Advancements in Real-Time Sign Language Translation

by

Eshtiak Alam Shihab
21301502
Md Shamsur Shafi Nur E Aziz
21301432
Kazi Israrul Karim
21301509
Tashfia Saad
21301320
Neelavro Shafin Qais
21301501

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
February 2025

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Eshtiak Alam Shihab
21301502

Md Shamsur Shafi Nur E Aziz
21301432

Kazi Israrul Karim
21301509

Tashfia Saad
21301320

Neelavro Shafin Qais
21301501

Approval

The thesis/project titled "Transforming Sign Language Recognition: Leveraging Deep Learning for Improved Accuracy and Real-Time Translation" submitted by

1. Eshtiak Alam Shihab(21301502)
2. Md Shamsur Shafi Nur E Aziz(21301432)
3. Kazi Israrul Karim(21301509)
4. Tashfia Saad(21301320)
5. Neelavro Shafin Qais(21301501)

Of Fall, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on February 03, 2025.

Examining Committee:
Supervisor:
(Member)

Dr. Farig Yousuf Sadeque
Associate Professor
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

The need for effective sign language recognition and translation has become more critical to create a more inclusive society that addresses the communication concerns of the Deaf community. In recent years, the field has seen a revolutionary progress arc, spearheaded by the development of transformative Deep Learning based approaches such as reinforcement learning, spatio-temporal residual networks, temporal convolution modules, iterative alignment networks, and attention mechanisms. Yet, vision-based real time continuous sign language recognition (CSLR) continues to face several application challenges, encompassing its visual, sequential, and alignment modules. As such, we propose an end-to-end training model inspired by the recent successes of transfer learning and attention-based mechanisms in particular to achieve new state-of-the-art performance on current benchmarks. Our paper includes two variations of approaches to dealing with continuous sign language videos: a classification approach and a translation generation approach. It eventually highlights the suitability of the translation-based approach for this domain of research. A comparative analysis between Classification based and generation based model highlights the superior efficiency and accuracy of the latter, making it the most suitable model for real-time, sentence-level sign language-to-text translation. Furthermore, our optimized inference strategy significantly reduces latency, ensuring real-time translation speeds, which is a crucial requirement for practical applications in accessibility and assistive communication technologies.

Keywords: Sign Language Recognition; Continuous Sign Language Recognition; Deep Learning; Transfomers; ResNet50; Video Vision Transformer (ViViT); TimeSformer; Connectionist Temporal Classification; Spatio-temporal Residual Networks; Attention-based Mechanisms; Transformers

Table of Contents

Declaration i

Approval iii

Abstract iv

Table of Contents v

List of Figures vii

1 Introduction 2
- 1.1 Aims and Objectives . . . 3
- 1.2 Research Objectives . . . 3
  - 1.2.1 Data Collection and Preprocessing . . . 3
  - 1.2.2 Model Development and Implementation . . . 4
  - 1.2.3 Performance Evaluation . . . 4

2 Literature Review 5
- 2.1 Survey Methodology . . . 5
- 2.2 Analysis of Sign Language . . . 5
  - 2.2.1 Isolated Sign Language Recognition (ISLR) . . . 7
  - 2.2.2 Continuous Sign Language Recognition . . . 10
- 2.3 Challenges of Sign Language Recognition . . . 12
  - 2.3.1 Feature Extraction and Gloss Segmentation . . . 12
  - 2.3.2 Alignment . . . 14
  - 2.3.3 Translation . . . 16
  - 2.3.4 Multilinguality . . . 21
  - 2.3.5 Dataset . . . 21

3 Work Plan 24

4 Dataset 25
- 4.1 Dataset Analysis . . . 25
- 4.2 Data Pre-processing . . . 30

5 Technology Overview 31
- 5.1 ResNet50 . . . 31
- 5.2 Connectionist Temporal Classification (CTC) . . . 32
- 5.3 Long Short-Term Memory (LSTM) . . . 33
- 5.4 Video Vision Transformer (ViViT) . . . 34

5.5 TimeSformer 35
5.6 Bidirectional Encoder Representations from Transformers (BERT) . . 36

6 Methodology **38**
6.1 Classification using Neural Network 38
6.2 Generation Based model using Transformers 39

7 Result Analysis **48**
7.1 Inference Time and Efficiency Analysis 51
7.2 Analysis of Prediction Accuracy 53
7.2.1 Selection of Best Model 55

8 Limitations and Future Work **56**

9 Conclusion **58**

Bibliography **62**

List of Figures

1.1 Overview and Pipeline for Sign Language Recognition 4

3.1 Workplan Gantt Chart 24

4.1 Example of frames from a video segment 27
4.2 KDE Plot of Frames per Gloss Annotation 28
4.3 Bar chart showing each signer's most frequently used word and its usage count. 28
4.4 Optical flow visualization representing motion magnitude between consecutive frames. 29
4.5 Bar chart of the top 50 words in the PHOENIX 2014T "orth" column, sorted by frequency. 29
4.6 MediaPipe detected landmarks including body joints, facial key points, and hand landmarks overlaid on an image. 30

5.1 Connectionist Temporal Classification (CTC) architecture for sign language translation. 33
5.2 LSTM Architecture [32] 33
5.3 A two-stage Factorised encoder : the first encoder captures spatial interactions per time step, while the second models temporal dependencies, enabling late fusion of spatial and temporal information by [1]. 35
5.4 The different attention mechanisms of TimeSformer [2]. 36
5.5 BERT pre-training and fine-tuning process by [5]. 37

6.1 Illustration of the overall pipeline. 39
6.2 Attention Mechanism 42
6.3 The BiLSTM architecture [23] 44
6.4 Overview of the Sign Language Translation Pipeline using ViViT as the encoder 46
6.5 Overview of the Sign Language Translation Pipeline incorporating TimeSformer. 47

7.1 Comparing Test Image with Closest Match from Training Dataset . . 48
7.2 Training Progress: Loss vs. Epochs for Convergence. 50
7.3 Inference time comparison between Transformer and TimeSformer Encoders using Bar chart. 52
7.4 Comparison between a randomly selected frame from LSA64 and PHOENIX 2014-T datasets to show some of the additional challenges associate with training a model on the PHOENIX dataset 54

7.5 Comparison between Traditional Pipeline, Transformer and TimeSformer Encoders using Bar chart. 55

Chapter 1

Introduction

Sign Language is the fundamental mode of communication for around 6.1% of the entire population of the earth, the Deaf community, according to the World Health Organization. There exists regional variations in dialect and certain linguistic aspects within the sign language communities owing to the unique characteristics of American, Bengali, German, Chinese, Russian and other sign languages. These languages not only exist as a linguistic tool, but also as a medium which allows people with hearing and speech disabilities to carry out regular everyday activities such as going to work, providing service to others, expressing their needs in required times, etc. From individuals who are born with hearing loss to those who lost their hearing midway through their lives, Sign Language provides the opportunity to remain an active and interactive individual of the community.

However, unlike spoken language, sign language expressions incorporate a manual component such as hand shape, pose etc alongside non-manual ones pertaining to facial expressions. It also comes with its own set of vocabulary, grammar and linguistic nuances. As such, there exists a communication gap between a signer and a speaking-hearing person which can be bridged effectively by Automatic Sign Language Recognition [11]. The sensor-based approach to achieve this has been increasingly replaced by vision-based ones in recent years. Subcategories of the latter involve Isolated Sign Language Recognition (ISLR) which aims at identifying the gloss illustrated by a specific video segment. In contrast, Continuous Sign Language Recognition (CSLR) takes on the more complex challenge of predicting an appropriate gloss sequence from a sign language video.

Despite remarkable outcomes, CSLR is still undergoing progressive improvements. The traditional approaches involved hand-crafted feature extraction followed by naive Machine Learning models such as Support Vector Machines. The standardised approach today involves more robust frameworks that utilise Deep Learning for automatic feature extraction. The functionality of these frameworks is distributed among 3 modules: visual, sequential and alignment. The purpose of the visual module is to extract visual cues from video frames. The sequential model then uses the extracted features to mine correlation between glosses [31]. Eventually, the alignment module maps the correct video frames and gloss sequences. An objective function of Connectionist Temporal Loss (CTC) is usually associated with these three modules to leverage its ability of automatic alignment to allow for end-to-end

training of models. The initial models relied on Hidden Markov Model (HMM) but have been replaced by Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) architectures lately. Long Short-Term Memory (LSTM) is also integrated into most frameworks. The resultant models have achieved great success on frequently used large benchmark datasets such as the PHOENIX-2014, PHOENIX-2014-T, and the CSL datasets. Our goal is to contribute to the same cause by improving a Sign Language recognition and translation system with high accuracy and addressing some barriers that continue to pose challenges.

The following sections of our paper encompass our research statement and objectives. Figure 1.1a presents an overview of the different approaches to Sign Language Recognition (SLR), categorizing them into isolated sign language recognition (ISLR) and continuous sign language recognition (CSLR). The diagram further breaks down CSLR into sub-methods, showcasing the evolution from traditional methods to deep learning models. Then, Figure 7.4b illustrates the pipeline for continuous sign language recognition and translation, detailing the process from input to output. The visual module captures and processes the sign language input, followed by the alignment module, and culminating in the translation module that converts the signs into English sentences. These are followed by a deep-dive into existing works in this field, particularly in the past 3 to 5 years. We explore the prominent breakthroughs which influence the research landscapes of isolated and continuous sign language recognition, focusing more on the latter. We categorically look into recent studies trying to resolve the more challenging sub-domains within CSLR: feature extraction and gloss segmentation, alignment, translation, multilinguality and dataset. Finally, we end with our tentative work plan and a concluding remark on the work conducted to achieve accurate and representative CSLR so far.

1.1 Aims and Objectives

The research addresses the current challenges in the evolving landscape of Continuous Sign Language Recognition by analysing the latest developments and progress in the field, ensuring a comprehensive understanding. As such, the research objective aims to enhance Real-Time Sign Language Recognition by leveraging advancements in deep learning models and transformers, thereby facilitating seamless communication between the deaf and hard of hearing communities.

1.2 Research Objectives

1.2.1 Data Collection and Preprocessing

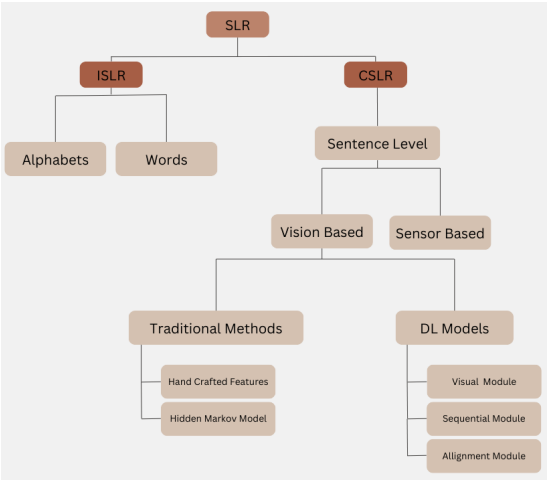
- Collect datasets which cover the sentence-level video structure of Sign Language Recognition (SLR), as they contain continuous data suitable for ISLR & CSLR.
- Apply standard video and image preprocessing techniques such as noise reduction, segmentation, and data augmentation for accurate video fragmentation and feature extraction.

1.2.2 Model Development and Implementation

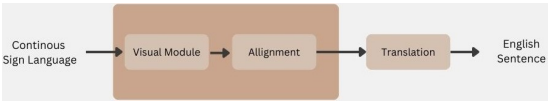
- Implement state-of-the-art machine learning techniques and deep learning models to extract relevant frames and glosses from video input.
- Employ an advanced sequence-to-sequence (seq2seq) modeling technique for translation.
- Enhance an existing framework to serve as a real-time sign language interpreter.

1.2.3 Performance Evaluation

- Assess the performance and effectiveness of the proposed models and methods using appropriate metrics and methodology.
- Differentiate the performance and results of our model with other current state-of-the-art models.



(a) An overview of different approaches to Sign Language Recognition (SLR)



(b) Pipeline for Continuous Sign Language Recognition

Figure 1.1: Overview and Pipeline for Sign Language Recognition

Chapter 2

Literature Review

2.1 Survey Methodology

We thoroughly read many technical research articles related to Sign Language Recognition. We began our research by first understanding the complex nature of sign language and how they differ from our spoken languages. Then we moved on to the latest review papers to understand the challenges faced in this field. Finally, we narrowed down our research to understand how different state-of-the-art models are working on these challenges and developing more effective and accurate Sign Language Recognition models. Furthermore, our paper selection method was to consider relatable papers with large numbers of citations excluding very similar research. We highly focused on recent publications as well. By systematically categorising these selected papers according to the stages of CSLR, we can construct a coherent and comprehensive body of knowledge that will underpin our future research endeavours.

2.2 Analysis of Sign Language

To begin with, Sign Language recognition and translation is an extensive research field, with the primary goal - to accurately curate translations of Sign Language, which has been approached in various ways by different researchers. Achieving this goal comprehensively by addressing every single aspect has proved to be a significant challenge, even with modern cutting-edge technology. Hence, several different studies have each tackled one or numerous facets of the broader problem.

The review paper by Manoharan and Roy [25] mentions the branches into which SLR can be divided and summarises the advancements done in each sub-field between 2009 and 2021. It also discusses the major limitations of these models and approaches. In almost every research there are six major features with respect to SLR - firstly, SLR Translation can either be isolated alphabet/word recognisable for which an isolated dataset has to be used, or it can be continuous SLR Translation for which continuous dataset has to be used. Secondly, the user input can be obtained either via camera or likewise components, i.e vision related hardware or via sensory gloves, i.e sensor based hardware. Next, there are two kinds of features that can be processed for the recognition and translation, manual markers (i.e hand signals), and non-manual markers (i.e eyebrows, nose, chin, mouth, nose, cheek, etc.). Usu-

ally, manual markers are used for the translation itself, while non-manual markers are used for a more comprehensive and contextual analysis behind the translation. To research on these aspects, many different models like deep learning models, traditional methods, hybrid methods can be implemented. However, their application comes with challenges. Other than the gesture recognition and translation process being complex, there is the pressing issue of context. Comprehensive understanding can only be achieved by considering the several non-manual markers simultaneously with the gestures. Some other significant barriers include scaling and orienting the image properly in case of different signers with variable physical attributes, light intensity, and the level of interference by non-uniform and dynamic backgrounds. These issues can be found in both vision and sensor-based input methods. From this paper, we also get an outline of the Isolated Manual SLR research during the period. Chronologically, the challenges faced by earlier papers have been addressed and resolved over the years. As such, in the last few years, with model accuracy reaching up to 99.93%, the research demands have shifted away from Isolated SLR. Instead, researchers have been intrigued by the fact that accuracy obtained by recent papers have not met the highest standards when it comes to Isolated Non-Manual SLR work, reaching a maximum of 78%. Even when trained by ML or Deep Learning models, there remain significant barriers concerning visual sensitivity, and insufficient dataset and real-world experimental testing. Inadequate real-time accurate classification and translation, and heavy computational time and resource requirements also present challenges. Research overview on continuous SLR shows that even accuracy involving manual works are as high as 98% in recent papers. Hence, there remains very limited opportunities to contribute any further improvements here too. On the contrary, for Continuous Non-Manual SLR, the accuracy obtained using recent technology are comparatively much lower, reaching a highest of only 28%. Older methodologies have obtained 92% accuracy, but failed to address some crucial classification issues that have surfaced later. Analysing the statistics have revealed that even with DL models, sufficient continuous datasets are extremely limited. As such, several models perform well with the training dataset but do not perform similarly on other ones (overfitting). Other barriers include high training time and complexity, sensitivity to overlapping during classification and unalignment problems.

It is evident that there has already been extensive research on both isolated and continuous manual marker dependent research for SLR [25]. In their study, Wali, Shariq, Shoaib, *et al.* [38] delves deeper into some of the aforementioned branches within this field. They examine the primary obstacles in Sign language recognition (SLR) that are associated with written text units, including letters, words, and sentences. Every unit poses individual challenges and necessitates specific strategies. At the letter level, SLR addresses challenges involving the recognition of characters, shapes, or objects. Pre-trained models like VGG-16, have demonstrated commendable outcomes by utilising transfer learning. Ultimately, the sole prevailing issue in letter-level pertains to the identification of shapes in dynamic environments. Word-level spoken language recognition can be categorised into two main types: static recognition and sequential recognition. Static word recognition is identifying individual signs that are captured in a single frame. In contrast, sequential word recognition involves temporal dimension, adding complexity to the task. Wali, Shariq, Shoaib,

et al. [38] recognize that identifying words in continuous signing sequences is particularly strenuous as there are no noticeable breaks between signs. Consequently, this level presents various obstacles, such as the identification of hands, classification of sequences with multiple inputs, and the presence of changing backgrounds. Finally, for sentence-level or CSLR, the most complex amongst the challenges, a general three-step dissection procedure is presented: converting signed videos into gloss alignment, recognizing individual glosses (the basic lexical units in sign language), and translating these glosses into coherent sentences. This mechanism faces several barriers such as consecutive signs having no visible pauses, making boundary detection difficult, challenges of sequence-to-sequence (seq2seq) modelling, and gloss segmentation. Then, the study suggests that while Transformers show great potential, they typically require extensive datasets to be effective. Therefore, further research is recommended to identify the most suitable pre-trained Transformer models for SLR and to explore how these models can be adapted to achieve improved results. Reiterating the premise established in the previous paper about the significance of non-manual markers in SLR, the need for incorporating facial expressions into the segmentation module has been highlighted for future research here as well. Ultimately, the fundamental problem in current CSLR research has been the lack of cohesive studies building upon one another to refine methods and approaches. However, as we will be seeing in our paper later this landscape is changing rapidly in the last three years. In conclusion, while significant progress has been made in SLR, the field faces ongoing challenges related to segmentation, alignment, and the integration of non-manual signals. Cohesive research efforts and the exploration of advanced models like transformers are essential for future advancements.

2.2.1 Isolated Sign Language Recognition (ISLR)

One of the most highlighted works in the field of ISLR was published by Joshi, Sierra, and Arzuaga [17]. Their paper consists of some of the most fundamentally relevant and simple techniques to translate ASL gestures taken via user video input to produce both English characters and English words using manual markers entirely. The authors address the problem in the massive gap between the communication of individuals from the Deaf community and the rest of the world by creating a medium that accepts inputs from generally used devices such as computers and other generic webcams. After ensuring that the model is user friendly by not requiring complicated input hardware, the authors create a software which analyses the video input frame by frame and compares each frame's images to their dataset, the characters or words are formulated. Finally, they formulate the whole word by applying a gesture recognition method. This involves calculating the cross-correlation between the frames to eventually deduce the full word. For their research, the authors create their own dataset which consists of two parts. Firstly, the Alphabet Database (consisting of ASL A-Z images) which is created with the help of their aforementioned sign image acquisition software. The software takes image inputs of the signs for each alphabet, then uses image segmentation to convert the image inputs from RGB to binary (black and white). This helps eliminate any interference from the surrounding background. To further remove noise, they carry out morphological filtering. The original image input is also passed through edge detection algorithm

and the two images (image with edge detection and morphologically filtered image) are combined to be finally stored as a comparison model to be used for a particular alphabet. They create 10 such samples for each alphabet. Secondly, to curate the ASL Word Database, they record video input, on which the same steps as Alphabet Database is carried out and then every frame is matched with the alphabets and a cross-correlation matrix is used to find the similarity between the two images to come to a conclusion. Instead of using any particular Machine Learning model, the authors carry out the translation procedure using their software. For Alphabet translation, again image segmentation, morphological filtering and edge detection is carried out on the input image and finally correlation matrix used to come to an identifiable conclusion. For word/sentence translation, the video input is divided into frames and for each frame, the same steps as before are carried out to give an output of the sentence. For alphabet translation and word translation, the accuracy obtained is 94.23% and 92%, respectively.

