalization. No data augmentation was applied, and each experiment was repeated with ten preset seeds, averaging the resulting metrics.

Table 6.1 displays the average accuracy, precision, recall, and F1-score for each model under both pre-trained and NPT settings, revealing that models pre-trained on Kinetics-400 achieve notably higher validation metrics compared to those trained solely on MINDS. From Table 6.1 and Table 6.2, it is evident that the VideoMAE model outperforms all others in both MINDS validation and MALTA-LIBRAS testing. We attribute this advantage to its masked auto-encoding method, where random portions of the input are concealed and the model is trained to reconstruct these hidden segments, effectively learning more robust representations.

Though these models attain high validation scores, their metrics on MALTA-LIBRAS remains lower. This result indicates that the models may be learning patterns in the training data that do not generalize well to the unseen test set, revealing an overfitting scenario.

An additional evaluation examined performance metrics across each data source in MALTA-LIBRAS, as summarized in Table 6.2. Across these different conditions, the Video-

MAE model consistently surpassed ViViT and TimeSformer, achieving higher accuracy and showing effectiveness in managing sign variety and other differences found in MALTA-LIBRAS. Moreover, the metrics appeared stable across these sources, indicating no obvious biases that might complicate model performance on any single subset.

Even though models trained exclusively on MINDS struggled to generalize, pre-training on the Kinetics-400 dataset played a major role in bridging this gap. A comparison of validation and test outcomes in Table 6.1 reveals that NPT models achieved accuracy equivalent to random guessing over the 20 classes of the test set, whereas pre-trained counterparts achieved accuracy levels above random chance.

This finding addresses RQ1 by showing that pre-training on action recognition tasks can indeed benefit LIBRAS ISLR. Despite MALTA-LIBRAS having varied signers and recording conditions, knowledge from general human motion appears valuable for feature extraction. All models improved top-1 accuracy and F1-score when leveraging Kinetics-400 pre-training, suggesting that large-scale motion representations from general action data transfer effectively to sign recognition, especially when training is limited by data constraints such as those in MINDS.

6.2 Data Augmentation Experiment

RQ2 focused on the effect of different data augmentation approaches on LIBRAS ISLR. Due to computational constraints and previous experiment outcomes, we use the VideoMAE model for subsequent experiments, as it is less resource-intensive than TimeSformer or ViViT.

In contrast to the original MINDS setup, which made limited use of data augmentation, our study employed a more extensive range of augmentation methods and intensities. Specifically, we tested Color Jitter (changing brightness, contrast, saturation, and hue), Mixup [103] (mixing training samples with samples from other classes to introduce controlled noise), Random Rotation, and Random Perspective for simulating camera angle variations. Each method was examined at intensities ranging from 0.05 to 0.50 in incre-

Table 6.1 – Validation and test results of pretrained and non-pretrained (NPT) models.

Model	MINDS Validation				MALTA-LIBRAS Test			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
VideoMAE	**.994**	**.995**	**.994**	**.994**	**.290**	.256	**.258**	**.227**
TimeSformer	.980	.983	.980	.980	.254	**.257**	.224	.207
ViVit	.984	.986	.984	.984	.181	.177	.155	.134
VideoMAE NPT	.767	**.791**	.767	.766	.048	.020	.043	.022
TimeSformer NPT	**.774**	.783	**.774**	**.769**	.051	**.040**	**.059**	.028
ViVit NPT	.552	.563	.552	.549	**.055**	.036	.054	**.034**

Table 6.2 – Test metrics for each data source within MALTA-LIBRAS of all pretrained and non-pretrained (NPT) models.

Source		VideoMAE	VideoMAE NPT	TimeSformer	TimeSformer NPT	ViVit	ViVit NPT
Spread The Sign	Accuracy	.293	.056	**.312**	.031	.206	.081
	Precision	.179	.022	**.236**	.017	.159	.024
	Recall	.272	.046	**.289**	.024	.189	.064
	F1-score	.207	.026	**.252**	.019	.169	.030
INES Dictionary 2.0	Accuracy	**.234**	.061	.219	.080	.219	.050
	Precision	**.177**	.007	.158	.028	.146	.023
	Recall	**.248**	.044	.222	.078	.226	.046
	F1-score	**.191**	.011	.167	.030	.158	.026
INES Dictionary 3.0	Accuracy	**.300**	.050	.287	.045	.137	.050
	Precision	**.262**	.012	.242	.012	.118	.012
	Recall	**.280**	.055	.270	.055	.139	.044
	F1-score	**.244**	.014	.236	.017	.111	.016
V-LIBRASIL	Accuracy	**.314**	.040	.214	.044	.148	.051
	Precision	**.150**	.006	.122	.014	.073	.032
	Recall	**.214**	.036	.166	.035	.112	.037
	F1-score	**.160**	.010	.114	.018	.064	.027
Signbank	Accuracy	**.278**	.057	.257	.063	.215	.052
	Precision	**.233**	.005	.202	.019	.137	.022
	Recall	**.248**	.061	.223	.065	.184	.054
	F1-score	**.229**	.009	.205	.024	.146	.026
LIBRAS Corpus	Accuracy	**.331**	.026	.263	.036	.178	.057
	Precision	**.266**	.001	.202	.009	.119	.018
	Recall	**.306**	.027	.236	.038	.153	.054
	F1-score	**.276**	.002	.205	.012	.119	.025

ments of 0.05, repeated with ten different seeds to address random fluctuations. Models were trained on MINDS (75% training, 25% validation) and tested on MALTA-LIBRAS. The highest accuracy from each augmentation was then merged into a single pipeline.

