Warwick University

CS907 Dissertation Project

Final Report

Mar Galiana Fernández

Supervised by: Dr Ahir Bhalerao

Abstract

This report presents the progress of the sign language recognition system developed for the dissertation project. The goal of the system is to convert American Sign Language gestures into written English text using videos of sign language interpreters as input. The report includes an overview of previous research in this field, details about the methodology used, and information about the completed experiments. It also outlines the planned next steps for the project.

2nd September 2023

Contents

List of Figures

List of Tables

1 Introduction

This chapter gives an overview of the techniques used for Sign Language Recognition. It describes the aim of the thesis and the motivation behind it.

1.1 Overview

Sign Language is the communication tool used for those having a speech or hearing impairment. There is no universal sign language, to be precise, there are almost 140 according to the Ethnologue [2], and nearly every country has its own national sign language and finger-spelling alphabet. Hence there is a need for systems capable of recognising the sign gestures and conveying the message to the population that does have no knowledge of it. These systems are called Sign Language Recognition (SLR). In this report, the main objective will be to evaluate the state-of-the-art, compare it with our results and identify the next steps.

1.2 Motivation

Sign languages are not studied in school and only a tiny part of the world's population is proficient in them. This causes a problem when a person with a speech or hearing impairment tries to communicate an emergency or an everyday task. Nowadays, people are more aware of this situation and attempt to solve it by learning about them. However, at the moment, it is not enough to build a society that includes people living with these disabilities. [3, 4]

This research project aims to accelerate this process in order to reach this state of inclusivity sooner.

1.3 Aim of the Thesis

The aim of this thesis is to develop sign language recognition based on American Sign Language (ASL). We have chosen ASL as it is the one for which the most data is available. The inputs of the system to implement will be videos of an interpreter and the texts corresponding to each sign will be the outputs. We have decided to implement this system using videos as input data due to the fact that these can be obtained using a camera, an easy and economical

device. As we will see in section 2, most of the existing research has been done using expensive and complicated to use sensors. Our mission is to help with the communication process for those people who cannot communicate with verbal language, so it is important to try to minimise and facilitate as much as possible the use of the system to be developed.

In this project, existing Sign Language Recognition systems will be studied and evaluated in order to find the most promising one according to our requirements, which are explained in section 4. We will test several data pre-processing techniques, algorithms, optimisations and evaluation metrics, in order to come up with a proposal for an improved SLR system.

The report is organized as follows: Section 2 provides an overview of past research and advancements made in the same fields. In Section 4, the progress made so far and the results of the conducted experiments are presented. Section NONE outlines the upcoming steps and future directions of the project. Subsequent sections discuss the limitations, ethical considerations, and project management aspects. The report concludes with Section 8, which presents the achieved conclusions.

1.4 Proposed Idea

This project aims to develop a sign language recognition system based on ASL. The system will take videos of individual interpreters performing ASL gestures as input and provide translations of these signs into English text as output. Throughout the project, a series of experiments will be conducted, incorporating existing studies and novel approaches to enhance the system. These experiments can be categorised into four phases: data selection, data processing, model selection, and optimisation.

Two experiments have been conducted, utilising the same dataset but employing different models and data processing techniques. The first experiment uses a 3D Convolutional Neural Network (3DCNN), while the second experiment employs a SlowFast Neural Network. Each experiment will be elaborated on in the following sections, aligned with the four established phases.

The 3DCNN model was chosen because of its high performance proven in the state-of-the-art literature. In contrast, the SlowFast Neural Network incorporates two channels: one dedicated to detecting the slow frequencies of the video, and the other channel focuses on capturing the high frequencies.

This approach has demonstrated significant utility in action recognition [5] as well as Sign Language Recognition [6]. In the context of sign language interpretation, interpreter videos primarily exhibit movement in specific regions, which can be efficiently processed through the fast channel. Therefore, we have high expectations for the promising results that can be achieved through the implementation of this technique.

2 Background and research

Various techniques have been used in Sign Language Recognition (SLR) based on system requirements. Wadhawan and Kumar [1] summarised the rates of different papers on American Sign Language (ASL) training models. They compared the output rates based on five characteristics:

- Data Acquisition: Different methods were used to acquire data, including cameras and sensors. Cameras provided either images or videos, while sensors included gloves, Kinect, arm sensors, electroencephalogram, and leap motion.
- Type of Signs: Signs can be static (no movement, e.g., ASL alphabet) or dynamic (requiring movement for interpretation). Static signs often used camera-acquired images, while dynamic signs used videos.
- Modelling Algorithm: Various algorithms and techniques were employed. Previous papers using cameras and dynamic signs used Dynamic Time Warping (DTW) and Support Vector Machine (SVM).

Wadhawan and Kumar [1] also discuss other aspects regarding the dataset used to train the model. The main characteristics are:

- Interpreter mode. Comparing whether it is either isolated, continuous or both. Isolated signing refers to independent signs without any connections to preceding or succeeding signs. Continuous signing involves several signs without distinct pauses between them.
- Number of hands needed to interpret the signs. It can be single- or double-handed.

Table 1 summarises the above information. It is taken from the article by Wadhawan and Kumar [1], although only the information relevant to this article has been retained. Only the articles training the model with data containing dynamic signs or both (dynamic and static) are presented in the table.

Paper	**Data Acquisition**	**Gestures**	**Technique**	**Rate**
Oz and leu [7]	Gloves	Dynamic	NN	95%
Sun et al [8]	Kinect	Dynamic	Latent SVM	86%
Sun et al. [9]	Kinect	Dynamic	Adaboost	86.8%
Jangyodsuk et al. [10]	Camera	Both	DTW	93.38%
	Kinect	Both	DTW	92.54%
Wu et al. [11]	Arm sensors	Dynamic	Decision tree	81.88%
			SVM	99.09%
			NN	98.56%
			Naïve Bayes	84.11%
Usachokcharoen et al. [12]	Kinect	Dynamic	SVM	95%
Savur and Sahin [13]	Arm band	Both	SVM	82.3%
Sun et al. [14]	Kinect	Both	Latent SVM	86%
Kumar et al. [15, 16]	Camera	Static	SVM	93%
		Dynamic	SVM	100%
Savur and Sahin [17]	Armband	Dynamic	SVM and ensemble learner	60.85%
AlQattan and Sepulveda [18]	Electroence-phalogram	Dynamic	LDA	75%
			SVM	76%

Table 1: Summarised review of ASL recognition systems comparing the Reported Rate obtained. This table is located in the Sign Language Recognition Systems: A Decade Systematic Literature Review article [1].

From table 1, the most remarkable papers for this report are the ones from Jangyodsuk et al. [10] and Kumar et al. [15, 16]. Both researchers have used cameras as the data acquisition method, and the data set used to train the model uses dynamic signs.

Kumar et al. [15, 16] developed a sign language recognition system for recognising both static and dynamic signs in American Sign Language. They focused on predicting the letters a-z (where only the letters j and z are dynamic). The system was able to perform dynamic backgrounds with minimal decorations, as it relies on skin colour segmentation to identify gestures. Since the signs they wanted to predict do not use a facial expression, they removed this section of the video using Viola-Jones face detection followed by subtraction of the detected region. Once the data was processed, they extracted a curved feature vector, following previous work from Bhuyan et al.,

[19]. Afterwards, these feature vectors were classified using pre-trained SVM classifiers. Static and dynamic gestures were differentiated by measuring the distance travelled by the hand in subsequent frames. Dynamic gesture recognition was performed using four gestures for testing, which are: "j", "z", "no", and "goodbye", achieving an accuracy of 100%.

Jangyodsuk et al. [10] employed a dataset consisting of videos taken with a standard RGB camera, with three signers each making 1,113 signs, for a total of 3,339 different signs. They followed previous work by Dalal et al. [20], who applied the Dynamic Time Warping (DTW) method to compare the hand trajectory using a Histogram of Oriented Gradient (HoG) to represent hand shape. However, they standardised the features to have a mean value of 0 and a standard deviation of 1. With this improvement, their accuracy increased, on average, by about 10%.

Using Hand trajectory matching with hand shape distance using HoG features as shape representation, they archived an accuracy of 93.38

There exist other papers published after the Wadhawan and Kumar article [1] was released which have other interesting methods applied.

Borg and Camilleri [21] implemented, in 2020, a two-stage system, which has the hand key points features obtained via OpenPose as the input to the model. They use Hidden Markov models (HMMs) to obtain the subunits of sign concatenation (SU). The SU descriptors are employed to train the SU Recurrent Neural Networks (RNNs), using a Connectionist Temporal Classification (CTC) framework to handle the temporal sequence. Once the SU RNNs are trained, a second-level RNN is added for sign recognition. RWTH-Phoenix Weather [22] was the dataset used which contained 1230 unique signs with 9 signers. With this implementation, they archived an accuracy of 71.9%.

Zheng et al. [23] proposed a model that reduced the training size data by 9.3%, compared with the state-of-the-art of 2020. This model used the Frame Stream Density Compression (FSDC) algorithm to detect and reduce redundant similar frames, which shortens long sign sentences without losing information. They further implemented a temporal convolution (T-Conv) connected to a dynamic hierarchical bidirectional GRU (DH-BiGRU). The RWTH-Phoenix Weather Dataset [22] was used, which is the same as the one utilised by Borg and Camilleri [21].

Al-Hammadi et al. [24] used a 3D Convolutional Neural Network (3DCNN)

where three instances of the 3DCNN structure were trained to extract the hand gesture features from the beginning, middle, and end of the video sample. Afterwards, they studied three techniques for feature fusion: multilayer perceptron (MLP) neural network, long short-term memory (LSTM) network, and stacked autoencoder. Using the 3DCNN and MLP in a 40 classes dataset called KSU-SSL dataset, they reached a recognition rate of 84.38%.

Finally, Hassan et al. [6] conducted an experiment using various models, with the most successful one being the SlowFast Neural Network. They utilized a pre-trained model called *SLOWFAST_8×8_R50* obtained from the PySlowFast GitHub repository [5]. The experiment utilized the WLASL [25] dataset and achieved predictions for 300 different labels, resulting in a TOP 1 accuracy of 79.34%, which implied an improvement of 23.2% over the previous state-of-the-art performance. The researchers mentioned that the limitations they encountered were related to the time-consuming nature of model training, which required a total of twenty-four hours spread across multiple days, as well as some hardware limitations.

Even though there is intensive research conducted on gesture recognition, the majority of them have been executed using data coming from complex hardware and inconvenient for the user to carry on a daily basis. Research so far indicates that the architectures being used are vastly differing, therefore it is necessary to bring all the information together and try to get the best out of each one of them.

3 Experiments

3.1 Experiment 1: First model implementation

The objective of this experiment was to develop an initial model as a preliminary attempt, aimed at understanding the complexities of the problem being addressed.

As the primary focus of this experiment was not to achieve maximum performance, the chosen model configuration was not optimised. The number of epochs and batch size were intentionally kept at their minimum values to accelerate the results without enduring an extended training duration.

3.1.1 Data selection

To begin with, it was essential to select an appropriate dataset that could be utilised in future experiments. Additionally, another significant choice to make was the proportion of training, testing, and validation data, which would be contingent on the number of videos obtained from each label.

Numerous datasets were evaluated, all sourced from other papers that had also implemented an SLR using the ASL. This choice was deliberate as it provides a more accurate basis for comparing the performance of other models with that presented in this report under similar conditions.

The dataset needed to contain labelled videos, as the data acquisition and labelling are beyond the scope of this project. The following datasets have been examined in order to determine which was going to be the most accurate to the project requirements.

- Word-Level American Sign Language (WLASL) [25]
- MS-ASL dataset [26]
- The American Sign Language Lexicon Video Dataset (ASLLVD) [27].
- Datasets from the Kaggle [1] community.

[1] https://www.kaggle.com

The ASLLVD was gathered at Boston University and encompassed more than 3,000 signs produced by 1-6 native ASL signers. Sadly, this dataset had to be excluded due to the requirement of submitting a petition for download, which was never approved. Among the options available on Kaggle, the sign count in each dataset was considerably lower and less reliable compared to other datasets.

The samples within the MS-ASL dataset were collected by Microsoft, featuring 1,000 distinct labels and involving over 200 signers. Furthermore, the WLASL dataset offers a wider range of backgrounds, speaker speeds, physiques, and camera orientations. Additionally, it is the largest ASL dataset available, featuring 2,000 common words in ASL. Both the MS-ASL and the WLASL datasets were viable options, but eventually, the WLASL was chosen, as it was the most commonly used dataset, which helps compare the results of the proposed model with the results from other papers.

Once the dataset was downloaded, 10 random labels were chosen: drink, trade, before, bowling, computer, cool, go, thin, help and tall. There were 12 samples for each label, leaving 8 for training and 2 for testing and validation purposes. Although this number of samples may not be sufficient to train and validate a robust model, it is important to note that the purpose of this experiment is to train a model for the first time, understand the data processing involved, and anticipate the type of results that will be obtained. Specifically, 70% of the samples were allocated for training, while 20% were used for validation and the remaining 20% for testing.

