**Abstract**

Sign language is the only medium of communication between the people with dis-abilities in hearing and speaking. Even if they need to present their thoughts or ideas to any individual, they will do that using actions. It may not be understood accurately and efficiently by the individual, which may result in misunderstanding leading to greater problems.

In this project, we have aimed at converting their actions, the sign language, to text. The actions will be understood by the system, depending upon the knowledge of the actions, and then translated accordingly to the required text. The text can be easily read and thus favouring communication.

To enable the detection of gestures, we are making use of a Convolutional neural network (CNN). A CNN is highly efficient in tackling computer vision problems and is capable of detecting the desired features with a high degree of accuracy upon sufficient training.

The software also aims at saving time by encoding some signs to a text that would normally be large in size. We just need to use the desired sign and it will be converted automatically to the large text. This can also be used for security purposes, for developing one’s own Sign language.

**Acknowledgement**

I take this opportunity to express my heartfelt gratitude to all the individuals for their invaluable support and encouragement throughout the course of this project.

I am deeply thankful to **Dr. Hemlata K. Bagla, Vice Chancellor of HSNC University, Mumbai** for providing me with the opportunity to pursue my studies at this esteemed institution and for being a constant source of inspiration.

I extend my sincere gratitude to **Dr. Tejashree V. Shanbhag, Principal of K.C. College, Mumbai,** for generously providing all the resources and unwavering support that have been instrumental in successful completion of this project.

My gratitude also goes to **Dr. Shalini R. Sinha, Vice Principal and Co-ordinator of the Computer Science Department,** for her constant guidance and support.

I am indebted to my project guide **Ms. Ritika Sharma and Mr. Aditya Agarwal** for their mentorship, expertise and continuous encouragement for successful completion of this project. I also sincerely thank **Ms. Geeta N. Brijwani** for imparting their knowledge and wisdom and immense support.

I would also like to thank **Mr. Ajit Vishwakarma** for their valuable insights and contributions from an industry perspective.

I am grateful to the non-teaching staff **Mr. Ashish Gawande, Mr. Vishal Pandey, and Mr. K.B. Shukla of the Computer Science department** for their continuous support and assistance in providing lab facilities to complete the project in time.

I sincerely thank my dedicated team member **Pranav Surve** for his collaboration, hard work and shared enthusiasm that were crucial to the success of this project.

Last but not least, I want to express my heartfelt appreciation to my family and classmates for their support, understanding and encouragement throughout the journey. This project would not have been possible without their love and collective support.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr No** | **Title** | **Page Number** |
| **1** | **Introduction** | **5** |
|  | Motivation of the project | **6** |
|  | Key Features and objectives |  |
| **2** | **Literary Survey/ Related Work** | **8** |
| **3** | **Scope of Work** | **10** |
|  | Solution overview |  |
|  | Features and benefits of the proposed solution | **11** |
|  | Objectives | **7** |
| **4** | **Problem Statement Formulation** |  |
| **5** | **System Design** | 12 |
|  | Object Diagram | **14** |
|  | Activity diagram | **15** |
|  | Deployment Diagram | **16** |
| Sr No | Title | **Page Number** |
| 6 | Methodology | **19** |
|  | Explanation of modules, algorithms, and techniques used | **20** |
| 7 | Implementation and Code | **24** |
|  | Code snippets or code-related discussions |  |
| 3 | Testing | **28** |
| 4 | Experimental Setup | **30** |
|  | Information on the setup used for experiments or testing | **31** |
|  | Describe software used for testing | **32** |
| 5 | Results and Discussion | **33** |
|  | Presentation of results | **33** |
|  | In-depth discussion and analysis of results | **34** |
| 6 | Conclusion | **37** |
|  | Summary of the results |  |
|  | Concluding remarks |  |
| 7 | Future Scope and Potential Advancements | **38** |
| 8 | References | **39** |
| 9 | Self-attested copy of Plagiarism Report from any open-source tool | **40** |

**Introduction**

Today, there are almost 2 million people classified as Deaf and Dumb. They have great difficulty in communicating with each other and with other individuals as the only means of communication is sign language. They need to learn this sign language. It is extremely difficult for a person who is unaware of this sign language to understand and decode their actions.

It is impossible to identify anything without it’s prior knowledge. Even for computers, they need to have information in their memory to identify and provide data related to any object. Now one particular object may differ from another similar type of object in shape, size, orientation or even visual effects may differ. But all the different forms of 1 type of object must be classified in the same category.

Making this our aim in abstracting, modifying, analyse and identify the various signals used by the Deaf and Mute to communicate, we have developed this model. The major technologies used are IMAGE PROCESSING and CNN (Convolutional Neural Network)

A number of images of some gestures are taken and processed to make the dataset. The CNN model is then trained using Keras on these captured and modified images. The signs to be translated are then fed to the software which matches it with the existing images and classifies it.

Sign language translation is one of the amongst most growing line of research nowadays and its miles the maximum natural manner of communication for the humans with hearing impairments. A hand gesture recognition gadget can offer an opportunity for deaf people to talk with vocal humans without the need of an interpreter. The system is built for the automated conversion of ASL into textual content and speech

**Motivation**

It is very difficult for the deaf people to communicate with the hearing person and there are not many options available to help them. And all of the alternatives have some major flaws. Interpreters are not usually available and are expensive. Pen and paper is also not a good idea, it is uncomfortable, messy and even time consuming, both for the deaf and the hearing person.

With the evolution of IoT, everything around is getting automated. The demand for Machine Learning and it’s applications is very high. The accuracy and efficiency of any algorithm and the model developed must be very high to make it useful. The knowledge of Machine Learning thus becomes very important.

In the era of Machine Learning where everything is getting automated, the need for an Interpreter to translate Sign language to text is just waste of resources. The classification of objects, object detection and image processing plays a very vital role. Thus, the main aim is to bridge the gap between the normal and the deaf-and mute individuals by providing an automatic translation system.

The key aim here is to detect the sign in the video sequence and translate it into text and speech that people can understand. Normal people have a hard time understanding hearing-impaired peoples language , so a system that understands signs and gesture and relays information to normal people is needed

**Objectives**

Deaf people do not have that many options for communicating with a hearing person, and all of the alternatives have some major flaws. Interpreters are not available easily, and also can be expensive.

Affordable and always available interpreter services are in huge demand in the deaf community. Every day thousands of local businesses around the globe face problems with providing their services to deaf. According to the National Deaf Association (NAD), 18 million people are estimated to be deaf in India.

The main objective is to design a software that will help drive inclusivity at the workplace by removing communication barriers between the disabled and able. The 5 new application can find use in a B2B setting, where businesses who want to employ deaf and mute employees can use it to convey employee messages to the end consumer.

An easy to use innovative digital translator that is compellingly fast, easy, comfortable and economical is the need of the hour.

Having to communicate between deaf people and normal public has become a difficult task now days and to implement a such as the society lacks a good translator for it and having an app for it in our mobile phones is like having a dream at day.

Having proposed something great for the deaf community or hearing aid community by providing an app for the communication. But making an app for it is no simple task at it requires lot of efforts like memory utilization and a perfectly fined design to implement a such. What their application does is that they take a picture of a sign gesture and later converts is to a meaningful word.

At first, they have compared the gesture using histogram that has been related to the sample test and moreover samples that are obliged to BRIEF to basically reduce the weight on the CPU and its time. They have explained a process on which on their app, its very easy to add up a gesture and store it in their database for further and expand detection set. So lastly, they came strong with having an app as a translator instead of several applications that are being used lately by users.

**Related Work / Literature Review**

Text classification is one of the most used application of Machine Learning. It is used to automatically assign predefined categories(labels) to text documents. The purpose of text classification is to organize conceptually a large collection of documents.

It has become more relevant with exponential growth of the data, with wide applicability in real world applications. There are apps available in the market for converting signs into text, but all of them use old technology are slow to operate. Messengers and texting are used, but the problem is still not solved, which is translation , and do not offer neat, confident and comfortable way to communicate.

Google has developed an app called GnoSys [2], that uses neural network and computer vision to recognise the video of sign language speaker ,and then smart algorithms translate it into speech.

A glove was developed at the University of California, San Diego ,in July 2017 which can convert the 26 letters of American Sign Language (ASL) into text on a smartphone or computer screen .But it was limited only to 26 letters of English alphabet.

Start-ups working with NAD (National Deaf Association) have collaborated with India Accelerator to gather sign language data for India. All the available apps have limited vocabulary of signs and hence research is going on in this field.

This paper proposes a framework for Sign Language Recognition (SLR) based on Hidden Markov Models (HMMs). The proposed framework utilizes trajectories and hand-shape features of sign videos to translate sign language into text or speech. The authors introduce a new trajectory feature called "enhanced shape context" to capture spatio-temporal information and fetch hand regions using Kinect mapping functions, which are then described by HOG (pre-processed by PCA).

Over the past few decades, lots of research is going on in gesture recognition as it can be used in various application domains like smart home applications, Human Computer Interaction (HCI), gaming, medical systems, etc. Solutions proposed by different researchers are of two types: solutions based on Hardware and solutions based on Soft- ware. Solutions based on hardware include gesture recognition using gloves, wrist bands, etc. These hardware solutions contain sensors as they are necessary to track hand movements. Google has developed wristbands which are able to recognize gestures by tracking hand movements and user is able to hear the recognized word/sentence through a mobile device as the mobile device is connected to the wristband

In another proposed method, data captured by gloves is sent to the neural network and is processed for classification

InerTouchHand System is proposed for Human Machine Interaction (HMI) and uses distributed inertial sensors, vibro tactile simulators

Glove based systems may give wrong results as time goes on depending on sensor quality. Software based solutions include gesture recognition using Support Vector Machines(SVM), Neural Networks(NN), Hidden Markov Models (HMMs), etc. Software based solutions require image processing before classifying gesture images. Amazon Alexa also is able to respond to sign language gestures

Histogram of Gradients (HOG) and Scale Invariant Feature Transform (SIFT) features are drawn out from the images of hand gestures and are fed to Support Vector Machines (SVM) for training which is then used to classify new hand gesture images [9]. They used a dataset containing images of different orientations for accurate classification. HOG along with Local Binary Pattern (LBP) features are used together to classify hand gestures and this system attained an accuracy of 92%

**Scope of Work**

**Scope**

This system is primarily intended for making a Interpreter. This will have applications in Business who want to employ deaf and mute employees can use it to convey employee messages to the end consumer. It will be used majorly by the deaf and mute to communicate.

The applications can further be extended to security purposes, by developing a sign language of your own. And even observing and analysing any suspicious actions.

**Purpose**

The aim of this document is to provide a detailed description of the translator of Sign language to text. It will cover the applications and features of the system, the interfaces of the system, what the system is expected to do, the constraints that the project will work under and how it behaves in response to external stimuli. This is intended for both the developers and the users of this system

The objectives of the project

Image Acquisition

The gestures are captured through the web camera. This OpenCV video stream is used to capture the entire signing duration. The frames are extracted from the stream and are processed as grayscale images with the dimension of 50\*50. This dimension is consistent throughout the project as the entire dataset is sized exactly the same.

