CONTINUOUS AMERICAN SIGN LANGUAGE TRANSLATION WITH ENGLISH SPEECH SYNTHESIS USING ENCODER-DECODER APPROACH

by

PREETHAM GANESH

THESIS

Submitted in partial fulfillment of the requirements

for the degree of Master of Science in Computer Science at

The University of Texas at Arlington

May, 2021

Arlington, Texas

Supervising Committee:

Vassilis Athitsos, Supervising Professor

Christopher Conly

Alex Dillhoff

ABSTRACT

CONTINUOUS AMERICAN SIGN LANGUAGE TRANSLATION WITH ENGLISH SPEECH SYNTHESIS USING ENCODER-DECODER APPROACH

Preetham Ganesh, M.S.

The University of Texas at Arlington, 2021.

Supervising Professor: Vassilis Athitsos

Interaction between human beings brings about improvements in science and technology. However, the interaction is limited for people who are deaf or hard-of-hearing, as they can only communicate with others who also know their sign language. With the help of recent technologies, such as Deep Learning, the gap can be bridged by converting Sentence-based Sign Language videos into English language speech. The methods discussed in this thesis are taking a step closer to solve that problem. There are four steps involved in converting ASL (American Sign Language) videos to English language speech. Step 1 is to recognize the phrases performed by the user in the videos. Step 2 is to convert the SVO (Subject-Verb-Object) phrases in the ASL glossary to English language text format. Step 3 would be to convert the English language text (graphemes) to English language phonemes. Step 4 would be to convert the English language phonemes to English language spectrogram.

We developed the Video-to-Gloss module by constructing a Sentence-based ASL dataset using word-based WLASL (Word-level American Sign Language) dataset as the base dataset, where the WLASL dataset was used for generating random phrases from the videos. We used 2D (2-Dimensional) human pose-based approach for extracting keypoint information from videos, and the extracted information were fed into the Seq2Seq (Sequence-to-Sequence) architecture to convert the signs from videos into words (ASL gloss). We developed the Gloss-to-Grapheme module using the ASLG-L12 dataset, where the Attention-based Seq2Seq & Transformer architectures were used for training the models. We developed the Grapheme-to-Phoneme module using the CMUDict dataset, where the models were trained similar to the Gloss-to-Grapheme module, i.e., using the Attention-based Seq2Seq architectures were used to train the model. We developed the Phoneme-to-Spectrogram model using the LJSpeech dataset, where the Transformer architecture was used for training the model.

KEYWORDS: Sign Language Recognition, English Speech Synthesis, ASL Translation, Seq2Seq model, Attention Mechanism.

ACKNOWLEDGEMENT

Firstly, I would like to thank Dr. Vassilis Athitsos for allowing me to prospect various parts of research and constantly motivate me and encourage me throughout the study. I would also like to thank Dr. Christopher Conly and Dr. Alex Dillhoff for accepting my invitation to be on the supervising committee and providing valuable feedback on my research. I want to thank my lab mates Balasubramaniam Theetharappan, Saif Sayed, and Farnaz Farahanipad for sharing their knowledge with me during my hard times.

I would like to extend my sincere thanks to my family and friends for supporting me during my research. I would also like to thank all those who have helped, guided, or indirectly encouraged us during the study. Finally, I would like to thank God for his blessings, which helped me complete my research.

Dedicated

to

My Dearest Parents,

Mrs. Kalpana Ganesh, and Mr. Ganesh Ramalingam

and

My best friends.

LIST OF ACRONYMS

2D	2-Dimensional
3D	3-Dimensional
ASCII	American Standard Code for Information Interchange
ASL	American Sign Language
ASL-LVD	American Sign Language Lexicon Video Dataset
AUSLAN	Australian Sign Language
Bi	Bidirectional
BLEU	Bi-Lingual Evaluation Understudy
BSL	British Sign Language
BSL-1K	British Sign Language - 1K
BSLCP	British Sign Language Corpus Project
CNN	Convolutional Neural Network
CTC	Connectionist Temporal Classification
DNN	Dense Neural Network
GFE	Gloss Feature Enhancement
GPS	Global Positioning system
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
HTML	Hypertext Markup Language

I3D	Inflated 3D ConvNet
IoT	Internet of Things
ISL	Indian Sign Language
LSTM	Long Short-Term Memory
METEOR	Metric for Evaluation of Translation with Explicit Ordering
MOS	Mean Opinion Score
MSASL	Microsoft American Sign Language
NN	Neural Network
PER	Phoneme Error Rate
POS	Parts-of-Speech
PS	Precision Score
RC	Residual-Connection
RGB	Red-Green-Blue
RNN	Recurrent Neural Network
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
Seq2Seq	Sequence-to-Sequence
SER	Sentence/Phrase Error Rate
SFD	Stochastic Frame Dropping

SFL	Stochastic Fine-Grained Label
SGD	Stochastic Gradient Descent
SVO	Subject-Verb-Object
Uni	Unidirectional
WER	Word Error Rate
WLASL	Word-level American Sign Language

LIST OF SYMBOLS

b	Backward layer
c	Carry State
dm	Number of units
f	Forward layer
ff	Number of units in feed-forward layer
FP	False Positives
h	Number of attention heads
i	ith layer
k	Number of keypoints
m	Memory State
N	Number of layers
t	Timesteps
TP	True Positives
W	Weight
x	Input or Output

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

Sign language is a tool used by people who are deaf or hard-of-hearing. Different countries or regions follow no universal sign language. Around the world, there are more than 135 sign languages. Most countries have their sign languages such as ASL (American Sign Language), BSL (British Sign Language), AUSLAN (Australian Sign Language), ISL (Indian Sign Language), and many more. ASL is a proper, natural language with similar semantic properties as spoken languages. ASL is mainly communicated with the help of movements in the face and hands. If an outsider wants to interact with a person from the sign language community, they may have to learn their corresponding sign language and then converse, which is time-consuming and requires much effort. A solution to this would be to use a translator who would know the corresponding sign language, but it can be expensive and intrusive.

An efficient solution would be to use an application that can recognize the sign language in the videos and convert them into English language Speech with the help of the latest technologies such as Computer Vision and Natural Language Processing. In this thesis, we have designed a pipeline that would convert sentence-based ASL videos to English language speech. It mainly requires the following four steps:

1. Recognize the words in the Sign Language videos (Video-to-Gloss module).
2. Convert the ASL phrases from SVO (Subject-Verb-Object) format to English language text format (Gloss-to-Grapheme module).
3. Convert the English language test (graphemes) to English language phonemes (Grapheme-to-Phoneme module).
4. Convert the English language phonemes to English Language spectrogram (Phoneme-to-Spectrogram module).

Sign Language Recognition can be done in 3 different ways: (1) Character-level Sign Language Recognition, (2) Word-level Sign Language Recognition (isolated), and (3) Sentence-level Sign Language Recognition (continuous). Characters-level Sign Language has 36 signs, where 26 are for the English alphabets, and the remaining ten are numerals (0-9). Although there have been many works in the area of Character-level Sign Language Recognition [1, 2], it is intensive to spell every word in the sign language and hence is not used in day to day life. A more straightforward approach would be to use Word-level Sign Language Recognition or Sentence-level Sign Language Recognition. However, there are a few difficulties in these approaches, which are listed below:

- The sign language vocabulary in daily use is relatively high (mostly in thousands), making it challenging to develop a system capable of capturing the features in all the signs.
- Despite the extensive vocabulary, there are a few words that may not be present in it, such as people's names; in such cases, it may be necessary to use character-level signs to represent those words.
- The recognition of signs mainly depends on the mixture of body, hand, and head movement. There can be two signs which may make just a subtle difference, which, if not appropriately recognized, may lead to incorrect classification.

ASL gloss translation is necessary because someone outside the sign language community can not comprehend all SVO format sentences. This can be illustrated with the help of 2 examples:

1. - ASL Gloss: Ginger should not eat beef
   - English Text: Ginger should not eat beef
2. - ASL Gloss: Tennis I like play not
   - English Text: I don't like to play tennis

In example 1, it can be understood that the words in both sentence structures remain the same; however, in example 2, it can be understood that the order of words in both the sentences are different, along with the choice of words. Hence, the translation of SVO format text to English language text is necessary.

The English Speech Synthesis is another critical component of the thesis, where the English language text is converted from text to speech using deep neural networks. It is used in many fields such as GPS (Global Positioning system), devices with speech capability, and people with visual and reading disabilities [3]. In other words, it helps people interact with IoT (Internet of Things) devices, where the interaction does not require any physical interface; instead, a person's voice would act as the interface. There are multiple stages in the process of converting any language text to its corresponding speech, namely, Grapheme-to-Phoneme conversion model, phoneme segmentation model, phoneme duration model, fundamental frequency model, and audio synthesis model [4]. Due to its complexity in the number of stages, Speech Synthesis is still one of the top researched topics in deep learning.

The methodology mentioned in this thesis is a step towards solving the problems/difficulties mentioned above. We propose a novel state-of-the-art end-to-end approach to the translation of ASL videos to English language speech. The contributions of this research as follows:

- First exploration of ASL videos to English language speech problem.
- We introduce a sizeable sentence-based dataset containing short phrases (length = 4) using WLASL as the base dataset.
- Performance improvement to Sign language Recognition on the WLASL dataset.
- First exploration on usage of POS (Parts-of-Speech) Tagging approach to ASL Gloss to English language text translation.
- Performance improvement to ASL Gloss to English language text on the ASLG-PC12 dataset.
- We also provide an ablation study on testing different parameter ranges and combinations such as Attention mechanism, model complexity, and many more.

The structure of the proposed work is as follows: Chapter 2 discusses the related works in the field of Sign language recognition and English speech synthesis techniques used by researchers; Chapter 3 explains the methodology used for developing the proposed system; Chapter 4 describes in detail the datasets used, performance measures used, results of the proposed system; Chapter 5 concludes the report based on the derived results.

Chapter 2

RELATED WORK

This chapter reviews the different approaches to Sign Language Recognition, ASL Gloss Translation, Grapheme-to-Phoneme conversion, and English Speech Synthesis, and its state-of-the-art results.

2.1 VIDEO-TO-GLOSS DATASETS

There are six publicly available Word-based Sign Language datasets, such as Purdue RVL-SLLL American Sign Language dataset [5], ASL-LVD (American Sign Language Lexicon Video Dataset) [6], BSLCP (British Sign Language Corpus Project) dataset [7], MSASL (Microsoft American Sign Language) dataset [8], WLASL (Word-level American Sign Language) dataset [9], and BSL-1K (British Sign Language - 1K) dataset [10]. Table 2.1 contains the details of the Word-based Sign Language datasets mentioned above.

