INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, ALLAHABAD



Course Project Report

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Hand Gesture Recognition and Communication Software**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Submitted By:**

Aditya Raj - IIT2017005

Sameer Kathal - IIT2017010

Aman Saxena - IIT2017021

Nishant Kumar - IIT2017023

Sneh Sameer - IIT2017028

**Under the Supervision of**

Prof. Anupam Agarwal

**Candidate’s Declaration**

We hereby declare that the project work entitled ”**Hand Gesture Recognition and Communication Software**” submitted at Indian Institute of Information Technology, Allahabad, is the bonafide work of Aditya Raj (IIT2017005) , Sameer Kathal (IIT2017010), Aman Saxena (IIT2017021) , Nishant Kumar (IIT2017023), Sneh Sameer (IIT2017028) . It is a genuine record of our study carried out from February 2020 till present under the guidance of Prof. Anupam Agarwal. Due acknowledgements have been made in the text to all the materials used.

April 24, 2020 Prof. Anupam Agarwal Professor , IIIT Allahabad

**Contents**

1. **Abstract**
2. **Introduction**
3. **Literature Review**
4. **Problem Definition**
5. **Proposed Methodology**
   1. Dataset Collection
   2. Data Preprocessing
   3. Feature Extraction
   4. Gesture Recognition
   5. Advantages
6. **Implementation Plan (TimeLine)**
7. **Conclusion And Expected Outcome**
8. **References**

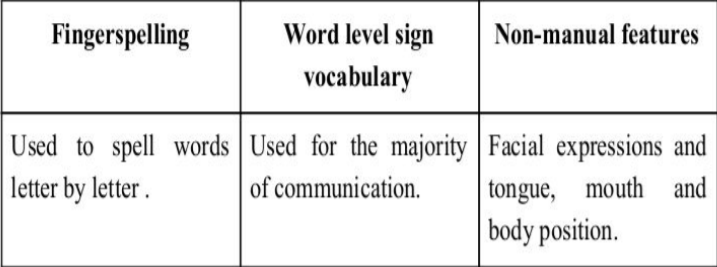
**1. Abstract**

This report describes the 6th-semester group project assignment that our group is working on, titled “**Hand Gesture Recognition and Communication Software**”. This project aims at analyzing and recognizing single hand gestures that can be converted to text in order to facilitate communication with differently-abled people. This has been a key challenge for communication with/between differently-abled people. Many different approaches have been formulated trying to solve this problem including  *Principal Component Analysis(PCA), Finger Peak and Angle Calculation,* *and Support Vector Machine (SVM).* Here we propose to recognize hand gestures using *Background Subtraction and Convolution Neural Network (CNN)* for Classification of Images into Text. Our method provides 98.28 % accuracy for 28 classes of characters with the assumption that the background is clear with no edges and lines.

**2. Introduction**

The communication with and between differently-abled people, specifically deaf and dumb, has been a problem ever since. They need to learn different techniques for expressing their thoughts to other people. There are different Sign languages like American Sign Language and Indian Sign Language that have been developed to facilitate the communication between these people. Since these languages may vary depending on region, people of different regions may use different languages. This creates a problem in communication between people using different sign languages. Also, normal people have difficulty communicating with differently-abled people.

Sign language is a visual language and consists of 3 major components :



In our project, we basically focus on producing a model that can recognize Fingerspelling based hand gestures in order to form a complete word by combining each gesture which is custom for every user. This eliminates the problem of lack of communication with and between differently-abled people as this system overcomes the problem of sign language dependency.

**3. Literature Review**

We studied the existing project related to the topic and the system behavior. A group of researchers have worked upon a robust technique to generate background by taking weighted averages. This is quite robust and works for most of the lighting conditions. Background subtraction was the most common way to extract the gestures out , after the background is generated. Several other methods such as, depth map for hand segmentation, hue saturation color model have been used for hand tracking and hand segmentation. Another method which was used by another group of people was based on canny edge detection after skin color segmentation, for edge detection for edge detection, Principal Component Analysis (PCA) for feature extraction and then Support Vector Machine (SVM) for classification. In one of the papers, they have proposed a method in which the angle and peak calculation approach is used to extract the features of hand gestures by using MATLAB and then they convert the recognized gesture into speech using MATLAB inbuilt command.

**4. Problem Definition**

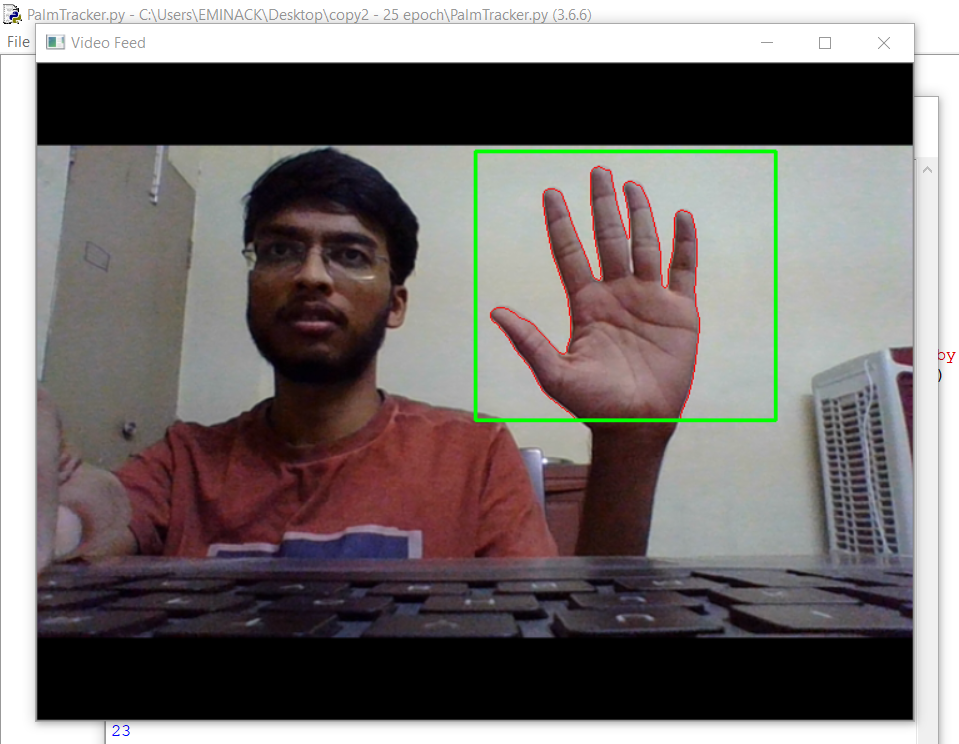
The crux of the problem is given an image of a hand gesture, we have to tell the character to which it is mapped. The images are stored in the database So, we can formally define the problem to be the prediction of the contextual meaning of a particular hand gesture, via learning of an appropriate system of related data.

