DEEP LEARNING BASED SIGN LANGUAGE PREDICTION

Author

RUQAIYA ALI

Regn No. 00000276700

Supervisor

Dr. HASAN SAJID

ROBOTICS AND INTELLIGENT MACHINE ENGINEERING

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

AUGUST 2022

DEEP LEARNING BASED SIGN LANGUAGE PREDICTION

Author

RUQAIYA ALI

Regn Number

00000276700

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Robotics and Intelligent Machine Engineering

Thesis Supervisor:

Dr. Hasan Sajid

Thesis Supervisor's Signature: ________________________________

ROBOTICS AND INTELLIGENT MACHINE ENGINEERING
SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY
ISLAMABAD
AUGUST 2022

FORM TH-4

Thesis Acceptance Ceritficate

Declaration

I certify that this research work titled “*Deep Learning Based Sign Language Prediction”* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources has been properly acknowledged / referred.

Signature of Student

RUQAIYA ALI

2018-NUST-MS-RIME-00000276700

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student
RUQAYA ALI
00000276700

Signature of Supervisor

Copyright Statement

Acknowledgements

I am thankful to my Creator Allah Subhana-Watala to have guided me throughout this work at every step and for every new thought which You set up in my mind to improve it. Indeed, I could have done nothing without Your priceless help and guidance. Whosoever helped me throughout the course of my thesis, whether my parents or any other individual was Your will, so indeed none be worthy of praise but You.

I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout every department of my life.

I would like to express special thanks to my supervisor “Dr. Hasan Sajid” for trusting in me, by providing me the opportunity to work with him. His guidance, support, and availability at every part of the research phase made me complete this work. I would like to thank him for guiding me in all the events other than my research work which has helped me in grooming my technical skills and being motivated. His efforts and dedication throughout my research work were the key element for the accomplishment.

I would also like to acknowledge my committee members Dr. Muhammad Jawad Khan and Dr. Karam Dad Kallu for their support and encouragement throughout my course work and research work.

I am very thankful to my family and friends for supporting and loving me throughout my life by the best they could do for me.

Dedicated to my exceptional parents and adored siblings for their endless love and tremendous support that led me to this wonderful accomplishment.

Abstract

Speech and hearing impairment is a condition that limits a person's capacity to communicate verbally and audibly. Individuals who are impacted by this adopt sign language and various alternative forms of communication. Even though sign language has become more widely used recently, it is still difficult for non-signers to engage with the individuals that use sign language.

There has been promising improvement in the disciplines of motion and gesture detection combining techniques of computer vision and deep learning. This study aims to put forward an approach that uses deep learning techniques to automate the recognition of American Sign Language, thereby lowering barriers to effective communication among the hard of hearing individuals and hearing communities. Previously, several techniques of deep learning were employed for sign language gesture recognition. Video sequences are used as an input for extraction of spatial and temporal information. Word-level sign language recognition (WSLR) technology advancements can drastically reduce the necessity for human translators and enable the signers and non-signers to easily communicate. The majority of methods currently in use rely on the use of extra equipment like sensor devices, gloves, or depth cameras. The ease of usage in real life situations is, however, constrained by these limitations. Such situations may benefit from deep learning techniques that are entirely vision-based and non-intrusive. American Sign Language has its own rules for syntax and grammar, much like any other spoken language. ASL, like every other language, is a living language that evolves and develops through time. The majority of ASL users are found in both Canada and the United States of America. In order to complete their current and "international" degree requirements, most schools and institutions across the US accept ASL. This study uses deep learning methods to predict American Sign Language using the WLASL (word-level American Sign Language) dataset. For the dataset, a subset of 50 classes was chosen from WLASL. This study used a combination of VGG16-LSTM and ConvLSTM based to work with spatio-temporal features. These models were chosen due to their ability to work with spatial and temporal features. We observed that VGG16-LSTM outperformed the ConvLSTM architecture. Both models' performances are examined using accuracy as an evaluation metric and judged according to how well they perform on test videos.

Key Words: *Sign Language Recognition, Deep Learning, Neural Network, Convolutional Neural Network, Long Short-term Memory, VGG16*

Table of Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

The deaf people around the world use sign language to communicate. Sign language includes a variety of hand postures and motions that are coordinated with a predetermined vocabulary and lexicon, in addition to the use of face expressions and body language. Although sign language is widely used by the deaf, the hearing community does not comprehend it, creating a communication gap between those who do and those who do not. Automatic systems for recognizing sign language have been suggested as a solution to this problem, and research into these systems is ongoing[1]. The categorization of isolated signs and continuous signing can be used to classify the research on automatic sign language recognition[2, 3]. Several strategies have been proposed for interpreting sign language from videos involving computer vision techniques. There are three different methods for recognizing signs in sign language: (1) Character-level Sign Language Recognition, isolated sign language recognition (word-level), and continuous sign language recognition (sentence-level). In character-level sign language, there are 36 signs total, including 26 signs for the English alphabet and ten numerals (0–9). Even though character-level sign language recognition has been the subject of countless studies[4, 5], it is time-consuming to write down every sign language word, hence it is not commonly employed. A simpler way would be to employ Word-level or Sentence-level Sign Language Recognition. These methods do have some drawbacks, some of which are stated below:

1. It is difficult to create a system that can capture the features in all the signs because the vocabulary of sign language that is used on a daily basis is rather large (usually in the thousands).
2. While having a large vocabulary, some words, such as names of persons, might not be included; in these situations, it might be essential to indicate those words with character-level signs.
3. The combination of body, hand, and head movement plays a major role in the recognition of signs.
4. Two signs could differ by just a little bit, and if they aren't properly identified, they could result in the wrong classification[6].

More than 135 sign languages exist in different parts of the world and American Sign Language is one of them. The most common sign language in the developed applications for Sign Language Recognition is American Sign Language (ASL), which is practiced by the biggest sign language community in the world and is also the subject of this review. There are regions of West Africa and Southeast Asia where American Sign Language is used. ASL is used as a first language by about 500,000 individuals in the United States [7], and since English and ASL have major linguistic distinctions, it is acceptable to be proficient in one but not the other. Most deaf individuals in the U.S. have lower levels of English literacy, which experts have linked to a number of academic characteristics and early exposure to language [8, 9].

American Sign Language, Portuguese Sign Language, Indian Sign Language, and many others are among the sign languages used in the majority of nations. The semantic characteristics of spoken languages are identical to those of ASL. Face and hand gestures are the primary means of communication in ASL. It may be necessary for a non-signer to learn the related sign language before engaging in conversation with someone from the deaf community, which takes time and effort. The use of a translator who is conversant in the relevant sign language would be a solution to this, but it can be costly and intrusive[6]. The primary building block for comprehending sign language phrases, word-level sign identification is also exceedingly difficult because the meaning of signs mostly depends on the mixture of body movements, manual gestures, and head postures, and small deviations may translate into diverse meanings. For instance, the only difference between the signs for "dance" and "read" is how the hands are positioned. A significant number—often thousands—of signs are used on a daily basis. Comparatively, only a small number of categories are present in related tasks like gesture recognition and action recognition. This poses a serious problem for the scalability of recognition techniques. Natural language counterparts for a sign language term could have several variations. For instance, based on the circumstances, the sign "want" may be perceived as "hungry." Furthermore , words that share a lemma with a noun or verb may share the same sign. These nuances are not adequately reflected in existing small-scale databases[10].

Using the most recent advances in computer vision, it would be effective to implement a system that can convert sign language seen in video into English words.

Given that ASL is widely used by deaf communities in more than 40 countries worldwide, we focus on word-level recognition tasks for ASL. In this study, we use ConvLSTM and VGG16-LSTM to perform word-level sign language recognition for American Sign Language.

1.1 Problem Statement

430 million individuals, which is more than 5 percent of the earth's total population, suffer from hearing impairment. According to WHO (Word Health Organization) More than seven hundred

million individuals, or 1 out of 10 persons, are predicted to have hearing problems by 2050. Considering sign language is difficult to learn and time consuming, it creates a communication gap between the hearing individuals and the hard of hearing. To interact with deaf persons, sign language must constantly be translated into natural speech by an interpreter. To identify, recognize, and translate gestures into meaningful terms or sentences, an automated SLR technique is required. This will greatly benefit the hard of hearing community and will also help in bridging the communication gap.

1.2 Objective

The main purpose of this study is to provide methods and to create an automated system that can recognize gestures from American Sign Language videos and in doing so contribute towards the individuals that suffer from hearing loss and promote their social inclusion.

1.3 Areas of Application

According to a review on Sign Language Recognition (SLR), American Sign Language (ASL), as the largest sign language community in the World, is the most-used sign language in the developed applications for SLR.

Major areas of the application for sign language recognition systems are the following:

- Translation Service for Medical Appointments
- Translation Service for Educational purpose
- Translation Service for on-the-job training
- Important Events (conferences and meetings)

1.4 Contributions

The following is a list of contributions made to this thesis.

- To categorise the videos, a ConvLSTM network and VGG-LSTM was proposed.
- To get the optimum model for ASL sign predictions, hyperparameters were adjusted.
- In order to assess performance for accuracy of prediction, a set of test videos were used for classification of different gestures.

1.5 Thesis Overview

Section 2 of this work describes the previous work conducted by multiple researchers on the study of various methodologies used for Sign Language Recognition in several languages, including American Sign Language.

The entire technique and implementation, including the dataset, data pre-processing, and full workflow, are contained in Section 3. Section 4 contains the results that were achieved using the proposed techniques for recognition of Sign language. Section 5 includes the discussion of the complete work. Section 6 describes the potential future work that could be conducted in this field.

CHAPTER 2: Literature Review

Parallel to the advancement of the neural network during the past few years, there have been numerous effective methods for recognizing isolated actions[11]. Researchers employed fundamental machine learning algorithms for action categorization in the previous research on human action recognition, such as Random forest [12], CRF [13], Naive Bayes [14], KNN [15], SVM [16], HMM [17] and Decision Tree [18], and used traditional hand-crafted features. Recently, they switched to end-to-end deep learning techniques that can automatically generate features using their neural networks rather than requiring a data scientist to correctly identify them. Deep learning-based approaches extract useful abstract features from sensor inputs or a collection of images, in contrast to Machine Learning methods that rely on hand-crafted features for training. The entire gesture identification process for both methods involves pre-processing, characterization, gesture acquisition, and gesture identification, with the latter being the most important stage [19].

Most techniques for identifying hand movements can be generally categorized as being based on measured values by sensing gloves and being vision-based. Whereas glove-based approach relies on external gear for recognition of gestures, the vision-based technique interacts with both humans and computers to recognize gestures [20]. Recently, major advancements in this domain have been made [21-26]. A technique for supervised modification of self-organizing maps called ProbSom was used by Ronchetti et al. [23] to classify the shape of hands after extracting descriptors from images. With this method, they were able to translate Argentinean Sign Language with an accuracy rate of more than 90%.

A technique relying on eigenvectors was provided by Joyeeta and Karen [25]. Pre-preprocessing phase involved skin filtering and histogram comparison. They employed an eigenvalue-weighted Euclidean distance-based classification method. They managed to identify 24 unique Indian Sign Language alphabets with a 96 percent accuracy rate. A technique to identify motions in Italian sign language was suggested by Lionel et al. [22]. Convolutional neural network (CNN) powered by a graphic processing unit (GPU) and Microsoft Kinect were employed (GPU). dataset comprising 20 classes of Italian gestures and were able to cross-validate with an accuracy of about 92 percent. A precisely calibrated portable gadget was suggested by Rajat et al. [24] as a remedy for the issue of reducing the communication barrier between those with normal abilities and those with disabilities. Three embedded algorithms that intended for quick, simple, and effective communication were used to explain the device's architecture and functioning.

Convolutional neural networks have been used in another study by S. Masood et al. [5] to recognize characters in American Sign Language. On a collection of 2524 ASL actions, the Convolutional neural model in this study managed to obtain 96% accuracy rating.

Additionally, the imaginative works of people such as W. Vicars [27] have aided in improving comprehension in the discipline of recognizing American Sign Language.

The concept of distinguishing sign language from video involves following steps: extraction of features, spatiotemporal localization of patterns, and categorization [6]. Previous scholars have experimented with various feature extraction techniques for the recognition of sign language, including classification of hand-crafted features [28, 29], appearance of image-based recognition using CNN [30, 31], classification using body parts [32-34], and recognition that relies on facial features of a person [35, 36].

Rather than obtaining the spatial and temporal information from a videotape individually, studies have attempted mapping them together using 3-D CNN models [9, 37].

The CNN component of the image appearance-based model is utilized to extract spatial features for classification from the input images and then passes the recovered flattened features into fully connected layers [38, 39]. As time went on, the Convolution layers began retrieving more complicated data from visuals, such as temporal and spatial characteristics from a series of pictures, or recordings [40, 41].

In order to extract the pose information or regions of interest in the frames of video using a deep CNN [42] and mapping the temporal information among frames applying an RNN model [43], or to apply non-maximal suppression method on the estimation of heatmap based pose information or key points, there are two different types of pose based recognition models.

The two methods for pose-based recognition include using a deep neural network to extract regions of interest from video frames and then using an RNN to map those characteristics throughout the frames. It can also be accomplished by using a non-maximum suppression strategy to the estimation of key points and pose information based on heatmaps.

The RWTH PHOENIX Weather 2014 dataset was used by Cui et al. [44] in combination with Recurrent CNN-based extraction of features, Bi-LSTM, and Detection Net. The conclusion made by the contributors was that their method acquired distributed portrayal between different people who signed and adequately controlled inter-signer variances. Later, Seq2Seq attention-based models were used by Camgoz et al. to enhance their research [45]. They adopted a tactic in which they approached sign language as a distinct language and thought about language translation as a potential remedy. The researchers found that their system could translate more precisely than the state-of-the-art, but it had one flaw: their model was unable to read or catch crucial data like dates, numerals, and locations [10].

The fingerspelling alphabet for ASL was implemented by Gracia and Alarcon [46]. GoogleNet was used in their study, which was built using the ASL datasets from Surrey University and Massey University as well as the ILSVRC2012 sample. While the ILSVRC uses 1000 different items or classes to create their data, t hey used hands to create their data in 24 distinct orientations, To regulate the input dimensionality of the data of the Google Neural Network model, they downsized 256x256 and removed arbitrary cropping of 224x224. The input data was further normalized and fed into the system for classification. They achieved 100% a-e Top-5 Val-accuracy.

To detect American Sign Language, Rim Barioul et al. [47] employed four commercial FMG sensors. The sensors serve as the bracelet for an FSR sensor. The study examined how well the Extreme Learning Machine and raw FMG recognized nine ASL alphabets (ELM). With the use of five-fold cross-validation and ELM training, an accuracy of 89.65% was attained, compared to a raw FMG accuracy of 69.69%.

From videos, Al Amin Hosain et al. [48] recognized American Sign Language (ASL) motions. From 2D skeleton data trajectories calculated from films, they suggested a pattern Recursive Neural Network (RNN) gesture detection model. By adding hands from another hand form recognition model, the model was expanded. For skeleton and hand picture data, the final model combines two LSTM RNN models. The GMU-ASL51 dataset, which consisted of 12 users and 51 ASL gestures, was used to train the model. Fusion LSTMs provided an average accuracy of 89%. This 2D skeleton-based model is superior to 3D models and sensor-based models, according to the research.

Using deep learning to recognize signs Passi and Goswami [49] created the CNN model, in which they provide approaches for identifying hand movements, color of skin, and hand shapes. For alphabets with different angle, light levels, and backgrounds with clutter, the model reached 99%

accuracy. Because there were many training images and high-quality images were used, a high accuracy was attained.

This research and WLASL Recognition by [10] Dongxu Li et al. are correlated. They put up a brand-new dataset for ASL recognition that comprises of 2000 classes performed by 119 signers in RGB format. The collection is gathered from publicly accessible videos on YouTube and other websites. They trained I3D, Pose-GRU, Pose-TGCN, VGG-GRU, and others on the new dataset, and they came to the conclusion that I3D outperforms all others with the greatest Top-10 accuracy of 89.92% across 100 classes. By selecting 50 sequential frames from each video and cutting them to 224x224 size, they performed preprocessing on the video dataset.

The majority of word-level sign language recognition models now in use are trained and assessed on either limited size datasets less than 100 words [50-57] or private [50, 51, 53, 57]dataset.

Unfortunately, only small-scale datasets are used to test these approaches. Therefore, it is unknown how well those procedures generalize. Furthermore, the results of various approaches tested on numerous small datasets is incomparable and might not accurately reflect the true effectiveness of models because there isn't a standard big scale dataset for word level sign language [10].

CHAPTER 3: Methodology and Implementation

3.1 Proposed Scheme

In the suggested scheme, we begin by downloading the WLASL dataset. The ASL videos is converted to frames first. For our VGG16-LSTM model, Data Augmentation technique is applied to expand the size of our training set which is discussed in detail in this Chapter's section 3.3. The augmented data is then fed into VGG16 (Pre-trained model) for spatial features extraction which is further forwarded to LSTM for extraction of temporal features. The output of LSTM is carried into fully connected layers using softmax as a classifier.

For our second method, the sequence of frames goes as an input to ConvLSTM model for feature extraction and classification. We used several unseen ASL videos from online sources to test the model's ability to recognize gestures and obtain the final results. At the conclusion, test data are used to evaluate each model, and performance is checked using average prediction accuracy on the test videos for different classes.

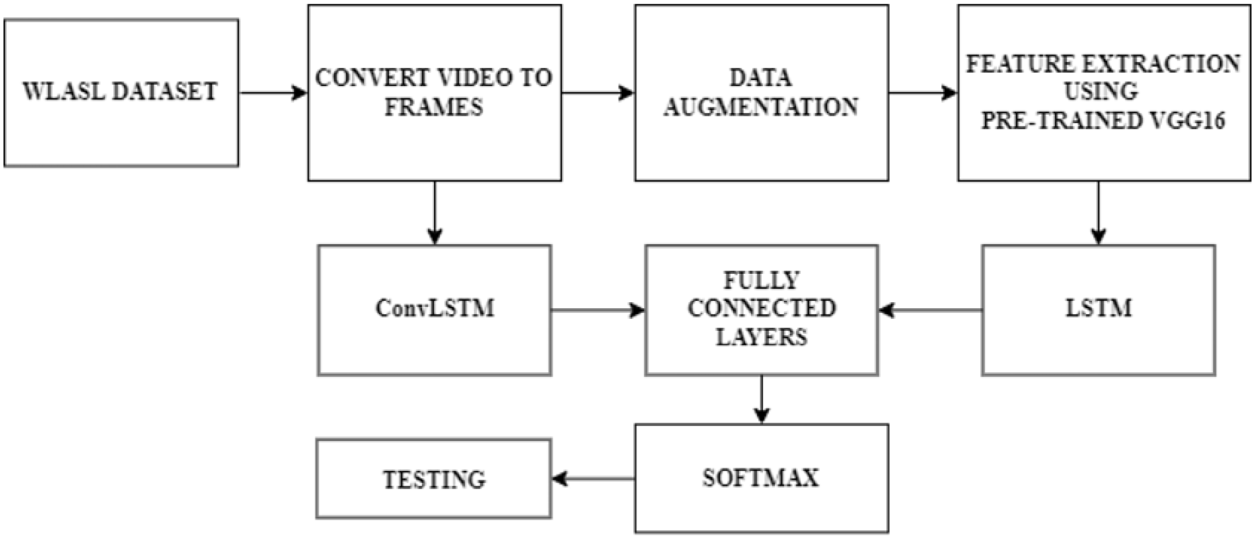


Figure 1 Complete Process Diagram

3.2 Dataset

Detection of sign language from videos can be challenging due to scarcity of large-scale annotated datasets for word-level sign language recognition. The majority of word-level sign language recognition methods work with datasets that don't accurately reflect real-world environments since there isn't much change in the conditions such as background, signers, inter signer variation and lighting [58]. We used the WLASL dataset since it does a great job in bypassing these constraints. It features a variety of signers, lighting, signers' distances from the camera, and backgrounds.

