SIGN LANGUAGE RECOGNITION

Project Report

PROJECT WORK PHASE-2 (ECS899)

Degree

BACHELOR OF TECHNOLOGY (CSE)

PROJECT GUIDE:

MR. AADITYA JAIN

SUBMITTED BY:

MOHIT CHAUHAN (TCA1809064)
MOHD. ZEESHAAN (TCA1911002)
PIYUSH VERMA (TCA1911008)

Session 2021-2022

FACULTY OF ENGINEERING & COMPUTING SCIENCES

TEERTHANKER MAHAVEER UNIVERSITY, MORADABAD

DECLARATION

We hereby declare that this Project Report titled **Sign Language Recognition** submitted by us and approved by our project guide, to the College of Computing Sciences and Information Technology (CCSIT), Teerthanker Mahaveer University, Moradabad, is a bonafide work undertaken by us and it is not submitted to any other University or Institution for the award of any degree diploma / certificate or published any time before.

Project ID :	**17**	
Student Name:	**MOHD. ZEESHAAN**	**ZEESHAAN**
Student Name:	**MOHIT CHAUHAN**	**MOHIT**
Student Name:	**PIYUSH VERMA**	**PIYUSH**
Project Guide :	**MR. AADITYA JAIN**	Signature

ACKNOWLEDGEMNT

I would like to thank the Computer Science Department for giving me the opportunity to work on this project. This was quite a great experience and I learned a lot from it. It helped me to explore my skills and increased my interest in this project.

Special thanks to my guide Mr.Aaditya Jain for being so accommodating and obliging and for his confidence in my abilities. He has guided me throughout the project and provided his insights.

Further I would like to thanks my project partners who help me through their project because without their help, the project work is not possible to be completed.

ABSTRACT

Sign language is one of the oldest and most natural form of language for communication, but since most people do not know sign language and interpreters are very difficult to come by, we have come up with a real time method using neural networks for finger Spelling based American sign language. In our method, the hand is first passed through a filter and after the filter is applied the hand is passed through a classifier which predicts the class of the hand gestures. Our method provides 80 % accuracy for the 26 letters of the alphabet.

A hand gesture recognition system offer a chance for deal folks to speak with vocal human while not the requirement of AN interpreter. The system is made for the machine-driven conversion of sign language into the matter content and its speech

Table Of Contents

- **Introduction:**

American Sign Language is a predominant Sign Language Since the only disability Deaf and Dumb people have to communicate related and since they could not speak language. Communication is that method of exchange the thoughts and message in varied ways that like to speech, Signals , behaviour and visuals.

D&M folks create use of their hand to precise totally different gestures to precise their concepts with others. Gestures are the non-verbally changed message and these gestures to precise their concepts with others. Gestures are the non-verbally changed message and these gesture are understood with vision.

Minimizing the articulation gap among D&M and Non- D&M folks turns into a need to foam a bound effective speech among all language translation is among the foremost growing lines of analysis and it perm it's the uttermost one among the foremost growing lines of analysis and its permits the uttermost natural manner of communication for those with hearing impairments.

A hand gesture recognition system offer a chance for deal folks to speak with vocal human while not the requirement of AN interpreter. The system is made for the machine-driven conversion of sign language into the matter content and its speech.

The goal of this project to create a neural network ready using CNN to classify that letter of the sign which in the format of (ASL) alphabet is being signed, given a picture of language hand a attain table sign translator , which might take communication in sing and translate them into written and oral language.

Such a translator would greatly lower the barrier and oral language. Mute people to be ready to higher communicate with other in day to day interactions. There are giant barriers that deeply have an effect on life quality

steam from the communication disconnected between the deaf and therefore the hearing.

In our project we have a tendency to primarily specialise in manufacturing a model which might acknowledge sign language based mostly hand gestures so as to create an entire word by combining every gesture. The gestures we have a tendency to aim to coach ar as given within the image below.

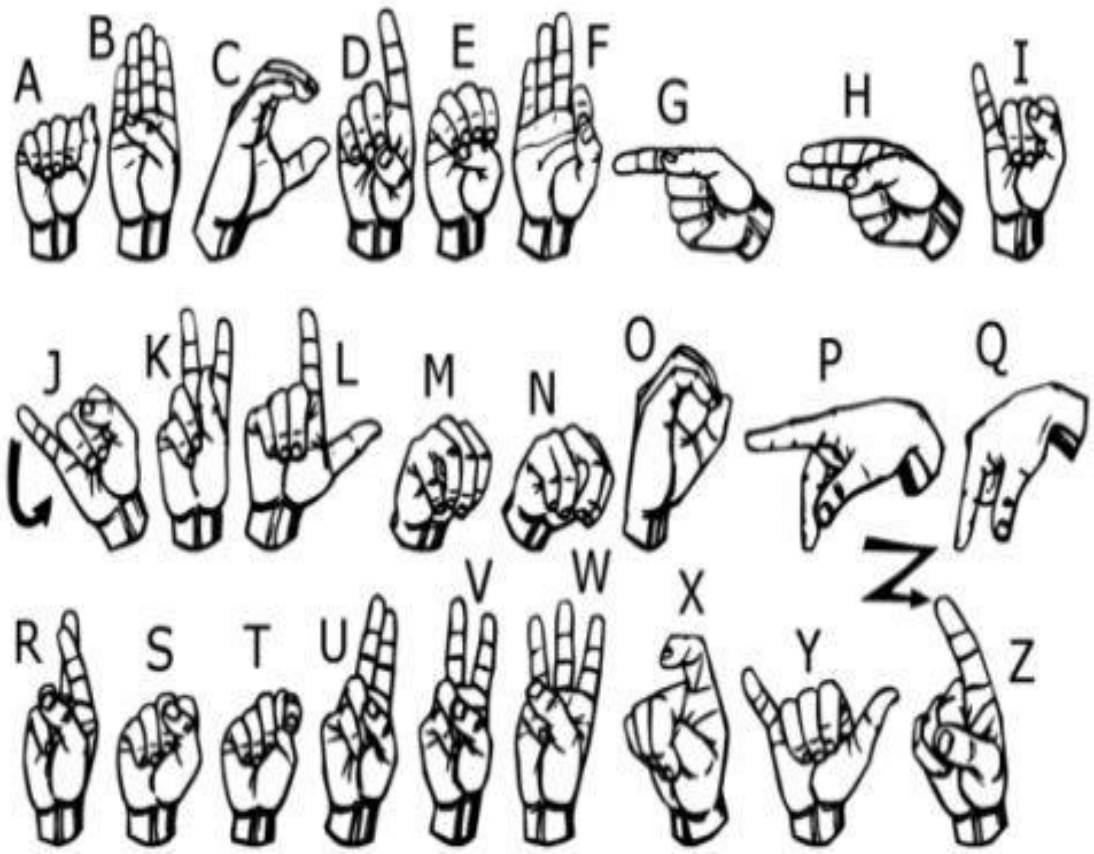


Figure – 1 ASL Sign

- **Motivation:**

For interaction between ordinary people and D&M people a language barrier is created as sign language structure since it is complete different from normal text. So they depend on vision-based communication for interaction. If there is a common interface that converts the sign language to text, then the gestures can be simply understood by non-D&M people. So, research has been made for a vision-based interface system wherever D&M people can enjoy communication without really knowing each other's language.