**5. Proposed Methodology**

## **5.1 Dataset Collection**

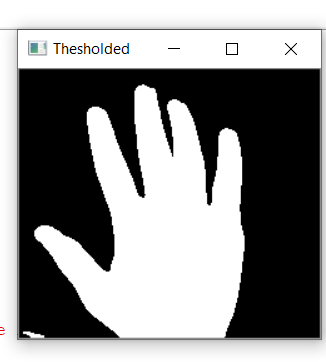
The dataset is custom generated. We used Open computer

vision(OpenCV) library in order to produce our dataset. Firstly we captured around 1000 images of each of the symbols in the English alphabet for training purposes and around 100 images per symbol for the testing purpose. First, we capture each frame shown by the webcam of our computer. In each frame, we define a region of interest (ROI) which is denoted by a green bounded square as shown in the image below.



**5.2 Data Preprocessing**

The grayscale image recorded is passed through a **Gaussian blur filter** of 7x7. After that **Background subtraction** is applied onto them to extract out the hand. The obtained image is then resized to 100x89 which is then stored into the system.**[5]**



**Background subtraction :**

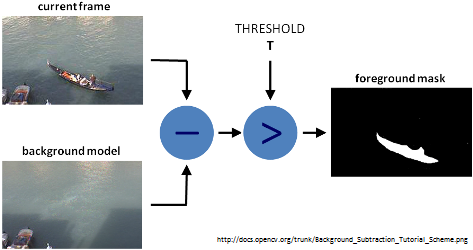
We make our system look over a particular scene (Empty Background) for 30

frames. During this period, we compute the running average over the current frame and the previous frames. Now we use the current frame which holds the foreground object (hand in our case) in addition to the background. We calculate the absolute difference between the background model (updated over time) and the current frame (which has our hand) to obtain a difference image that holds the newly added foreground object (which is our hand). This is what Background Subtraction is all about. The **threshold** value for the absolute difference is set as 25.

If **x(n)** represents the pixel intensity of an input image at a particular pixel

coordinate, then threshold decides how nicely we are going to segment/threshold the image into a binary image.

If 𝗫**(n) =**



**5.3 Training phase**

* For the training phase, we had 28000 images of 100x89 fed into CNN.

The network had the following structure:-

|  |  |  |  |
| --- | --- | --- | --- |
| **Name Layer** | **Layer type** | **No. of units** | **Activation function** |
| Input layer | - | - | - |
| *1st Conv* | *Convolution layer + max pooling* | *32 filters of 2x2 ,*  *2x2 max pooling* | *Relu* |
| *2nd Conv* | *Convolution layer + max pooling* | *64 filters of 2x2 ,*  *2x2 max pooling* | *Relu* |
| *3rd Conv* | *Convolution layer + max pooling* | *128 filters of 2x2,*  *2x2 max pooling* | *Relu* |
| *4th Conv* | *Convolution layer + max pooling* | *256 filters of 2x2 ,*  *2x2 max pooling* | *Relu* |
| *5th Conv* | *Convolution layer + max pooling* | *256 filters of 2x2 ,*  *2x2 max pooling* | *Relu* |
| *6th Conv* | *Convolution layer + max pooling* | *128 filters of 2x2 ,*  *2x2 max pooling* | *Relu* |
| *7th Conv* | *Convolution layer + max pooling* | *64 filters of 2x2 ,*  *2x2 max pooling* | *Relu* |
| *1st hidden Layer* | *Fully Connected Layer* | *1000* | *Relu* |
| *2nd hidden Layer* | Dropout Layer | 25% dropout | - |
| *Output Layer* | *Fully Connected Layer* | *28* | *softmax* |

This contributed to **689144** number of trainable parameters. The neural

network was trained using a loss Function of **Categorical Cross entropy**

and **mini batch** **Gradient Descent** was used to update the weights and biases with Learning rate of **0.001** . The model was trained for **14 epochs** with batch size **64**.

**5.4 Testing Phase**

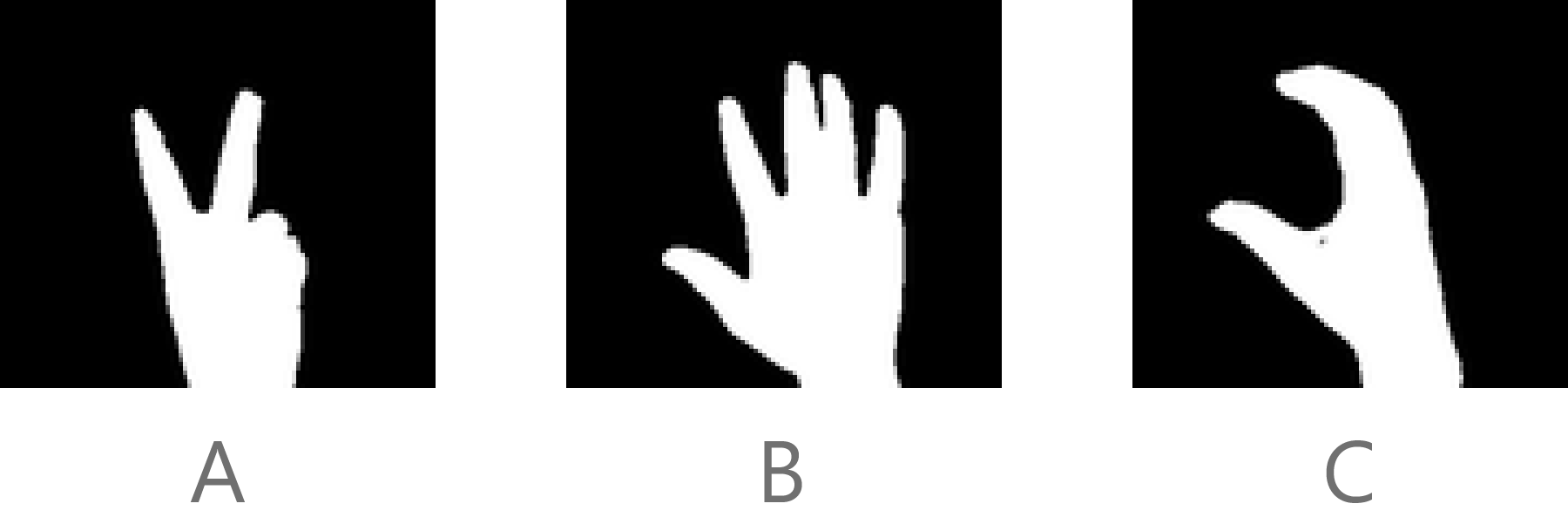
* For testing of the network 10% of data was used . This data was passed through the same preprocessing. After that the features were passed through the neural network to generate the predicted result.