Amongst later attempts at recognising signs as independent entities, Matlani, Dadlani, Dumbre, *et al.* [27] achieved some interesting results. Their paper acknowledges two methods for sign language recognition: from images or by using sensors to track the data. It proposes a method based on machine learning and neural networks for real-time sign language recognition followed by converting it into speech or text and vice versa.Matlani, Dadlani, Dumbre, *et al.* [27] observes that the performance of RNN, CNN, ANN and HMM models is compromised in the absence of large datasets. Hence, they opt for a different approach by using a google open source framework, Mediapipe, to recognise human body parts. For the dataset, they use Indian Sign Language (ISL) and ASL, consisting of signed images of each character in the English alphabet and numbers from 0-9. The input video is captured using either a webcam or via real time video capturing which contains hand signs. The hand trajectories are later tracked using google's library mediapipe where mediapipe found out coordinates associated with the sign captured. These coordinates, referred to as key points, then undergo a 2-stage pre-processing for training purposes. Consequently, data files are separated into training and test sets. Finally, DT, KNN, Random Forest, SVM and XGBoost were used for training. A variety of regression models such as Linear Regression, SVR and XGBoost regressor were also implemented and the paper recommends using a light NN for eventually matching the signs with appropriate letters. It also states that this system is resilient and cost effective because it achieves its goal of ISLR without using any smart sensors or expensive computational power. Moreover, the research reveals that Mediapipe can be effectively used to recognise complex hand signs exactly with very high accuracy, ranging between 78% to 100%.

More recently, Kezar, Thomason, and Sehyr [19] provide a transformative novel approach of improving the accuracy of ISR by leveraging the use of phonological features of sign language such as identifying the minor locations (chin), the shape of the hand, and path movement of the hand through space (circular) etc. For the dataset, they combine the phonological annotation in ASL-LEX 2.0 with the signs in WLASL 2000 ISLR benchmark, in total adding 16 new columns each indicating different phonemes. A graph convolutional network based on SL-GCN and a Transformer based model is used.The SL-GCN uses spatial and temporal attention to

learn the different pose- estimation. Then it is passed on to the transformer which uses 5-layers to learn the sequences of the coordinates with respect to time. Using these models helps to capture the phoneme even if it is for a short amount of time. They modify the decoder to classifier by implementing n fully-connected layers as it not only apprehends the targeted gloss but also the phonemes. Furthermore, the model is trained until it improves the validation accuracy in the last 30 epochs.Training the 16 labelled phonemes was not cost efficient so to further improve the model they used an utility function:

$$P^*(n) = \left\{ \arg\max_P U(P) : |P| = n \right\}$$

The models were trained with their auxiliary loss for $P^*(n)$, $n \in \{2, 5, 9, 16\}$. It was concluded that handshapes and minor location were the most significant phonemes as they improved the result the most. Evaluation metrics used were top-1, top-3 and MRR (mean reciprocal rank). The model shows a 3-9% gain in top-1, 5-11% gain in top-3 and 0.04-0.09 gain on MRR. However there were limitations as the existing dataset had incorrect labels, information of the signers like age, race, fluency is also missing. To conclude, the use of phonemes during training enables the models to learn a more holistic approach to ISLR rather than just learning the target gloss, overall showing a significant improvement in the results.

On the other hand, Jamwal, Vasukidevi, Malleswari, *et al.* [16] explores a new avenue of ISLR by incorporating emotional analysis using ML models into their research on sign language recognition and translation. Previously, as evident from our investigations, the key association of non-manual markers with the emotions expressed through signs had been well established. Hence, this paper uses non-manual markers namely eyes, eyebrows, nose tip, lips to classify the sentiment. For training the models, Jamwal, Vasukidevi, Malleswari, *et al.* [16] use a static hand sign model dataset whose details have not been included. For the gesture translation itself, the authors used the MediaPipe Regression model that was trained with over 30,000 real world images. Eventually, the gesture has been split into five categories - angry, happy, sad, surprise and neutral. They use the Support Vector Machine (SVM) model and a custom Convolutional Neural Network (CNN) model alongside the MediaPipe to obtain a 0.77 recall and F1-score, and a 0.81 average score. The paper reflects upon a heavy dependency on the CNN model for both translation and emotion classification. Additionally, it should be noted that a major limitation of this approach was that the emotional recognition was solely independent of the translation. So for instance, if a positive statement was signed with a sad facial expression, the system would give the translation with a sad emotion, which is contradictory. Regardless, Jamwal, Vasukidevi, Malleswari, *et al.* [16] achieves significant strides in integrating emotion analysis with sign language recognition. In another research, Jalaja and Shekar [15] build a two-way communication platform for the Deaf community aimed at sentiment analysis based on the text context. Their work is of significance since it includes a class of Deaf people who may not be literate to understand the English language. The system involves character level sign recognition and translation. No detailed information is given about the dataset used, other than the fact that it is composed of all 26 alphabets of the English Language. For the model, they use a custom CNN model along with RNN models in order to display predictive text for faster typing in the user interface. Firstly by implementing CNN and RNN

architecture, they create a sign language recognition and translation system. Consequently, they propose the design for a robotic arm, which would perform the gesture of the text inputted by the user into the software interface. The robotic arm is 3D, portable and printable, and can be further optimised using advanced controllers such as RaspberryPi, etc. Eventually, using the DenseNet architecture, Jalaja and Shekar [15] reach an accuracy of 98.07%. Their sentiment analysis model achieves a remarkable accuracy of 93%. Although there exists future scope for implementing other architectures such as ResNet to see the impact on accuracy, this model sets a high benchmark in sentiment analysis from ISLR with its impressive accuracy rates.

2.2.2 Continuous Sign Language Recognition

Driven by the potential for contribution in continuous sign language recognition (CSLR) and upon careful consideration of its potential challenges, we are motivated to work in this field. Majority of our following research concerns the recent breakthroughs in this branch of sign language recognition.

In 2019,Cui, Liu, and Zhang [4] introduce a sophisticated framework designed to enhance continuous sign language recognition (CSLR) through the application of deep neural networks. This framework directly transcribes videos of SL sentences into ordered sequences of gloss labels, addressing the limitations of traditional methods that rely heavily on hidden Markov models (HMMs) and hand-crafted features. These traditional approaches often fail to capture the complex temporal dynamics of SL and require costly temporal boundary annotations for each gesture. The proposed architecture is composed of two primary modules: spatiotemporal feature extraction and sequence learning. The spatiotemporal feature extraction module employs deep convolutional neural networks (CNNs) with stacked temporal fusion layers to effectively capture both spatial and temporal features from video data. Meanwhile, the sequence learning module utilises bidirectional recurrent neural networks (RNNs) to learn temporal dependencies and align sequences of gloss labels with video frames, ensuring coherent and accurate transcription. A notable innovation in this research is the iterative optimization process designed to maximise the representation capabilities of deep neural networks, particularly when data is limited. This process begins with an initial end-to-end training phase to generate alignment proposals. These proposals then serve as strong supervisory signals for fine-tuning the feature extraction module. By iteratively refining this process, the framework progressively enhances recognition performance. Specifically, the iterative training involves repeating the alignment generation and fine-tuning steps multiple times, each iteration improving upon the last. Furthermore, the authors explore the RGB image and optical flow data fusion to enhance recognition accuracy. Optical flow provides motion information that complements the spatial cues captured by RGB images, leading to a more comprehensive understanding of dynamic gestures in SL videos. This multimodal approach significantly boosts the system's ability to capture and interpret the nuances of SL. The framework is rigorously evaluated on two challenging SLR benchmarks: RWTH-PHOENIX-Weather 2014 and CSL (Chinese Sign Language) datasets. The results demonstrated substantial improvements over state-of-the-art techniques, with a relative performance increase of over 15%

on both datasets. On the RWTH-PHOENIX-Weather 2014 dataset, the proposed method achieved a word error rate (WER) reduction from 27.1% to 23.0%, while on the CSL dataset, the WER was reduced from 38.3% to 32.2%. These significant performance gains underscore the effectiveness of the deep neural network architecture and the iterative training strategy. In comparison with previous methods, the proposed framework shows clear superiority. Traditional HMM-based methods and non-iterative deep learning approaches often fail to capture the intricate temporal dependencies and dynamic variations in SL videos. The iterative process and multimodal fusion approach adopted in this study address these shortcomings, leading to more accurate and reliable SL recognition. In terms of future scope, the framework's reliance on deep neural networks opens numerous possibilities for further research and improvement. Future work could explore the integration of more advanced neural network architectures, such as transformers, which have shown remarkable performance in various sequence-to-sequence tasks. Additionally, refining the multimodal fusion process by incorporating additional data modalities, such as depth information or skeletal data, could further enhance recognition accuracy. Expanding the dataset to include more diverse SL videos from different sign languages and contexts would also improve the model's generalizability and robustness. Overall, the proposed framework presents a robust and efficient approach for CSLR, providing a strong foundation for related applications in multimedia and computer vision in the future as well.

On the other hand, research done in 2020 emphasises the contribution of non-manual markers in CSLR and translation [30]. The paper works on K-RSL because in this language, non-manual markers can alter the context of a sentence entirely. To differentiate, the sentences need to be analysed on two levels: lexical and morphological. For lexical distinction between signs that have similar manual features, facial expressions, particularly mouthing, are crucial. On the other hand, at the morphological level, these facial expressions and mouth patterns convey information equivalent to adjectives and adverbs [30]. To elaborate, many sign languages rely on head movements to express negation, while eyebrow and head positions are key factors in determining whether a sentence is a statement or an inquiry. To test the contribution of these less-considered aspects of sign language, the research team creates their own Russian/Kazakh Sign Language dataset. It is relatively smaller than popular benchmark datasets used and consists of 5200 isolated frequently-used sign samples as well as sentences signed by five native Kazakh signers. The recorded signs are chosen such that although the gestures are similar, involvement of non-manual cues alters the meaning of the phrase or sentence in K-RSL. Their approach relies on a ML model, Logistic Regression, to compare the accuracy between including and excluding non-manual markers from the translation process. Their model achieves a mean accuracy score of 73.4% and 77% for testing with and without non-manual features, alongside manual ones, respectively, suggesting a 3.6% improvement in the overall translation module [30]. Alongside, 8 additional question sentence samples, which were misclassified using only manual features, are correctly identified when non-manual cues are considered. Among statement samples, there is also an increase of 5 in accurate classification. These findings underscore the relevance and significance of integrating non-manual markers to enhance the accuracy of sign language translation systems.

2.3 Challenges of Sign Language Recognition

SLR continues to present multiple challenges that researchers are yet to fully resolve. The four most prominent hindrances, alongside their corresponding most promising progresses, have been addressed below.

2.3.1 Feature Extraction and Gloss Segmentation

Fang, Wang, Lin, *et al.* [8] came up with an interesting hypothesis that visual similarities between sign sequences can be attributed to the similarity in semantics expressed by them. Such a claim was based on real-world observations. They designed an end-to-end framework where at first, input of video streams are passed on the 2D CNN to extract the frame-level features from each frame. Then it is passed onto the KPBR structure for short-term temporal feature learning. To break down the KPBR structure, K stands for 1D convolution on the time dimension, P stands for the 1D-max pooling layer, lastly B and R stands for batch normalization and ReLU activation respectively. The clip level features are passed onto the BiLSTM for long-term feature learning. Lastly, CTC loss aligns the prediction matrix. After that comes the most important part of the architecture, the Reinforcement Learning (RL) model which segments the clip-level features into small groups, giving them pseudo labels. The minimum cosine distance among the features allows the RL agent to create the groups of similar results.Then using gradient ascent for RL and backward propagation an optimized and fully working end-to-end model is created. The dataset used to evaluate the model was once again the popularly used RWTH-PHOENIX-Weather 2014 and CSL dataset. A total of four experiments were carried out to compare the results, first an end-to-end model without auxiliary task, second experiment uses auxiliary task but only decodes a prediction sequence and uses Cross-entropy. The third experiment was the designed VFS-RL method without the CTC auxiliary loss. Finally in the fourth experiment the intact VFS-RL model was used. For the evaluation metric, the commonly used WER was used to compare the results, and the VFS-RL method showed the best WER result of 24.4 in Dev and 24.9 in test. Apart from the experimented results, the VFS-RL model was also compared to the other respective state-of-the-art end-to-end models which carried out their work on the CSL dataset and obtained the best result of 1.5 WER(%) while other methods like STMC, FCN had a WER(%) of 2.1,3 respectively. To conclude, the use of RL contributes to a more discriminative feature extractor which also excels at grouping similar gestures together.

Meanwhile,Moryossef, Jiang, Müller, *et al.* [29] utilizes linguistic cues observed in sign language corpora to introduce a novel approach to segmentation problems. Although spoken language text can be divided into a linear sequence of words with the help of punctuation marks, sign language's simultaneity obstructs it to such linear equations. Previous segmentation work typically used binary classification to predict if each frame or pixel is part of a segment, neglecting the seamless transitions in continuous signing where boundaries are not clearly defined. This paper replaces

this predominant Inside-Outside (IO) tagging scheme with Beginning-Inside-Outside (BIO) tagging leveraging linguistic prosodic cues. These cues are also known as non manual markers that have repeatedly come up in our discussions so far. This approach unifies individual sign segments and phrase segments (larger units comprising several signs) using a mechanism where in addition to predicting a frame to be in or out of a segment, it also classifies the beginning of the segments. This paper confronts phrase boundaries using optical flow features as a proxy for prosodic processes and sign boundaries utilizing linguistic information such as limited number of hand shapes in a sign and dominant hand's signal. This paper uses The Public DGS Corpus dataset which is a comprehensive linguistic dataset that includes both accurate sign-level annotation from continuous signing, and well-aligned phrase-level translation in spoken language. The proposed sign language segmentation method involves adjusting video to 25 fps, estimating and normalizing poses with MediaPipe Holistic, applying optical flow and 3D hand normalization, encoding sequences with an LSTM, using BIO tagging for classification, and decoding segments using a greedy algorithm. The experimental results show significant improvements in segmentation performance. The model achieves an F1 score and IoU score of 63% and 69% respectively for sign segmentation, alongside achieving 65% and 85% respectively for phrase segmentation. These results outperform previous IO tagging models. Moreover, the model demonstrates robustness in out-of-domain settings, maintaining competitive segmentation scores in zero-shot evaluations on different signed languages. The inclusion of optical flow and 3D hand normalization enhances the model's robustness in such contexts. In conclusion, this study presents an effective approach to sign language segmentation that significantly improves accuracy and shows promise for real-world applications. The findings suggest that incorporating linguistic cues and advanced modelling techniques can enhance the performance and robustness of sign language processing systems.

On the other hand, Rao, Sun, Wang, *et al.* [35] address the need for improvement in gloss representations. They present a simpler yet very effective technique of cross-sentence gloss consistency. Their strategy is to not only focus on the gloss correlation in an individual sentence but also consider the relationship among glosses from different sentences to achieve a richer correlation. This would eventually boost feature extraction performances significantly. To begin with, the proposed framework consists of three main components: gloss prototype (GP), gloss contrastive loss function and an auxiliary similarity fusion strategy (ASFS). The implementation was done by first aligning the visual backbone ResNet18, through which the gloss features are extracted. The GP stores prototypical features of all gloss categories, it serves as a comprehensive feature dictionary and is initialized by empirical/statistical distribution. It serves as a prototype memory bank for referral and to distil representative prototypes from diverse gloss samples. Alongside, it repressively refine the GP via momentum update, providing gloss feature references and gloss contrastive learning. A further use of gloss contrastive loss function reduces the distance between gloss sample point and its belonging prototype making it more distinguishable. ASFS used to fuse the intra-sentence recognition clues and cross-sentence clues using Softmax and CTC. Finally, sequences of words are generated using two-layer BiLSTM. The model was tested using the PHOENIX 14 and CSL Daily datasets and the evaluation metric used was WER. To be precise their model improves the current

state of the art performances on the Dev sets of PHOENIX14, PHOENIX14-T, and CSL-Daily by 1.6%, 2.4%, and 5.7% respectively, and by 1.4%, 1.5%, 5.3% on Test sets respectively. The primary limitation of the conducted research was the narrow domain range in the dataset as they mostly focused on weather-type information. To summarise, the idea of cross-sentences validation gives us well-distinguished gloss prototypes. Simultaneously, it improves the gloss discrimination with a gloss contrastive loss and an auxiliary similarity fusion strategy revealing notable improvements amongst the results.

2.3.2 Alignment

Alignment issues have long persisted in this domain of research. The lack of rigid annotation of words to videos is one of its contributing factors. It complicates the process of converting continuous Sign Language to accurate phrases. Pu, Zhou, and Li [34] tackled this challenge that presented itself under a CSLR setting. They proposed the idea of dividing the process into two approaches: feature extraction and an encoder-decoder network with CTC for sequence modelling. They conducted their experiments on two public datasets RWITH-PHEONIX and CSL. Firstly, they used a CNN model 3D-resnet to process the video frames to extract important spatial and temporal features. Therefore, this process converted the video into an order of feature vectors. Secondly, these vectors were sent to an Bi-LSTM network which caught dependencies in the video forwards and backwards in time. As a result, this provided an adequate comprehension of the sequence. Thirdly, they decoded the sequence by using a CTC decoder and LSTM decoder. The CTC decoder implemented the alignment of sequences and the LSTM used the attention tool to pinpoint the important parts of the input sequence to generate each word in the output. Lastly, they used Soft-DTW to make sure that the two sequences from CTC and LSTM are aligned perfectly. By merging the uses of the mentioned models, their architecture to recognise Sign Language ensured promising results. After training the models with 94 sentences, they tested their system using different sentences which were not included in the training set but had vocabularies from it. Their system recorded 0.327 WER, which was substantially lower than many state-of-the-art methods such as LSTM CTC, S2VT and HLSTM. In addition, they did an independent signers test where they trained their architecture with videos of 40 signers and tested with videos of 10 signers. In this test, the sentences of the training and testing set were the same. Therefore, they achieved 0.980 BLEU(Bilingual Evaluation Understudy) which was better than existing methods which gave 0.936 - 0.948 BLEU.

Similarly, the weak supervised sign language labels has been another prominent impediment for CSLR model performances. Hence, Xue, Yu, Yan, *et al.* [39] addresses it using an attention mechanism and an iterative alignment network. The sign recognition process comprises two main modules: feature extraction and sequence learning modules. The feature extraction based on iterative alignment network consists of spatio-temporal residual network (STRN), time series convolution module and a sequence learning module. Firstly, the sign language is given as input in block, the STRN then extracts the block level features, then fed into the TCN(Temporal Correlation Enhancement) which enhances the correlation between block-level fea-

tures and a three-layer bidirectional gated neural network. CTC is then used to learn the mapping relationship between the features and final sign language labels. Using this iterative approach the feature extraction is considerably refined, especially at word level feature. The word level features are then used as input into the encoder-decoder network. The encoder is a three layer LSTM unit and the decoder is based on an attention mechanism which generates the target sequence. The designed model is tested primarily on two datasets: RWTH-Weather-Phoenix 2014 (i.e. PHOENIX2014), and CSL. Once again, WER was employed as the evaluation metric. During result comparison, a few compromises were made. The 3D-ResNet uses a large number of parameters. As a result, the processing requires considerable computational power. As a countermeasure, Xue, Yu, Yan, *et al.* [39] adopted three-dimensional residual connections which reduces the cost. The final error rate emerged as low as 25.65% compared to the previous state-of-the-art model which had an error rate of 35.7% in the RWTH-Weather-Phoenix 2014 DATASET. In the CSL dataset, the WER(%) achieved was 1.7 compared to 2.1 under the STMC model (discussed later). To conclude, this framework uses its effective sequence mapping technique and the iterative alignment network to boost the accuracy of CSLR. However, it is worth noting that there are further scopes of using multi-modal fusion methods to improve the overall performance.

Amidst growing concerns over the alignment problem, cross-modality augmentation in the CSLR model emerged as a potential turn-around strategy. However, in 2024, a research identifies some of the key problems with this popular approach and provides a tentative solutions to it [10]. Their work is noteworthy because they do so by not only considering manual markers, but also non-manual markers to some extent. The traditional approach is non optimal because it ignores some crucial facial gestures in the video input during feature extraction[33]. This proves detrimental for accurate translation. This paper uses denoising-diffusion alignment (DDA) to tackle this problem. The DDA consists of two parts: a partial noise processor and a denoising-diffusion autoencoder. The former adds random Gaussian noise to the video part. This strategy enables the text modality to be a natural condition in the denoising process. Simply, the authors attempted to match up visual representations from videos with sequences of sign language glosses. The Denoising-Diffusion Autoencoder naturally conditions the textual sequence representations to denoising the part-noisy visual representation to achieve global alignment [10]. The DDA also has the capability of spontaneously enforcing the CSLR model to learn contextual information within and between both modalities, enhancing the generalisation of visual representation. The three datasets they worked on are PHOENIX-2014, PHOENIX-2014T and CSL-Daily. DDA outperforms the key points supervised TwoStream-SLR by 1.1% and 0.9% WERs on the dev and test set of PHOENIX-2014 and even surpasses it by 0.5% and 0.6% WERs on the dev and test set of PHOENIX-2014T [10]. Furthermore, it shows significant improvement compared to other multi-cue methods, which employ the pre-captured hands, face, key points, or heartmaps as supervision. Overall, the model is sensitive to sign movements and can capture the sign movements of the mouth, palm, head, and other sign events much more effectively to improve the sign recognition ability [10].