The aim is to develop a easy to use Human Computer Interface (HCI) wherever the computer understands the human sign language. There are various sign languages all over the world, namely American Sign Language (ASL), French Sign Language, British Sign Language (BSL), Indian Sign language, Japanese Sign Language and work has been done on other languages all around the world. Communication is one of the essential demand for survival in society. Deaf and dumb natives communicate among themselves using sign language but normal people find it difficult to know their language. extensive work has been done on American sign language recognition but Indian sign language differs significantly from American sign language.ISL uses two hands for communicating(20 out of 26) whereas ASL uses single hand for communicating. Using each hands often leads to darkness of features due to overlapping of hands. In addition to this, lack of datasets along with variance in sign language with locality has resulted in restrained efforts in ISL gesture detection. Our project aims at taking the basic step in bridging the communication gap between normal group and deaf and dumb people using Indian sign language. Effective extension of this project to words and familiar expressions might not solely create the deaf and dumb people communicate faster and easier with outer world, however also afford a boost in going up self-sufficient systems for understanding.

- **Literature Survey:**

  In the recent years, there has been done research done on the hand gesture recognition. With the help of literature survey, we realized that the basic steps in hand gesture recognition are: -

  - Data acquisition
  - Data pre-processing
  - Feature extraction
  - Gesture classification

- **Data acquisition:**

  The different approaches to acquire data about the hand gesture can be done in the following ways:

- **Use of sensory devices:**

  It uses mechanical devices to produce correct hand configuration, and position. Totally Different glove-based approaches are often used to extract information. However it is expensive and not user friendly.

- **Vision based approach:**

  In vision-based method, the computer webcam is the input device used for observe the information of hands or else fingers. The Vision based method required just a camera, thus realize a natural interface between humans and computers without the use of any extra devices, thereby reducing cost. These systems be likely to match biological vision by describing artificial vision systems that are implemented in software and hardware. The main challenge of vision-based hand detection range from cope with the large

changeability of the human hand's form due to a huge number of hand movements, to different skin-color possibilities as well as to the variation in viewpoints, balance, as well as speed of the camera capturing the scene.

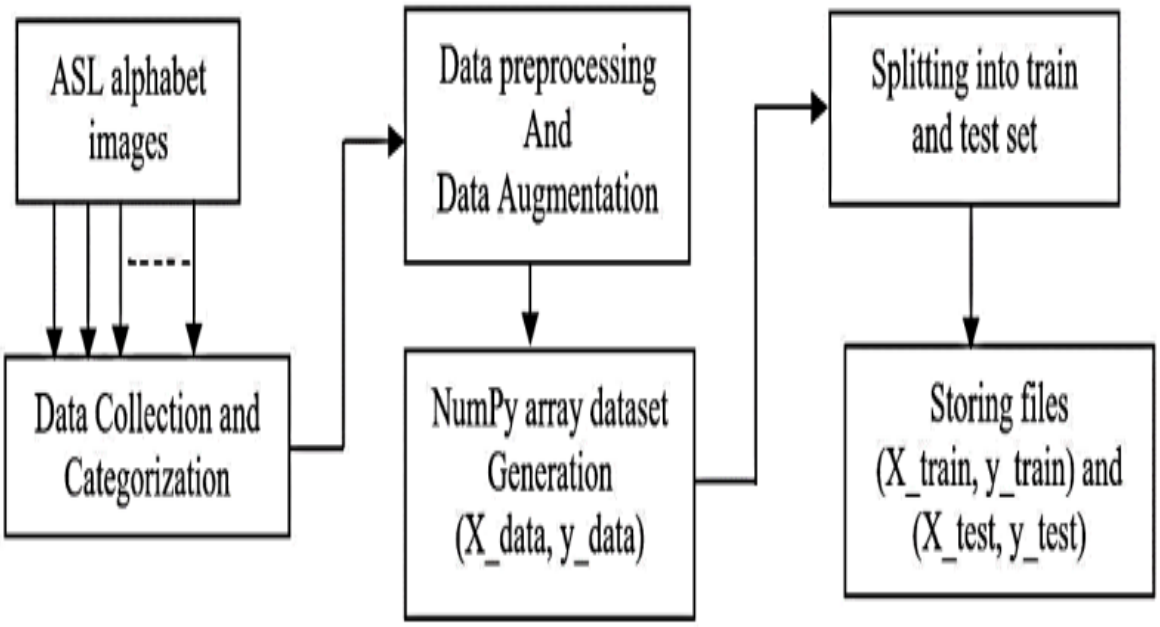


Figure 2: Block diagram of data gaining

- **Data Pre-Processing and 3.3 Feature extraction for vision-based approach:**

- In [1] the approach for hand detection combines threshold-based color detection by background calculation. We can utilize face detector to make a distinction between faces and hands while they both involved parallel skin-color.

- We can pull out necessary image which is to be train by means of applies a filter call Gaussian Blur (also known as Gaussian smoothing). The filter be capable of easily useful using open computer vision (also known as OpenCV) and is describe in .

- For extract necessary image which is toward be skilled we can use instrumented gloves as mentioned. This help decrease computation time for Pre-Processing and give us more brief and accurate data compare to applying filters on data received from video extraction.

- We tried doing the hand segmentation of an image by means of color segmentation techniques but skin color in addition to tone is very dependent on the lighting conditions due to which output, we get for the segmentation we try to do were rejection so great. Moreover, we have a massive number of symbols to be trained for our project lots of of which looks alike to each other like the gesture for symbol 'V' and digit '2', hence we resolute that in order to produce improved accuracies for our massive number of symbols, pretty than segmenting the hand out of a chance background we keep background of hand a stable single color so that we don't require to segment it on the source of skin color. This would help us to get improved results.

- **Gesture Classification:**

- In [1] Hidden Markov Models (HMM) is used for the classification of the gestures. Gestures are extracted from a sequence of video images by track the skin-color blobs parallel to the hand into a body– face space centre on the face of the user.