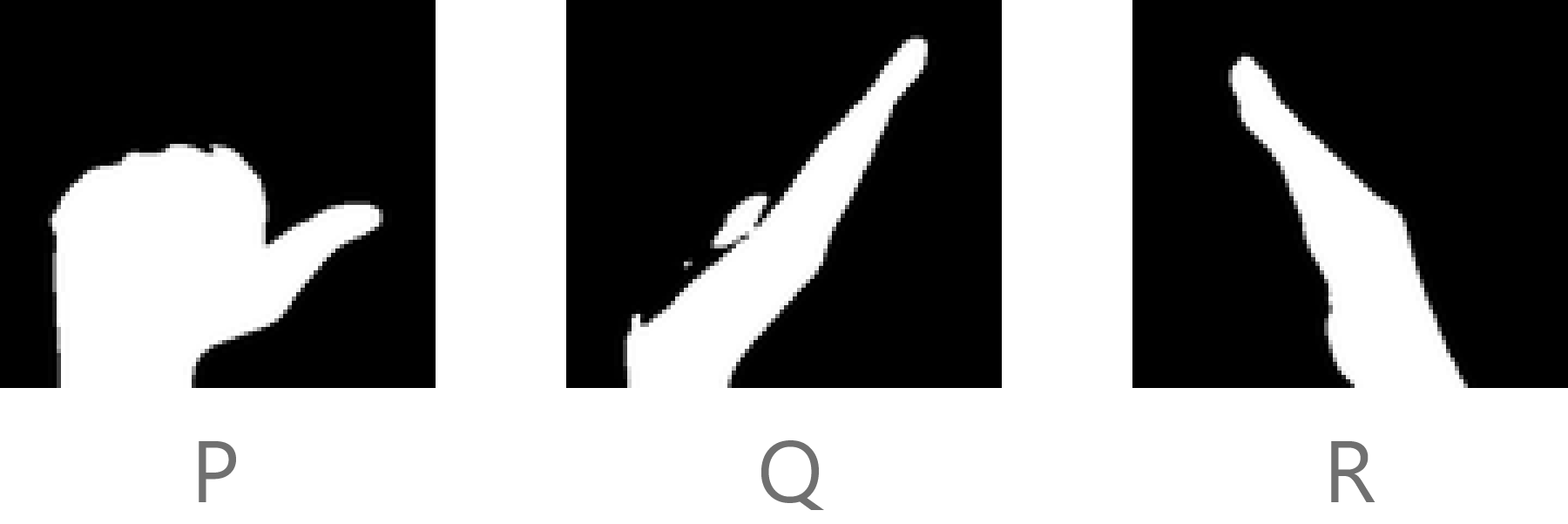
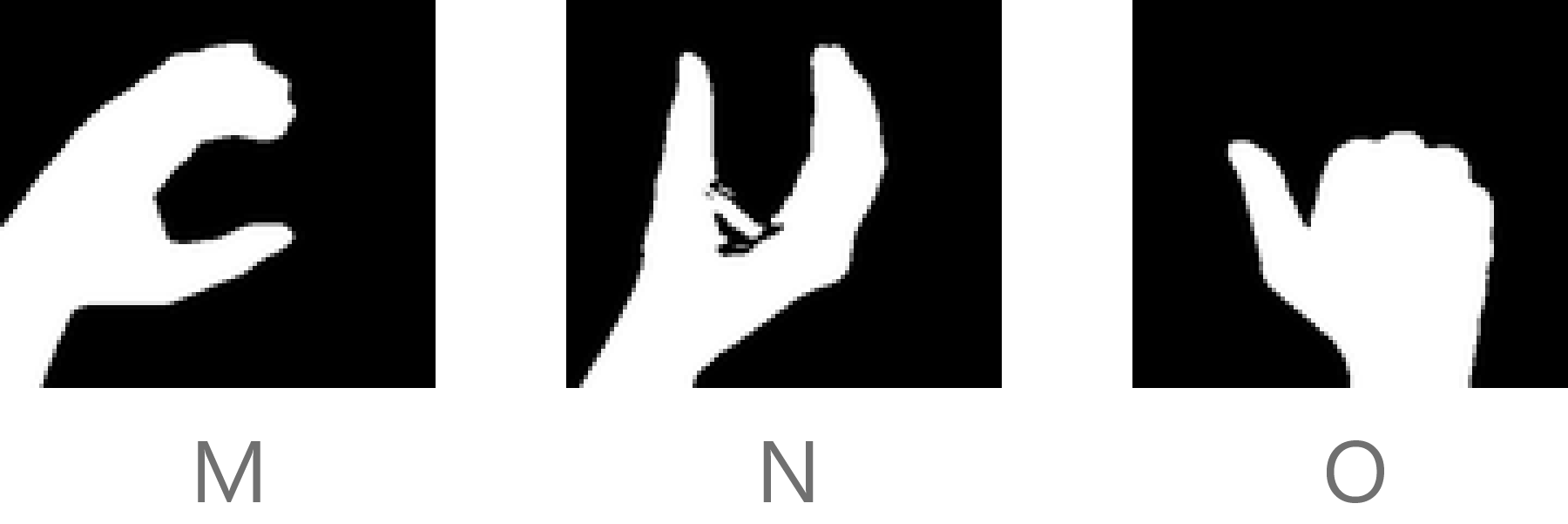
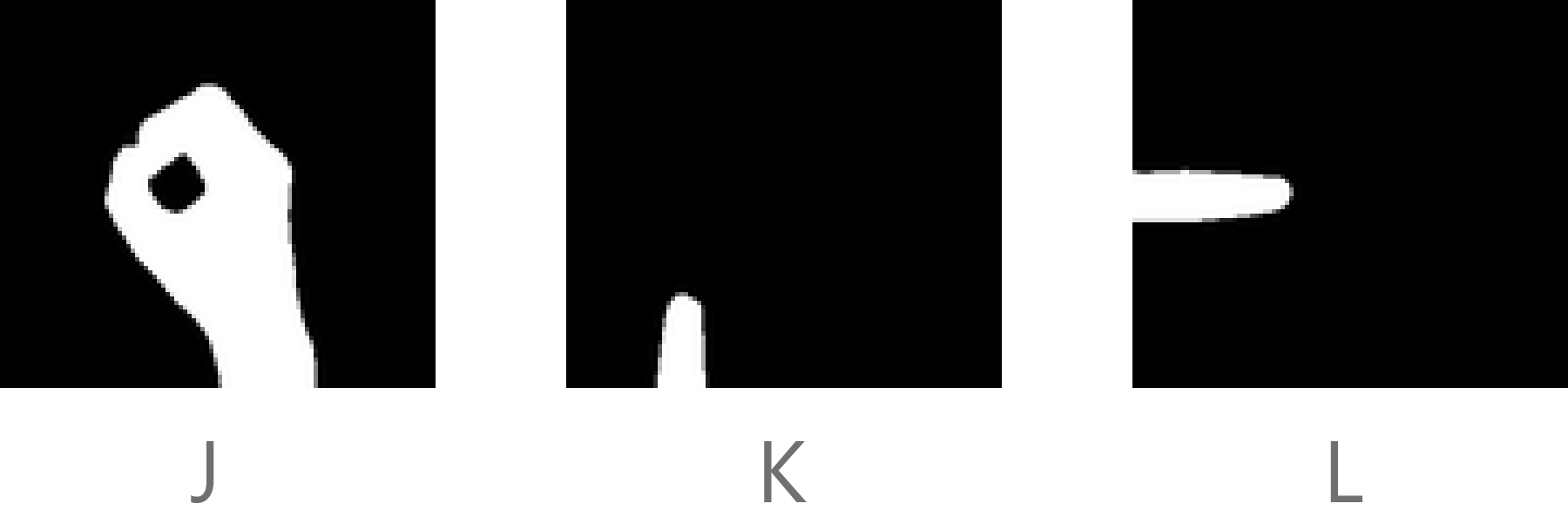