3.1.2 Data processing

In the first experiment, a series of data processing tests were conducted to determine the relevant portion of the sign interpretation within the video. For each video, 20 consecutive frames were extracted from four different starting points. These frames were obtained from the beginning, middle, and end of the video, as well as randomly selected positions. This approach helped reduce the data size by 10 frames per video while capturing different sections of the sign interpretation.

3.1.3 Model selection

After an in-depth exploration of the state-of-the-art in section 2, a range of promising techniques came under evaluation. These encompassed:

- Support Vector Machine [15, 16].
- Dynamic Time Warping Using a Histogram of the Orientated Gradient [10].
- Hidden Markov models combined with Recurrent Neural Network [21].
- Temporal convolution network connected to a dynamic hierarchical bi-directional GRU. [23].
- 3D Convolutional Neural Network using a multilayer perceptron for feature fusion. [24].
- Slow Fast Neural Network [6].

Based on the outcomes derived from these techniques, the 3DCNN model emerged as the preferred choice for the initial experiment. Specifically, the selected model was the pre-trained MoViNet-A0-Base model sourced from TensorFlow [2]. This particular model had undergone training on the kinetics dataset, achieving a TOP 1 accuracy of 72.28% while operating with an input size of 50x172x172 pixels.

In this context, the model was re-trained using the 10 distinct labels mentioned in section 3.1.1. This retraining involved using a batch size of 8 over the course of 10 epochs, using 20 frames from each video in the training process. Each label was assigned 12 samples for retraining.

3.1.4 Results

The outcomes of each data processing test are meticulously documented in table 2.

A notable observation is the 20% decrease in model accuracy when utilizing the final frames of the video. This outcome implies that the pivotal segment of the video lies within both the initial and middle sections, whereas

[2]https://github.com/Atze00/MoViNet-pytorch

Test	Frames position	Accuracy
1	Beginning	40%
2	Middle	40%
3	End	20%
4	Random Start	40%

Table 2: Results of the second experiment after optimising the model

the end of the video proves less informative. In particular, even when starting from random positions, the accuracy remains consistent at 40%, confirming the importance of both the beginning and middle sections to capture essential information for accurate interpretation.

The results findings of this experiment are as expected due to the fact that at the last moments of each video, the signers often await the recording's conclusion without contributing any valuable sign content.

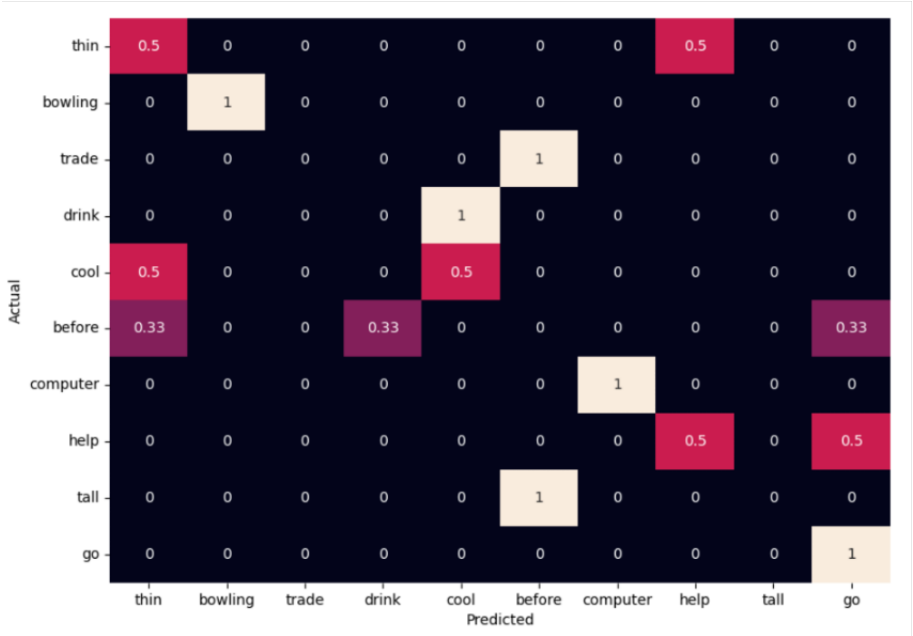

Figure 1: The confusion matrix when testing the 3DCNN model in the first experiment.

Looking at the results from the confusion matrix in figure 1, it's noticeable that the labels "Trade," "Tall," and "Before" consistently get mixed up. After closely examining these labels, it's clear that they all involve signs performed in the same space using both hands and without much movement

in other parts of the body. This similarity in how these signs are executed is likely causing the model to confuse them with each other.

3.2 Experiment 2: Compare the SlowFast neural network to 3DCNN

The aim of the second experiment was to introduce the SlowFast Neural Network and compare its performance with that of the 3DCNN, specifically by focusing on the body areas of the videos that hold the sign information.

The intuition behind this experiment stems from the inherent model architecture of the SlowFast Neural Network. This architecture is believed to offer advantages in the realm of sign language recognition, as discussed in Section 1.4. Moreover, the experiment hypothesises that by concentrating on the primary movement areas within signs, performance enhancements can be achieved.

3.2.1 Data selection

For this experiment, the number of labels was increased by 10, resulting in a total of 20 labels. These labels were randomly selected and included words such as book, drink, computer, before, chair, go, clothes, who, candy, cousin, dead, fine, no, thin, walk, year, yes, all, black, and help.

Additionally, the number of samples per video was elevated. Models were trained using 22 videos for each label, all obtained from the WLASL dataset. The data allocation consisted of 70% for training, 15% for testing, and the remaining 15% for validation.

3.2.2 Data processing

As discussed in Section 1, signs are distinguishable by factors such as hand gestures, facial expressions, and body movements. Building upon this understanding, the videos were processed to highlight these specific aspects of the signer's body, with a focus on detecting hands and the face in each frame of

every sample.

Each video frame underwent additional processing to test four different improvements and determine which ones enhanced the models' performance. These enhancements were selected based on the body parts visible in the original image. The various types of enhancements can be seen in Figures 2, 4, 3, and 5. These figures illustrate the four distinct ways in which the images were altered. In each image, a person is shown using sign language. On the left, the unmodified image is visible, while on the right, the adjusted version designed to facilitate model comprehension is presented.

The first type, referred to as "ALL" during the experiment, is depicted in Figure 2. This image shows a person using sign language to convey the word "secretary." The modified image is divided into four sections: the upper-left part focuses on the left hand, the upper-right part accentuates the right hand, the lower-left part highlights the face, and the lower-right part displays the entire image with a subtle blur. The blur is applied to help highlight additional details while retaining the main context.

Figure 2: Visual representation of video processing in Experiment 2. The image on the left presents the untouched original, while the image on the right showcases the result after employing the "ALL" processing approach.

In the second type, which will be named "FACE AND HANDS" during the execution of the experiment, the input frames are transformed to the model as shown in the example illustrated in Figure 3. The illustration portrays an interpreter performing the ASL sign for the word "angry". This

processed image is divided into three sections: the top left zooms in on the left hand, the top right emphasises the right hand, an the total weight of the bottom highlights the face. The intuition behind this experiment is that the significant part of the sign interpretation relies on the hands gesture and face expression, the rest of the body is static, ergo it is irrelevant information that the model does not need to understand the sign.

Figure 3: Visual representation of video processing in Experiment 2. The image on the left presents the untouched original, while the image on the right showcases the result after employing the "FACE AND HANDS" processing approach.

The third category, denoted as "BODY AND HANDS" throughout the experiment, involves transforming the input frames for the model, as exemplified in the illustration of Figure 4. This depiction features an interpreter enacting the ASL sign for the word "book." The altered image shares similarities with the "FACE AND HANDS" processing approach. However, in this instance, the entirety of the bottom section shows the complete image with a subtle blur. The rationale behind this processing experiment is that although certain signs require a facial expression to be represented accurately, these instances are relatively infrequent. Most often, the uniqueness of a sign resides in the hand movements themselves. As such, the face need not be prominently highlighted; including the entire image in the lower part of the frame provides sufficient context for interpreting the sign. The introduction of the blur serves to emphasise other relevant aspects while preserving the overall context.

Figure 4: Visual representation of video processing in Experiment 2. The image on the left presents the untouched original, while the image on the right showcases the result after employing the "BODY AND HANDS" processing approach.

The final processing technique, referred to as "HANDS" throughout the experiment, involves transforming the input frames for the model. This technique is illustrated in Figure 5, where an interpreter is depicted performing the ASL sign for the word "help." The adjusted frame comprises two images: one showcasing the left hand and the other displaying the right hand. Both the face and the original image have been omitted from this version. This approach is based on the idea behind the "BODY AND HANDS" image processing technique, which aims to reduce the focus on the face. Furthermore, the exclusion of the original image is based on the understanding that the most important information in sign language is contained within the hand movements. By emphasising the hands over other visual elements, the model's attention is narrowed, thus improving its ability to recognise and accurately interpret the complex gestures of sign language.

To detect hands in the original images, a variety of libraries were used. Initially, OpenPose was used, a choice also made by Borg and Camilleri [21]. However, OpenPose's outcomes were rather imprecise, prompting the exploration of alternative libraries. Ultimately, Mediapipe was chosen for hand detection and differentiation between the left and right hands. Although slight errors may still exist in certain frames, Mediapipe showed considerable improvement compared to OpenPose.

For face detection, the library "face_recognition," developed using ad-

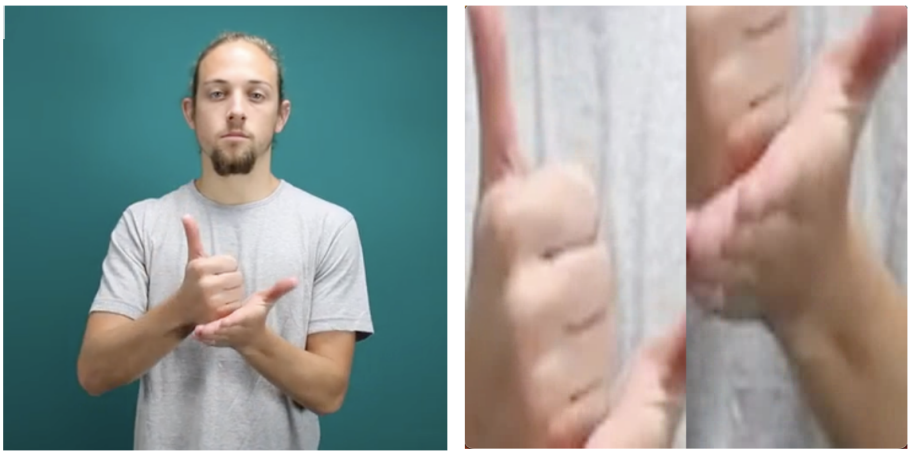

Figure 5: isual representation of video processing in Experiment 2. The image on the left presents the untouched original, while the image on the right showcases the result after employing the "HANDS" processing approach.

vanced techniques from dlib, was employed. This library consistently delivers accurate results in face-detecting tasks.

The completed implementation for this stage of the experiment is available within the script titled "processing/process_videos.py." It can be found in the specified GitHub repository in Section 1.

3.2.3 Model selection

Two models were employed for training within this experiment: the SlowFast Neural Network and the 3DCNN. Both models had previously been pre-training, which helped reduce the computational requirements when retraining them with the WLASL dataset.

The SlowFast Neural Network, created and trained by Facebook [28], was trained using the Kinetics-400 [29], Kinetics-600 [30], Charades [31], and AVA [32] datasets. Damen et. al. [5] achieved a TOP 1 accuracy of 77% using this model with the same dataset employed in this experiment.

For the 3DCNN model used in the initial experiment, its selection was based on its simplicity and efficiency in terms of memory and computational time. However, as the objective of this experiment was to attain reliable performance, a different pre-trained model was necessary. The chosen model

is known as R3D18 and was obtained from Torchvision [33]. This model had been pre-trained for action recognition purposes and achieved an accuracy of 74.3% using the Kinetics dataset [34].

3.2.4 Results

The precision obtained training the 3DCNN and the SlowFast Neural Network using the four types of processing techniques mentioned in Section 3.2.2 is summarised in Table 3.

Processing data type	**3DCNN**	**SlowFast**
All	14.58%	58.33%
Face and Hands	14.58%	62.50%
Body and Hands	18.37%	**63,27%**
Hands	**26,53%**	53.06%

Table 3: A comparison of accuracy performance between the 3DCNN and SlowFast Neural Network models, considering four different types of data processing.