Table 2.1: Details of the Word-based Sign Language datasets

Name	**Country**	**Classes**	**Samples**	**Subjects**
Purdue RVL-SLLL [5]	American	39	2576	184
ASL-LVD [6]	American	2742	9794	-
BSLCP [7]	British	≈ 5000	≈ 50000	249
MSASL [8]	American	1000	25513	222
WLASL [9]	American	2000	21083	119
BSL-1K [10]	British	1064	$\approx 273K$	49

The Purdue RVL-SLLL dataset [5] contains 2576 videos, where these signs in the videos were performed by 14 volunteers, with approximately 184 videos per volunteer. The vocabulary count of the words in the dataset is 39. All the videos were recorded in high-resolution with different lighting conditions to ensure that each signer had fewer shadows and enhanced contrast. Each video in the dataset was recorded in the RGB format with AVI extension where the frame size was 640 x 480 pixels.

The ASL-LVD [6] contains 9794 videos, with a unique vocabulary count of 2742 words, where each gloss had approximately 3.6 videos. The first frontal view and face view videos were recorded at 60 fps, with a frame size of 640 x 480 pixels, and the second frontal view videos were recorded at 30 fps with a frame size of 1600 x 1200 pixels. The BSLCP dataset [7] consists of nearly 50000 videos with a unique vocabulary count of approximately 5000. The average number of videos per gloss sign is 10, where 249 different signers performed the signs.

The MSASL dataset [8], contains 25513 videos, where the unique vocabulary count is 1000. The authors provided 4 subsets of the dataset based on the vocabulary count, i.e., gloss count = {100, 200, 500, 1000}. A total of 222 signers have performed the videos' actions, where the minimum number of videos per class is 11, and the mean number of videos per class is 25.5. A total of 16054 videos are in the training set, 5287 videos in the validation set, and 4172 videos in the testing set. The combined duration of the video dataset is 24 hours and 39 mins approximately.

The WLASL dataset [9], contains 21083 videos with a unique gloss count of 2000. The dataset comprises signs performed by 119 signers, with a mean number of 10.5 videos per gloss. The length of videos in the dataset ranges from 0.36 seconds to 8.12 seconds, where the average length of videos is 2.41 seconds. Similar to the MSASL dataset [8], the authors of the paper also split the dataset into 4 subsets based on the unique gloss count, gloss count = {100, 300, 1000, 2000}. The total duration of the video dataset was approximately 14 hours.

There are two publicly available datasets for Sentence-based Sign Language datasets, RWTH PHOENIX Weather 2014 dataset [11], and SIGNUM [12]. Table 2.2 contains the details of the Sentence-based Sign Language datasets mentioned above.

Table 2.2: Details of the Sentence-based Sign Language datasets

Name	**Country**	**Classes**	**Samples**	**Subjects**
PHOENIX 2014 [11]	German	1200	45760	9
SIGNUM [12]	German	450	33210	25

Since, there are no publicly available large phrase-based/sentence-based dataset for ASL, we construct a dataset using a word-based ASL dataset as the base dataset. There are four options for word-based ASL dataset, namely, Purdue RVL-SLLL ASL dataset [5], ASL-LVD [6], MSASL dataset [8], and WLASL dataset [9]. We chose WLASL dataset as the base dataset because, when compared in terms of number of classes it has more number of classes than Purdue RVL-SLLL ASL dataset, and MSASL dataset, and lesser number of classes than ASL-LVD. However, when WLASL and ASL-LVD dataset are compared in terms of number of videos per class, WLASL has higher number of videos per class.

2.2 SEQUENCE-TO-SEQUENCE APPROACHES

The Seq2Seq (Sequence-to-Sequence) model [13] mainly consists of 2 sub-models: An Encoder model and a Decoder model. The Encoder takes input from one format and encodes it with the help of RNN (Recurrent Neural Network). The results are passed into the decoder, which then decodes it to the output format with the help of RNN and a softmax layer. It has multiple uses such as response generation, language translation, image captioning, and text summarization [14]. Bahdanau et al. in [15] introduced the concept of Attention to the Seq2Seq architecture, as the normal Seq2Seq failed to work for longer sentences (i.e., more than 40 words or tokens). The authors' idea presented in the paper was to use the input and the previous timestep's output to predict the output of the current timestep. The authors concluded that their approach produced better results than the previous Seq2Seq model [13] for longer sentences.

Luong et al. in [16] provided different approach to the Attention-based Seq2Seq model. The Bahdanau Attention model is considered a Local Attention model, whereas the Luong Attention model is a Global Attention model. The difference between the Luong Attention model and the Bahdanau Attention model is that the Bahdanau Attention model uses the output for the top-most LSTM layer from the encoder model to calculate the context vector and concatenates the context vector with the top-most LSTM layer's previous timestep's output. However, the Luong Attention model takes the output of the top-most LSTM layer of both the Encoder and Decoder models for calculating the context vector.

Vaswani et al. in [17] proposed a Self-Attention based DNN (Dense Neural Network) Seq2Seq modeling, commonly called as Transformer architecture. It computes the context vectors based on a Multi-layer DNN model where each layer DNN layer is attached to a Self-Attention layer. The Self-Attention layer allows the input to interact to identify the words that need more attention. There are two advantages of Self-Attention over other Attention architectures: (1) Capability to perform parallel computing (in comparison with RNN based Encoder and Decoder models); (2) Lesser need for Deep RNN architectures, which take more time compared to Deep DNN architectures. It helps in a more effortless flow of gradients through all the states, which helps solve the vanishing gradient problem to some extent. The Multi-head attention helps the model cooperatively attend to the statistics from unique representation subspaces at unique positions.

2.3 VIDEO-TO-GLOSS APPROACHES

There are three steps involved in the video-based sign language recognition process, namely (1) feature extraction, (2) temporal mapping of features, and (3) classification. Many researchers have tried different approaches for feature extraction in sign language recognition, such as hand-engineered feature-based classification [18, 19], body part-based classification [20, 21, 22], facial features based recognition such as [12, 23], image appearance-based CNN (Convolutional Neural Network) based recognition [24, 25], and pose information-based recognition [26, 27].

There are many approaches performed by researchers for the temporal mapping of features from CNN for sign language recognition such as HMM (Hidden Markov Model) based approach [28, 29], GRU (Gated Recurrent Unit) based approach [9, 30], and LSTM (Long Short-Term Memory) [20, 31]. Researchers have tried 3D (3-Dimensional) CNN models to map the spatial and temporal features from a video in a combined manner instead of extracting separately [32, 33].

The image appearance-based model consists of a CNN, used to extract spatial features from images given as input, and pass the extracted features (flattened or average pooled) into fully connected layers for classification [34, 35]. Over time, the CNN models started extracting complex features from images, i.e., these models could extract spatial features from images and extract temporal features from a sequence of photos, i.e., videos [36, 37].

There are two ways to extract the temporal features, (1) combine a 2D (2-Dimensional) CNN model to extract the spatial features with a RNN (Recurrent Neural Network) model to extract the temporal features, or (2) use a 3D CNN model to extract the spatial features and temporal features. The pose based recognition model consists of two types of approaches, namely: (1) extract the pose information or keypoints of humans in the video frames using a deep CNN [38] and map the temporal features across frames using RNN model [39], or (2) use non-maximal suppression technique on the prediction of heatmap based pose information or keypoints. Table 2.3 describes the current state-of-the-art approaches for Sign language recognition and translation in a detailed manner.

Cui et al. in [40] used Recurrent CNN-based feature extraction along with Bi-LSTM and Detection Net on RWTH PHOENIX Weather 2014 dataset. The authors concluded that their model learned distributed portrayal among various signers and handled inter-signer variations to a reasonable extent. Camgoz et al. in [41] improved the work mentioned above by using Attention-based Seq2Seq models. Their approach was to view sign language as an independent language and that using a language translation approach might help solve the problem. The authors concluded that their model could produce better translation than the state-of-the-art; however, there was one problem with their approach, i.e., their model could not translate/capture essential information such as date, numbers, and places.

Pu et al. in [42] improved the work mentioned above by using 3D Convolution Residual Neural Network-based feature extraction, and Attention-based Seq2Seq model on the dataset. The authors approach was to use a 3D-ResNet-based Deep CNN architecture for feature extraction, unlike

Table 2.3: Description of the current state-of-the-art works for Sign Language Recognition and Translation

Paper	Datasets	Feature Extraction	Algorithms	Performance Measures	Results
Cui et al. in [40]	RWTH PHOENIX Weather 2014	Recurrent CNN	Bi-LSTM & Detection Net	WER	Val WER = 39.4, Test WER = 38.7
Camgoz et al. in [41]		CNN	Attention-based Seq2Seq	ROUGE, BLEU	Val BLEU = 18.40, Test BLEU = 18.13
Pu et al. in [42]		3D Convolution Residual NN	Attention-based Seq2Seq	WER	Val WER = 37.1, Test WER = 36.7
Camgoz et al. in [43]		CNN	Multi-head Attention & Transformer	WER, BLEU	Val WER = 24.98, Val BLEU = 22.38, Test WER = 26.16, Test BLEU = 21.32
Saunders et al. in [44]		-	Symbolic & Progressive Transformer	ROUGE, BLEU	Val BLEU = 20.23, Test BLEU = 19.10
Niu et al. in [45]		CNN	Transformer	WER	Val WER = 24.9, Test WER = 25.3
Cheng et al. in [46]		CNN	Gloss Feature Encoder & Enhancement Decoder	WER	Val WER = 23.7, Test WER = 23.9
Albanie et al. in [10]	BSL-1K	CNN	I3D, Pose-COCO	Top-K Accuracy	Top-1 Accuracy = 64.71%
Li et al. in [9]	WLASL	Image-based CNN, Pose-based CNN	GRU, I3D, Temporal Graph CNN	Top-k Accuracy	Test Top-10 Accuracy = 66.31

Camgoz et al. in [41], where the authors used 2D CNN for extracting features. The authors also used CTC (Connectionist Temporal Classification)-based learning with the help of repetitive optimization. It was concluded that their model was able to produce better translation than the state-of-the-art. Camgoz et al. in [43] enhanced the result from the previous work with the help of Transformer architecture-based Sign language translation. The authors used spatial embedding to help with the positioning of the frames in the videos representing signs. The authors concluded that their model could produce better translation than the state-of-the-art; however, there was one problem with their approach, it was not handling the standard grammar, which is essential in certain cases.

Niu et al. in [45] improved the work mentioned above by using ResNet-based Visual Encoder, and Transformer Encoder as the contextual model, and SFL (Stochastic Fine-Grained Label)-based CTC as the alignment model. The authors were able to address the issue of unsatisfactory performance of CNN-based Transformer model with CTC loss, by introducing SFD (Stochastic Frame Dropping) and SGD (Stochastic Gradient Descent). It was concluded that their model was able to produce better translation than state-of-the-art. Cheng et al. in [46] enhanced the result in the previous work by using CNN-based Gloss Feature Encoder, and Enhancement Decoder Seq2Seq model. The authors introduced a GFE (Gloss Feature Enhancement) module for enforcing better sequence alignment learning without any pre-training. It was concluded that their model was able to produce better translation than state-of-the-art.