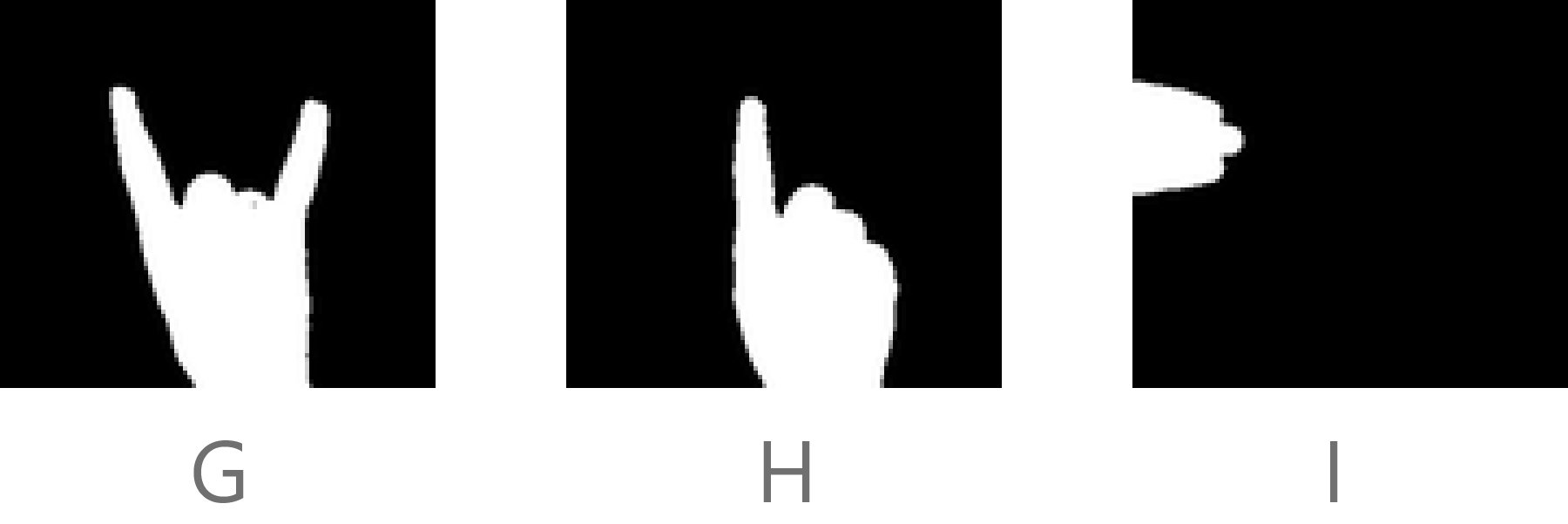
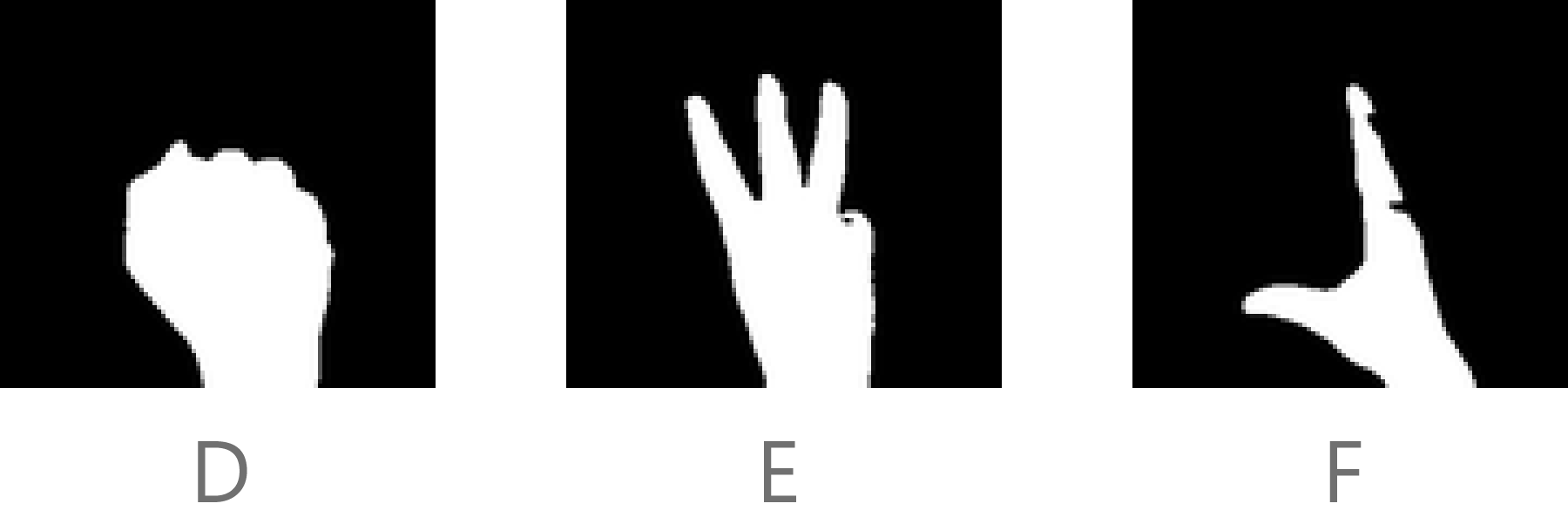
**5.5 Output generation**