2.3.3 Translation

Amongst early attempts at conducting continuous sign language translation effectively, weakly-supervised learning have garnered popularity. However, the weakly-supervised learning approach have problems like weak labels and noisy data. Koller, Camgoz, Ney, *et al.* [21] propose a notion to resolve these aspects. This involves building a robust system for recognizing sign languages using a multi stream HMM to jointly align and synchronize mouth and hand shape patterns. Their research involves using the RWTH-PHOENIX Weather dataset. However, since this dataset does not have mouth shape annotations, they create an extension of the dataset by adding the mouth shape sequences from observing visible mouth shapes on it. Firstly, they used a parallel sequence learning where they divided the problem of mouth and hand shape recognition into sub tasks which worked parallel to each other as independent streams. Then these independent streams are modeled using a multi-stream HMM. Each stream handled sequence constraints specific to that channel of sign language. Consequently, to leverage parallelism, they create synchronization points among the streams which successfully aligned the independent sequences appropriately. Later CNN-LSTM models were integrated into the HMM streams which created a hybrid CNN-LSTM-HMM system. This, in turn, combines the advantages of deep learning for extracting features alongside utilising a probabilistic approach. The researchers use this hybrid model to learn and extract features from weak labels and noisy data. By applying this method, they get 26.5% WER on the first stream, 25.4% WER on the second stream and 24.1% WER on the third stream . Lastly, they have a 73.4% accuracy on their designed system where a previously published result on the same dataset are 62.8%. Therefore, despite being one of the earlier models, the hybrid CNN-LSTM-HMM system evidently addresses the challenges of weak labels and noisy data in continuous sign language recognition effectively.

Later Yin and Read [40] emerges with a breakthrough when they contradict with the notion that gloss-based modelling is pivotal to achieve remarkable outcomes in overall SLR translation. While this strategy has been effective in some of the papers we have investigated as well, Yin and Read [40] state that a reliance on gloss-based systems present a fundamental problem since ground truth glosses are sub-optimal for capturing the nuances of sign language. The problem lies in the assumption that GT glosses provide an ideal intermediate representation for sign language, which may overlook the complexities of the language. The lack of universal standard for glosses and imprecise representation of sign language leads to a bottleneck for the multidimensional stream of data that sign language possesses. This issue is addressed through the introduction of the STMC-Transformer model for video-to-text translation, a novel solution aimed at enhancing translation accuracy in SLT systems. They propose the use of a Spatial-Temporal Multi-Cue (STMC) network with a self-contained pose estimation branch that breaks down the input video into spatial features of multiple visual cues. Then, it identifies temporal correlation between cues within signs and beyond it. Leveraging recent advancements in Neural Machine Translation (NMT), particularly a two-layered Transformer maximising the log-likelihood, their technology facilitates more accurate translation by better preserving the intricacies of sign language expressions. Through extensive experiments on the PHOENIX-Weather 2014T and ASLG-PC12

datasets, Yin and Read [40] demonstrate that their STMC-Transformer model outperforms both recurrent networks and Transformer models translating GT glosses, yielding a substantial improvement in translation accuracy. The model achieves significant improvements, surpassing the previous state-of-the-art by over 5 and 7 BLEU scores on gloss-to-text and video-to-text translation tasks, respectively, for the PHOENIX-Weather dataset, and over 16 BLEU on the ASLG-PC12 corpus. Notably, the STMC-Transformer's video-to-text translation outperforms the translation of ground truth (GT) glosses, challenging the assumption that GT gloss translation is the upper bound for SLT performance. This suggests that glosses may be an inefficient representation of sign language. For future work, the authors recommend focusing on end-to-end training of recognition and translation models or experimenting with alternative sign language annotation schemes. Overall, the study establishes a new benchmark for SLT and opens new avenues for further research and development.

Continuing with further developments on this model, the translation subsystem has undergone a transformative performance upgrade. Deep Learning has played a pivotal role behind this achievement. DL models are often more popular because they focus on the stronger features to achieve quick convergence. However, certain visual cues may be overlooked in the process. This is an important limitation because the comprehensive analysis of continuous sign language requires addressing the manual (hand shape) and non-manual (facial expressions, upper body posture) features simultaneously ([21], as cited in [41]). As such, Zhou, Zhou, Zhou, *et al.* [41] introduces a Spatial MultiCue (SMC) and a Temporal MultiCue (TMC) module dedicated to spatial representations and temporal correlations respectively. Combined, they contribute as part of the Spatial-Temporal MultiCue (STMC) model to achieve a high performing CSLR model. Firstly, the SMC module processes each video frame to extract spatial features from cues (full-frame, hand, face, pose). This involves pose estimation followed by patch cropping to eventually generate feature sequences of cues. The TMC module utilizes these to integrate information about the unique features of each cue (intra-cue path) with learning about the combination of different cues at different timings (inter-cue path) [41]. The output from TMC modelling is then passed to BiLSTM encoders and CTC layers. This achieves sequence learning and inference. Additionally, a joint optimization strategy is employed to ensure that STMC network sequence learning is end-to-end. Zhou, Zhou, Zhou, *et al.* [41] implements this network architecture using PyTorch. The resultant framework has remarkable results with large-scale prominent CSLR datasets: PHOENIX-2014, PHOENIX-2014-T, and CSL. It records state-of-the-art performance on all three datasets, achieving a 20.7% WER on PHOENIX-2014 while outperforming its best competitor on CSL by 4.1% on WER [41]. It also manages to have a 17.6% and 5.3% edge over the recent multi-cue models LS-HAN and CNN-LSTM-HMM respectively. Extensive experimentation on the PHOENIX-2014 dataset also reveals that SMC can achieve a 3% improvement on this test set in comparison to the assumed baseline model. However, the unsupervised TMC module showed no such progress over 1D-CNN. Only when CTC loss helps its intra-cue path learn about the temporal dependency amongst cues, there is an improvement of 1.6% and 1.7% upon this development and test sets respectively. The STMC network itself manages to reduce WER by 4.8%. Alongside, the self-contained pose estimation branch plays

a crucial role in regularisation and preventing overfitting. It reduces inference time by 44% which is an upgrade even upon the performance of an off-the-shelf model. However, it is worth noting that the model struggled to distinguish sign gestures from upper-body pose features. It performed best when considering hand-gestures and full frames alongside different cues along the inter-cue path. In summary, the STMC model's innovative aggregation of spatial and temporal features into an integrated architecture significantly enhances CSLR performance and efficiency.

In further attempts to improve CSLR accuracy, Min, Hao, Chai, *et al.* [28] focuses upon the iterative training scheme to address the prominent issue of overfitting. They discover that for overfitting problems, sufficiently training the feature extractor is necessary. Hence, a Visual Alignment Constraint (VAC) is proposed. This packs two auxiliary losses, one focusing on visual features and the one enforcing prediction alignment between the feature extractor and the alignment module. Before starting to work on the model, CTC-based CSLR models were revisited, which led to an interesting observation: only a few frames played a key role in training. So constraining the feature space would be critical to train a CSLR model. The framework proposed, has three compartments: a feature extractor, an alignment module and an auxiliary classifier. The feature extractor applies 1D-CNN on the sorted out frame-wise features from the images to extracting the information and then propagates the features to the next compartment. Two auxiliary losses are chosen during training. Initially, the visual enhancement loss which primarily aligns visual features and the target sequences. Secondly, the visual alignment loss which aligns the short and long term context prediction through a method called knowledge distillation. To test the model, the RWTH-PHOENIX-Weather-2014 and CSL datatsets are used and alike previous researches we have explored so far, the prediction inconsistency is evaluated using WER. Min, Hao, Chai, *et al.* [28] performs the experiment in four parts. For the first experiment, the baseline model is used, resulting in Dev and Test WER(%) of 25.4 and 26.6 respectively. The second experiment using Baseline+VE model obtains Dev and Test WER(%) of 23.3 and 23.8 respectively. Following that, the third experiment consisting of the Baseline+VA model achieves Dev and Test WER(%) of 24.5 and 25.1 respectively. Finally, the proposed Baseline+VAC model gains an impressive Dev and Test WER(%) of 21.2 and 22.3 respectively. Moreover, comparisons made with the state-of-the-art models of the time such as STMC and FCN recorded WER(%) of 2.1, 3 respectively compared to the WER score of 1.6 achieved by the proposed model. To conclude, the VAC end-to-end model proposed addresses the overfitting problem, showing promising results and continuing improved CSLR outcomes.

Zuo and Mak [43] continue to address the challenges associated with DL-based models. However, their approach is driven by an incentive to enhance the consistency of CSLR backbones. They target the visual and sequential modules and contribute an auxiliary constraint to each. HRNet pre-trained on MPII is used to extract key points before it undergoes a post-processing stage for heatmap refinement. The inclusion of these keypoint heat maps allows the visual module to achieve spatial attention consistency (SAC) constraint. This spatial attention module builds on the concept proposed in CBAM by dynamically assigning weights to imply channel importance. This way, it offers an improvement in terms of feature-extraction ef-

fectiveness and fewer parameters over former CSLR models that use an off-the-shelf pose detector and a multi-stream architecture. The module works similar to attention mechanisms and focuses on informative regions. To further improve recognition accuracy, they build on previous models like Visual Alignment Constraint (VAC) and Self-Mutual Knowledge Distillation (SMKD) which portray the benefit of enforcing consistency between visual and sequential modules to improve inter-module cooperation. Since both output features from both these modules represent the same sentence, Zuo and Mak [43] introduce a sentence embedding consistency (SEC) constraint. The Sentence Embedding Extractor (SEE) is modeled based on QANet and it improves the overall feature representation power. It is also worth noting that the overall loss function of C²SLR is a combination of SAC and SEC loss alongside traditional CTC. The effectiveness of these changes are evident in the results from experiments on CSLR backbones like VTB (VGG11, TCN and BiLSTM), CT (CNN and TCN) and VLT (VGG11 and Local Transformer). C²SLR also achieves the revolutionary feat of being the first end-to-end method that outperforms models utilizing state optimization strategy. It records new state-of-the-art art performance on the PHOENIX-2014 dataset, surpassing STMC. It also achieves comparable performance on CSL against the state-of-the-art model MSeqGraph. With PHONEIX-2014-T, its consistent performance on the development and test sets, alongside it outperforming the state of the art performance on the latter, is evidence to its effectiveness in real-world scenarios for dealing with unseen data [43]. As such, this approach significantly enhances CSLR backbones by introducing constraints that achieve superior performance across various datasets.

More recent attempts at outperforming existing models like STMC have led to the emergence of attention-based CSLR models. Amongst these, SENet, CBAM, NLNet have yielded positive outcomes, not without their fair share of limitations. Eventually, Hu, Gao, Liu, *et al.* [14] implements a correlation network (CorrNet) that outperforms all three of these aforementioned models. Besides, they achieve better results at aggregating spatial-temporal information than techniques like I3D and TSM. They address the problem that the prevalent technology for capturing spatial features in CSLR models processes the video frames individually. As such, adjacent frame correlations and cross-frame signing trajectories get overlooked. Even if 3D CNN is used despite its compromised capacity to provide precise gloss boundaries, its success is limited to combining information within nearby video frames. The rigidity of the overall structure also means that it is not adequate for adapting to the role of finding the most important information from different videos. Using temporal convolutions for this task suffers from the same disadvantages. To improve this status quo, CorrNet focuses primarily on body trajectories. The network involves two modules. Initially the correlation module generates correlation maps to track the movement of spatial feature patches across frames. The size of this correlation map is varied depending on a need for semantic consistency or fetching information from far-apart frames. Alongside, the size of feature patches can be adjusted: reducing it to even one pixel depending on computational constraints. Later these correlation maps are used by the identification module to focus upon the most informative spatial regions (hands and face) and closely correlated spatial-temporal features dynamically [14]. However, this is achieved without any large 3D spatial-temporal kernel. Instead, multiple convolutions along parallel branches, each with

progressively increasing dilation rates, are employed. These help reduce computational costs and contribute to model capacity. Consequently, the branches undergo group convolutions to reduce parameter and computation requirements. Eventually, the branch-wise extracted features are regulated using a learnable coefficient, alpha, depending on importance and groupwise convolution of different branches to achieve feature extraction from multiple spatial-temporal neighbourhoods. During training, Hu, Gao, Liu, *et al.* [14] maintained the same settings as the former state-of-the-art models. Yet, CorrNet achieved new state-of-the-art accuracy on all four of their datasets: PHOENIX14, PHOENIX14-T, CSLDaily and CSL. It records an 18.8% and 19.4% WER on PHOENIX14 development and testing datasets respectively. Similarly, it achieves 18.9% and 20.5% WER on the corresponding datasets of the PHOENIX14-T dataset. It even outperforms normal convolution consisting of more parameters and computation during the ablative study on the development and testing sets of PHOENIX14 dataset. Alongside, it offers an advantage over previous models that relied on hand and facial features from multistream inputs, body keypoints and pre-extracted hand patches. For instance, unlike STMC it does not require an additional pose-estimation network. The need for extra supervision or training stages is also alleviated yet CorrNet can achieve end-to-end dynamic training of the model as before.

Meanwhile, Zuo, Wei, and Mak [44] has recently approached the limitations in the translation subsystem from a different perspective. They observed that the CTC based models usually took the entire signed video as input for making predictions. Therefore, they developed a system to shift away from offline recognition towards effectively conducting SLR online. They used the same datasets mentioned in the majority of the research discussed so far: Phoenix-2014 (P-2014), Phoenix 2014T (P-2014T) , and CSL-Daily. Firstly, they constructed a sign dictionary where a pre-trained CSLR model TwoStream-SLR was used. It segmented the input videos into isolated signs. Consequently, DTW (Dynamic Time Warping) was employed to identify the signs by aligning the frames with the gloss labels. Since the segmented signs were not consistent in terms of accuracy, augmented signs were incorporated to enhance the training data. Secondly, they used ISLR models for encoding the sign videos. Additionally, gloss sampling was utilised to sample the glosses first and then multiple instances of the glosses were selected to form batches. In order to improve overall sign prediction, two types of cross-entropy loss functions (instance level and gloss level) played an important role. A saliency loss was also incorporated to focus on only the frames with signs. , he went into detecting signs in real time by using a sliding window approach. In addition, He used a voting system to remove any duplicate signs and background noise. Thus, the predictions were better filtered. Lastly, he added a gloss to the text network which translated the signs into actual text phrases or sentences in real time. By implementing his architecture he got 70/90% (dev/test) accuracy in recognizing the isolated signs. The online system here can, in turn, boost the offline models by aligning the dimensions of their features. In summary, this innovative approach not only enhances real-time sign language recognition but also improves the accuracy and efficiency of offline models.

2.3.4 Multilinguality

Maalej [24] addresses it as a fallacy that there is a universal sign language. It is also widely acclaimed that SLR models can only be designed specific to individual languages. This is because of the noticeable differences in dialect, signing and other nuances that make sign languages in specific regions develop with unique characteristics.

Hu, Pu, Zhou, *et al.* [13] challenges this concept by tackling the problem of multilingual sign language recognition using a unified framework. Their system used a shared combination of technologies including visual encoder and several multiple language dependent sequential modules. This contributed to temporal learning. A probabilistic decoding method was subsequently used to align the sign videos with the sign words. For the datasets, they use PHOENIX-2014, CSL and GSL-SD (which is a Greek sign language dataset). Firstly, the features are extracted from the videos using the shared encoder for all the languages. Each language is then allocated a separate sequential models. Therefore, each language gets its own model to learn the features and then verify them as sign words. Eventually, CNN-TCN is used as an additional shared layer to extract the features and before passing it to LSTM to do sequence modelling. Furthermore, the output is sent through a softmax layer where probabilities and CTC is utilised for training. Hu, Pu, Zhou, *et al.* [13] tests their framework for a case consisting of two languages and another consisting of three languages. In case 1 (recognizing two languages), the model achieved a BLEU score of 0.840 and a WER of 0.191, outperforming state-of-the-art models, which reported BLEU scores ranging from 0.332 to 0.795 and WERs from 0.245 to 0.757. For case 2 (recognizing three languages), the model attained a BLEU score of 0.852 and a WER of 0.181.R.

2.3.5 Dataset

Majority of the significant research that we have covered so far uses one or more of the datasets mentioned below. All of these large public datasets have been instrumental in advancing recent progress in continuous sign language recognition. Hence, we have summarised their key dataset characteristics below.

RWTH-PHOENIX is German weather forecast recordings involving nine actors. It is composed of 1295 hand gestures which make up 6841 sentences. It is divided into 5672 training, 629 testing, and 540 development samples. PHOENIX2014 is very similar but it features the nine signers against a clear background with a resolution of 210×260. Its sign language data has a specific structure and annotation, making it more suitable for sign language recognition tasks. To summarise, RWTH-PHOENIX is a broader term that may refer to the initial corpus provided by RWTH Aachen University while PHOENIX2014 is a specific version of the dataset. The latter also has a training-development-testing split of 5672-540-629. PHOENIX2014-T comes as an extension of PHOENIX2014, providing additional data and annotations to support both CSLR and translation research. Consisting of 1085 gestures and 8247 sentences, it is further divided into 7096 training, 519 development, and 642 testing instances. CSL, a Chinese Sign Language dataset, has 50 signers representing a

vocabulary size of 178 signs across 100 sentences gathered in a laboratory environment. The dataset contains 25000 videos, all of which are split into a 8:2 train-test split ratio, gathered in a laboratory environment. In contrast, CSLDaily reflects more natural daily activities. Videos are recorded indoors at 30 frames per second by 10 signers and the dataset includes 20654 sentences, with 18401 training, 1077 development, and 1176 testing samples.

However, as Manoharan and Roy [25] has mentioned there is not enough continuous datasets available for research. They state how several models perform well with these training datasets but suffer from a case of overfitting and fail to perform according to expectations on unseen data. Besides, Min, Hao, Chai, *et al.* [28] state that most CSLR datasets consist of sentence-level annotations only, owing to frame-level annotations being an expensive alternative. Thus, the current progress in this domain is hindered by the lack of high-quality datasets.

Hence, several research groups are now increasingly more concerned with improving the dataset in this field of research. Kim and O'Neill-Brown [20] introduced us to the idea of synthetic data, which is the mimic of real world-data created with the help of various algorithms. Their approach utilises a baseline prototype ASLR based on DeepHands using the Kinect Sensor and a graphical user interface. The dataset involved is the Sign Language Lexicon Video Dataset (ASLLVD), comprising of 10,000 ASL signs by 6 native ASL signers and human-annotated linguistic information such as gloss and handshape labels. For feature extraction, they use OpenPose which provides the different models pre-trained on publicly available datasets. Additionally, 25-point body pose and 20-point hand detection models are involved. During preprocessing, two main types of features are extracted from all frames: 2D coordinate of the hand followed by the DeepHand hand-shape. The latter is created using a CNN-based sign language recognizer trained on approximately one million images. Later K-means clustering is used for classification. Consequently, for Data Augmentation, several manipulation strategies such as adding noise, changing brightness change, rotation and zoom were considered. However, for the scope of the proposed experiment, only Rotate and Zoom is used. The experiment is carried out in several stages using varying amounts of synthetic data (0%, 100%, 300%, 500% and 1000%) and accuracy is used as the primary evaluation metric. The best performance was recorded with 1000% synthetic data with K-means cluster size of 3000. However, interestingly, the relationship between performance and synthetic data quantity appears non-linear. Performance dropped significantly until 300% but showed significant improvement when 1000% was surpassed. Overall, the finding of the research was that with careful data augmentation, we can significantly enhance SLR recognition models.