2.4 GLOSS-TO-GRAPHEME APPROACHES

The main approach used by authors for translating ASL Gloss to English language text is with the help of Seq2Seq [13], and Transformer architecture [17]. The authors modify the Seq2Seq approach with the help of various attention mechanisms to extract essential information from input and target text, namely, Bahdanau Attention [15], and Luong Attention [16] mechanisms. Table 2.4 describes the current state-of-the-art approaches for translation of ASL Gloss to English language text in a detailed manner.

N. Arvantis et al. [47] used Luong Attention-based Seq2Seq architecture to translate ASL Gloss phrases to English language text sentences. The authors concluded that their 4-layer Luong Attention-based Seq2Seq architecture produced a BLEU score 65.0 on the testing set. The authors also concluded that their model could not capture the large vocabulary due to the smaller

Table 2.4: Description of the current state-of-the-art approaches for translation of ASL Gloss to English language text

Paper	Dataset	Algorithms	Performance Measures	Results
N. Arvantis et al. [47]	ASLG-PC12	Luong Attention-based Seq2Seq	BLEU	Test BLEU = 65.0
K. Yin and J. Read in [48]	ASLG-PC12	Pre-trained Embedding, Ensemble Transformer	BLEU, ROUGE, and METEOR	Test BLEU = 82.87, Test ROUGE = 96.22, and Test METEOR = 96.6

dataset size and could not produce quality translations for longer sentences. K. Yin and J. Read in [48] enhanced the result of the above work by using a Pre-trained Embedding and Ensemble Transformer-based approach for translation of ASL Gloss to English language text. The authors reduced the vocabulary size by dropping words from the dataset when the number of occurrences was less than 5. The problem with this approach is that the model was trained on a smaller vocabulary (due to heavy data pre-processing). Also, due to end-to-end single model training, the model facing a lot of information loss.

2.5 GRAPHEME-TO-PHONEME APPROACHES

Similar to ASL Gloss to English language text, the main approach used by authors for translating English language Graphemes to English language Phonemes with the help of Seq2Seq [13], and Transformer architecture [17]. The authors modify the Seq2Seq approach with the help of various attention mechanisms to extract essential information from input and target text, namely, Bahdanau Attention [15], and Luong Attention [16] mechanisms. Table 2.5 describes the current state-of-the-art approaches for translating English language graphemes to English language phonemes in a detailed manner.

M. Bisani et al. in [49] used a Joint Sequence model on the CMUDict dataset, where the authors used PER (Phoneme Error Rate), and SER (Sentence/Phrase Error Rate) metrics to evaluate the model. It works based on sequence alignment between graphemes and phonemes and predicts

Table 2.5: Description of the current state-of-the-art approaches for translation of English language graphemes to English language phonemes on the CMUDict dataset

Paper	**Algorithms**	**Performance Measures**	**Results**
M. Bisani et al. in [49]	Joint Sequence model	PER, SER	PER = 5.88, SER = 24.53
K. Yao et al. in [50]	Encoder-decoder LSTM	PER, SER	PER = 5.45, SER = 23.55
A. E. Mousa et al. in [51]	Deep Bi-LSTM with many-to-many alignment	PER, SER	PER = 5.37, SER = 23.23
S. Yolchuyeva et al. in [52]	Encoder CNN, decoder Bi-LSTM	PER, SER	PER = 4.81, SER = 25.13
S. Yolchuyeva et al. in [53]	Transformer model	PER, SER	PER = 5.23, SER = 22.1

based on a combined n-gram language model over phrases. The authors concluded that the model produced a PER of 5.88 and a SER of 24.53. K. Yao et al. in [50] improved the work mentioned above by using LSTM-based Seq2Seq model on the dataset. The authors introduced the Seq2Seq approach to the Grapheme-to-Phoneme model. The authors concluded that the model produced a PER of 5.45 and a SER of 23.55. Since the authors did not use the Attention mechanism, the model could not capture important information.

A. E. Mousa et al. in [51] enhanced the result of the work mentioned above by using Deep Bi-LSTM with a many-to-many alignment model on the dataset. The authors used three types of alignment constraints, namely, 1 grapheme to 1 or 2 phonemes, 1 or 2 graphemes to 1 or 3 phonemes, and many graphemes to many phonemes. The authors concluded that many graphemes to many phonemes alignment model produced the best results. Similar to [50], the authors did not use the Attention mechanism; the model could not capture essential information.

S. Yolchuyeva et al. in [52] enhanced the work mentioned above by using Encoder CNN, decoder Bi-LSTM model on the dataset. The authors introduced a residual CNN based model to grapheme-to-phoneme conversion, where they concluded that their model produced a lower PER score than the state-of-the-art approaches. However, the model posed two types of error; namely, the model generated unwanted phonemes multiple times, and the other is sparsely represented graphemes during the training stage. S. Yolchuyeva et al. in [53] improved the work mentioned above by using the Transformer model on the dataset. The authors concluded that the model produced a PER of 5.23 and a Phrase Error of 22.1. Similar to [52], the model still generated unwanted phonemes multiple times.

2.6 PHONEME-TO-SPECTROGRAM APPROACHES

English Speech Synthesis is complex research, where many researchers have tried to provide multiple approaches to solving shortcomings mentioned in Chapter 1. Table 2.6 describes the current state-of-the-art approaches for translating English Speech Synthesis in a detailed manner.

Table 2.6: Description of the current state-of-the-art for English Speech Synthesis

Paper	Datasets	Algorithms	Performance Measures	Results
Arik et al. in [4]	Blizzard 2013	Attention-based Seq2Seq, Wavenet-based	MOS	Val MOS = 4.65 ± 0.13, Test MOS = 2.67 ± 0.37
Arik et al. in [54]	Internal & VCTK dataset	Wavenet	MOS	MOS = 2.96 ± 0.38
Ping et al. in [55]	LibriSpeech ASR	Attention-based Seq2Seq, Convolutional Sequence Learning	MOS	MOS = 2.96 ± 0.38
Ren et al. in [56]	LJ Speech	Autoregressive Transformer	MOS	MOS = 3.84 ± 0.08

Arik et al. in [4] used Attention-based Seq2Seq, Wavenet-based models on Blizzard 2013 dataset, where the authors used MOS (Mean Opinion Score) as the performance measure. One of the problems faced by the authors is that the pipeline used was highly segmented, i.e., it was a multi-stage pipeline. The other problem was that the performance of the duration and frequency model was hindered due to the smaller dataset, which resulted in less than natural voice generation. Arik et al. in [54] improved the previous result by using the Wavenet model on an Internal dataset and VCTK dataset. The authors concluded that the model produced an MOS = 2.96 ± 0.38. The problem with this approach is that it was not able to produce results in near-real-time.

Ping et al. in [55] used Attention-based Seq2Seq, Convolutional Sequence Learning on LibriSpeech ASR dataset. The authors concluded that the model produced an MOS = 2.96 ± 0.38. The authors inferred that using a combined training with a neural vocoder and training on larger and cleaner datasets helped improve model performance. Ren et al. in [56] used Autoregressive Transformer on the LJ Speech dataset. The authors concluded that the model produced an MOS = 3.84 ± 0.08. The problem with this approach is that it does not work well with low-resource settings.

Chapter 3

PROPOSED WORK

This chapter describes the proposed method for converting ASL videos to English language speech. It also explains how the proposed solution coherently solves the problems faced by the methods mentioned in Chapter 2.

3.1 PIPELINE DESCRIPTION

The pipeline used for converting ASL videos to English language speech is given in Fig. 3.1. The pipeline consists of 4 modules, namely, 1) Video-to-Gloss module, 2) Gloss-to-Grapheme module, 3) Grapheme-to-Phoneme module, 4) Phoneme-to-Spectrogram module. The input for the system is a sentence/continuous sign language video, and the output of the system is English language speech. The pipeline used for training the neural network models in each module in Fig. 3.1 is given in Fig. 3.2. The steps followed in Fig. 3.2 for each module are explained below.

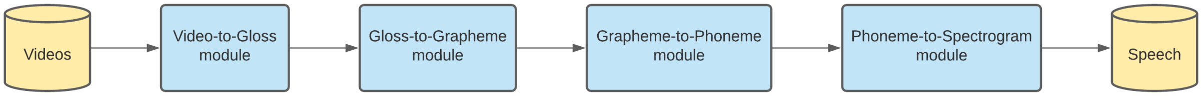

Figure 3.1: Pipeline for converting ASL videos to English language speech

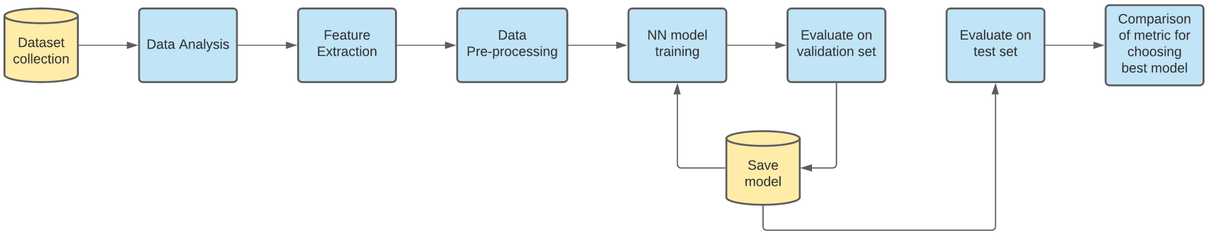

Figure 3.2: Pipeline for training the neural network models in each module

3.2 FEATURE EXTRACTION

The feature extraction step is used only in the Video-to-Gloss module. The features extracted from the videos are human pose 2D Keypoints. Since the frame size for each of the videos in the sign language dataset collected was different, the videos' frames had to be resized to a diagonal size of 256. We mainly extract 18 body pose keypoints [57], and 42 hand pose keypoints (21 for each hand) [58], i.e., a total of 60 keypoints. Since the video only covers the signer's upper midriff, the six keypoints below the midriff are dropped, i.e., hips, knees, and ankles.

The left-hand pose keypoint detected by [57] is also dropped and replaced with left-hand pose keypoints detected by [58]. It is because the keypoint provided/detected by [57] is approximately an average of the corresponding hand keypoints detected by [58]. In order to reduce the duplicate/value manipulated data and get better feature alignment, the keypoint is dropped/removed. A similar process is done for the right-hand side of the body. The facial keypoints detected by [57] are attached next to the nose keypoint detected. In other words, the keypoints are ordered in limb sequence fashion (sorted based on nearest keypoint). The ordered 2D keypoints are concatenated at each joint as the input feature. The end resulting array for a video would be of shape (t, k). The entire process is performed every single available video from the WLASL dataset.

