Spring 2023

Video Sign Language Recognition using Pose Extraction and Deep Learning Models

Shayla Luong
San Jose State University

Video Sign Language Recognition using Pose Extraction and Deep Learning Models

A Project

Presented to

The Faculty of the Department of Computer Science

Department of Computer Science

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By Shayla Luong

May 2023

The Designated Committee Approves the Master's Project Titled

Video Sign Language Recognition using Pose Extraction and Deep Learning Models

By

Shayla Luong

Approved for the Department of Computer Science

San José State University

April 2023

Dr. Nada Attar	Department of Computer Science
Dr. Ching-Seh Wu	Department of Computer Science
Dr. Mashhour Solh	Department of Computer Science

ABSTRACT

Sign language recognition (SLR) has long been a studied subject and research field within the Computer Vision domain. Appearance-based and pose-based approaches are two ways to tackle SLR tasks. Various models from traditional to current state-of-the-art including HOG-based features, Convolutional Neural Network, Recurrent Neural Network, Transformer, and Graph Convolutional Network have been utilized to tackle the area of SLR. While classifying alphabet letters in sign language has shown high accuracy rates, recognizing words presents its set of difficulties including the large vocabulary size, the subtleties in body motions and hand orientations, and regional dialects and variations. The emergence of deep learning has created opportunities for improved word-level sign recognition, but challenges such as overfitting and limited training data remain. Techniques such as data augmentation, feature engineering, hyperparameter tuning, optimization, and ensemble methods have been used to overcome these challenges and improve the accuracy and generalization ability of ASL classification models. We explore various methods to improve the accuracy and performance in this project. From the approach, we were able to first reproduce a baseline accuracy of 43.02% on the WLASL dataset and further achieve an improvement in accuracy at 55.96%. We also extended the work to a different dataset to gain a comprehensive understanding of our work.

Keywords: classification task, data augmentation, deep learning models, ensemble techniques, sign language recognition, skeletal keypoints, pose extraction

Table of Contents

List of Figures

List of Tables

I. Problem Statement

Sign language has long been developing into a complex and sophisticated method of visual communication to meet the needs of people with impaired auditory sensories all around the world. Its unique aspect as a visual language makes use of a wide range of hand gestures, facial expressions, and body movements to convey meaning and information in a highly expressive and nuanced way [1]. One example of a widely adopted sign language is the American Sign Language (ASL), which has a vast vocabulary consisting of thousands of different signs. Sign languages can be analyzed at various levels of complexity, including alphabet-level, word-level, and sentence-level. At the alphabet-level, the vocabulary in sign language only consists of 26 letters, which is a small amount of variability. Various past studies have demonstrated significantly high performance in accurately identifying all the letters in the American Sign Language alphabet, at accuracies reaching as high as 99% accuracy [2].

Unfortunately, the complexity of the language system grows significantly when moving beyond the alphabet level. Word-level Sign Language Recognition (SLR) is a much studied research area within the Computer Vision field. Understanding word-level sign recognition in ASL requires overcoming various challenges, including the subtlety of body motions and hand orientations, the large vocabulary size, and the different word interpretations of the same sign depending on the context [3]. In addition, sign language can be further complicated by regional dialects or variations, which may make it harder to detect slight differences or biases in signing. These challenges present unique difficulties for not just real-life but also Machine Learning perspectives.

The advancement in areas of deep learning has opened up new opportunities for major improvements in classifications of word-level sign recognition using various state of the art architectures. However, despite these technological advancements, there are still many challenges that researchers in the field face when it comes to understanding and modeling sign languages. The challenges include the variability of the signs and the limited amount of training data, modeling and classifying ASL signs using deep learning is often prone to overfitting of the training data. To overcome these challenges, researchers have developed and applied many state-of-the-art Machine Learning techniques such as contrastive learning, data augmentation, and self-supervised learning. These techniques are used to improve the accuracy and generalization ability of the models, enabling them to better recognize ASL signs in a variety of contexts and situations.

II. Related Works

There are two main approaches to hand sign recognition tasks: appearance-based and pose-based. From traditional to state-of-the-art techniques, researchers have explored various methods to address the challenge of sign language recognition using both these types of approaches. Additionally, either approaches can be tackled using either a model trained from scratch or pretrained models that have been specialized for the sign language recognition task to improve performance. By leveraging pre-trained models, researchers could alleviate the shortage of data available for sign language.

A. Appearance-based Recognition

In the appearance-based approach, early works employ hand-crafted Histogram of Oriented Gradients (HOG)-based or Scale-Invariant Feature Transform (SIFT)-based features to represent the spatial arrangement of static hand poses [4]. Then, classical sequence learning models such as Hidden Markov Model (HMM) were used for temporal modeling. Finally, Support Vector Machine (SVM) or kNN is used to classify the hand gesture video into a specific gloss. While this approach has shown promise in early studies with static image-based classification tasks, the models are limited in their ability to handle larger scales of data. This is a challenge as more complex sign language recognition tasks require the processing of much larger and more diverse datasets.

One of the most well-known models when it comes to training visual data is the Convolutional Neural Network (CNN) [5]. CNN has become increasingly popular for its ability to perform well in various computer vision applications like medical image analysis, facial recognition, and object detection since its inception. Taking a similar approach to action recognition and 2D human pose, some recent works use Convolutional Neural Networks (CNN) to extract holistic features from video frames and then feed the extracted features into a Recurrent Neural Network (RNN) [6, 7] for further classification. An RNN is necessary for processing sequential or time series data because it retains and models the past with future information, as opposed to traditional feedforward networks that models independent data points. CNNs and RNNs have shown significant improvement over traditional models at handling larger scale and complex data. The approach from [8] employs a pre-trained VGG16 model trained on ImageNet to extract spatial features from input video frames. The next step is feeding the

extracted features into a stacked GRU (Gated Recurrent Units) for modeling dependency between inputs and performing final classifications.

Another approach is to use 3D CNNs that also capture spatial and temporal features together in a hierarchical fashion. A popular tactic such as one in [9] inflates 2D filters of a pretrained Inception network to obtain 3D filters that are well-initialized. These 3D filters are then fine-tuned on the Kinetics dataset to better capture both the temporal and spatial information in a video. For the feature extraction process, in addition to manual features such as hand shape, position, orientation of palm or fingers, non-manual features such as eye gaze, head-nods, shoulder orientation, and various facial expressions must also be captured [3]. [10] employs the I3D network architecture and performs fine-tuning on ImageNet and Kinetics-400 following the approach in [7]. Another network that has shown great performance is ResNet2+1D due to it being able to decouple spatial and temporal convolution and then perform the operation one after the other [11]. One recent work [12] proposes a video Swin Transformer consisting of a backbone architecture. Differently from typical a 3D CNN network where a 3D convolution is barely a direct extension of 2D convolution to incorporate the next spatial and temporal modeling, the Swin Transformer network takes advantage of spatiotemporal locality of videos in which pixels close in spatiotemporal distance are more likely to be correlated. The model could be pre trained on large-scale image and video datasets.

B. Pose-based Recognition

For the pose-based approach, some recent works use an off-the-shelf pose extractor to obtain hand priors represented as visual tokens. The tokens typically contain embedded pose information including hand state, temporal, and chirality. After this pose information is extracted

from extractors such as OpenHands [13], the next part of the network can include attention-based Transformer network or BERT LSTM network to model the sequence-based data [14]. Pose extractors help the training process on labeled data more efficiently when the poses are used as modality. In addition to using hand poses, some recent works have introduced skeleton-based methods into the hand sign recognition task. In action recognition and pose estimation areas, the skeleton-based approach has been shown effective due to its nature of being able to isolate dynamic foreground movements from a busy background. In past works, the approach relies on obtaining ground-truth skeleton annotations captured by physical motion sensor systems. This greatly limits the amount of available training data, especially for hand skeleton poses, as the system is difficult to set up and not as available. S. Jiang et al [15] propose a novel network of skeleton graphs that could model the spatial and temporal dynamic movements of the hands as well as whole-body key points while still maintaining their relative spatial features. Some works specifically extract body key points, by localizing the joints of human bodies, and concatenate them into a feature vector. This feature vector is then fed into an RNN or other sequential-data modeling systems for recognizing video sequences of signs. Although this approach captures the temporal movements and trajectories of the poses, it is not fully able to capture spatial information between the body key points. One related work attempts to overcome this by employing a Graph Convolutional Network (GCN) to capture both temporal and spatial dependencies simultaneously [16]. The Temporal GCN (TGCN) represents the human body as a fully connected graph network of vertices and edges.