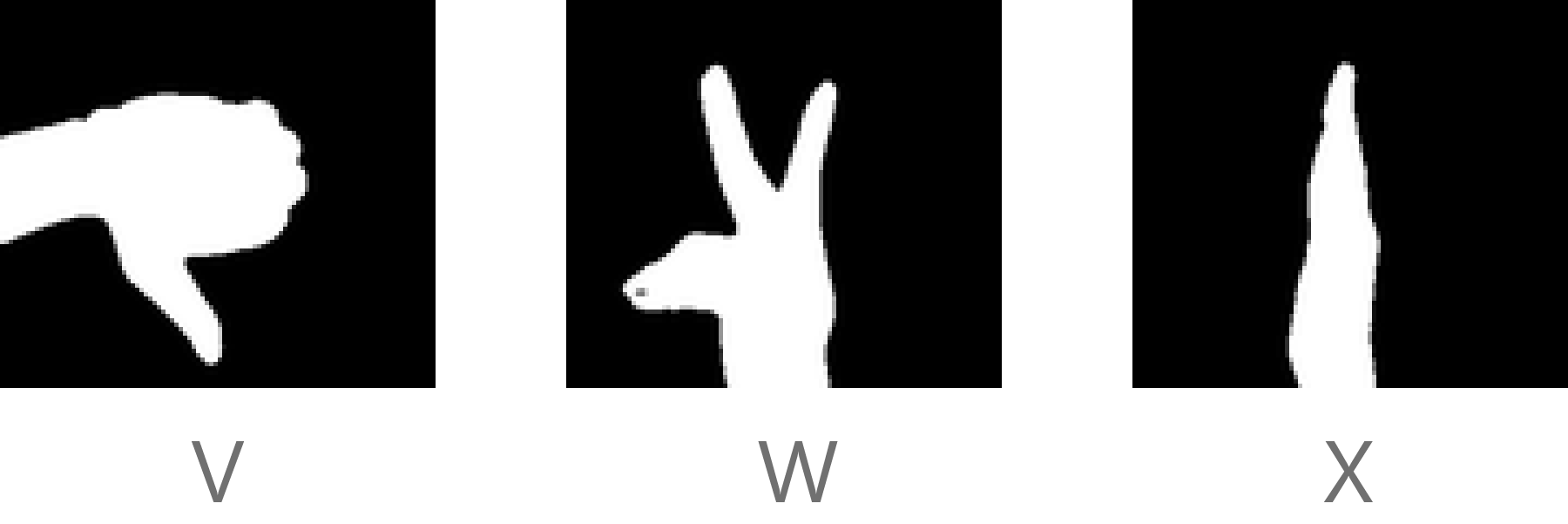
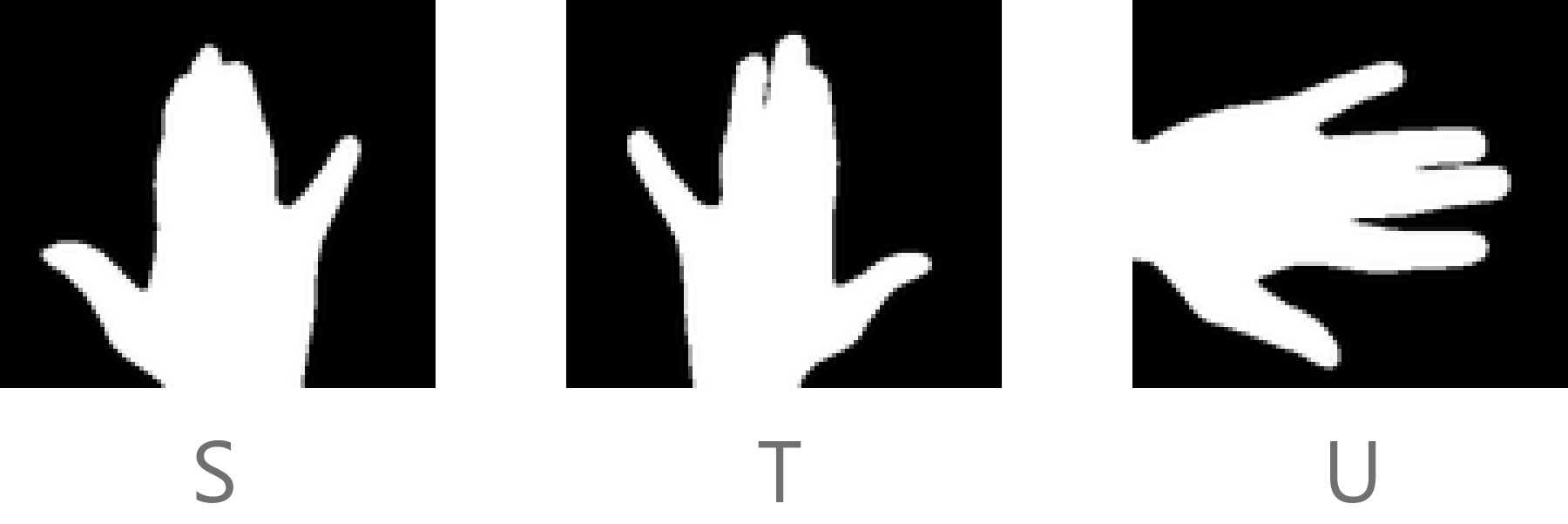
* When the application starts , running average of 30 frames is taken to learn the background .
* Now when the user shows the gesture , it is converted into a Grayscale image and passed through the Background subtractor , then resized and passed into the model for prediction.
* The predicted class is passed through a function to get the corresponding character of that class which is displayed onto the screen along with its confidence.