3.3 DATA CONSTRUCTION

Similar to Chapter 3.2, the data construction step is also used only in the Video-to-Gloss module. This is one of our contributions to sign language translation research. The initial approach used for creating the phrase dataset consisted of using phrases from ASLG-PC12 dataset [59] and videos from the WLASL dataset [9]. However, when an intersection was performed on the ASL vocabulary from the ASLG-PC12 and WLASL datasets, the common vocabulary size was drastically reduced by almost 50%. Hence, the phrases generated are in random order with certain checkpoints.

The first checkpoint is the length of the phrase, i.e., since every frame from every video is considered, the length of the phrase randomly generated is set to 4. It is because of the memory constraint within the GPU (Graphics Processing Unit) during the model's training. The second checkpoint is that no word should be repeated in a phrase from ASL Gloss vocabulary. The third checkpoint is that the phrases' combination should not be repeated in the generated dataset. These checkpoints are used to ensure the resulting dataset consists of various phrases with different word combinations. The output phrase would consist of '<s>' as the starting token, and '</s>' as the ending token apart from the four words in the phrases. Similarly, the input would consist of frames from 4 videos; hence the starting frame ('<s>') consists of ones, the ending frame ('</s>') consists of twos, and the padding frame consists of zeros. The input (keypoints from a video) shape for the Seq2Seq models would be of shape (t, k).

3.4 DATA PRE-PROCESSING

3.4.1 GLOSS-TO-GRAPHEME MODULE

The text pre-processing step for the Gloss-to-Grapheme module consists of the following steps:

- Remove HTML (Hypertext Markup Language) lines from sentences.
- Lowercase all characters in the input and target lines.
- Perform Unicode to ASCII (American Standard Code for Information Interchange) text normalization.
- Remove all prefixes from words such as 'desc-', and 'x-'.
- Remove mathematical sentences from input and target lines.
- Remove brackets in symbol format and word format such as 'lrb', and 'rrb'.
- Remove duplicate translations for a sentence.
- Use POS tagging to extract and convert Proper Nouns from sentences.

- Input and Target lines are paired and shuffled. Divided the dataset into training, validation and testing set.
- Tokenize sentences using SentencePiece model [60] for the Seq2Seq models, and Subword Text Encoder for the Transformer models, where the new vocabulary size is set as approximately 4000.

Another contribution of ours to sign language translation research is the usage of POS-tagging-based approach to extract essential features such as proper nouns and rare words. During the data pre-processing stage in [48], the authors dropped words from vocabulary if the frequency is less than 5, which drastically reduced vocabulary size as mentioned in Chapter 2. To solve this problem, we use Spacy [61] to extract Proper Nouns from sentences and assign a random probability to the extracted words, provided if the word does not contain digits and only contains alphabets. If the assigned probability is greater than 0.5, the word is converted from 'stark' to '<s#t#a#r#k>,' i.e., unified format.

3.4.2 GRAPHEME-TO-PHONEME MODULE

The text pre-processing step for the Grapheme-to-Phoneme module consists of the following steps:

- Lowercase all characters in the input and target lines.
- Remove duplicate translations for a word.
- Input and Target lines are paired and shuffled. Divided the dataset into training, validation and testing set.

3.4.3 PHONEME-TO-SPECTROGRAM MODULE

The text pre-processing step for the Phoneme-to-Spectrogram module consists of the following steps:

- Lowercase all characters in the input lines.

- Text normalization is performed on input lines.
- The best Grapheme-to-Phoneme model is used for converting text normalized sentences into phoneme sequences.
- Certain longer sequences are dropped.

3.5 NEURAL NETWORK ARCHITECTURE

There are two main types of neural network architectures used for building the neural network models for each module; namely Seq2Seq architecture [13], and Transformer architecture [17]. The Seq2Seq architecture was mainly tested with two Attention mechanisms, namely, Bahdanau Attention [15], and Luong Attention [16]. The hyperparameters tuned for each of these models in all the modules are given in Table 3.1. The model configuration used for developing all the models in each module is given in Table 3.2.

Table 3.1: Hyperparameters tuned for all the models

Parameters	**Values**
Seq2Seq Attention mechanism	Bahdanau/Luong
Encoder Layers (N)	1/2/3/4
Decoder Layers (N)	1/2/3/4
Transformer dm	512/1024
h	8/16
ff	2048/4096
Dropout	0.1/0.3
Seq2Seq Encoder layer types	Bi-LSTM/Uni-LSTM
Seq2Seq dm	Bi- LSTM = 256/Uni-LSTM = 512

For all the Luong Attention-based Seq2Seq models and Bahdanau Attention-based Seq2Seq models, the number of units in each uni-directional LSTM layer was set as 512, the number of units in each bi-directional LSTM layer as 256, and the dropout value during training stage as 0.3. The dropout value for Transformer model was set as 0.1 when $dm = 512$, $ff = 2048$, $h = 8$, and 0.3 when $dm = 1024$, $ff = 4096$, $h = 16$. The learning rate for all the Luong Attention-based

Table 3.2: Model Configuration used for training models

Model	Configuration	Name
Bahdanau	(1 Uni x 1 Uni) LSTM	b-1
	(2 Uni x 2 Uni) LSTM	b-2
	(1 Bi x 2 Uni) LSTM	b-3
	(3 Uni x 3 Uni) LSTM	b-4
	((1 Bi, 1 Uni) x 3 Uni) LSTM	b-5
	(4 Uni x 4 Uni) LSTM	b-6
	((1 Bi, 2 Uni) x 4 Uni) LSTM	b-7
	((1 Bi, 2 Uni) x 4 Uni) LSTM & RC	b-8
Luong	(1 Uni x 1 Uni) LSTM	l-1
	(2 Uni x 2 Uni) LSTM	l-2
	(1 Bi x 2 Uni) LSTM	l-3
	(3 Uni x 3 Uni) LSTM	l-4
	((1 Bi, 1 Uni) x 3 Uni) LSTM	l-5
	(4 Uni x 4 Uni) LSTM	l-6
	((1 Bi, 2 Uni) x 4 Uni) LSTM	l-7
	((1 Bi, 2 Uni) x 4 Uni) LSTM & RC	l-8
Transformer	(1 x 1), dm = 512, ff = 2048, h = 8	t-1
	(2 x 2), dm = 512, ff = 2048, h = 8	t-2
	(3 x 3), dm = 512, ff = 2048, h = 8	t-3
	(4 x 4), dm = 512, ff = 2048, h = 8	t-4
	(1 x 1), dm = 1024, ff = 4096, h = 16	t-5
	(2 x 2), dm = 1024, ff = 4096, h = 16	t-6
	(3 x 3), dm = 1024, ff = 4096, h = 16	t-7
	(4 x 4), dm = 1024, ff = 4096, h = 16	t-8

Seq2Seq models and Bahdanau Attention-based Seq2Seq models was set as 0.001. The learning rate schedule used for the Transformer model is given in (3.1) [17], beta_1 = 0.9, beta_2 = 0.98, and epsilon = 1e-9.

$$rate = dm^{-0.5} * min(step\_num^{-0.5}, warmup\_step^{-1.5}) \tag{3.1}$$

In the bi-directional LSTM, the first layer traverses from left-to-right (one forward LSTM layer), while the next layer traverses from right-to-left (one backward LSTM layer). A bi-directional layer was chosen as a hyperparameter only for the first layer in the encoder model because it helps obtain the context and other essential features such as the description of context and its changes across timesteps [15]. The equations used for calculating the bi-directional layer's output and states are given in (3.2).

$$
\begin{aligned}
x_t^f, m_t^f, c_t^f &= LSTM_f(x_t^0, m_t^0, c_t^0; W^f) \\
x_t^b, m_t^b, c_t^b &= LSTM_b(x_t^0, m_t^0, c_t^0; W^b) \\
x_t^1 &= concat(x_t^f, x_t^b) \\
m_t^1 &= concat(m_t^f, m_t^b) \\
c_t^1 &= concat(c_t^f, c_t^b)
\end{aligned} \tag{3.2}
$$

Both the encoder model and the decoder model contain residual connections between the 3rd and 4th LSTM layers. These connections are used because when more LSTM layers are stacked together, the model suffers from vanishing gradient problems [62], [63]. In other words, the deeper the model, the more it forgets about the information it has seen previously. The equation used for calculating the residual output is given in (3.3). For all the models, Adam [64] was used as the optimizer during the training stage.

$$
\begin{aligned}
x_t^i, m_t^i, c_t^i &= LSTM_i(x_t^{i-1}, m_t^{i-1}, c_t^{i-1}; W^i) \\
x_t^i &= x_t^i + x_t^{i-1} \\
m_t^i &= m_t^i + m_t^{i-1} \\
c_t^i &= c_t^i + c_t^{i-1} \\
x_t^{i+1}, m_t^{i+1}, c_t^{i+1} &= LSTM_{i+1}(x_t^i, m_t^i, c_t^i; W^{i+1})
\end{aligned} \tag{3.3}
$$

3.5.1 VIDEO-TO-GLOSS MODULE

The Luong Attention-based Seq2Seq model was used for developing the Video-to-Gloss module, where the encoder had input as keypoints from Chapter 3.3, and target as ASL Gloss phrase. The l-1 architecture used for developing the Video-to-Gloss module is given in Fig. 3.3. The encoder model consists of only a LSTM layer(s) which changes as per the configuration in Table 3.2. The decoder model consists of a word embedding layer of size 512, LSTM layer(s) of size 512, followed by the Luong Attention layer, a Dense layer with 'tanh' activation layer, and a Dense layer with 'softmax' activation layer, which changes as per the configuration in Table 3.2. The models were trained for 30 epochs with a batch size of 50.