More recently, Kezar, Thomason, and Sehyr [19] address this crucial problem primarily for ISR. We have previously discussed in page 8 the significance of phonology for accurate ISLR in reference to another paper co-authored by Kezar, Pontecorvo, Daniels, *et al.* [18]. In this paper, that relevance is reestablished by stating that models considering signs as a sequence of linguistic components improve ISLR by up to 6%. As such, they introduce a new dataset, the Sem-Lex Benchmark, comprising 84,568 videos of isolated sign production, the largest of its kind under the

status quo. The objective of this dataset is to cater to the growing number of researchers who no longer treat SLR as an entirely vision-based problem. Alongside its data size, the dataset boasts using ethical and consistent sourcing. Deaf fluent signers are involved to counter heterogeneity and inconsistencies in sign articulation of videos. Additionally, a final inter-rater reliability threshold of only 0.7 was used for selecting labellers. However, its most prominent contribution is adding representative and high-quality data amidst a profound absence of well-annotated datasets.This is achieved by incorporating linguistic information accompanying each sign alongside a new annotation scheme. They rely on videos of ASL signs to label ASL signs themselves Kezar, Thomason, and Sehyr [19]. Utilising the lexical databases in this way allowed their new resultant dataset to have sign labels aligned with other available linguistic resources: ASL-LEX, ASL Citizen, and ASL SignBank. In turn, phonological descriptions that are recorded in ASL-LEX can now be utilised as phonological information of ISR dataset without manually annotating it. Similarly, the cross-compatibility of this dataset with ASL SignBank has enabled labelling of continuous signing videos. However, this benchmark is yet to incorporate grammatical features as well as coarticulation. Simultaneous phonological feature and gloss recognition has already shown to improve few-shot ISR accuracy by 6% and overall ISR accuracy by 2% respectively Kezar, Thomason, and Sehyr [19]. Further inclusion of more aspects of ASL to address the current limitations of this dataset is expected to transform it into a prominent driving force behind the future research on CSLR.

Chapter 3

Work Plan

The work plan for our thesis consists of three main phases, starting with Pre-Thesis 1 and then Pre-Thesis 2, and finally the defense. Pre-Thesis 1 lasted for four months which involved member selection and team building, selecting a topic and supervisor, writing an abstract, reading relevant papers and summarising them into a literature review, and finally writing a report. Secondly, pre-thesis 2 will also likely last for four months and it involves collecting and analyzing data, building and evaluating models, and also writing a report and a poster. The final phase of the thesis will involve analyzing the results, completing the paper, and preparing for the final defense. The Gantt chart [3.1] visually represents the time frame and level of completion for each individual task within the overall work plan. The time frame has also been color coded to signify the three different phases, and we have used green to showcase that the first phase is over, whereas the second phase and final phase have yet to be completed.

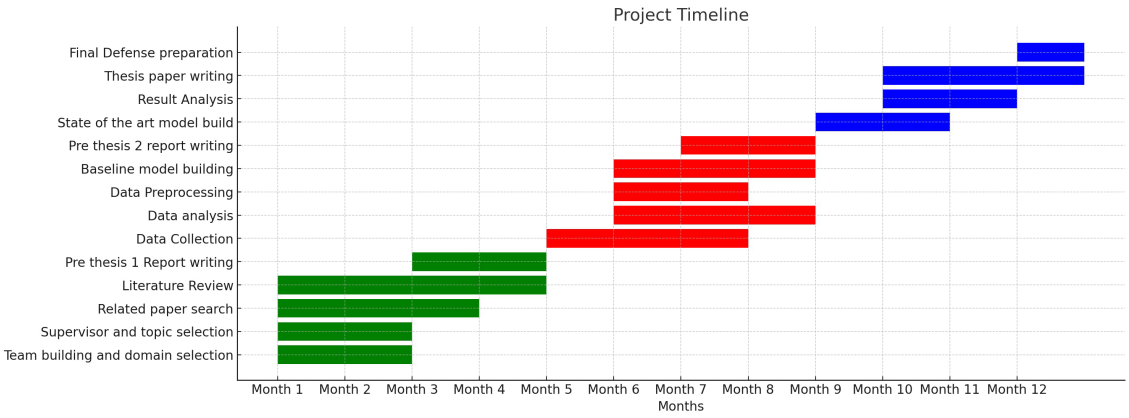


Figure 3.1: Workplan Gantt Chart

Chapter 4

Dataset

4.1 Dataset Analysis

The PHOENIX 2014-T dataset is a large-scale compilation of German weather forecast recordings, specifically assembled for tasks related to CSLR and Sign Language Translation Camgöz, Hadfield, Koller, *et al.* [3]. The dataset comprises weather forecast recordings of 9 different signers using German Sign Language (deutsche Gebärdensprache, DGS). In total, it includes 8,257 sentence-level recordings, along with glosses and sentence-translations. Table 4.1 provides the volume of data into perspective.

Set Type	**No. of Sentences**	**No. of Glosses**	**No. of Signs**
Training	7,096	1,085	-
Development	519	-	-
Test	642	-	-
Total	**8,257**	**1,085**	**1,295**

Table 4.1: Dataset statistics: Summary of distribution for the entire dataset

Our motivation behind choosing this is that it is widely regarded as a benchmark for CSLR and SLT research. Our previous findings have highlighted that research in this field has long been bottlenecked by datasets lacking sufficient size, vocabulary variation, or signing complexity. However, the PHOENIX 2014-T dataset has a comparatively larger training set that maximizes the learning capability of models across a diverse range of signing styles. The structured division into training, development, and test sets further facilitates robust model training and evaluation. Additionally, since the dataset is based on real-world weather forecasts, its practical nature allows it to be leveraged for substantial research and applied to a variety of real-world scenarios, broadening its potential for learning and improving sign language technologies. It should be noted, however, that the test set is smaller in contrast, indicating the necessity of distinct validation procedures to eradicate overfitting.
In addition, this dataset contains gloss annotation. Glossing involves associating each sign to a corresponding gloss, which is of significance because it provides a linear representation of sign language. The annotations present a sentence-level translation corresponding to a signer's gestures with respect to words, sentences in a tokenized representation. Both training and evaluation of different CSLR models

based on this dataset depend upon these annotations. These are recorded in csv files containing 5 columns:

- **Name**: A unique identifier for each recording.
- **Video**: The file path of the recording corresponding to the sign language translation.
- **Start**: The starting timestamp of the video segment.
- **End**: The ending timestamp of the video segment.
- **Speaker**: The signer identifier; each signer is numbered (e.g., Signer01, Signer02, etc.). There are 9 signers in total in the dataset.
- **Orth**: The direct German translation of the sign, which may or may not follow the spoken language structure. This is referred to as the *gloss sequence*. Each row corresponds to a collection of glosses signed in sequence across the frames.
- **Translation**: The natural German translation in spoken format of the direct translation.

To get a better insight into our training dataset, we have separate folders, each containing images of consecutive frames corresponding to original video segments. These frames, each of size 210 by 260 pixels, have been collected by sampling the videos at 25 frames per second. For instance:
Figure 4.1 shows the 127 frames in the folder labeled "14August_2009_Friday_tagesschau-64". The training corpus contains a row identified by this label, which bears the gloss annotation for this gloss sequence: *NORD WOLKE IX ZWISCHEN SCHOTTLAND* (English: *NORTH CLOUD IX BETWEEN SCOTLAND*). It also includes the German translation for this gloss sequence: "der Norden wird allerdings von den Wolken eines Tiefs bei Schottland gestreift" (English: *However, the north is touched by the clouds of a depression near Scotland*).
It is also interesting to note that an additional training corpus with more detailed and complex annotations has been provided. For instance, for the same folder mentioned above, the gloss sequence in this CSV file is *loc-NORD ___EMP___ WOLKE IX ZWISCHEN SCHOTTLAND*. Instead of just *NORD* (English: *NORTH*), this new annotation, with *loc* preceding *NORD*, indicates a location marker, specifying that the next part of the gloss relates to a specific geographical area. Alongside this, the extra *EMP* serves as a marker for emphasis, indicating that the signer is stressing or emphasizing the word that follows. This additional information collectively contributes to a further extensive understanding of the nuances of sign language, which, unlike text-based languages, is non-linear and multi-dimensional.
Further investigations into the PHOENIX 2014-T dataset reveal several important aspects. The average video length with gestures is 116 frames, meaning the dataset uses an average of 116 frames to represent each sentence. Since the dataset consists of individual frames rather than raw videos, this resolution allows for detailed analysis. The average number of frames representing a single gloss, or sign for a particular word, is 15, indicating a high level of detail per gloss. Additionally, the "Orth" column, which contains unordered direct translations, averages 51.33 words

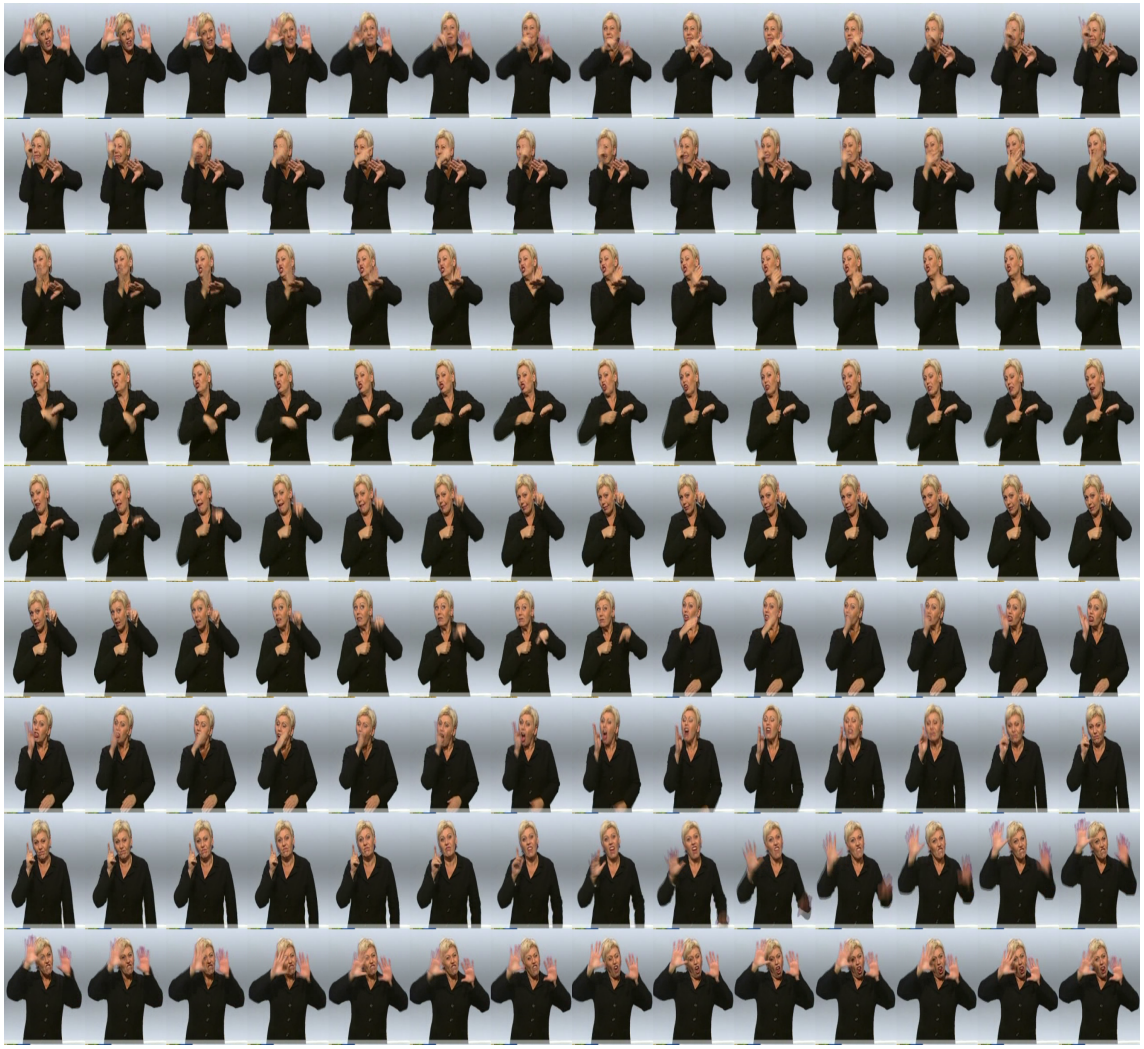


Figure 4.1: Example of frames from a video segment

per sentence. Each word in the dataset is represented by approximately 15.52 frames on average, demonstrating the extensive frame usage for word representation.The average gloss-to-word ratio of 0.58 highlights the constructional variations between sign language and spoken language, indicating that there are generally fewer glosses than words. This means that, on average, each word in the spoken language translation corresponds to less than one gloss in sign language. The KDE plot in Figure 4.2 illustrates the distribution of frames per gloss annotation, with an average of approximately 120 frames per gloss. Figure 4.5 presents the top 50 most frequently occurring glosses in the dataset.

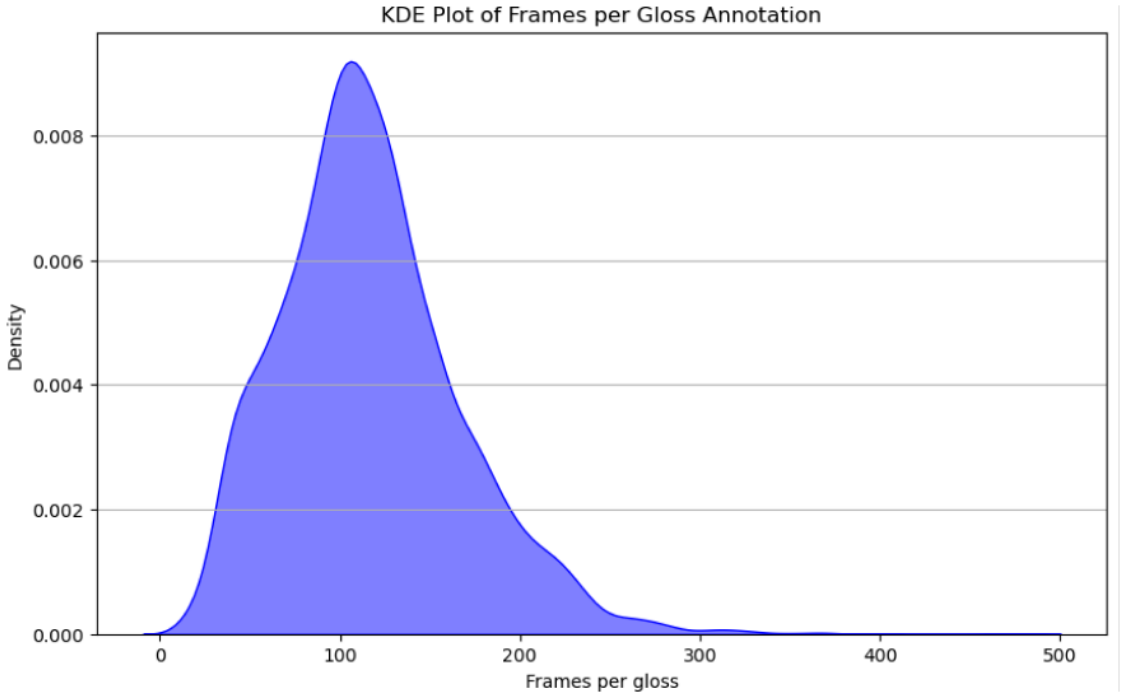


Figure 4.2: KDE Plot of Frames per Gloss Annotation

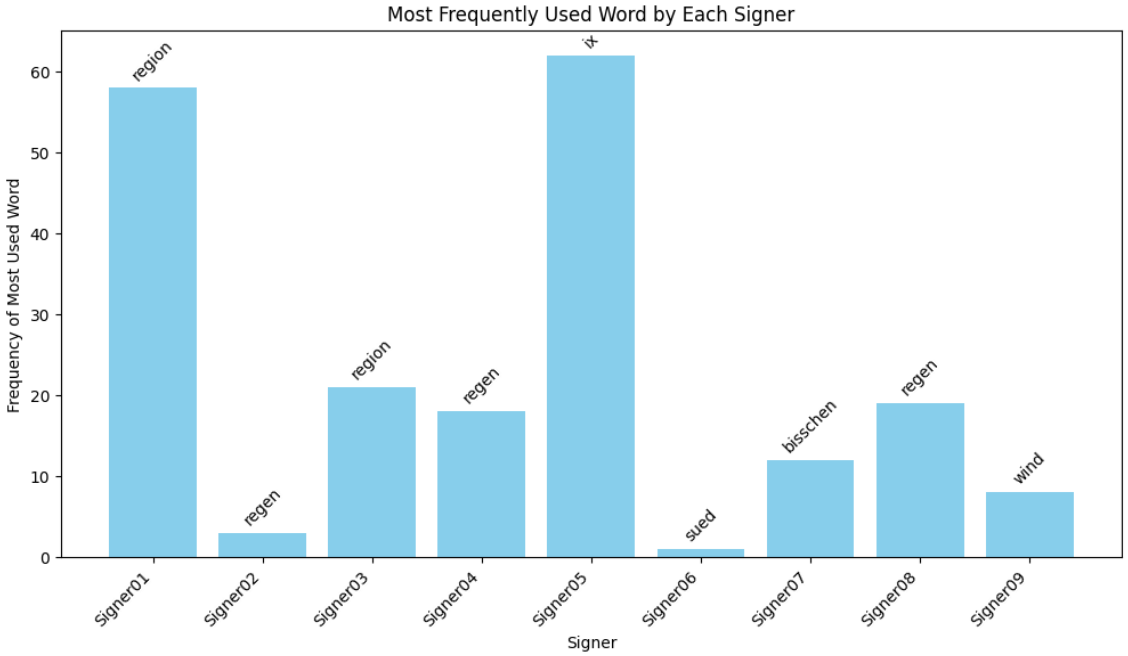


Figure 4.3: Bar chart showing each signer's most frequently used word and its usage count.

The optical flow visualization shown below represents the motion magnitude between two consecutive frames of a video sequence during a sign language gesture. This optical flow analysis can help identify the dynamic manual markers used in sign language while also highlighting the regions of the body that exhibit subtle motion related to non-manual markers. By observing the differences in movement intensity,

we can differentiate between key signing actions (high motion) and supportive non-manual elements (low motion).

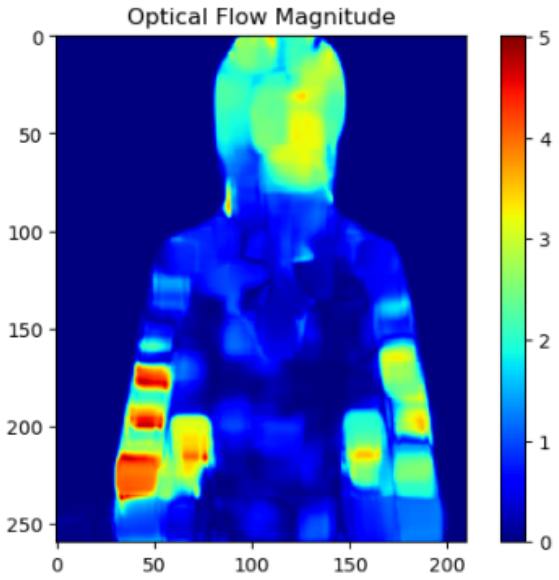


Figure 4.4: Optical flow visualization representing motion magnitude between consecutive frames.