**Generation of words and sentences**

1. When the ROI( Region of Interest) is green , then the user needs to show his gesture for a period of 6 sec , the count of predicted character for 180 frames is stored in the dictionary. At the end of this, the character with maximum count is appended at the end of the word.
2. When the ROI(Region of Interest) is blue , that means now the user has to wait for 2 second for the next character to be predicted.
3. If class “SPACE” is predicted means the word has ended and needs to be appended at the end of the sentence. The word is set to an empty string after that.









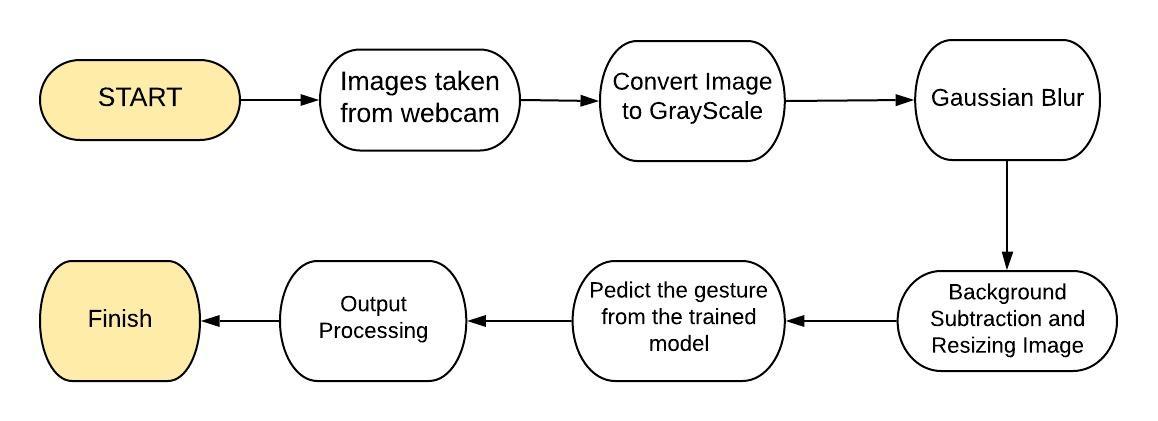
BACKSPACE

**5.6 ADVANTAGES**

* Eliminates the dependency of sign language between the users using it, since each person can customize the actions according to his/her needs.
* User can also create a custom database which can give the flexibility to :