Some publicly available word-based ASL datasets are, ASL-LVD [59], MSASL dataset [60], the Purdue RVL-SLLL ASL dataset [61], and WLASL dataset [10] . We selected the WLASL dataset because it has a variety of classes and number of samples for each gesture compared to the other datasets.

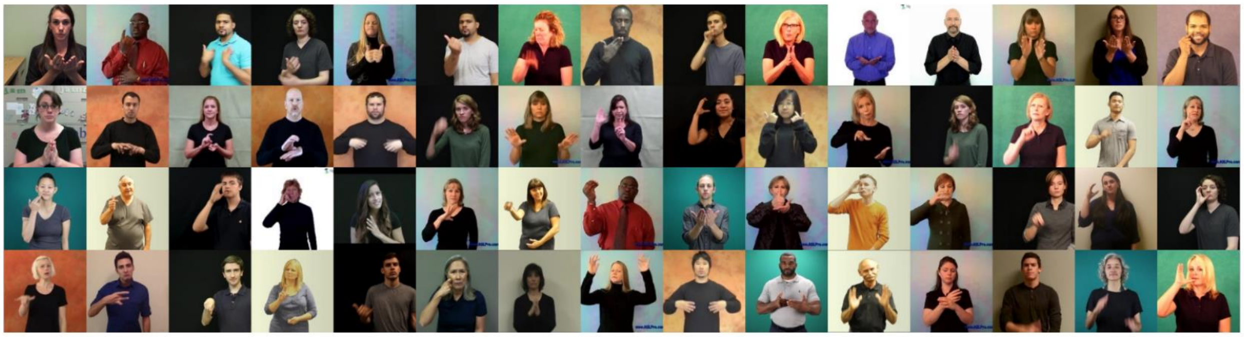


Figure 2 Illustration of the diversity of WLASL Dataset

3.3 Acquiring and Preparation of Dataset

Using the Dongxu.li author's recommended method, the WLASL Dataset was downloaded. All the video files are contained in a single folder without labels, and the downloaded material has already been pre-processed to convert them all to mp4 format. Each video file's information is contained in a JSON file that is part of the library. Each video file's class and video id label are provided under the tag and label, respectively, "gloss" and "video id," respectively. Python's OS libraries and JSON are used to gather the data. To read and create folders using loops, the operating system library is utilised. For each class, a single folder is made and loaded with numerous videos from that class.

The 50 classes used in this study are listed in the list below:

‘again’, ’birthday’, ’book’, ’buy’, ’bye’, ’can’, ’care’, ’coffee’, ’confused’, ’cook’, ’eat’, ’excited’, ’fine’, ’food’, ’go’, ’good’, ‘goodbye’, ’happy’, ’hello’, ’how’, ’later’, ’learn’, ’maybe’, ’me’, ’meet’, ’morning’, ’my’, ’name’, ‘nice’ , ‘night’, ‘no’, ‘please’, ‘pretty’, ‘sad’, ‘see’, ‘sign’, ‘slow’, ‘smile’, ‘sorry’, ‘stationary’, ‘take’, ‘text’, ‘thankyou’, ‘to’, ‘understand’, ‘want’, ‘what’, ‘yes’, ‘you’, ‘your’

3.4 Data Pre-Processing

First, the dataset was checked for missing values. We made the following conclusions while exploring the video examples in our dataset:

1. While some signers took their time to complete a sign, others did so quickly.
2. Several videos contained overlapping and signs from different classes as well as signs from the same class.
3. Longer video snippets contained beginning and ending frames without any sign being performed. We eliminated frames from classes that weren't the targets, and empty frames.

Pre-processing steps involves converting the video to a series of RGB frames. The dimensions for each frame are same. The frames were normalized and resized to fixed width and height.

3.5 Regularization

Any sophisticated network with a huge number of learning parameters, such as convolutional neural networks, is quite complex. Consequently, the likelihood of overfitting in such networks is a challenging problem. Overfitting is a critical issue since it makes it difficult to achieve a higher functioning model in the necessary amount of time for Convolutional Neural Networks. Thus, overfitting during the training of a neural network can happen rather readily and depends on a number of variables. A technique called Regularization allows us to modify the training and learning process for the model to train itself in a more global manner, which also enhances the performance of the model on unobserved data. In order to prevent overfitting, regularization penalizes the coefficients, just like in machine learning methods. The weights of each node at all layers are adjusted by regularization in deep learning, similarly. Convolutional Neural Networks can be regularized using a variety of techniques following dropout, data augmentation and batch normalization.

3.6 Data Augmentation

Another less complicated but very effective method to prevent overfitting in sophisticated neural networks is data augmentation. Data augmentation can be used to enhance the sample size of training data in Convolutional Neural Networks, but it is not possible to do so in machine learning. When dealing with images, the quantity of training data samples can be enhanced by applying alteration on images such as scaling, shearing, rotating, width or height shifting and horizontal or vertical flipping.

The accuracy of Convolutional Neural Network models can be improved significantly with the use of this regularization technique, which is referred to as data augmentation.

Data can be enhanced in a number of ways, primarily through geometric transformations that build on simple visual modifications. The ease of implementation makes data augmentation techniques for geometric transformations quite popular. Geometric transformation approaches only have two drawbacks: increased training time and computing expenses. Different image processing operations are implemented during geometric transformation, and they are explained below.

Flipping: includes flipping the image on its horizontal and vertical axes. It is the simplest and most practical geometric data augmentation approach. More often than flipping on the vertical axis, flipping occurs on the horizontal axis.

Color Space: A single colour is isolated from a three-channel colour space RGB using the colour space-augmentation approach. The approach known as "colour space augmentation" is also quite successful and practical.

Cropping: Cropping can be used as an useful image processing technique to extract the core portion of each image when the images have heterogeneous heights and width dimensions.

Rotation: The image can be rotated between 1° and 359° in either the clockwise or counterclockwise direction.

Translation: Similar to rotation augmentation, photos can also be translated to the left, right, up, or down to minimize positional bias caused by the photographs showing the object in a different location.

Noise Injection: Noise injection is a different augmentation technique that includes injecting a matrix of random data points, typically drawn from a Gaussian distribution or another distribution.

For our VGG16-LSTM model, Data augmentation was done using Keras' ImageDataGenerator. By using data augmentation, we hope to improve the model's generalizability where each image in the batch was subjected to a series of arbitrary changes that included rotation, horizontal flipping, resizing, changes in scale and shearing etc.

3.7 Dataset Distribution

In our final dataset, each class has 1716 images belonging to 50 classes. We made a distribution of 80% videos for the training set and 20% for the validation set in the separate folders.

3.8 Feature Selection

With the help of their neural networks, deep learning techniques are able to automatically detect features, negating the need for a data scientist or engineer to do so. In our proposed architectures, ConvLSTM and VGG-LSTM, the spatiotemporal characteristics are automatically learnt, and the learned features are subsequently classified using an LSTM network.

3.9 Proposed Methods

3.9.1 Architectural innovations and applications of CNN

The study being conducted by CNN has a lot of potential to advance. From 2015 to 2019, CNN's success underwent the most significant adjustments. Several studies have demonstrated that cutting-edge deep architectures demonstrated promising results for challenging classification and localization tasks.

According on how the architecture has been modified, CNNs can be roughly categorized into seven different varieties, including CNNs that depend on attention and those that use feature maps and spatial exploitation.

3.9.2 Convolutional Neural Network – VGG16

In 2014, [47] proposed the VGG16 network architecture, which was created for the ILSVRC and trained on one thousand classes. A convolutional neural network with 16 layers is called VGG-16. After yielding such good results, VGG16 is considered one of the best vision model architectures ever developed. The most distinctive aspect of VGG16 is that instead of concentrating on having a large number of hyper-parameters, they prioritised having CNN layers with a 3x3 filter and kept the padding at 2x2 filter maxpool, stride 2. Throughout the entire network, convolutional layers and max pool are arranged in the same manner. The output is provided by a softmax after two fully connected layers (FC). There are 16 layers with weights, as indicated by the 16 in VGG16. With 138 million parameters, this network is quite big.

The local features that VGG-16 retrieved from video frames are fed into LSTM in in order to access spatiotemporal data. By gathering features, a pre-trained VGG16 extracts the characteristics of the video. Each vector from the prior step is processed by two LSTM layers with 256 units. The output stage is made up of a dense layer with 50 nodes as its final component.

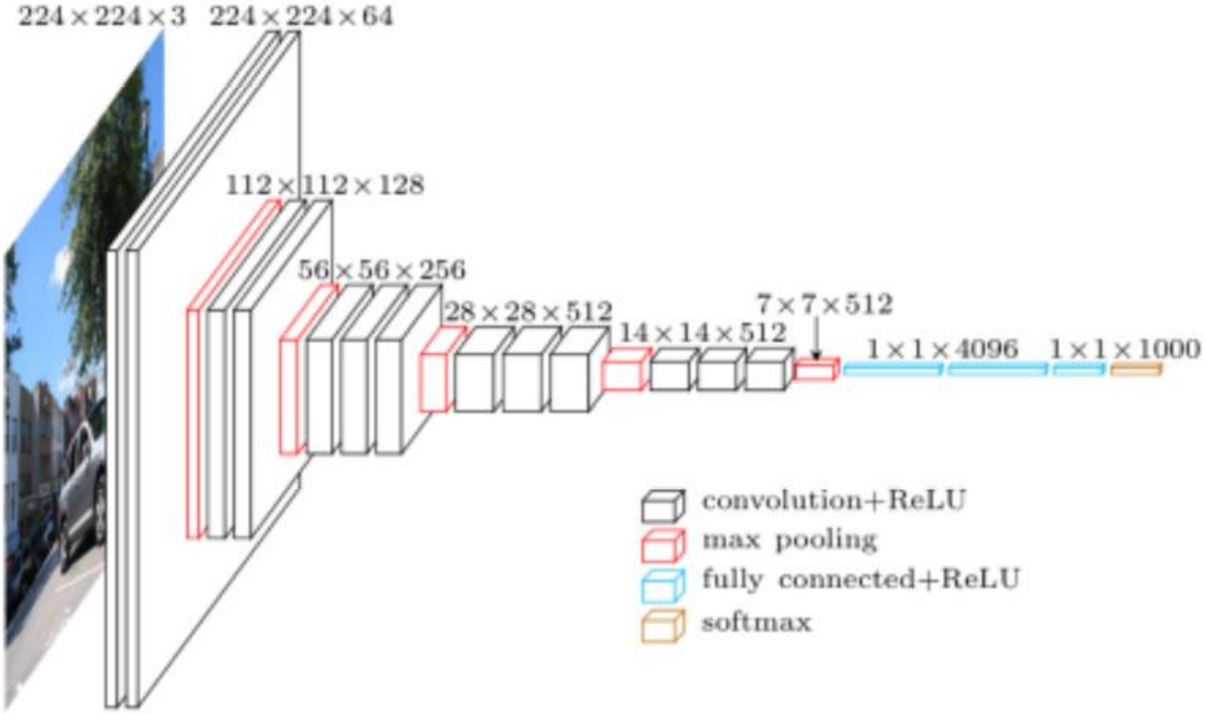


Figure 3 VGG16 Architecture [62]

3.9.3 ConvLSTM

A time series is a collection of data gathered across a number of time periods. In these circumstances, using a network built on LSTM (Long Short Term Memory) is an intriguing approach. The model transfers the prior hidden state to the following step in the sequence in this type of design. As a result, the network keeps track of earlier data and uses it to influence judgments. In other words, the sequence of the data is crucial.

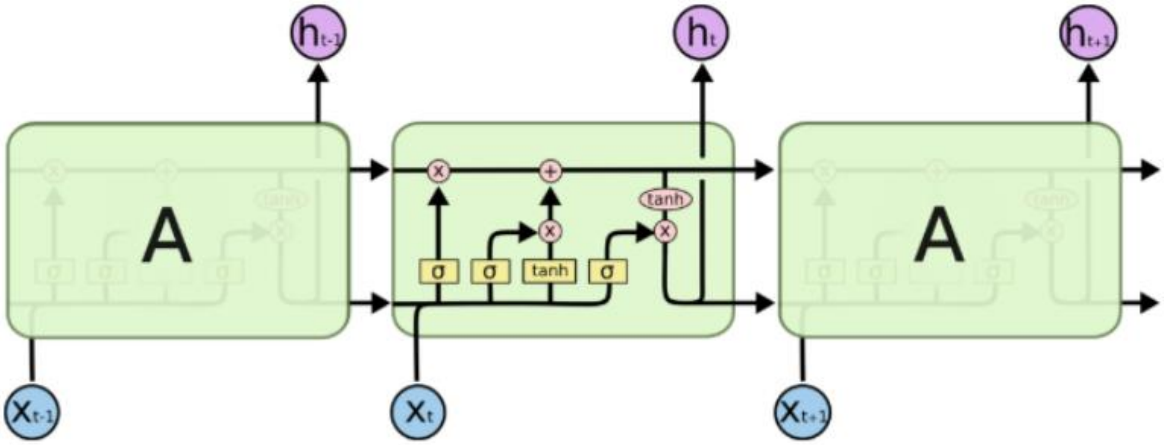
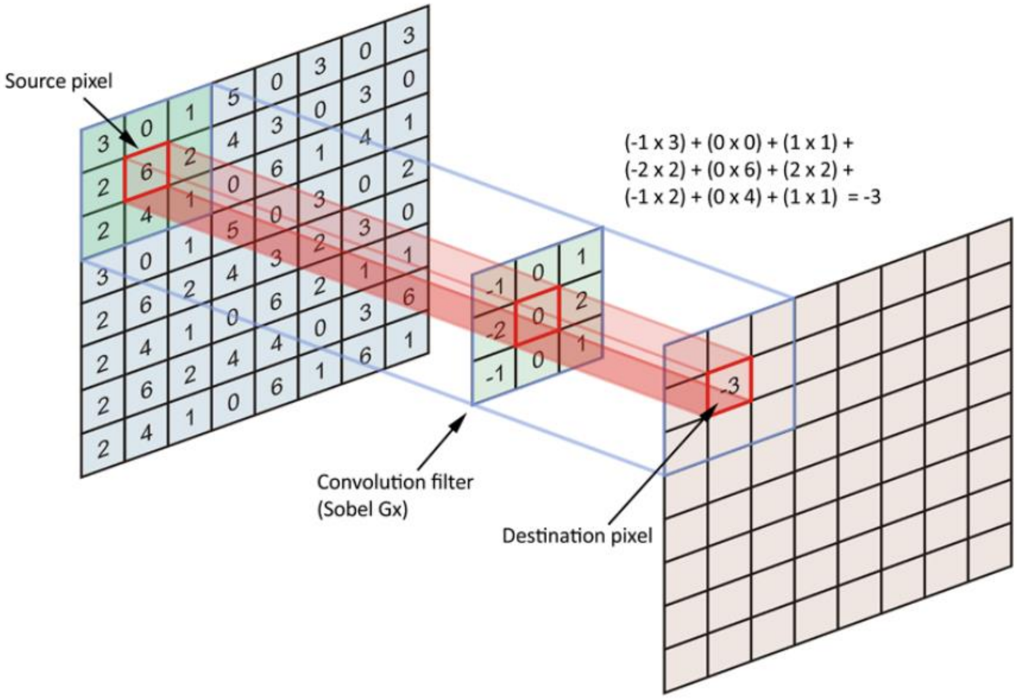


Figure 4 LSTM Architecture [63]

The ideal method for dealing with images is a CNN (Convolutional Neural Network) architecture. Convolutional layers, which the image passes through to extract significant information, are used. The output is joined to a fully-connected Dense network after going through a few convolutional layers sequentially.



Source pixel
(-1 x 3) + (0 x 0) + (1 x 1) +
(-2 x 2) + (0 x 6) + (2 x 2) +
(-1 x 2) + (0 x 4) + (1 x 1) = -3
-3
Convolution filter
(Sobel Gx)
Destination pixel

Figure 5 Convolution of an image with one filter [65]

ConvLSTM layers are one strategy we might use in our scenario of sequential photos. It is a recurrent layer, precisely like the LSTM, but convolutional operations are used instead of inside mathematical operations. As an outcome, instead of being just a 1-dimensional vector representing features, the data that flows through the ConvLSTM cells retains the 3 dimensional input.

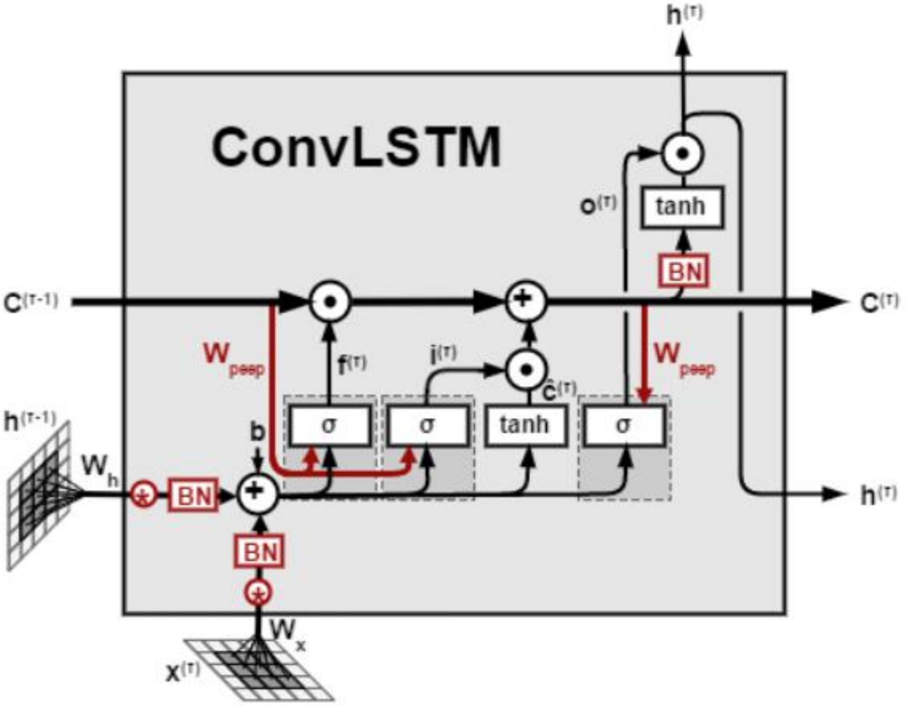


Figure 6 A ConvLSTM cell [65]

A Convolutional LSTM model, which takes a different approach from a ConvLSTM and runs an image into convolutional layers before flattening it into a 1D array with the resulting features, is one example of a ConvLSTM. A number of characteristics across time is produced by applying this procedure repeatedly to all of the images in the time set; this output serves as the input for the LSTM layer.