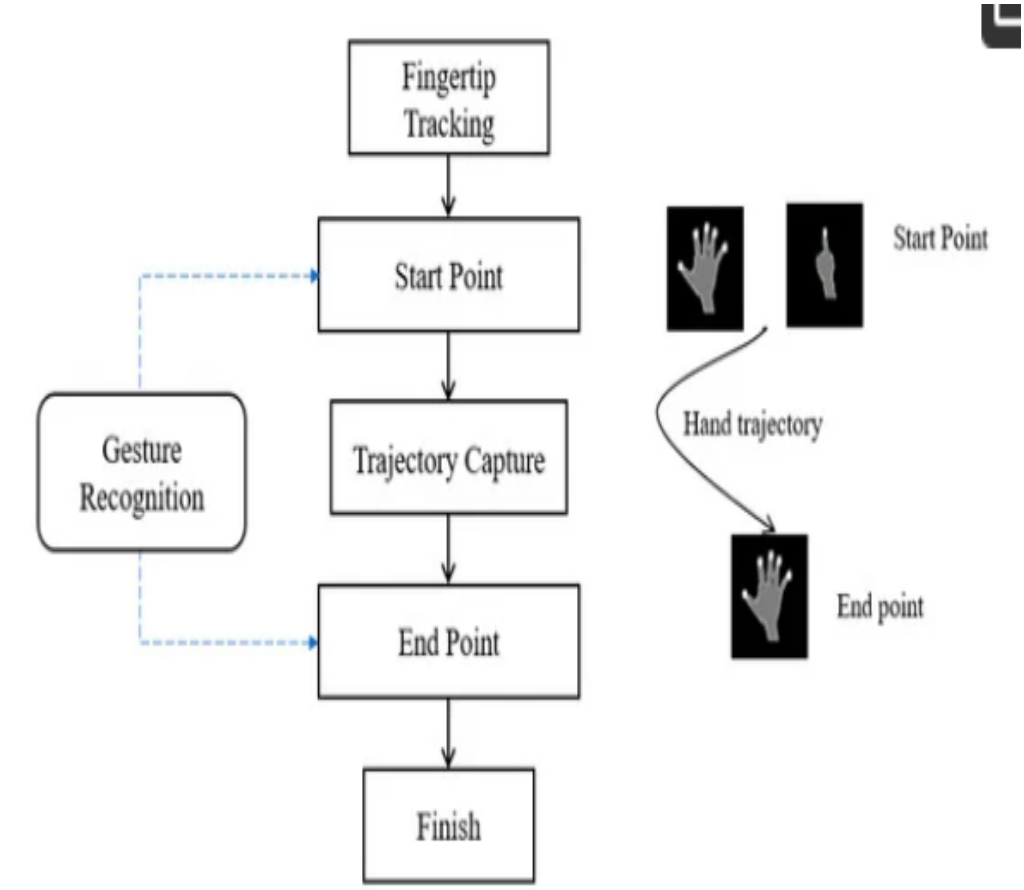


Figure 3 Proposed schemes for gesture spotting.

- Thus, dissimilar many other recognition methods, this method is not needy on skin color. And sign direction .The gestures are extract from each frame of the video, by means of a static background. The first step is to segment in addition to label the objects of interest and to remove numerical invariants from them used which is an effective and quick method for static hand gesture recognition. It is based on classifying the dissimilar gestures according to geometric base invariants which are obtain on or after image data after segmentation.

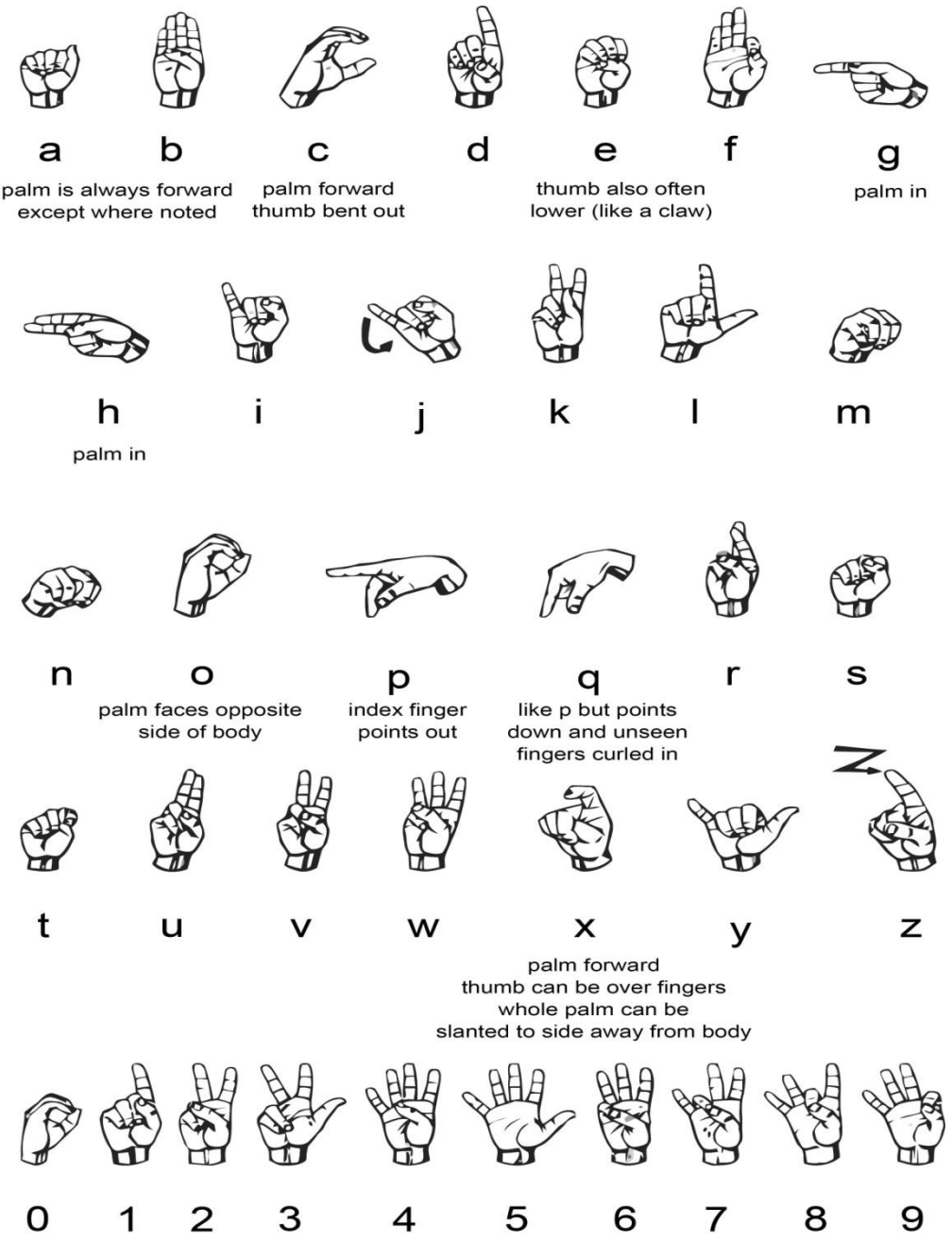


Figure 4. Pose of symbols

- Thus, dissimilar many other recognition methods, this method is not dependent on skin color. The gestures are extracted from each frame of the video, with a stationary background. The first step is to slice and labeled the objects of attention and to extract geometric invariants from them. Next step is the classification of gestures by using a K nearest neighbor algorithm aid with distance weighting algorithm (KNNDW) to give suitable data for a locally weighted Naïve Bayes classifier.