For the ALL processing data type, the SlowFast Neural Network achieved an accuracy of 58.33%, as depicted in Figure 7 for the loss function and Figure 6 for the confusion matrix in Section A.1. Using the 3DCNN approach, an accuracy of 14.58% was attained, with its loss function showcased in Figure 9 and the corresponding confusion matrix in Figure 8, both available in Section A.2.

For your convenience, the code utilised in training the 3DCNN can be accessed through the following code repository: GitHub link. To replicate the SlowFast experiment, the relevant code can be found within this experiment's code repository: GitHub link.

Similarly, for the FACE AND HAND processing data type, the SlowFast Neural Network achieved an accuracy of 62.50% as shown in Figure 15 for the loss function and Figure 14 for the confusion matrix in Section A.5. Utilizing the 3DCNN approach yielded an accuracy of 14.58% with its loss function illustrated in Figure 17 and the corresponding confusion matrix in Figure 16, both available in Section A.6. The relevant code of this experiment can be found in the following code repository GitHub link.

In the case of the BODY AND HAND processing data type, the SlowFast Neural Network achieved an accuracy of 63.27% as highlighted in Figure 11 for the loss function and Figure 10 for the confusion matrix in Section A.3. The 3DCNN approach resulted in an accuracy of 18.37% with its loss function depicted in Figure 13 and the corresponding confusion matrix in Figure 12, both available in Section A.4. The relevant code of this experiment can be found in the following code repository GitHub link.

Lastly, employing the HANDS processing data type led to a SlowFast neural network accuracy of 53. 06%, with the loss function displayed in Figure 19 and the confusion matrix in Figure 18 within Section A.7. Using the 3DCNN approach resulted in an accuracy of 26.53%, with its loss function presented in Figure 21 and the corresponding confusion matrix in Figure 20, both available in Section A.8. The relevant code for this experiment can be found in the code repository GitHub link.

Analysing the results obtained, it can be observed how the SlowFast Neural Network gives a higher performance in all data types processed, archiving a top accuracy of 63.27% when using the body and hands processing type. And taking a look at the 3DCNN, it seems like it is not even learning from the training, given the low accuracy that it is returning. The 3DCNN is a commonly used approach for video recognition, as mentioned in the state-of-the-art section 2, which implies that it should have a higher accuracy than the one obtained from this experiment. The following experiments will be focused on understanding what is happening to the model and try to improve it.

Upon analysing the results obtained, it becomes evident that the SlowFast Neural Network consistently outperforms the 3DCNN across all processing data types. In particular, the SlowFast neural network achieved a remarkable accuracy peak of 63.27% when applied to the body and hands processing type.

However, a closer examination of the 3DCNN's performance reveals a concerning trend: it appears to struggle in acquiring meaningful insights from the training data, leading to a noticeably low accuracy rate. It can be verified evaluating the loss function and confusion matrix of the four different data processing techniques when using the 3DCNN model. Given the significance of 3DCNN as a widely used technique for video recognition, as mentioned in Section 2, the disparity between its anticipated performance and the observed results is concerning.

The following experiments will be directed towards uncovering the factors contributing to this underperformance and formulating strategies to enhance the 3DCNN's efficacy. This proactive investigation is crucial for gaining insights into the model's limitations and, ultimately, for steering improvements in its functionality.

3.3 Experiment 3: Join Datasets

The primary objective of the third experiment was to enhance the performance of the 3DCNN model. The rationale behind this initiative stems from the observation that the model's learning capacity was potentially limited because of the restricted number of available samples. As detailed in the analysis mentioned in Section 3.1.1, a preliminary study was conducted on the existing ASL dataset.

The proposed solution was to augment the existing dataset with a new one. This augmentation aimed not only to increase the dataset's overall sample size but also to facilitate improved generalisation by providing the model with a more diverse range of examples to learn from.

Given the use of two different datasets, this experiment also explored how combining them can enhance the model's performance. This evaluation involved deciding whether one dataset should be primarily used for training, or if both should be divided and used for complementary purposes.

3.3.1 Experiments dataset requirement

The primary dataset used in this experiment remained consistent with the dataset employed in previous experiments, named the WLASL dataset [25]. MSASL [26] was the secondary dataset. As detailed in Section 3.1.1, the choice of this dataset was identified as the most suitable option.

More labels were added to train the model, with a total of 50 labels. Given the necessity of downloading data, it was logical to do so collectively. However, this time the labels were manually selected, taking into account the type of gesture and the significance of the word. Priority was given to

words that require early acquisition.

Upcoming experiments will delve into how various gestures impact data processing. Therefore, for this experiment, it was ensured that three types of signs were included: face motions (expressing emotions such as anger or happiness), body motions (involving hands, shoulders and arms, like the sign for "big"), and hand gestures (primarily involving hand movement). A significant proportion of signs involved hand movement, encompassing 68% of the selected signs, while face and body gestures each constituted 16%. Specifics can be found in Table 4.

Gesture Type	**Included Labels**
Face Motion	Laugh, cry, angry, surprise, happy, think and delicious.
Body Motion	Dance, fly, hello, teacher, swim, important, crazy and swing.
Hand Motion	Sister, bird, book, friend, doctor, eat, nice, yes, learn, no, like, want, deaf, school, finish, white, fish, sad, table, father, milk, brother, paper, mother, water, help, yellow, hungry, drink, careful, coffee, phone and more.

Table 4: Labels categorised by gesture type in the 50-label dataset.

For the execution of this experiment, acquiring the new dataset required certain steps. This process demanded increased computational memory and time. Although downloading around 22 videos for each of the 50 labels in the WLASL dataset took approximately 2 hours, the new dataset required approximately 10 hours. This increase in download duration can be attributed to two primary factors. Firstly, the MSASL dataset contained double the number of videos compared to the WLASL dataset. Secondly, the new dataset's videos were notably longer in duration, often extending up to five minutes, in contrast to the typical three-second duration of regular sign videos.

A distinctive feature of the videos in the new dataset was their extended content, incorporating multiple signs within a single video. This inherent characteristic resulted in the prolonged duration of the video. The challenge lay in the fact that the downloaded videos needed subsequent editing to retain only the pertinent sections.

Throughout this experiment, the ALL data processing technique was applied consistently. The plan was to evaluate other methods once the model demonstrates successful learning.

3.3.2 Using WLASL dataset for Training

While the primary goal of this experiment was to enhance the performance of the 3DCNN model, it will also be conducted on the SlowFast neural network to assess whether increasing the sample size could lead to improvements. The resulting performance of each model, now that the sample size has been doubled, is summarised in Table 5, which presents the outcomes of this experiment.

Training Dataset	**Validation/Test Dataset**	**3DCNN**	**SlowFast**
WLASL	WLASL	**14.58%**	**58.33%**
WLASL	MSASL	3.60%	44.44%

Table 5: Accuracy achieved by the 3DCNN and SlowFast models, trained using the WLASL dataset and evaluated on the MSASL dataset. Both datasets were previously processed using the ALL data processing type.

For the 3DCNN, the confusion matrix and loss function can be found in Figures 24 and 25 respectively, detailed in Section B.2. These visualisations highlight that the model is still struggling to learn, as evidenced by the accuracy dropping to 3.60%. On the other hand, the SlowFast model exhibits learning capabilities, as indicated by the confusion matrix in Figure 22 and the loss function in Figure 23, both available in Section B.1. However, the precision has decreased from 58.33% to 44.44%. The code for this experiment can be found in the repository GitHub link.

The reduction in accuracy in both models, despite the augmented sample size, may be attributed to the substantial disparities between the MSASL and WLASL datasets. This intuition will be further explored in upcoming experiments.

3.3.3 Increase batch size using mixed precision training

As increasing the sample size did not yield significant improvements in the 3DCNN model's performance, another potential issue could be the batch size used during the training process. In previous experiments, the batch size was set to 5. Attempting to increase it to 16 resulted in a CUDA out of memory error. Upon further investigation, it was discovered that employing mixed precision training, which reduces the number of decimal places in the neural network's weights to conserve memory, might be a viable solution.

Implementing mixed precision training implies that the model may not attain its peak performance, but it raises the possibility that, when combined with an increased batch size, it could outperform the existing small batch size configuration.

Shifting to CPU-based training rather than utilising the GPU was not a viable alternative. Such an approach would have considerably prolonged the training time, imposing substantial constraints on the project's scope and feasibility. Unfortunately, the CUDA capacity was only capable of training the models using a batch size of 10, any higher number would result in an out of memory error.

Table 6 provides a summary of the accuracy achieved in the various trial runs of the current experiment. Notably, the accuracy of the 3DCNN increased from the previous trial; however, it was insufficient to deem this training successful. This is evident from its loss function in Figure 29 and the confusion matrix in Figure 28, both detailed in Section B.4. The model is not effectively learning.

Conversely, the SlowFast Neural Network's performance deteriorated by 6.6% and showed no improvement when the batch size was increased. The loss function during training can be found in Figure 27, and the confusion matrix is presented in Figure 26, both available in Section B.3.

It's worth noting that increasing the batch size without employing Mixed Precision would likely result in performance improvement rather than degradation. However, due to GPU memory limitations, this test could not be conducted.

The code from this experiment can be found in the repository GitHub

Training Dataset	**Val/Test Dataset**	**3DCNN**	SlowFast
MSASL	WLASL	14.58%	58.33%
WLASL	MSASL	3.60%	44.44%
WLASL increasing batch size and using Mixed Precision	MSASL	12.01%	37.84%

Table 6: Accuracy achieved by the 3DCNN and SlowFast models, trained using the WLASL dataset and evaluated on the MSASL dataset increasing the batch size to 10 and applying the mixed precision technique. Both datasets were previously processed using the ALL data processing type.

link.

3.3.4 Exploring Training and Testing Dataset Combinations

Despite being a commonly used model for video recognition with high accuracy expectations, the 3DCNN model was not performing as anticipated under the same conditions as the SlowFast Neural Network. Consequently, a decision was made to switch to a different pre-trained model.

The selected pre-trained model, "i3d_r50" [35], was developed by Facebook and is accessible through PyTorch. This model, trained on the Kinetics dataset, demonstrated impressive performance in action recognition, boasting an accuracy of 74.3% [36].

This experiment also sought to identify the most suitable dataset for both training and testing. It explored whether the MSASL dataset was more appropriate for training and the WLASL dataset for testing, vice versa, or if a balanced combination of both for training and testing would yield the best results. The performance of the pre-trained model under these various dataset combinations is detailed in Table 7.

Given the larger size of the MSASL dataset, when used for training, it necessitated reducing the batch size to 5 and activating Mixed Precision, which in turn resulted in lower accuracy, as evidenced in Section 3.3.3. To maintain consistency across all trials in this experiment, the low batch size and Mixed Precision configurations were retained, even when the WLASL dataset was used for training.

Training Dataset	Val/Test Dataset	3DCNN	SlowFast
MSASL	WLASL	55.24%	36.36%
WLASL	MSASL	20.94%	32.53%
MIX	MIX	**62.84%**	**57.55%**

Table 7: Performance Comparison of 3DCNN and SlowFast Models Using Different Training and Testing Datasets

The performance of the pre-trained model was indeed influenced by the choice of training and testing datasets, as demonstrated in Table 7. It's noteworthy that the accuracy of the new pre-trained model improved compared to previous trials, rising from 14.58% (Table 6) to 20.94% (Table 7). Figures 32 and 33 in Section B.5.2 showcase the model's confusion matrix and loss function, respectively, during training with the WLASL dataset. These visualisations indicate that the model is indeed learning from the training set, despite the relatively low accuracy. However, this accuracy is insufficient to declare this model successful.

Conversely, when trained using the MSASL dataset, the 3DCNN's accuracy increased substantially to 55.25%, as evidenced by Figures 36 and 37 in Section B.5.4. The highest accuracy of 62.84% was achieved when both datasets were used for testing and training, as seen in Figures 40 and 41 in Section B.5.6. This result is remarkable, considering that the original model was achieving 74.3% accuracy during the training process and that the current model had not yet been optimised.

In the case of the SlowFast Neural Network, the reduction in batch size to 5 due to the necessity of conducting several tests in this trial resulted in a decrease in accuracy when training with the WLASL dataset. The accuracy dropped to 32.53%, a decrease of approximately 12% compared to when the batch size was set to 10. These findings are illustrated in Figure 30 (confusion matrix) and Figure 31 (loss function), both available in Section B.5.1. This outcome suggests that increasing the batch size, as observed in the previous experiment (Section 3.3.3), had a positive impact on the model by counteracting the accuracy decrease caused by Mixed Precision.

However, when the MSASL dataset was used for training, the SlowFast accuracy increased to 36.36%. Figure 34 (confusion matrix) and Figure 35 (loss function), both located in Section B.5.3, demonstrate this improvement.