Table 6.3 shows the intensities, highlighting the best mean accuracy and standard deviation for each technique. A key observation emerged with Mixup: moderate intensities (approximately 0.20–0.30) tended to yield higher mean accuracy than lower or higher intensities. Minimal intensities may help generalization, while high intensities often distort signs to the extent that classification becomes challenging. This pattern aligns with the idea that overly strong mixing can produce samples that the model struggles to recognize.

For Color Jitter, all parameters (brightness, contrast, saturation, and hue) were varied uniformly, making it difficult to assess the contribution of each aspect individually. Nonetheless, higher Color Jitter intensities improved generalization. Intensities of 0.40 and 0.50 both achieved an accuracy of 0.310, though the 0.40 setting displayed lower variance, suggesting it is preferable overall.

Turning to Random Perspective, higher intensities similarly aided model generalization, with 0.50 yielding 0.293 in accuracy. All techniques boosted performance except for Random Rotation. Across all rotation intensities, 0.35 recorded 0.274 in accuracy, which is lower than the baseline without augmentation (see Table 6.1). Sign gestures rely significantly on hand angles, so strong rotation may alter sign shape or orientation, leading to misclassification.

Based on these findings, the following intensities were chosen for the final data augmentation pipeline: 0.40 for Color Jitter, 0.25 for Mixup, 0.35 for Random Rotation, and 0.50 for Random Perspective. Using these intensities across 10 preset seeds, a new set of runs was performed on the VideoMAE model, both with (PT) and without pre-training (NPT).

Table 6.4 shows that data augmentation by itself does not surpass the advantages of pre-training. In addition, the NPT VideoMAE still struggles despite the new pipeline, while the PT model improves from an accuracy of 0.29 to 0.336 when using these techniques.

By analyzing these results, we can address RQ2 and conclude that introducing variation during training helps models manage diverse backgrounds, lighting conditions, and signer characteristics. However, these improvements did not eliminate the performance gap between MINDS and MALTA-LIBRAS, suggesting that additional variety in training data or alternative approaches, including more specialized video augmentations, may be necessary for robust generalization.

All explored augmentation methods enhanced model accuracy, but each had an optimal range, after which further transformations were no longer helpful. Moderate transformations appear to generate more varied training instances without obscuring key sign features. These outcomes indicate that careful data augmentation can be beneficial for SLR, particularly in resource-limited contexts.

Table 6.3 – Mean and standard deviation of MALTA-LIBRAS test accuracy for the VideoMAE model, pretrained on Kinetics-400 and fine-tuned on MINDS, on various intensities for each data augmentation technique.

Intensity	*Color Jitter*	*Mixup*	Random Rotation	Random Perspective
0.05	.290 ± .016	.286 ± .026	.271 ± .027	.288 ± .025
0.10	.288 ± .018	.291 ± .035	.271 ± .027	.279 ± .025
0.15	.301 ± .026	.279 ± .018	.271 ± .027	.277 ± .022
0.20	.295 ± .021	.289 ± .038	.271 ± .027	.288 ± .036
0.25	.305 ± .025	**.298 ± .027**	.271 ± .027	.283 ± .021
0.30	.305 ± .014	.293 ± .022	.258 ± .029	.286 ± .019
0.35	.302 ± .024	.254 ± .042	**.274 ± .028**	.276 ± .029
0.40	**.310 ± .014**	.287 ± .047	.266 ± .039	.286 ± .015
0.45	.296 ± .028	.277 ± .031	.274 ± .031	.283 ± .029
0.50	.310 ± .029	.273 ± .042	.268 ± .033	**.293 ± .021**

Table 6.4 – Test accuracy on MALTA-LIBRAS with and without the augmentation pipeline.

Model	Without Augmentation		With Augmentation	
	Acc	Top-5 Acc	Acc	Top-5 Acc
VideoMAE PT	.290 ± .021	.541 ± .027	.336 ± .013	.523 ± .028
VideoMAE NPT	.048 ± .012	.260 ± .021	.074 ± .016	.269 ± .017

6.3 Transfer Learning Experiment

The final experiment investigated the potential for cross-dataset or cross-lingual knowledge transfer to support LIBRAS ISLR. A VideoMAE model, pre-trained on Kinetics-400, was further pre-trained (PT) on the SLOVO dataset following the author's split but merging the validation and test sets into a single validation set, then fine-tuned (FN) on MINDS. Separately, the model was PT on the WLASL-1000 subset, maintaining the author's original splits, then FT on MINDS. Finally, another scenario involved PT on both SLOVO and WLASL-1000 before FT on MINDS.

No augmentation was used during the SLOVO or WLASL PT phases, though the RQ2 pipeline was applied during MINDS fine-tuning. Only one seed was used in each experiment to reduce training times, creating a common validation approach across both datasets and facilitating direct comparisons.

This step addressed RQ3 by assessing whether sign-based knowledge could improve performance in LIBRAS ISLR. Table 6.5 shows mean accuracy in MALTA-LIBRAS for each PT scenario, including a baseline with no cross-dataset PT.