3.9.4 Proposed Architecture: VGG16-LSTM

The proposed architecture for VGG16-LSTM is shown in **Figure 7**, where an input of shape (240,240,3) is fed into the model for feature extraction and classification. The detailed steps are given in the **Chapter 4 Results and Analysis Section 4.2.**

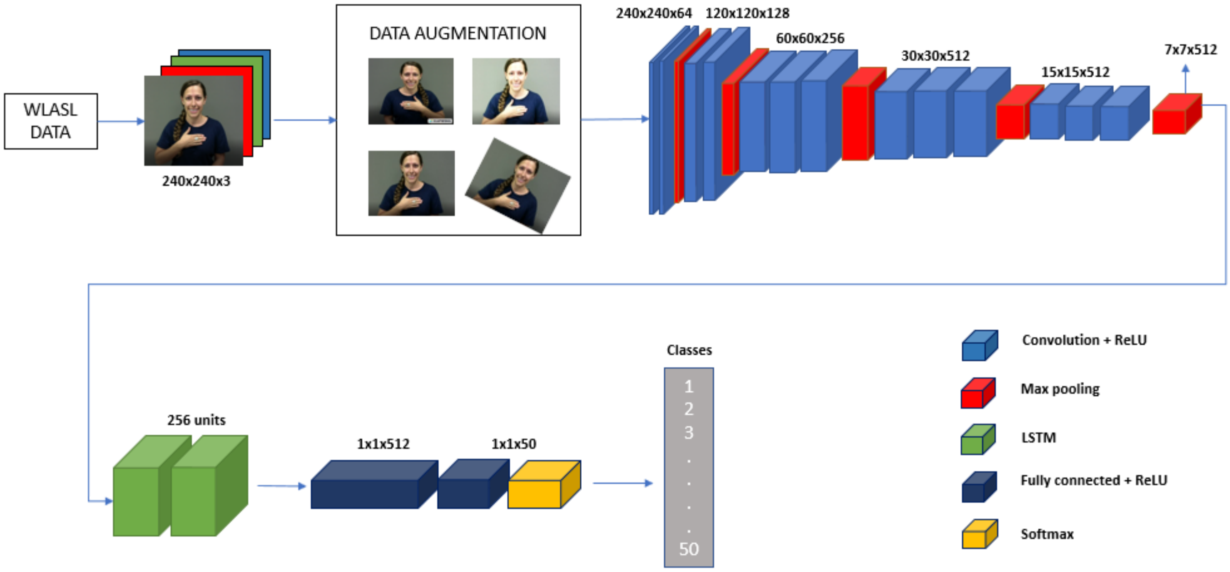


Figure 7 Proposed Architecture VGG16-LSTM

Optimizer	Stochastic Gradient Descent
Activation Function	relu, Softmax
Learning Rate	0.0001
Dropout	0.3
Number of epochs	50
Batch Size	32
Early Stopping Monitor	Loss = categorical crossentropy

Table 1 VGG16-LSTM Configuration

INPUT SHAPE	**(None, 240, 240, 3)**	
LAYER TYPE	**OUTPUT SHAPE**	**NO. OF PARAMETERS**
block1_conv1 (Conv2D)	(None, 240, 240, 64)	1792
block1_conv2 (Conv2D)	(None, 240, 240, 64)	36928
block1_pool (MaxPooling2D)	(None, 120, 120, 64)	0
block2_conv1 (Conv2D)	(None, 120, 120, 128)	73856
block2_conv2 (Conv2D)	(None, 120, 120, 128)	147584
block2_pool (MaxPooling2D)	(None, 60, 60, 128)	0
block3_conv1 (Conv2D)	(None, 60, 60, 256)	295168

block3_conv2 (Conv2D)	(None, 60, 60, 256)	590080
block3_conv3 (Conv2D)	(None, 60, 60, 256)	590080
block3_pool (MaxPooling2D)	(None, 30, 30, 256)	0
block4_conv1 (Conv2D)	(None, 30, 30, 512)	1180160
block4_conv2 (Conv2D)	(None, 30, 30, 512)	2359808
block4_conv3 (Conv2D)	(None, 30, 30, 512)	2359808
block4_pool (MaxPooling2D)	(None, 15, 15, 512)	0
block5_conv1 (Conv2D)	(None, 15, 15, 512)	2359808
block5_conv2 (Conv2D)	(None, 15, 15, 512)	2359808
block5_conv3 (Conv2D)	(None, 15, 15, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
reshape_6 (Reshape)	(None, 49, 512)	0
lstm_14 (LSTM)	(None, 49, 256)	787456
lstm_15 (LSTM)	(None, 256)	525312
dense_16 (Dense)	(None, 512)	131584
dense_17 (Dense)	(None, 50)	25650
TOTAL PARAMETERS	16,184,690	
TRAINABLE PARAMETERS	8,549,426	
NON-TRAINABLE	7,635,264	

Table 2 VGG16-LSTM Configuration on WLASL Dataset

3.9.5 Proposed Architecture: ConvLSTM

The proposed architecture for ConvLSTM is shown in **Figure 8**, where sequence of frames with input shape of 1x150x150x4 is fed into the model for feature extraction and classification. The detailed steps are given in the **Chapter 4 Results and Analysis Section 4.2.**

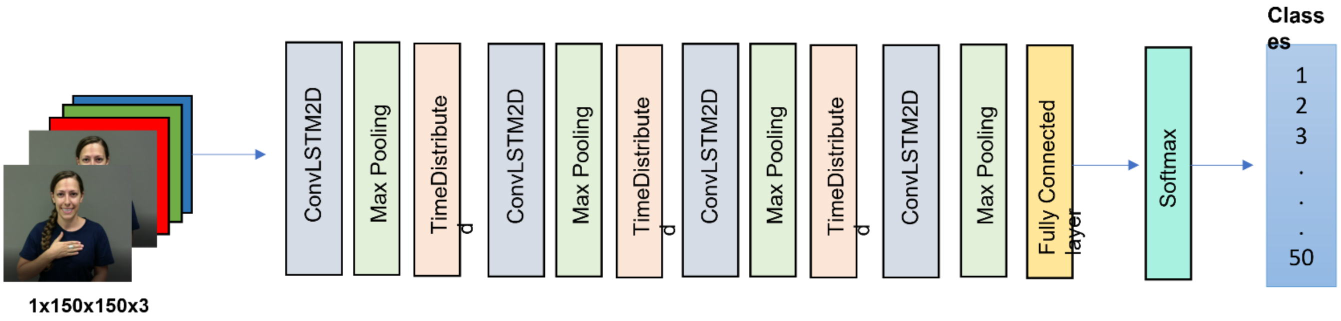


Figure 8 Proposed ConvLSTM Architecture

Optimizer	Adam
Activation Function	tanh
Dropout	0.2
Kernel Size	(3, 3)
Number of epochs	70
Early Stopping Monitor	Loss = categorical crossentropy

Table 3 ConvLSTM Configuration

INPUT SHAPE	**(None, 1, 150, 150, 4)**	
LAYER TYPE	**OUTPUT SHAPE**	**NO. OF PARAMETERS**
CONVLSTM2D	(None, 1, 148, 148, 4)	1024
MAXPOOLING3D	(None, 1, 74, 74, 4)	0
TIMEDISTRIBUTED	(None, 1, 74, 74, 4)	0
CONVLSTM2D	(None, 1, 72, 72, 8)	3488
MAXPOOLING3D	(None, 1, 36, 36, 8)	0
TIMEDISTRIBUTED	(None, 1, 36, 36, 8)	0
CONVLSTM2D	(None, 1, 34, 34, 14)	11144
MAXPOOLING3D	(None, 1, 17, 17, 14)	0
TIMEDISTRIBUTED	(None, 1, 17, 17, 14)	0
CONVLSTM2D	(None, 1, 15, 15, 16)	17344
MAXPOOLING3D	(None, 1, 8, 8, 16)	0
FLATTEN	(None, 1024)	0F
DENSE	(None, 53)	54325

TOTAL PARAMETERS	87,325
TRAINABLE PARAMETERS	87,325
NON-TRAINABLE	0

Table 4 ConvLSTM Configuration on WLASL Dataset

CHAPTER 4: Results and Analysis

4.1 Preliminaries

This study used a subset of the WLASL Dataset. Because the original dataset contained several discrepancies, a reduced selection was used. Due to comparable qualities being removed during the training process for various signs, the presence of several interpretations for a single sign gesture and ambiguity in the signs causes the accuracy of the model to be reduced. The 50 ASL terms that are used the most frequently were used. Before using each sequence as an input for the model, preprocessing was done on each one. To boost the training set size and improve performance, the dataset was augmented. For testing purpose, we collected 20 ASL videos for several gestures from publicly available online sources.

4.2 Model Testing

First, we obtained the World-Level American Sign Language dataset (WLASL). There are 2000 classes in WLASL. We selected 50 classes for this study from those 2000 classes. The collected ASL videos are then transformed into frames in the following step. A split of 80:20 was performed, with saving 80% for training and 20% for validation set in separate folders portion of the data set was divided into 80:20, with 80% used for training and 20% for validation videos in a different folder. To assess how well the model can generalize, a separate collection of 20 ASL test videos were collected from web resources.

In order to expand the size of the training data for the Convolutional Neutral Network (VGG16), the frames we collected are subsequently processed by using the technique of data augmentation. The 240x240 input frames with 3 RGB channels that resulted from the use of the data augmentation approach are utilized as the input frames for the model (VGG16). The max pooling layer output is sent into a huge network of two LSTM layers with 256 units. There are two fully connected layers after the large layer with 256 LSTM units. This layer connects all of the neurons in its preceding layers to each neuron in the layers above it. Following the fully connected dense layers, a SoftMax layer is trained for the final prediction.

The categorical cross entropy of the provided loss function was optimized via a stochastic gradient descent with momentum. A ReLU activation function was employed. Back-propagations and numerous iterations are applied using the loss function for convergence and learning the training dataset. During the compilation, which comes after epochs and a dropout, the model's parameters were adjusted. By running several epochs and comparing the prediction error, best suitable dropout was chosen. To achieve better accuracy and avoid overfitting, dropout was modified from 0.1 to 0.3, and the number of epochs was raised to 50. In order to use the test videos for prediction, it was transformed into NumPy array first. The average accuracy was computed based on the total number of predicted videos.

For the ConvLSTM model, we used input sequence frames with a size of 150x150 and three channels of RGB as an input to the first layer of ConvLSTM with kernel size (3,3) coupled with a time-distributed layer without using data augmentation. There is one fully connected layer after the ConvLSTM layers. Following the fully connected layer, a SoftMax layer is trained for the final classification. The categorical cross entropy of the provided loss function was optimized using Adam. Dropout was modified to 0.2, and the number of epochs was raised to 70. The same procedure was used for all the testing's subsequent steps as mentioned above.

4.3 Results and Discussion

Similar sign movements with different meanings can occur in ASL, leading to misclassification. Experimenting with various combinations of the number of LSTM layers, LSTM nodes, and dropout variance led to the selection of the best model. The accuracy of class prediction was used as the evaluation metric for model comparison. Comparatively, VGG16-LSTM, the variant with 256 units performed the best of the available options. The training accuracy reported for VGG16-LSTM is reported as 95% and an average accuracy of 57% was calculated on 20 ASL test videos for prediction of sign words while the ConvLSTM reported 93% training accuracy and was not able to generalize new ASL videos reporting only 22% test accuracy on test videos. We tested both models on 20 word-level ASL videos taken from online sources and 12 of them were correctly classified by VGG16-LSTM. We also evaluated our best model on some continuous videos of ASL and it surprisingly classified some of the sign keywords from ASL sentence videos quite well. Additionally, it has been observed that choosing lesser number of classes resulted in significantly higher accuracy overall, but choosing more classes resulted in lower accuracy. The amount of sample videos provided for each word and the inconsistency in the gestures for various interpretations are the main causes of this. Several classes are there which contains both facial features along with hand movements.

Without the use of keypoints for the face, hand joints, and pose-based algorithms, predicting such complex gestures with any degree of accuracy is quite impossible.

Below are the results of bar plots for prediction count and scatter plots for probability of classes being classified by our model VGG16-LSTM from test videos of word-level ASL and continuous-level ASL videos shown in Figure 9 to Figure 41.

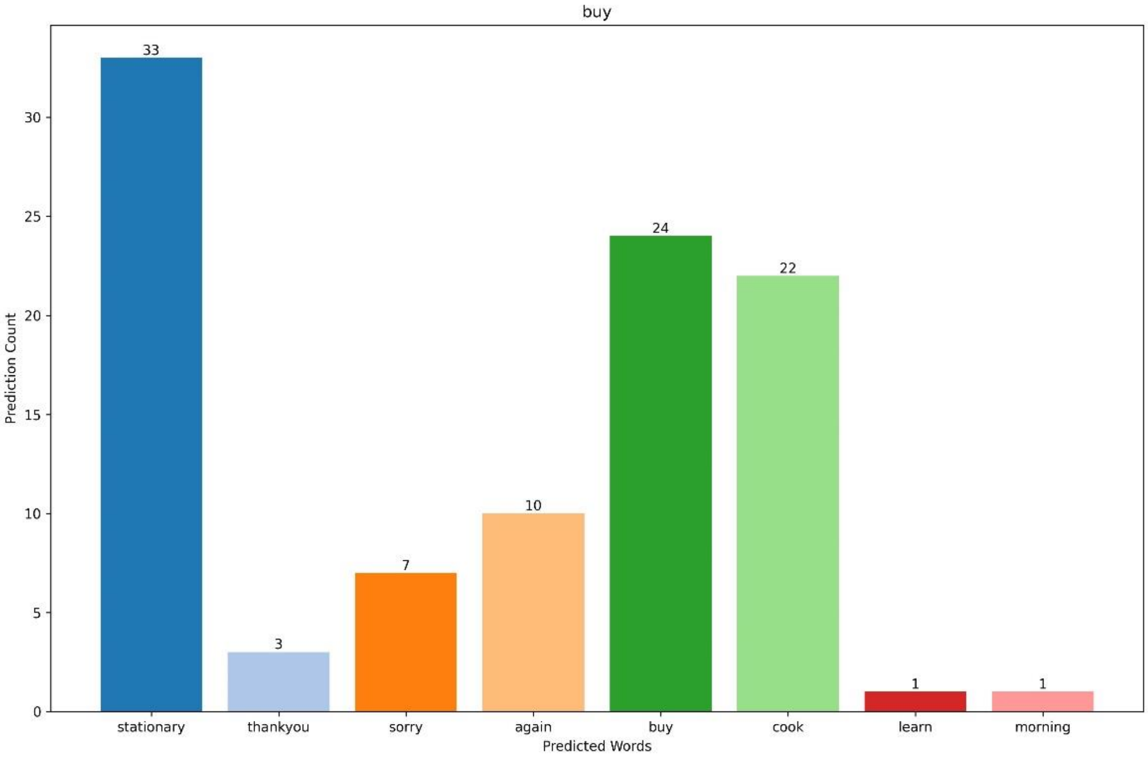


Figure 9 Prediction count plot for word "buy"

In this bar plot, it is clearly shown that word “buy” has been predicted 24 times and the prediction count for class “stationary” is 33, the reason behind this is the pace variation where the signer starts performing the sign after some seconds where he/she is in an idle position (not moving) therefore the stationary class has higher prediction count than “buy”. We notice that there has been some classification as well where the model confuses the word “buy” with “cook”, “again” and some other classes but their counts are lower than “buy”. The cause of overlapping is the similarity between the signs for two different words. Similarly, the plot results for different classes are shown below where it is observed how many times a gesture has been predicted and on number of frames.

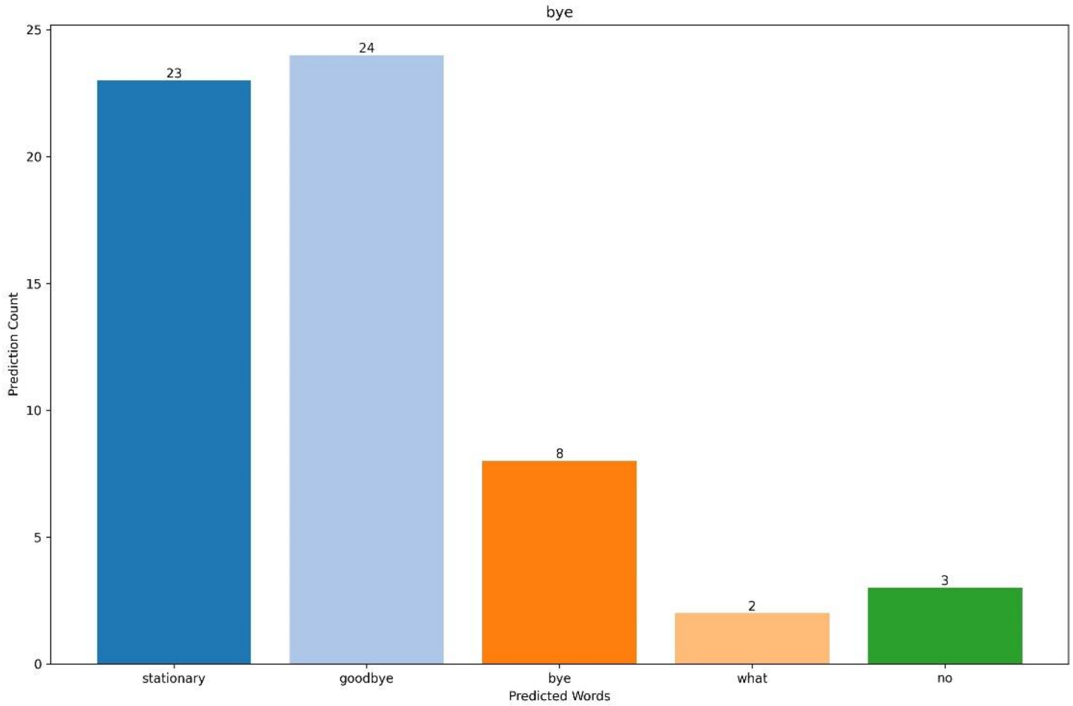


Figure 10 Prediction count plot for word "bye"

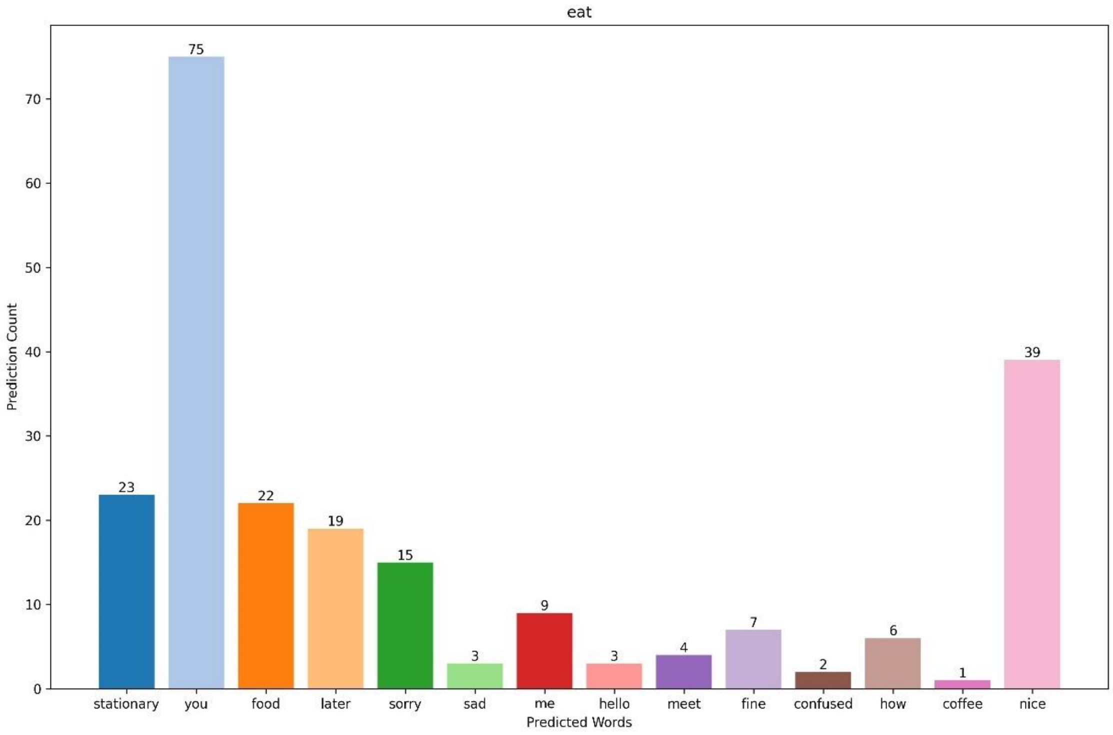


Figure 11 Prediction count plot for word "eat" misclassified into different classes

The study’s findings demonstrated that some classes were predicted quite accurately than the others due to having more number of examples but some classes even with lesser number of samples were predicted with very less error and the interesting reason behind this was that those classes had distinct features than the others such as for word “hello” , “understand”, “sad” , “smile”, the model was able to identify these gestures from different videos.