The highest accuracy was achieved when both datasets were combined for training and testing, mirroring the 3DCNN results. This combination yielded an accuracy of 57.55%, as depicted in Figure 38 (confusion matrix) and Figure 39 (loss function), both accessible in Section B.5.5. This outcome aligns with expectations, as training on diverse situations, camera orientations, and with different interpreters likely contributes to improved model performance. Additionally, the training set becomes more similar to the test set when both datasets are used, potentially explaining the enhanced accuracy.

Notably, both models exhibited higher accuracy when the MSASL dataset was used for training rather than testing, which makes sense given its larger sample size. The code implemented for this experiment is available in the following repository: GitHub link.

4 Results

5 Project Management

The project has been managed and monitored using a dedicated GitHub repository, which will be made accessible for public viewing upon the final submission of the dissertation project report. For implementation purposes, I have utilised my personal computer as well as the servers provided by the university to leverage additional computational capacity.

Supervision of the project is being conducted by Dr Ahir Bhalerao, with whom I have regular meetings every two weeks. These meetings serve as opportunities to review the progress, discuss new developments, analyse results, and plan the next steps of the project. These regular meetings with my supervisor ensure effective guidance and facilitate a smooth progression of the research work.

6 Appraisal and reflection

During the course of the mentioned experiments (see 4), several challenges have been encountered. Initially, hardware limitations necessitated the use of university-provided servers for implementing the experiments. However, each student had limited space allocated, and since the dataset primarily consisted of videos (which occupy more storage compared to images), it posed a challenge. After multiple communications with the university's IT team, they increased the storage capacity assigned to me, enabling the training of models on the remote server.

Another challenge arose while working with pre-trained models, as not all research papers provided comprehensive details on the model architecture and expected input data format. This made it challenging to retrain the models and accurately understand the required input data format. It required multiple attempts and extensive investigation to achieve successful training. However, debugging code on the remote server is complex due to limited accessibility.

In addition, it is often challenging to locate pre-trained models as they are not always readily available. Many research papers do not provide specific details or direct links to the pre-trained models they utilised. Luckily, when a paper employs a pre-trained model, they typically cite the source from which it was obtained. This citation serves as a helpful reference, facilitating the search for suitable pre-trained models that are accessible for download.

Despite these challenges, proactive measures were taken to overcome them and ensure the smooth progression of the experiments.

7 Ethics

The WLASL dataset [25] employed in this study involves secondary analysis of publicly available data. The dataset creators explicitly state that "all the WLASL data is intended for academic and computational use only" [25]. Since this report utilises the dataset for academic purposes and properly cites it, there is no violation of ethical consent.

8 Conclusion

In conclusion, this dissertation project aims to develop an effective American Sign Language recognition system that utilises video input data in a signer-independent mode. The primary objective is to alleviate communication challenges faced by individuals with speech and hearing impairments.

Two experiments have been conducted thus far using the WLASL dataset. The first experiment involved utilising a 3DCNN model and testing the most relevant frames to understand where the crucial information lies within the signing process. The second experiment incorporates a SlowFast neural network, which detects hand movements and performs facial recognition to assign greater importance to the relevant features during model training.

Looking ahead, several further steps will be taken to enhance the system. These steps involve implementing various approaches, including testing different models to identify the optimal performance and experimenting with different types of data augmentation techniques to improve the model's generalisation capabilities.

By continuously refining and expanding the system, we strive to develop a robust and inclusive solution that empowers individuals with speech and hearing impairments to communicate more effectively through American Sign Language recognition.

A Second Experiment

A.1 SlowFast Neural Network with ALL Data Processing

Figure 6 portrays the confusion matrix for the SlowFast Neural Network of the second experiment using the "ALL" data processing technique. Additionally, Figure 7 depicts the model's loss function during training. This particular model attained an accuracy level of 58.33%.

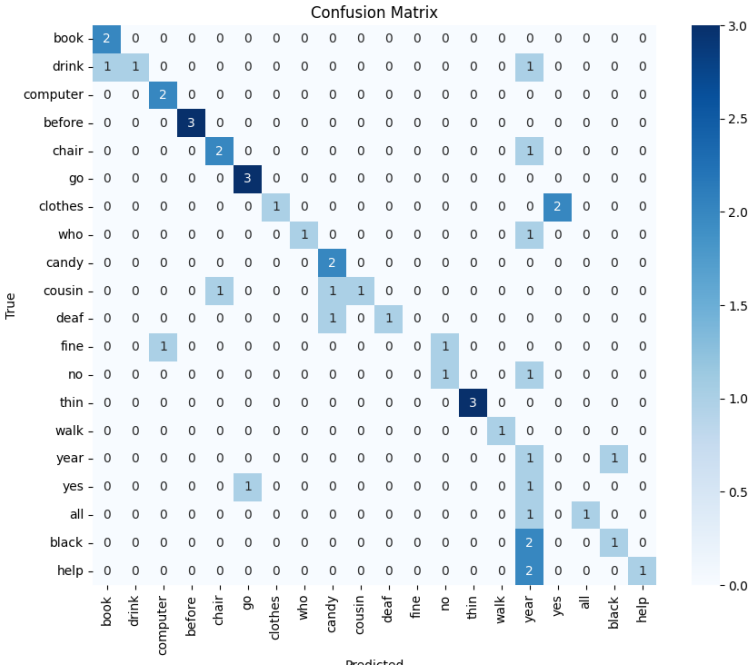

Figure 6: The confusion matrix for the second experiment's SlowFast Neural Network, employing the "ALL" data processing technique.

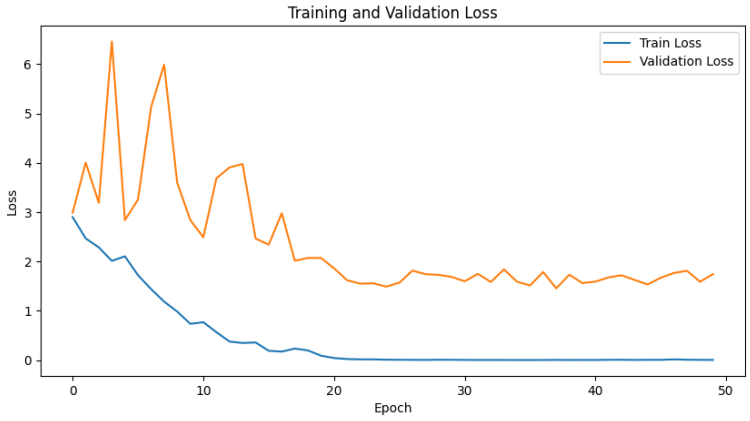

Figure 7: The loss function graph obtained from training the SlowFast Neural Network in the second experiment, employing the "ALL" data processing technique.

A.2 3DCNN with ALL Data Processing

Figure 8 shows the confusion matrix for the 3DCNN of the second experiment using the "ALL" data processing technique. Additionally, Figure 9 depicts the model's loss function during training. This particular model attained an accuracy level of 14.58%.

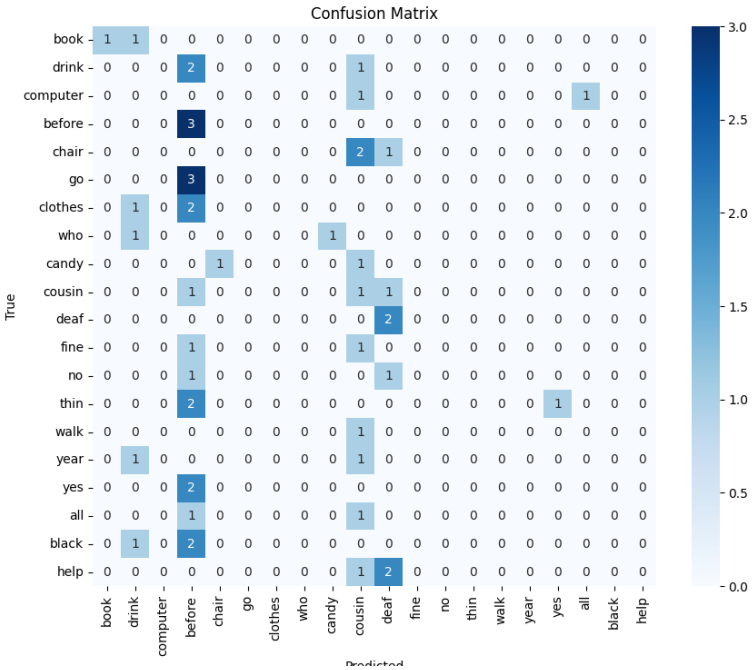

Figure 8: The confusion matrix for the second experiment's 3DCNN, employing the "ALL" data processing technique.

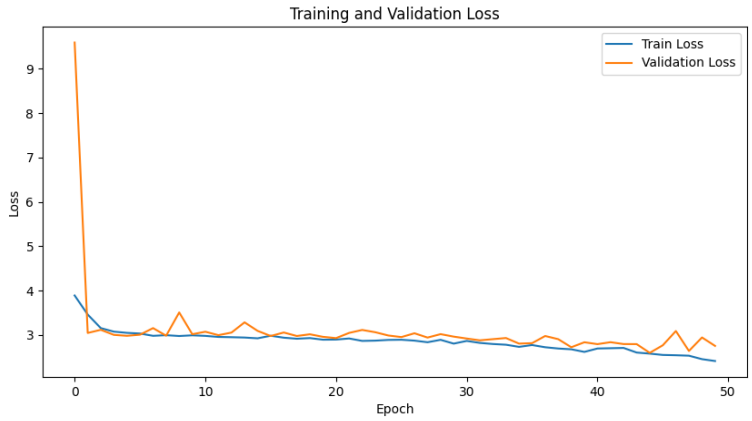

Figure 9: The loss function graph obtained from training the 3DCNN in the second experiment, employing the "ALL" data processing technique.

A.3 SlowFast Neural Network with BODY AND HANDS Data Processing

Figure 10 portrays the confusion matrix for the SlowFast Neural Network of the second experiment using the "BODY AND HANDS" data processing technique. Additionally, Figure 11 depicts the model's loss function during training. This particular model attained an accuracy level of 63.27%.

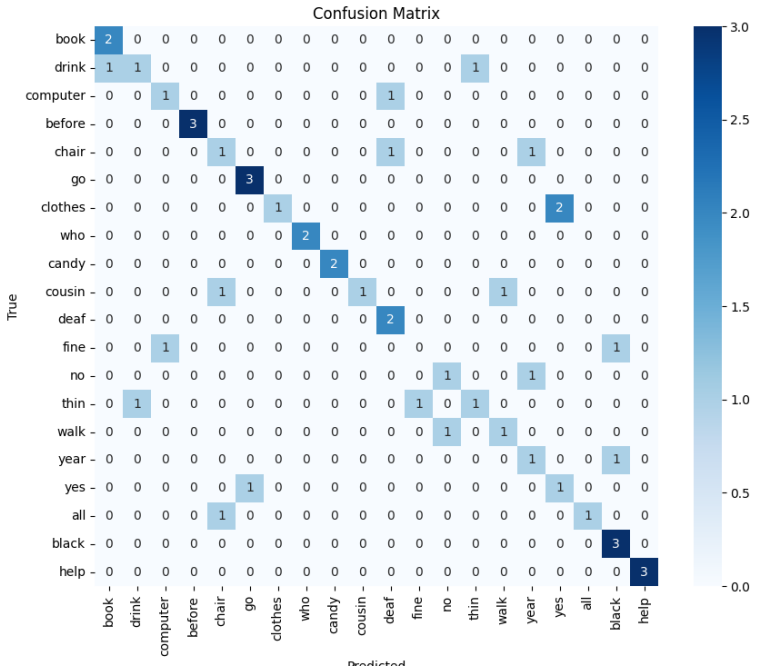

Figure 10: The confusion matrix for the second experiment's SlowFast Neural Network, employing the "BODY AND HANDS" data processing technique.

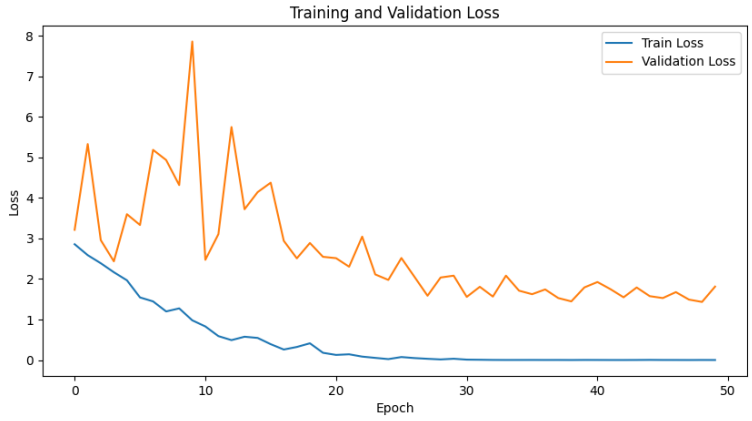

Figure 11: The loss function graph obtained from training the SlowFast Neural Network in the second experiment, employing the "BODY AND HANDS" data processing technique.