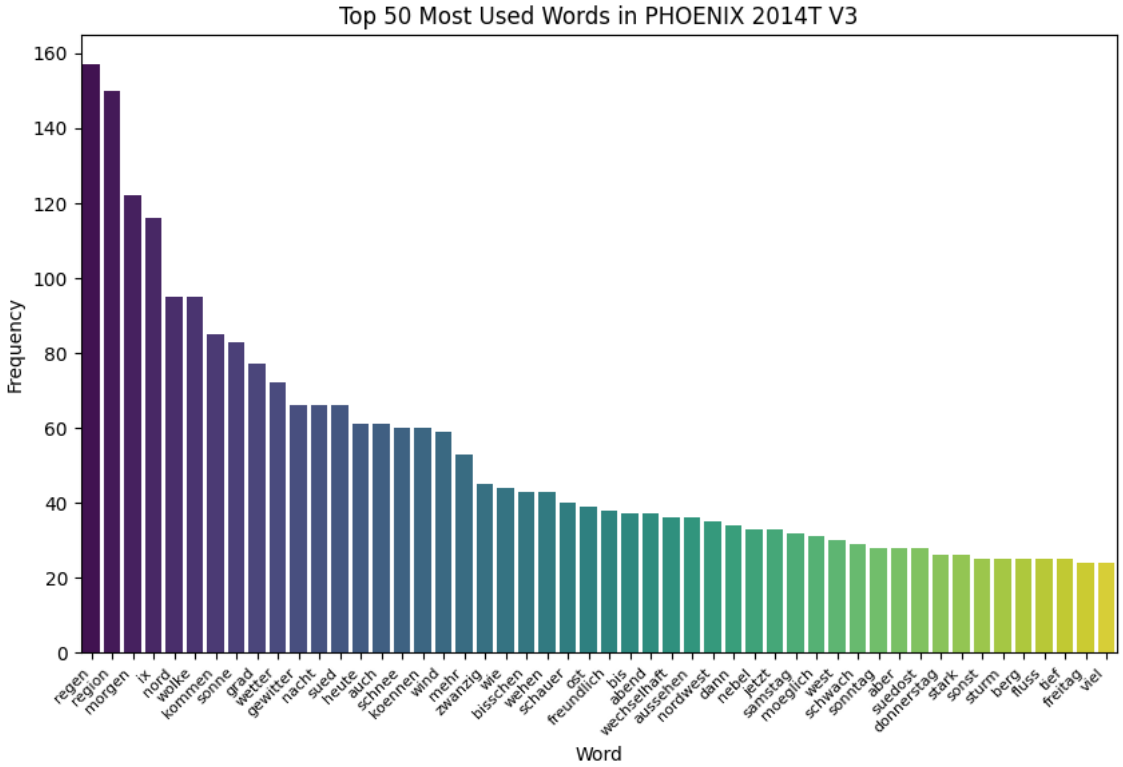


Figure 4.5: Bar chart of the top 50 words in the PHOENIX 2014T "orth" column, sorted by frequency.

4.2 Data Pre-processing

Since multi-modality is a pivotal aspect of sign language, we have decided to focus beyond hand shapes and motion for our baseline model. We have utilized Mediapipe, a framework developed by Google, which has pre-trained models that can achieve remarkable performances efficiently. For our model, we employed the `mp_pose` module for detecting body joints (e.g., shoulders, elbows, etc.). Alongside, the `mp_face_mesh` module is used to detect facial landmarks (e.g., eyes, mouth, etc.) in the form of 468 key points. Moreover, `mp_hands` detects the hand landmarks: 21 key points on each hand.
Each of these modules runs on every image frame of the training dataset, one folder at a time. However, the frame images are processed independently instead of being considered as part of a video sequence since the `static_image_mode` parameter of Mediapipe models is initially set to True. Eventually, for each processed image, all the detected landmarks are drawn upon the original image using different colors and thickness so the distinct categories of landmarks can still be identified. Afterwards, these processed images are converted back to RGB format and stored in a list that contains all the frame images corresponding to an individual folder of the training dataset.

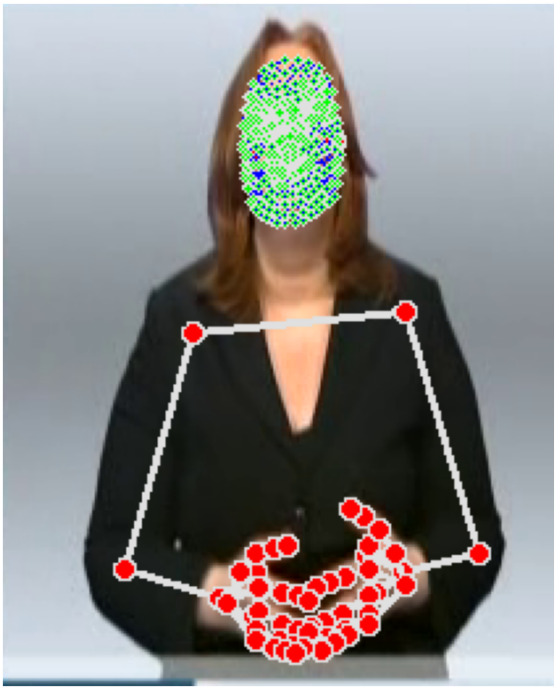


Figure 4.6: MediaPipe detected landmarks including body joints, facial key points, and hand landmarks overlaid on an image.

Chapter 5

Technology Overview

5.1 ResNet50

To efficiently extract features from image frames, we employed the pretrained ResNet50 model,a state-of-the-art deep learning architecture. After extracting key points from the image frames using MediaPipe, we move on to training our neural network. The feature extraction process is made more efficient by employing ResNet50, which is renowned for achieving an error rate as low as 22.85%.
At the core of its architecture, ResNet50 uses deep convolutional neural networks (CNNs) with skip connections, also known as residual connections. These connections allow the model to learn deeper architectures without encountering the vanishing gradient problem. By enabling data to flow directly from the input to the output, bypassing certain layers, the model does not need to learn the entire mapping from scratch.
For our baseline model, we fine-tuned the layers of ResNet50 and incorporated different pooling techniques, along with padding, to optimize performance. The process can be summarized as follows:
Initially, the convolutional layer accepts 3 input channels (corresponding to RGB images) and outputs 64 feature maps using a 7×7 kernel with a stride of 2. To prevent excessive shrinking of the output size, padding of 3 is applied. Importantly, no bias term is used in this layer, as the subsequent batch normalization handles any shifts in the data distribution by normalizing the activations. This normalization step helps the model converge faster during training.
Next, an activation function, ReLU (Rectified Linear Unit), is implemented for the purpose of non-linearity by setting all negative values to zero, allowing the model to capture non-linear patterns in the data. Following this, a max pooling layer reduces the spatial resolution, decreasing computational complexity while retaining essential features.
A key feature of ResNet50 is its use of residual blocks, which consist of three convolutional layers arranged in a bottleneck structure. These layers include a 1×1 convolution to reduce the number of channels, acting as a compression step that facilitates faster computations in subsequent layers. This structure is repeated across layers 2, 3, and 4.
Finally, each feature map is reduced to a single layer by leveraging an average pooling layer. As a result, a 1×1 feature map per channel is produced. The output from this pooling layer is inserted into a fully connected layer. A sigmoid activation function

resides there to perform image classification.
This approach, leveraging the power of ResNet50's residual connections and bottleneck structure, enables efficient feature extraction and accurate image classification.

5.2 Connectionist Temporal Classification (CTC)

One of the biggest challenges in our dataset was that we were given N frames of pictures with M glosses, where the glosses corresponding to each picture were completely unknown to us. For example, in the training dataset, the folder named *"01April_2010_Thursday_heute-6694"* contains a total of 53 images of consecutive video frames. The training corpus provided the corresponding label for all of these images as a gloss sequence *"LIEB ZUSCHAUER ABEND."* However, it was unclear which specific frames corresponded to which words in the label, making alignment a significant problem due to the absence of ground truth.
To overcome this problem, we deployed an algorithm known as CTC, which is commonly used for training deep neural networks with sequential problems where no explicit information about the connection between the input and output is given. The CTC algorithm is mainly divided into three parts.
Firstly, the alignment algorithm merges all the duplicating characters into a single character and a special token, known as the blank symbol (ϵ). This helps in mapping sequences that may vary in length between the input (frames) and output (glosses). Secondly, CTC loss is determined by calculating the probability of generating the output sequence. A modification is made to the input sequence to include blank symbols between each pair of characters to provide the flexibility to align the input sequence with the output sequence. Lastly, the inference module is employed to decode the sequence.
A simple approach of greedy decoding, where the most probable output token is selected at each time step without considering the overall sequence of outputs, could address this task. However, this naive approach does not treat each time step independently and can result in multiple possible alignments collapsing into the same output sequence. As such, we employ beam search, a more advanced decoding algorithm, to keep track of the multiple possible output sequences. This contributes to improving the model's alignment sequence, which is crucial for our baseline model since alignment variability plays a significant role in determining the best output. Hence, after we obtain the classification from the fully connected layer of ResNet50, we apply the CTC algorithm to compute the CTC loss, which is then fed back to the model at the beginning of each epoch. This process allows the model to iteratively learn the correct alignment between image sequences and gloss labels, improving its ability to predict the correct gloss sequence for each set of frames.

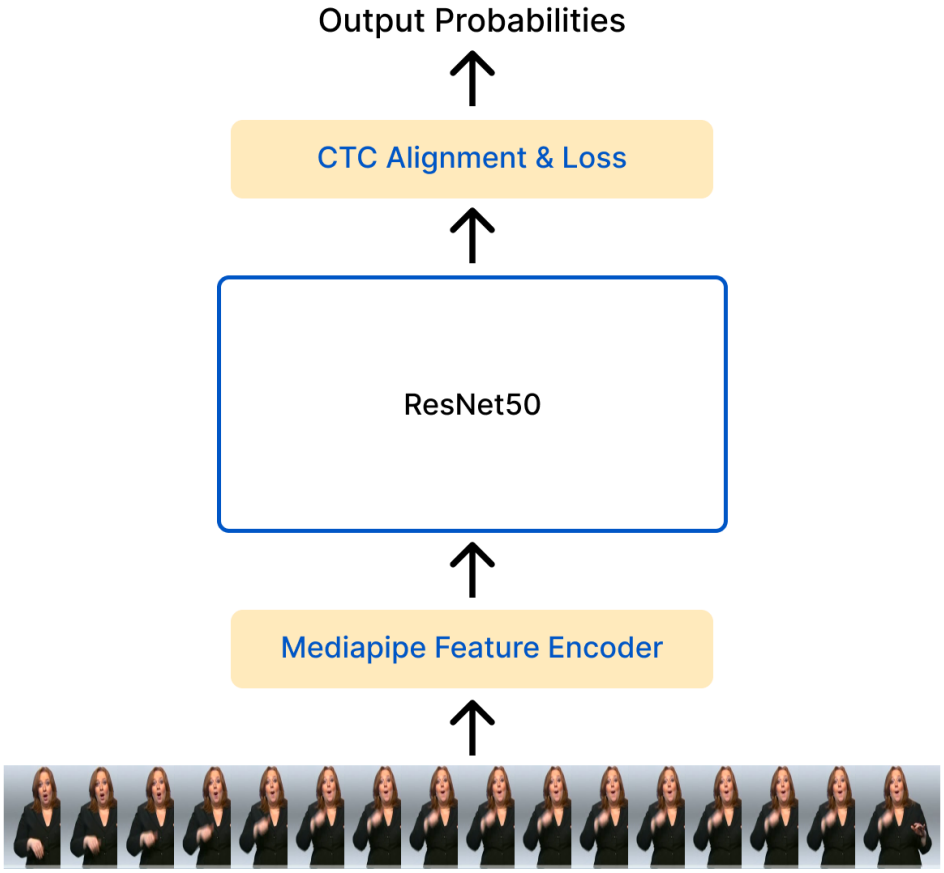
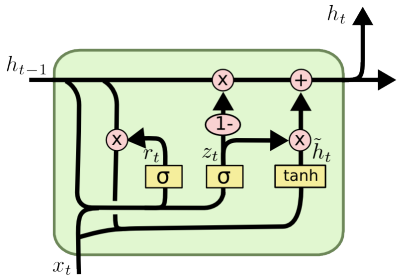


Figure 5.1: Connectionist Temporal Classification (CTC) architecture for sign language translation.

5.3 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a widely used recurrent neural network (RNN) designed to capture long-term dependencies between contexts in sequential data, which traditional RNNs struggle to retain. It achieves this by utilizing a memory cell controlled by three functional gates.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 5.2: LSTM Architecture [32]

The working mechanism of these gates is almost identical to those of a Bidirectional LSTM (Bi-LSTM). There are three sigmoid-based gates, each providing a specific

function:

- **Input Gate**: Balances the volume of data to be retained from the previous layer. The closer the value of this gate is to 1, the more important the current information is deemed.
- **Forget Gate**: Filters the most relevant information by excluding unnecessary details from the data. The closer the value of this gate is to 1, the less important the information is considered, resulting in less retention.
- **Output Gate**: Determines which information about the current layer will be passed on to the next layer for learning. This ensures that only the most important parts are carried forward in the network for optimal learning.

By strategically regulating information flow using these gates, the notorious vanishing gradient problem gets solved by LSTMs and learning enhances in long sequence of data.

5.4 Video Vision Transformer (ViViT)

Video Visual Transformer (ViViT) is another architecture that can be leveraged for feature extraction. It is an extension of ViT, which was initially modified to extract spatio-temporal features directly from videos [7]. ViViT relies mainly on two video tokenization methods: Uniform Frame Sampling and Tubelet Embedding [1]. In Uniform Frame Sampling, the tokenization is similar to ViT in the sense that once uniform frames are sampled from the video, they are split into non-overlapping patches and embedded into a sequence of tokens like before [1]. On the other hand, in Tubelet Embedding, spatio-temporal features like patch spanning times, height, width, etc. (called tubelets) are extracted. These are then embedded in 3D to retain both spatial and temporal information. The two-stage factorised encoder separates spatial and temporal processing. First it encodes frame-wise features before aggregating temporal information. ViViT factorizes self-attention across spatial and temporal dimensions to reduce computation overhead and uphold long-range dependencies in video sequences. Moreover, it leverages pretrained image models to get superior generalization on small datasets by utilizing regularization techniques like stochastic depth and data augmentation. This purely transformer-based design enables more meaningful and enhanced feature extraction and temporal reasoning. The spatial module in ViViT captures information from spatial dimensions such as edges, shapes, etc., to deduce the object each frame is representing. Meanwhile, the temporal module captures the trajectory of the object using optical flow to determine the direction of movement.

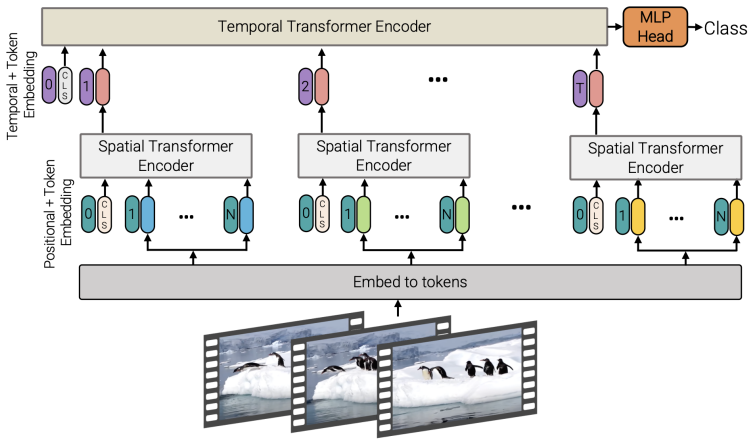


Figure 5.3: A two-stage Factorised encoder : the first encoder captures spatial interactions per time step, while the second models temporal dependencies, enabling late fusion of spatial and temporal information by [1].

5.5 TimeSformer

TimeSformer is essentially an efficient transformer-based model that was built for classifying video. It has a functioning space-time attention module that is divided to balance both speed and accuracy [2].
First, an RGB video is fed as input into the model, where each frame is split into a number of non-overlapping patches. Upon flattening the patches into vectors, they are then linearly embedded to form feature vectors. Layer normalization (LN) is applied. Then, similar to ViT, each patch is assigned a Q (query), K (key), and V (value) for computing attention in many layers for refinement.
Two main types of attention modules are involved here: full spatio-temporal attention and divided space-time attention. In the full spatio-temporal attention, efficient temporal information is captured because each patch attends to all other patches across space and time. However, this makes the mechanism more expensive. On the other hand, in the divided space-time attention, first each patch attends to the same spatial location across different frames to capture temporal attention. Next, attention is applied inside each frame to capture spatial attention. This helps to reduce the overall computational complexity, thereby making the attention mechanism more efficient. Other types of attention mechanisms, such as the sparse local-global mechanism, compute attention in small regions and then use a stride-based approach to compute attention globally. There is also the axial mechanism, where attention is applied individually over time, width, and height.

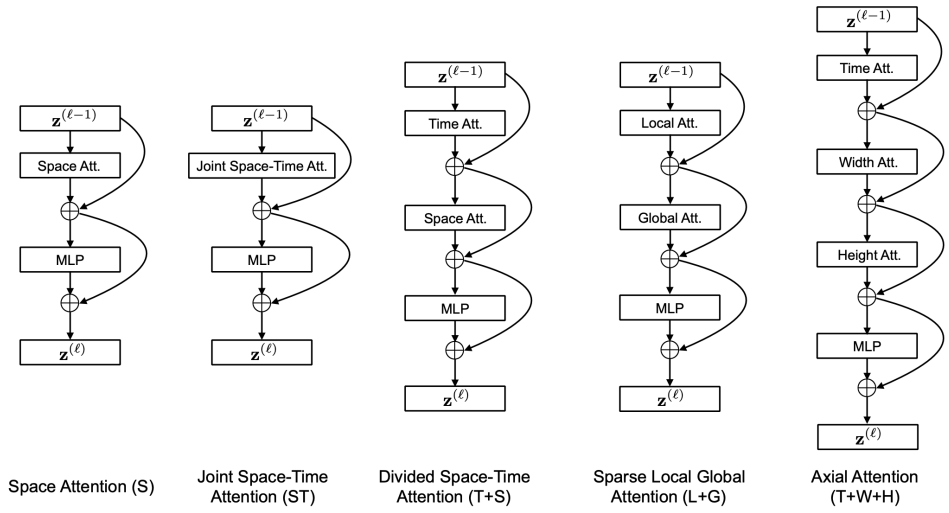


Figure 5.4: The different attention mechanisms of TimeSformer [2].

Finally, to summarize the whole video, a unique classification token is used, and the final feature vector is passed through an MLP to classify the input video.

5.6 Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (BERT) is a pre-trained, transformer-based neural network that can learn, understand, and generate spoken language. Unlike a traditional transformer, which contains both encoder and decoder modules, BERT is based solely on an encoder module. This design allows BERT to focus on learning and understanding sequential input data rather than generating human-like output. Its architecture enables it to learn text bidirectionally, allowing for more refined learning. Instead of analyzing text from left to right or vice versa, it processes text from both directions simultaneously to better understand the context. BERT's architecture comprises a multilayer bidirectional transformer encoder that employs self-attention. First, it takes a classification token (CLS), followed by a text sequence as input [5]. The input is passed through the layers in the encoder stack, where each layer applies its own attention head and sends it to the feedforward networks, which then pass it to the next encoder layer [5]. This process is repeated across all encoder layers. The final output is a vector that can be used for various downstream tasks, such as classification and translation.

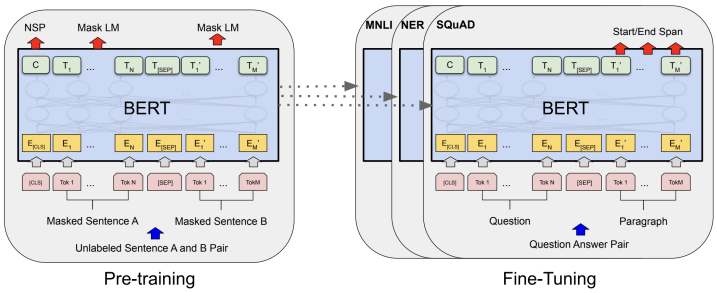


Figure 5.5: BERT pre-training and fine-tuning process by [5].

BERT's key working mechanism involves predicting masked words, which resembles filling in the gaps of a sentence. BERT adds a classification layer on top of the encoder output. The output from the classification layer is then multiplied by an embedding matrix to keep a coherent dimensions between the output and the vocabulary space. A probability for each masked word in the vocabulary is calculated using the Softmax activation function [5]. Finally, a loss function trains BERT based on the deviations between its predictions and the true values.

Chapter 6

Methodology

This paper investigates two distinct approaches to create CSLT pipelines to compare their effectiveness in the context of real-time translations.