There were some other conditions as well, where in certain videos the model could not recognize the sign gestures because of the poor quality or a lot of background noise and the lighting issues.

In Figure 12, the gesture for word “how” is predicted with highest score with “stationary” class which was again due to the signer being idle for some time.

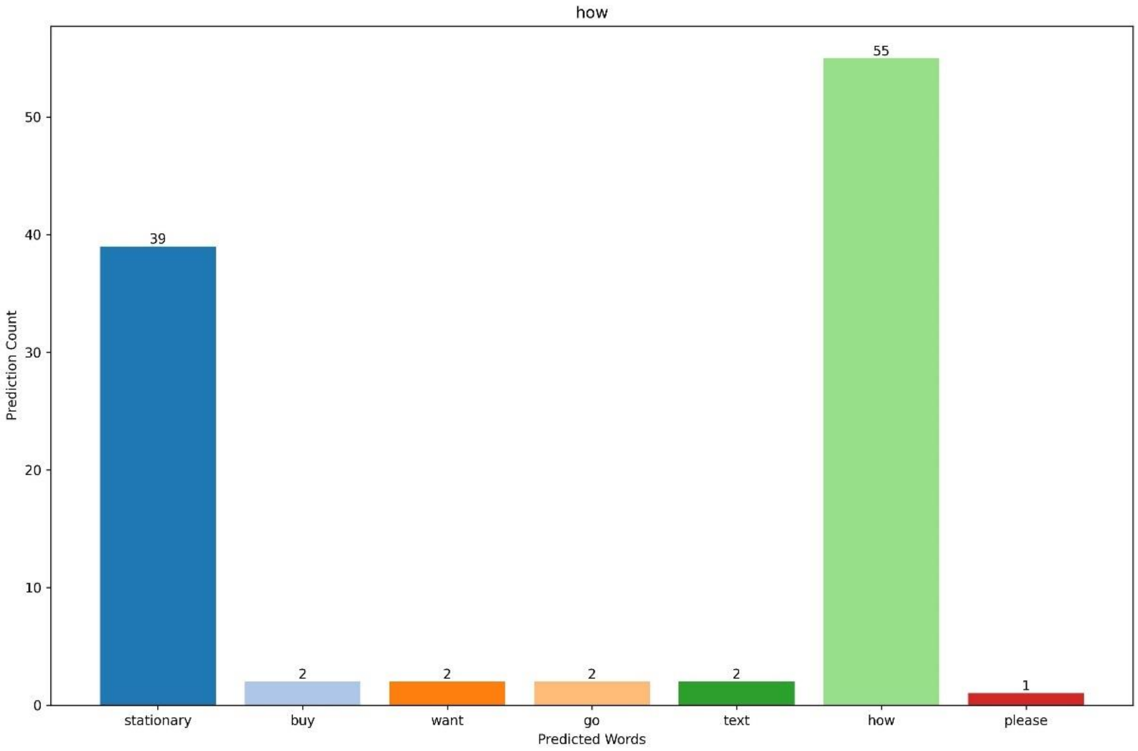


Figure 12 Prediction count plot for word "how"

Figure 13, 14 and 15 shows the prediction count for the words “learn” and “please” and “sad” is the highest among others, showing the model’s generalization ability for these gestures.

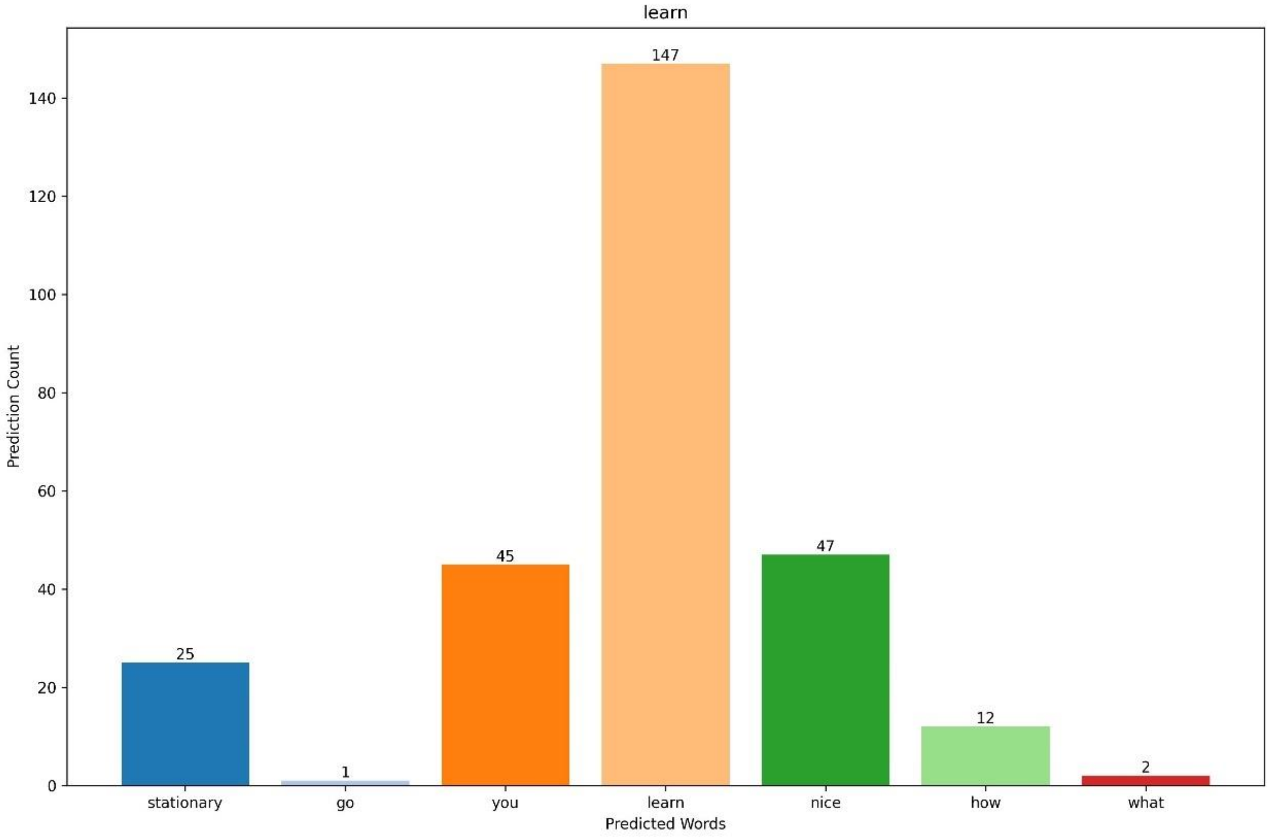


Figure 13 Prediction count plot for word "learn"

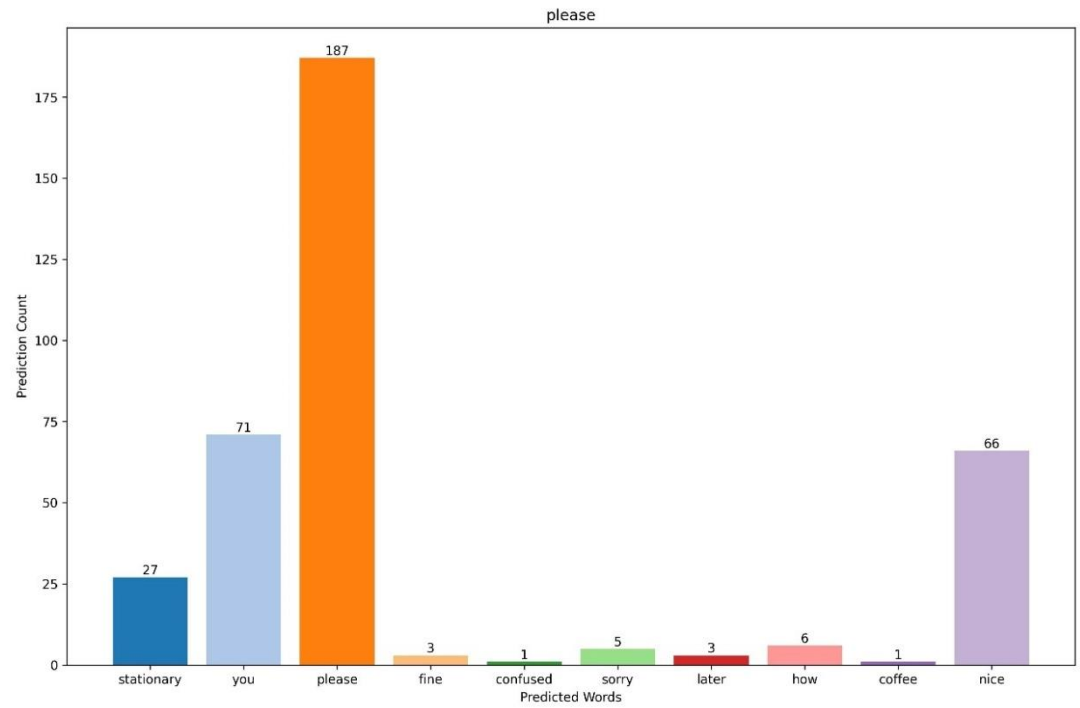


Figure 14 Prediction count plot for word "please"

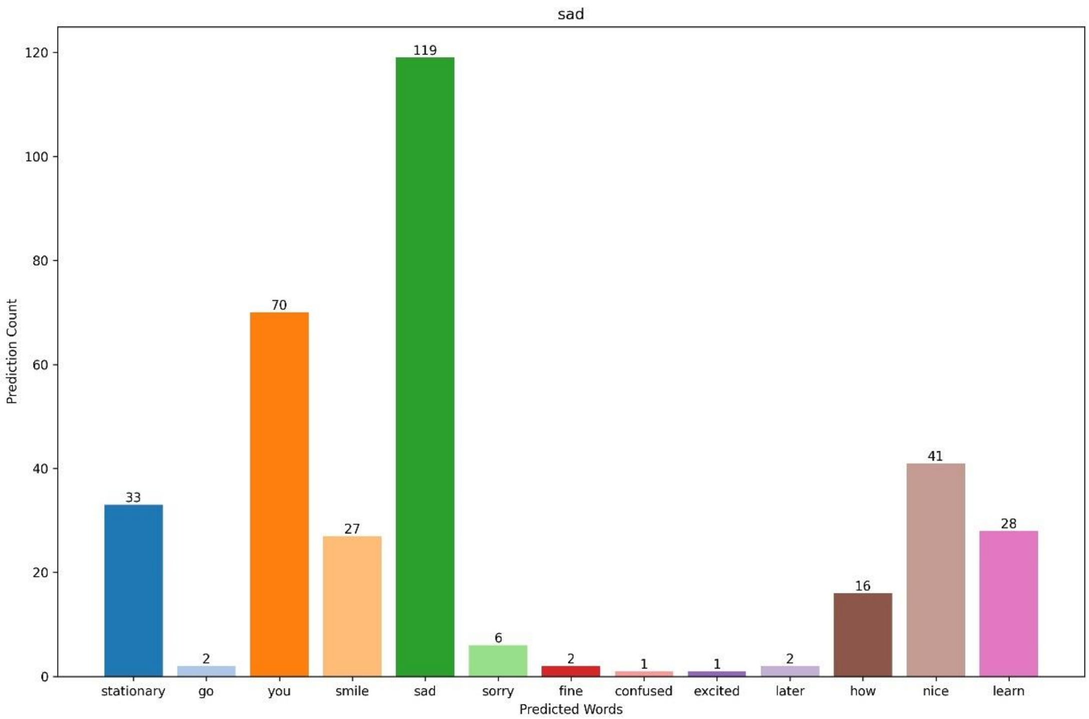


Figure 15 Prediction count plot for word "sad"

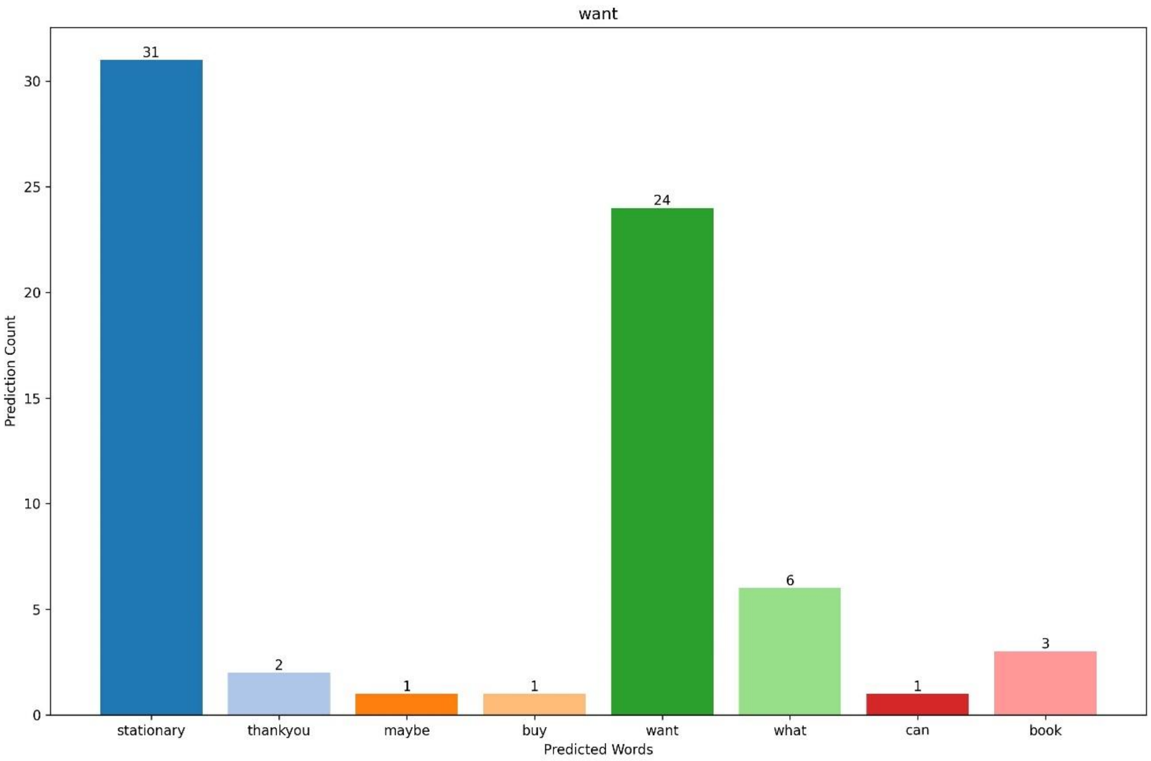


Figure 16 Prediction count plot for word "want"

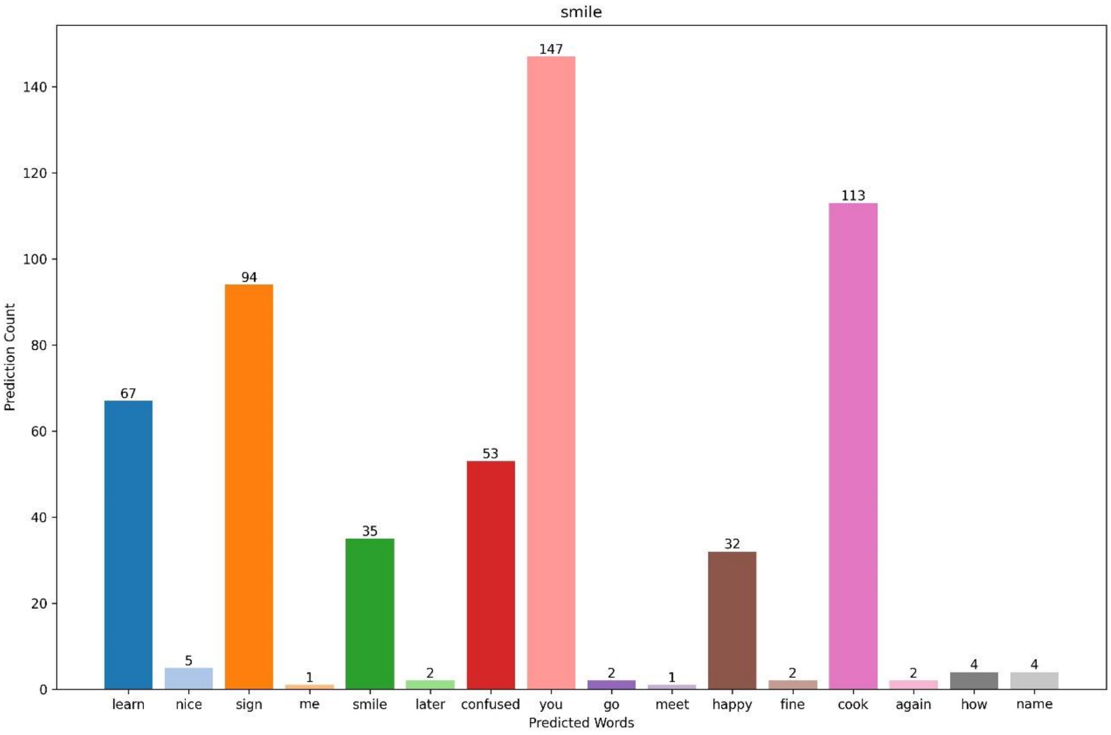


Figure 17 Prediction count plot for word "smile" misclassified into different classes

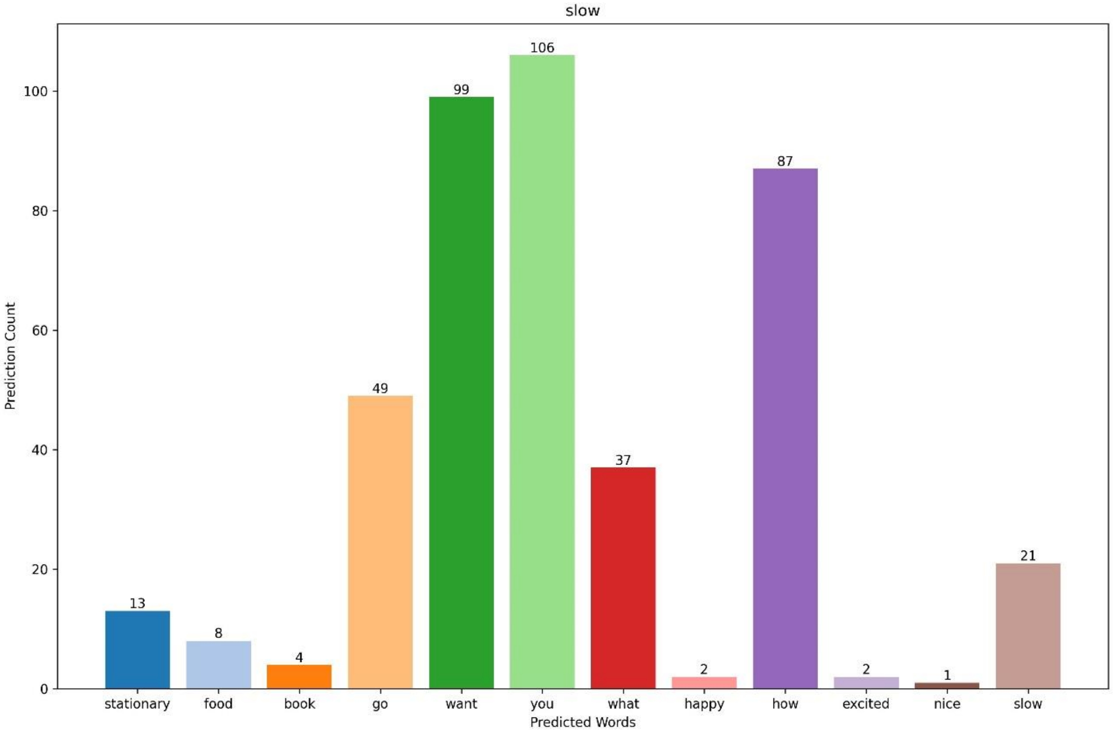


Figure 18 Prediction count plot for word "slow"misclassified in 3 to 4 different classes.