A.4 3DCNN with BODY AND HANDS Data Processing

Figure 12 shows the confusion matrix for the 3DCNN of the second experiment using the "BODY AND HANDS" data processing technique. Additionally, Figure 13 shows the loss function of the model during training. This particular model attained an accuracy level of 18.37%.

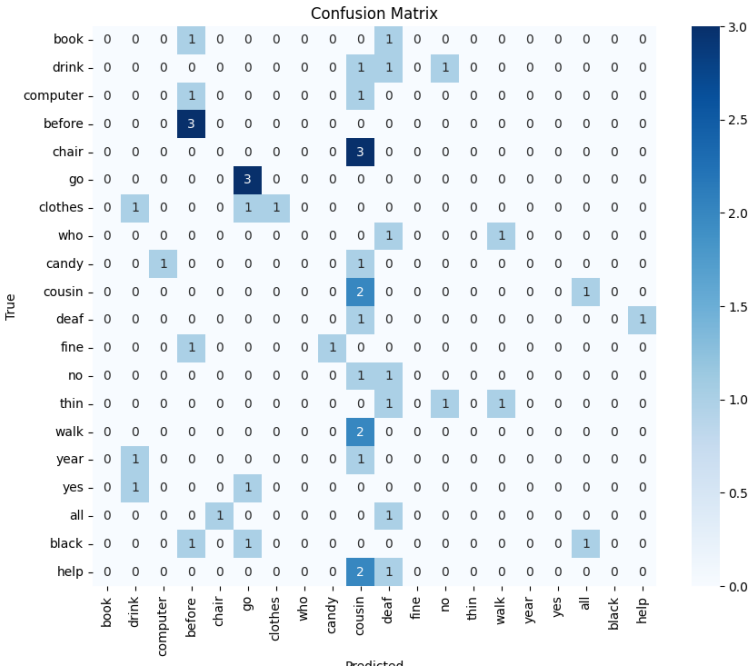

Figure 12: The confusion matrix for the second experiment's 3DCNN, employing the "BODY AND HANDS" data processing technique.

Figure 13: The loss function graph obtained from training the 3DCNN in the second experiment, employing the "BODY AND HANDS" data processing technique.

A.5 SlowFast Neural Network with FACE AND HANDS Data Processing

Figure 14 portrays the confusion matrix for the SlowFast Neural Network of the second experiment using the "FACE AND HANDS" data processing technique. Additionally, Figure 15 depicts the model's loss function during training. This particular model attained an accuracy level of 62.50%.

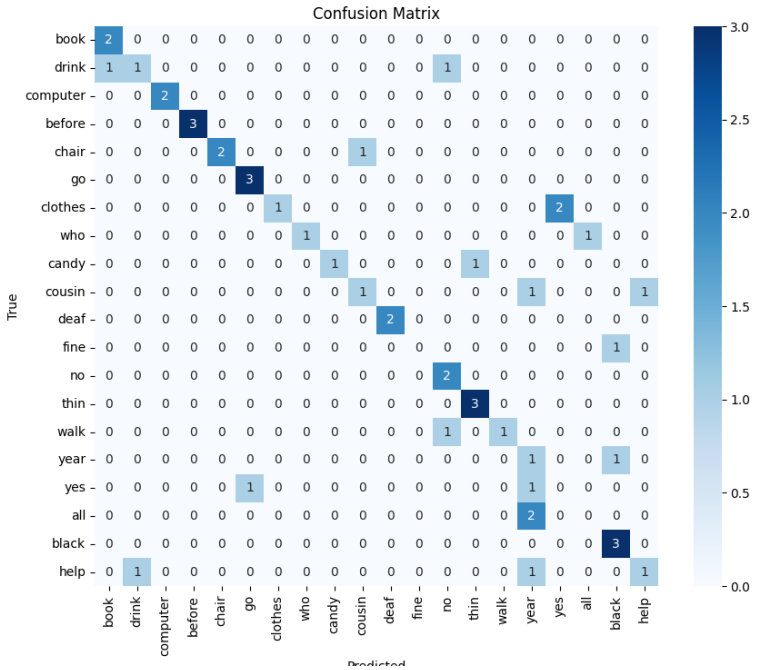

Figure 14: The confusion matrix for the second experiment's SlowFast Neural Network, employing the "FACE AND HANDS" data processing technique.

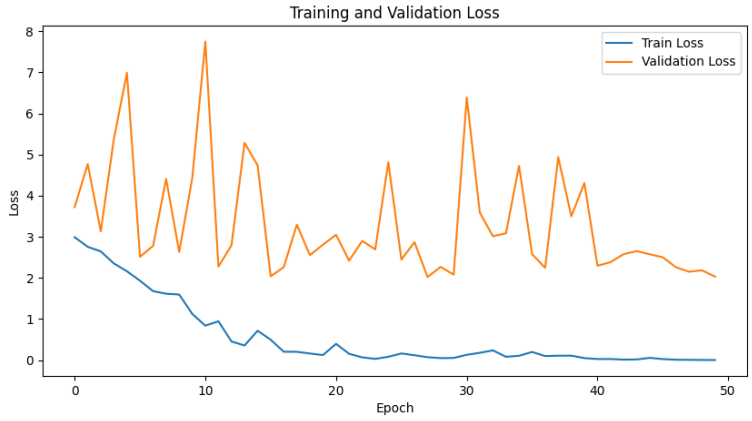

Figure 15: The loss function graph obtained from training the SlowFast Neural Network in the second experiment, employing the "FACE AND HANDS" data processing technique.

A.6 3DCNN with FACE AND HANDS Data Processing

Figure 16 shows the confusion matrix for the 3DCNN of the second experiment using the "FACE AND HANDS" data processing technique. Additionally, Figure 17 shows the loss function of the model during training. This particular model attained an accuracy level of 14.58%.

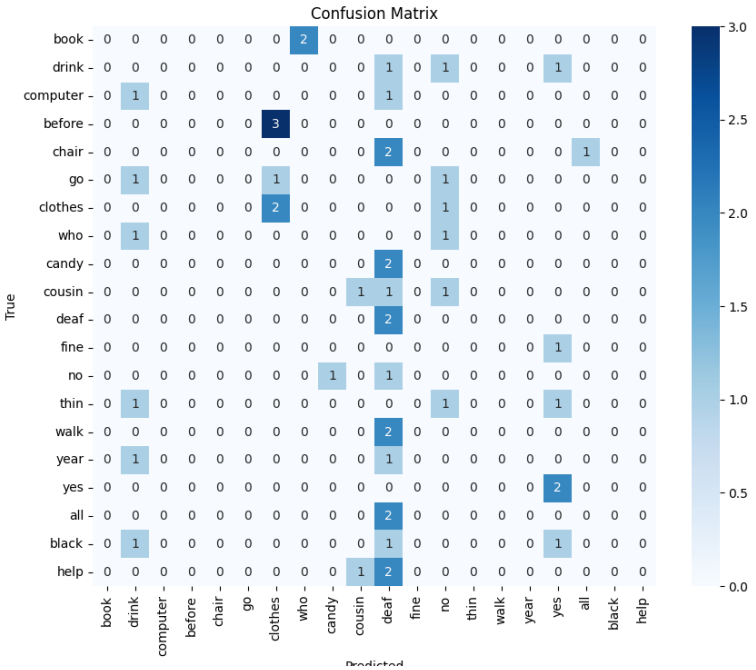

Figure 16: The confusion matrix for the second experiment's 3DCNN, employing the "FACE AND HANDS" data processing technique.

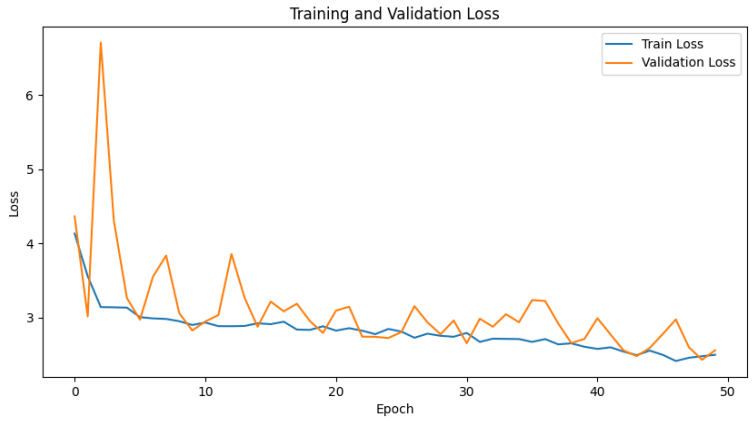

Figure 17: The loss function graph obtained from training the 3DCNN in the second experiment, employing the "FACE AND HANDS" data processing technique.

A.7 SlowFast Neural Network with HANDS Data Processing

Figure 18 portrays the confusion matrix for the SlowFast Neural Network of the second experiment using the "HANDS" data processing technique. Additionally, Figure 19 depicts the model's loss function during training. This particular model attained an accuracy level of 53.06%.

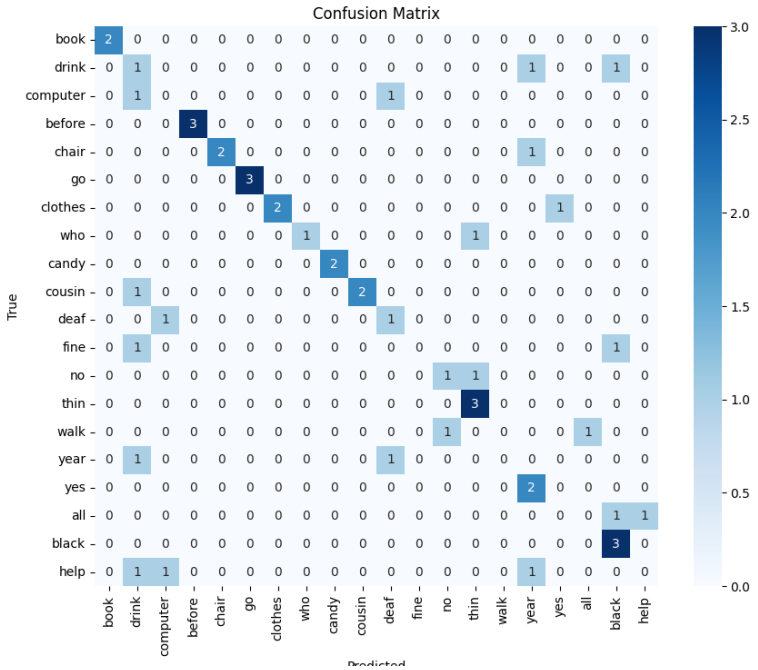

Figure 18: The confusion matrix for the second experiment's SlowFast Neural Network, employing the "HANDS" data processing technique.

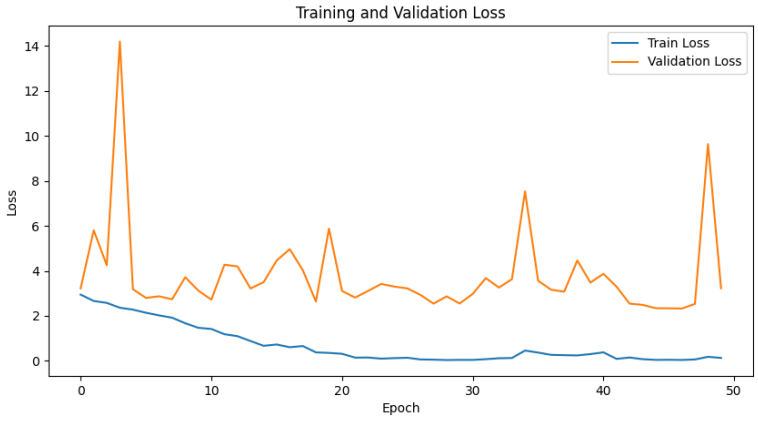

Figure 19: The loss function graph obtained from training the SlowFast Neural Network in the second experiment, employing the "HANDS" data processing technique.

A.8 3DCNN with HANDS Data Processing

Figure 20 shows the confusion matrix for the 3DCNN of the second experiment using the "HANDS" data processing technique. Additionally, Figure 21 shows the loss function of the model during training. This particular model attained an accuracy level of 26.53%.