Table 6.5 – Results for the VideoMAE pre-trained (PT) on Russian and American Sign Languages then fine-tuned (FT) on MINDS.

Model	MINDS Validation				MALTA-LIBRAS Test			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
SLOVO PT + MINDS FT	1	1	1	1	**.317**	.242	**.274**	.226
WLASL PT + MINDS FT	.996	.996	.996	.996	.290	**.263**	.263	**.238**
SLOVO PT + WLASL PT + MINDS FT	.996	.996	.996	.996	.290	**.263**	.263	**.238**
MINDS FT	.994	.995	.994	.994	.290	.256	.258	.227

Contrary to initial expectations, pre-training on SLOVO, WLASL, or both offered limited gains compared to a baseline PT model. In Table 6.5, only PT with SLOVO provided a slight boost in test accuracy, even though validation achieved perfect performance. Cross-lingual SL data yielded smaller benefits than pre-training on action recognition, indicating that any spatiotemporal overlap across SLs might be overshadowed by the broader motion patterns learned from general action datasets.

Variations in linguistic structure, sign formation, and visual appearance likely reduced the potential for transferring features across different SLs. These findings point to the complexity of cross-lingual adaptation and the need for approaches that directly address language or domain-specific differences.

7. CONCLUSION

This dissertation tackled thetask of improving ISLR in LIBRAS, an endeavor vital for breaking down communication barriers and fostering broader accessibility for Deaf communities. Through focused explorations in three different approaches, leveraging action recognition pre-training (RQ1), broadening data augmentation strategies (RQ2), and pursuing cross-dataset transfer learning across diverse Sign Languages (RQ3), this work aimed to enhance model robustness and generalizability. Experiments were conducted on the MINDS dataset for training and validation, while a subset of our novel MALTA-LIBRAS dataset served as a real-world test set to evaluate metrics on a more varied set of videos and signers. Three attention-based video classification architectures, TimeSformer, ViViT, and VideoMAE, were employed to capture the subtle spatial and temporal cues that define sign articulation. By integrating these techniques and models, this study demonstrates a promising path toward more accurate, inclusive, and scalable LIBRAS recognition systems. We present next our RQs and their respective findings:

- RQ1: How does pre-training in action recognition tasks affect LIBRAS ISLR?

Models pre-trained on Kinetics-400 performed better than those trained solely on MINDS. The difference was clear for the VideoMAE architecture, which achieved higher metrics across multiple MALTA-LIBRAS dictionaries, consistently outperforming other models. These findings suggest that large-scale human action datasets capture motion features that can transfer to LIBRAS signs. While MINDS is narrow in scope, it may still benefit from these motion priors, improving convergence and mitigating overfitting.

- RQ2: How different data augmentation techniques impacts LIBRAS ISLR?

Random perspective transformations, color jitter, and Mixup helped improve generalization from MINDS to MALTA-LIBRAS. Although the gains did not fully address the overfitting issue, the observed rise in accuracy and F1-score shows the benefits of a structured data augmentation pipeline tailored for SLR. Moderate intensities in most techniques provided better results, indicating that excessive transformations can obscure important sign features.

- RQ3: How cross-dataset transfer learning, between ASL and RLS, affect LIBRAS ISLR?

Pre-training with the RSL-based SLOVO dataset yielded improvements in accuracy and recall, while the ASL-based pre-training and the combination of both SLOVO and ASL benefited precision and F1-score, compared to a baseline without cross-lingual data. These gains, however, were smaller than those from Kinetics-400. This finding implies that while certain features may be shared across different SLs, broader motion representations learned from general action datasets may carry more influence.

7.1 Implications

Training on a small, controlled dataset such as MINDS can lead to overfitting, though performance can be improved by leveraging pre-training on large-scale action datasets or by employing carefully chosen data augmentation strategies. Cross-lingual pre-training offers less benefit than expected, possibly reflecting limited overlap across different SLs.

Introducing additional data or domain-specific augmentations can help address the lack of variety in existing LIBRAS datasets, producing tangible gains. The MALTA-LIBRAS dataset emphasizes the importance of evaluation sets that differ from controlled conditions, providing a more realistic measure of model performance. Large-scale human action pre-training remains valuable for capturing motion cues relevant to SLs, yet only small improvements arose from cross-lingual transfer, indicating a need for deeper collaborations or more consistent labeling protocols to fully utilize multi-language sign data.

7.2 Limitations

7.3 Limitations

The methods described in this work face challenges related to data diversity, model capacity, and computational resources, limiting both the conclusions regarding SLR in LIBRAS and their applicability to broader SL contexts. In addition, the chosen evaluation strategy relies on a fixed holdout set rather than more robust statistical methods such as k-fold cross-validation. While the holdout approach allows for a simpler workflow, it may leave open questions about model variance and the representativeness of the chosen splits, especially when datasets are small or unbalanced. Future studies could explore more comprehensive sampling strategies to gain deeper insights into model performance and generalization.

7.3.1 Datasets

MINDS is smaller than resources such as WLASL and has limited environmental variety compared to datasets like SLOVO, leading to pronounced overfitting. MALTA-LIBRAS broadens signer and recording conditions but remains constrained in both total videos and samples per gloss. Both datasets concentrate on only 20 signs, covering a minimal fraction of LIBRAS vocabulary and possibly overlooking the broader complexity found in day-to-day signing.