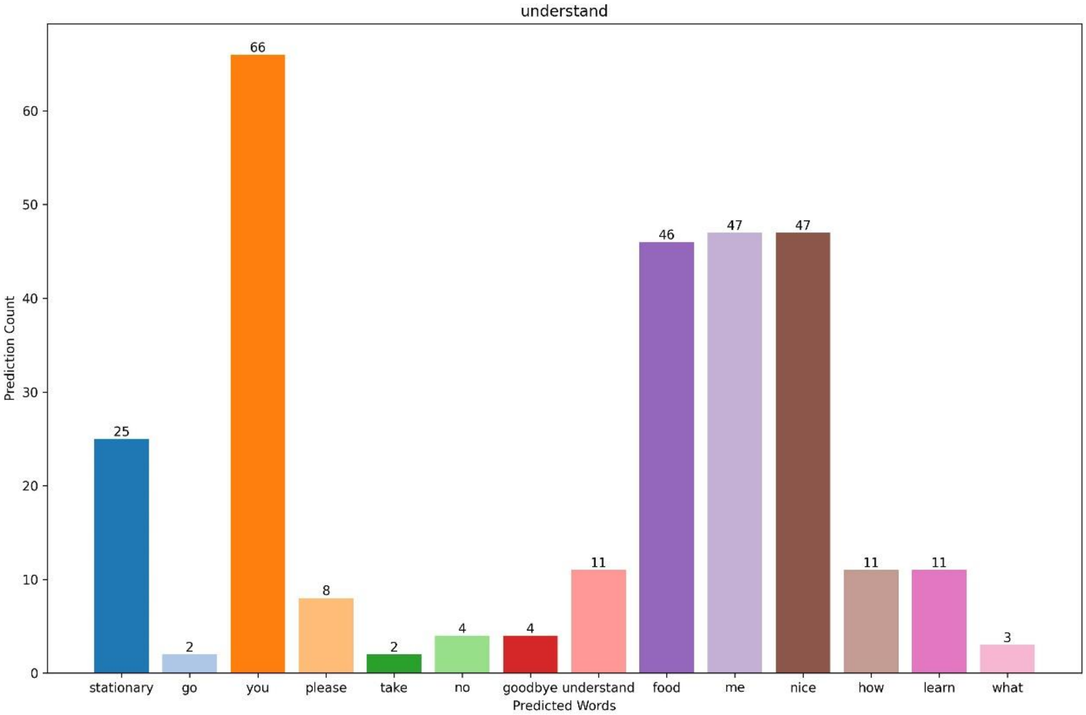


Figure 19 Prediction count plot for word "understand"

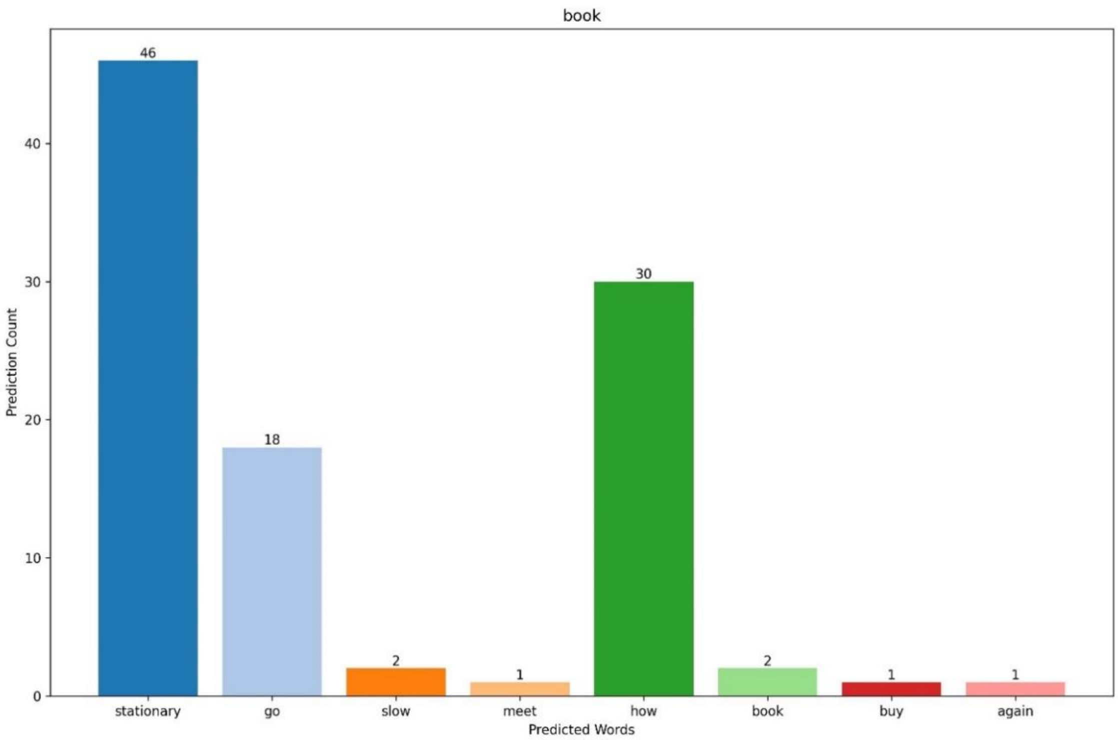


Figure 20 Prediction count plot for word "book"

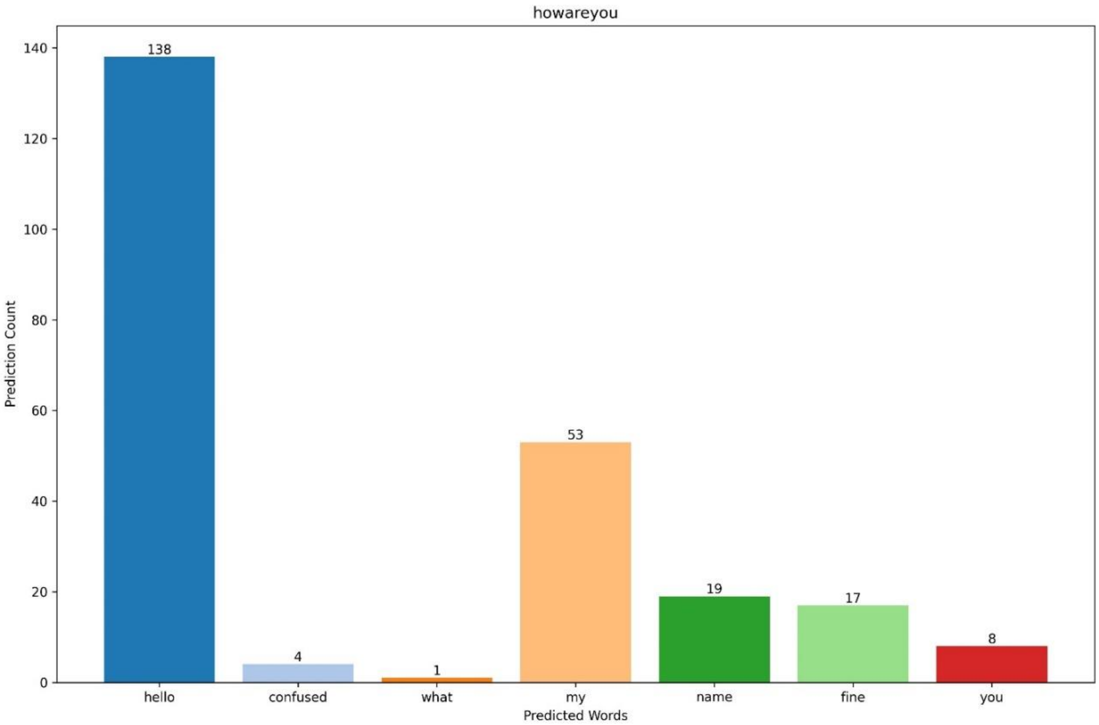


Figure 21 Prediction count plot for ASL sentence "how are you"
Sign keyword spotting from continuous video, localization result for word "you"

We tested a few ASL sentence videos as well, **Figure 21** is the plot for prediction count for the words “me” and “happy” from the continuous video containing more than one gesture. The system was able to localize and spot the sign keywords “me” and “happy” very well. In ASL, “I am happy” is signed as “me happy” which has the same meaning.

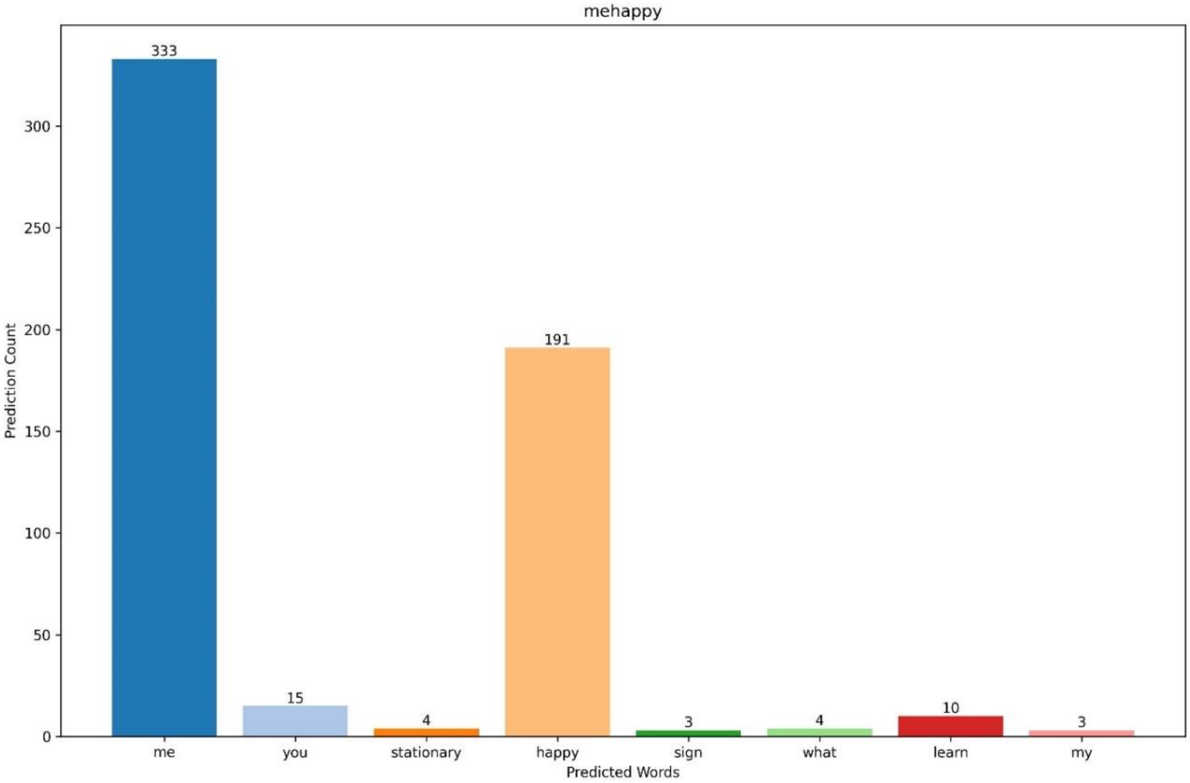


Figure 21 Prediction count plot for ASL sentence "me happy"
Sign keyword spotting from continuous video, localization result for the words “me” and “happy”

A few more results of ASL sentences are shown below predicted by our proposed model VGG16-LSTM, **Figure 22** demonstrates the ability of model to spot the sign keywords “hello”, “my” and “name” with good results from the continuous video.

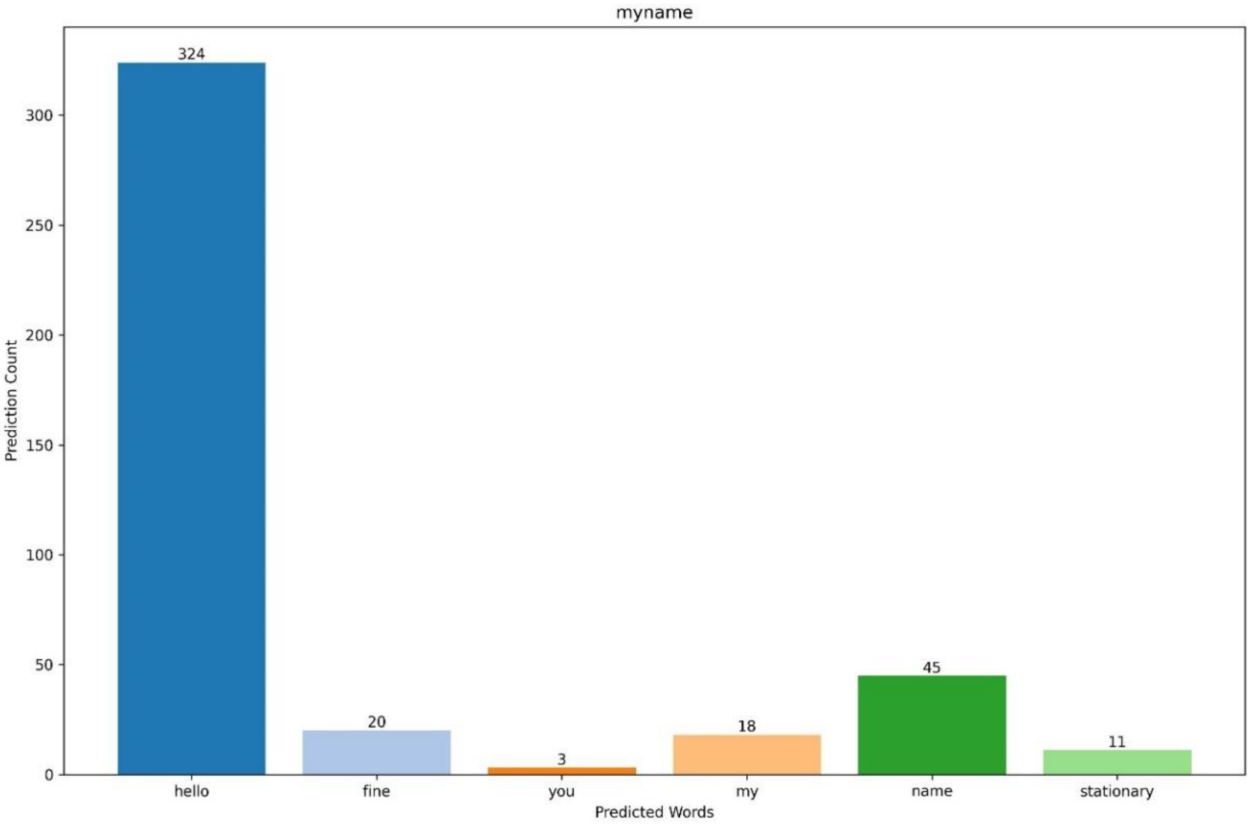


Figure 22 Prediction count plot for ASL sentence "hello my name"
Sign keyword spotting from continuous video, localization result for the words “hello”,”my”,”name”

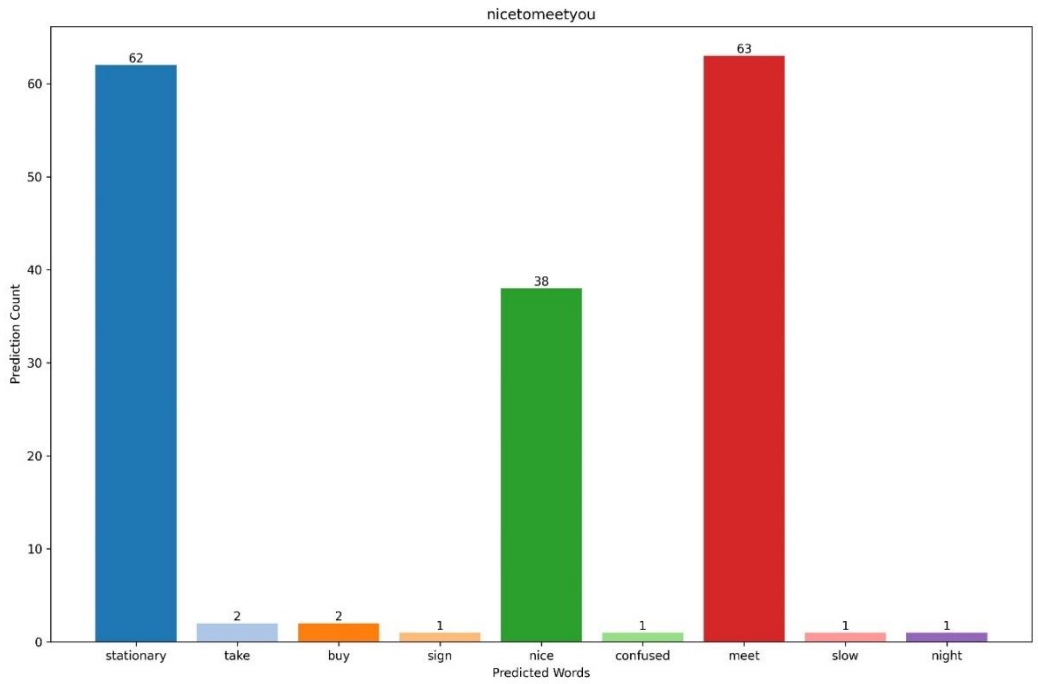


Figure 23 Prediction count plot for ASL sentence "nice to meet you"
Sign keyword spotting from continuous video, localization result for the words "nice","meet"

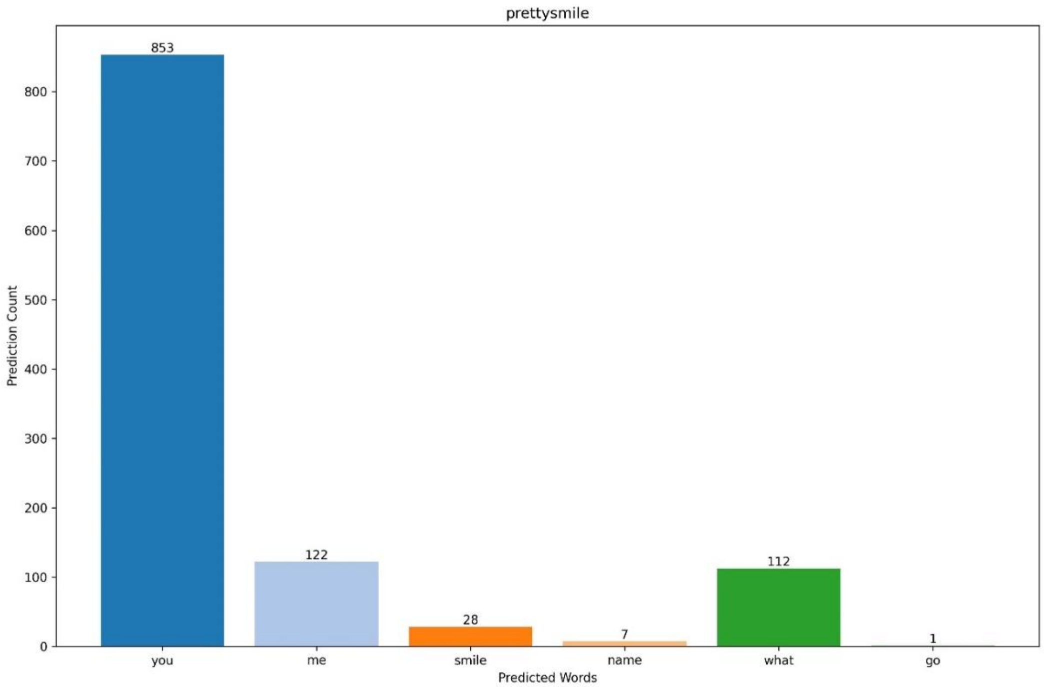


Figure 24 Prediction count plot for ASL sentence "Your smile is pretty"
Sign keyword spotting from continuous video, localization result for the words "you", "smile"