- According to the document on “Human Hand Gesture Recognition by means of a Convolution Neural Network” by Hsien-I Lin, Ming-Hsiang Hsu, and Wei-Kai Chen (graduates of Institute of Automation Technology National Taipei University of Technology Taipei, Taiwan), they contain construct a skin model to extracted the hands not in the image and then apply binary threshold toward the whole image. After obtaining the threshold image they calibrate it about the principal axis in organize to centre the image about the axis. They contribution this image to a convolutional neural network model in order to train as well as predict the outputs. They contain trained their model of hand gestures and using this model they produced an accuracy.

- **Key words and Definitions:**

- **Feature Extraction and Representation:**

  The representation of an image as a 3D matrix having measurement as of height and width of the image as well as the value of each pixel as deepness in case of Grayscale and 3 in case of RGB). additional these pixel values are use for extracting helpful features by means of CNN.

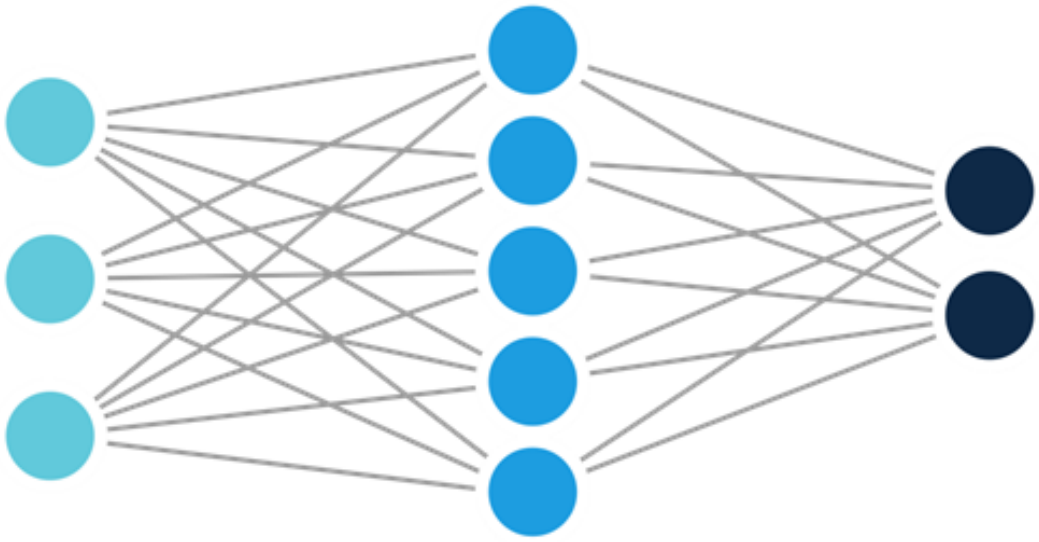
- **Artificial Neural Network (ANN):**

  Artificial Neural Network is a relation of neurons, replicating the structure of human brain. Every connection of neuron transfers information to a different neuron. Inputs are fed keen on first layer of neuron which process it as well transfers to an additional layer of neurons called as hidden layers. After giving out of information through multiple layers of hidden layers, information is passed to final output layer.

  Artificial neural network covers all the aspect related to the artificial neural network. In this, we will talk about ANNs, Adaptive resonance theory,

Kohonen self-organizing map, Building blocks, unsupervised learning, Genetic algorithm, etc.

Artificial Neural Network could be finest represented as a weighted directed graph, wherever the artificial neurons form the nodes. The association among the neurons outputs also neuron inputs can be viewed as the directed edges with weights. The Artificial Neural Network receive the input signal from the outside source in the form of a pattern and image in the figure of a vector. These inputs are mathematically assigned by the notations x(n) designed for every n number of inputs.



Input layer Hidden layer Output Layer

Artificial Neural Network Architecture

Figure – 5

These are able of learning and have to be trained. There are different learning strategies:

- Unsupervised Learning
- Supervised Learning
- Reinforcement Learning

- **Convolutional Neural Network (CNN):**

Nothing like regular Neural Networks, in the layers of CNN, the neuron are prearranged in 3 dimensions: width, height, depth. The neuron in a layer determination only be connect to a little region of the layer (window size) earlier than it, instead of every one of the neurons in a fully-connect technique. Furthermore the final production layer would have dimensions (number of classes), because by the finish of the CNN planning we resolve reduced the full image into a single vector of class scores.

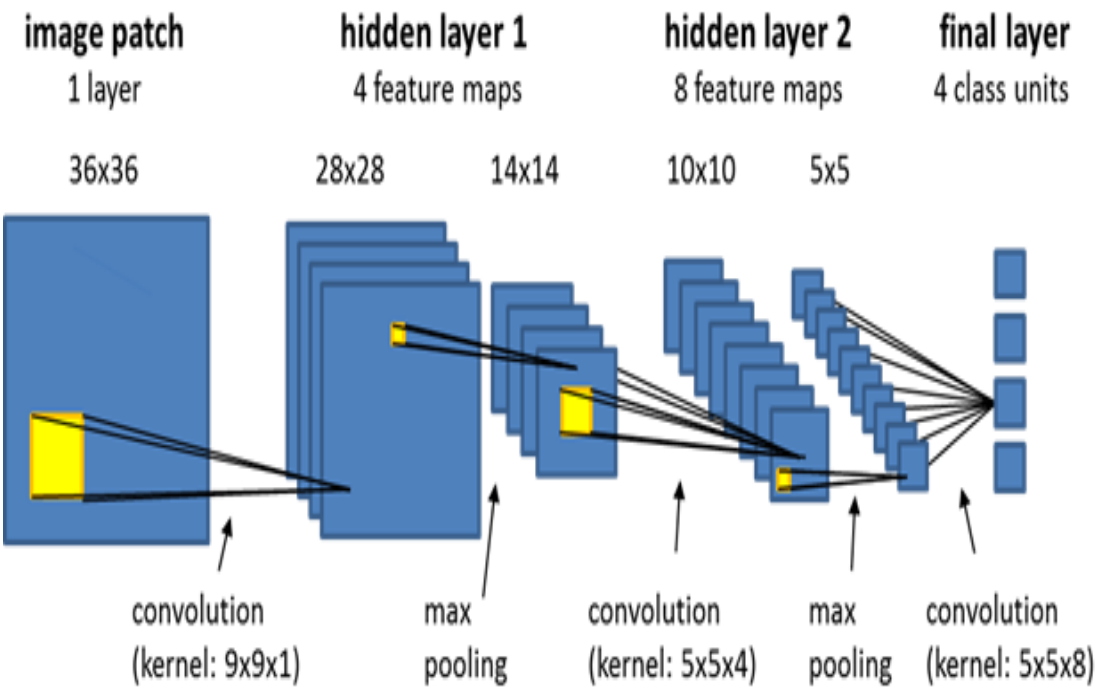


Figure – 6 CNN Layers

- **Convolution Layer:**

In convolution layer we take a tiny window size that extends to the depth of the input matrix. The layer consists of learnable filters of window size. Through every iteration we slid the window by stride size , and calculate the dot product of clean entries and input values at a known position.