Hand Region Segmentation & Hand Detection and Tracking The captured images are scanned for hand gestures. This is a part of preprocessing before the image is fed to the model to obtain the prediction. The segments containing gestures are made more pronounced. This increases the chances of prediction by many folds.

Hand Posture Recognition

The preprocessed images are fed to the keras CNN model. The model that has already been trained generates the predicted label. All the gesture labels are assigned with a probability. The label with the highest probability is treated

To be the predicted label.

Display as Text & Speech

The model accumulates the recognized gesture to words. The recognized words are converted into the corresponding speech using the pyttsx3 library. The text to speech result is a simple work around but is an invaluable feature as it gives a feel of an actual verbal conversation.

**Features and benefits of the proposed solution.**

There are two methods for recognising sign language: glove-based recognition and vision-based recognition .This system employs a vision-based recognition mechanism that is non-invasive.

There are two methods for achieving vision-based recognition. The entry point file will be manually run on any modern browser, notably the most recent updated Chrome browser classifier.

This system includes a camera unit for recording and generating, as well as training for hearing and speech challenged people's gestures. The pictures scanned from the raw videos are fed into the system with the correct environmental configuration

To ensure that all of the movies are similar in size, the picture frames are adjusted. Feature extraction and classification are done using Convolution Neural Networks. To access web camera, the programme will display a notification. We can read American sign language using camera and write the character on a screen after allowing webcams access . Using unique hand movements any two words can be separated by space

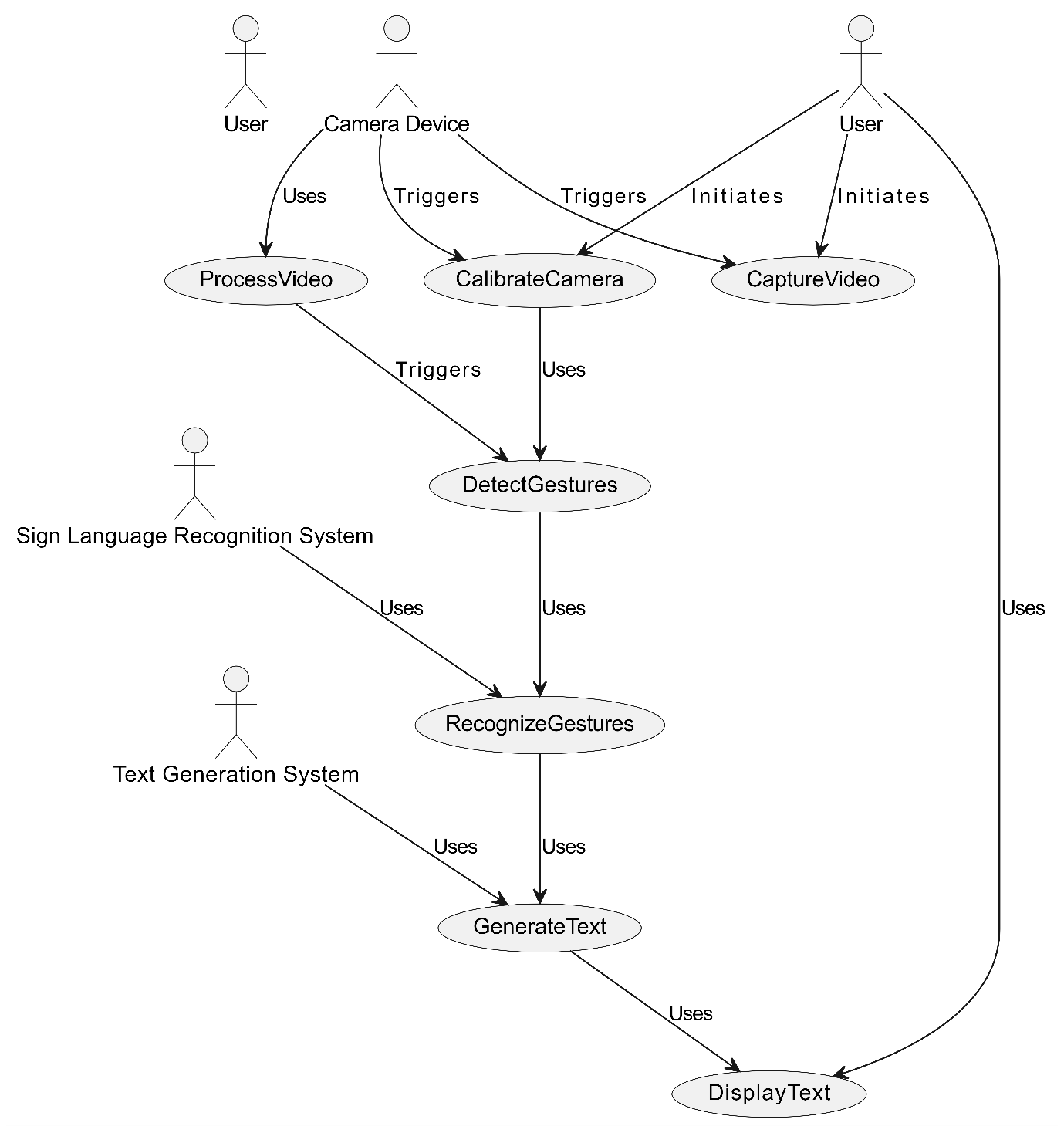
Building the Model The CNN is a type of Artificial Neural Network that is used in Image recognition and processing. It detects specific features in the images and then classifies them accordingly based on the presence of those features.

In our model we have 5 layers in total including 3 Convolutional layers and 2 fully connected Dense layers. The first CNN layer consists of 16 neurons, the second layer consists of 32 neurons and the third layer consists of 64 neurons.

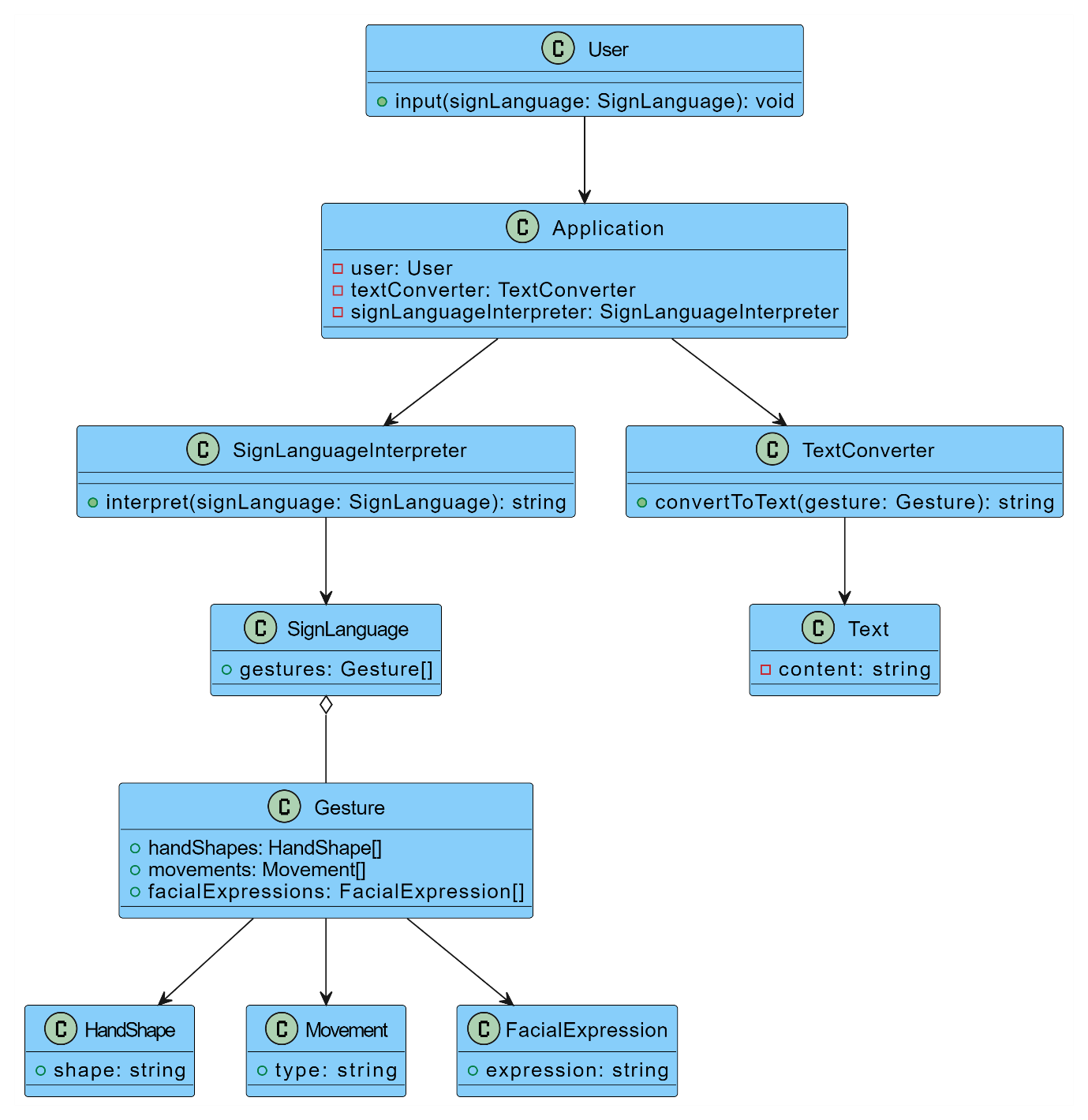
The first dense layers consists of 128 neurons and uses the Rectifier function as the Activation function. The second dense layer uses Softmax function as the Activation function.