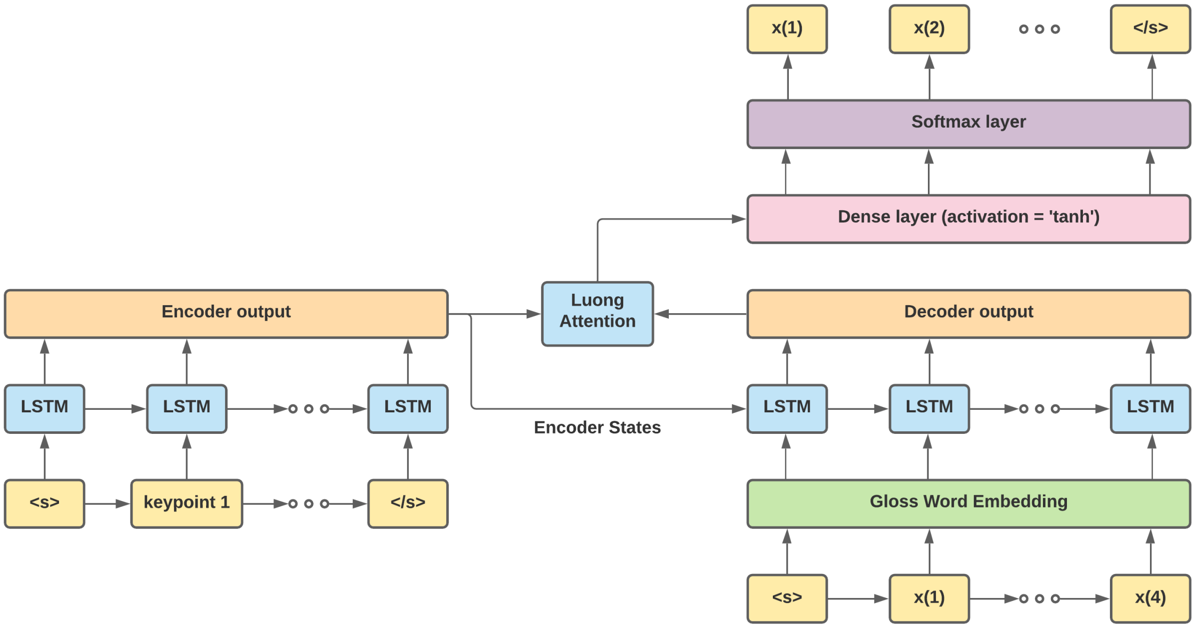

Figure 3.3: l-1 architecture used for developing the Video-to-Gloss module

3.5.2 GLOSS-TO-GRAPHEME MODULE

The Luong Attention-based Seq2Seq model, Bahdanau Attention-based Seq2Seq model, and Transformer model were used to develop the Gloss-to-Grapheme module. The Seq2Seq models are similar to the ones mentioned in Chapter 3.5.1, except for the encoder model in Chapter 3.5.1, which did not have a word embedding layer as the input was keypoints. However, since this module has both input and output as text, the encoder model has a word embedding layer. The Bahdanau Attention-based Seq2Seq has an encoder model similar to the Luong Attention-based Seq2Seq mentioned above. The decoder model consists of a word embedding layer of size 512, Bahdanau

Attention layer, LSTM layer(s) of size 512, and a Dense layer with ’softmax’ activation layer, which changes as per the configuration in Table 3.2. The b-1 architecture used for developing the Gloss-to-Grapheme module is given in Fig. 3.4. The warmup step count for all the Transformer models developed was 4000.

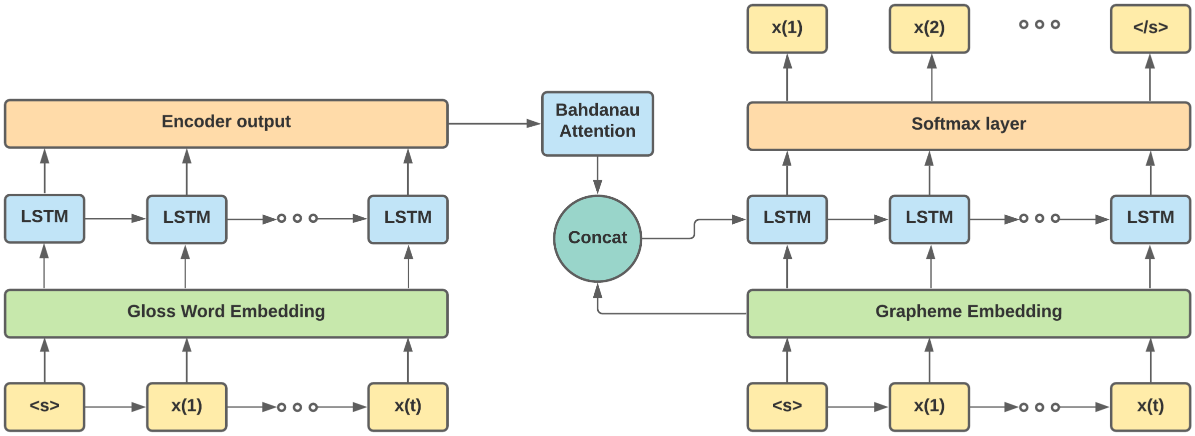

Figure 3.4: b-1 architecture used for developing the Gloss-to-Grapheme module

3.5.3 GRAPHEME-TO-PHONEME MODULE

The Luong Attention-based Seq2Seq model, and Bahdanau Attention-based Seq2Seq model were used to develop the Grapheme-to-Phoneme module. Both the Luong Attention-based Seq2Seq model, and Bahdanau Attention-based Seq2Seq model have a similar configuration to the models in Chapter 3.5.2.

3.5.4 PHONEME-TO-SPECTROGRAM MODULE

The Transformer based architecture was used to develop the Phoneme-to-Spectrogram module. The Transformer model consisted of an Encoder, Decoder, Decoder Pre-Net, and Decoder Post-Net. The Encoder and Decoder models consisted of fully connected layers with Multi-Head Attention, where the number of layers was set as 4, number of heads as 8, number of units in each layer in Encoder model as 512, and number of units in each layer in Decoder model as 256. The

Decoder Pre-net is a DNN consisting of two fully connected layers, with dimension size as 256, and activation as ’relu.’ The Decoder Post-net consists of 2 linear projections (mel-linear, and stop-linear) and a post-net, for which a 5-layer CNN is used to help in the better reconstruction of mel spectrogram. The model was trained for 200k training steps with a learning rate of 0.0001.

Chapter 4

EXPERIMENTAL EVALUATION

This chapter describes in details the datasets used, performance measures used, and results obtained on using methodology mentioned in Chapter 3.

4.1 DATASET DESCRIPTION

4.1.1 WLASL (Word-level American Sign Language) DATASET

The WLASL dataset [9], contains 21083 videos with a unique gloss count of 2000. The dataset comprises of signs performed by 119 signers, with a mean number of 10.5 videos per gloss. The length of videos in the dataset ranges from 0.36 seconds to 8.12 seconds, where the average length of videos is 2.41 seconds. The authors of the paper also split the dataset into 4 subsets based on the unique ASL gloss count, $glosscount = \{100, 300, 1000, 2000\}$. The total duration of the video dataset is approximately 14 hours.

4.1.2 ASLG-PC12 DATASET

It consists of 87,710 examples in the dataset [59]. The American Sign Language vocabulary size is approximately 15k and the English Vocabulary size is approximately 21k.

4.1.3 CMUDICT DATASET

The CMUDict dataset [65] is an open-source machine-readable pronunciation dictionary for North American English that contains over 134,000 words and their pronunciations. Its entries are particularly useful for speech recognition and synthesis, as it has mappings from words to their pronunciations in the ARPAbet phoneme set, a standard for English pronunciation. The current phoneme set contains 39 phonemes.

4.1.4 LJSPEECH DATASET

It is a public domain speech dataset consisting of 13,100 short audio clips of a single speaker reading passages from 7 non-fiction books [66]. A transcription is provided for each clip. Clips vary in length from 1 to 10 seconds and have a total length of approximately 24 hours.

4.2 PERFORMANCE MEASURES

4.2.1 SPARSE CATEGORICAL TOP K-ACCURACY

It calculates the number of times the integer target classes are in the top k predictions produced by the neural network model. In this thesis, we use three different k values for estimating the quality of the model, i.e., k = 1, 5, 10.

4.2.2 PS (Precision Score)

Precision is the fragment of correctly predicted positive statements to the total predicted positive statements as given in (4.1).

$$Precision = \frac{TP}{TP+FP} \tag{4.1}$$

4.2.3 BLEU (Bi-Lingual Evaluation Understudy)

BLEU is used to calculate the variation between the machine-translated output, and human translated output [67]. It works on complimenting n-grams in the machine-translated output to n-grams in the human-translated output. The score ranges from 0.0 to 1.0 or 0% to 100%.

4.2.4 METEOR (Metric for Evaluation of Translation with Explicit Ordering)

METEOR is used for evaluating the quality of the machine-translated output, where the quality is estimated based on the harmonic mean of unigram precision and recall [68]. The weight of recall is higher than precision.

4.2.5 WER (Word Error Rate)

WER is used for evaluating the performance of the neural network model at the word level. It is used in the field of Machine Translation and Speech Recognition. It is evaluated with Levenshtein distance [69], where the result is divided by the length reference or human-translated output.

4.2.6 PER (Phoneme Error Rate)

PER is similar to WER, where the model's performance is evaluated at the phoneme level instead of word level, and the output from the Levenshtein distance [69] is divided by the number of phonemes in the reference or human-translated output.

4.2.7 SER (Sentence/Phrase Error Rate)

It is the fraction of incorrectly translated phrases to the total number of phrases.

4.2.8 MOS (Mean Opinion Score)

MOS is a human evaluated quality of an event, i.e., the voice in text-to-speech systems. It is the mean of multiple human-evaluated parameters. It ranges from 1 (worst) to 5 (excellent).

4.3 RESULTS

4.3.1 VIDEO-TO-GLOSS MODULE

The details of the WLASL dataset used for developing phrase dataset for training the Video-to-Gloss model are given in Table 4.1. The details of the dataset used for training the Video-to-Gloss models are given in Table 4.2.

The gloss-level performance of the optimization of the models mentioned in Chapter 3.5.1 for the testing set is given in Table 4.3. Table 4.3 also includes the ablation study on the performance of the models. The phrase-level performance of the optimization of the models mentioned Chapter

Table 4.1: Details of the WLASL dataset used for developing phrase dataset for training the Video-to-Gloss models

Dataset	**WLASL**			
	100	**300**	**1000**	**2000**
Original	2308	5118	13174	21095
Files available	1963	4955	12851	20645
Feature Extraction	1664	4278	11077	17721

Table 4.2: Details of the phrase dataset used for training the Video-to-Gloss models

Dataset	**No. of Phrases**		
	Train	**Validation**	**Test**
WLASL-100	150000	2000	2000
WLASL-300	200000	2000	2000
WLASL-1000	250000	2000	2000
WLASL-2000	250000	2000	2000

3.5.1 for the testing set is given in Table 4.4. Table 4.4 also includes the ablation study on the performance of the models.

4.3.2 GLOSS-TO-GRAPHEME MODULE

The details of the dataset used for training the Gloss-to-Grapheme model are given in Table 4.5. It also contains the details of the dataset after performing the data pre-processing mentioned in Chapter 3.4.