6.1 Classification using Neural Network

The architecture and working mechanism of ResNet-50, which has already been discussed above, makes it optimal for classifying images. Hence, it is utilized in our attempts to classify sign language gloss. This paper uses a ResNet-50 model which was pre-trained on the ImageNet dataset. However, some preprocessing steps are needed. Firstly, individual frames are extracted from sign language videos to feed static images to the model, since ResNet-50 is designed to work on still images. Consequently, each of the frames is resized to 224×224 pixels to ensure maximum compatibility with ResNet-50's input dimensions. As a final step of preprocessing, the frames are converted to NumPy arrays, and their pixel values are scaled using the normalization metric of ImageNet. Frames are then processed in batches of 32 to make optimal use of memory and computational power.
In order to use the pre-trained ResNet model for feature extraction instead of direct classification, the fully connected classification head is removed, and the global average pooling is skipped. Eventually, a 2048-dimensional feature vector is generated for each frame. This contains the high-level spatial features of the signer's face, hands, and body posture, thereby serving as a form of encoding.
Next, a fully connected neural network (MLP classifier) is applied to map the extracted features into corresponding gloss categories. For each layer, batch normalization is also ensured to speed up the convergence rate and stabilize training. A consistent dropout rate is maintained to prevent overfitting by deactivating a percentage of neurons during the training phase. To minimize classification errors, a backpropagation method is used to adjust MLP weights. For regularization, dropout layers are deployed to ensure optimal generalization of the model to unseen sign language videos. The separate validation set of the dataset was used to evaluate the trained model.
To generate the outputs, the model uses a Softmax function to convert the logits from the MLP layer into probabilistic distributions. The model consists of:

- 1024 hidden units with ReLU, batch normalization, with a dropout rate of 50%.

- 512 hidden units with ReLU, batch normalization, with a dropout rate of 40%.
- 256 hidden units with ReLU, batch normalization, with a dropout rate of 40%.
- 128 hidden units with ReLU, batch normalization, with a dropout rate of 30%.
- 64 hidden units with ReLU, batch normalization, with a dropout rate of 20%.
- An output layer consisting of a Softmax layer with 1085 output classes (one class for each gloss).

Categorical cross-entropy is used as the loss function for this model.
The glosses generated by the final Softmax layer are connected to a Neural Machine Translation (NMT) model to generate the output sequences. This raises the chance of error propagation when converting from gloss to a sentence.

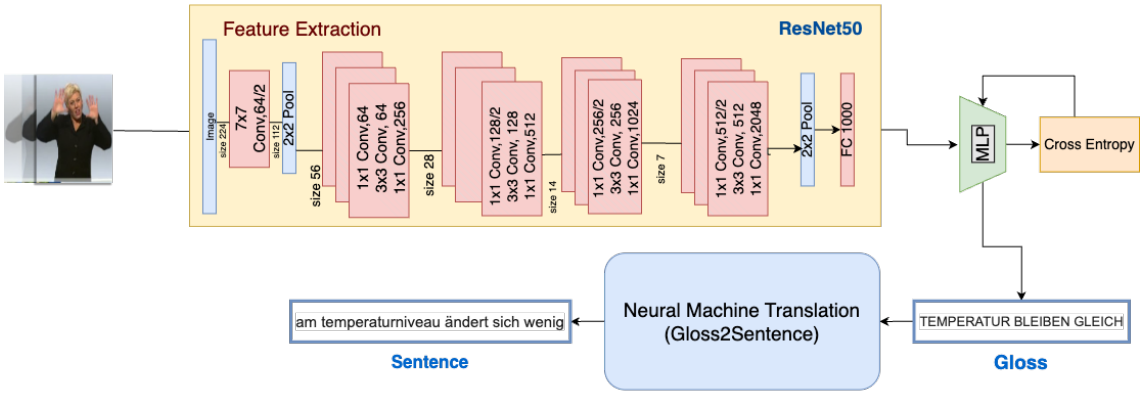


Figure 6.1: Illustration of the overall pipeline.

6.2 Generation Based model using Transformers

The original design of ViViT, aimed at video-based classification tasks, has been discussed above. However, the objective of this research is not within the paradigms of simple classification but rather translation: taking an input sequence of video frames and generating an output sequence of words transcribing the signs in it. This involves the additional complexity of sequential pattern recognition. Hence, the order of the input frame sequence, whose length is neither fixed nor identical to the output sequence, is pivotal in determining the final translation. This also implies that the temporal dependencies amongst continuous frames of gestures and facial expressions need to be studied by the model to derive accurate contextual relationships. However, ViViT has been designed to produce a single, fixed-size output corresponding to a feature vector or classification for each video. While this feature vector does well to capture overall video context, ViViT fails to preserve the sequential temporal dependencies since it considers each spatiotemporal patch (tubelet) as non-overlapping and does not provide a time-step-wise intermediate representation of the frames either. Besides, it is unable to adapt to variable-length sentence generation tasks, which are crucial for our goal.

As such, to continue to leverage the feature extraction abilities of ViViT, we bypassed this problem by moving away from returning a single pooled feature vector to generate frame-wise feature embeddings. Our backbone is a variant of this model previously trained on Kinetics-400, ViViT-B/16×2. Its core architecture has been largely kept unchanged by freezing the ViViT encoder. This is because we did not want to change the pre-trained weights, which had already been obtained by training on a significantly larger dataset than ours, in a way that compromises its performance. Instead, we get rid of the final classification head and retain the last hidden-state representations of the transformer encoder for each frame. These frame-level feature maps bear the dimensions of

```
[batch_size, num_frames, num_patches, features_dim]
```

thereby allowing us to align features to words in a sentence instead of the entire sentence.

Transformers like ViViT also come with the limitation that they do not have an inherent sense of order. They require extra steps to define the position of each video frame so that our requirements of having an ordered input sequence can be achieved. To do so, we have integrated an enhanced positional encoding scheme as part of our pipeline. Standard positional embeddings are designed to assign an encoding to each frame based on its position in the input sequence.

However, fixed positional embeddings only work for fixed-length sequences. As we have established previously, sign gestures can vary in duration based on the sign as well as the signer, and our dataset includes signing videos from different signers. Besides, this is such a time-sensitive task that a sign's meaning can change even if the same one has been performed faster in one instance and slower in another. As such, if these variations are not captured during training, our model may overfit to specific frames instead of learning the true gesture timings. Hence, our implementation of positional encoding relies on introducing a framewise gesture-specific bias and a dropout-based regularization to prevent this overfitting.

Traditional positional encoding used in transformers like ViViT is based on sine and cosine functions and assumes uniform time intervals between frames. Since this does not map well to sign language videos, we make the positional encoding more flexible using a learnable bias for each frame. We initialize the bias as a random Gaussian-distributed trainable tensor. During training, each gesture sequence influences the bias value to allow our model to dynamically adjust the positional encoding for each frame. This adaptive encoding scheme allows for context-aware adjustments and improved generalizations for real-world signing speed variations.

Additionally, since these learnable biases can cause our model to overfit to the exact timing of specific gesture sequences, we have applied dropout on the combined positional encoding along with its learning bias. We set the dropout probability range to 0.1 to 0.3. This range has been chosen to account for small datasets having considerable consistency in gesture styles as well as ones having a lot of noise or variability.

Our encoding scheme can be summarized as follows. Let V be a video sequence of T frames:

$$V = \{F_1, F_2, ..., F_T\}$$

We apply standard positional encoding:

$$PE_t = PositionalEncoding(F_t)$$

We introduce a learnable timing bias, B:

$$PE'_t = PE_t + B_t$$

We apply dropout to the result:

$$E_t = Dropout(PE'_t) = Dropout(PE_t + B_t)$$

This final enhanced encoding is more ideal for our task since it learns which frames are important for each sign and makes adjustments for timing variation among signers. It also allows our model to generalize well for pauses and overlapping signs, which are common in real-world sign language videos.

Moreover, we have introduced temporal and spatial modules to the ViViT architecture to facilitate sign language modeling. As discussed previously, the ViViT encoder is designed to use self-attention on all patches[37].

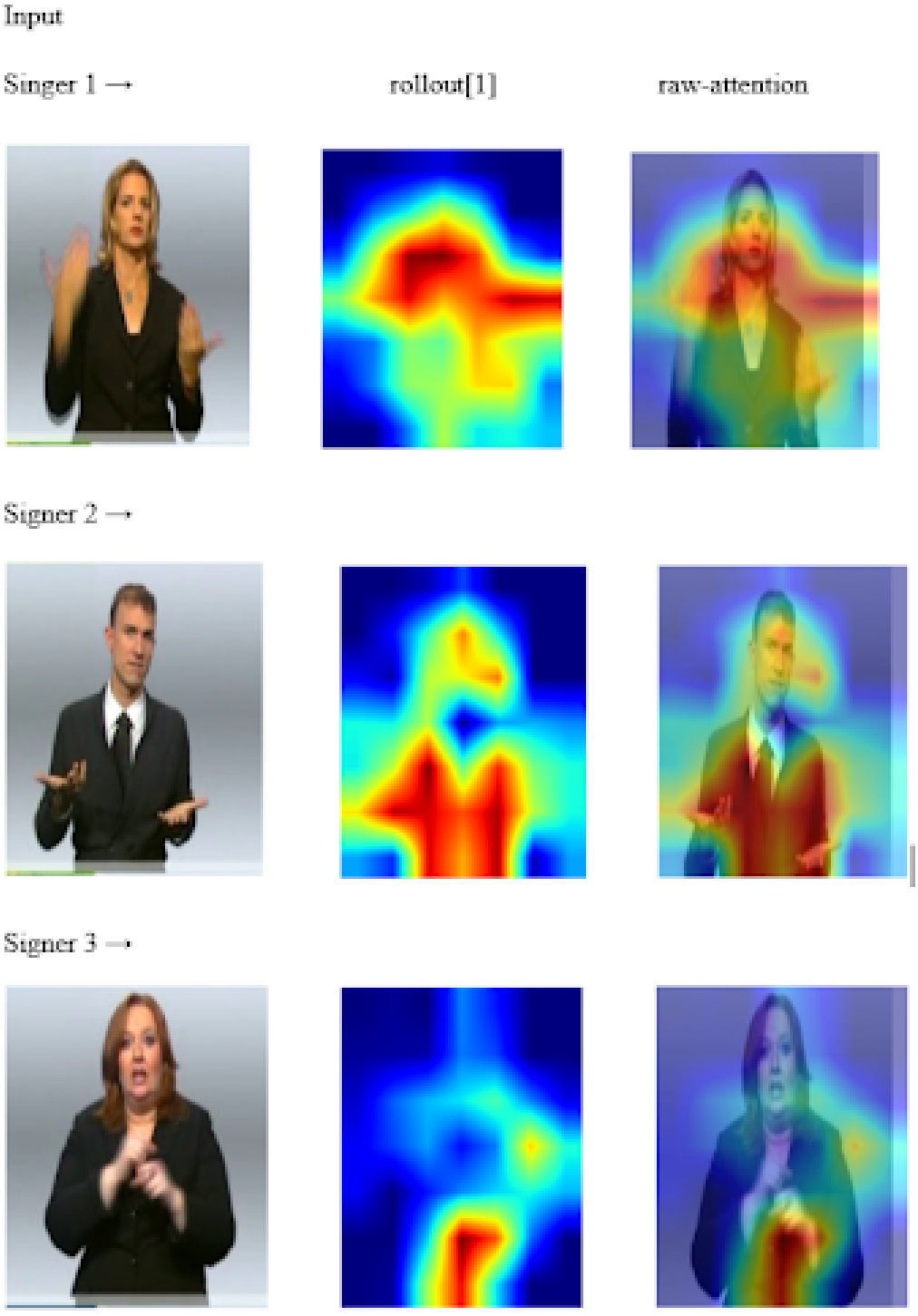


Figure 6.2: Attention Mechanism

This means that it distributes equal attention to consider each patch from a video frame to be of uniform relative relevance. While this does allow the model to learn contextual relationships, i.e, the relative position of the thumb compared the upper lip, the spatial dependencies it learns across different patches are more relevant for extracting overall video-based feature vectors. However, this is suboptimal for our sign language-based translation task because the linguistic nuances of sign language,

based on manual and nonmanual markers, can be better captured by focusing more on specific movements of the upper body parts, such as the hands, face, and shoulders. As such, we introduced an extra lightweight attention module to further refine the frame-wise feature maps. A multilayer perceptron (MLP) with learnable weights achieves this by emphasizing upper body and hand patches more than other spatial aspects. Eventually, the learned attention weights are applied to the frame-wise feature embeddings before further processing. This enhances the spatial representation of our model to better align with our objectives.

To account for the fact that we are now generating frame-wise feature maps, we have focused extensively on robust end-to-end temporal modeling to capture cross-frame dependencies.

Firstly, even before passing the video frames to the ViViT encoder and the consequent temporal modules, we employ a sliding window-based feature aggregation technique.

This adjustment was made in consideration of ViViT's fixed-length input sequence. Since sentences in our dataset, as well as sign gestures in general, vary in duration, this aspect of the model presents a significant limitation. Particularly for long input sequences, if we attempt to process an entire video at once, the compression required to fit the frames within the input window can distort the temporal resolution, which is critical for our model to make predictions. It will also lead to significant computational costs and memory constraints. Instead, the input video frames are distributed into overlapping segments of 32 frames per window with a stride of 16. These numbers have been chosen to achieve optimum context awareness from overlapping windows while ensuring essential transitions are not lost. Each of these windows is then processed independently through the transformer encoder. In this way, we extract localized temporal segments to enable variable-length input video processing without significant computational or memory overhead.

Secondly, to understand context from the frame-wise features, we incorporated a Bidirectional Long Short-Term Memory, also known as BiLSTM, to create a pipeline specifically curated to handle sign language translation.

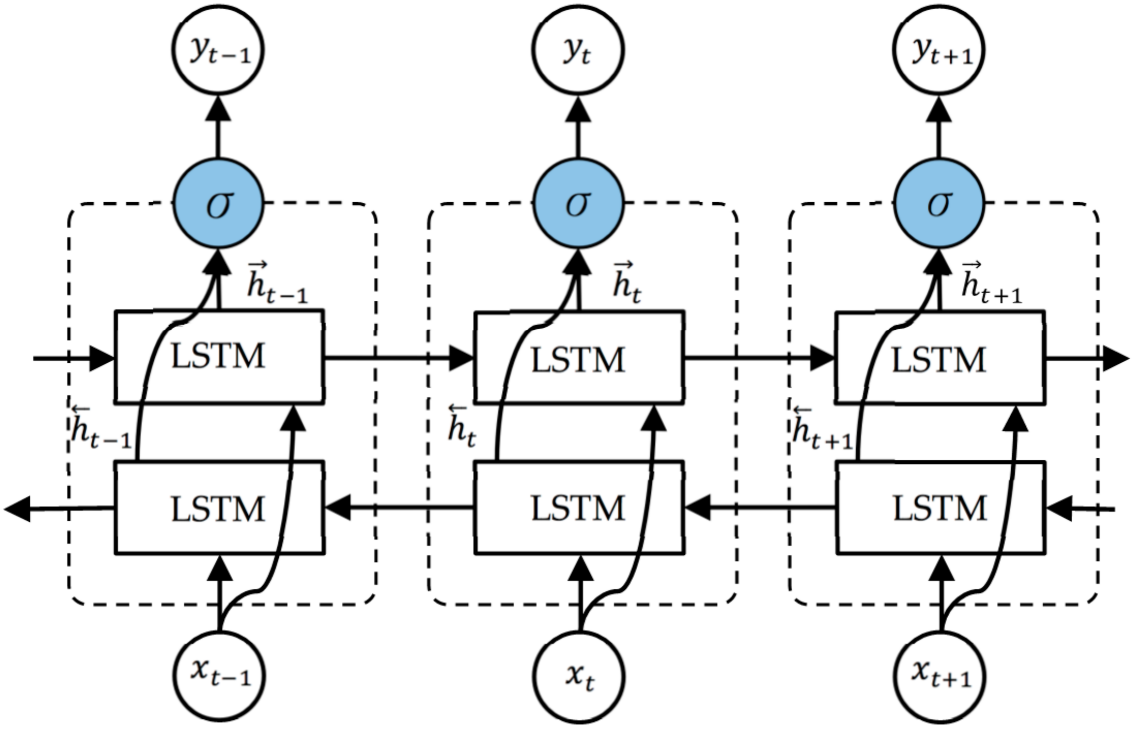


Figure 6.3: The BiLSTM architecture [23]

The features extracted from the transformer after applying our additional spatial attention module on individual frame windows can only capture spatial and local temporal patterns. In order to model long-range dependencies, BiLSTM plays a significant role. It concatenates hidden states by running in both forward and backward directions to output a 1024-dimensional context-aware vector representation for each frame. The bidirectional context is pivotal for understanding transitions between signs, as hand movements depend on a consecutive sequence of frames. Furthermore, since signing gestures are variable-length, making word boundaries fluid in nature, both preceding and succeeding frames influence the intended meaning captured by a frame.
However, it is also important to consider that BiLSTM models these long-range temporal dependencies by prioritizing high-level sequential patterns. This means that during its deeper abstraction layers, low-level spatial features, such as the precise orientation of fingers and hand shape, can become distorted. These features from ViViT are also critical for making accurate sentence predictions. As such, we introduce our third adjustment: adding an additional skip connection module to our pipeline.
This preserves the original ViViT features and combines them with the BiLSTM outputs. A learnable linear projection is used to match their dimensions before taking a weighted sum of the two to generate the final temporal feature representations.
Here, a common pipeline is designed but with the ViViT architecture in one pipeline is replaced with a TimeSformer to analyze its impact on feature extraction.
Let:

- V_t be ViViT features at time t.
- B_t be BiLSTM output at time t.

$$S_t = \text{LayerNorm}(WV_t + B_t) \tag{6.1}$$

where:

- W is a learnable projection matrix.
- LayerNorm ensures stable learning.

For the latter part of our pipeline, we utilised a fine-tuned version of BERT as the sign language decoder to handle the visual features extracted primarily from ViViT. These features serve as the encoder representation of the input sequence, as seen in traditional Transformer-based models. The decoder then follows a Transformer-based sequence-to-sequence (seq2seq) paradigm to autoregressively generate the German translation predictions. Apart from using BERT-based embeddings, key adaptations have been made to the decoder as well to improve its suitability for our task.

Firstly, while ViViT extracts 1024-dimensional feature vectors framewise, pre-trained BERT has a 768-dimensional hidden space. Any inconsistencies in input dimensionality can detrimentally impact training and also fail to utilize the extensive linguistic knowledge offered by pre-trained BERT embeddings. As such, a linear transformation followed by layer normalization is implemented to map the video features from ViViT to BERT. Following this mapping, the features undergo a bottleneck transformation. Here, feature map dimensionality is reduced to capture the most relevant features from learned representations, thereby making training less computationally expensive. Consequently, the feature maps are restored to original dimensions without having all redundant details. In this step, non-linear activation GELU and dropout regularization is also used to prevent overfitting when reducing feature complexity. Besides, this step is instrumental for guiding convergence by smoothening the feature space through activation functions. Alongside, a positional embedding scheme similar to the one we designed for our ViViT encoder is utilised in the decoder segment. It serves the same purpose of having a learnable embedding per frame to dynamically adjust frame importance. Reducing the chances of positional misalignment errors makes our decoder's temporal modelling significantly more robust.

Our Transformer-based decoder has also been adapted to predict words step by step, maintaining linguistic coherence in each step. This is achieved using self-attention and multi-head mechanisms to selectively focus on different parts of the encoded sequence. Hence, the output sequence predictions are more context-aware and the overall decoding is made more efficient by allowing parallel attention. Besides, to ensure that the decoder is not looking ahead in the sentence during training, a causal masking technique was used. This forces our model to make word predictions based on previously generated words only to progressively string together an entire sentence autoregressively. This was done to avoid any unwanted dependencies that would not be available to our model with new test data.

Moreover, a beam search algorithm has been used to select the most suitable word at each time step. At the start of decoding, the model is initialized with a start-of-sequence token. K identical active beams corresponding to independent prediction sequences are then set up with a score of 0. Consequently, at each decoding step, beam search predicts the next token's probabilities over the entire vocabulary for all active beams. The beams are expanded by appending multiple candidate tokens

to each one, and the corresponding scores are updated. Eventually, only the top K beams are passed on to the next iteration of predicting consequent tokens while the rest are pruned. This continues until the end-of-sequence token is reached or the maximum sequence length has been generated. The idea behind integrating this decoding algorithm to pretrained BERT is that it considers multiple sequences yet avoids a computationally expensive BFS search. Multiple beams generated during decoding can also be processed in parallel to each other. This allows for generating output sequences with good accuracy while not requiring too extensive computational time. This accuracy is further achieved because beam search allows exploring different words before selecting the best transition at each decoding step. This is essential for handling ambiguities better. Moreover, this algorithm outperforms greedy decoding for our task because instead of picking the most probable word at each step, it leaves room for experimenting with multiple possibilities. This also helps beam search to minimize repetitions and not get trapped in loops unlike greedy decoding.