**System Design**

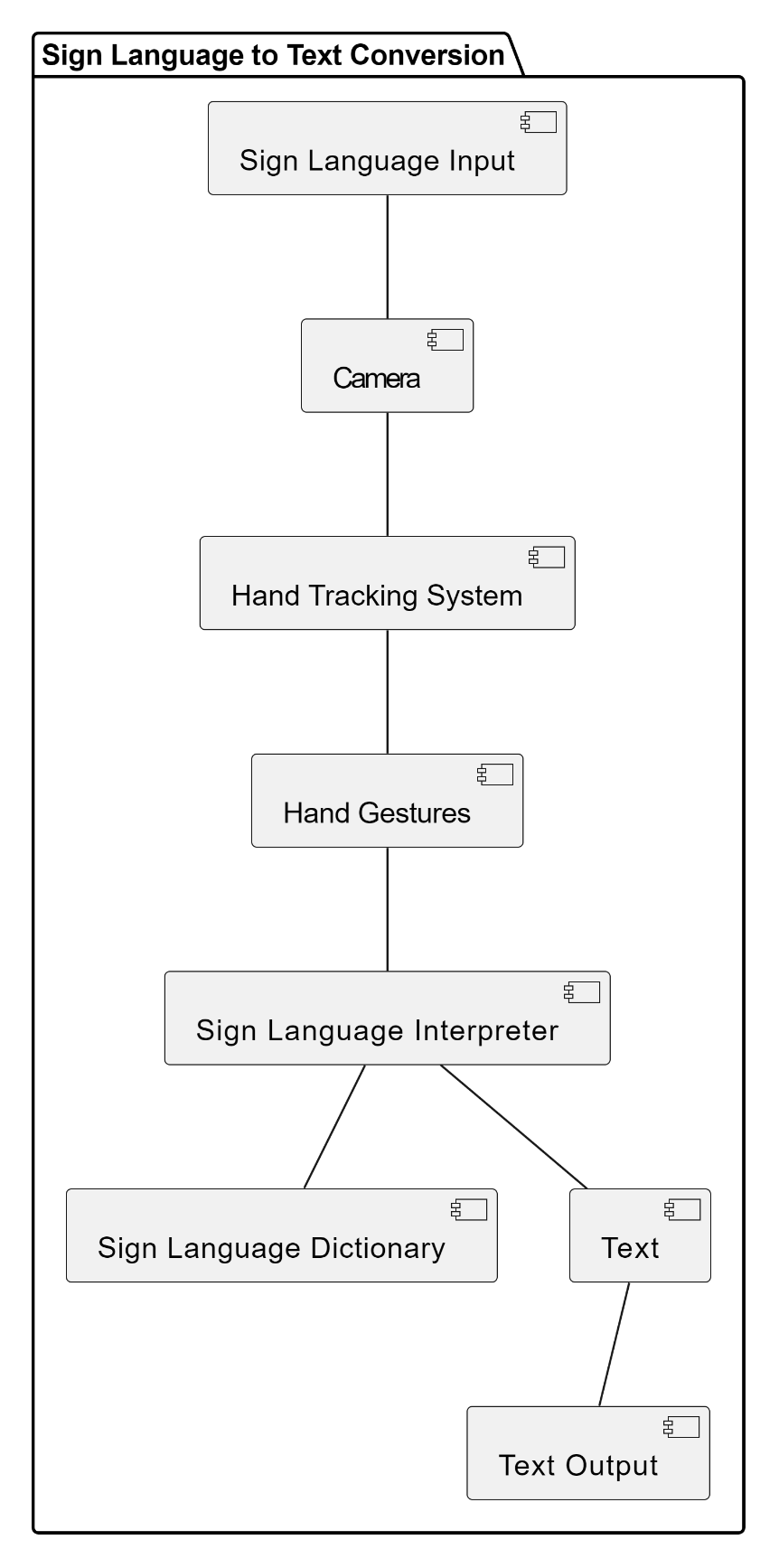
Use Case Diagram



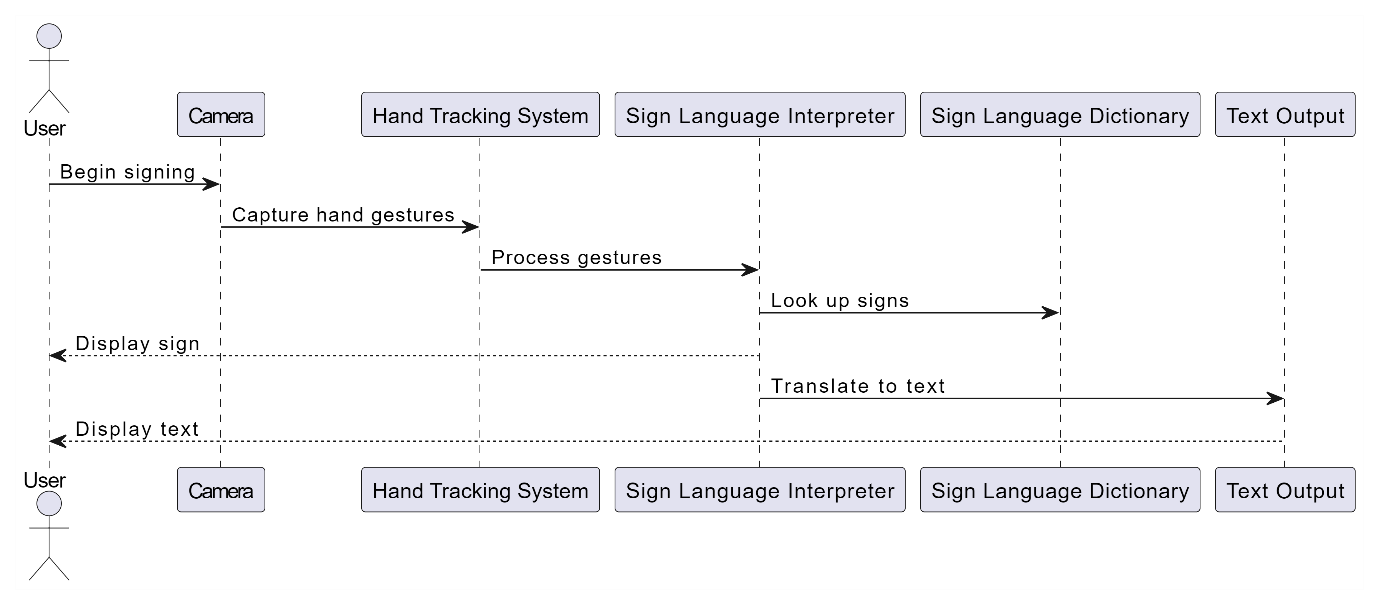
**Class Diagram**



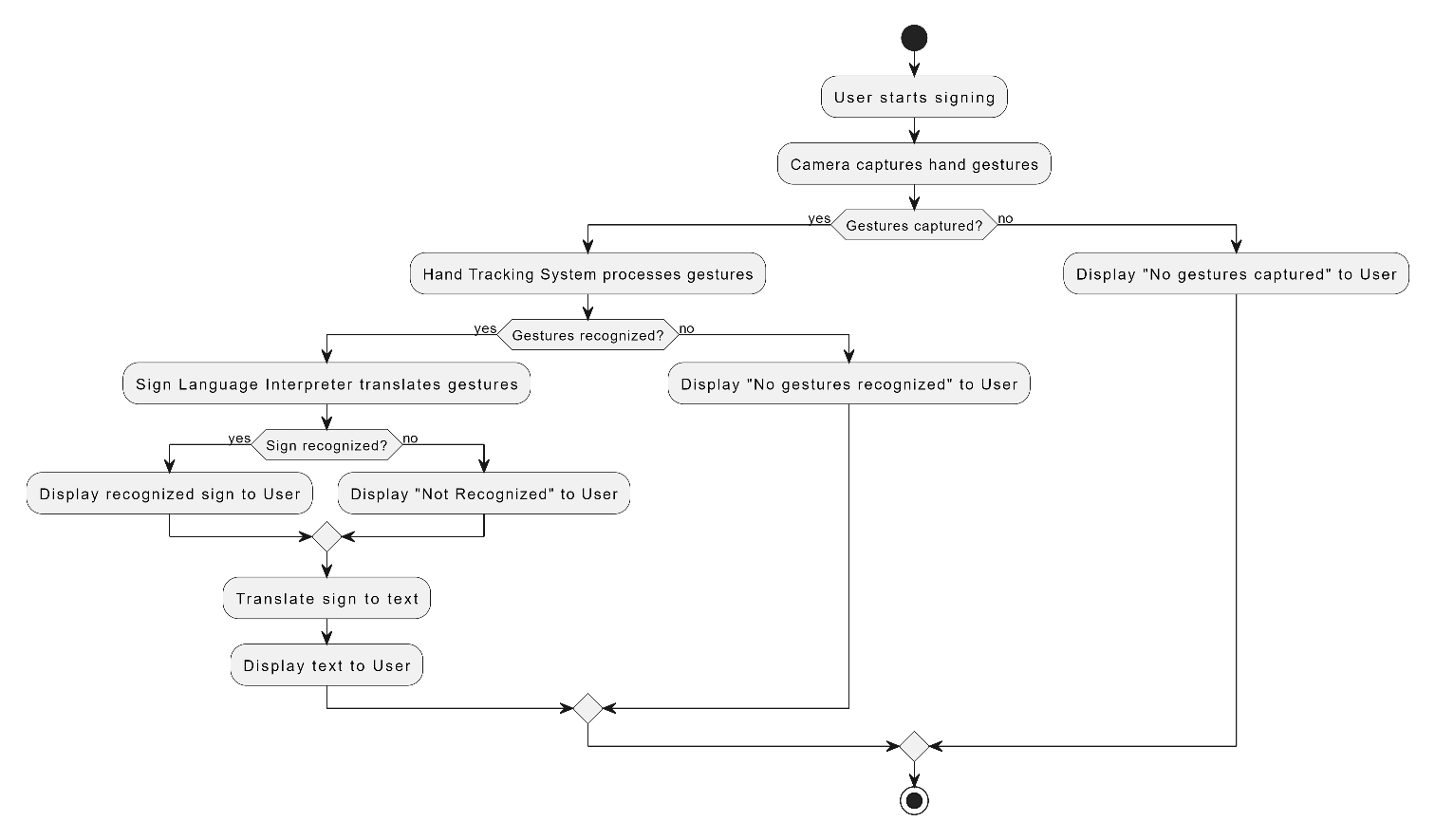
Object Diagram



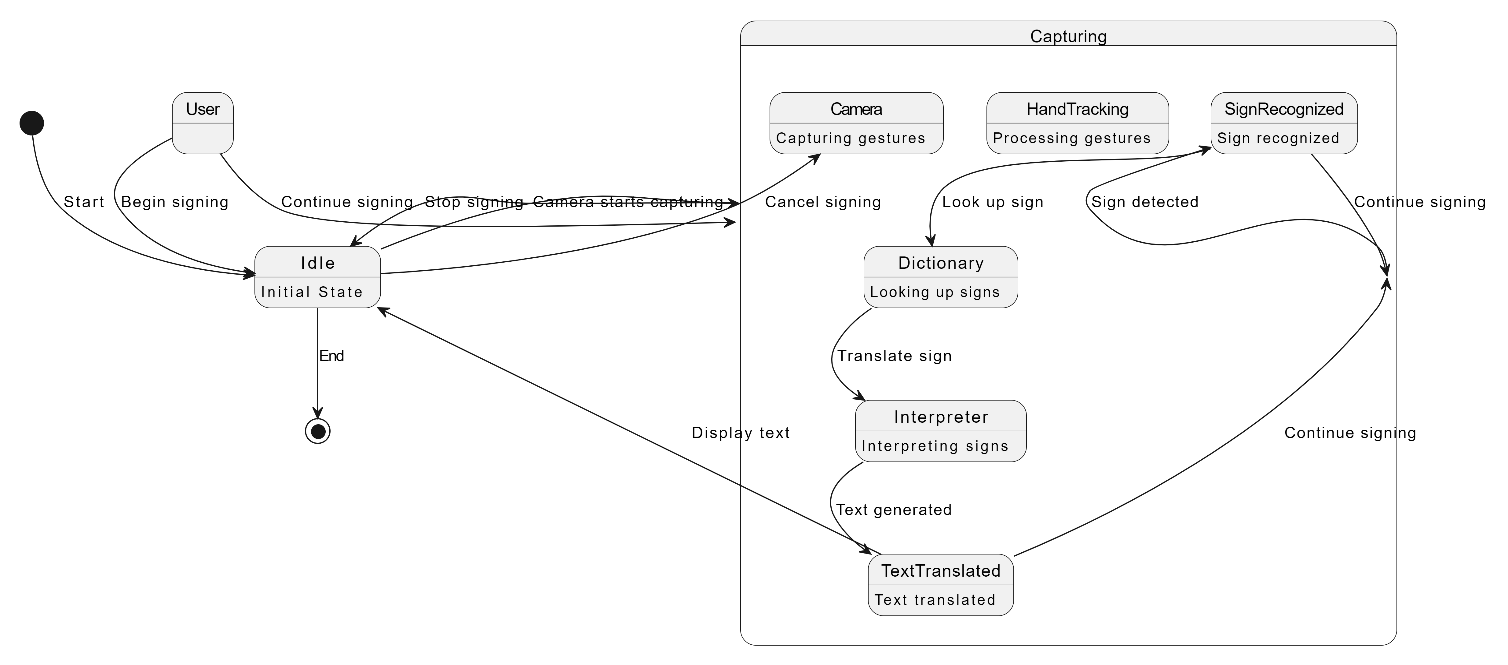
Sequence Diagram



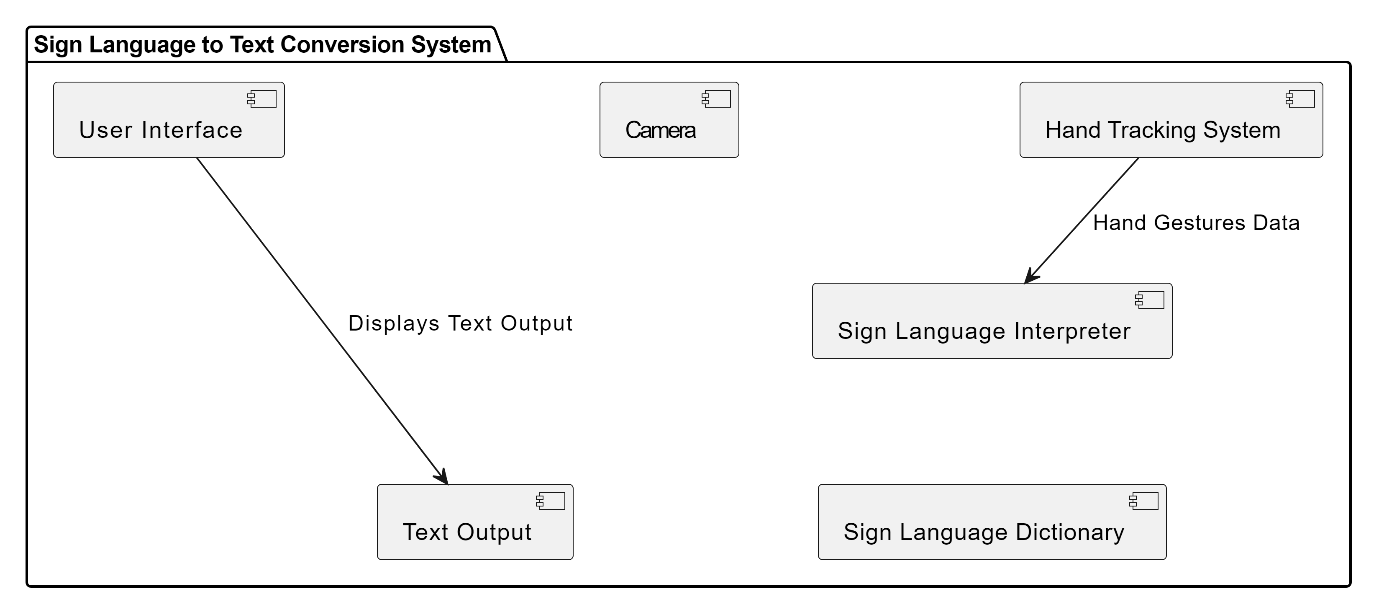
Activity Diagram



State Machine

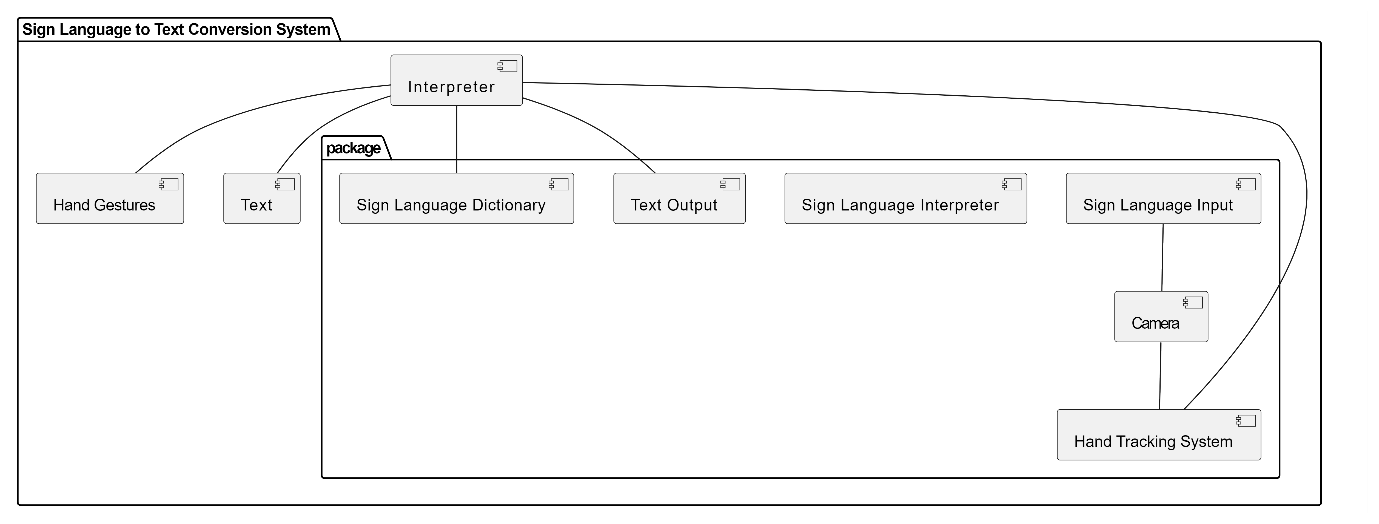


Comp Diagram



Deployment Diagram





**Methodology**

The system is a vision-based approach. All signs are represented with bare hands and so it eliminates the problem of using any artificial devices for interaction.