As we go on by means of this process we will create a 2-Dimensional activation matrix that gives the response of that matrix at each spatial position. With the point of the network we learn filters that activate when they observe some type of optical characteristic such as an border of some orientation or a dot of some colours.

- **Pooling Layer:**

We used pooling layer to reduce the size of activation matrix as well as ultimately reducing the learnable parameters. There are two types of pooling:

- **Max Pooling:** In the max pooling we take a window size [for example window of size 2*2], and no more than take the maximum of 4 values. Well cover this window and continue this process, so well finally get an activation matrix half of its original Size.
- **Average Pooling:** In average pooling, we take improvement of all the Values in a window.

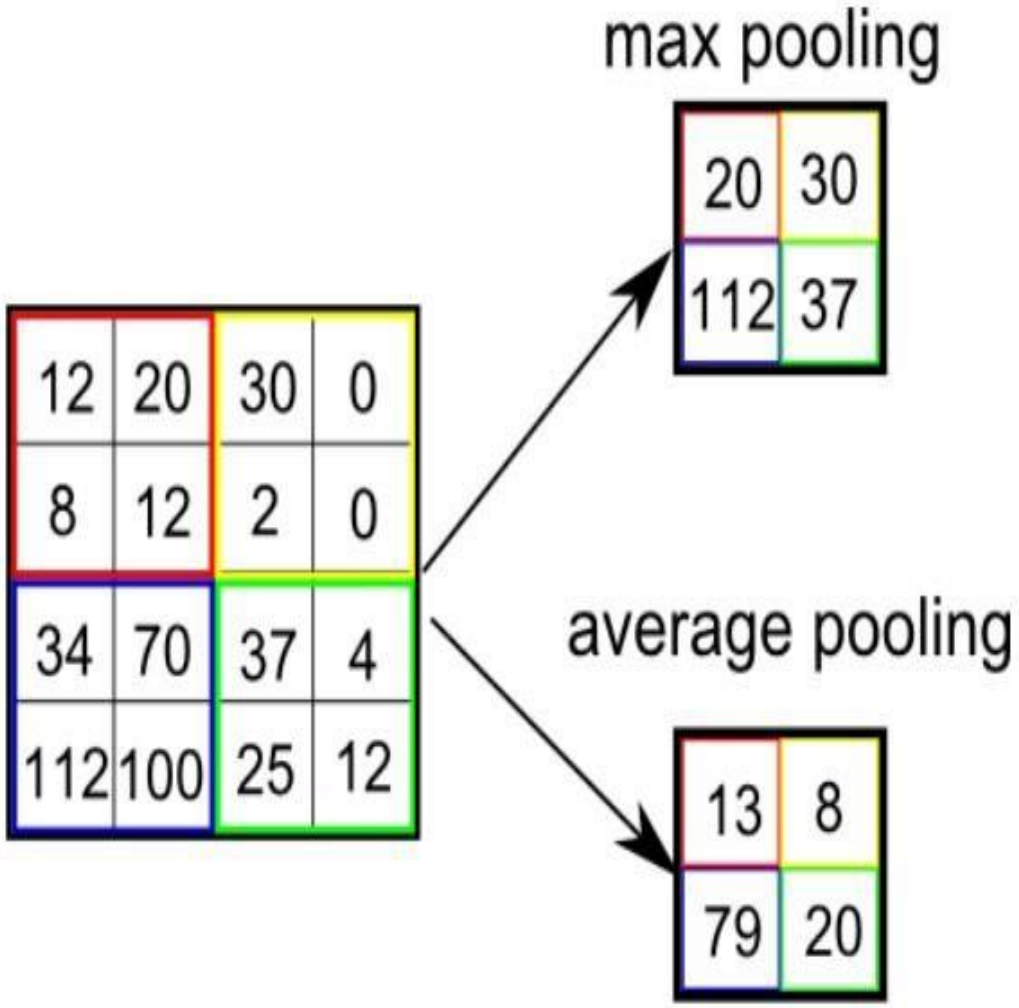


Figure – 5 Pooling Layer

- **Fully Connected Layer:**

  In convolution layer, neurons are connected only to a restricted region, at the same time as in a fully connected area , we will connect all the input to neurons.

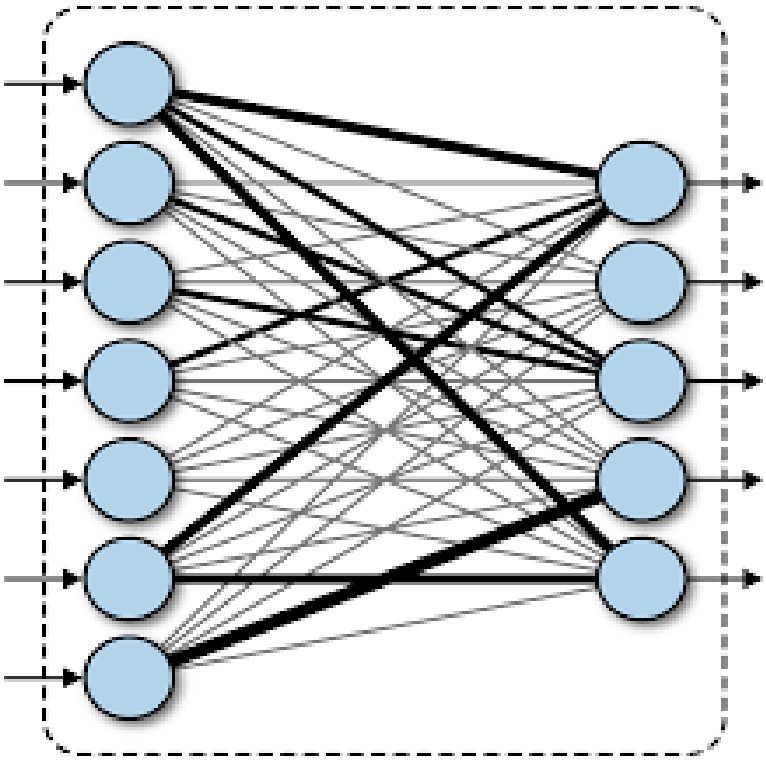


Figure – 6 Fully Connected layer

- **Final Output Layer:**

  After gaining values from the fully connected layer, we are connected them to the final layer of neuron [having counted the equivalent to total number of class], that will forecast the chance of each image to be in dissimilar classes.

- **TensorFlow:**

  TensorFlow is an end-to-end open-source platform for Machine Learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources by way of the intention of lets researchers move forward the state-of-the-art in Machine Learning and developers without difficulty build and deploy Machine Learning powered applications.

  TensorFlow present several levels of abstraction as a result you could choose the right one for your needs. Build and train model by means of the high-level Keras API, which make getting ongoing by way of TensorFlow and machine learning simple.

If you need more suppleness, eager execution allows for immediate iteration and intuitive debugging. For big ML training tasks, make use of the Distribution Strategy API for distributed training on the dissimilar hardware configurations with no changing the model definition.