7.3.2 Models

All tested architectures use attention-based patch embeddings for spatiotemporal modeling, and while these approaches worked well, integrating skeleton or pose-based features in a hybrid strategy may offer further gains. The large number of parameters can complicate training under limited data or hardware constraints.

Additionally, the same hyperparameters were applied across different experiments, leaving open the possibility that certain configurations might be less effective for specific data splits or augmentation procedures. Large networks can overfit when data is scarce, and they may benefit from stronger regularization or broader augmentation techniques.

7.3.3 Hardware

All experiments were conducted on a computational cluster with two NVIDIA A6000 GPUs using fixed parallelization strategies to ensure reproducibility. For RQ1 three transformer architectures (TimeSformer, ViViT, VideoMAE) were tested under two training settings (pre-trained on Kinetics-400 vs. He-initialized weights) across 10 preset seeds. This resulted in a total of 60 runs, 3 models, 2 pre-training settings across 10 preset seeds, requiring 480 GPU hours (±20 days) at approximately 8 hours per run.

RQ2 investigated data augmentation techniques, testing four strategies (color jitter, random rotation, random perspective, Mixup) at 10 intensity levels (0.05–0.50/) with 10 seeds per technique. After identifying optimal intensities, a combined pipeline was evaluated with 10 additional seeds, totaling 410 runs. This experiment used 3,280 GPU hours (±137 days).

For RQ3, cross-lingual transfer learning involved three steps:

1. SLOVO pretraining: VideoMAE trained on SLOVO with 10 seeds, followed by fine-tuning the best seed on MINDS.

2. WLASL pretraining: VideoMAE trained on WLASL-1k with 10 seeds, then fine-tuning the best seed on MINDS.

3. Combined pretraining: Sequential training on SLOVO (1 seed), then on WLASL-1k (1 seed), then fine-tuning on MINDS (1 seed).

This culminated in 25 runs for RQ3, requiring approximately 200 GPU hours (±8.3 days). Collectively, the study executed 495 runs with over 3,960 GPU hours (±165 days), highlighting the computational intensity of transformer-based video modeling in low-resource SLR.

7.4 Future Work

Several directions could broaden the scope of LIBRAS ISLR research. Emphasis may be placed on enlarging dataset resources, combining models that handle signing nuances, and refining transfer learning.

One direction involves collecting additional LIBRAS videos on a scale similar to other SL datasets, which would also support a shift from ISLR to CSLR and help limit overfitting. Involvement from Deaf communities or educational institutions may enhance the authenticity of collected data.

Another direction centers on methods that rely on 2D or 3D skeletal inputs, keypoint detection, or depth estimation. These steps can reduce the model's dependence on color or background settings. Incorporating pose data into an attention-based structure might address lexical and grammatical elements of LIBRAS. Finally, techniques such as domain adaptation, adversarial training, and multi-lingual SL embeddings may leverage cross-lingual data more effectively than direct fine-tuning.

7.4.1 Published Work and Fundings

This work was submitted to the 2025 International Joint Conference on Neural Networks. Future publications will likely focus on extended LIBRAS corpora and domain adaptation experiments to expand real-world usability of sign language systems. Funding for our work included the 2022 Google Award for Inclusion Research (AIR), with which we were able to obtain the GPU's used here. This study was also financed in part by the Coordination for the Improvement of Higher Education Personnel – Brazil (CAPES) – Finance Code 001 and Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS, grant nr. 22/2551-0000390-7).

This dissertation emphasizes both the opportunities and the difficulties in LIBRAS ISLR. Data scarcity remains a significant limitation, yet pre-training on general human motion datasets and carefully chosen augmentations can alleviate overfitting and contribute to moderate gains. Cross-lingual sign data yields only limited improvements, indicating the need for additional data collection efforts, cross-lingual alignment approaches, and specialized architectures for SLs. The MALTA-LIBRAS test set, along with the experimental findings, forms a starting point for further LIBRAS investigations and may guide expanded research in academic and practical domains.

REFERENCES

[1] Abdu, S. A.; Yousef, A. H.; Salem, A. “Multimodal video sentiment analysis using deep learning approaches, a survey”, *Information Fusion*, vol. 76, December 2021, pp. 204–226.

[2] Adaloglou, N.; Chatzis, T.; Papastratis, I.; Stergioulas, A.; Papadopoulos, G. T.; Zacharopoulou, V.; Xydopoulos, G. J.; Atzakas, K.; Papazachariou, D.; Daras, P. “A comprehensive study on deep learning-based methods for sign language recognition”, *IEEE Transactions on Multimedia*, vol. 24, April 2022, pp. 1750–1762.

[3] Agha, R. A. A. R.; Sefer, M. N.; Fattah, P. “A comprehensive study on sign languages recognition systems using (svm, knn, cnn and ann)”. In: International Conference on Data Science, E-Learning and Information Systems, 2018, pp. 1–6.

[4] Agência IBGE Notícias. “Pns 2019: país tem 17,3 milhões de pessoas com algum tipo de deficiência”. Accessed in: November 2023, Source: https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/31445-pns-2019-pais-tem-17-3-milhoes-de-pessoas-com-algum-tipo-de-deficiencia, 2021.

[5] Al-Qurishi, M.; Khalid, T.; Souissi, R. “Deep learning for sign language recognition: Current techniques, benchmarks, and open issues”, *IEEE Access*, vol. 9, September 2021, pp. 126917–126951.