**A.** Explanation of modules, algorithms, and techniques used.6803

**Data Set Generation:**

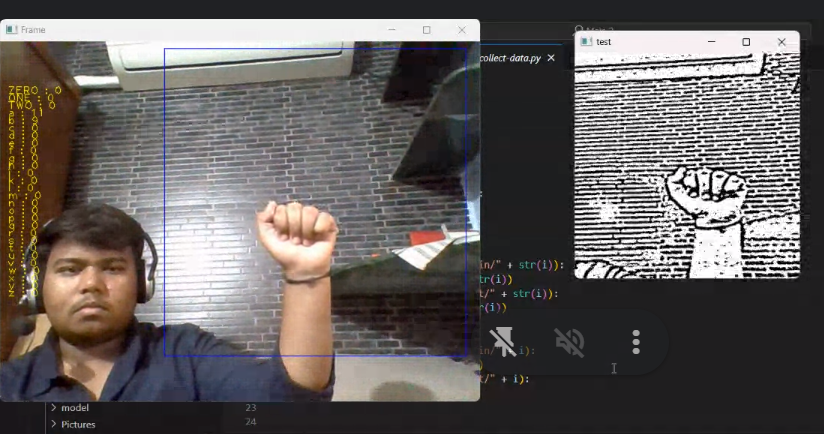
For the project we tried to find already made datasets but we couldn’t find dataset in the form of raw images that matched our requirements. All we could find were the datasets in the form of RGB values. Hence, we decided to create our own data set. Steps we followed to create our da

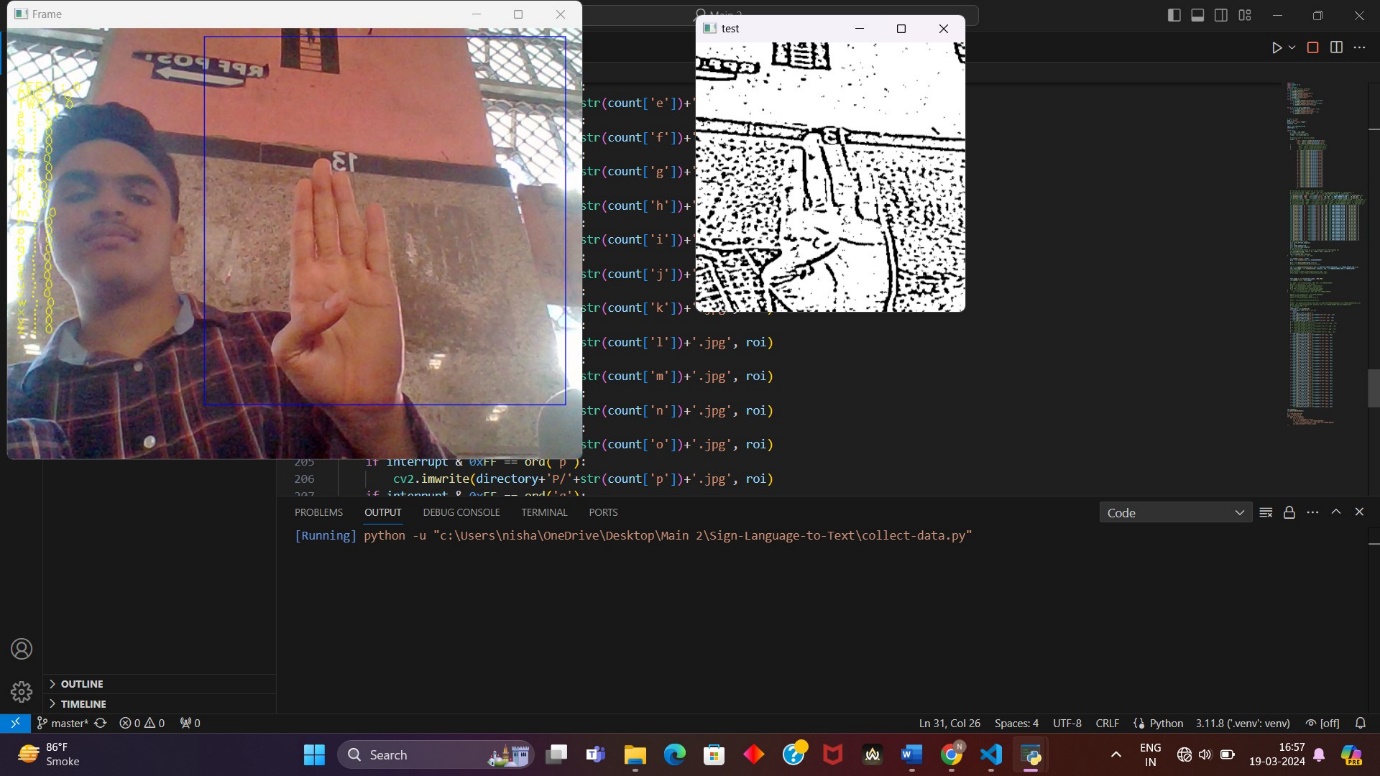
+ta set are as follows.