However, there is a limited amount of labeled training data available that might lead to lower performance or cause the models to overfit. In order to tackle the challenge of having limited labeled data, some recent works employ self-supervised pre-training on unlabeled data

using available datasets of different languages [17]. These datasets include NMFs-CSL, SLR500, MSASL, and WLASL. A similar challenge in Natural Language Processing (NLP) was mitigated through self-supervised pre-training strategies on large text corpus. The Bidirectional Encoder Representations from Transformers (BERT) model has seen much success in this area due to its simplicity and high performance. Following this idea, Jiang et al. [18] propose the SignBERT model which is trained through a self-supervised process on large amounts of hand pose data obtained from pose extractors. After the pre-training process on unlabeled data, the model could then be trained on labeled dataset and achieve better results.

Additionally, recent advances in deep generative models enable labels to be omitted during the training phase while still producing accurate predictions if the dynamics and content of the data is captured to a good extent. Unsupervised techniques are preferable over supervised ones since data annotation for video analysis is more costly compared to static images. Various recent works employ deep auto-encoder frameworks to capture and synthesize patterns during the pre training process. Some works include Dynencoder, LSTM Autoencoder, and GAN models [19]. These models can learn to represent the spatiotemporal information of a video in a compact way during the training process. Then, the reconstruction error in autoencoders can be used as a tool for classification.

Similarly, the authors of [20] propose an approach that targets skeleton-based action representation learning in an unsupervised manner. The approach focuses on separating the input representation into multiple levels of features. The levels include instance level, domain level, clip level, and part level. The proposed technique named Hierarchical Contrast (HiCo) framework uses sequence-to-sequence encoders and downsampling techniques to obtain features from both temporal and spatial domains. Following the same approach as the skeleton-based

feature extractions and training mentioned above, similar techniques could be adapted to train specific tasks like ASL video recognition. This could be proven useful given the limited labeled data available and the available pre-trained models.

III. DATASET

A. Word-Level American Sign Language (WLASL)

The WLASL dataset consists of a collection of video recordings of people performing ASL signs that correspond to individual English words. The dataset was originally proposed by Dongxu et al [21]. The recordings are captured in various lighting, contexts, and angles to provide more comprehensible views of the hands, elbows, and body motions in each gesture. The videos are hand-annotated with consistent metadata that provides information including the gloss, bounding box of the signer, and start and end frames of the word being signed. There is also a designated ID to group and distinguish different signers. All the metadata fields are listed here:

- gloss
- bbox
- fps
- frame_start
- frame_end
- validation_id
- instance_id
- signer_id
- source
- split
- url
- video_id

1. Constituting Subsets

The authors of [21] also split the dataset into four subsets, namely WLASL100, WLASL300, WLASL1000, and WLASL2000. These subsets are generated by selecting the

top-K (k=100, 300, 1000 and 2000) glosses as specified in the criteria defined in the paper. For the scope of this project, we'll be working with the WLASL100 subset to demonstrate the results of our approach and lay the groundwork for future work with other datasets.

2. Obtaining Videos

Each data point in the WLASL dataset contains a URL link where the video could be downloaded from, the majority of which are from Youtube while various others are hosted on other video streaming platforms or Cloud storage services. The authors provided source code which includes a feature to load the dataset and enables users to download videos directly from the provided links. This simple feature provides convenience and simplifies the process of accessing the dataset.

B. Missing Data

Due to many videos being taken down or only providing private access, we were only able to obtain a fraction of the WLASL100 subset. From the baseline, the number of videos originally obtained were 2,038 split into 100 glosses. However, we were only able to obtain 1,323 videos due to broken links. Within this fraction, the split for training is 1036, validation is 166, and for testing is 121 videos.

C. Data Preprocessing

After the videos are downloaded from URL links, the raw videos undergo a series of data pre-processing steps including conversion to mp4 format, video trimming, cropping, resizing, and label encoding. The metadata that accompanies the dataset contains precise details on when to trim the videos, so that only the sections showing the sign language gestures are included.

Cropping and resizing the videos ensures that the data formats are uniform and consistent throughout. Finally, the dataset is partitioned into training, validation, and test sets based on a predetermined split and annotated with numerical labels.

D. Preliminary Experiments with Video Data

In order to gain a comprehensive understanding of the scope and opportunities the dataset presents, we conducted preliminary groundwork. This involved training and evaluating a series of baseline models and analyzing the resulting data. By undertaking this preliminary work, we gained insights into the strengths and limitations of the dataset and used this information to guide our research project and future experimentation work.

Following the work detailed in [21], we begin creating a baseline for the ASL recognition task. Since the training dataset is of video format where each hand sign is presented in a 3 to 4 seconds long clip, the Machine Learning model must be able to capture this sequential dependency between multiple frames to achieve good performance. Thus dividing this task into a two-part network, one dealing primarily with feature extractions of per-frame content while the other takes care of temporal relationships between these data points.

In a preliminary implementation, we attempted to build a single-layer shallow Convolutional Neural Network for the classification task. The approach has previously shown great results on classifying the ASL alphabet, producing near perfect accuracy. However, the alphabet consisting of simple characters is much simpler to classify compared to word-level classification due to the simplicity and distinctiveness of hand motions required to produce the signal. Additionally, the ASL alphabet contains only 26 characters as opposed to the large ASL vocabulary consisting of thousands of words. Attempting the same approach, we implemented a

shallow network consisting of 2 Convolutional layers with Max Pooling and Batch Normalization and a final dense layer for classification. In order to obtain the input images for the CNN, we extracted 5 separate frames from each training video from the 100-gloss subset, namely the WLASL100. The frames are each pre-processed to a fixed 224x224 size and assigned the label corresponding to their gloss. The model is trained for 50 epochs using categorical cross-entropy loss and Adam optimizer. This model produces subpar results with accuracy score of 0.03 when evaluated against the test set.

Next, we added the temporal component to a new network and experimented with a similar approach to action recognition tasks. In the feature extraction step, we used an InceptionV3 model with weights pre-trained on the ImageNet dataset. By excluding the fully connected top layer, the model will output high-level features from images instead of classifications. The input consists of 748 videos, each with 20 frames, after extraction, the output consists of 2,048 features per frame. After obtaining extracted features, the next component of the network is a Recurrent Neural Network that captures the temporal aspect of the data. In this implementation, stacked GRU layers are used to model the temporal relationship between the frames of each video and a dense final layer is used for classification. The architecture of this model is illustrated in the figure below.

```
Model: "model_5"
_________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #     Connected to
=================================================================================================
 input_13 (InputLayer)       [(None, 25, 2048)]        0           []

 input_14 (InputLayer)       [(None, 25)]              0           []

 gru_10 (GRU)                (None, 25, 16)            99168       ['input_13[0][0]',
                                                                    'input_14[0][0]']

 gru_11 (GRU)                (None, 8)                 624         ['gru_10[0][0]']

 dropout_5 (Dropout)         (None, 8)                 0           ['gru_11[0][0]']

 dense_10 (Dense)            (None, 8)                 72          ['dropout_5[0][0]']

 dense_11 (Dense)            (None, 100)               900         ['dense_10[0][0]']

=================================================================================================
Total params: 100,764
Trainable params: 100,764
Non-trainable params: 0
_________________________________________________________________________________________________
```

Figure 1: Inceptionv3 + GRU Model Summary

During training, both the training and validation accuracy fluctuate back and forth but there is an upward trend for the training set, meaning that the model is able to learn from training data, however there is a barrier to learn and generalize to new data. The result inference to test data only achieves 0.10 accuracy.