- **Keras:**

Keras is a high-level neural networks library written in python that works as a wrapper to TensorFlow. It be used in cases where we would like to quickly build and test the neural network with least lines of code. It contains implementations of normally used neural network elements like layers, objective, activation functions, optimizers , and tools to create working with images and text data easier.

- **OpenCV:**

OpenCV (Open-Source Computer Vision) is an open-source library of programming functions used for real-time computer-vision.

It is mostly used for image processing, video capture and analysis for features like face as well as object recognition .which is its primary interface, however bindings are obtainable designed for Python, Java.

- ## Methodology:

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

Proposed Methodology, The overall system comprises of two sections, back end and front-end. The back end framework comprise of 3 modules: Camera module, Detection module and Interface module as appeared in Fig. 8. They are summed up as follows:

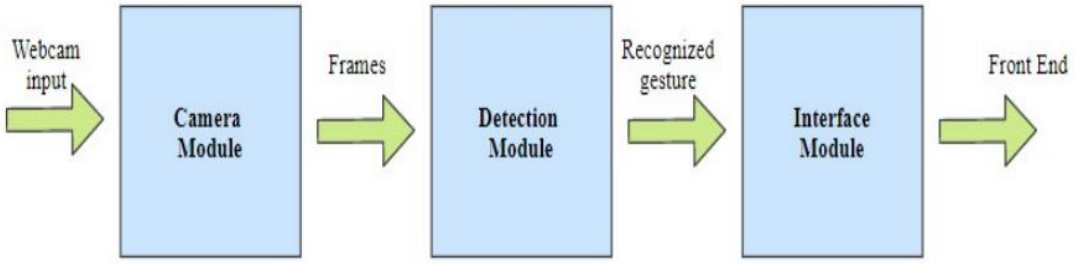


Figure 8 Back And Architecture

- **Camera module**

This module is focus for interfacing and capturing input from side to side of the different sorts of picture markers and sends this picture to the detection module designed for handling as frame. The normally utilized techniques of capturing along with recognizing input are hand belts, data gloves and cameras. In our framework, we use the inbuilt webcam which is financially sense to see static signs.

- **Data Set Generation:**

  For the project, we tried to get already made datasets but we couldn't find dataset in the variety of raw images that match our requirements. All we can find the datasets in the form of RGB values.

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

  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:

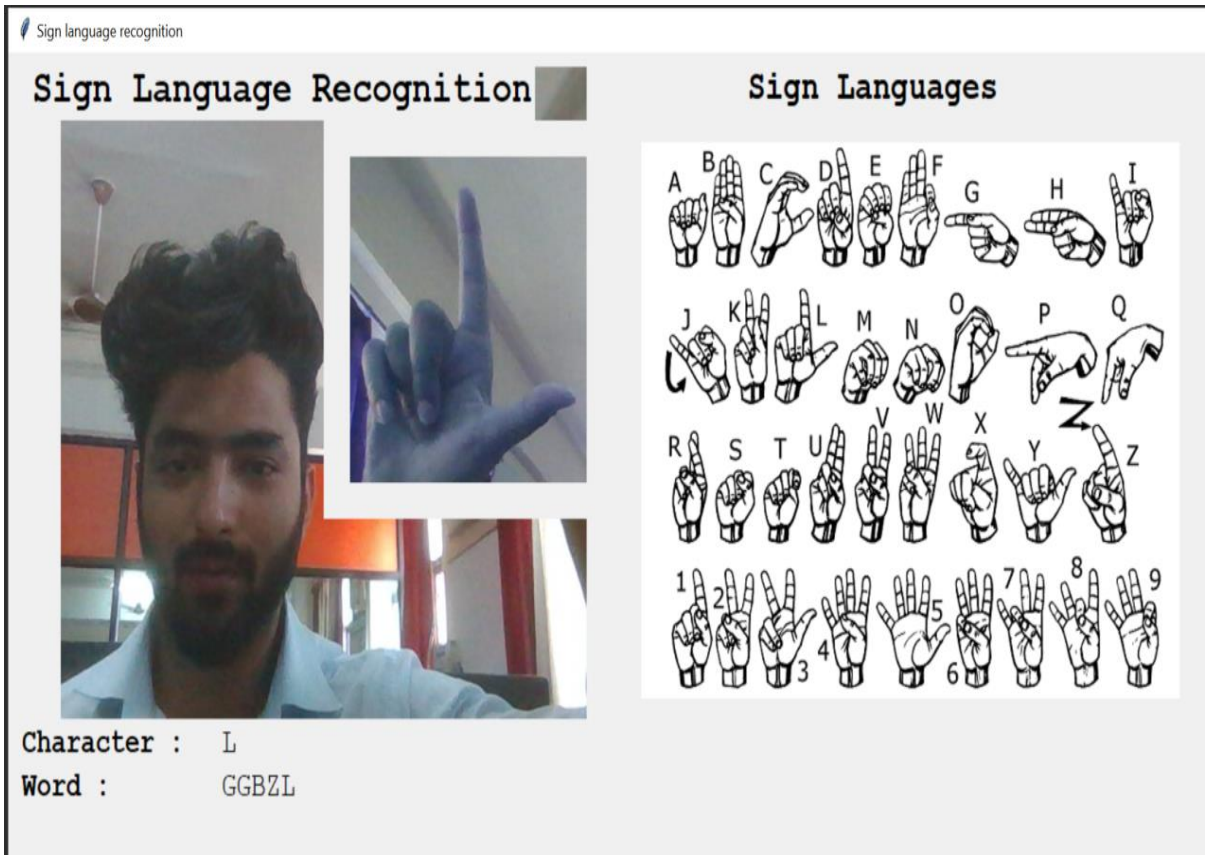


Figure – 7 Detection

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:

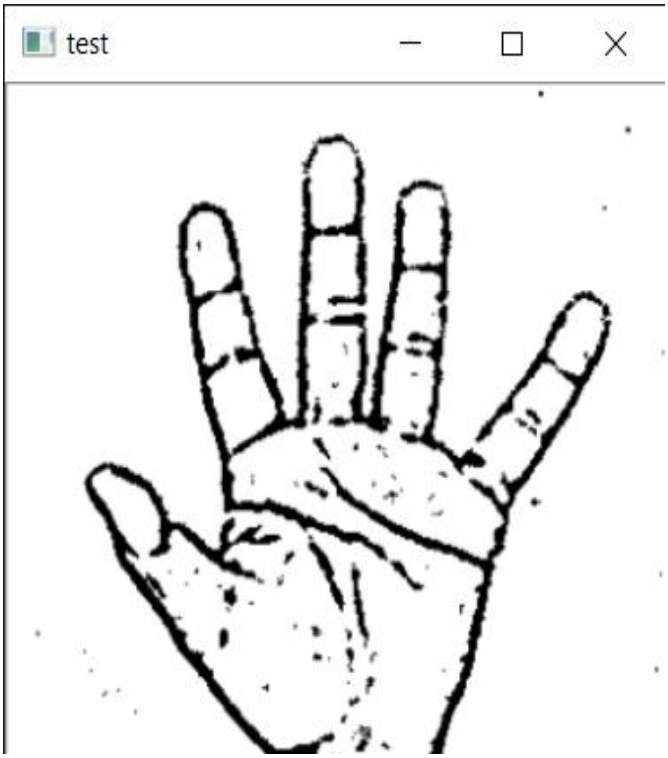


Figure – 8

- ## Gesture Classification:

Detection Module

This module is responsible for the image processing. The output from camera module is present on the way to different images handling methods, for case in point, color conversion, noise removal, thresholding following which the image goes through contour extraction. In the event that the image contains defects, at that point convexity defects are found by which the gesture is recognized. In the event that there are no defects, at that point the image is classified .Our approach uses layer of algorithm to predict the final symbol of the user.