We used Open computer vision (OpenCV) library in order to produce our dataset.

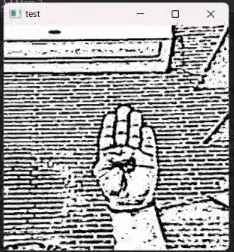
Firstly, we captured around 800 images of each of the symbol in ASL (American Sign Language) for training purposes and around 200 images per symbol for testing purpose.

First, we capture each frame shown by the webcam of our machine. In each frame we define a Region Of Interest (ROI) which is denoted by a blue bounded square as shown in the image below:





Then, we apply Gaussian Blur Filter to our image which helps us extract various features of our image. The image, after applying Gaussian Blur, looks as follows:



**Algorithm Layer 1:**

**1.** Apply Gaussian Blur filter and threshold to the frame taken with openCV to get the processed image after feature extraction.

**2.** This processed image is passed to the CNN model for prediction and if a letter is detected for more than 50 frames then the letter is printed and taken into consideration for forming the word.

**3.** Space between the words is considered using the blank symbol.

**Algorithm Layer 2:**

**1.** We detect various sets of symbols which show similar results on getting detected.

**2.** We then classify between those sets using classifiers made for those sets only.

**Layer 1:**

l **CNN Model:**

**1.** **1st Convolution Layer:** The input picture has resolution of 128x128 pixels. It is first processed in the first convolutional layer using 32 filter weights (3x3 pixels each). This will result in a 126X126 pixel image, one for each Filter-weights.

**2.** **1st Pooling Layer:** The pictures are down sampled using max pooling of 2x2 i.e we keep the highest value in the 2x2 square of array. Therefore, our picture is down sampled to 63x63 pixels.

**3.** **2nd Convolution Layer:** Now, these 63 x 63 from the output of the first pooling layer is served as an input to the second convolutional layer. It is processed in the second convolutional layer using 32 filter weights (3x3 pixels each). This will result in a 60 x 60 pixel image.

**4.** **2nd Pooling Layer:** The resulting images are down sampled again using max pool of 2x2 and is reduced to 30 x 30 resolution of images.

**5.** **1st Densely Connected Layer:** Now these images are used as an input to a fully connected layer with 128 neurons and the output from the second convolutional layer is reshaped to an array of 30x30x32 =28800 values. The input to this layer is an array of 28800 values. The output of these layer is fed to the 2nd Densely Connected Layer. We are using a dropout layer of value 0.5 to avoid overfitting.

**6.** **2nd Densely Connected Layer:** Now the output from the 1st Densely Connected Layer is used as an input to a fully connected layer with 96 neurons.

**7.** **Final layer:** The output of the 2nd Densely Connected Layer serves as an input for the final layer which will have the number of neurons as the number of classes we are classifying (alphabets + blank symbol).

· **Activation Function:**

We have used ReLU (Rectified Linear Unit) in each of the layers (convolutional as well as fully connected neurons).

ReLU calculates max(x,0) for each input pixel. This adds nonlinearity to the formula and helps to learn more complicated features. It helps in removing the vanishing gradient problem and speeding up the training by reducing the computation time.

· **Pooling Layer:**

We apply **Max** pooling to the input image with a pool size of (2, 2) with ReLU activation function. This reduces the amount of parameters thus lessening the computation cost and reduces overfitting.

· **Dropout Layers:**

The problem of overfitting, where after training, the weights of the network are so tuned to the training examples they are given that the network doesn’t perform well when given new examples. This layer “drops out” a random set of activations in that layer by setting them to zero. The network should be able to provide the right classification or output for a specific example even if some of the activations are dropped out [5].

· **Optimizer:**

We have used Adam optimizer for updating the model in response to the output of the loss function.

Adam optimizer combines the advantages of two extensions of two stochastic gradient descent algorithms namely adaptive gradient algorithm (ADA GRAD) and root mean square propagation (RMSProp).

**Layer 2:**

We are using two layers of algorithms to verify and predict symbols which are more similar to each other so that we can get us close as we can get to detect the symbol shown. In our testing we found that following symbols were not showing properly and were giving other symbols also:

**1.** **For D : R and U**

**2.** **For U : D and R**

**3.** **For I : T, D, K and I**

**4.** **For S : M and N**

So, to handle above cases we made three different classifiers for classifying these sets:

**1.** **{D, R, U}**

**2.** **{T, K, D, I}**

**3.** **{S, M, N}**

**5.3**  **Finger Spelling Sentence Formation Implementation:**

**1.** Whenever the count of a letter detected exceeds a specific value and no other letter is close to it by a threshold, we print the letter and add it to the current string (In our code we kept the value as 50 and difference threshold as 20).

**2.** Otherwise, we clear the current dictionary which has the count of detections of present symbol to avoid the probability of a wrong letter getting predicted.

**3.** Whenever the count of a blank (plain background) detected exceeds a specific value and if the current buffer is empty no spaces are detected.

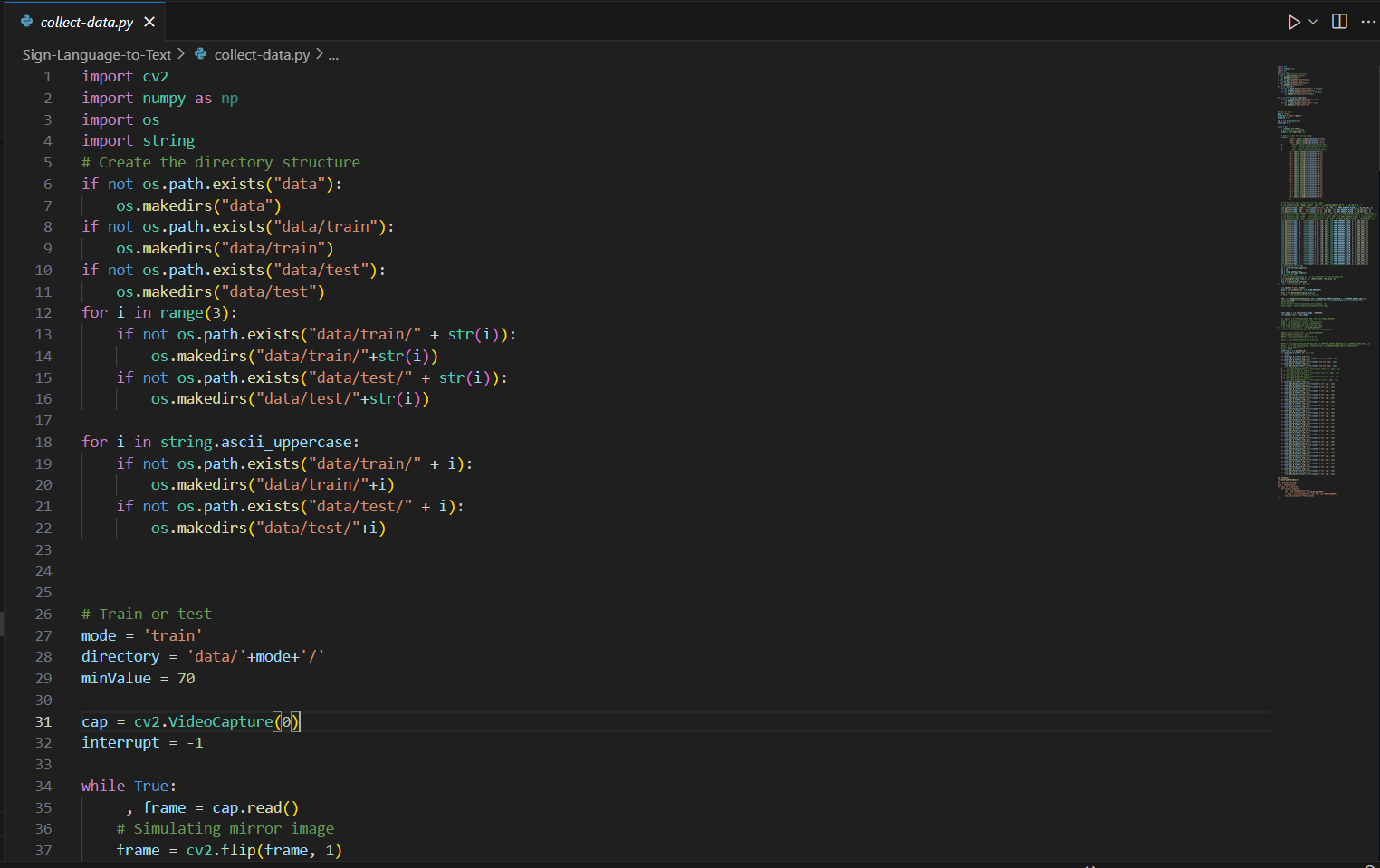
**4.** In other case it predicts the end of word by printing a space and the current gets appended to the sentence below.