It is also important to note that since the decoder is designed to output a sequence of hidden states, a linear transformation was performed to convert these states into a probability distribution over the entire vocabulary. The hidden states carry contextual cues from surrounding frames. This allows our model to predict which learned word any new vocabulary's temporal pattern most closely resembles, even if the vocabulary is unknown.

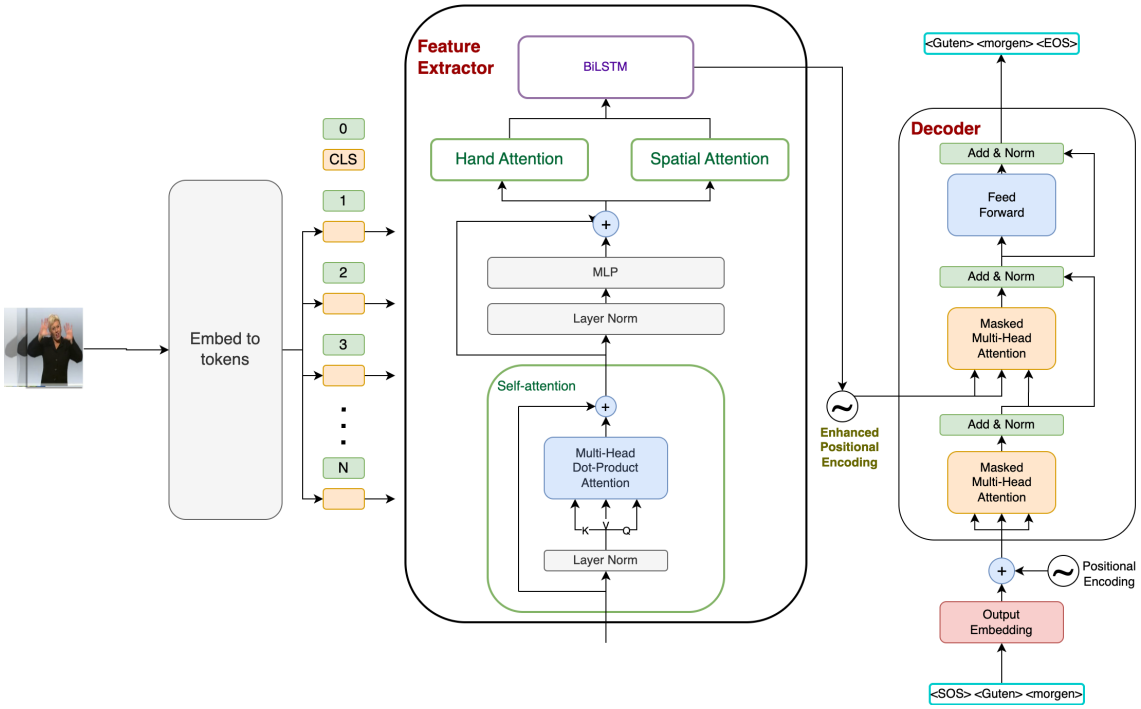


Figure 6.4: Overview of the Sign Language Translation Pipeline using ViViT as the encoder

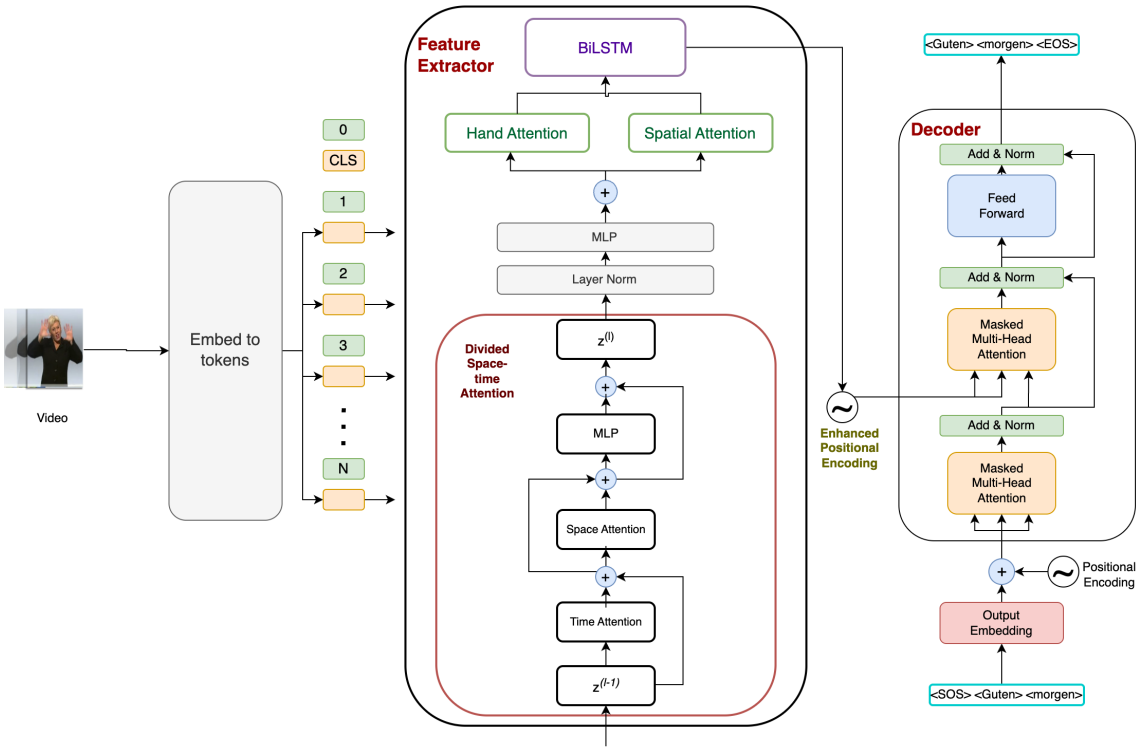
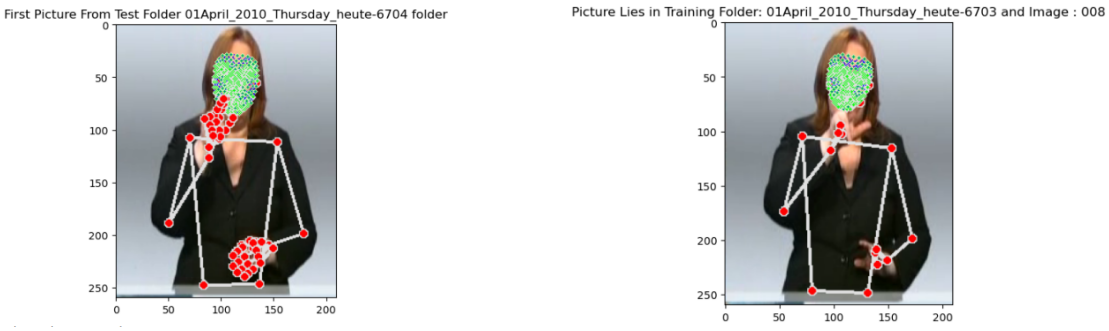


Figure 6.5: Overview of the Sign Language Translation Pipeline incorporating TimeSformer.

Chapter 7

Result Analysis

The following section summarizes and highlights key findings from our investigation into the baseline model for Continuous Sign Language Recognition (CSLR). Before achieving CSLR, implementing accurate vision techniques is crucial. All the data (training, development, and testing) in our PHOENIX 2014-T dataset exists as images of consecutive frames. Our strategy involves utilizing the pre-trained preprocessing modules of MediaPipe to extract essential features, such as pose, hand shapes, and facial expressions, from these frames in the form of keypoint representations. These representations are then fed into a fine-tuned ResNet50 architecture to complete the individual image classification phase.
The combination of MediaPipe and ResNet50 works well for a couple of reasons. MediaPipe can effectively extract fine-grained features that can serve as the foundation for image classification. Alongside, MediaPipe reduces the need for manual data augmentation or correction by robustly preprocessing key visual cues, keeping them consistent across different backgrounds and illuminated conditions. Meanwhile, ResNet50, as a deep convolutional neural network, already excels at handling complex image recognition tasks. By fine-tuning a pretrained ResNet50, we leverage its ability to capture hierarchical image features.



(a) Test Data (b) Prediction based on Test data

Figure 7.1: Comparing Test Image with Closest Match from Training Dataset

In Figure 7.1, we illustrate the successful implementation of this approach. Given the organization of data in our training corpus, the baseline model was trained to recognize which gloss a particular image in the training dataset belongs to. This mapping of an image to its gloss can then be extended to map it to the corresponding gloss sequence and German translation provided in the training corpus. Once the

training phase was complete, we tested the model's performance on unseen images from the test dataset. As demonstrated above, the model accurately identified which gloss contains an image most similar to the test subject by comparing their respective keypoint distributions. This result implies that the test image has the potential to belong to a gloss sequence similar to that of the identified training file.

This outcome represents a significant step forward. When the model encounters an image from an unknown sign, it can narrow down the search to the closest matching signs encountered during training. This information can be used to map similarities between the remaining frames of the test folder and those in the predicted training folders.

The results indicate that our approach of iteratively refining the predicted gloss sequence using CTC loss has resulted in diminishing training loss despite marginal improvement in testing accuracy (4.49% currently). Several factors could explain this outcome. The most apparent is overfitting during model training, a challenge we had anticipated when analyzing the dataset. We observed a substantial imbalance between the sizes of the training and testing datasets. As the model becomes more familiar with the training data, it may reduce training loss by memorizing specific patterns rather than learning generalizable features. This leads to poor predictive abilities when tested on unseen data. A second explanation could involve the CTC alignment process. Our primary motivation behind computing CTC loss was to align predicted sequences with target sequences, handling variability in sequence lengths. However, if the model becomes too rigid in this alignment during training, it may struggle to capture the natural variability in sign language gestures when tested with unfamiliar data. This misalignment could be a reason for the limited improvement in testing accuracy, even as training loss continues to decline. To address this, we plan to explore additional regularization techniques, such as dropout, and adjust model hyperparameters to encourage better generalization during training.

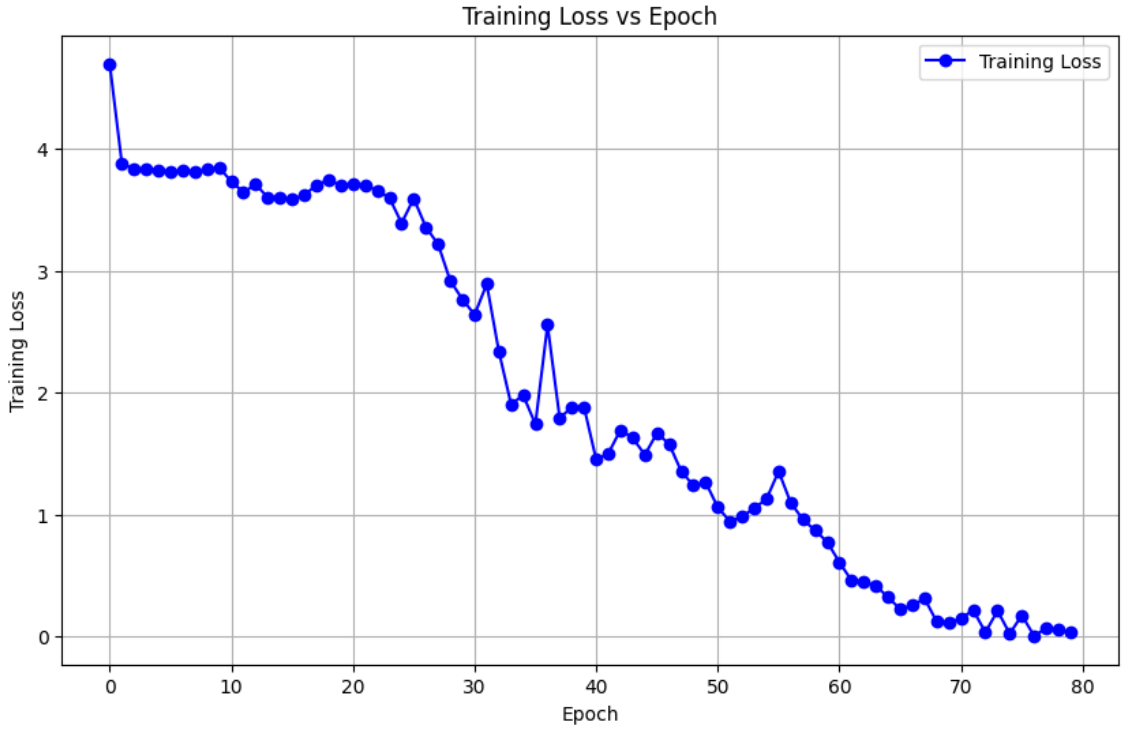


Figure 7.2: Training Progress: Loss vs. Epochs for Convergence.

Here, Figure 7.5 depicts the convergence of loss within 100 epochs. In summary, our model has not yet fully grasped the temporal relationships between consecutive frames, which are crucial for Continuous Sign Language Recognition (CSLR). While we have achieved some success in frame-level classification, this shortcoming highlights the need for a more sophisticated temporal understanding, which is essential for accurately recognizing sign language over time. A significant limitation to our progress has been the absence of precise frame-level annotations for each gloss in the sequence. Without exact frame-to-gloss mappings, the model has to rely on broad temporal windows, which reduces its ability to accurately align glosses with specific frames. This is one of the most pressing challenges of using this dataset. To address this, we are considering manually annotating the boundaries of each gloss in the sequence, which would significantly improve the model's ability to associate frames with the correct glosses. However, manual annotation is both time-consuming and requires specialized expertise, so we are also exploring alternative end-to-end approaches that could potentially reduce the model's dependence on manual gloss annotations.

To further improve our model, we are also investigating the integration of MiDAS for extracting depth features from images. Depth information could help capture spatio-temporal features that enhance sequence learning. By combining MiDAS depth features with the keypoint-based features extracted from MediaPipe and ResNet, we hope to capture a broader set of visual cues important for sign language recognition. We propose a fusion module with an attention mechanism that dynamically adjusts the relative importance of these feature streams, allowing the model to focus on the most relevant aspects of the input. This combined approach is expected to improve overall recognition performance.

In conclusion, while our current model demonstrates significant progress in reducing

training loss, challenges such as overfitting, rigid CTC alignment, and the lack of frame-level gloss annotations have limited improvements in testing accuracy. Moving forward, we plan to address these issues through regularization techniques, enhanced temporal modeling, and potential manual annotation of glosses. We are also exploring innovative approaches, such as incorporating depth features via MiDAS and attention-based fusion modules, to further enhance our model's performance. By refining these areas, we hope to advance the state of Continuous Sign Language Recognition and improve the accuracy and robustness of our model.

7.1 Inference Time and Efficiency Analysis

The Encoder-Decoder based approach towards Continuous Sign Language Translation (CSLT) managed to achieve a balance between accuracy and inference time, which is promising for the future prospects of vision-based real-time translators. The timing-based results of the Transformer-based and TimeSformer-based pipelines have been summarized in Table 7.1.

Generation-based Pipeline	**ViViT**		**TimeSformer**	
	Validation	**Test**	**Validation**	**Test**
Feature Extraction Mean	1.2874	1.2806	0.6118	0.5849
Translation Generation Mean	0.6201	0.6192	0.3308	0.3499
Total Processing Mean	1.908	1.8998	0.9426	0.9348

Table 7.1: Comparison of Processing Metrics between Transformer and TimeSformer Encoders.

It can be seen from the table that both the chosen architectures are efficient for CSLT tasks on the PHOENIX-2014 dataset since the results are consistent across both validation and test datasets. This is true despite ViViT and TimeSformer handling nearly 105 and around 138 million training parameters respectively. The reason that such an extensive parameter count does not translate to an equally extensive computational overhead is because these models are better able to leverage GPU parallelism.
However, it is evident that using the TimeSformer encoder clearly outperforms the ViViT encoder pipeline on the metric of overall inference time: one is almost half of the other's value. Although they both suffer from a nearly identical proportion of efficiency bottleneck because of their feature extractor, there is a marked improvement in both average feature extraction time and translation generation duration for the pipeline having the TimeSformer. The feature extraction mean drops from near 1.2 seconds to 0.6 seconds while the translation generation mean also drops from around 0.6 to 0.3 seconds. These results derive from our pipelines because of the different working principles of ViViT and TimeSformer. In the case of the former, parallel processing of frames is not applicable since ViViT applies spatial attention followed by temporal attention on small spatiotemporal patches from the video, i.e. tubelets. Dealing with tubelets also invites extra computations for tubelet embedding and reshaping. On the other hand, TimeSformer allows effective parallelization when first applying self-attention to each frame to capture spatial representations and relationships. Then it applies temporal attention across corresponding patches

from different frames but the reduced redundant computation improves feature extraction efficiency. Moreover, the feature representations generated by TimeSformer are also less complex and detailed, thereby being easier to process during translation generation. Together, the overall inference time is reduced

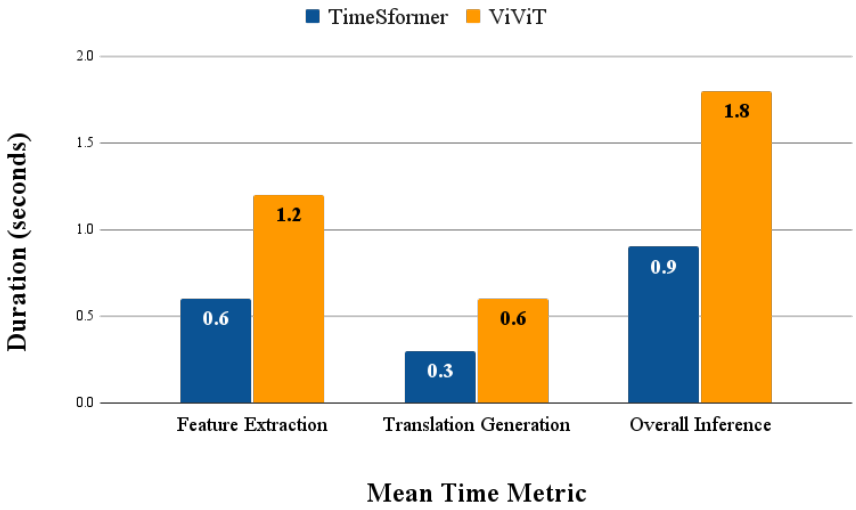


Figure 7.3: Inference time comparison between Transformer and TimeSformer Encoders using Bar chart.

Beyond near-instantaneous inferences, these pipelines are also able to predict sentences with considerable resemblance to the ground truths. The evaluation metric chosen for this study is word-level cosine similarity. This choice is due to our decoder's stepwise generation of words using causal masking to autoregressively generate translations. Since it generates one word at a time while selecting the most probable sequence through beam search, we do not wish to misrepresent our pipelines' performance as poor whenever they fail to generate the exact expected output sequence. Instead, if the predicted words are similar to those present in the ground truths while preserving the corresponding word ordering, this is still conclusive evidence of relevant predictions being made. The overall cosine values for sentence-level translations generated by the ViViT and TimeSformer-based pipelines can be seen in the subsequent table.
The table below presents the word-level cosine similarity scores for translations generated by the ViViT and TimeSformer pipelines on the PHOENIX-2014 dataset.

Pipeline	**Dataset**	**Top 5 Highest Cosine Similarities**	**Bottom 5 Lowest Cosine Similarities**	**Average Similarity**
ViViT	Validation	1.000, 0.968, 0.962, 0.947, 0.941	0.074, 0.110, 0.189, 0.213, 0.225	**0.645**
ViViT	Testing	1.000, 0.990, 0.978, 0.971, 0.968	0.091, 0.140, 0.179, 0.186, 0.189	**0.632**
TimeSformer	Validation	0.999, 0.968, 0.960, 0.936, 0.929	0.321, 0.330, 0.352, 0.356, 0.370	**0.616**
TimeSformer	Testing	1.000, 0.943, 0.918, 0.915, 0.909	0.186, 0.298, 0.326, 0.336, 0.347	**0.612**

Table 7.2: Comparison of Word-Level Cosine Similarities for ViViT and TimeSformer Pipelines.

7.2 Analysis of Prediction Accuracy

The differences in inference time for the two pipelines imply that both the chosen architectures are efficient for CSLT task on the PHOENIX-2014 dataset but TimeSformer Encoder infers at half the time. This is true despite ViViT and TimeSformer handling nearly 105 and around 138 million training parameters respectively. The reason that such an extensive parameter count does not translate to the an equally extensive computational overhead is because these models are better able to leverage GPU parallelism. The models are trained using batch sizes of 16 and over 50 epochs.