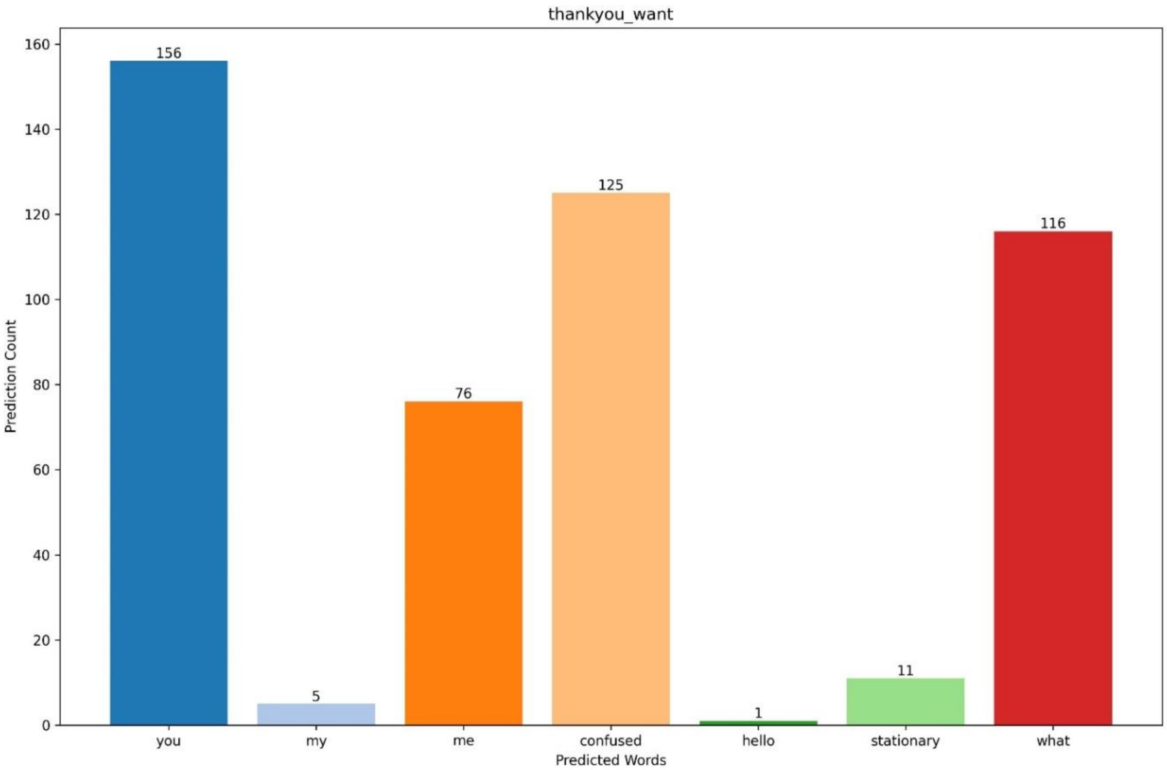


Figure 25 Prediction count plot for continuous ASL video "thankyou and want" localization result for the words "you", "want" is misclassified as "what"

The scatter plots were obtained by calculating the highest probability of each class predicted on the test videos. The figure below shows the scatter plots for high confidence predicted probability of class “book” among others on each frame.

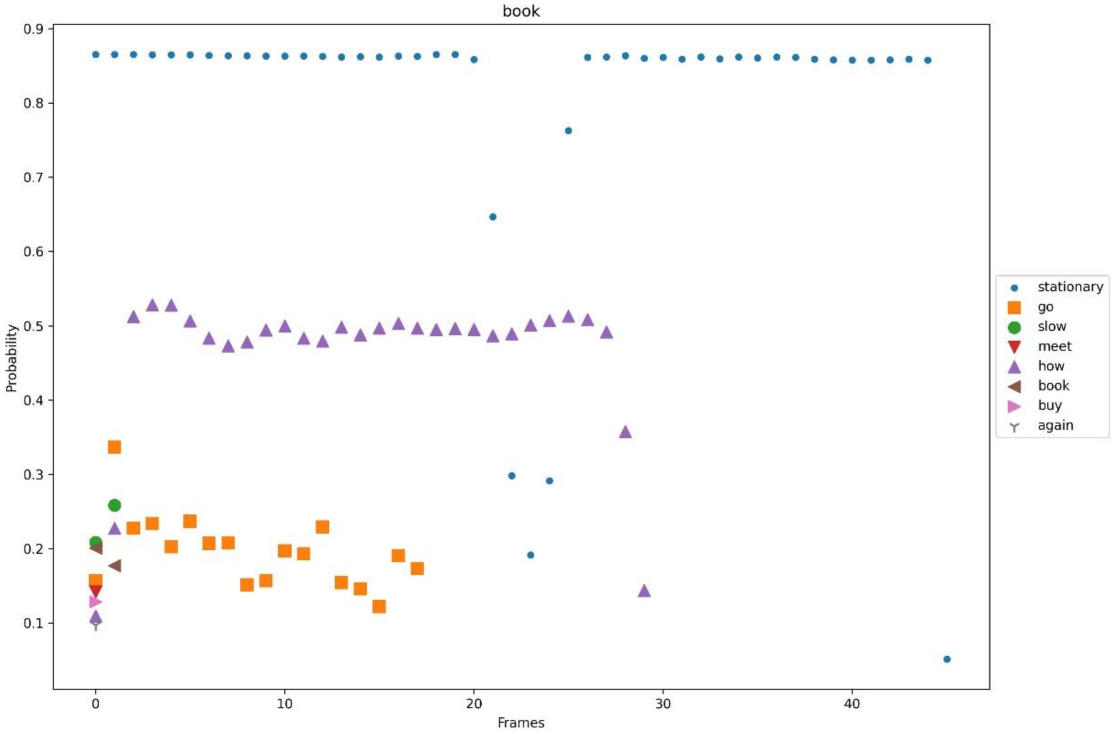


Figure 26 Scatter plot for probability of predicted class "book"

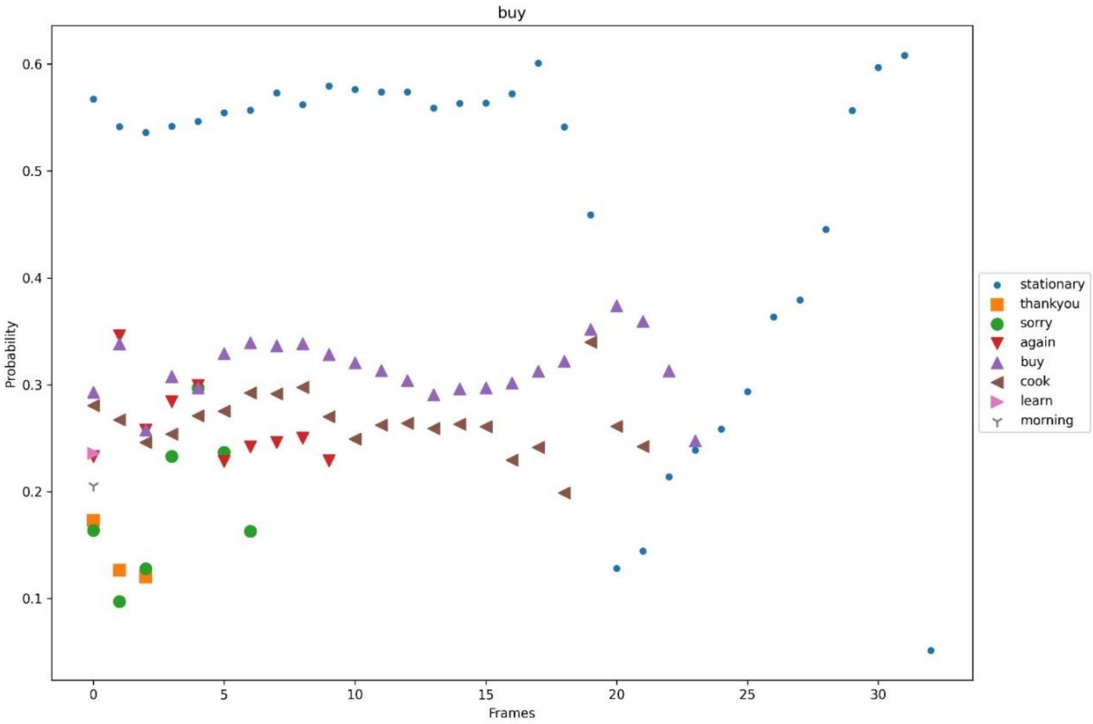


Figure 27 Scatter plot for probability of predicted class "buy"

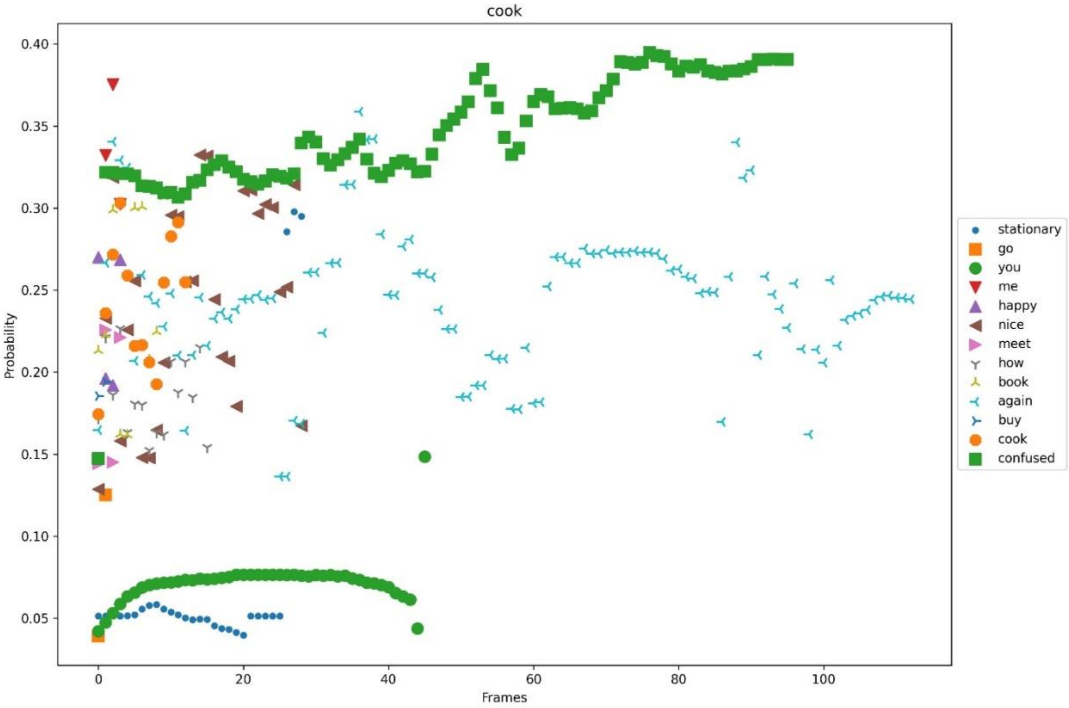


Figure 28 Scatter plot for probability of predicted class "cook"

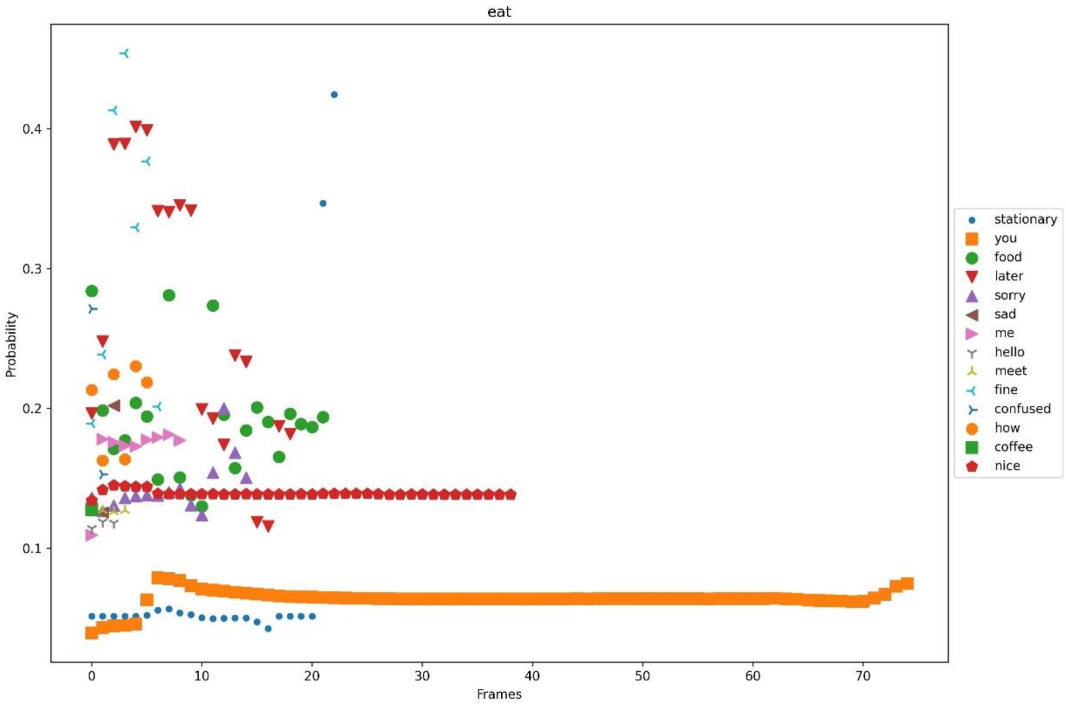


Figure 29 Scatter plot for probability of predicted class "eat"

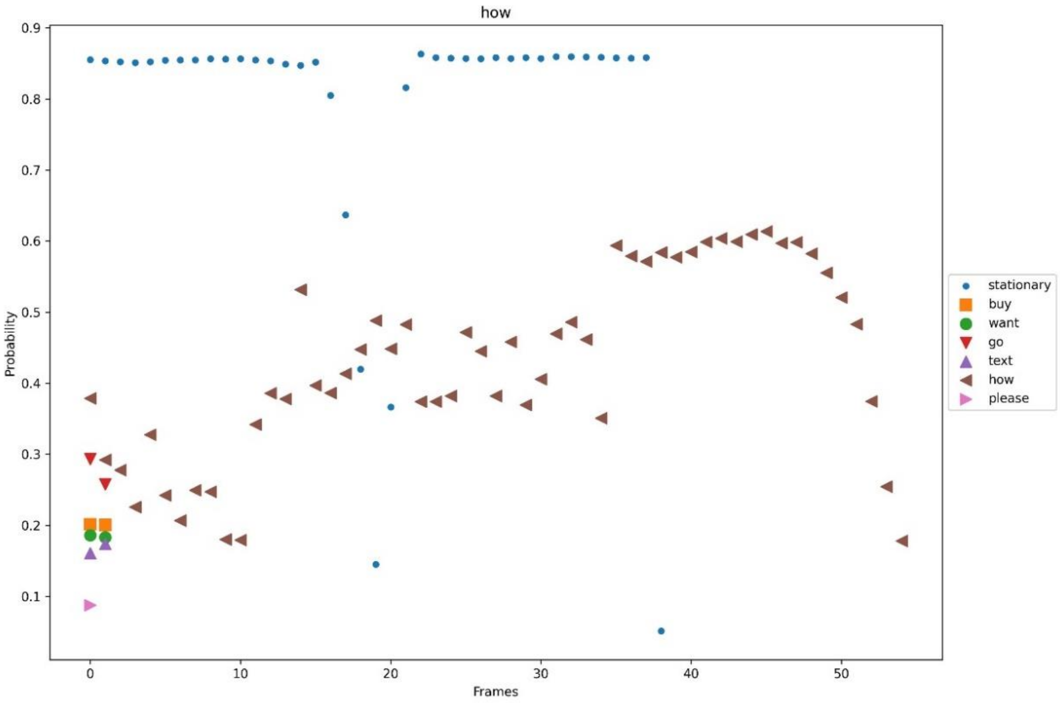


Figure 30 Scatter plot for probability of predicted class "how"

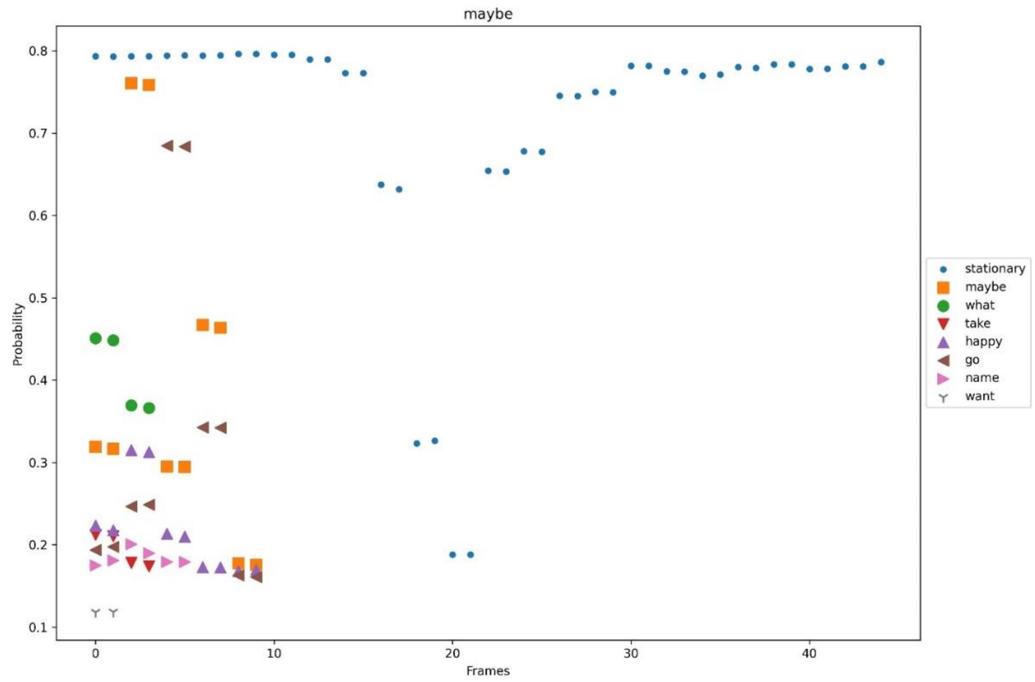


Figure 31 Scatter plot for probability of predicted class "maybe"