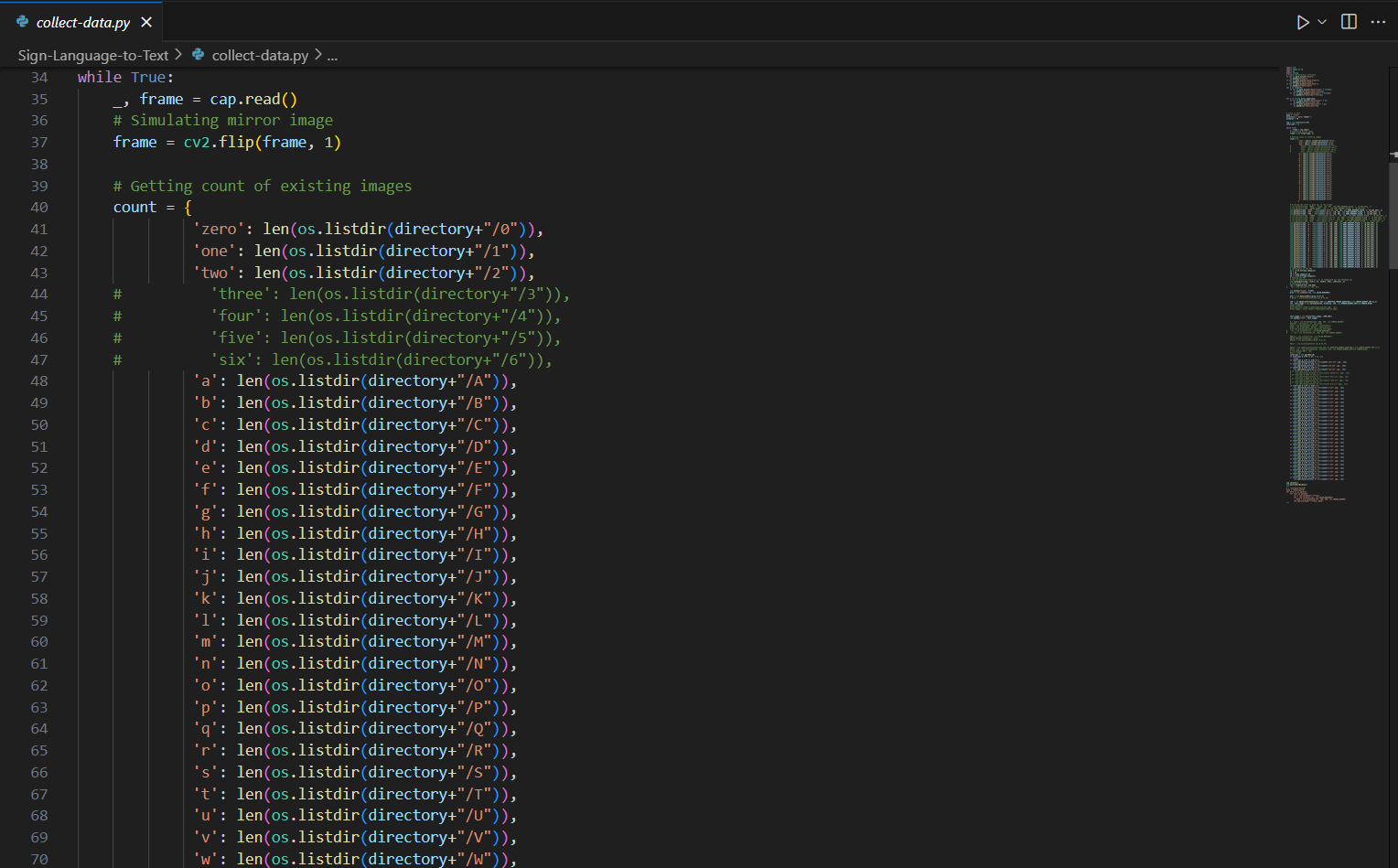
**5.4**  **AutoCorrect Feature:**

A python library **Hunspell\_suggest** is used to suggest correct alternatives for each (incorrect) input word and we display a set of words matching the current word in which the user can select a word to append it to the current sentence. This helps in reducing mistakes committed in spellings and assists in predicting complex words.

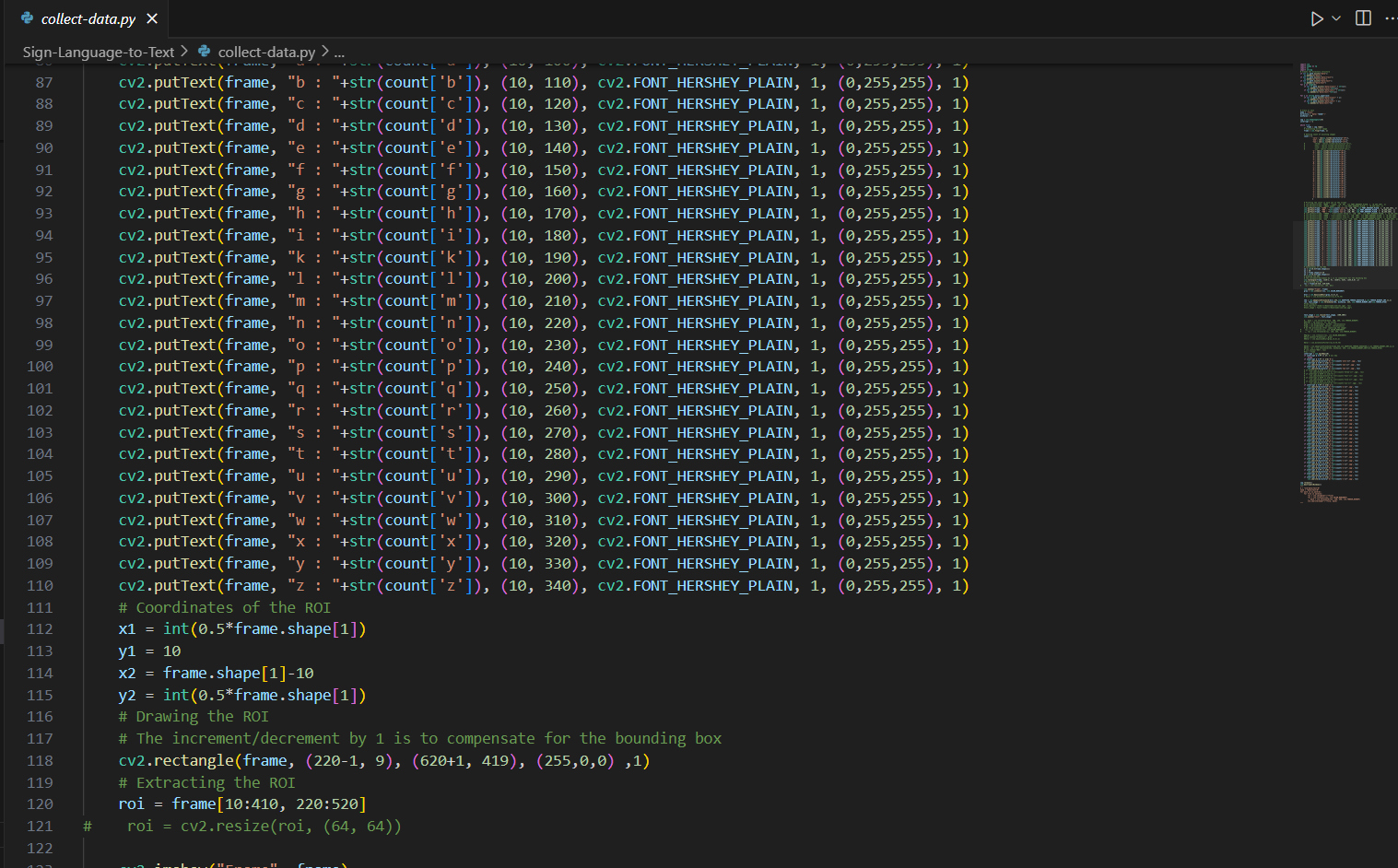
**Implementation and code**

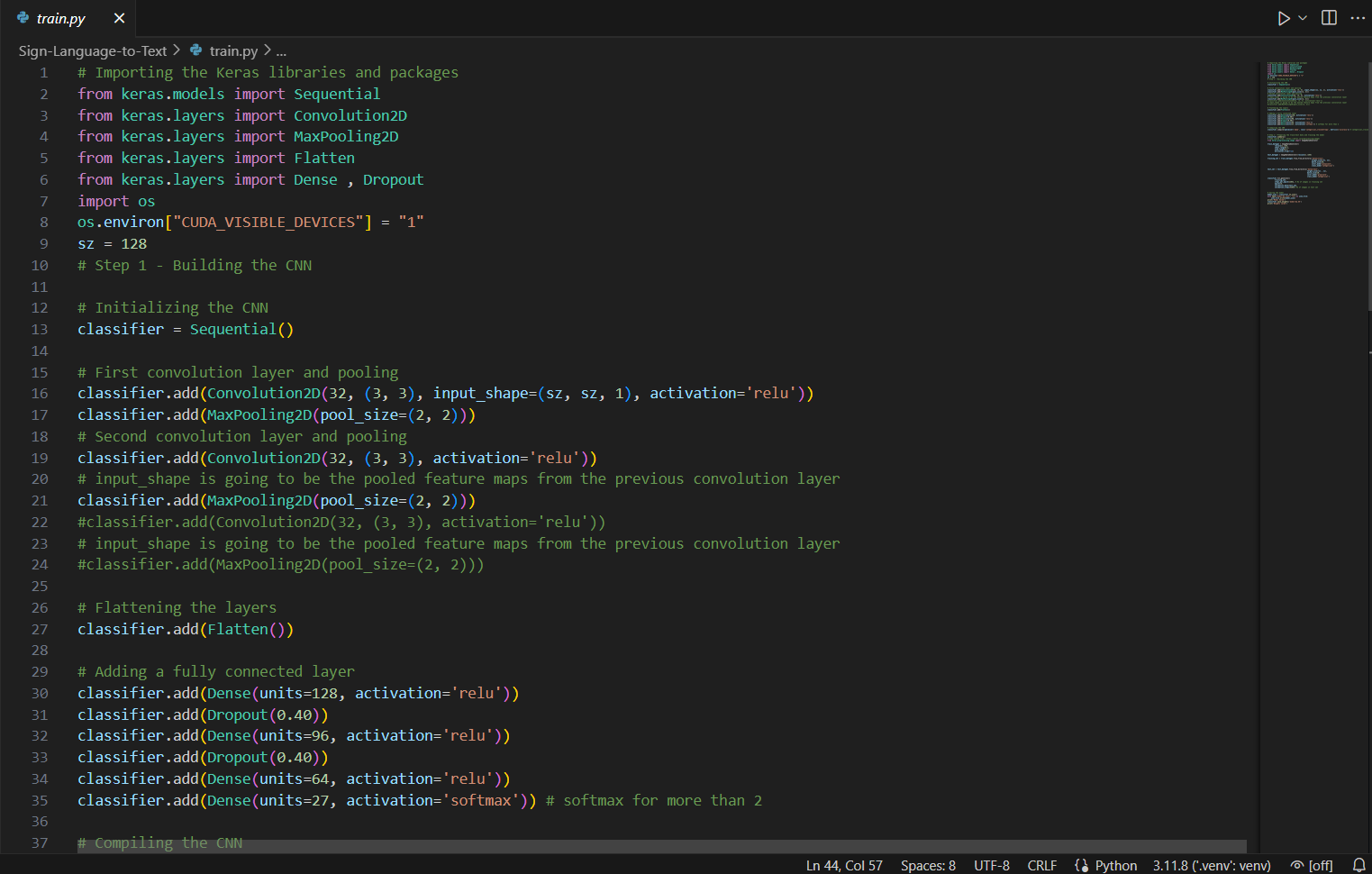
The first step in building this project was of creating the folders for storing the training and testing data