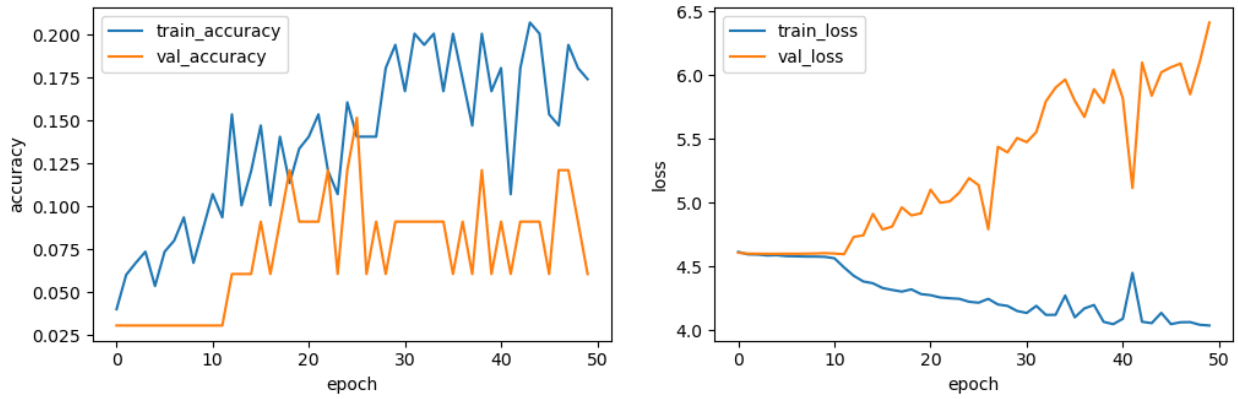

Figure 2: Inceptionv3 + GRU Train vs Validation Accuracy and Loss Graphs

The low accuracy achieved in the models could be attributed to the lack of consistent and good data. It appears that the model can learn from the training data, but it fails to generalize effectively when applied to validation and testing data. This suggests that the quality of data used for training the model plays a crucial role in its ability to perform well on unseen data. After this preliminary work, we decided to discontinue the approach of using appearance-based models due to the fact that it requires a significant amount of training data, which we do not possess.

From surveying recent works, the next approach would be to extend works such as ones presented in [11, 13, 16]. The main components are detailed as follows.

- Feature extraction using off-the-shelf pose extractor. This extraction process focuses on two main aspects, the hand or joint position poses as well as the composition of the signers' facial expressions and body gestures.
- Model the temporal relationship between frames of the same sequence of motion. This can be done using the Transformer-based model.

Our baseline and final project implementation is based on this new approach.

IV. BASELINE RECONSTRUCTION

The baseline work is based on the work done by Bohacek and Hruz presented in their paper "Sign Pose-Based Transformer for Word-Level Sign Language Recognition" (SPOTER) [22]. The backbone of SPOTER is a Transformer model which has been on the rise with demonstrated high performance in various domains such as natural language processing (NLP) applications such as language comprehension, machine translation, and text generation. As of recent, the Transformer model has received significant attention and been researched extensively

in the field of Computer Vision as well. Transformer-based models typically consist of an encoder that extracts features from an input sequence and a decoder that translates the features back into a new output sequence. The "attention" mechanism is the fundamental component of a Transformer model. It is responsible for computing weights for each input element based on its significance to the current processing stage. These weights are utilized to determine a weighted sum of the input elements, which in turn serves as the input to the subsequent layer of the model. Transformers differ from Recurrent Neural Network models in that they do not process sequential data in order to model temporal data. Instead, their self-attention mechanism enables them to concentrate on the most significant components of the input at each processing stage. This helps to capture the dependencies between frames of the skeletal input data.

A. Input Data

The input data makes up of extracted pose skeletal information using an off-the-shelf pose extractor, namely Vision API. The pose data consists of 104 features that capture information such as head and body main key points as well as hand joints throughout the videos. The Vision API produces confidences output scores for each recognized body landmarks, the points with confidence less than specified threshold are zeroed out. The coordinates are within the range of (0,0) for the bottom left of the image and (1,1) for the top right. The data has also been pre-processed using techniques such as normalization and transformation to standardize the dataset. The WLASL100 data subset contains 100 glosses and a total of 2,038 videos. This information is captured for each frame of the video and then is fed into the training network.

Table 1: Extracted Features for Baseline Input Data

Joints	Hands	Head
right elbow left elbow right wrist left wrist	index tip index dip index pip index mcp middle tip middle dip middle pip middle mcp ring tip ring dip ring pip ring mcp little tip little dip little pip little mcp thumb tip thumb dip thumb pip thumb mcp	nose neck right eye left eye right shoulder left shoulder

B. Baseline Architecture

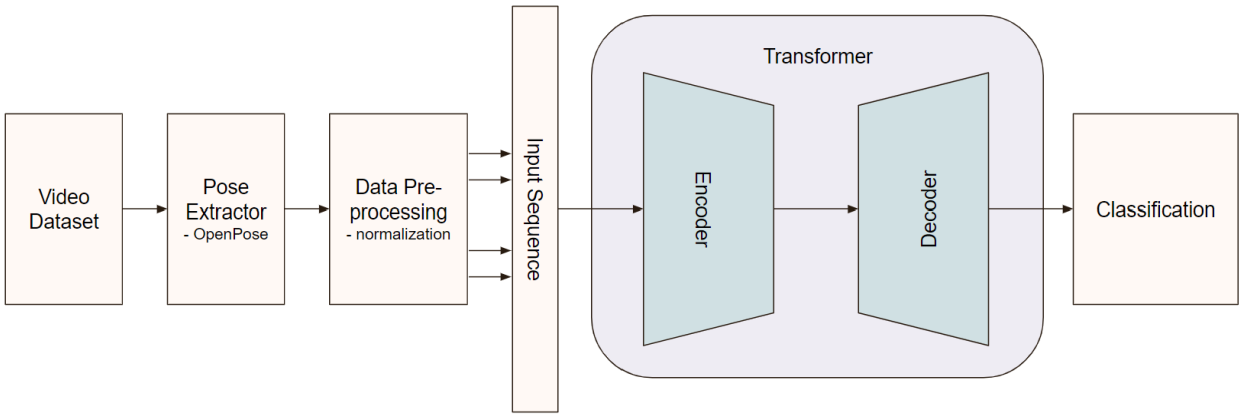

Figure 3: Baseline Architecture

The architecture consists of various steps including video data pre-processing, pose extraction, pose data pre-processing, Transformer model training, and final classification and evaluation as shown in the diagram above.

C. Model Evaluation Metric

The performance of machine learning models concerning classification tasks are typically evaluated using the Top-1 accuracy metric. It measures the percentage of times that the model correctly predicts the most likely classification for a given input, that is the model's prediction for one out of all possible classes matches the true label for the input. The metric is used commonly due to it being simple and fast to evaluate. It also provides a clear measure of how well the model is able to distinguish between different classes. The drawback of this metric is that it's not able to account for the ranking of the correct class compared to other classes. This topic is explored briefly at the end of this report. All models throughout this report are evaluated using Top-1 accuracy. Since the dataset for our experiments contains 100 glosses, each prediction will be one of 100 possible labels, thus achieving high performance using the top-1 accuracy metric can be challenging.

D. Reproducing Baseline

We attempted to replicate the training process outlined in the paper by using the hyperparameters provided without making any changes to them. The values for these hyperparameters are presented in the table below. One variation we made was training the replicated model for 100 epochs, rather than the 250 epochs stated in the paper. Due to the time constraints and the observation that the model's training accuracy and validation accuracy have

stopped improving in any significant amount after 50 epochs of training, we opted to use 100 epochs for the subsequent training and hyperparameter selection.

Table 2: Training Parameters for Baseline Reproduction

Training Parameters	
Learning Rate	0.001
Dropout Rate	0
Epochs	100
Loss	Cross Entropy
Optimizer	SGD

Due to having significantly less training data, the reproduced result could only reach 43.02% in accuracy. This is a significant decrease from the original result which achieved 63.18% in accuracy. As the missing training data cannot be accessed, we will consider the reproduced accuracy as our baseline for analysis going forward.