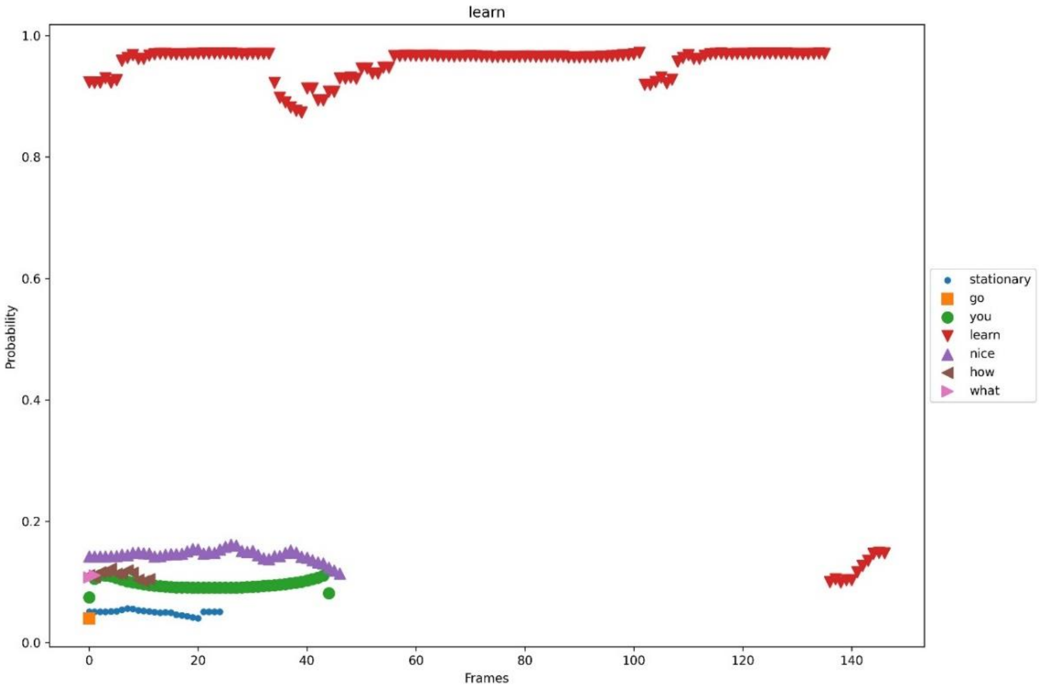


Figure 32 Scatter plot for probability of class "learn"

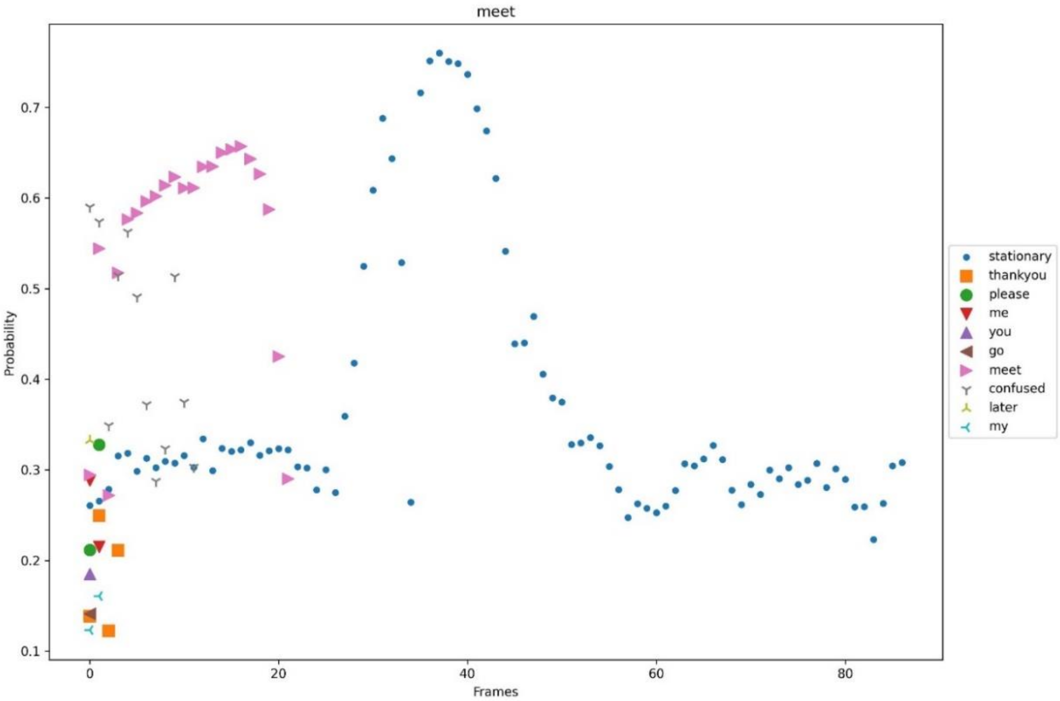


Figure 32 Scatter plot for probability of class "meet"

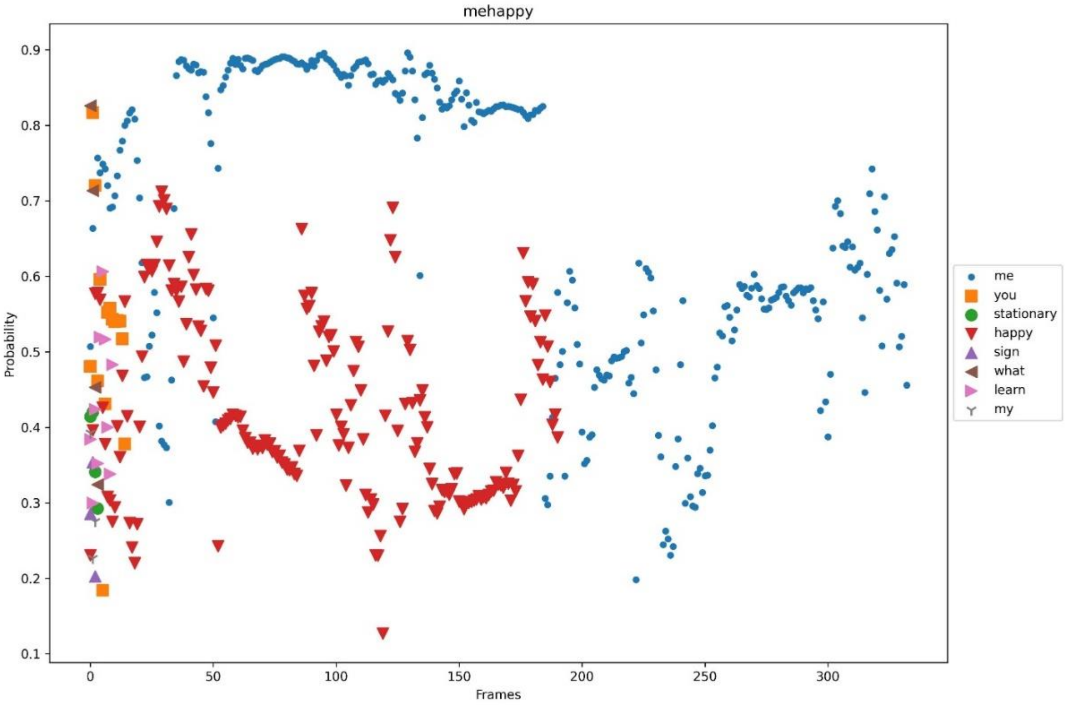


Figure 33 Scatter plot for probability of class "me" and “happy”

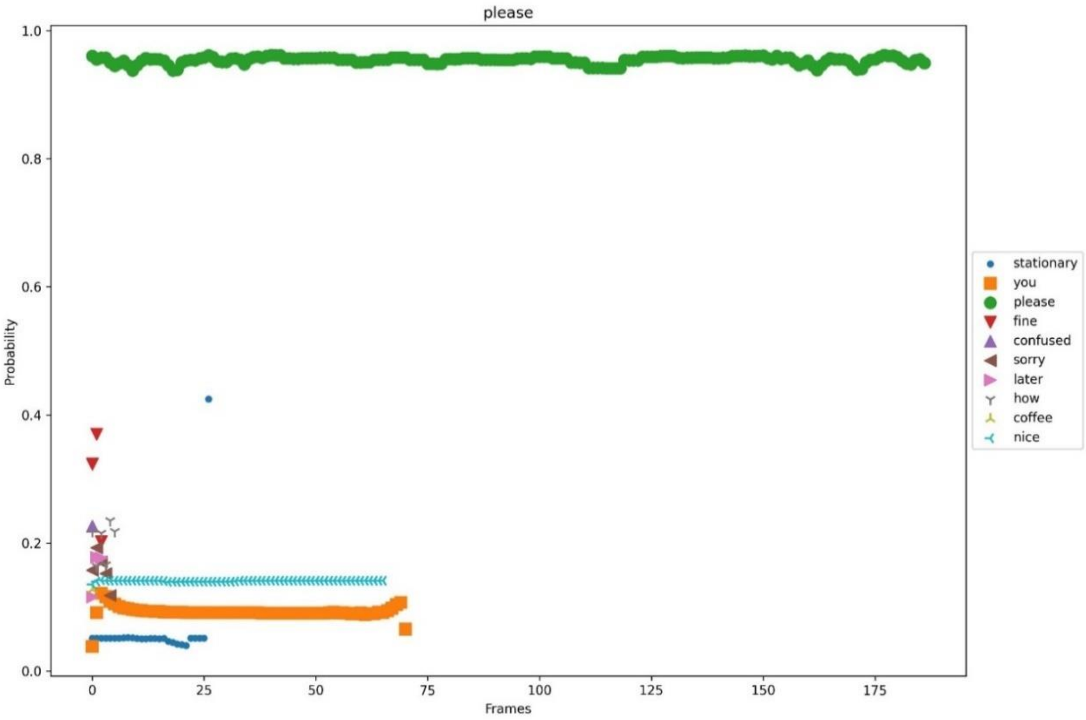


Figure 34 Scatter plot for probability of class "please''

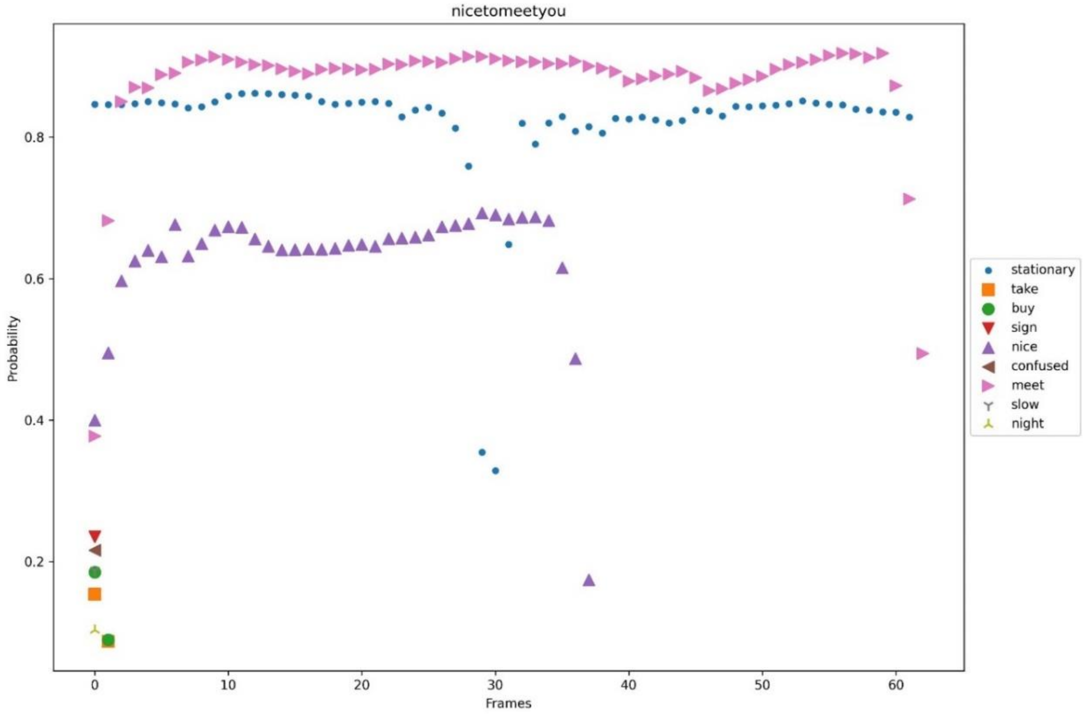


Figure 35 Scatter plot for probability of class "meet''and ''nice'' from ASL sentence "nice to meet you''

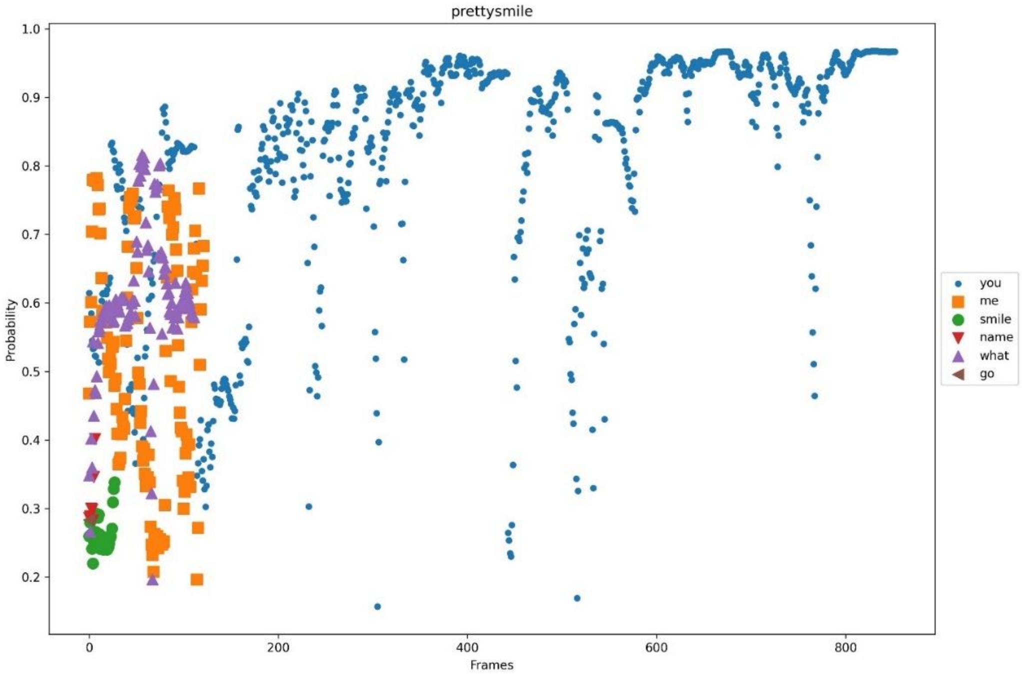


Figure 36 Scatter plot for probability of class "smile"and "you" from ASL sentence "you have pretty smile"

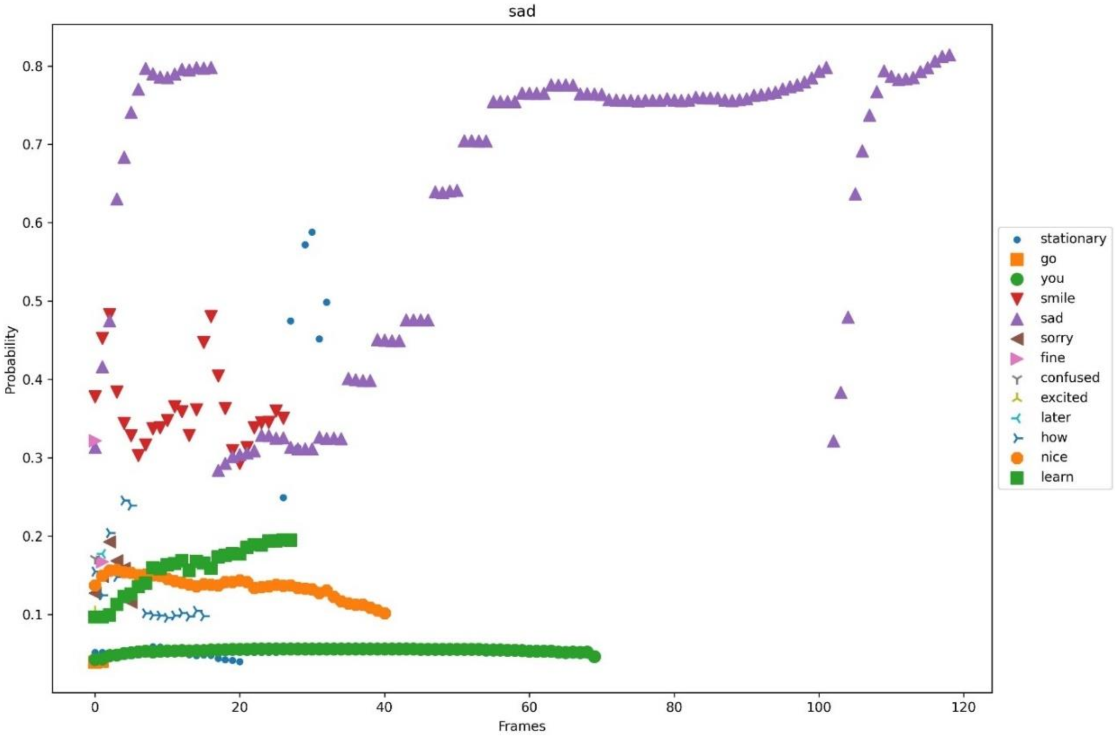


Figure 37 Scatter plot for probability of class "sad"

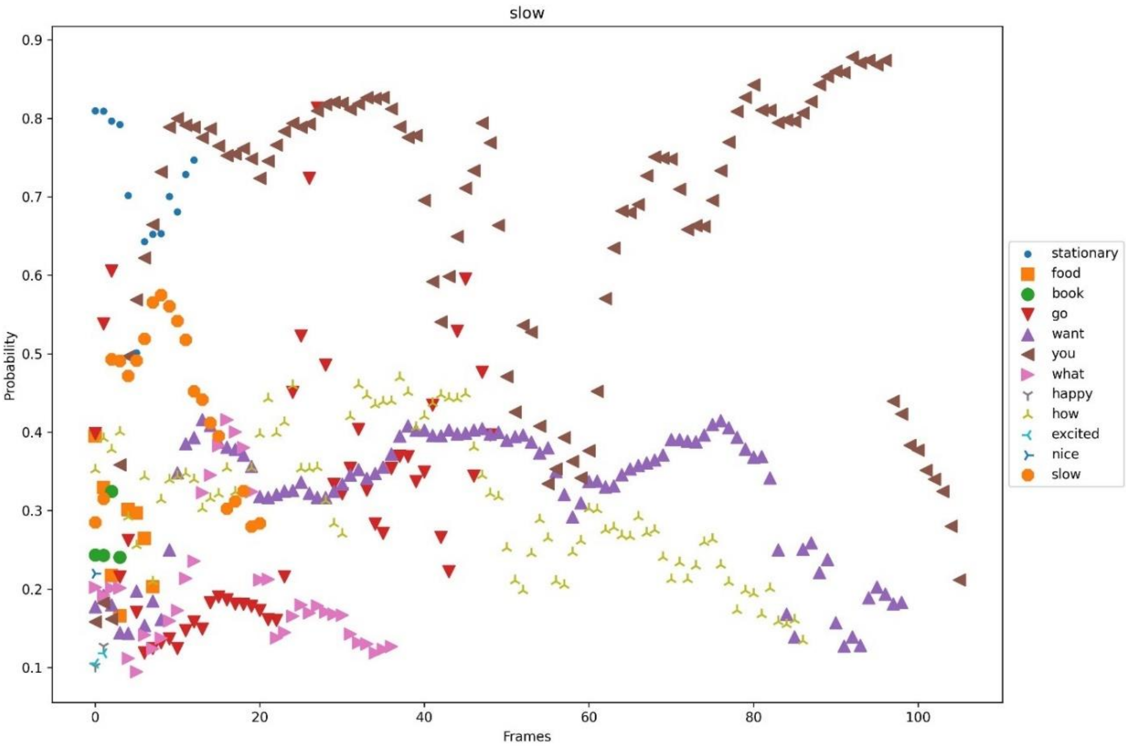


Figure 38 Scatter plot for probability of class "slow"