[6] Arnab, A.; Dehghani, M.; Heigold, G.; Sun, C.; Lucic, M.; Schmid, C. “Vivit: A video vision transformer”. In: IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 6816–6826.

[7] Bahdanau, D.; Cho, K.; Bengio, Y. “Neural machine translation by jointly learning to align and translate”, *ArXiv*, vol. 1409, Setember 2014, pp. 15.

[8] Bahia, N. K.; Rani, R. “Multi-level taxonomy review for sign language recognition: Emphasis on indian sign language”, *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22–1, June 2023, pp. 1–39.

[9] Bengio, Y. “Learning Deep Architectures for AI (Foundations and Trends in Machine Learning)”. Now Publishers, 2009, 144p.

[10] Bertasius, G.; Wang, H.; Torresani, L. “Is space-time attention all you need for video understanding?” In: 38th International Conference on Machine Learning, 2021, pp. 813–824.

[11] Bishop, C. M. “Pattern Recognition and Machine Learning (Information Science and Statistics)”. Berlin, Heidelberg: Springer-Verlag, 2006, 738p.

[12] Boesch, G. “Vision transformers (vit) in image recognition - 2024 guide”. Accessed in: June 2023, Source: https://viso.ai/deep-learning/vision-transformer-vit/, 2024.

[13] Bragg, D.; Koller, O.; Bellard, M.; Berke, L.; Boudreault, P.; Braffort, A.; Caselli, N.; Huenerfauth, M.; Kacorri, H.; Verhoef, T.; Vogler, C.; Ringel Morris, M. “Sign language recognition, generation, and translation: An interdisciplinary perspective”. In: 21st International ACM SIGACCESS Conference on Computers and Accessibility, 2019, pp. 16–31.

[14] Brasil. “Lei nº 10.436, de 24 de abril de 2002”. Accessed in: Jan 2023, Source: https://www.planalto.gov.br/ccivil_03/leis/2002/l10436.htm, January 2023.

[15] Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al.. “Language models are few-shot learners”, *Advances in neural information processing systems*, vol. 33, 2020, pp. 1877–1901.

[16] Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.-E.; Sheikh, Y. “Openpose: Realtime multi-person 2d pose estimation using part affinity fields”, *IEEE transactions on pattern analysis and machine intelligence*, vol. 43–1, 2019, pp. 172–186.

[17] Caselli, N. K.; Sehyr, Z. S.; Cohen-Goldberg, A. M.; Emmorey, K. “Asl-lex: A lexical database of american sign language”, *Behavior Research Methods*, vol. 49–2, May 2016, pp. 784–801.

[18] Chapelle, O.; Scholkopf, B.; Zien, Eds., A. “Semi-supervised learning”, *IEEE Transactions on Neural Networks*, vol. 20–3, 2009, pp. 542–542.

[19] Chaveiro, N.; Porto, C. C.; Barbosa, M. A. “The relation between deaf patients and the doctor”, *Brazilian journal of otorhinolaryngology*, vol. 75–1, February 2009, pp. 147–150.

[20] Cihan Camgoz, N.; Koller, O.; Hadfield, S.; Bowden, R. “Sign language transformers: Joint end-to-end sign language recognition and translation”. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10020–10030.

[21] Cooper, H.; Holt, B.; Bowden, R. “Sign Language Recognition”. London: Springer London, 2011, chap. Applications, pp. 539–562.

[22] Courant, R.; Edberg, M.; Dufour, N.; Kalogeiton, V. “Transformers and Visual Transformers”. New York, NY: Springer US, 2023, chap. Machine Learning Fundamentals, pp. 193–229.

[23] Das, S.; Biswas, S. k.; Chakraborty, M.; Purkayastha, B. “A review on sign language recognition (slr) system: Ml and dl for slr”. In: IEEE International Conference on Intelligent Systems, Smart and Green Technologies (ICISSGT), 2021, pp. 177–182.

[24] De Coster, M.; Van Herreweghe, M.; Dambre, J. “Isolated sign recognition from rgb video using pose flow and self-attention”. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, pp. 3436–3445.

[25] de Quadros, R. M. “Phrase structure of brazilian sign language”. In: *Cross-Linguistic Perspectives in Sign Language Research: Selected Papers from TISLR*, Baker, A. E.; van den Bogaerde, B.; Crasborn, O. (Editors), Hamburg: Signum Press, 2003, pp. 141–162.

[26] Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. “BERT: Pre-training of deep bidirectional transformers for language understanding”. In: 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Burstein, J.; Doran, C.; Solorio, T. (Editors), 2019, pp. 4171–4186.

[27] Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; Houlsby, N. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: International Conference on Learning Representations, 2021.

[28] Falcon, William and The PyTorch Lightning team. “Pytorch lightning”. Accessed in: Jan 2023, Source: https://github.com/Lightning-AI/lightning, March 2019.

[29] Feng, Z.; Xu, S.; Zhang, X.; Jin, L.; Ye, Z.; Yang, W. “Real-time fingertip tracking and detection using kinect depth sensor for a new writing-in-the air system”. In: International Conference on Internet Multimedia Computing and Service, 2012, pp. 70–74.