Table 4.3: Gloss-level performance of the Video-to-Gloss models on the Testing set

Model	**Config**	**WLASL-100**			**WLASL-300**			**WLASL-1000**			**WLASL-2000**		
		Top-1	**Top-5**	**Top-10**	**Top-1**	**Top-5**	**Top-10**	**Top-1**	**Top-5**	**Top-10**	**Top-1**	**Top-5**	**Top-10**
Luong	l-1	89.65	94.13	95.41	81.25	89.9	91.97	68.8	81.35	85.23	66.53	80.42	84.45
	l-2	95.77	97.24	97.66	96.54	97.96	98.31	96.05	97.71	98.0	94.43	97.63	97.87
	l-3	89.73	94.3	95.54	92.56	95.76	96.51	84.93	92.01	93.75	81.58	90.75	92.3
	l-4	96.46	97.72	97.97	96.9	98.14	98.46	96.2	97.93	98.28	94.55	97.74	98.14
	l-5	96.39	97.65	97.9	96.43	97.78	98.16	95.66	97.47	97.84	94.31	97.44	97.75
	l-6	96.47	97.58	97.94	96.86	**98.27**	98.46	96.24	**98.29**	**98.53**	**94.75**	**97.92**	**98.3**
	l-7	**96.71**	**97.82**	**98.02**	**97.05**	98.17	**98.5**	**96.41**	98.05	98.31	94.5	97.62	98.02
	l-8	95.69	97.19	97.67	95.82	97.2	97.72	95.89	97.61	97.94	94.68	97.77	98.04

Table 4.4: Phrase-level performance of the Video-to-Gloss models on the Testing set

Model	**Config**	**WLASL-100**			**WLASL-300**			**WLASL-1000**			**WLASL-2000**		
		WER	**SER**	**BLEU**	**WER**	**SER**	**BLEU**	**WER**	**SER**	**BLEU**	**WER**	**SER**	**BLEU**
Luong	l-1	21.84	57.95	56.25	35.71	83.75	30.6	55.02	96.95	11.14	56.54	96.4	12.22
	l-2	8.0	23.5	84.91	6.35	19.05	87.15	6.89	20.15	86.33	9.57	28.0	81.56
	l-3	29.18	70.9	43.99	20.02	57.05	56.67	32.98	77.8	36.67	35.54	80.75	33.99
	l-4	6.29	19.8	87.18	6.52	19.7	86.98	7.87	22.4	85.13	10.07	29.7	80.43
	l-5	6.46	20.2	86.79	6.1	18.95	87.46	7.31	**21.12**	**85.56**	10.98	31.05	79.46
	l-6	**5.99**	**18.8**	**87.78**	6.22	18.75	87.59	**7.5**	21.9	85.61	10.93	31.2	79.23
	l-7	6.82	20.7	86.53	**5.6**	**17.3**	**88.73**	7.66	21.65	85.6	10.13	28.45	81.07
	l-8	7.71	23.35	84.46	7.44	22.6	84.7	7.55	21.85	85.34	**9.51**	**28.0**	**81.78**

Table 4.5: Details of the dataset used for training the Gloss-to-Grapheme model

Dataset	No. of lines	Words		Characters	
		Gloss	Grapheme	Gloss	Grapheme
Original	87710	16120	22071	64	125
Cleaned	87114	15616	20773	43	43
Non-Duplicates	80420	15616	20773	43	43
POS-Tagging	80420	17620	22402	46	46
Train	76420	16880	21933	46	46
Validation	2000	3291	4078	44	45
Test	2000	3369	4200	44	45

The performance of the optimization of the models mentioned in Chapter 3.5.2 for the testing set is given in Table 4.6. Table 4.6 also includes the ablation study on the performance of the models. The performance of the models on the testing set based on length of input sentence for WER and BLEU is given in Table 4.7. The performance of the models on the testing set based on length of input sentence for PS and METEOR is given in Table 4.8.

4.3.3 GRAPHEME-TO-PHONEME MODULE

The details of the dataset used for training the Gloss-to-Grapheme model are given in Table 4.9. It also contains the details of the dataset after performing the data pre-processing mentioned in Chapter 3.4.

The performance of the optimization of the models mentioned in Chapter 3.5.3 for the testing set is given in Table 4.10. Table 4.10 also includes the ablation study on the performance of the models.

4.3.4 PHONEME-TO-SPECTROGRAM MODULE

The details of the dataset used for training the Gloss-to-Grapheme model are given in Table 4.11. It also contains the details of the dataset after performing the data pre-processing mentioned in

Table 4.6: Performance of the Gloss-to-Grapheme models on the Testing set

Model	Config	WER	BLEU	PS	METEOR
Bahdanau	b-1	51.59	43.59	48.24	64.31
	b-2	49.14	40.12	40.12	65.41
	b-3	**24.11**	**59.87**	**64.5**	**84.32**
	b-4	34.02	53.36	58.36	75.49
	b-5	109.1	14.51	14.51	39.86
	b-6	103.62	7.67	7.67	24.31
	b-7	37.07	50.34	56.93	71.23
	b-8	114.8	11.16	11.16	33.71
Luong	l-1	26.97	61.54	61.54	86.04
	l-2	28.88	58.3	63.28	81.54
	l-3	**20.87**	**66.87**	69.97	**88.1**
	l-4	30.96	56.53	61.42	78.86
	l-5	34.21	60.72	62.34	80.77
	l-6	85.74	5.38	5.4	22.39
	l-7	27.66	65.19	67.63	84.51
	l-8	26.01	63.05	**70.18**	83.15
Transformer	t-1	14.41	72.76	74.93	92.98
	t-2	12.89	75.12	77.25	94
	t-3	**12.35**	**76.09**	**78.47**	**93.97**
	t-4	12.54	75.58	78.57	93.66
	t-5	13.52	73.89	76.91	93.12
	t-6	12.7	75.01	78.39	93.38
	t-7	12.37	75.45	78.57	93.64
	t-8	14.23	73.04	76.34	92.4

Chapter 3.4. The model results was evaluated with ten human testers, and the testers reported an MOS of 3.67 ± 0.06.

4.4 SENSITIVITY TO PARAMETERS

4.4.1 VIDEO-TO-GLOSS MODULE

It can be observed from Table 4.3 that as the number of layers in the uni-directional layer models (i.e., l-[1, 2, 4, 6]) increases, the top-1 accuracy increases. A similar pattern can be observed for all the models except for WLASL-100 in top-5 and top-10 accuracy. For WLASL-100, the top-5 and top-10 accuracy increase until 3 layers and drops when the number of layers is 4. Similarly,

Table 4.7: Performance of the models on the testing set based on length of input sentence for WER and BLEU metrics

Model	**Config**	**WER**				**BLEU**			
		0-10	**10-20**	**20-30**	**30-40**	**0-10**	**10-20**	**20-30**	**30-40**
Bahdanau	b-1	56.21	49.03	72.26	93.09	44.06	44.02	16.54	0
	b-2	45.03	50.93	79.3	83.41	45.97	38.49	16.9	3.4
	b-3	**23.54**	**24.15**	43.55	75.04	**62.13**	**59.77**	41.69	7.13
	b-4	39.89	30.94	**36.59**	**72.45**	48.89	54.74	**46.16**	9.63
	b-5	98.2	114.68	104.83	98.65	23.53	11.86	5.36	0
	b-6	100.71	105.14	102.2	91.74	10.71	6.58	5.63	0
	b-7	40	35.4	49.69	76.43	48.84	50.98	37.49	**12.38**
	b-8	106.67	119.13	90.69	90.39	18.05	9.17	7.03	0
Luong	l-1	28.3	26.13	39.61	61.37	60.09	61.94	42.65	23.97
	l-2	33.89	26.24	32.87	50.3	54.89	59.39	50.89	31.23
	l-3	**17.98**	**22.11**	36.83	65.54	**70.67**	66.31	53.84	30.93
	l-4	36.23	28.18	37.21	**50.23**	52.53	57.72	49.28	33.13
	l-5	44.71	28.85	31.81	57.17	52.65	62.02	59.33	25.06
	l-6	84.34	86.45	85.22	93.17	6.93	5.02	4.68	0
	l-7	36.72	23.07	**20.04**	48.95	58.82	**66.76**	**66.12**	**34.08**
	l-8	31.03	23.39	26.63	51.69	58.92	64.23	63.75	31.06
Transformer	t-1	13.9	14.53	19.02	51.88	73.63	72.84	68.73	33.48
	t-2	11.59	13.41	20.04	40.84	77.25	74.92	66.46	45.98
	t-3	11.86	**12.47**	16.18	42.12	**77.5**	**76.03**	70.94	44.96
	t-4	11.85	12.78	15.56	**39.41**	77	75.5	**72.29**	**46.4**
	t-5	12.58	13.88	18.93	39.45	75.76	73.74	66.3	46.79
	t-6	11.61	13.14	**15.1**	42.23	77.12	74.78	71.9	44.05
	t-7	**11.4**	12.72	19.21	43.58	76.97	75.4	69.6	39.62
	t-8	13.07	14.7	18.26	44.93	74.12	73.1	66.48	34.92

as the number of layers in the bi-directional layer models (i.e., l-[3, 5, 7]) increases, the top-1, top-5, and top-10 accuracy increases. It can be inferred from Table 4.3 that the influence of the bi-directional layer in the model increases as the number of layers increases. However, adding residual connections does not help in increasing the top-1, top-5, and top-10 accuracy. It can also be observed from Table 4.3, that for WLASL-100, WLASL-300, and WLASL-1000, the best model in terms of gloss accuracy is l-7; and for WLASL-2000, the best model in terms of gloss accuracy is l-6.