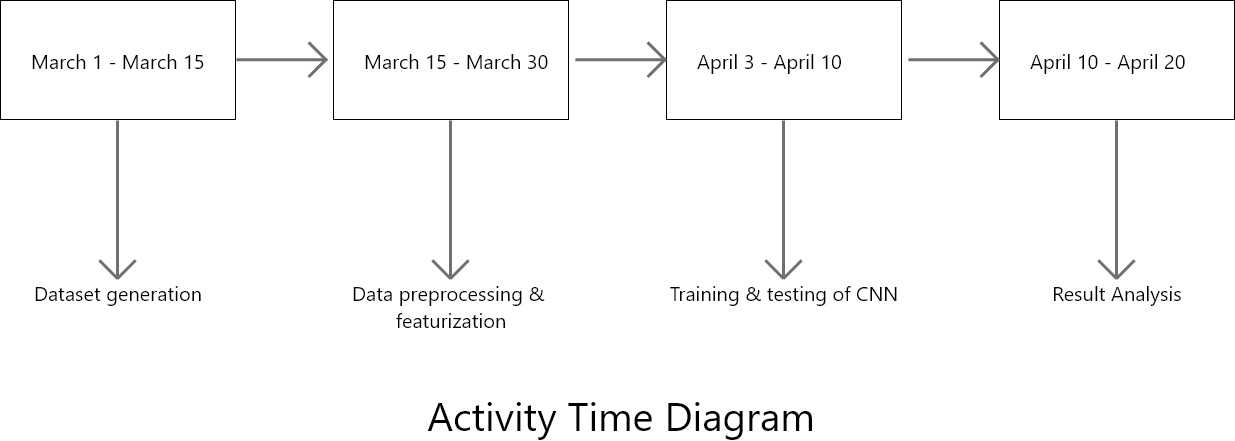
i) Use words

ii) Use custom templates

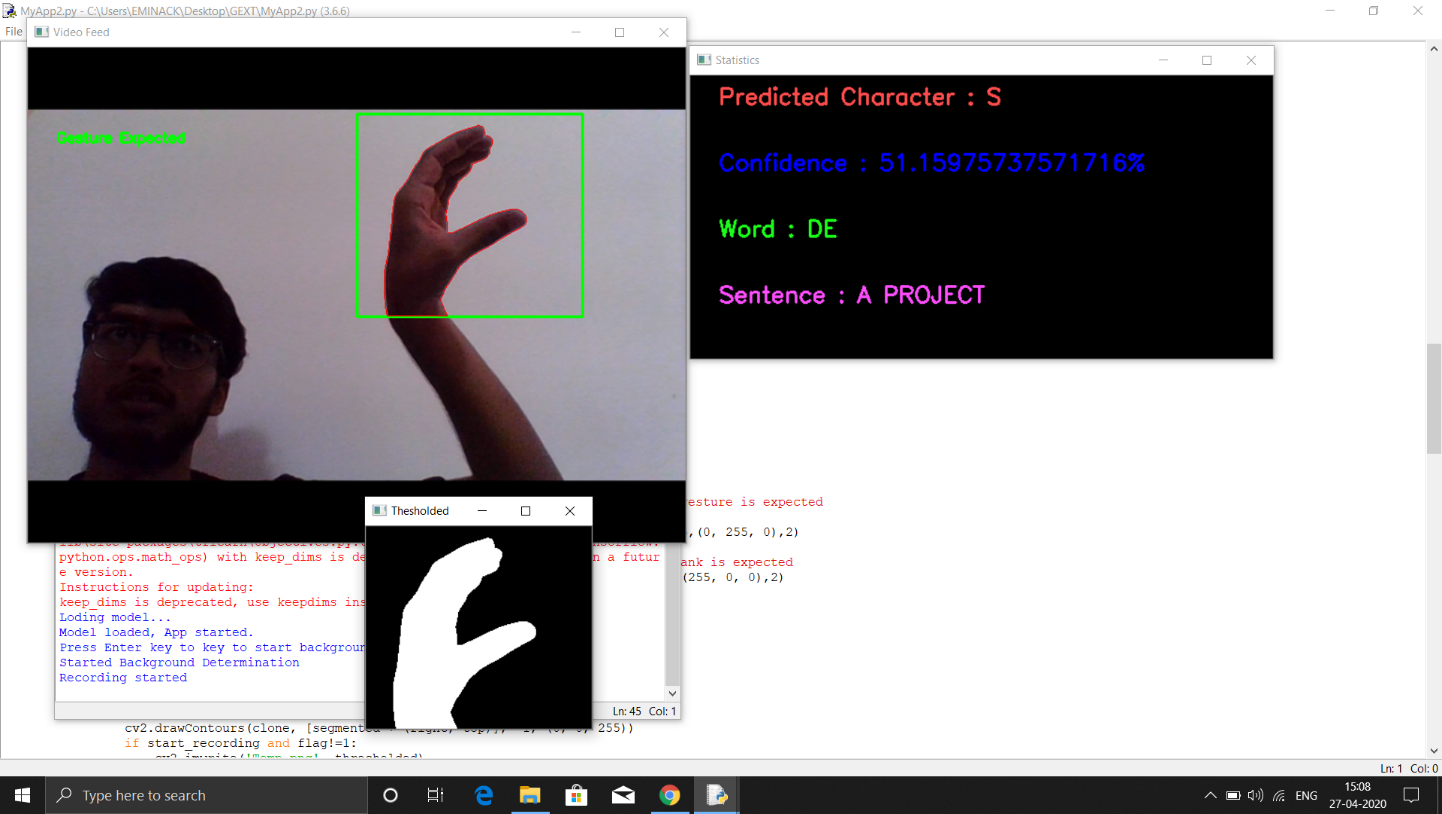
****

**Fig -1: System Architecture**

**7. Implementation Plan (Timeline)**

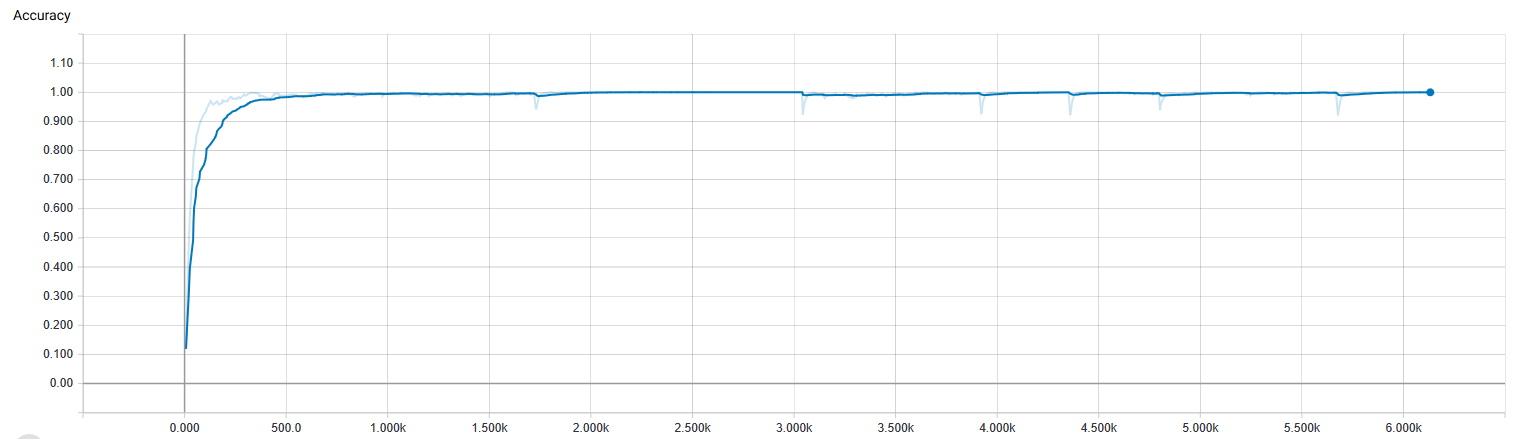
****

1. **Conclusion And Expected Results**

****

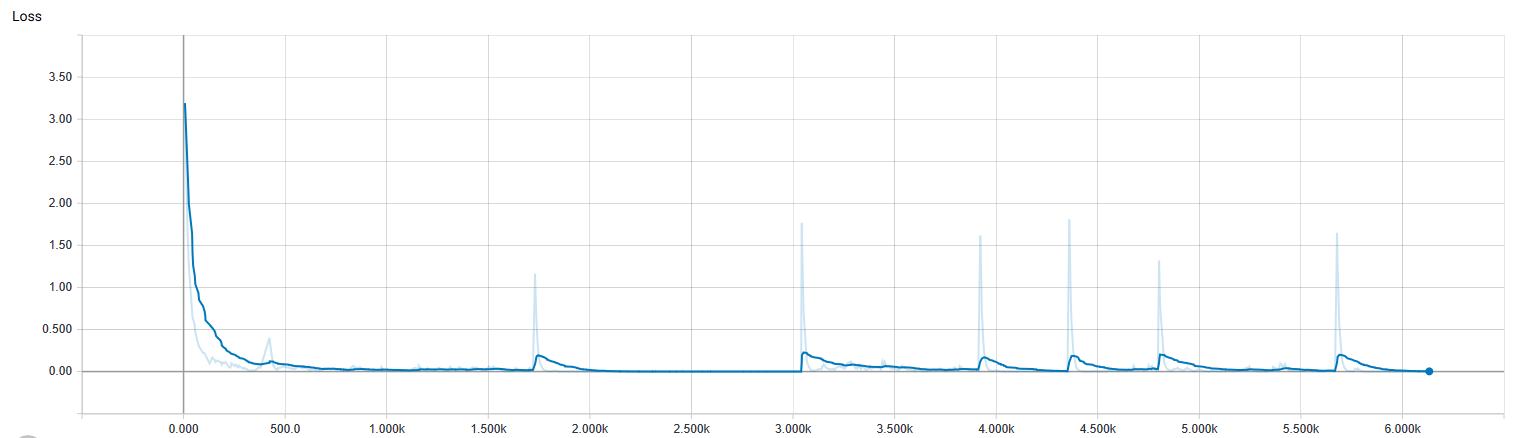
The expected outcome is a multiclass classifier constructed using Convolutional Neural Networks to predict the most appropriate textual information among a given set of contextual data.We achieved a validation accuracy of 98.28% for classification of 28 classes on the validation set. The ratio of train set to validation set is 10:1.