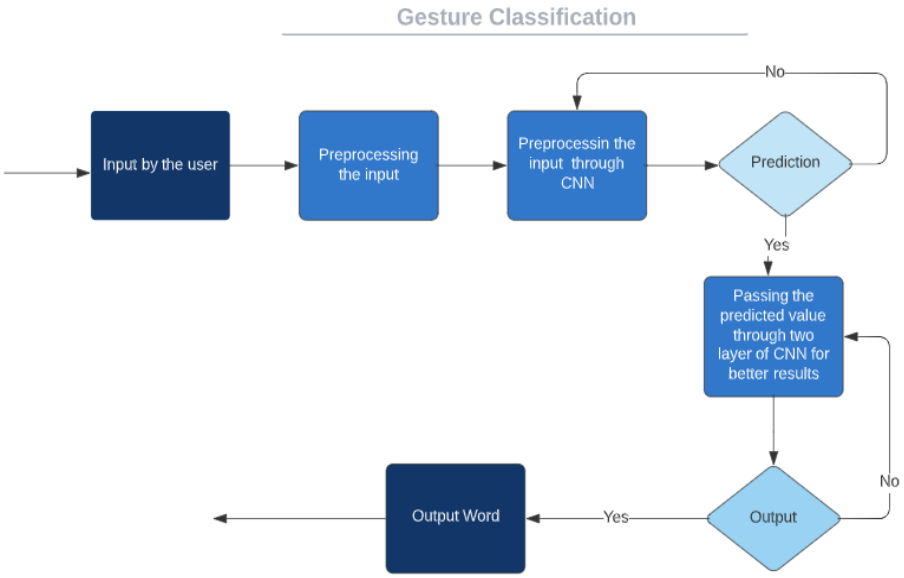


Figure – 9 Gesture Classification

- **Algorithm Layer :**
- Apply Gaussian Blur filter and threshold to the frame taken with OpenCV to get the process image after feature extraction.
- This processed image is accepted to the CNN model for prediction and if a letter is detected for more than fifty frames then the letter is printed and taken keen on deliberation for forming the word.
- Gap between the word is considered using the blank symbol.

Layer :

- **CNN Model:**

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

- **1st Pooling Layer:** The pictures are down sample using max pooling of 2x2 . We remain the highest value in the 2x2 square of array. Therefore, our picture is down sample to 63x63 pixels.
- **2nd Convolution Layer:** Now, these 63 x 63 from the production of the first pooling layer is serve as an input to the 2nd convolutional layer. It is process in the second convolutional layer using 32 filter weights (3x3 pixels each). This will result in a 60 x 60 pixel image.
- **2nd Pooling Layer:** The resulting images are down model again using max pool of 2x2 and are reduce to 30 x 30 resolution of images.
- **1st Densely Connected Layer:** Now these images be use as an input to a fully connecte layer with 128 neurons and the output from the second convolutional layer is reshape 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.
- **2nd Densely Connected Layer:** Now the output beginning the 1st Densely Connected Layer is used as an input to a completely connected layer with ninty-six neurons.
- **Final layer:** The output of the 2nd Densely Connected Layer serve as an input for the final layers which will contain the number of neurons as the numbers of class 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 every 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 over-fitting.

- **Dropout Layers:**

  The problem of over-fitting, 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.

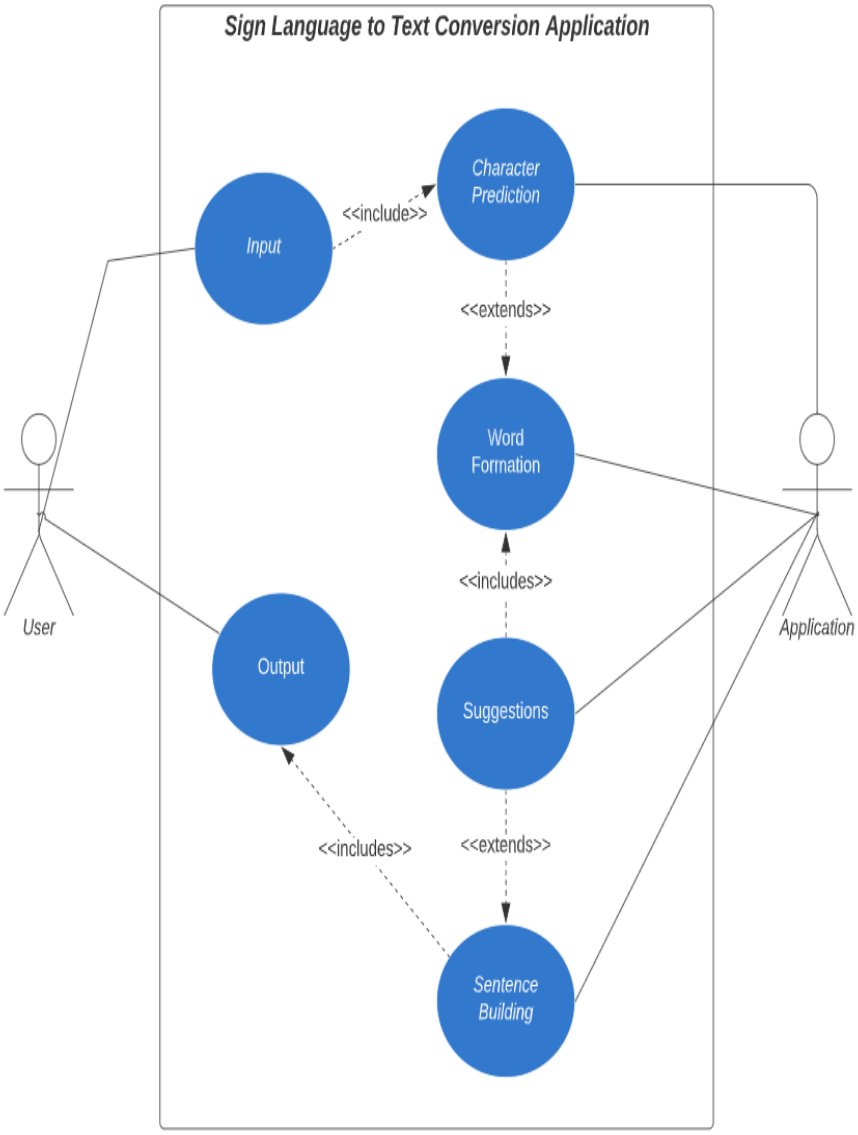


Figure – 10

- **Training and Testing:**

We provide for the input images after pre-processing to our model for training and testing after apply all the operations mentioned above.