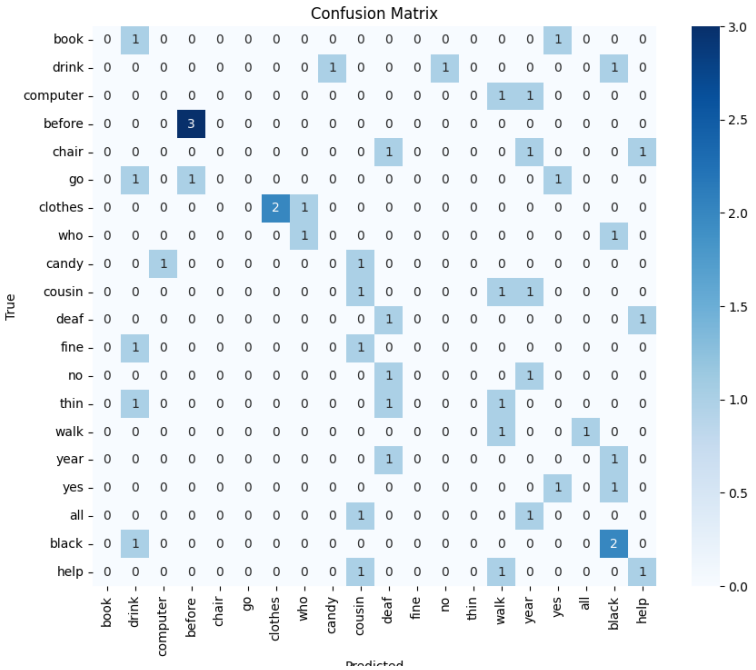

Figure 20: The confusion matrix for the second experiment's 3DCNN, employing the "HANDS" data processing technique.

Figure 21: The loss function graph obtained from training the 3DCNN in the second experiment, employing the "HANDS" data processing technique.

B Third Experiment

B.1 SlowFast Neural Network using the WLASL dataset for training

Figure 22 portrays the confusion matrix for the SlowFast Neural Network of the third experiment using the WLASL dataset for training and the MSASL dataset for testing. Additionally, Figure 23 depicts the model's loss function during training. This particular model attained an accuracy level of 44.44%.

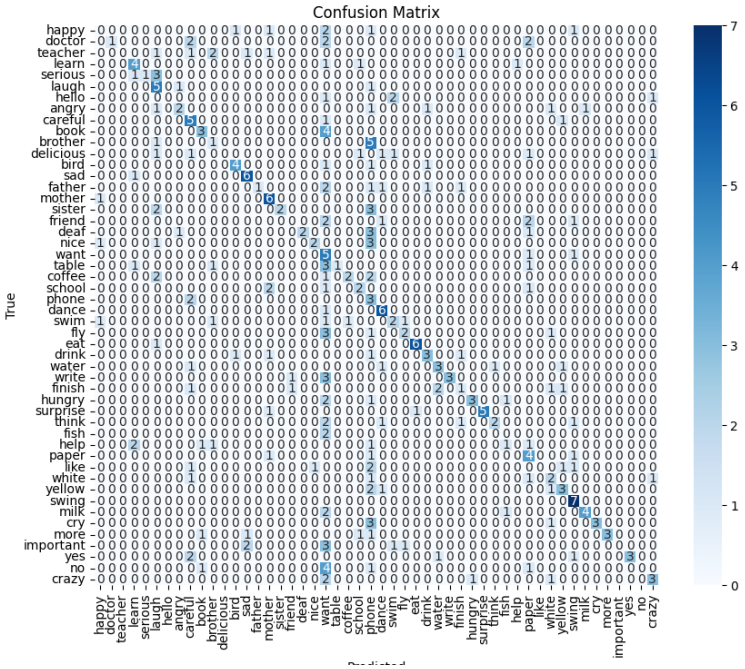

Figure 22: The confusion matrix for the third experiment's SlowFast Neural Network, using the WLASL dataset for training.

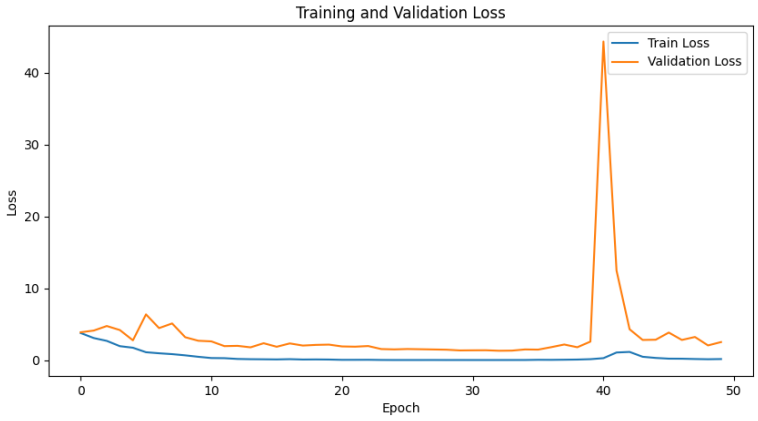

Figure 23: The loss function graph obtained from training the SlowFast Neural Network in the third experiment using the WLASL dataset.

B.2 3DCNN using the WLASL dataset for training

Figure 24 portrays the confusion matrix for the 3DCNN Neural Network of the third experiment using the WLASL dataset for training and the MSASL dataset for testing. Additionally, Figure 25 depicts the model's loss function during training. This particular model attained an accuracy level of 44.44%.

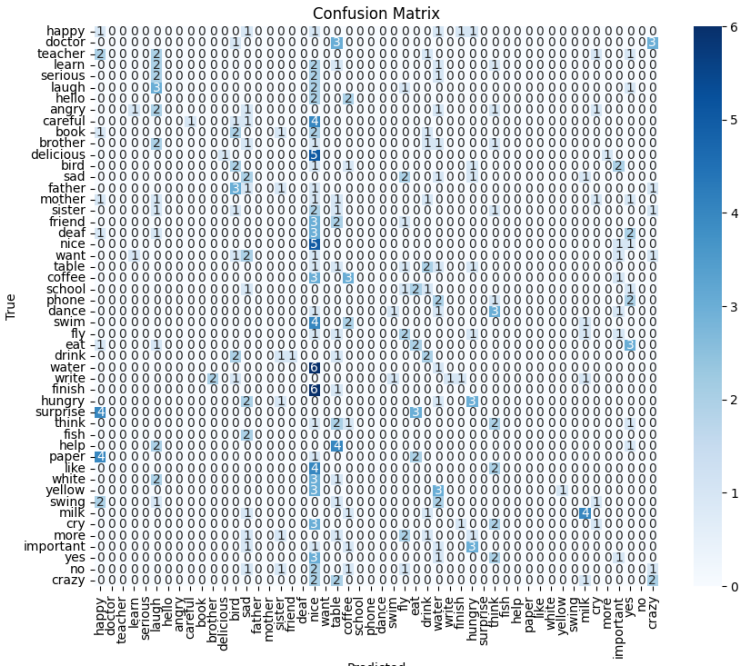

Figure 24: The confusion matrix for the third experiment's 3DCNN Neural Network, using the WLASL dataset for training.

Figure 25: The loss function graph obtained from training the 3DCNN Neural Network in the third experiment using the WLASL dataset.

B.3 SlowFast Neural Network using Mixed Precision and the WLASL dataset for training

Figure 26 portrays the confusion matrix for the SlowFast Neural Network of the third experiment using the WLASL dataset for training and the MSASL dataset for testing. The batch size was increased to 10, using the mixed precision technique. Additionally, Figure 27 depicts the model's loss function during training. This particular model achieved an accuracy level of 37.84%.

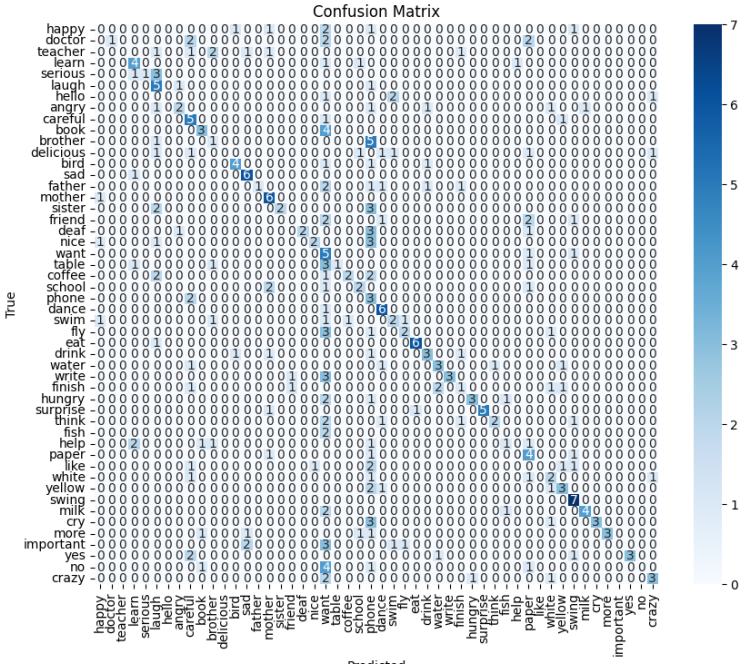

Figure 26: The confusion matrix for the third experiment's SlowFast Neural Network, using the WLASL dataset for training.

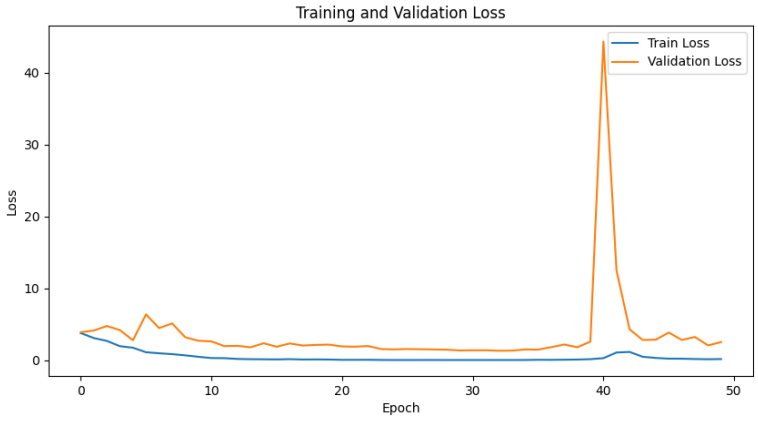

Figure 27: The loss function graph obtained from training the SlowFast Neural Network in the third experiment using the WLASL dataset.

B.4 3DCNN using Mixed Precision and the WLASL dataset for training

Figure 28 portrays the confusion matrix for the 3DCNN of the third experiment using the WLASL dataset for training and the MSASL dataset for testing. The batch size was increased to 10, using the mixed precision technique. Additionally, Figure 29 depicts the model's loss function during training. This particular model achieved an accuracy level of 12.01%.

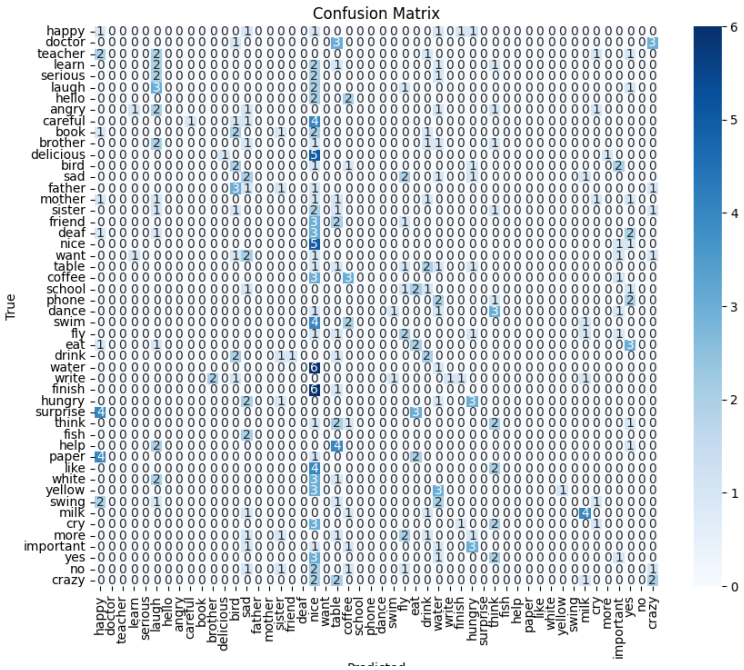

Figure 28: The confusion matrix for the third experiment's 3DCNN Neural Network, using the WLASL dataset for training.

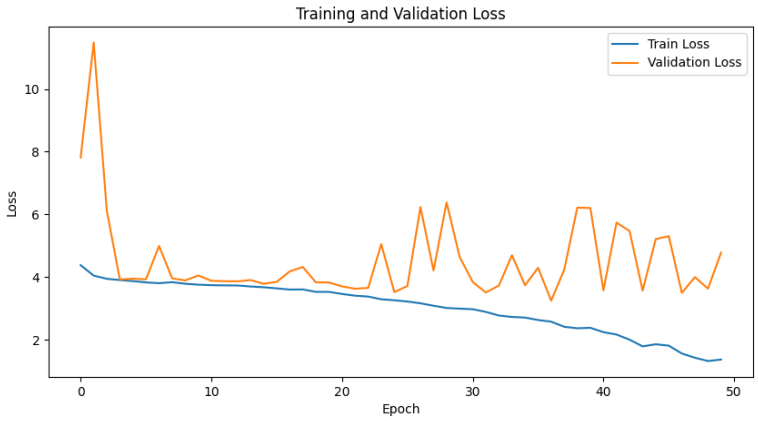

Figure 29: The loss function graph obtained from training the 3DCNN Neural Network in the third experiment using the WLASL dataset.