We can clearly see that model accuracy becomes constant after 6000 steps. Also validation loss decreases constantly meaning the model is trained well and is not overfitted.



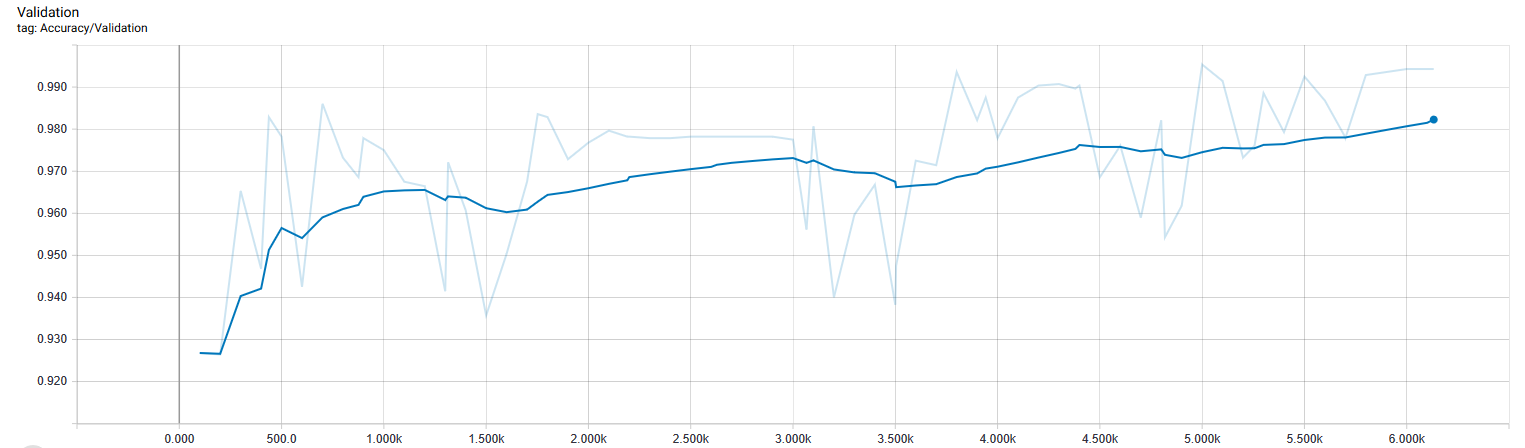
No of steps

Training - Accuracy



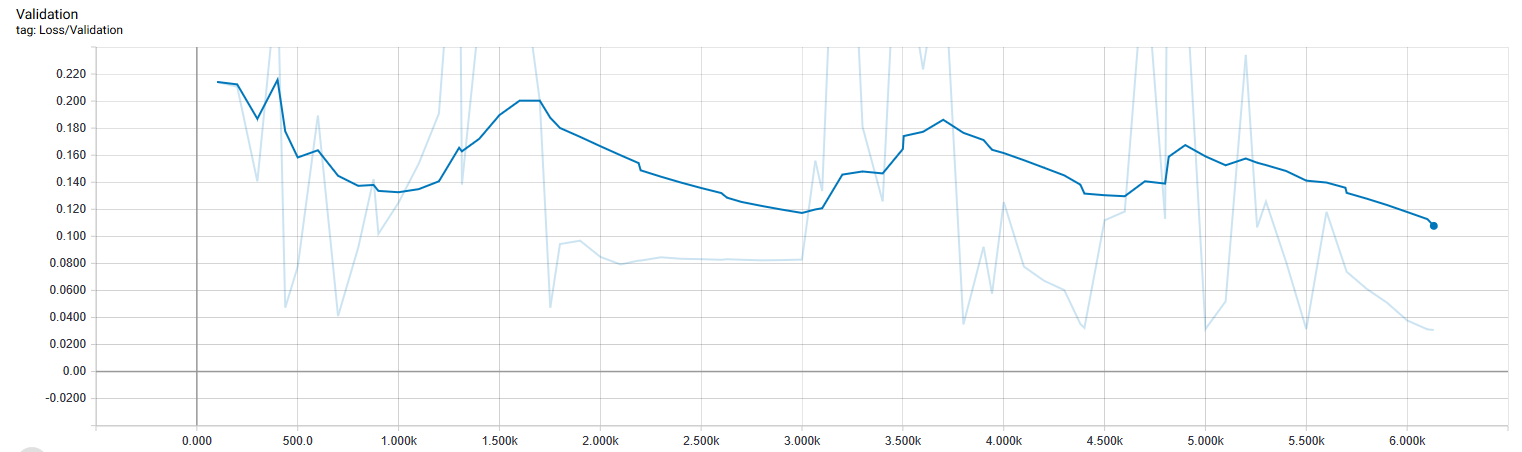
No of steps

Training-loss



No of steps

Validation Accuracy



No of steps

Validation loss

**9. References**

[1] Hazem Khaled , Samir G. Sayed El Sayed M. Saad and Hossam Ali. “Hand Gesture Recognition Using Modified 1$ and Background Subtraction Algorithms”. 2015 Hindawi, Mathematical Problems in Engineering, 2015.

[2] Byeongkeun Kang, Subarna Tripathi and Truong Q. Nguyen. “ Real-time sign language fingerspelling recognition using convolutional neural networks from depth map “. 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR).

[3] Shinde, Shweta S., Rajesh M. Autee, and Vitthal K. Bhosale. "Real-time two-way communication approach for hearing impaired and dumb person based on image processing." Computational Intelligence and Computing Research (ICCIC), 2016 IEEE International Conference on. IEEE, 2016.