Besides near-instantaneous inferences, these pipelines are also able to predict sentences with considerable resemblance to the ground truths. The evaluation metric chosen for this paper is word-level cosine similarity. This is because our decoder works in time-steps using causal masking technique to autoregressively generate words in the final translation. Hence, as it generates one word at a time based on choosing the most probable sequence using beam search, we do not want to accredit our pipelines' performance as poor whenever it fails to generate the exact output sequence expected. Instead, if it is able to predict words similar to the ones present in ground truths while maintaining corresponding word ordering, it is also conclusive evidence of relevant predictions being made. The overall cosine values for sentence-level translations generated by the ViViT and TimeSformer based pipelines can be seen below.

The table reflects that among the two, ViViT-encoder pipeline registered higher average cosine similarity for both the datasets besides having slightly higher top similarities. These indicate that its prediction sequences often more closely matched the actual input sentence captured in the videos. However, it is also worth noting that the TimeSformer-encoder pipeline is more consistent and reliable when dealing with difficult to comprehend video sequences since its lowest similarities are not as low as the ones for ViViT. These stable cosine values mean that the latter model will never drastically fail to understand an input sequence.

This prediction accuracy may seem underwhelming since understanding the correct context is also a key objective of the CSLT task. However, it is important to acknowledge that implementing TimeSformer in this context is still in its early days, being mostly constricted to word-level sign language recognition. Even the recent works by involved training models using datasets like WLASL that have isolated signs[12]. Not many have yet considered using a video-based transformer designed for classification tasks (ViViT) in this research domain either. In fact, the few recent works that have utilised the ViViT architecture have used the LSA64 dataset [22], [26], [36]. Despite having almost the same amount of training data compared to PHOENIX 2014-T, LSA64 has only 64 unique signs while the former has 1085 glosses only. This smaller vocabulary means there are fewer long term dependencies, as well as ambiguities, among signs in LSA64 [22]. Besides, it is a video dataset for isolated words and short phrases performed in a controlled setting with the additional advantage that each signer performs all 64 signs. In contrast, the PHOENIX 2014-T dataset videos are of full sentences from a real-world weather report.

This means that training a model with the dataset in this paper requires learning complex grammar rules and detailed feature extraction all while facing the challenge

(a) The sign database for the Argentinian Sign Language(LSA64)

(b) RWTH-PHOENIX Weather 2014 T Dataset

Figure 7.4: Comparison between a randomly selected frame from LSA64 and PHOENIX 2014-T datasets to show some of the additional challenges associate with training a model on the PHOENIX dataset

of more variability in lighting and occlusions. It should also be noted that transformers are not inherently well-adapted for longer input sequences (which is why this paper implements a sliding window technique). This is a challenge more relevant for working the long sentences in the PHOENIX 2014-T dataset than LSA64. As such, considering the complexity of achieving real-time CSLT by training on this dataset, their better results can not undermine the findings of this paper. This paper goes on to show how TimeSformer is the closest to achieving the speed to accuracy ratio that can work best in a real-world sentence-level sign language translation setting. On the other hand, the neural network based classification model yielded sub-optimal performance as a real-time translator. It did allow us to train our model on the entire dataset using around 2.8 million parameters. However, fewer parameters did not contribute to faster inference time in comparison to the Encoder-Decoder based approach. The results obtained have been summarized in the table below:

Metric	**Validation**	**Test**
Total Inference Time (s)	17.8842	18.001
Feature Extraction Time (s)	11.907	12.015
Prediction Time (s)	5.962	5.986

Table 7.3: Inference Time Comparison for Validation and Test Sets.

For both the validation and test datasets, the total inference times are similar. This overall stable performance indicates that the model did not overfit to the validation dataset. Additionally, it is evident that feature extraction is the main computational efficiency bottleneck, accounting for nearly two-thirds of the overall time taken. The decision to employ ResNet-50 as the feature extractor by transfer learning was originally influenced by our findings about recent research in this field that used the PHOENIX 2014-T dataset. In multiple such papers having promising outcomes and new state-of-the-art performances, ResNet-50 was used for a similar role [9], [42]. This indicated that it could best adapt to the data variance and other nuances of our dataset by transfer learning. However, this architecture's convolution layers are computationally expensive as they require sequential processing for each

layer. Even with batch processing, each frame has to undergo several forward passes through deep CNN layers. It also generates high-dimensional feature maps that can slow down inference due to memory bottleneck. Alongside, the MLP layers that later operate on the high-dimensional features extracted by ResNet-50 further prolong the inference time since each neuron has to consider the entire input from a previous layer. All these delays accumulate to give us a model that suffers from too much latency to fulfill real-time translation requirements.

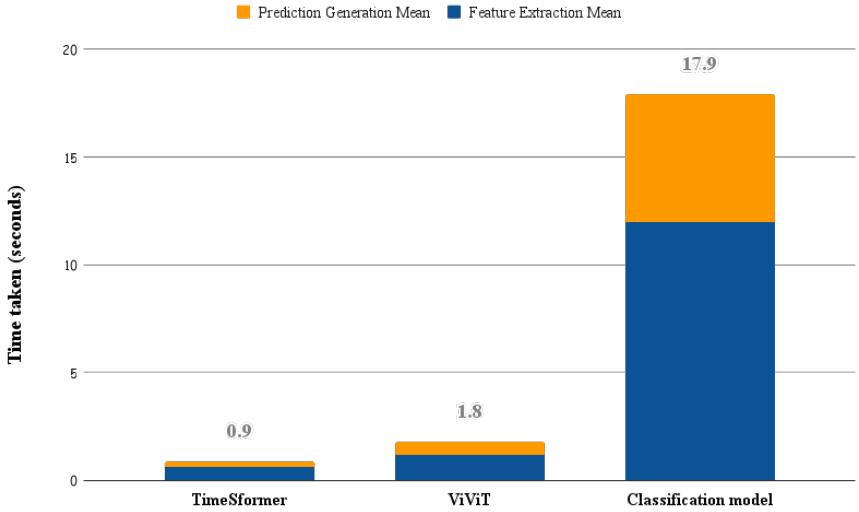


Figure 7.5: Comparison between Traditional Pipeline, Transformer and TimeSformer Encoders using Bar chart.

7.2.1 Selection of Best Model

In conclusion, the two encoder-decoder based models are more suitable for real-time Continuous Sign Language Translation (CSLT) than our classification based model. Between them, TimeSformer-encoded feature extraction gains a substantial advantage in terms of overall inference time since both the average feature extraction and translation generation durations are halved. On the other hand, ViViT can better support a goal of maximizing prediction and input sequence similarities on average. However, TimeSformer can provide more reliability and robustness when deployed in a real-world scenario since it avoids generating more incorrect output sequences than ViViT. Generating some cases where the output has almost no correlation with the input sequence would be more detrimental than having to compromise for a marginally lower average similarity between input and output. Hence, the pipeline with the TimeSformer integration appears as the more reliable candidate for this translation task.

Chapter 8

Limitations and Future Work

Despite the recent substantial advancements in sign language translation using vision-based deep learning models, several challenges still linger due to the inherent complexity and limited availability of sign language datasets. Compared to large-scale text corpora widely used in natural language processing (NLP), datasets such as Phoenix 2014-T are relatively small. This leads to potential issues with overfitting and inadequate generalization. We observe that our training loss saturates, while the validation loss overfits significantly. This underscores the overfit nature of the model.

Furthermore, the granularity of sign language data presents additional obstacles. Lack of universally standardized sign language still plagues this field. Naturally, the collection of dataset is arguably one of the most difficult problems in this field [6]. This limitation is further propagated by signer-to-signer versatility. Even in the Phoenix 2014-T dataset, there are 9 signers whose signing style, speed and articulation varies. This presents a difficult obstacle to generalize and maintain high translation accuracy simultaneously. Such deviation in handshape execution, movement dynamics, and facial expressions can trickle down to inconsistencies in the dataset and model learning. This adds to the challenge since neural models require a standardized dataset to recognize patterns but these granularities complicate their learning process. Additionally, grammar transfer stands as a vital challenge as well. Sign languages consist of unique syntactic structures that fundamentally deviate from spoken and written languages. Without the addition of more context in the dataset by the linguists with domain knowledge, technological mechanisms will barely be able to bridge these structural gaps. Our paper lacks contextual knowledge to perfectly capture such context. Therefore, explicit modeling of sign language grammar or supplementary linguistic cues is also an urgent requirement.

Fine-tuning CSLR models, particularly in alignment and attention mechanisms, also presents a future scope. Firstly, continuous signing when segmented to discrete token, regardless whether frames, feature vectors, or spatio-temporal-windows, always leaves the possibility of losing continuity across longer sequences. A well-calibrated model fills this shortcoming that is left by a more helpful tokenization system in the datasets of this paradigm of research. In absence of it, signs that span multiple frames may become misaligned, It can cause partial recognition and translation errors. Moreover, Spatial and hand-specific attention modules might enhance focus on relevant manual and non-manual markers. Yet they fail to robustly capture coarticulation effects and subtle shape-orientation changes. Spoken languages with its

diverse standardized vocabulary sets can project rhetoric in a more visible manner. Sign Language's emotional nuance and rhetoric tool is much subtle. So the model needs to dynamically allocate weight to focus on various markers based on the subtle changes in signs. If the model's attention weights are distributed too broadly across background or irrelevant features, sign recognition accuracy can significantly degrade.

Another major limitation this paper faces is the lack of pre-trained models for sign language translation. As such, sign language translation remains an excruciatingly niche task with limited external linguistic resources, unlike mainstream NLP. Specialized models benefit from large-scale pre-training on diverse text corpora. Hence, this scarcity of pre-trained models stagnates the ability to transfer knowledge from related language tasks. These challenges must be addressed in future research by developing larger, more diverse sign language datasets, improving model architectures to better capture temporal dependencies, and exploring multi-modal approaches that incorporate external linguistic and contextual knowledge.

Chapter 9

Conclusion

The objective of this research is to contribute to the motivation about enhancing the bridge between the communication of signers and the rest of the world. In this paper, we summarised our findings on the different problems that exist in creating Sign Language translation models and categorised them. For each category, multiple researches have been analysed, drawing conclusions on their limitations, barriers and scope for future work. Upon drawing conclusions on the advancements already established in this area and considering the scopes of further research, we hope to be able to make our own improvements. Now that we have summarised the barriers faced by recent researchers who have contributed to this case, our task remains to utilise the technologies in NLP, neural networks, and image processing to come up with a holistic approach to the recognition and translation issue, while careful consideration of tackling the barriers. In the age of digitalisation, communication, a fundamental living requirement, should not have to be a concern for anyone.Understanding and interpreting Sign Language is not only necessary for easy communication with signers, but also for giving the Deaf community the recognition and ease of life they deserve.

Bibliography

[1] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, *Vivit: A video vision transformer*, 2021. arXiv: 2103.15691 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2103.15691.

[2] G. Bertasius, H. Wang, and L. Torresani, *Is space-time attention all you need for video understanding?* 2021. arXiv: 2102.05095 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2102.05095.

[3] N. C. Camgöz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT: IEEE, 2018, pp. 1021–1030.

[4] R. Cui, H. Liu, and C. Zhang, "A deep neural framework for continuous sign language recognition by iterative training," *IEEE Transactions on Multimedia*, vol. 21, no. 7, pp. 1880–1891, 2019. DOI: 10.1109/TMM.2018.2889563.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1810.04805.

[6] S. E. Dhruvo, M. A. Rahman, M. K. Mandal, M. I. H. Shihab, A. A. N. Ansary, K. F. Shithi, S. Khanom, R. Akter, S. H. Arib, M. N. Ansary, S. Mehnaz, R. Sultana, S. Rahman, S. S. Chowdhury, S. A. Chowdhury, F. Sadeque, and A. Sushmit, *Bornil: An open-source sign language data crowdsourcing platform for ai enabled dialect-agnostic communication*, 2023. arXiv: 2308.15402 [cs.HC]. [Online]. Available: https://arxiv.org/abs/2308.15402.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, *An image is worth 16x16 words: Transformers for image recognition at scale*, 2021. arXiv: 2010.11929 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2010.11929.

[8] Y. Fang, L. Wang, S. Lin, *et al.*, "Visual feature segmentation with reinforcement learning for continuous sign language recognition," *International Journal of Multimedia Information Retrieval*, vol. 12, no. 39, 2023. DOI: 10.1007/s13735-023-00302-8. [Online]. Available: https://doi.org/10.1007/s13735-023-00302-8.

[9] C. Feichtenhofer, H. Fan, J. Malik, and K. He, *Slowfast networks for video recognition*, 2019. arXiv: 1812.03982 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1812.03982.

[10] L. Guo, Z. Wang, Q. Liu, W. Feng, L. Wang, and H. Liang, "Denoising-diffusion alignment for continuous sign language recognition," *ArXiv*, 2024. [Online]. Available: https://arxiv.org/abs/2305.03614.

[11] A. Hao, Y. Min, and X. Chen, "Self-mutual distillation learning for continuous sign language recognition," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, 2021, pp. 11 283–11 292. DOI: 10.1109/ICCV48922.2021.01111.

[12] M. Hrúz, I. Gruber, J. Kanis, M. Boháček, M. Hlaváč, and Z. Krňoul, "One model is not enough: Ensembles for isolated sign language recognition," *Sensors*, vol. 22, no. 13, p. 5043, Jul. 2022. DOI: 10.3390/s22135043.

[13] H. Hu, J. Pu, W. Zhou, and H. Li, "Collaborative multilingual continuous sign language recognition: A unified framework," *IEEE Transactions on Multimedia*, vol. 25, pp. 7559–7570, 2023. DOI: 10.1109/TMM.2022.3223260.

[14] L. Hu, L. Gao, Z. Liu, and W. Feng, "Continuous sign language recognition with correlation network," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2023, pp. 2529–2539.

[15] S. Jalaja and K. Shekar, "Robotic arm for sign language interpretation with sentiment analysis and auto-complete text features," *International Journal of Engineering Research and Applications*, vol. 12, no. 10, pp. 92–99, 2022. [Online]. Available: https://www.ijera.com.

[16] A. Jamwal, G. Vasukidevi, T. Malleswari, T. Vijayakumar, L. S. Reddy, and A. S. A. L. G. G. Gupta, "Real time conversion of american sign language to text with emotion using machine learning," in *2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, IEEE, 2022, pp. 603–609. DOI: 10.1109/I-SMAC55078.2022.9987362.

[17] A. Joshi, H. Sierra, and E. Arzuaga, "American sign language translation using edge detection and cross correlation," in *2017 IEEE Colombian Conference on Communications and Computing (COLCOM)*, IEEE, 2017, pp. 1–6. DOI: 10.1109/ColComCon.2017.8088212.

[18] L. Kezar, E. Pontecorvo, A. Daniels, C. Baer, R. Ferster, L. Berger, J. Thomason, Z. Sevcikova Sehyr, and N. Caselli, "The sem-lex benchmark: Modeling asl signs and their phonemes," *ArXiv*, 2023. DOI: 10.48550/arXiv.2310.00196.

[19] L. Kezar, J. Thomason, and Z. S. Sehyr, "Improving sign recognition with phonology," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, 2023, pp. 2732–2737.

[20] J. Kim and P. O'Neill-Brown, "Improving american sign language recognition with synthetic data," in *Proceedings of Machine Translation Summit XVII: Research Track*, European Association for Machine Translation, 2019, pp. 151–161.

[21] O. Koller, N. C. Camgoz, H. Ney, and R. Bowden, "Weakly supervised learning with multi-stream cnn-lstm-hmms to discover sequential parallelism in sign language videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 9, pp. 2306–2320, 2020. DOI: 10.1109/TPAMI.2019.2911077.

[22] N. Kuznietsova and S. Smirnov, "Application of vision transformers and 3d convolutional neural networks for sign language cluster recognition," in *International Workshop on Computer Modeling and Intelligent Systems*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258744193.

[23] Y.-H. Li, L. N. Harfiya, K. Purwandari, and Y.-D. Lin, "Real-time cuffless continuous blood pressure estimation using deep learning model," *Sensors*, vol. 20, no. 19, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/19/5606.

[24] Z. Maalej, "Book review: Language, cognition, and the brain: Insights from sign language research," *The Linguist List*, 2002. [Online]. Available: http://www.linguistlist.org/issues/13/13-1631.html.

[25] M. Manoharan and P. Roy, "A comprehensive review of sign language recognition: Different types, modalities, and datasets," 2022.

[26] M. Marais, D. Brown, J. Connan, and A. Boby, "Spatiotemporal convolutions and video vision transformers for signer-independent sign language recognition," in *Proceedings of the International Conference on Advances in Big Data, Computing, and Data Communication Systems (icABCD)*, Aug. 2023, pp. 1–6. DOI: 10.1109/icABCD59051.2023.10220534.

[27] R. Matlani, R. Dadlani, S. Dumbre, S. Mishra, and M. A. Tewari, "Real-time sign language recognition using machine learning and neural network," in *Proceedings of the 2022 International Conference on Electronics and Renewable Systems (ICEARS)*, IEEE, 2022, pp. 1381–1386. DOI: 10.1109/ICEARS53579.2022.9752213.

[28] Y. Min, A. Hao, X. Chai, and X. Chen, "Visual alignment constraint for continuous sign language recognition," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, 2021, pp. 11 522–11 531.

[29] A. Moryossef, Z. Jiang, M. Müller, S. Ebling, and Y. Goldberg, "Linguistically motivated sign language segmentation," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, Association for Computational Linguistics, 2023, pp. 12 703–12 724.

[30] M. Mukushev, A. Sabyrov, A. Imashev, K. Koishybay, V. Kimmelman, and A. Sandygulova, "Evaluation of manual and non-manual components for sign language recognition," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, European Language Resources Association, 2020, pp. 6073–6078.

[31] Z.-Y. Niu and B.-K. Mak, "Stochastic fine-grained labeling of multi-state sign glosses for continuous sign language recognition," in *Proceedings of the European Conference on Computer Vision*, 2020.

[32] C. Olah, *Understanding lstms*, 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[33] J. Pu, W. Zhou, H. Hu, and H. Li, "Boosting continuous sign language recognition via cross modality augmentation," *ArXiv*, 2020. DOI: 10.1145/3394171.3413931.

[34] J. Pu, W. Zhou, and H. Li, "Iterative alignment network for continuous sign language recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2019, pp. 4160–4169.

[35] Q. Rao, K. Sun, X. Wang, Q. Wang, and B. Zhang, "Cross-sentence gloss consistency for continuous sign language recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 4650–4657.

[36] N. Singh, V. Shinde, and P. Kanani, "American sign language recognition using video vision transformers," in *Proceedings of the International Conference on Advances in Computing, Communication and Technology Applications (ICACTA)*, Oct. 2023, pp. 1–6. DOI: 10.1109/ICACTA58201.2023.10392991.

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[38] A. Wali, R. Shariq, S. Shoaib, S. Amir, and A. A. Farhan, "Recent progress in sign language recognition: A review," *Machine Vision and Applications*, vol. 34, pp. 1–20, 2023.

[39] C. Xue, M. Yu, G. Yan, *et al.*, "Continuous sign language recognition based on iterative alignment network and attention mechanism," *Multimedia Tools and Applications*, vol. 82, no. 15, pp. 17 195–17 212, 2023. DOI: 10.1007/s11042-022-13705-7.

[40] K. Yin and J. Read, "Better sign language translation with stmc-transformer," in *Proceedings of the 28th International Conference on Computational Linguistics*, International Committee on Computational Linguistics, 2020, pp. 5975–5989.

[41] H. Zhou, W. Zhou, Y. Zhou, and H. Li, "Spatial-temporal multi-cue network for continuous sign language recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[42] Q. Zhu, J. Li, F. Yuan, and Q. Gan, *Continuous sign language recognition based on cross-resolution knowledge distillation*, 2023. arXiv: 2303.06820 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2303.06820.

[43] R. Zuo and B.-K. Mak, "C²slr: Consistency-enhanced continuous sign language recognition," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2022, pp. 5121–5130. DOI: 10.1109/CVPR52688.2022.00507.

[44] R. Zuo, F. Wei, and B.-K. Mak, "Towards online sign language recognition and translation," *ArXiv*, 2024. [Online]. Available: https://arxiv.org/abs/2401.05336.