B.5 Comparison of the different training ans testing datasets

B.5.1 SlowFast Neural Network using the WLASL dataset for training

Figure 30 portrays the confusion matrix for the SlowFast Neural Network of the third experiment using the WLASL dataset for training. Additionally, Figure 27 depicts the model's loss function during training. This particular model achieved an accuracy level of 32.53%.

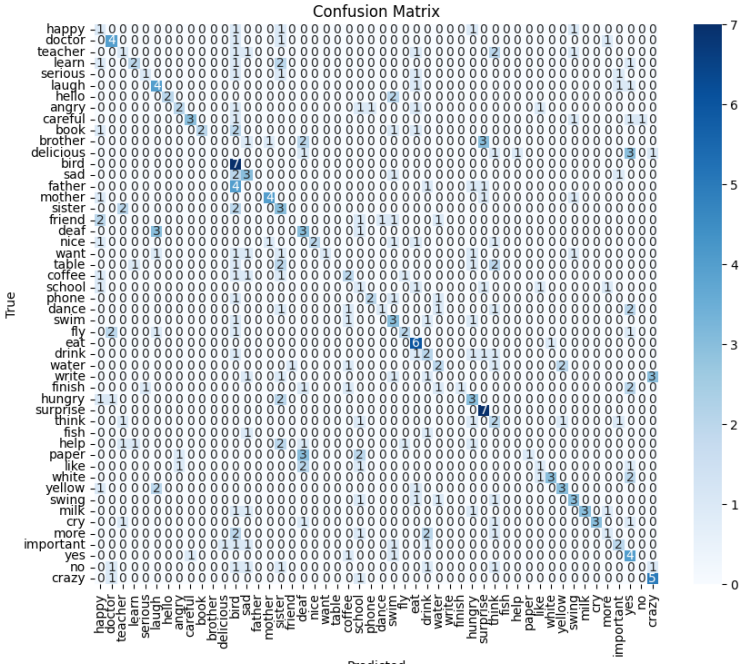

Figure 30: The confusion matrix for the third experiment's SlowFast Neural Network, using the WLASL dataset for training.

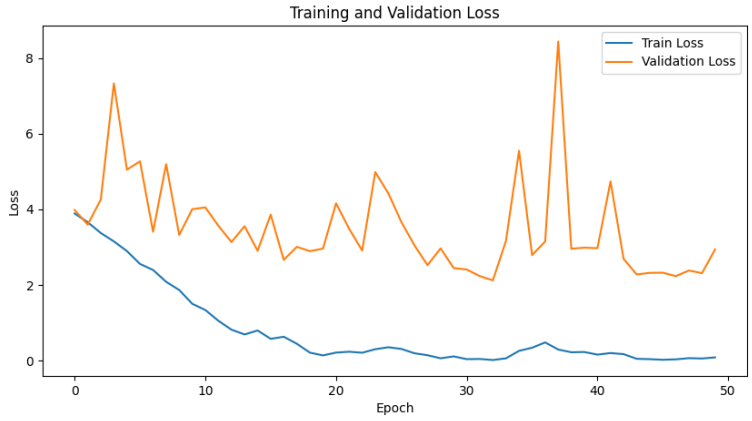

Figure 31: The loss function graph obtained from training the SlowFast Neural Network in the third experiment using the WLASL dataset.

B.5.2 Facebook 3DCNN model using the WLASL dataset for training

Figure 32 portrays the confusion matrix for the 3DCNN model from Facebook of the third experiment using the WLASL dataset for training. Additionally, Figure 29 depicts the model's loss function during training. This particular model achieved an accuracy level of 20.94%.

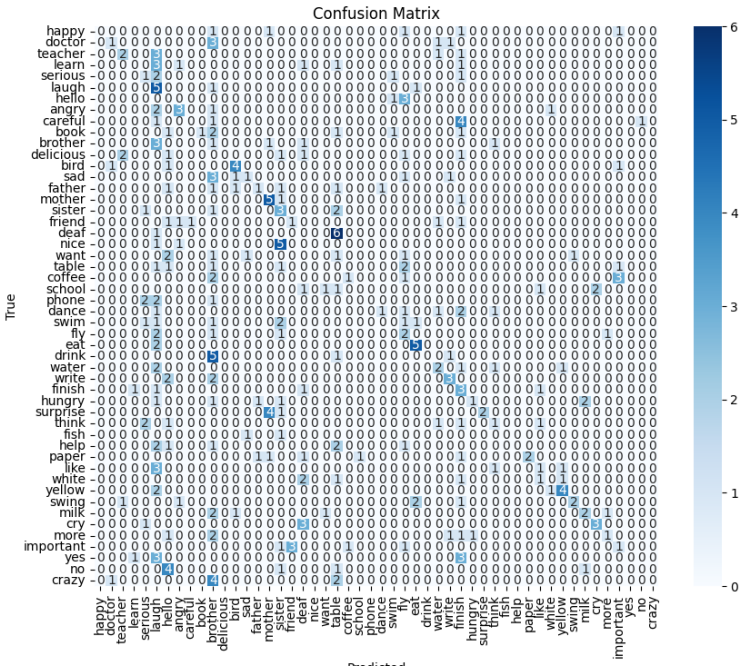

Figure 32: The confusion matrix for the third experiment's 3DCNN pre-trained by Facebook, using the WLASL dataset for training.

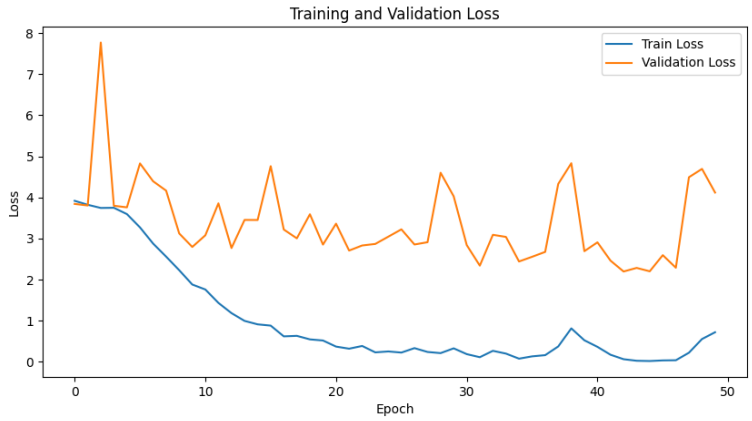

Figure 33: The loss function graph obtained from training the 3DCNN pre-trained by Facebook in the third experiment, using the WLASL dataset.

B.5.3 SlowFast Neural Network using the MSASL dataset for training

Figure 34 portrays the confusion matrix for the SlowFast Neural Network of the third experiment using the MSASL dataset for training. Additionally, Figure 35 depicts the model's loss function during training. This particular model achieved an accuracy level of 32.53%.

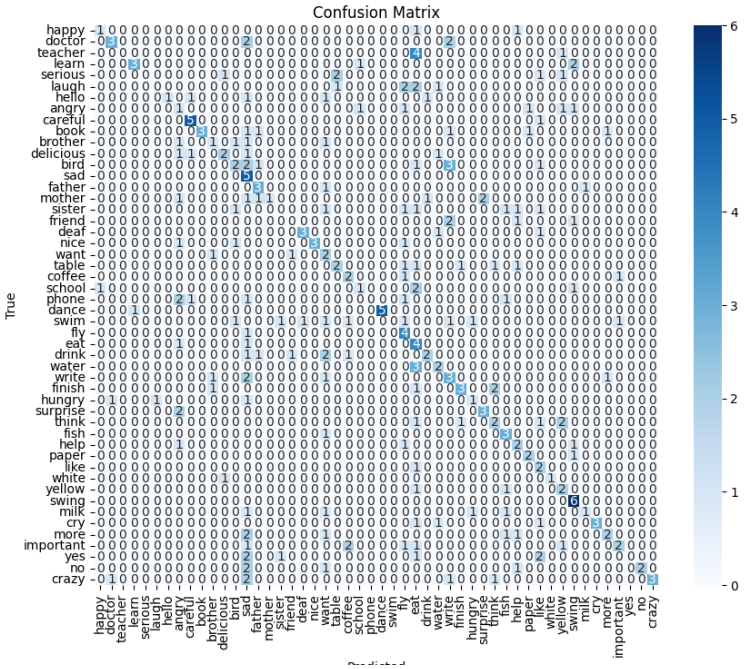

Figure 34: The confusion matrix for the third experiment's SlowFast Neural Network, using the MSASL dataset for training.

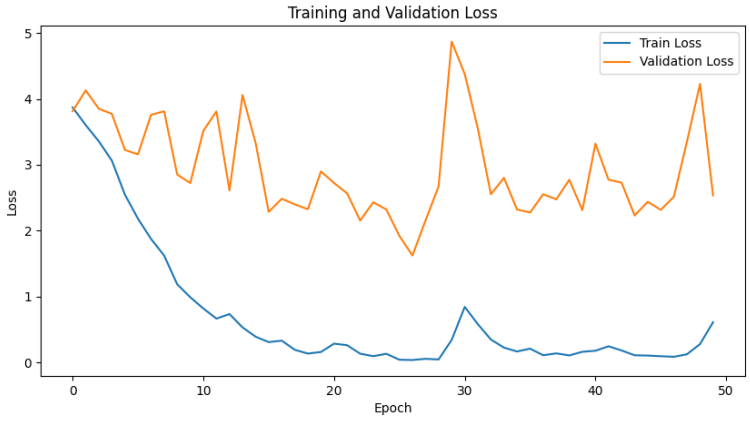

Figure 35: The loss function graph obtained from training the SlowFast Neural Network in the third experiment using the MSASL dataset.

B.5.4 Facebook 3DCNN model using the MSASL dataset for training

Figure 36 portrays the confusion matrix for the 3DCNN model from Facebook of the third experiment using the MSASL dataset for training. Additionally, Figure 37 depicts the model's loss function during training. This particular model achieved an accuracy level of 20.94%.

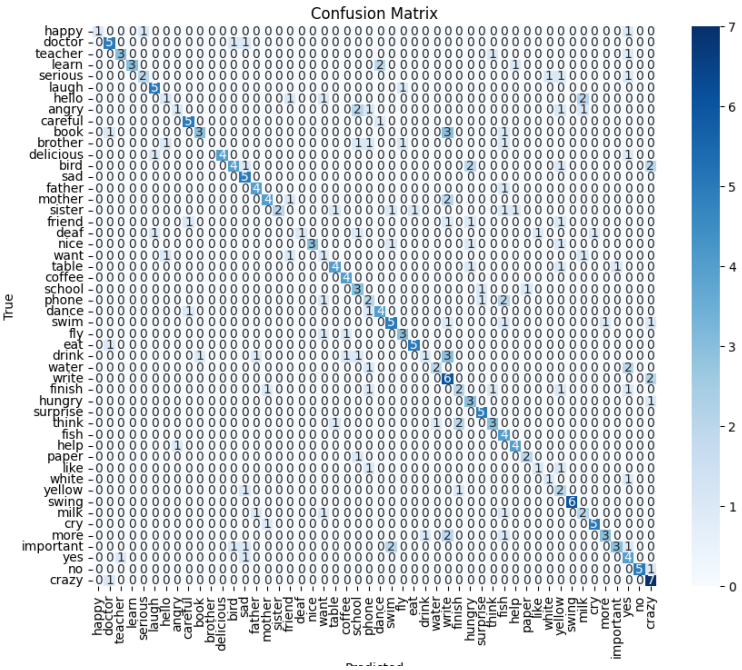

Figure 36: The confusion matrix for the third experiment's 3DCNN pre-trained by Facebook, using the MSASL dataset for training.

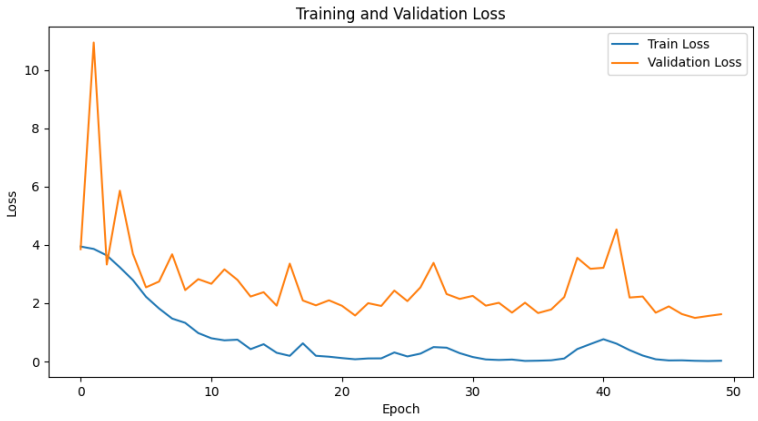

Figure 37: The loss function graph obtained from training the 3DCNN pre-trained by Facebook in the third experiment, using the MSASL dataset.