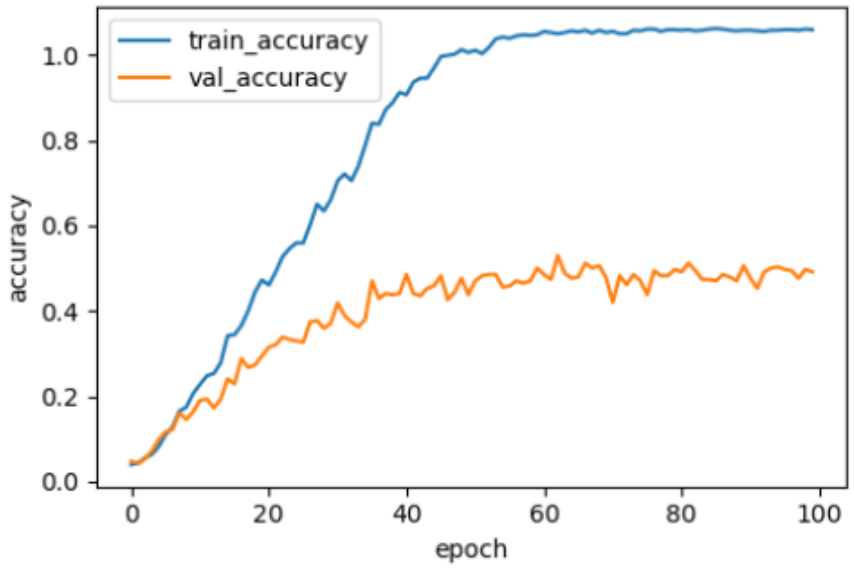

Figure 4: Train vs Validation Graph for Baseline Reproduction

Compared to appearance-based models, pose-based models tend to have lower dimensional feature extractions, which results in their training being more efficient. However, this approach often sacrifices performance accuracy in exchange for efficiency. In this experiment, the model has shown improvement on both aspects when compared to the CNN and GRU models presented in the preliminary works section. The model trained faster and performed better in terms of accuracy during test data inference. This improvement can be attributed to the fact that the training data is more specialized, allowing the model to train on more representative features, which was likely not feasible for the preliminary models due to limited training.

V. APPROACH AND EXPERIMENTS

In order to potentially enhance the accuracy of the model, we will be implementing our architecture and performing experiments with focus on these areas: data augmentation, hyperparameter tuning, pose data extraction, and ensemble learning.

A. Data Augmentation and Hyperparameter Tuning

One significant limitation for this project is the low amount of data available, which poses a challenge for achieving better model performance. Therefore, incorporating data augmentation techniques is a considerable approach to enhance the model's ability to capture diverse representations in the data. Moreover, considering the inherent variability in sign language, incorporating data from a variety of perspectives can further enhance the model's ability to generalize to new examples.

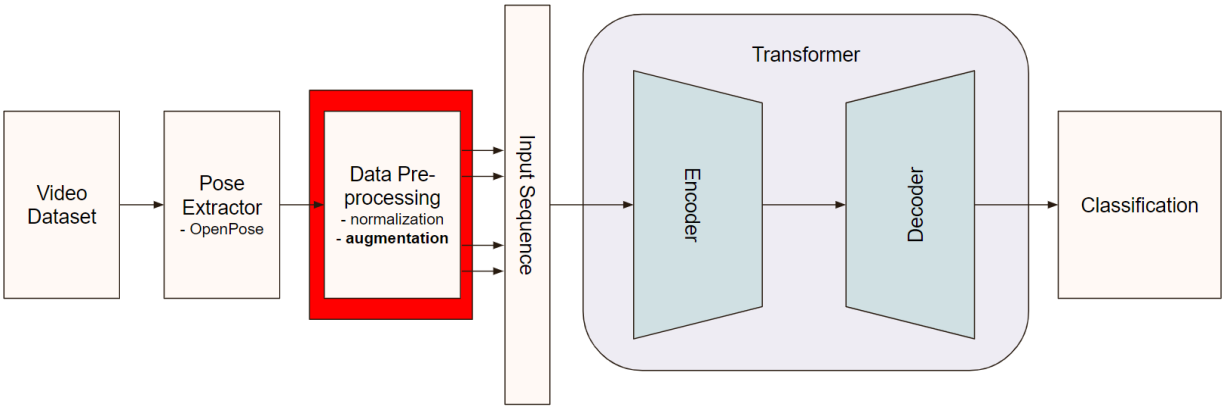

Figure 5: Architecture with Focus on Data Augmentation

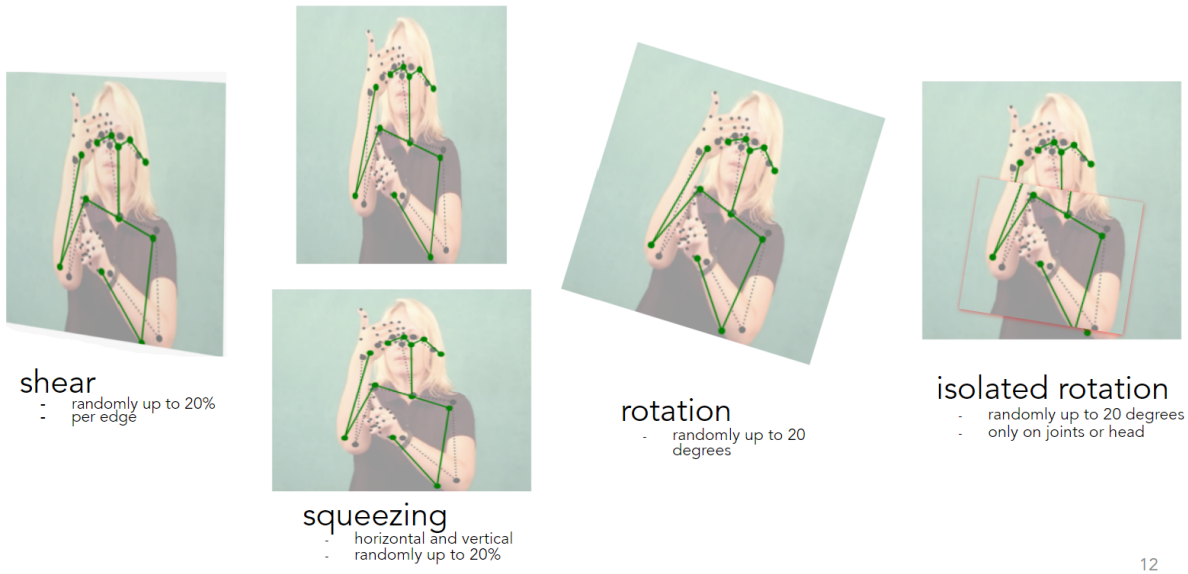

Figure 6: Data Augmentation Visual Examples

Some data augmentations included are random cropping, squeezing, rotation, and shearing. These are applied on the extracted 2D skeletal key points as demonstrated in the figure above. Since the hand and elbow joints could move independently of the head, we additionally performed isolated rotations by rotating only the hands and elbow joints while keeping the head positions stationary.

After incorporating data augmentation into our model pipeline, we saw improvements in the prediction results on the test data. Additionally, we utilized the validation data to tune our hyperparameters by tweaking parameters including learning rate, momentum, and weight decay as a regularization technique to reduce overfitting. The accuracies obtained as a result of these optimizations are reported in the table below.

Table 3: Result Comparison for Data Augmentation

Model	Test Accuracy
Baseline	43.02%
with Data Augmentation and Hyperparameter Tuning	46.80%