[30] Forster, J.; Schmidt, C.; Hoyoux, T.; Koller, O.; Zelle, U.; Piater, J.; Ney, H. “Rwth-phoenix-weather: A large vocabulary sign language recognition and translation corpus”. In: Eighth International Conference on Language Resources and Evaluation (LREC’12), Calzolari, N.; Choukri, K.; Declerck, T.; Dogan, M. U.; Maegaard, B.; Mariani, J.; Moreno, A.; Odijk, J.; Piperidis, S. (Editors), 2012, pp. 3785–3789.

[31] Forster, J.; Schmidt, C.; Koller, O.; Bellgardt, M.; Ney, H. “Extensions of the sign language recognition and translation corpus RWTH-PHOENIX-weather”. In: Ninth International Conference on Language Resources and Evaluation (LREC’14), Calzolari, N.; Choukri, K.; Declerck, T.; Loftsson, H.; Maegaard, B.; Mariani, J.; Moreno, A.; Odijk, J.; Piperidis, S. (Editors), 2014, pp. 1911–1916.

[32] Fundação de Articulação e Desenvolvimento de Políticas Públicas para Pessoas Portadoras de Deficiência e de Altas Habilidades no Rio Grande do Sul – FADERS. “Oficialização da libras como língua oficial do brasil completa 21 anos”. Accessed in: November 2023, Source: http://bit.ly/3FYPCGt, 2023.

[33] Gediel, A. L. B.; Mourão, Victor Luiz Alves, e. a. “Dicionário online bilíngue libras/português”, Technical Report, Universidade Federal de Viçosa - UFV, Brasília, Brasil, 2017, 29p.

[34] Gheisari, M.; Wang, G.; Bhuiyan, M. Z. A. “A survey on deep learning in big data”. In: IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), 2017, pp. 173–180.

[35] Glasser, A.; Mande, V.; Huenerfauth, M. “Accessibility for deaf and hard of hearing users: Sign language conversational user interfaces”. In: Conference on Conversational User Interfaces (CUI ’20), 2020, pp. 1–3.

[36] Goodfellow, I.; Bengio, Y.; Courville, A. “Deep Learning”. Cambridge, MA: MIT Press, 2016, 800p.

[37] Gui, J.; Chen, T.; Zhang, J.; Cao, Q.; Sun, Z.; Luo, H.; Tao, D. “A survey on self-supervised learning: Algorithms, applications, and future trends”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46–12, June 2024, pp. 9052–9071.

[38] He, K.; Zhang, X.; Ren, S.; Sun, J. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1026–1034.

[39] He, S. “Research of a sign language translation system based on deep learning”. In: International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), 2019, pp. 392–396.

[40] Hendrycks, D.; Gimpel, K. “Gaussian error linear units (gelus)”, *arXiv*, 2016.

[41] Hill, J. “Do deaf communities actually want sign language gloves?”, *Nature Electronics*, vol. 3, July 2020, pp. 512–513.

[42] Hilzensauer, M.; Krammer, K. “A multilingual dictionary for sign languages: "spreadthesign"”. In: International Conference of Education, Research and Innovation, 2015, pp. 7826–7834.

[43] Hochreiter, S.; Schmidhuber, J. “Long short-term memory”, *Neural Comput.*, vol. 9–8, November 1997, pp. 1735–1780.

[44] Huang, J.; Zhou, W.; Zhang, Q.; Li, H.; Li, W. “Video-based sign language recognition without temporal segmentation”, *AAAI Conference on Artificial Intelligence*, vol. 32–1, April 2018.

[45] Jhuo, I.-H.; Lee, D. “Video event detection via multi-modality deep learning”. In: 22nd International Conference on Pattern Recognition, 2014, pp. 666–671.

[46] Jiménez-Salas, J.; Chacón-Rivas, M. “A systematic mapping of computer vision-based sign language recognition”. In: International Conference on Inclusive Technologies and Education (CONTIE), 2022, pp. 1–11.

[47] Kahlon, N. K.; Singh, W. “Machine translation from text to sign language: a systematic review”, *Universal Access in the Information Society*, vol. 22–1, July 2021, pp. 1–35.

[48] Kamath, U.; Graham, K.; Emara, W. “Transformers for Machine Learning: A Deep Dive”. CRC Press, 2022, 257p.

[49] Kapitanov, A.; Karina, K.; Nagaev, A.; Elizaveta, P. “Slovo: Russian sign language dataset”. In: International Conference on Computer Vision Systems, 2023, pp. 63–73.

[50] Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. “Large-scale video classification with convolutional neural networks”. In: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.

[51] Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al.. “The kinetics human action video dataset”, *arXiv*, 2017.

[52] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. “Imagenet classification with deep convolutional neural networks”. In: Advances in Neural Information Processing Systems, 2012, pp. 1–9.

[53] Li, D.; Opazo, C.; Yu, X.; Li, H. “Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison”. In: IEEE Winter Conference on Applications of Computer Vision (WACV), 2020, pp. 1448–1458.

[54] Li, Y.; Miao, Q.; Tian, K.; Fan, Y.; Xu, X.; Ma, Z.; Song, J. “Large-scale gesture recognition with a fusion of rgb-d data based on optical flow and the c3d model”, *Pattern recognition letters*, vol. 119, December 2019, pp. 187–194.