B.5.5 SlowFast Neural Network using both datasets for training

Figure 38 portrays the confusion matrix for the SlowFast Neural Network of the third experiment using both datasets for training. Additionally, Figure 39 depicts the model's loss function during training. This particular model achieved an accuracy level of 32.53%.

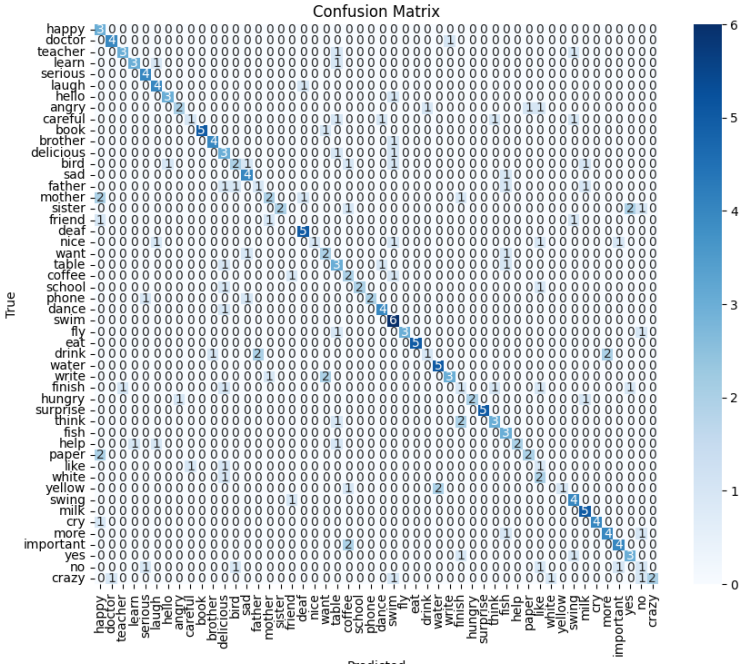

Figure 38: The confusion matrix for the third experiment's SlowFast Neural Network, using both datasets for training.

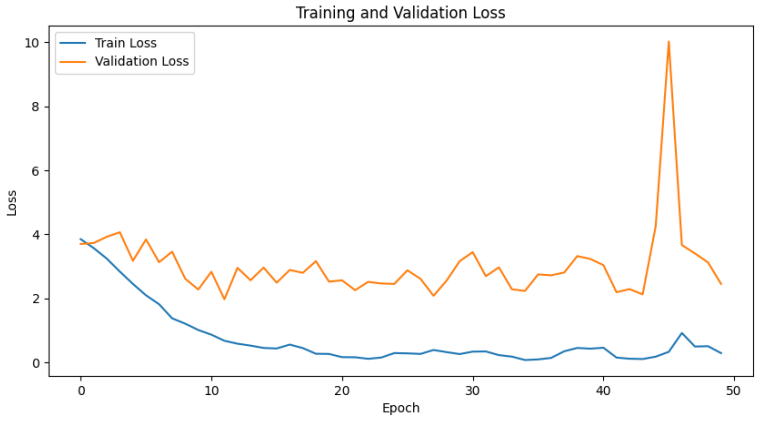

Figure 39: The loss function graph obtained from training the SlowFast Neural Network in the third experiment using both datasets.

B.5.6 Facebook 3DCNN model using both datasets for training

Figure 40 portrays the confusion matrix for the 3DCNN model from Facebook of the third experiment using both datasets for training. Additionally, Figure 41 depicts the model's loss function during training. This particular model achieved an accuracy level of 20.94%.

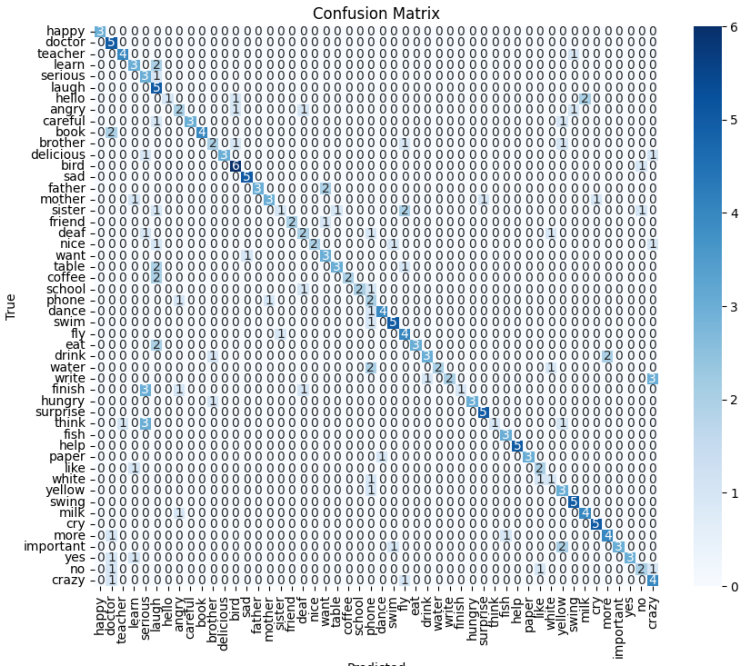

Figure 40: The confusion matrix for the third experiment's 3DCNN pre-trained by Facebook, using both datasets for training.

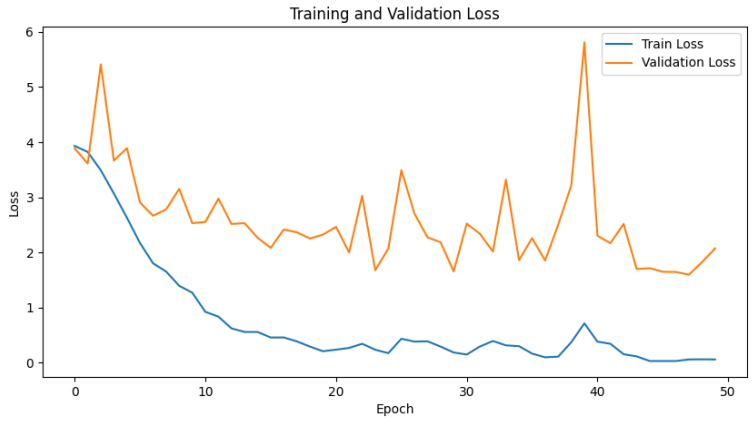

Figure 41: The loss function graph obtained from training the 3DCNN pre-trained by Facebook in the third experiment, using both datasets.

References

[1] Ankita Wadhawan and Parteek Kumar. Sign language recognition systems: A decade systematic literature review. *Archives of Computational Methods in Engineering*, 28:785 – 813, 2019.

[2] Alexey Karpov, Irina Kipyatkova, and Milos Zelezny. Automatic technologies for processing spoken sign languages. *Procedia Computer Science*, 81:201–207, 2016. SLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced languages 09-12 May 2016 Yogyakarta, Indonesia.

[3] Barbara R Schirmer. *Psychological, social, and educational dimensions of deafness.* Allyn & Bacon, 2001.

[4] Annalene Van Staden, Gerhard Badenhorst, and Elaine Ridge. The benefits of sign language for deaf learners with language challenges. *Per Linguam: a Journal of Language Learning= Per Linguam: Tydskrif vir Taalaanleer*, 25(1):44–60, 2009.

[5] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. *CoRR*, abs/2006.13256, 2020.

[6] Ahmed Hassan, Ahmed Elgabry, and Elsayed Hemayed. Enhanced dynamic sign language recognition using slowfast networks. In *2021 17th International Computer Engineering Conference (ICENCO)*, pages 124–128. IEEE, 2021.

[7] Cemil Oz and Ming C. Leu. Linguistic properties based on american sign language isolated word recognition with artificial neural networks using a sensory glove and motion tracker. *Neurocomputing*, 70(16):2891–2901, 2007. Neural Network Applications in Electrical Engineering Selected papers from the 3rd International Work-Conference on Artificial Neural Networks (IWANN 2005).

[8] Chao Sun, Tianzhu Zhang, Bingkun Bao, and Changsheng Xu. Latent support vector machine for sign language recognition with kinect. *2013 IEEE International Conference on Image Processing*, pages 4190–4194, 2013.

[9] Chao Sun, Tianzhu Zhang, Bing-Kun Bao, Changsheng Xu, and Tao Mei. Discriminative exemplar coding for sign language recognition with kinect. *IEEE Transactions on Cybernetics*, 43(5):1418–1428, 2013.

[10] Pat Jangyodsuk, Christopher Conly, and Vassilis Athitsos. Sign language recognition using dynamic time warping and hand shape distance based on histogram of oriented gradient features. In *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '14, New York, NY, USA, 2014. Association for Computing Machinery.

[11] Jian Wu, Zhongjun Tian, Lu Sun, Leonardo Estevez, and Roozbeh Jafari. Real-time american sign language recognition using wrist-worn motion and surface emg sensors. In *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 1–6, 2015.

[12] Panupon Usachokcharoen, Yoshikazu Washizawa, and Kitsuchart Pasupa. Sign language recognition with microsoft kinect's depth and colour sensors. *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 186–190, 2015.

[13] Celal Savur and Ferat Sahin. Real-time american sign language recognition system using surface emg signal. *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 497–502, 2015.

[14] Chao Sun, Tianzhu Zhang, and Changsheng Xu. Latent support vector machine modeling for sign language recognition with kinect. *ACM Trans. Intell. Syst. Technol.*, 6(2), mar 2015.

[15] Anup Kumar, Karun Thankachan, and Mevin M. Dominic. Sign language recognition. *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, pages 422–428, 2016.

[16] D. Anil Kumar, Polurie Venkata Vijay Kishore, A. S. Chandrasekhara Sastry, and P. Reddy Gurunatha Swamy. Selfie continuous sign language recognition using neural network. *2016 IEEE Annual India Conference (INDICON)*, pages 1–6, 2016.

[17] Celal Savur and Ferat Sahin. American sign language recognition system by using surface emg signal. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 002872–002877, 2016.

[18] Duaa AlQattan and Francisco Sepulveda. Towards sign language recognition using eeg-based motor imagery brain computer interface. *2017 5th International Winter Conference on Brain-Computer Interface (BCI)*, pages 5–8, 2017.

[19] M.K. Bhuyan, D. Ghosh, and P.K. Bora. Feature extraction from 2d gesture trajectory in dynamic hand gesture recognition. In *2006 IEEE Conference on Cybernetics and Intelligent Systems*, pages 1–6, 2006.

[20] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.

[21] Mark Borg and Kenneth P. Camilleri. Phonologically-meaningful subunits for deep learning-based sign language recognition. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision – ECCV 2020 Workshops*, pages 199–217, Cham, 2020. Springer International Publishing.

[22] 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.

[23] Jiangbin Zheng, Zheng Zhao, Min Chen, Jing Chen, Chong Wu, Yidong Chen, Xiaodong Shi, and Yiqi Tong. An improved sign language translation model with explainable adaptations for processing long sign sentences. In *Computational Intelligence and Neuroscience*, volume 2020, page 11, 2020.

[24] Muneer Al-Hammadi, Ghulam Muhammad, Wadood Abdul, Mansour Alsulaiman, Mohamed A. Bencherif, and Mohamed Amine Mekhtiche. Hand gesture recognition for sign language using 3dcnn. *IEEE Access*, 8:79491–79509, 2020.

[25] Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1459–1469, 2020.

[26] 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.

[27] Vassilis Athitsos, Carol Neidle, Stan Sclaroff, Joan Nash, Alexandra Stefan, Quan Yuan, and Ashwin Thangali. The american sign language lexicon video dataset. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, 2008.

[28] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. Pyslowfast. `https://github.com/facebookresearch/slowfast`, 2020.

[29] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[30] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018.

[31] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 510–526. Springer, 2016.

[32] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6047–6056, 2018.

[33] TorchVision maintainers and contributors. Torchvision: Pytorch's computer vision library. `https://github.com/pytorch/vision`, 2016.

[34] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[35] Haoqi Fan, Tullie Murrell, Heng Wang, Kalyan Vasudev Alwala, Yanghao Li, Yilei Li, Bo Xiong, Nikhila Ravi, Meng Li, Haichuan Yang, Jitendra Malik, Ross Girshick, Matt Feiszli, Aaron Adcock, Wan-Yen

Lo, and Christoph Feichtenhofer. PyTorchVideo: A deep learning library for video understanding. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2021. `https://pytorchvideo.org/`.

[36] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.