Similar to Table 4.3, it can be observed from Table 4.4, that as the number of layers in the

Table 4.8: Performance of the models on the testing set based on length of input sentence for PS and METEOR metrics

Model	**Config**	**PS**				**METEOR**			
		0-10	**10-20**	**20-30**	**30-40**	**0-10**	**10-20**	**20-30**	**30-40**
Bahdanau	b-1	48.02	48.65	21.45	0	60.37	66.6	37.71	5.07
	b-2	45.97	38.92	18.4	10.7	71.44	62.7	32.43	14.28
	b-3	**65.57**	**64.38**	**53.39**	**54.92**	**83.66**	**84.91**	65.46	25.62
	b-4	54.24	59.41	52.89	30.32	68.91	78.95	**71.19**	**34.04**
	b-5	23.53	11.86	9.3	0	55.79	32.03	16.31	9.08
	b-6	10.71	6.72	7.07	0	32.15	20.44	15.31	6
	b-7	52.49	58.11	51.38	41.6	68.06	73.04	59.47	26.58
	b-8	18.05	9.17	7.69	0	46.55	27.36	19.48	11.13
Luong	l-1	60.09	62.02	51.38	60.22	86.6	85.97	71.92	40.1
	l-2	59.59	64.24	56.24	74.68	76.7	84.11	79.35	53.79
	l-3	**72.72**	69.46	60.11	59.78	**89.38**	**87.67**	71.19	44.47
	l-4	57.58	62.31	59.03	79.21	73.69	81.6	72.7	51.39
	l-5	52.65	64.86	65.24	66.3	75.28	83.64	79.99	47.72
	l-6	7.06	5.02	5.12	0	22.37	22.44	20.39	9.29
	l-7	58.82	**69.73**	**73.45**	**81.5**	78.52	87.61	**85.59**	**55.25**
	l-8	67.07	70.8	75.49	78.01	78.27	85.73	78.4	**55.25**
Transformer	t-1	74.75	75.05	72.74	62.37	92.67	93.27	88.36	60.49
	t-2	78.63	76.99	71.68	**85.67**	**93.98**	94.13	**90.95**	61.58
	t-3	78.9	78.4	77.62	83.76	93.69	**94.24**	90.31	61.55
	t-4	79.31	**78.41**	**79.5**	80.68	93.39	93.94	88.45	61.92
	t-5	78.03	76.7	72.55	84.15	93	93.3	88.79	61.38
	t-6	**79.74**	78.09	78.67	82.06	93.31	93.53	90.16	61.33
	t-7	79.3	78.5	73.33	71.26	93.51	93.82	88.64	**65.31**
	t-8	76.75	76.3	75.38	62.8	92.61	92.42	87.83	64.4

Table 4.9: Details of the dataset used for training the Grapheme-to-Phoneme model

Dataset	**No. of lines**
Original	134374
Cleaned	134373
Duplicates Dropped	130808
Train	126808
Validation	2000
Test	2000

uni-directional layer models for WLASL-100 (i.e. l-[1, 2, 4, 6]) increases, WER, and SER decreases, and BLEU increases; for WLASL-300 and WLASL-1000, WER, and SER decreases, and

Table 4.10: Performance of the Grapheme-to-Phoneme models on the Testing set

Model	Config	PER	SER	BLEU	METEOR
Bahdanau	b-1	8.17	35.17	82.2	90.32
	b-2	7.86	33.97	83.13	90.76
	b-3	7.92	33.72	83.17	90.73
	b-4	7.86	34.32	82.73	90.85
	b-5	7.74	33.52	83.2	91.02
	b-6	7.85	33.07	83.19	90.86
	b-7	8.1	34.22	82.94	90.57
	b-8	**7.5**	**33.57**	**83.39**	**91.07**
Luong	l-1	8.31	35.32	81.73	90.4
	l-2	8.39	35.42	81.94	90.22
	l-3	8.64	36.32	81.63	89.85
	l-4	8.02	34.57	82.36	90.73
	l-5	7.66	33.12	83.9	91.04
	l-6	8.32	35.67	82	90.4
	l-7	**7.2**	**31.76**	**84.34**	**91.6**
	l-8	7.85	33.42	83.17	90.76

Table 4.11: Details of the dataset used for training the Phoneme-to-Spectrogram model

Dataset	No. of lines
Original	13100
Cleaned	8508
Train	7808
Validation	400
Test	400

BLEU increases except when number of layers = 3; for WLASL-2000, WER, and SER increases, and BLEU decreases, indicating that as the vocabulary size increases the increase in number of layers in the uni-directional models has a bad influence on the performance. Similarly, when the number of layers in the bi-directional layer models (i.e., l-[3, 5, 7]) for WLASL-300, and WLASL-2000 increases, WER, and SER decreases, and BLEU increases; WLASL-100 and WLASL-1000, WER, and SER decreases, and BLEU increases except when number of layers equal to 3, where it reverses.

Unlike, gloss performance in Table 4.3, adding residual connections helped in reducing the WER, and SER, and increasing BLEU for WLASL-2000 dataset. It can also be observed from Table 4.4, that for WLASL-100, the best model in terms of phrase level performance is l-6. Similarly, for WLASL-300 it is l-7, WLASL-1000 it is l-5, and WLASL-2000 it is l-8.

4.4.2 GLOSS-TO-GRAPHEME MODULE

It can be observed from Table 4.6 that the transformer models' performance is better than the Luong Attention-based Seq2Seq models and the Bahdanau Attention-based Seq2Seq models. When compared within the Seq2Seq models, the Luong Attention-based Seq2Seq models performed much better than the Bahdanau Attention-based Seq2Seq models.

It can be observed from Table 4.6 that as the number of layers in the uni-directional layer models, (i.e., b-[1, 2, 4, 6] and l-[1, 2, 4, 6]) increases, the performance of the models decreases, except for n_layers = 2. Similarly, as the number of layers in the bi-directional layer models (i.e., b-[3, 5, 7], and l-[3, 5, 7]) increases, the performance of the models decreases. It can be observed from the Luong Attention-based Seq2Seq models that the residual connections helped in the model's performance. The two types of the Transformer models (i.e., t-[1, 2, 3, 4], and t-[5, 6, 7, 8]) have performed similarly, i.e., the performance improves as the number of layers increases, the performance of the increases except when n_layers is 4. It can be observed from Tables 4.7, and 4.8 that as the length of the input sentence increases the performance of the all the models decreases. However, the performance drop in the Transformer model is less than the Luong Attention-based Seq2Seq models and the Bahdanau Attention-based Seq2Seq models.

It can be observed that for the Bahdanau Attention-based Seq2Seq model, the b-3 configuration produces the best results. Similar to the Luong Attention-based Seq2Seq model and the Transformer model, l-3 and t-3 configurations produce the best results.

4.4.3 GRAPHEME-TO-PHONEME MODULE

It can be observed from Table 4.10, that the Luong Attention-based Seq2Seq models performed much better the Bahdanau Attention-based Seq2Seq models. For Bahdanau Attention-based Seq2Seq models, it can be observed that as the number of layers in the uni-directional layer models (i.e., b-[1, 2, 4, 6]) increases, the performance of the models increases, except for n_layers = 3. Similarly, as the number of layers in the bi-directional layer models (i.e., b-[3, 5, 7]) increases, the performance of the model increases, except for n_layers = 4. It can also be observed that residual connections helped in improving the performance of the model. For Luong Attention-based Seq2Seq models, it can be observed that as the number of layers in the uni-directional layer models (i.e., l-[1, 2, 4, 6]) increases, the performance of the models decreases. Similarly, as the number of layers in the bi-directional layer models (i.e., l-[3, 5, 7]) increases, the performance of the models increases. It can be observed that the RC did not improve the performance of the model.

It can be observed that for the Bahdanau Attention-based Seq2Seq model, the b-8 configuration produced the best results. Similarly, for the Luong Attention-based Seq2Seq model, the l-7 configuration produced the best results.

4.5 COMPARISON TO COMPETITORS

4.5.1 VIDEO-TO-GLOSS MODULE

The gloss-level performance comparison with competitors on the WLASL dataset is given in Table 4.12.

It can be observed from Table 4.12, that our model i.e. Luong Attention-based Seq2Seq approach is performing better than the current state-of-the-art model i.e. I3D from [9]. It can be because of the data augmentation method used on the WLASL dataset, i.e., creation of phrases using random index generation with checkpoints mentioned in Chapter 3.3, reordering of keypoint in limb sequence fashion, and the model architecture proposed in Chapter 3.5.1.

Table 4.12: Gloss-level performance comparison with competitors on the WLASL dataset

Dataset	Model	Accuracy		
		Top-1	Top-5	Top-10
WLASL-100	I3D in [9]	65.89	84.11	89.92
	Our model (l-7)	**96.71**	**97.82**	**98.02**
WLASL-300	I3D in [9]	56.14	79.94	86.98
	Our model (l-7)	**97.05**	**98.17**	**98.5**
WLASL-1000	I3D in [9]	47.33	76.44	84.33
	Our model (l-7)	**96.41**	**98.05**	**98.31**
WLASL-2000	I3D in [9]	32.48	57.31	66.31
	Our model (l-6)	**94.75**	**97.92**	**98.3**

4.5.2 GLOSS-TO-GRAPHEME MODULE

The performance comparison with competitors on the ASLG-PC12 dataset based on metrics is given in Table 4.13. The performance comparison with competitors on the ASLG-PC12 dataset based on sample sentences are given in Table 4.14.

Table 4.13: Performance comparison with competitors on the ASLG-PC12 dataset based on metrics

Model	English Vocab size	BLEU	METEOR
Seq2Seq in [47]	-	65.9	-
Transformer Ensemble in [48]	7712	82.87	96.60
Our model (t-3)	20773	76.09	93.97

In Table 4.13, it can be observed that our model works better than the best model in [47]. However, the performance of our model is slightly lower than [48]. It can be observed that our model has a more extensive vocabulary than the best model in [48], i.e., it is approximately three times larger. Multiple research works show that as the vocabulary size in the Natural Language Processing problems increases, the model's performance drastically decreases. However, it can be observed that our model only slightly underperforms the best model in [48] but handles a more extensive vocabulary size. It can be observed from Table 4.14 that our model produces slightly better sentences than the ones produced by the best model from [48]. It should also be noted that

Table 4.14: Performance comparison with competitors on the ASLG-PC12 dataset based on sample sentences

S. No.	Type	Sentence
1	ASL	this pressure be desc-particularly desc-great along union x-poss desc-sourn and desc-eastern border
	Ground Truth	this pressure is particularly great along the union's southern and eastern borders .
	Transformer [48]	this pressure is particularly great along the union's southern and eastern borders .
	Our model (t-3)	this pressure is particularly great along the union's southern and eastern borders
2	ASL	more woman die from aggression desc-direct against x-y than die from cancer .
	Ground Truth	more women die from **the** aggression directed against **them** than die from cancer .
	Transformer [48]	more women die from aggression directed against them than die from cancer .
	Our model (t-3)	more women die from the aggression directed against **y** than die from cancer .
3	ASL	x-it fuel war in cambodium in 1990 and x-it be enemy democracy
	Ground Truth	it fuelled the war in cambodia in **the** 1990s and it is the enemy **of** democracy .
	Transformer [48]	it **fuel** war in the **cambodium** in **1990** and it is an enemy of democracy .
	Our model (t-3)	it fuelled war in cambodia in 1990s and it is an enemy democracy
4	ASL	desc-n chief investigator x-himself be target and house card collapse .
	Ground Truth	then **the** chief investigator himself is targeted and **the** house of cards collapses .
	Transformer [48]	then chief investigator himself is a target and a house card collapse .
	Our model (t-3)	then the chief investigator himself is targeted and **housecards collapse** .

the sentences in Table 4.14 were not a part of the training set.

4.5.3 GRAPHEME-TO-PHONEME MODULE

The performance comparison with competitors on the CMUDict dataset based on metrics is given in Table 4.15. The performance comparison with competitors on the CMUDict dataset based on sample sentences are given in Table 4.16.