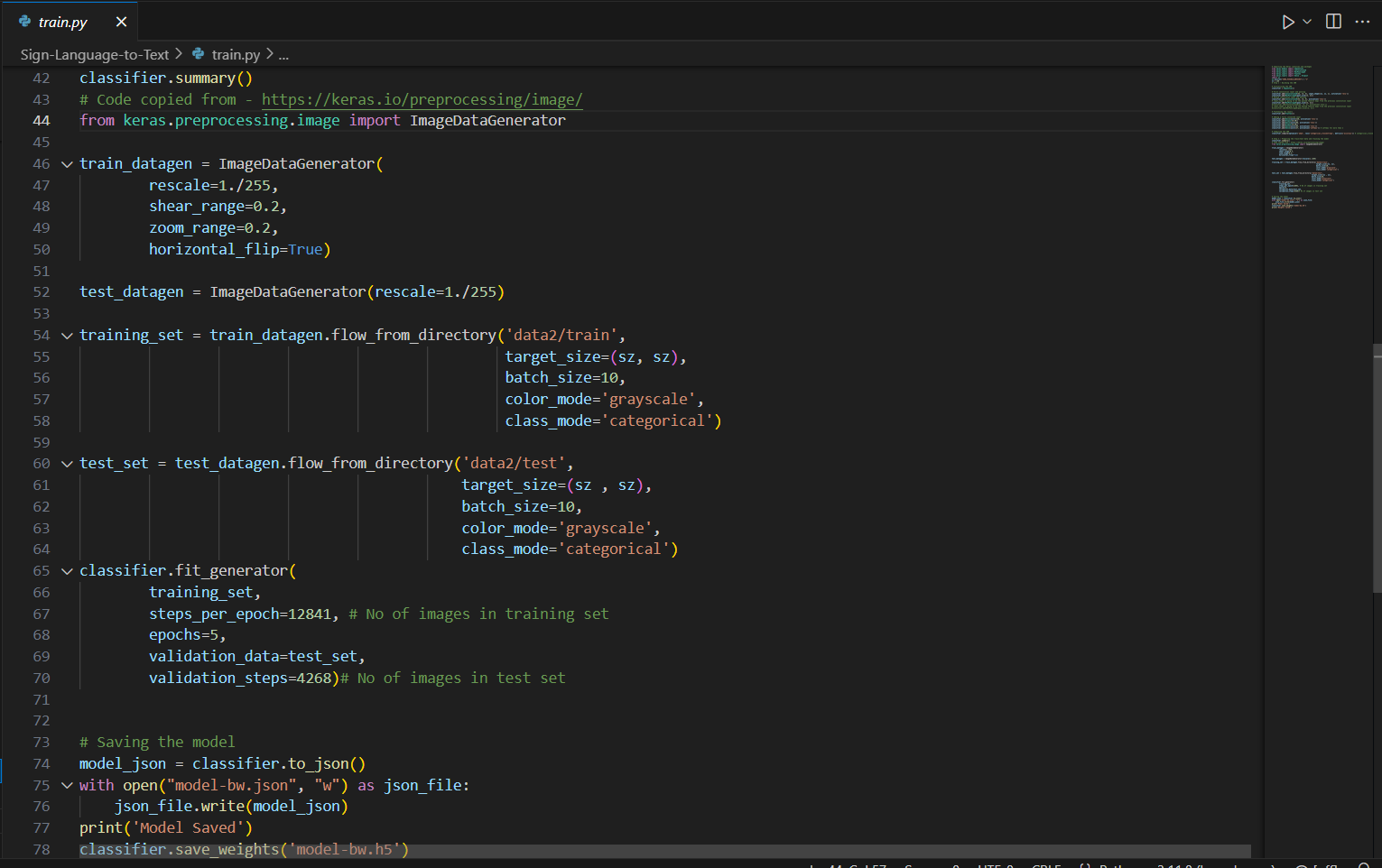


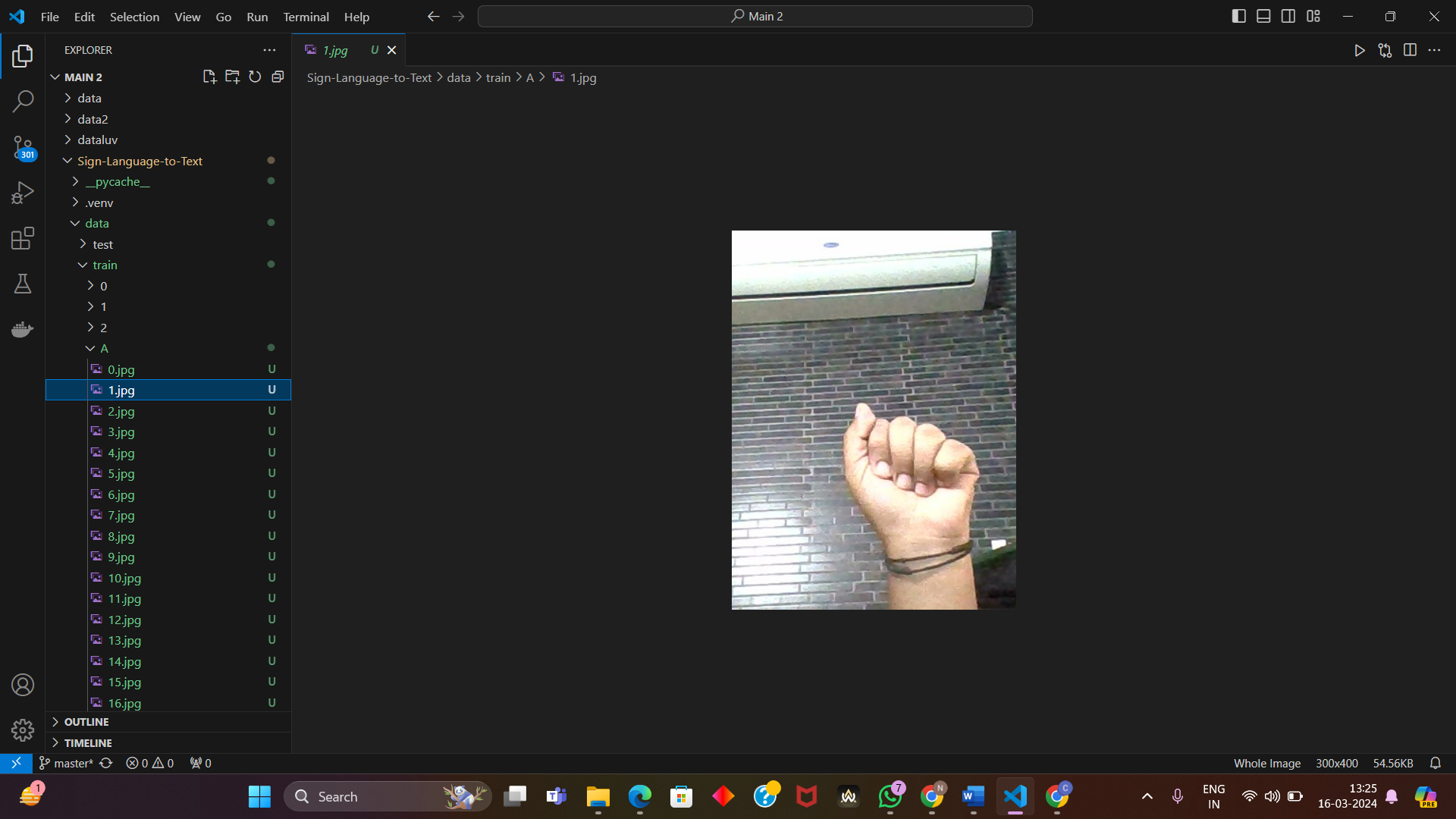
The second step after folder creation is of creating the training and testing data set





After creation of training and testing data the third step is creating a model for training. Here we have used Convolutional Neural Network(CNN) for building this model





**Testing**

While testing the applications I found out that some of the symbol predictions were coming out wrong.

So, I used two layers of algorithms to verify and predict symbols which are more similar to each other so that I can get close as I can to detect the symbol shown.

In my testing the following symbols were not showing properly and were giving output as other symbols :

For D : R and U

For U : D and R

For I : T, D, K and I

For S : M and N

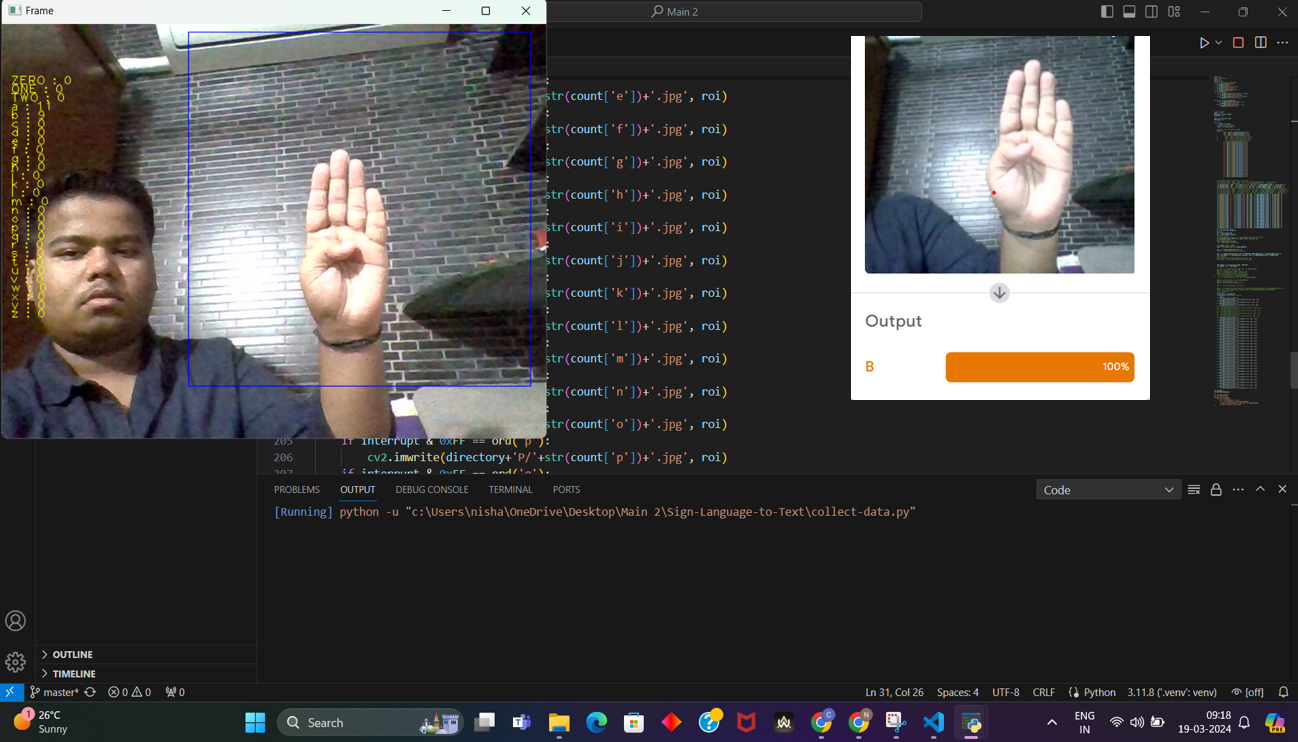
So, to handle above cases I made three different classifiers for classifying these sets:

{D, R, U}

{T, K, D, I}

{S, M, N}

The GUI(Graphical User Interface) of the application is as shown below :



**Experimental setup**

A .Information on the setup used for experiments or testing

B .Describe the software used for testing

**Environmental Setup**

***-Local Machine***

***Mediapipe***

MediaPipe is a Framework for building machine learning pipelines for processing time-series data like video, audio, etc. This cross-platform Framework works on Desktop/Server, Android, iOS, and embedded devices like Raspberry Pi and Jetson Nano.