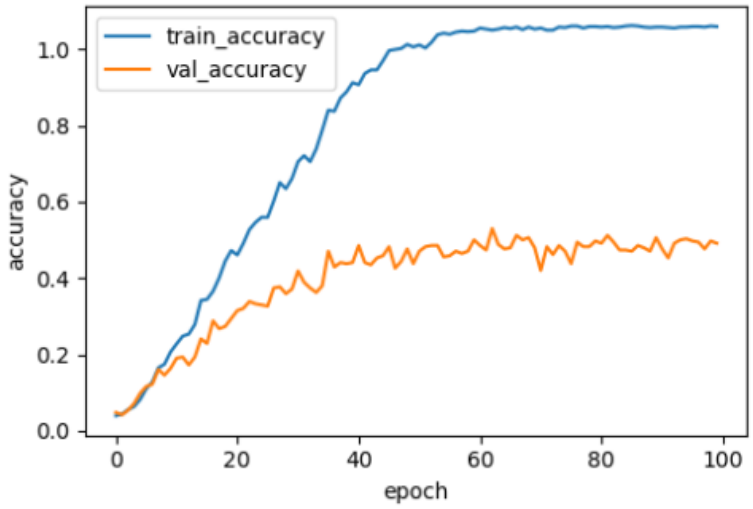

Figure 7: Train vs Validation Graph for Model with Data Augmentation

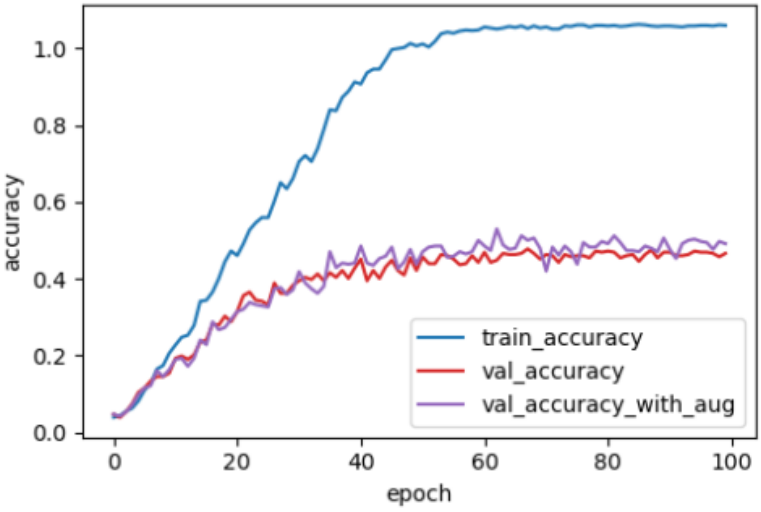

Figure 8: Comparison of Validation Graphs for Model with Data Augmentation

The graphs for training and validation accuracy above indicate that we are able to obtain higher accuracy on the validation set when applying the data augmentation optimization techniques.

B. Pose Extractors

The baseline paper utilized only the OpenPose extractor to obtain skeletal data from the video dataset. However, other extractors such as MediaPipe and SSTCN, which has been developed more specifically for Sign Language, have gained popularity due to their effectiveness in extracting accurate body keypoints.

The OpenPose was developed based on the CNN architecture to first detect where body parts and joints are in an image or video [23]. The detected keypoints include head, shoulders, elbows, wrists, hips, knees, and ankles. Then, an algorithm is applied to refine and improve the accuracy of the detection. It's able to highly accurately detect multiple people and body parts in complex scenes where there might be obstacles such as occlusions or varying lighting conditions.

MediaPipe Pose Estimation has a HRNet backbone. It uses a multi-stage CNN, similarly to OpenPose to detect and localize different body joints [24]. These joints are then connected to form body pose skeletons. The difference between OpenPose and MediaPipe is that the former uses a bottom-up approach while the latter uses a top-down approach to estimate the human pose. MediaPipe takes the entire body in as its input and estimates all the keypoints at the same time instead of estimating keypoints for each body part and then combining them.

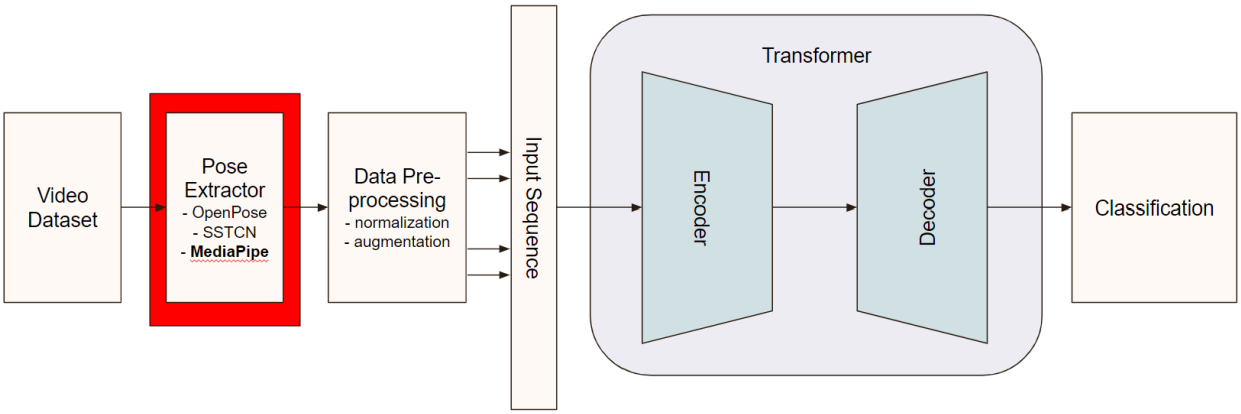

Figure 9: Architecture with Focus on Pose Extraction Step

Below is the table report of accuracies achieved by varying the pose extraction models. We see that by extracting more accurate pose positions, we were able to significantly improve the accuracy when training our models.

Table 4: Comparison of Different Pose Extraction Tools

Model	OpenPose	MediaPipe	SSTCN
Full body	46.80%	55.31%	42.53%
No head keypoints	45.45%	48.44%	41.20%

C. Ensemble Model

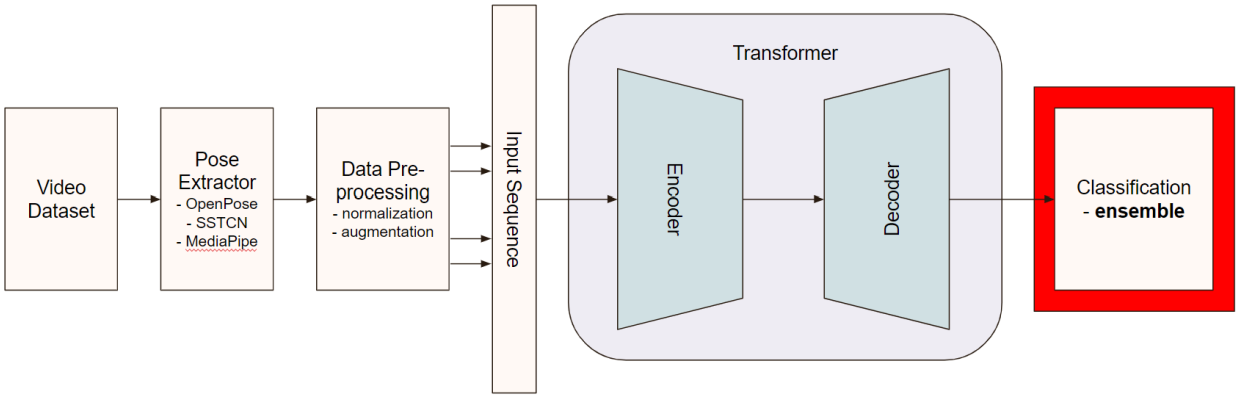

Figure 10: Architecture with Focus on Ensemble Technique

Ensemble models involve combining the predictions of multiple individual models to improve the overall performance, in both aspects of accuracy and reliability. For our architecture, the ensemble technique is applied at the last step as demonstrated in the figure above. There are several types of ensemble techniques including taking the majority vote or by averaging the prediction values. Each model within the ensemble can be trained with varying hyperparameters or data augmentation techniques. We employed a simple algorithm of averaging the predictions of our models using different pose extractors and data augmentation techniques. We were able to improve the accuracy of our overall model in multiple instances. The results are reported in the table below.

Table 5: Comparison of Ensemble Model Accuracies

Model	Test Accuracy
Baseline	43.02%
Ensemble of all 3 pose extractors	52.20%
Ensemble of MediaPipe and OpenPose predictions	55.96%

The 100 glosses confusion matrix for the dataset is presented below. There's indication that the model seems to have been able to predict well with no particular pattern to mispredictions. As shown in Figure 5, some glosses had a prediction accuracy of 100% while various others of 0%, suggesting that certain glosses were more challenging to predict than others. This finding provides some initial insights, and further analysis may reveal ways to better prepare the dataset. Therefore, we consider this as a starting point for future work to enhance the performance of the model.