Table 4.15: Performance comparison with competitors on the CMUDict dataset based on metrics

Model	PER	Phrase Error
Transformer model [53]	5.23	22.1
Our model (l-7)	7.2	31.76

Table 4.16: Performance comparison with competitors on the CMUDict dataset based on sample words

S.No.	Type	Word
1	Grapheme	n a t i o n a l i z a t i o n .
	Ground Truth	n ae sh ah n ah l ah z ey sh ah n
	Transformer [53]	n ae sh **n ah** l ah z ey sh ah n
	Our model (l-7)	n ae sh ah n ah l ah z ey sh ah n
2	Grapheme	k o r z e n i e w s k i
	Ground Truth	k ao r z ah n uw f s k iy
	Transformer [53]	k **er** z ah n uw **s** k iy
	Our model (l-7)	k **er** z **ih** n uw **s** k iy
3	Grapheme	g r a n d f a t h e r s
	Ground Truth	g r ae n d f aa dh er z
	Transformer [53]	g r ae **n f** aa dh er z
	Our model (l-7)	g r ae n d f aa dh er z

It can be seen from Table 4.15 that our model did not outperform the current state-of-the-art in terms of the metrics. However, it can be observed from Table 4.16 that our model produces better translations in terms of the sample words. It should be noted that the words used in Table 4.16 were not a part of the training set used for training the models.

4.5.4 PHONEME-TO-SPECTROGRAM MODULE

The performance comparison with competitors on the LJSpeech dataset based on metrics is given in Table 4.17. It can be observed from Table 4.17 that our model underperforms than the current state-of-the-art.

Table 4.17: Performance comparison with competitors on the LJSpeech dataset based on metrics

Model	**MOS**
Autoregressive Transformer model [56]	3.84 ± 0.08
Our model	3.67 ± 0.06

Chapter 5

DISCUSSION

This thesis aimed to develop a novel end-to-end pipeline for converting continuous/sentence-based ASL videos to English language speech. The proposed method discussed in this thesis consists of 4 sub-modules, namely, Video-to-Gloss module, Gloss-to-Grapheme module, Grapheme-to-Phoneme module, and Phoneme-to-Spectrogram module. In each of these modules, Seq2Seq approach was used for developing the modules. All the modules were developed using public datasets such as WLASL dataset, ASLG-PC12 dataset, CMUDict dataset, and LJSpeech dataset. Since there are multiple modules, the models might suffer from information loss; hence the proposed approach mainly focuses on the first three modules. This approach would help people from ASL community to converse with everyone in a more accessible and hassle-free manner. The proposed approach works for the first three modules, which substantially improved the performance in many cases than the current state-of-the-art models. However, the proposed method did not improve the performance of the model in the phoneme-to-spectrogram module. Future work would be to improve the Transformer model in the phoneme-to-spectrogram module and create a sentence-based dataset for ASL Video-to-Gloss module. An approach that can be considered for the Phoneme-to-Spectrogram module would be to use the deep transformer model; however, we would need a more extensive infrastructure to implement.

REFERENCES

[1] Sarfaraz Masood, Harish Thuwal, and Adhyan Srivastava. *American Sign Language Character Recognition Using Convolution Neural Network*, pages 403–412. 10 2018.

[2] D. Aryanie and Y. Heryadi. American sign language-based finger-spelling recognition using k-nearest neighbors classifier. In *2015 3rd International Conference on Information and Communication Technology (ICoICT)*, pages 533–536, 2015.

[3] Wikipedia contributors. Speech synthesis — Wikipedia, the free encyclopedia, 2020. [Online; accessed 10-December-2020].

[4] Sercan O. Arik, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, Shubho Sengupta, and Mohammad Shoeybi. Deep voice: Real-time neural text-to-speech, 2017.

[5] A. M. Martinez, R. B. Wilbur, R. Shay, and A. C. Kak. Purdue rvl-slll asl database for automatic recognition of american sign language. In *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces*, pages 167–172, 2002.

[6] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan, Quan Yuan, and A. Thangali. The american sign language lexicon video dataset. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2008.

[7] Schembri A. The british sign language corpus project, Jun 2018.

[8] Hamid Vaezi Joze and Oscar Koller. Ms-asl: A large-scale data set and benchmark for understanding american sign language. In *The British Machine Vision Conference (BMVC)*, September 2019.

[9] Dongxu Li, Cristian Rodriguez Opazo, Xin Yu, and Hongdong Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison, 2020.

[10] Samuel Albanie, Gül Varol, Liliane Momeni, Triantafyllos Afouras, Joon Son Chung, Neil Fox, and Andrew Zisserman. Bsl-1k: Scaling up co-articulated sign language recognition using mouthing cues, 2020.

[11] Jens Forster, Christoph Schmidt, Thomas Hoyoux, Oscar Koller, Uwe Zelle, Justus Piater, and Hermann Ney. RWTH-PHOENIX-weather: A large vocabulary sign language recognition and translation corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3785–3789, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

[12] Oscar Koller, Jens Forster, and Hermann Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141:108 – 125, 2015. Pose & Gesture.

[13] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[14] Wikipedia contributors. Seq2seq — Wikipedia, the free encyclopedia, 2020. [Online; accessed 21-September-2020].

[15] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[16] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[17] A Vaswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, AN Gomez, L Kaiser, and I Polosukhin. Attention is all you need. arxiv 2017. *arXiv preprint arXiv:1706.03762*, 2017.

[18] A. Farhadi, D. Forsyth, and R. White. Transfer learning in sign language. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[19] H. Fillbrandt, S. Akyol, and K. . Kraiss. Extraction of 3d hand shape and posture from image sequences for sign language recognition. In *2003 IEEE International SOI Conference. Proceedings (Cat. No.03CH37443)*, pages 181–186, 2003.

[20] N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden. Subunets: End-to-end hand shape and continuous sign language recognition. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3075–3084, 2017.

[21] A. S. Nikam and A. G. Ambekar. Sign language recognition using image based hand gesture recognition techniques. In *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, pages 1–5, 2016.

[22] Patrick Buehler, Andrew Zisserman, and Mark Everingham. Learning sign language by watching tv (using weakly aligned subtitles). In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2961–2968. IEEE, 2009.

[23] T. D. Nguyen and S. Ranganath. Tracking facial features under occlusions and recognizing facial expressions in sign language. In *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–7, 2008.

[24] PVV Kishore, G Anantha Rao, E Kiran Kumar, M Teja Kiran Kumar, and D Anil Kumar. Selfie sign language recognition with convolutional neural networks. *International Journal of Intelligent Systems and Applications*, 10(10):63, 2018.

[25] Hyojoo Shin, Woo Je Kim, and Kyoung-ae Jang. Korean sign language recognition based on image and convolution neural network. In *Proceedings of the 2nd International Conference on Image and Graphics Processing*, ICIGP '19, page 52–55, New York, NY, USA, 2019. Association for Computing Machinery.

[26] Sang-Ki Ko, Chang Jo Kim, Hyedong Jung, and Choongsang Cho. Neural sign language translation based on human keypoint estimation, 2019.

[27] Sang-Ki Ko, Jae Gi Son, and Hyedong Jung. Sign language recognition with recurrent neural network using human keypoint detection. In *Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems*, RACS '18, page 326–328, New York, NY, USA, 2018. Association for Computing Machinery.

[28] Ulrich Von Agris, Jörg Zieren, Ulrich Canzler, Britta Bauer, and Karl-Friedrich Kraiss. Recent developments in visual sign language recognition. *Universal Access in the Information Society*, 6(4):323–362, 2008.

[29] Ali Farhadi and David Forsyth. Aligning asl for statistical translation using a discriminative word model. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1471–1476. IEEE, 2006.

[30] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7784–7793, 2018.

[31] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. Video-based sign language recognition without temporal segmentation, 2018.

[32] Yuancheng Ye, Yingli Tian, Matt Huenerfauth, and Jingya Liu. Recognizing american sign language gestures from within continuous videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2064–2073, 2018.

[33] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[34] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

[36] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[37] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. In *Proceedings of the British Machine Vision Conference*, pages 124.1–124.11. BMVA Press, 2009. doi:10.5244/C.23.124.

[38] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun 2014.

[39] Julieta Martinez, Michael J. Black, and Javier Romero. On human motion prediction using recurrent neural networks, 2017.

[40] R. Cui, H. Liu, and C. Zhang. Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1610–1618, 2017.

[41] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden. Neural sign language translation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7784–7793, 2018.

[42] 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)*, pages 4160–4169, 2019.

[43] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation. *arXiv e-prints*, page arXiv:2003.13830, March 2020.

[44] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Progressive transformers for end-to-end sign language production, 2020.

[45] Zhe Niu and Brian Mak. Stochastic fine-grained labeling of multi-state sign glosses for continuous sign language recognition. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 172–186, Cham, 2020. Springer International Publishing.

[46] Ka Leong Cheng, Zhaoyang Yang, Qifeng Chen, and Yu-Wing Tai. Fully convolutional networks for continuous sign language recognition. In *European Conference on Computer Vision*, pages 697–714. Springer, 2020.

[47] Nikolaos Arvanitis, Constantinos Constantinopoulos, and Dimitris Kosmopoulos. Translation of sign language glosses to text using sequence-to-sequence attention models. pages 296–302, 11 2019.

[48] Kayo Yin and Jesse Read. Better sign language translation with stmc-transformer, 2020.

[49] Maximilian Bisani and Hermann Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451, 2008.

[50] Kaisheng Yao and Geoffrey Zweig. Sequence-to-sequence neural net models for grapheme-to-phoneme conversion, 2015.

[51] Amr El-Desoky Mousa and Björn Schuller. Deep bidirectional long short-term memory recurrent neural networks for grapheme-to-phoneme conversion utilizing complex many-to-many alignments. In *Interspeech 2016*, pages 2836–2840, 2016.

[52] Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. Grapheme-to-phoneme conversion with convolutional neural networks. *Applied Sciences*, 9(6), 2019.

[53] Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. Transformer based grapheme-to-phoneme conversion. *arXiv preprint arXiv:2004.06338*, 2020.

[54] Sercan Arik, Gregory Diamos, Andrew Gibiansky, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou. Deep voice 2: Multi-speaker neural text-to-speech, 2017.

[55] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O. Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep voice 3: Scaling text-to-speech with convolutional sequence learning, 2018.

[56] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech: Fast, robust and controllable text to speech, 2019.

[57] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[58] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.

[59] Achraf Othman and Mohamed Jemni. English-asl gloss parallel corpus 2012: Aslg-pc12. In *5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon LREC*, 2012.

[60] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71. Association for Computational Linguistics, November 2018.

[61] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020.

[62] J. F. Kolen and S. C. Kremer. *Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies*, pages 237–243. 2001.

[63] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. *ArXiv*, abs/1211.5063, 2012.

[64] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[65] R Weide. The cmu pronunciation dictionary.

[66] Keith Ito. The lj speech dataset, 2017.

[67] Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. Bleu: a method for automatic evaluation of machine translation. pages 311–318, 2002.

[68] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[69] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, February 1966.