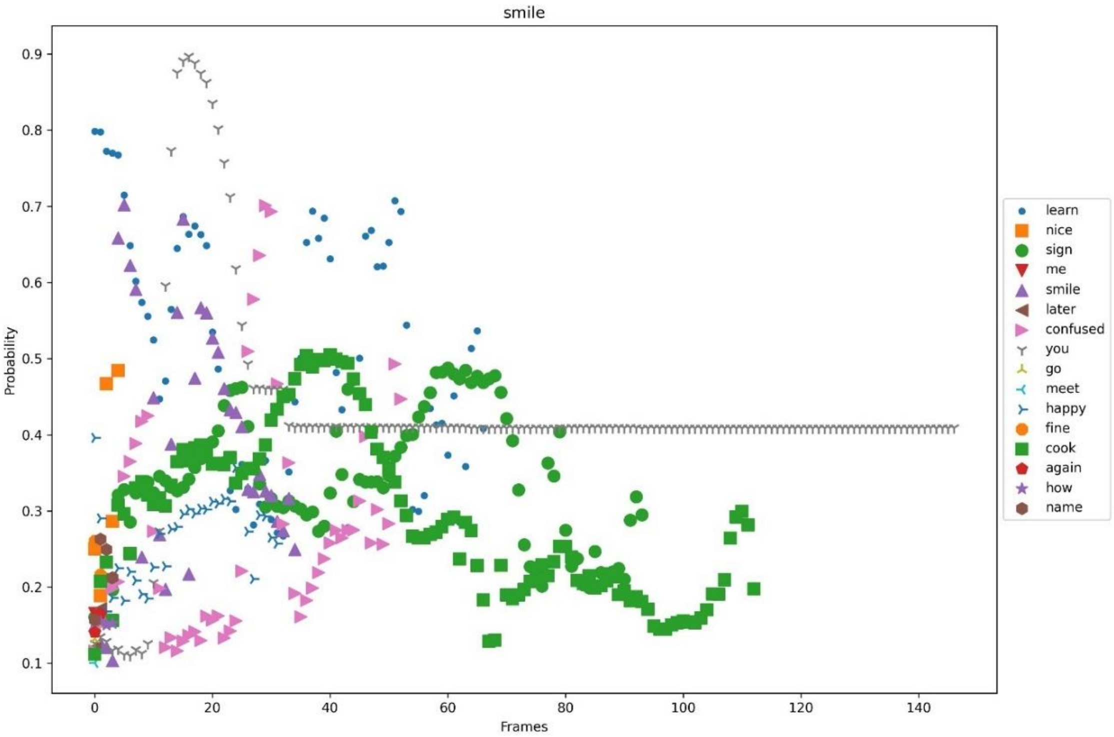


Figure 39 Scatter plot for probability of class "smile"

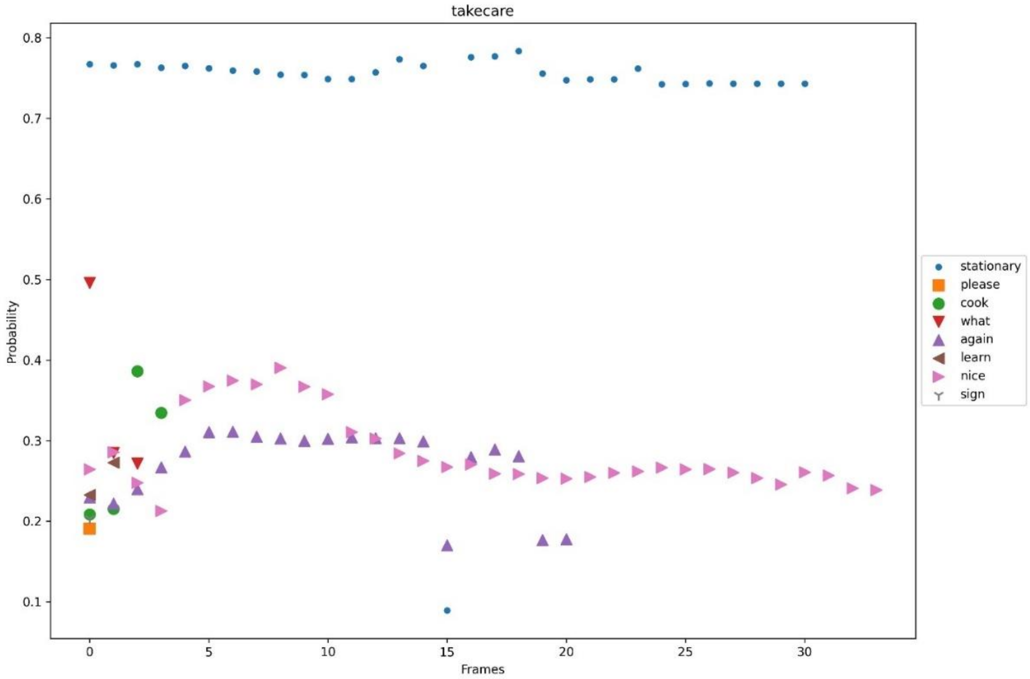


Figure 40 Scatter plot for probability of class "take care"

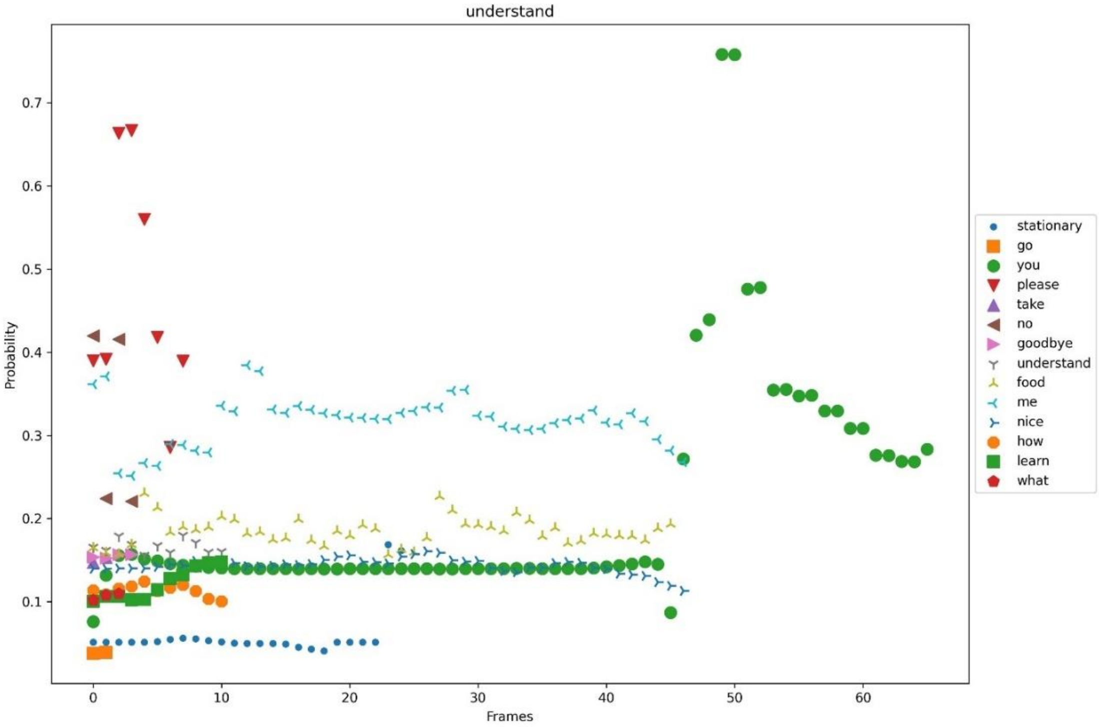


Figure 41 Scatter plot for probability of class "understand"

CHAPTER 5: Future Work

For future work, we can focus on developing an understanding of sentence-level sign language recognition and sign language translation from continuous videos. We plan to work with larger set of data having ample amount of sample videos for each class. The better the dataset, the better the results will be.

One potential improvement for word-level sign recognition can be possible to experiment with Gated Recurrent Unit (GRU) incorporated with RNN (Independent) to improve the efficiency of the max pooling. Also, a dataset that includes more example videos for each class is needed for better outcomes. Pose-based approaches where meaningful information or features like keypoints of hand and arm joints are extracted may provide a way to aid in the development of a reliable model. Since extracting the features of facial expressions from videos is more difficult than extracting the features of hand gestures, facial expression identification is a potential area for future research.

CHAPTER 6: Conclusion

This research used deep learning methods to identify the best model for word-level American Sign Language Recognition. To improve generalization ability, two deep learning architectures namely VGG16-LSTM and ConvLSTM were modified and tested. The American Sign language recognition techniques proposed in this study were implemented using an Intel Core i5 12400F 4.40GHz system with Nvidia RTX 3070 8GB Palit Gamerock Graphic Card and 32 GB RAM. An average test accuracy of 57% was attained on 20 ASL videos in experiments using normal speed sign detection videos collected from the online sources. The proposed model in this study makes accurate predictions for some classes but not for many others. It is not a good model to be used for commercial reasons since the prediction is not reliable, but it can be improved with a large dataset. One of the causes of misclassification is overlapping characteristics and features in different classes that results in increased ambiguity. The identification step was challenging since the computational cost of the system depended on the size of classes and the number of example videos present in each class. This deterioration in the performance of model inspires further study in this domain.

CHAPTER 7: REFERENCES

1. Gupta, R. and S. Rajan, *Comparative analysis of convolution neural network models for continuous indian sign language classification.* Procedia Computer Science, 2020. **171**: p. 1542-1550.
2. Ahmed, M.A., et al., *A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017.* Sensors, 2018. **18**(7): p. 2208.
3. Cheok, M.J., Z. Omar, and M.H. Jaward, *A review of hand gesture and sign language recognition techniques.* International Journal of Machine Learning and Cybernetics, 2019. **10**(1): p. 131-153.
4. Aryanie, D. and Y. Heryadi. *American sign language-based finger-spelling recognition using k-Nearest Neighbors classifier*. in *2015 3rd International Conference on Information and Communication Technology (ICoICT)*. 2015. IEEE.
5. Masood, S., H.C. Thuwal, and A. Srivastava, *American sign language character recognition using convolution neural network*, in *Smart Computing and Informatics*. 2018, Springer. p. 403-412.
6. Ganesh, P., *CONTINUOUS AMERICAN SIGN LANGUAGE TRANSLATION WITH ENGLISH SPEECH SYNTHESIS USING ENCODER-DECODER APPROACH*. 2021.
7. Mitchell, R.E., et al., *How many people use ASL in the United States? Why estimates need updating.* Sign Language Studies, 2006. **6**(3): p. 306-335.
8. Traxler, C.B., *The Stanford Achievement Test: National norming and performance standards for deaf and hard-of-hearing students.* Journal of deaf studies and deaf education, 2000. **5**(4): p. 337-348.
9. Ye, Y., et al. *Recognizing american sign language gestures from within continuous videos*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018.
10. Li, D., et al. *Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison*. in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2020.
11. Mahmoud, R., S. Belgacem, and M.N. Omri, *Towards an end-to-end isolated and continuous deep gesture recognition process.* Neural Computing and Applications, 2022: p. 1-20.
12. Breiman, L., *Bagging predictors.* Machine learning, 1996. **24**(2): p. 123-140.
13. Lafferty, J., A. McCallum, and F.C. Pereira, *Conditional random fields: Probabilistic models for segmenting and labeling sequence data.* 2001.
14. Friedman, N., D. Geiger, and M. Goldszmidt, *Bayesian network classifiers.* Machine learning, 1997. **29**(2): p. 131-163.
15. Cunningham, P. and S.J. Delany, *K-nearest neighbour classifiers-a tutorial.* ACM Computing Surveys (CSUR), 2021. **54**(6): p. 1-25.
16. Schuldt, C., I. Laptev, and B. Caputo. *Recognizing human actions: a local SVM approach*. in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* 2004. IEEE.
17. Rabiner, L.R., *A tutorial on hidden Markov models and selected applications in speech recognition.* Proceedings of the IEEE, 1989. **77**(2): p. 257-286.
18. Quinlan, J., *Induction of decision trees. mach. learn.* 1986.
19. Li, Q., et al., *Continuous dynamic gesture spotting algorithm based on Dempster–Shafer Theory in the augmented reality human computer interaction.* The International Journal of Medical Robotics and Computer Assisted Surgery, 2018. **14**(5): p. e1931.
20. Masood, S., et al., *Real-time sign language gesture (word) recognition from video sequences using CNN and RNN*, in *Intelligent Engineering Informatics*. 2018, Springer. p. 623-632.
21. Nandy, A., et al. *Recognition of isolated indian sign language gesture in real time*. in *International conference on business administration and information processing*. 2010. Springer.
22. Pigou, L., et al. *Sign language recognition using convolutional neural networks*. in *European conference on computer vision*. 2014. Springer.
23. Ronchetti, F., et al., *Handshape recognition for argentinian sign language using probsom.* Journal of Computer Science & Technology, 2016. **16**.
24. Sharma, R., et al. *Communication device for differently abled people: a prototype model*. in *Proceedings of the International Conference on Data Engineering and Communication Technology*. 2017. Springer.
25. Singha, J. and K. Das, *Automatic Indian Sign Language recognition for continuous video sequence.* ADBU Journal of Engineering Technology, 2015. **2**(1).
26. Tripathi, K. and N.B.G. Nandi, *Continuous Indian sign language gesture recognition and sentence formation.* Procedia Computer Science, 2015. **54**: p. 523-531.

27. Vicars, W., *Sign language resources at LifePrint. com*. 2017.

28. Farhadi, A., D. Forsyth, and R. White. *Transfer learning in sign language*. in *2007 IEEE conference on computer vision and pattern recognition*. 2007. IEEE.

29. Fillbrandt, H., S. Akyol, and K.-F. Kraiss. *Extraction of 3D hand shape and posture from image sequences for sign language recognition*. in *2003 IEEE International SOI Conference. Proceedings (Cat. No. 03CH37443)*. 2003. IEEE.

30. Kishore, P., et al., *Selfie sign language recognition with convolutional neural networks.* International Journal of Intelligent Systems and Applications, 2018. **10**(10): p. 63.

31. Shin, H., W.J. Kim, and K.-a. Jang. *Korean sign language recognition based on image and convolution neural network*. in *Proceedings of the 2nd International Conference on Image and Graphics Processing*. 2019.

32. Buehler, P., A. Zisserman, and M. Everingham. *Learning sign language by watching TV (using weakly aligned subtitles)*. in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009. IEEE.

33. Cihan Camgoz, N., et al. *Subunets: End-to-end hand shape and continuous sign language recognition*. in *Proceedings of the IEEE international conference on computer vision*. 2017.

34. Nikam, A.S. and A.G. Ambekar. *Sign language recognition using image based hand gesture recognition techniques*. in *2016 online international conference on green engineering and technologies (IC-GET)*. 2016. IEEE.

35. Koller, O., J. Forster, and H. Ney, *Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers.* Computer Vision and Image Understanding, 2015. **141**: p. 108-125.

36. Nguyen, T.D. and S. Ranganath. *Tracking facial features under occlusions and recognizing facial expressions in sign language*. in *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*. 2008. IEEE.

37. Carreira, J. and A. Zisserman. *Quo vadis, action recognition? a new model and the kinetics dataset*. in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

38. Krizhevsky, A., I. Sutskever, and G.E. Hinton, *Imagenet classification with deep convolutional neural networks.* Advances in neural information processing systems, 2012. **25**.

39. LeCun, Y., et al., *Gradient-based learning applied to document recognition.* Proceedings of the IEEE, 1998. **86**(11): p. 2278-2324.

40. Laptev, I., et al. *Learning realistic human actions from movies*. in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. 2008. IEEE.

41. Wang, H., et al. *Evaluation of local spatio-temporal features for action recognition*. in *Bmvc 2009-british machine vision conference*. 2009. BMVA Press.

42. Toshev, A. and C. Szegedy. *Deeppose: Human pose estimation via deep neural networks*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

43. Martinez, J., M.J. Black, and J. Romero. *On human motion prediction using recurrent neural networks*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

44. Cui, R., H. Liu, and C. Zhang. *Recurrent convolutional neural networks for continuous sign language recognition by staged optimization*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

45. Camgoz, N.C., et al. *Neural sign language translation*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

46. Garcia, B. and S.A. Viesca, *Real-time American sign language recognition with convolutional neural networks.* Convolutional Neural Networks for Visual Recognition, 2016. **2**: p. 225-232.

47. Simonyan, K. and A. Zisserman, *Very deep convolutional networks for large-scale image recognition.* arXiv preprint arXiv:1409.1556, 2014.

48. Santhalingam, P.S., et al. *Body Pose and Deep Hand-shape Feature Based American Sign Language Recognition*. in *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*. 2020. IEEE.

49. Passi, K. and S. Goswami. *Real time Static Gesture Detection Using Deep Learning*. in *International Conference on Big Data Analytics*. 2019. Springer.

50. Kulkarni, V.S. and S. Lokhande, *Appearance based recognition of american sign language using gesture segmentation.* International Journal on Computer Science and Engineering, 2010. **2**(03): p. 560-565.

51. Zafrulla, Z., et al. *American sign language recognition with the kinect*. in *Proceedings of the 13th international conference on multimodal interfaces*. 2011.

52. Xue, Q., et al., *Deep forest-based monocular visual sign language recognition.* Applied Sciences, 2019. **9**(9): p. 1945.

53. Pigou, L., M. Van Herreweghe, and J. Dambre. *Gesture and sign language recognition with temporal residual networks*. in *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017.

54. Metaxas, D., M. Dilsizian, and C. Neidle. *Scalable ASL sign recognition using model-based machine learning and linguistically annotated corpora*. in *sign-lang@ LREC 2018*. 2018. European Language Resources Association (ELRA).

55. Lim, K.M., A.W. Tan, and S.C. Tan, *Block-based histogram of optical flow for isolated sign language recognition.* Journal of Visual Communication and Image Representation, 2016. **40**: p. 538-545.

56. Huang, J., et al. *Sign language recognition using 3d convolutional neural networks*. in *2015 IEEE international conference on multimedia and expo (ICME)*. 2015. IEEE.

57. Grobel, K. and M. Assan. *Isolated sign language recognition using hidden Markov models*. in *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*. 1997. IEEE.

58. Radhakrishnan, S., et al. *Cross Transferring Activity Recognition to Word Level Sign Language Detection*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

59. Athitsos, V., et al. *The american sign language lexicon video dataset*. in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 2008. IEEE.

60. Joze, H.R.V. and O. Koller, *Ms-asl: A large-scale data set and benchmark for understanding american sign language.* arXiv preprint arXiv:1812.01053, 2018.

61. Wilbur, R. and A.C. Kak, *Purdue RVL-SLLL American sign language database.* 2006.

62. *https://blog.heuritech.com/2016/02/29/a-brief-report-of-the-heuritech-deep-learning-meetup-5/*

63. Understanding-architecture-of-lstm-cell-from-scratch-with-code.2018. *https://medium.com/hackernoon/understanding-architecture-of-lstm-cell-from-scratch-with-code-8da40f0b71f4*

64. An introduction to ConvLSTM. 2019. *https://medium.com/neuronio/an-introduction-to-convlstm-55c9025563a7*