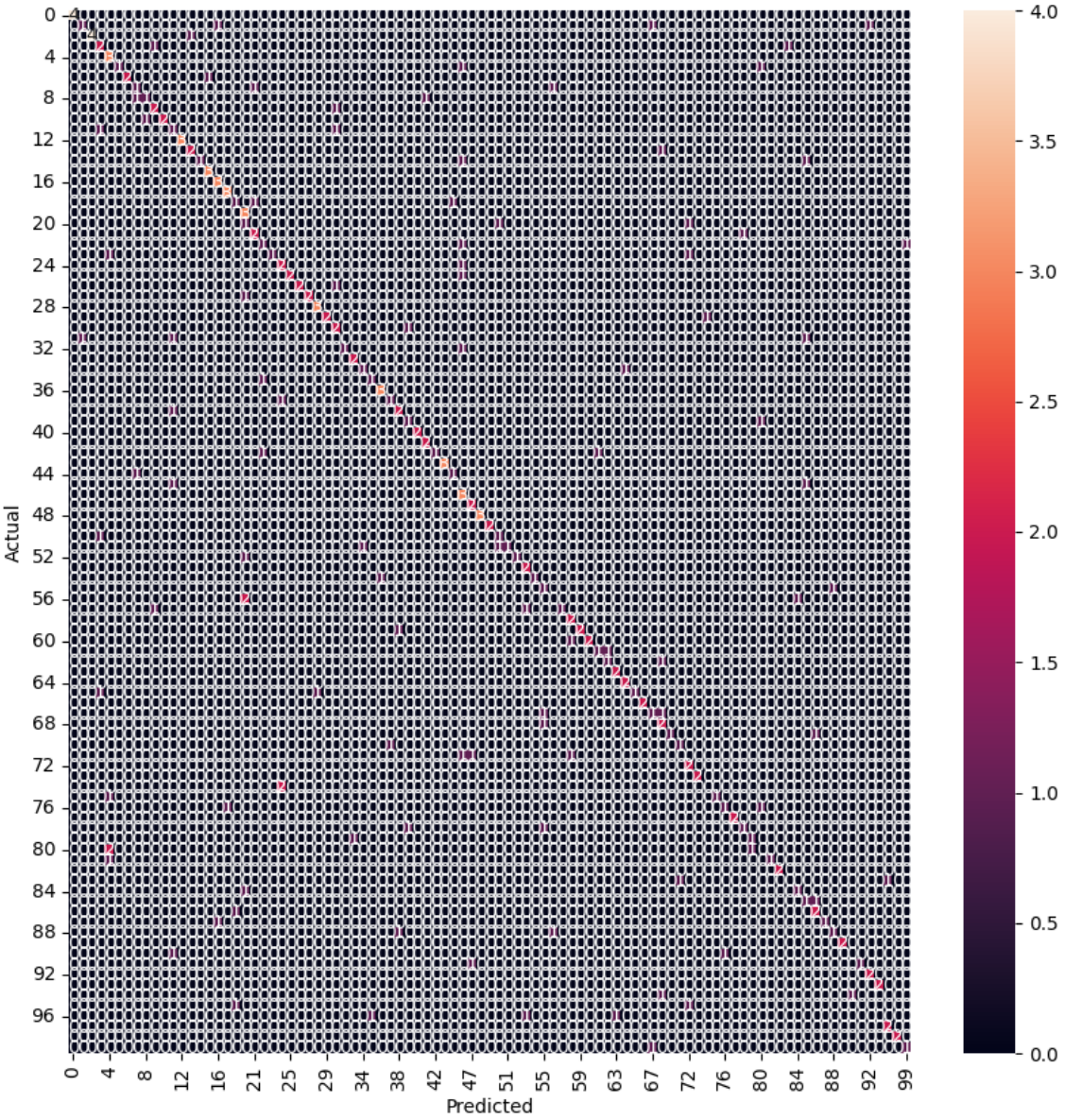

Figure 11: Confusion Matrix for Transformer Model Predictions of 100 Glosses

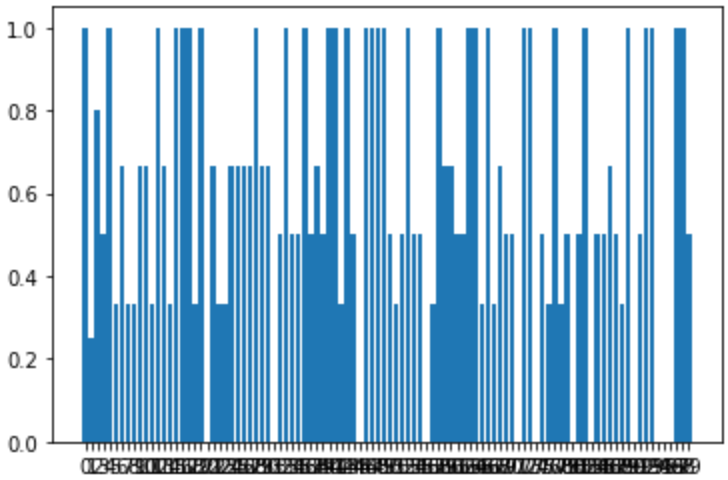

Figure 12: Predictability Score for Each Gloss

D. Additional Experiments on Other Datasets

Since the primary WLASL100 dataset used throughout this research project contains a large amount of missing data, we consider another dataset, namely AUTSL [25], to experiment our optimization approach with. AUTSL is a large-scale Turkish sign language dataset containing 38,336 videos from 43 different signers. The number of glosses available is 226. We followed the same split as the original paper to obtain 27,676 (72%) videos for the training set, 4,4884 (13%) for the validation set, and 5,776 (15%) for the test set. The number of training samples available in this dataset is significantly more than the 1,300 videos we previously acquired from the WLASL100. We conducted similar experiments as did with the WLASL100 dataset and reported the results in the table below.

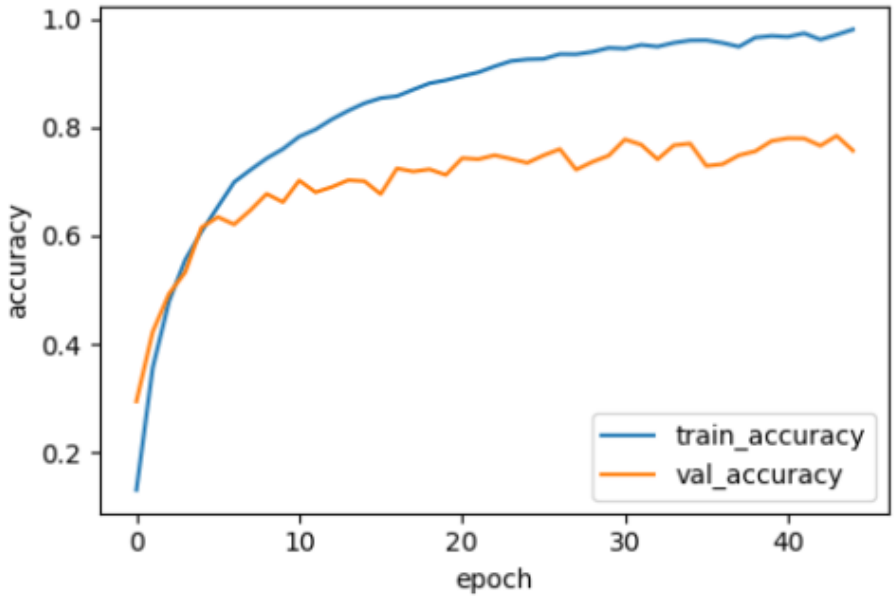

Figure 13: Train vs Validation Graph for Model using AUTSL Dataset

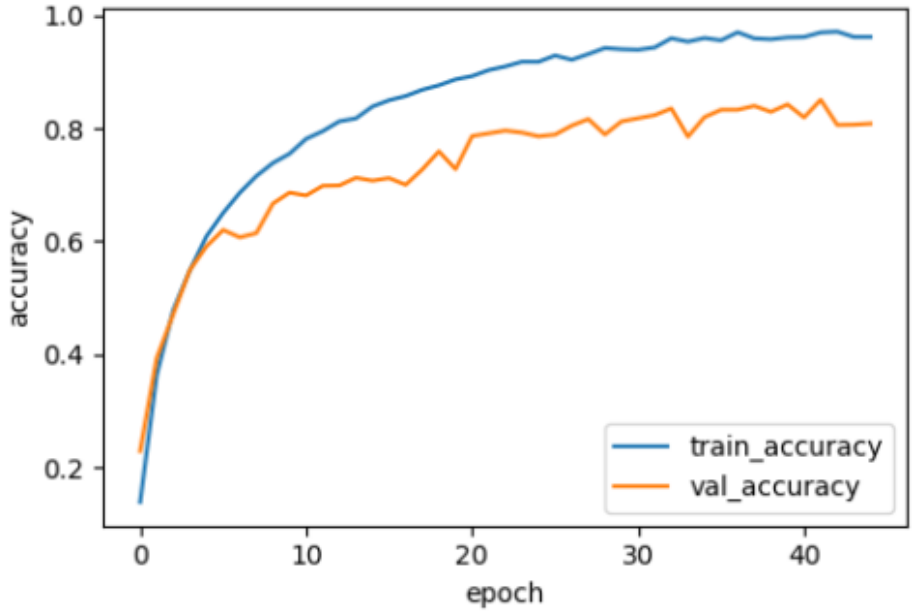

Figure 14: Train vs Validation Graph for Model using AUTSL Dataset with Data Augmentation