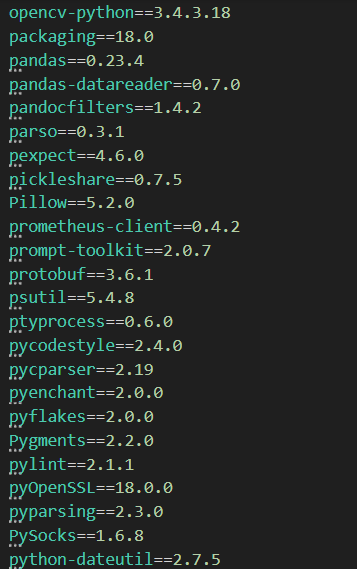
MediaPipe powers revolutionary products and services we use daily. Unlike power-hungry machine learning Frameworks, MediaPipe requires minimal resources. It is so tiny and efficient that even embedded IoT devices can run it. In 2019, MediaPipe opened up a new world of opportunity for researchers and developers following its public release.

***TensorFlow***

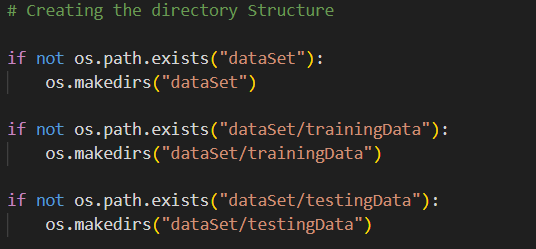
*TensorFlow is an open-source machine learning library developed by google. TensorFlow is used to build and train deep learning models as it facilities the creation of computational graphs and efficient execution on various hardware platforms.*

*TensorFlow is basically a software library for numerical computation using data flow graphs where nodes in the graph represent mathematical operation*

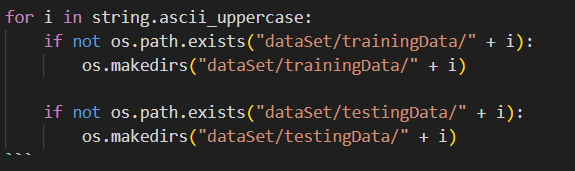
*Edges in the graph represent the multidemsional data arrays communicated between them*

****

**Creating directory structure**

****

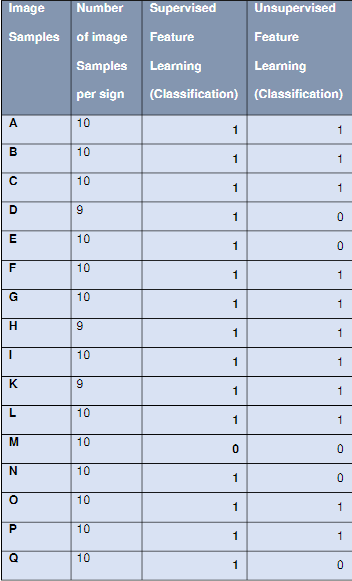
**Making folders from A-Z in the training and testing data folders respectively**

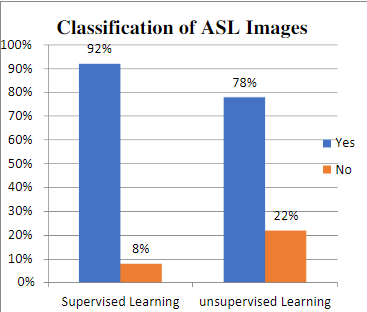


**Results and Discussions**

1. Presentation of results

Result of classification





1. In Depth discussion and analysis of results

We have achieved an accuracy of **95.8%** in our model using only layer 1 of our algorithm, and using the combination of **layer 1 and layer 2** we achieve an accuracy of **98.0%**, which is a better accuracy then most of the current research papers on American sign language.

Most of the research papers focus on using devices like Kinect for hand detection.

In [7] they build a recognition system for Flemish sign language using convolutional neural networks and Kinect and achieve an error rate of **2.5%.**

In [8] a recognition model is built using hidden Markov model classifier and a vocabulary of 30 words and they achieve an error rate of **10.90%**.

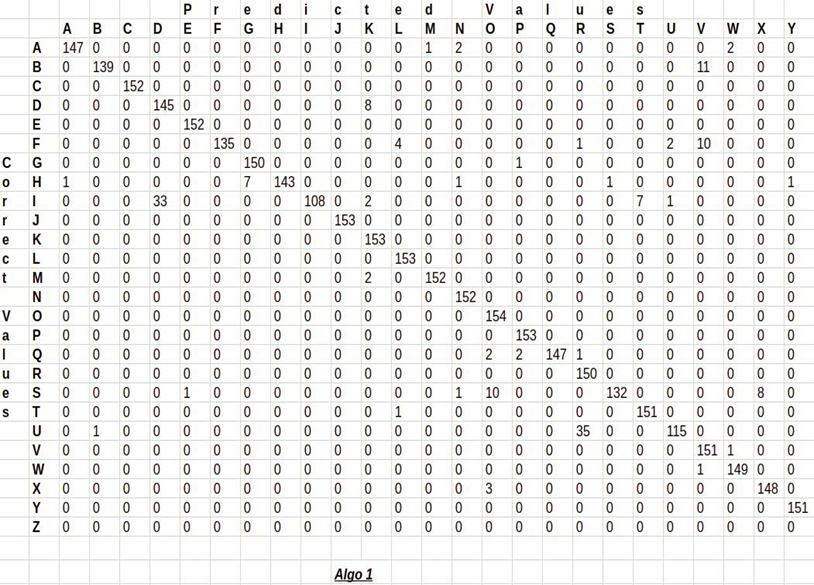
In [9] they achieve an average accuracy of **86%** for 41 static gestures in Japanese sign language.

Using depth sensors map [10] achieved an accuracy of **99.99%** for observed signers and **83.58%** and **85.49%** for new signers.

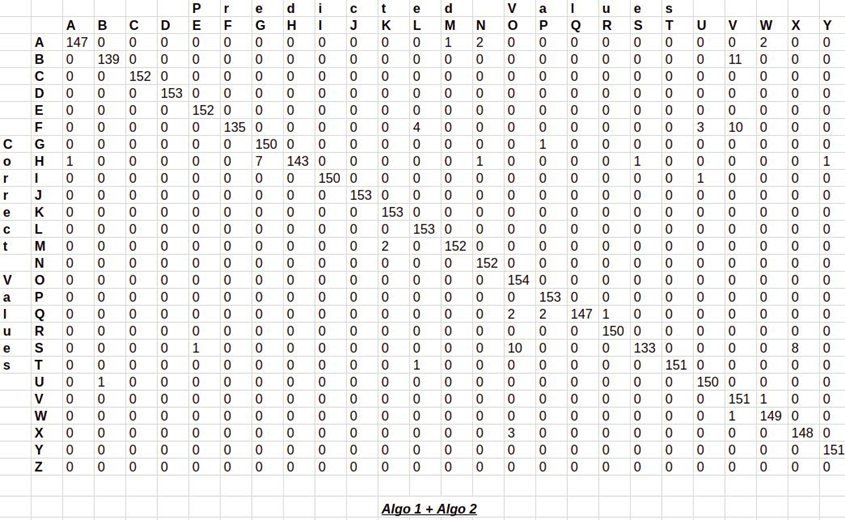
They also used CNN for their recognition system. One thing should be noted that our model doesn’t uses any background subtraction algorithm whiles some of the models present above do that.

So, once we try to implement background subtraction in our project the accuracies may vary. On the other hand, most of the above projects use Kinect devices but our main aim was to create a project which can be used with readily available resources. A sensor like Kinect not only isn’t readily available but also is expensive for most of audience to buy and our model uses a normal webcam of the laptop hence it is great plus point.

Below are the confusion matrices for our results



**Figure 11**

****

**Conclusion**

1. Summary of the results

In this report, a functional real time vision based American Sign Language recognition for D&M people have been developed for asl alphabets.

We achieved final accuracy of **98.0%** on our data set. We have improved our prediction after implementing two layers of algorithms wherein we have verified and predicted symbols which are more similar to each other.

This gives us the ability to detect almost all the symbols provided that they are shown properly, there is no noise in the background and lighting is adequate.

1. Concluding remarks

Hand Gestures are a powerful way for human communication, with lots of potential applications in the area of human computer interaction.

Vision based hand gesture recognition techniques have many proven advantages compared with traditional devices.

However, hand gesture recognition is a difficult problem and the current work is only a small contribution towards achieving the results needed in the field sign language gesture recognition. This Report presented a vision based system able to interpret isolated hand gestures from the American Sign Language(ASA).

The study findings show that American Sign Language (ASL) is commonly used in Nigeria by the hearing impaired hence; five hundred (500) ASL images were collected as training set. From the collection of images, a database of forty-nine (49) different signs was used. Having subjected the set of images to batch segmentation, features of each signs were detected and extracted from specific bounding-box of Region of Interest (ROI) to aid supervised learning. The combination FAST and SURF with a KNN of 10 also showed that unsupervised learning classification could determine the best matched feature from the existing database. In turn, the best match was converted to text as well as speech

**Future scopes and potential advancements**

We are planning to achieve higher accuracy even in case of complex backgrounds by trying out various background subtraction algorithms.

We are also thinking of improving the Pre Processing to predict gestures in low light conditions with a higher accuracy.

This project can be enhanced by being built as a web/mobile application for the users to conveniently access the project. Also, the existing project only works for ASL; it can be extended to work for other native sign languages with the right amount of data set and training. This project implements a finger spelling translator; however, sign languages are also spoken in a contextual basis where each gesture could represent an object, or verb. So, identifying this kind of a contextual signing would require a higher degree of processing and natural language processing (NLP).

**REFERENCES**

[1] V. Padmanabhan and M. Sornalatha, “Hand

gesture recognition and voice conversion

system for dumb people,” Int. J. Sci. Eng. Res.,

vol. 5, no. 5, 2014.

[2] C. Chen, J. Chen, and A. Ryan, “Scene

Segmentation of 3D Kinect Images with

Recursive Neural Networks,” 2011. [Online].

Available: http://cs.nyu.edu/. [Accessed: 14-

Mar-2017].

[3] J. C. Niebles, H. Wang, L. Fei-Fei, J. C.

Niebles, H. Wang, and L. Fei-Fei,

“Unsupervised Learning of Human Action

Categories Using Spatial-Temporal Words,” Int

J Comput Vis, 2008.

[4] P. A. Ajavon, “An Overview of Deaf Education

in Nigeria,” vol. 109, no. 1, pp. 5–10, 2006.

[5] D. Mart, “Sign Language Translator using

Microsoft Kinect XBOX 360 TM.”

[6] Xinapse, “Region of Interest (ROI) Algorithms,”

2018. .

[7] A. Li, W. Jiang, W. Yuan, D. Dai, S. Zhang, and

Z. Wei, “An Improved FAST + SURF Fast

Matching Algorithm,” Procedia - Procedia

Comput. Sci., vol. 107, no. Icict, pp. 306–312,

2017

Plagiarism Report