[55] Li, Y.; Wu, C.-Y.; Fan, H.; Mangalam, K.; Xiong, B.; Malik, J.; Feichtenhofer, C. “Mvitv2: Improved multiscale vision transformers for classification and detection”.

In: IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 4804–4814.

[56] Lira, G. d. A.; Souza, T. A. F. d. “Libras versão 2.0”. Acessed in: January 2024, Source: http://www.acessibilidadebrasil.org.br/libras/, 2014.

[57] Lira, G. d. A.; Souza, T. A. F. d. “Libras versão 3.0”. Accessed in: January 2024, Source: http://www.acessibilidadebrasil.org.br/libras_3/, 2014.

[58] Liu, K.; Hou, Y.; Guo, Z.; Yin, W.; Ren, Y. “Visual context learning based on cross-modal knowledge for continuous sign language recognition”, *The Visual Computer*, October 2024, pp. 1–15.

[59] Liu, Z.; Ning, J.; Cao, Y.; Wei, Y.; Zhang, Z.; Lin, S.; Hu, H. “Video swin transformer”. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 3202–3211.

[60] Loshchilov, I.; Hutter, F. “Decoupled weight decay regularization”. In: International Conference on Learning Representations, 2019, pp. 18.

[61] Lugaresi, C.; Tang, J.; Nash, H.; McClanahan, C.; Uboweja, E.; Hays, M.; Zhang, F.; Chang, C.-L.; Yong, M.; Lee, J.; Chang, W.-T.; Hua, W.; Georg, M.; Grundmann, M. “Mediapipe: A framework for perceiving and processing reality”. In: IEEE Computer Vision and Pattern Recognition (CVPR) Workshops, 2019.

[62] Luong, T.; Pham, H.; Manning, C. D. “Effective approaches to attention-based neural machine translation”. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, Marquez, L.; Callison-Burch, C.; Su, J. (Editors), 2015, pp. 1412–1421.

[63] Maćkiewicz, A.; Ratajczak, W. “Principal components analysis (pca)”, *Computers & Geosciences*, vol. 19–3, September 1993, pp. 303–342.

[64] Ministério da Saúde, Secretaria de Atenção à Saúde, Departamento de Ações Programáticas Estratégicas. “A Pessoa com Deficiência e o Sistema Único de Saúde”. Editora do Ministério da Saúde, 2007, 2 ed., 16p.

[65] Mitchell, T. M. “Machine learning”. McGraw-hill, 1997, 432p.

[66] Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; Hassabis, D. “Human-level control through deep reinforcement learning”, *Nature*, vol. 518–7540, February 2015, pp. 529–533.

[67] Molchanov, P.; Gupta, S.; Kim, K.; Kautz, J. “Hand gesture recognition with 3d convolutional neural networks”. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2015, pp. 1–7.

[68] Munea, T. L.; Jembre, Y. Z.; Weldegebriel, H. T.; Chen, L.; Huang, C.; Yang, C. “The progress of human pose estimation: A survey and taxonomy of models applied in 2d human pose estimation”, *IEEE Access*, vol. 8, July 2020, pp. 133330–133348.

[69] Murphy, K. P. “Probabilistic Machine Learning: An introduction”. MIT Press, 2022, 864p.

[70] Na, S.; Xumin, L.; Yong, G. “Research on k-means clustering algorithm: An improved k-means clustering algorithm”. In: Third International Symposium on Intelligent Information Technology and Security Informatics, 2010, pp. 63–67.

[71] Natarajan, B.; Rajalakshmi, E.; Elakkiya, R.; Kotecha, K.; Abraham, A.; Gabralla, L. A.; Subramaniyaswamy, V. “Development of an end-to-end deep learning framework for sign language recognition, translation, and video generation”, *IEEE Access*, vol. 10, September 2022, pp. 104358–104374.

[72] Parelli, M.; Papadimitriou, K.; Potamianos, G.; Pavlakos, G.; Maragos, P. “Spatio-temporal graph convolutional networks for continuous sign language recognition”. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 8457–8461.

[73] Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; Chintala, S. “PyTorch: an imperative style, high-performance deep learning library”. Red Hook, NY, USA: Curran Associates Inc., 2019.

[74] Pizzio, A. L.; Stumpf, M. R.; Lucinda, J. O.; Quadros, R. M. d.; Crasborn, O. “Signbank da libras”, *Fórum Linguístico*, vol. 17–4, December 2020, pp. 5475–5487.

[75] Quadros, R. M. “Brazilian sign language documentation”. In: Ibero-American Seminar on Linguistic Diversity, 2014, pp. 157–174.

[76] Radhakrishnan, S.; Mohan, N. C.; Varma, M.; Varma, J.; Pai, S. N. “Cross transferring activity recognition to word level sign language detection”. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2022, pp. 2445–2452.

[77] Redmon, J.; Farhadi, A. “Yolov3: An incremental improvement”, *arXiv*, 2018.

[78] Rezende, T. M.; Almeida, S. G. M.; Guimarães, F. G. “Development and validation of a brazilian sign language database for human gesture recognition”, *Neural Computing and Applications*, vol. 33–16, August 2021, pp. 10449–10467.

[79] Rodrigues, A. J. “V-librasil:uma base de dados com sinais na língua brasileira de sinais (libras)”, Master’s Thesis, UFPE, 2021, 162p.

[80] Russell, S.; Norvig, P.; Davis, E. “Artificial Intelligence: A Modern Approach”. Prentice Hall, 2010, 1132p.