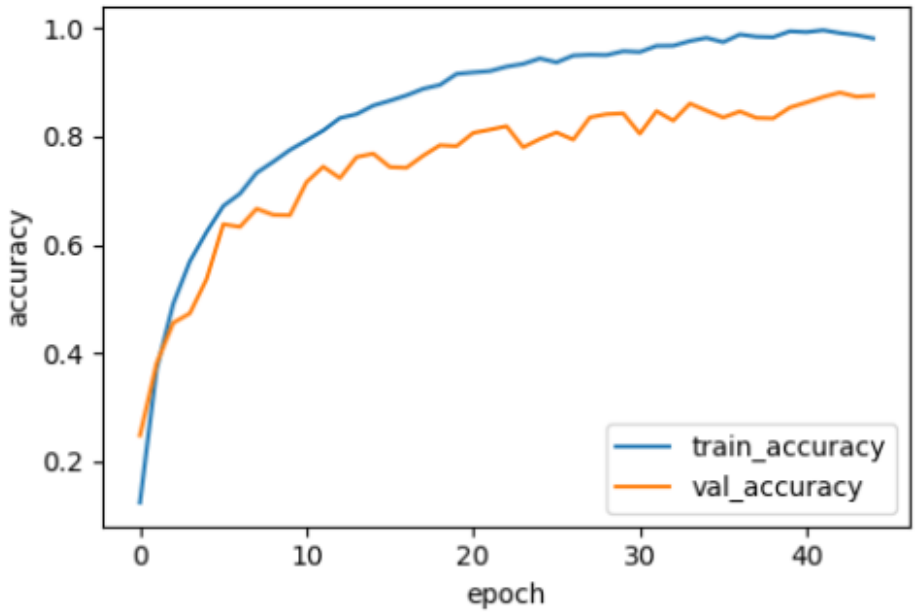

Figure 14: Train vs Validation Graph for Model using AUTSL Dataset with MediaPipe

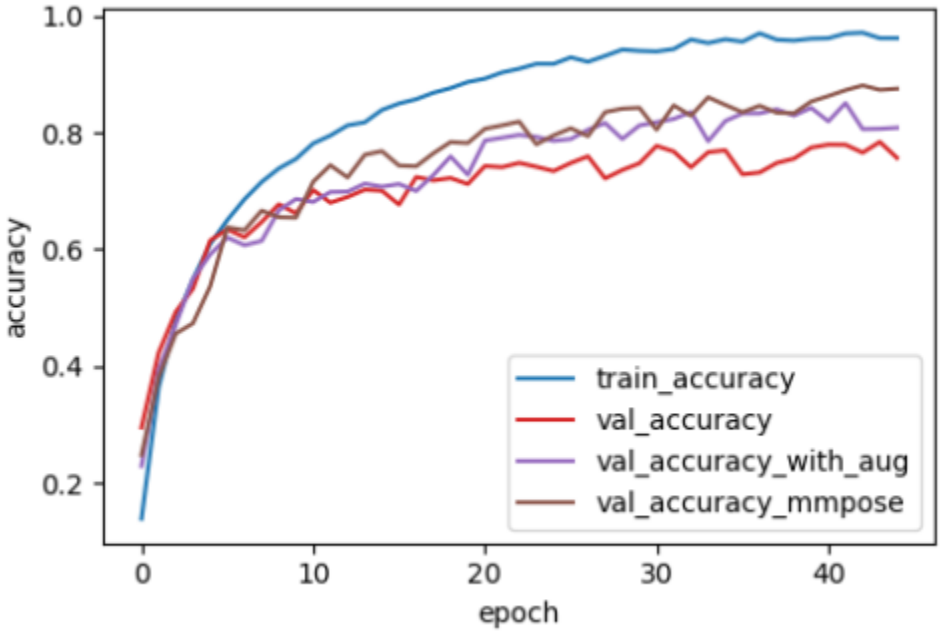

Figure 15: Comparison of Validation Graphs for Model using AUTSL Dataset

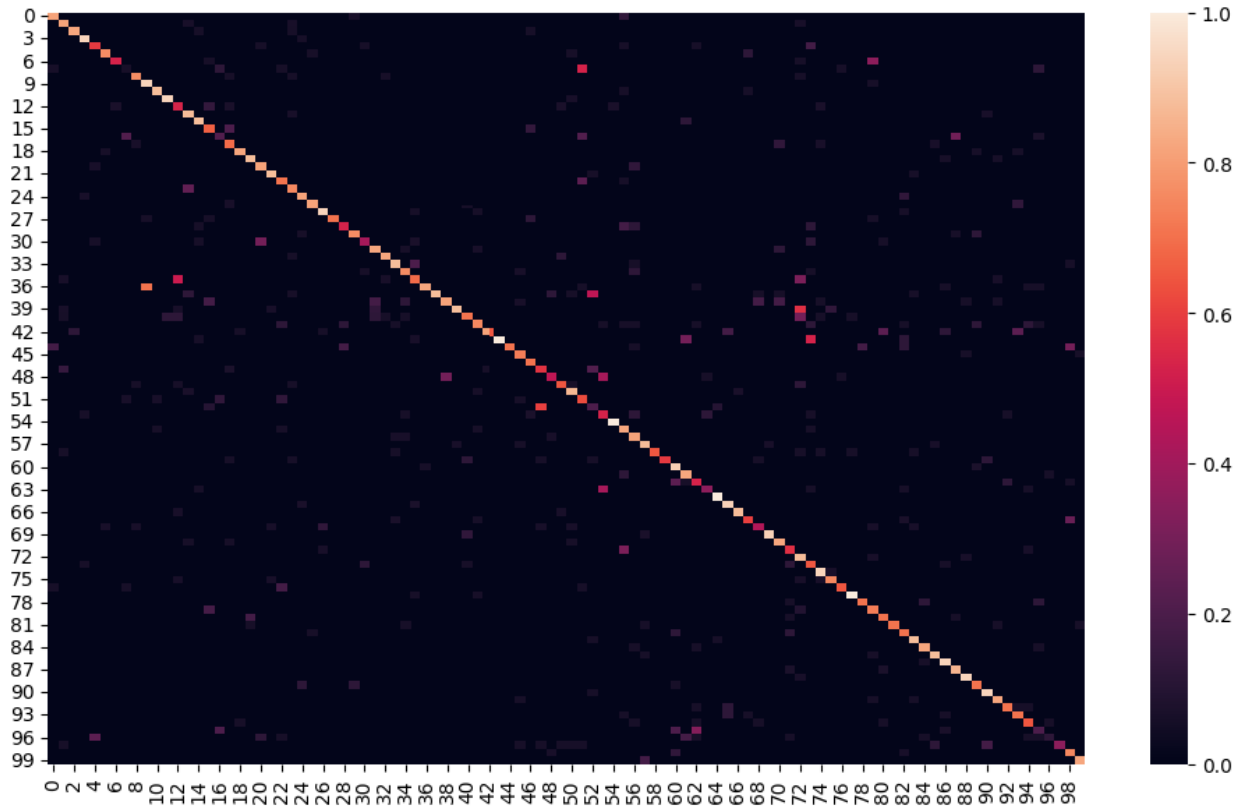

Figure 16: Confusion Matrix for Model using AUTSL Dataset

The following table presents the results of test accuracies obtained from our experiments with the additional dataset, AUTSL. The Transformer test accuracy is established by employing the model presented in section IV on the AUTSL dataset. The subsequent rows represent the performance of our approach.

Table 6: Comparisons of Optimization Methods on AUTSL Dataset

Model	Test Accuracy
Baseline - Transformer [22]	76.57%
with Data Augmentation and OpenPose	78.6%
with MediaPipe	81.08%
Ensemble of MediaPipe and OpenPose predictions	82.90%

VI. Discussion

From the results presented in Table 5, we observe that the performance in terms of accuracy was improved from employing different pose extractors as well as applying Machine Learning techniques such as data augmentation, hyperparameter tuning, and ensemble. Compared to our preliminary results with the appearance-based approach, the pose-based approach was vastly more efficient in rapidly learning identifying features. Since we were only able to obtain around 65% of the data compared to the original baseline work, our accuracies obtained from training limited data also suffered a significant decrease of 32%. However, using techniques such as data augmentation, we were able to improve the accuracy trained on the limited WLASL100 dataset by 8.78%. Further improvements were achieved by employing various pose extractors and ensemble technique. We can see the significant impact of having high quantity and quality of data that is apparent throughout our work.

Due to the lack of data we were able to obtain from the WLASL100 dataset, we extended the experiment to use the AUTSL dataset. The AUTSL dataset consists of ten times more data

points per gloss compared to WLASL100. We observe that the accuracies are much higher than obtained from the WLASL100 dataset. This suggests that the large amount of training data has a substantial impact on the model's ability to learn and generalize on SLR tasks.

When comparing our achieved results to other studies, we found that our top accuracy obtained was significantly higher than the original work on this dataset. The original test accuracy reported on Table 7 is obtained from [25] where the dataset was initially introduced. The proposed baseline model employs a multi-part CNN BLSTM architecture. A 2D CNN is used first to extract these features, followed by the integration of a feature pooling model (FPM) to obtain multi-scale feature representations. Finally, a BLSTM is used to model the spatio-temporal information for the SLR task.

Additionally, we consider some recent works that employ pose-based approach instead of appearance-based approach. These works include Holistic + OpenPose [26] and STGCN + LSTM by [27]. We report accuracy results of these works as well as our experiments in the table below. Our model’s performance was comparable to some recent works that focus on pose-based data.

Table 7: Survey of Performance on AUTSL Dataset

Model	Test Accuracy
CNN BLSTM [25]	63.22%
Holistic + OpenPose [26]	81.93%
ST-GCN + LSTM [27]	87.63%

VII. Conclusion and Future Work

In this work, we explored both major approaches to Sign Language Recognition. Appearance-based methods involve analyzing the visual features of the signer's hands and face, while pose-based methods typically focus on the position and movement of the signer's key points. The former approach typically requires a large amount of training data as well as extensive computational resources to achieve high performance. Conversely, the pose-based approach requires less computational power as much of the data is condensed into a set of 2D coordinate points. However, the accuracy of this approach tends to be lower as a result of the same abstraction process. In this project, we were able to achieve great performance and demonstrate the efficiency in training an SLR classifier using pose-based input data. Since the pose-extractors are pre-trained to extract key features, this takes away from the burden of an appearance-based model to learn and extract these features. Thus, a model trained from scratch using pose-based data requires less computing resources and is quicker at learning and inferring data while an appearance-based model would benefit to a greater degree from a pre-trained model.

Word-level sign language recognition is a fascinating yet demanding research area that poses challenges such as the subtleties in body motions and hand orientations, the vast size of the vocabulary, and the variability in sign interpretation depending on the context. In our research, we have explored various ML techniques, including data preprocessing, hyperparameter tuning, and multiple model implementations, to overcome these challenges and observed promising improvements in accuracy. Despite these improvements, there is still room for further research and development to better understand and model sign languages, particularly with regard to addressing issues of overfitting and limited training data.

Another approach to mitigate our limitation on shortage of available data on the WLASL100 set is to replace some glosses with lower video counts with glosses of higher count from the full WLASL2000 dataset. In consideration of time limitations, this aspect has not been incorporated in the current work but could be added in future work to obtain a more comprehensive view. Overall, the results observed on training the AUTSL dataset and evaluating the predictions exhibit a similar trend as the results of WLASL100. This indicates that the techniques employed in this research are consistent.

REFERENCES

[1] C. McCaskill, C. Lucas, R. Bayley, and J. Hill. "The Hidden Treasure of Black ASL: Its History and Structure." Gallaudet University Press Washington, DC, 2011.

[2] J. Shin, A. Matsuoka, M. Hasan, A. Srizo, "American Sign Language Alphabet Recognition by Extracting Feature from Hand Pose Estimation." *Sensors*. 2021; https://doi.org/10.3390/s21175856

[3] N. Adaloglou, et al. "A Comprehensive Study on Deep Learning-based Methods for Sign Language Recognition," in *IEEE Transactions on Multimedia*, 2021.

[4] P. Buehler, A. Zisserman and M. Everingham, "Learning sign language by watching TV (using weakly aligned subtitles)," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2961-2968, doi: 10.1109/CVPR.2009.5206523.

[5] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-scale Image Recognition." *arXiv preprint arXiv:1409.1556*, 2014.

[6] S. Herath, M. Harandi, and F. Porikli, "Going Deeper into Action Recognition: A Survey." *Image and Vision Computing,* 2017.

[7] R. Cui, H. Liu and C. Zhang, "Recurrent Convolutional Neural Networks for Continuous Sign Language Recognition by Staged Optimization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, doi: 10.1109/CVPR.2017.175.

[8] K. Cho, et al. "Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation." *arXiv preprint arXiv:1406.1078*, 2014.

[9] J. Carreira and A. Zisserman, “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset,” in *CVPR*, 2017.

[10] D. Li, et al. "Transferring Cross-Domain Knowledge For Video Sign Language Recognition." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[11] S. Yan, Y. Xiong, and D. Lin. "Spatial Temporal Graph Convolutional Networks For Skeleton-Based Action Recognition." *Thirty-Second AAAI Conference On Artificial Intelligence*, 2018.

[12] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden. “Multi-Channel Transformers For Multi Articulatory Sign Language Translation,” in *European Conference on Computer Vision*, pages 301–319. Springer, 2020

[13] P. Selvaraj, et al. "OpenHands: Making Sign Language Recognition Accessible with Pose-based Pretrained Models across Languages." *arXiv preprint arXiv:2110.05877*, 2021.

[14] J. Kenton, L.K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proceedings of NAACL-HLT*, June 2019.

[15] S. Jiang et al. “Skeleton Aware Multi-Modal Sign Language Recognition." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[16] A. Tunga, S. V. Nuthalapati, and J. Wachs. “Pose-based Sign Language Recognition using GCN and BERT,” in *IEEE Winter Conference on Applications of CV Workshops*, WACVW, 2021.

[17] R. Cui, H. Liu, and C. Zhang. “A Deep Neural Framework For Continuous Sign Language Recognition By Iterative Training.” *IEEE Transactions on Multimedia*, 2019.

[18] H. Hu, et al. "SignBERT: Pre-Training of Hand-Model-Aware Representation for Sign Language Recognition." Proceedings of the *IEEE/CVF International Conference on Computer Vision*. 2021.

[19] A. Hosain et al. “Hand Pose Guided 3d Pooling For Word-Level Sign Language Recognition.” In Proceedings of the *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3429–3439, 2021.

[20] J. Dong, et al. "Hierarchical Contrast for Unsupervised Skeleton-based Action Representation Learning." *arXiv preprint arXiv:2212.02082*, 2022.

[21] D. Li, et al. "Word-Level Deep Sign Language Recognition From Video: A New Large-Scale Dataset And Methods Comparison." Proceedings of the *IEEE/CFV Winter Conference on Applications of Computer Vision*, 2020.

[22] M. Boháček and M. Hrúz. "Sign Pose-Based Transformer for Word-Level Sign Language Recognition." Proceedings of the *IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022.

[23] Z. Cao et al. “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 15–20 June 2019.

[24] Contributors, “OpenMMLab Pose Estimation Toolbox and Benchmark.” 2020. Available online: https://github.com/openmmlab/MediaPipe (accessed on 22 February 2023)

[25] O.M. Sincan, H.Y. Keles, “AUTSL: A Large Scale Multi-Modal Turkish Sign Language Dataset and Baseline Methods.” *IEEE Access,* 2020.

[26] A. Moryossef, et al, “Evaluating the immediate applicability of pose estimation for sign language recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (IEEE)*, 3434–3440, 2021.

[27] O. Oğulcan, I. M. Baytaş, and L. Akarun, "Multi-cue Temporal Modeling for Skeleton-based Sign Language Recognition", in *Frontiers in Neuroscience*, vol.17, 2023.