[81] Sandler, W.; Lillo-Martin, D. “Sign Language and Linguistic Universals”. Cambridge University Press, 2006, 547p.

[82] Sarhan, N.; Frintrop, S. “Transfer learning for videos: From action recognition to sign language recognition”. In: IEEE International Conference on Image Processing (ICIP), 2020, pp. 1811–1815.

[83] Shi, B.; Rio, A. M. D.; Keane, J.; Brentari, D.; Shakhnarovich, G.; Livescu, K. “Fingerspelling recognition in the wild with iterative visual attention”. In: IEEE/CVF International Conference on Computer Vision, 2019, pp. 5400–5409.

[84] Shinde, P. P.; Shah, S. “A review of machine learning and deep learning applications”. In: Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, pp. 1–6.

[85] Simonyan, K.; Zisserman, A. “Two-stream convolutional networks for action recognition in videos”. In: International Conference on Neural Information Processing Systems, 2014, pp. 568–576.

[86] Sincan, O. M.; Keles, H. Y. “Autsl: A large scale multi-modal turkish sign language dataset and baseline methods”, *IEEE Access*, vol. 8, October 2020, pp. 181340–181355.

[87] Slimane, F.; Bouguessa, M. “Context matters: Self-attention for sign language recognition”. In: International Conference on Pattern Recognition (ICPR), 2021, pp. 7884–7891.

[88] Soomro, K.; Zamir, A. R.; Shah, M. “Ucf101: A dataset of 101 human action classes from videos in the wild.”, Technical Report, Center for Research in Computer Vision, University of Central Florida, Orlando, FL, USA, 2012, 7p.

[89] Stokoe, W. C. “Sign language structure”, *Annual review of anthropology*, vol. 9–1, October 1980, pp. 365–390.

[90] Stoyanov, D.; Taylor, Z.; Carneiro, G.; Syeda-Mahmood, T.; Martel, A.; Maier-Hein, L.; et al.. “Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support”. Granada, Spain: Springer Cham, 2018, 1 ed., *Lecture Notes in Computer Science*, vol. 11045, 387p.

[91] Sutton, R. S.; Barto, A. G.

“Reinforcement Learning”. Cambridge, MA: Bradford Books, 1998, 342p.

[92] Tan, P.-N.; Steinbach, M.; Karpatne, A.; Kumar, V. “Introduction to Data Mining (2nd Edition)”. Pearson, 2018, 864p.

[93] Tang, X.; Chang, X.; Chen, N.; Ni, Y. M.; LC, R.; Tong, X. “Community-driven information accessibility: Online sign language content creation within deaf communities”. In: CHI Conference on Human Factors in Computing Systems, 2023, pp. 1–24.

[94] Tavella, F.; Schlegel, V.; Romeo, M.; Galata, A.; Cangelosi, A. “Wlasl-lex: a dataset for recognising phonological properties in american sign language”. In: Annual Meeting of the Association for Computational Linguistics, Muresan, S.; Nakov, P.; Villavicencio, A. (Editors), 2022, pp. 453–463.

[95] Tong, Z.; Song, Y.; Wang, J.; Wang, L. “Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training”. In: Advances in Neural Information Processing Systems, Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; Oh, A. (Editors), 2022, pp. 10078–10093.

[96] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. “Attention is all you need”. In: Advances in neural information processing systems, 2017, pp. 5998–6008.

[97] Vicars, W. G. “American sign language university”. Accessed in: March 2023, Source: https://www.lifeprint.com/, 1997.

[98] von Agris, U.; Knorr, M.; Kraiss, K.-F. “The significance of facial features for automatic sign language recognition”. In: IEEE International Conference on Automatic Face & Gesture Recognition, 2008, pp. 1–6.

[99] Wilbur, R. “Phonological and prosodic layering of nonmanuals in American sign language.” Psychology Press, 2000, chap. Chapter IV, pp. 215–244.

[100] World Federation of the Deaf. “Our work”. Accessed in: Nov 2023, Source: http://wfdeaf.org/our-work/, 2024.

[101] World Health Organization. “World Report on Hearing”. World Health Organization, 2021, 252p.

[102] Xiong, S.; Zou, C.; Yun, J.; Jiang, D.; Huang, L.; Liu, Y.; Xie, Y. “Continuous sign language recognition enhanced by dynamic attention and maximum backtracking probability decoding”, *Signal, Image and Video Processing*, vol. 19–141, December 2024, pp. 1–13.

[103] Zhang, H.; Cisse, M.; Dauphin, Y. N.; Lopez-Paz, D. “mixup: Beyond empirical risk minimization”. In: International Conference on Learning Representations, 2018.

[104] Zhou, B.; Lapedriza, A.; Xiao, J.; Torralba, A.; Oliva, A. “Learning deep features for scene recognition using places database”. In: Advances in Neural Information Processing Systems, Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N.; Weinberger, K. (Editors), 2014, pp. 1–9.

[105] Zhu, X. “Semi-supervised learning literature survey”, Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005, 60p.

MARISTA PUCRS

Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Pesquisa e Pós-Graduação
Av. Ipiranga, 6681 – Prédio 1 – Térreo
Porto Alegre – RS – Brasil
Fone: (51) 3320-3513
E-mail: propesq@pucrs.br
Site: www.pucrs.br