The prediction layers estimate how expected the image will drop under one of the classes. So, the output is normalized between 0 and 1 such that the sum of each value in each class sums to 1. We have achieved this using function.

At first the output of the prediction layer will be somewhat far from the actual value. To make it better we have train the networks using labelled data as mention. The cross-entropy is a performance measurement used in the classification. It is a continuous function which is positive at values which is not same labelled value is zero exactly when it is equal to the labelled value. Therefore, we optimized the cross-entropy by minimizing it as close to zero. To do this in our network layer we adjust the weights of our neural networks. Tensorflow has an in-built function to calculate the cross entropy.

- **Challenges Faced:**

There were several challenges faced during the project. The very first issue we faced was that concerning and gathering of the data set. We wanted to deal with the images and those too square images as CNN, since it is much more convenient working with only square images.

We couldn't discover any existing knowledge data set as per our needs and thus we have a tendency to determine to create our own knowledge set. Second issue to select a filter which we could apply on our images so that proper features of the images could also be obtained and hence then we have a tendency to may offer that image as input for CNN model.

We tried a range of filters including binary threshold, smart edge detection. A lot of problems we have the tendency. More issues were faced relating to the accuracy of the model we had train in the previous phases. This problem was ultimately improved by ever-increasing the input image size and also by inhance the data set.

- This technology is one that enables a disabled community to supplementary integrate into an abled community, and may be viewed and bending to the rules of privileged community. This may reduce efforts of hearing people to accommodate for deaf people. • The dataset needed to be diverse enough in order to accommodate folk people of all skin tones and in all environments. A bias in data could perhaps disadvantage deaf people of a certain ethnic group.

- **Results:**

We have achieved an accuracy of **89%** in our model using only layer of our algorithm which is CNN, and using the arrangement of layers , which is a better accuracy then most of the current research papers . .

In [7] they construct a detection system for sign language detection system using convolutional neural networks as well as achieve an error rate of **2.5%.** .

They additionally used CNN for their recognition system. One factor ought to be noted that our model doesn't use any background subtraction algorithm whiles some of the models presented on top of 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 devices but our main aim was to create a project which can be used with readily available for resources. A sensor like not only isn't readily available for the users but also is costly for most of users to purchase along with our model uses a normal webcam of the laptop hence it is great plus point.

Below there is the confusion matrices for our results.

A confusion matrix gives the outline of prediction results on a classification problem. Every row corresponded to actual category class and every column of the matrix corresponds to a predicted class. It is desirable that a diagonal are obtained across the matrix, which suggests that categories are properly expected to be predicted.

						Predicted Values																			
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
	A	147	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	2	0	0
	B	0	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0
	C	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	D	0	0	0	145	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	E	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	F	0	0	0	0	0	135	0	0	0	0	0	4	0	0	0	0	0	1	0	0	2	10	0	0	0
Correct Values	G	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	H	1	0	0	0	0	0	7	143	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
	I	0	0	0	33	0	0	0	0	108	0	2	0	0	0	0	0	0	0	0	7	1	0	0	0	0
	J	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	K	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	L	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0
	M	0	0	0	0	0	0	0	0	0	0	2	0	152	0	0	0	0	0	0	0	0	0	0	0	0
	N	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0
Values	O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	0	0
	P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0
	Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	147	1	0	0	0	0	0	0	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0
	S	0	0	0	0	1	0	0	0	0	0	0	0	0	1	10	0	0	0	132	0	0	0	0	8	0
	T	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	151	0	0	0	0	0
	U	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0	0	115	0	0	0	0
	V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	1	0	0
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	149	0	0
	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	148	0
	Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151
	Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Algo 1

Figure – 11 Matrix

- **Conclusion:**

In this report, a purposeful real time vision based mainly American Sign Language recognition for D&M people have been developed for ASL alphabets. An American Sign Language Interpreter has been supported based on a Convolutional Neural Network (CNN). The system requirements some constraints, like a white background and thefore the palm of the hand to face the camera. Additional related symbols were sometimes misinterpreted for one another. This could be due to the limited training provided to the system. The system might overcome these limitations if a more detailed dataset in dissimilar environmental conditions is provided for training set. Also letters that required movement of the hands could'nt be properly identified.

We achieved final accuracy of **89%** on our knowledge 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 detected almost all the symbols provided that they are shown properly, there is no noise in the background and lighting is adequate

This system helps many dumb and deaf people. Though this CNN model is efficient, it needs more computations for building and loading model. The proposed model takes input image and recognizes the sign language. We proposed the model with an accuracy of 91.53%. In future work we can work on recognizing sentences and add voice system words in real time to ease the communication between the deaf or dumb and thefore the normal person.

- **Future Scope:**

  We are planning to achieve higher accuracy even in case of complex backgrounds by applying double or second layer for similar Alphabet Sign.

  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 will be extended to work for different native sign languages with the right amount of data set and training. This project implements a finger spelling translator; however, the sign languages are also spoken in a contextual basis where each gesture might represent an object, or verb. So, identify this kind of a signing would require a higher degree of process and natural language process (NLP).

- **References:**

1. T. Yang, Y. Xu, and “A., Hidden Markov Model for Gesture Recognition”, CMU-RI-TR-94 10, Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, May 1994.
2. Pujan Ziaie, Thomas M uller, Mary Ellen Foster, and Alois Knoll “A Na ̈ive Bayes Munich, Dept. of Informatics VI, Robotics and Embedded Systems, Boltzmannstr. 3, DE-85748 Garching, Germany.
3. https://docs.opencv.org/2.4/doc/tutorials/imgproc/gausian_median_blur_bilateral_filter/gausian_median_blur_bilateral_filter.html
4. Mohammed Waleed Kalous, Machine recognition of Auslan signs using PowerGloves: Towards large-lexicon recognition of sign language.
5. aeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural Networks-Part-2/
6. http://www-i6.informatik.rwth-aachen.de/~dreuw/database.php
7. Pigou L., Dieleman S., Kindermans PJ., Schrauwen B. (2015) Sign Language Recognition Using Convolutional Neural Networks. In: Agapito L., Bronstein M., Rother C. (eds) Computer Vision - ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science, vol 8925. Springer, Cham
8. Zaki, M.M., Shaheen, S.I.: Sign language recognition using a combination of new vision-based features. Pattern Recognition Letters 32(4), 572–577 (2011).
9. N. Mukai, N. Harada and Y. Chang, "Japanese Fingerspelling Recognition Based on Classification Tree and Machine Learning," *2017 Nicograph International (NicoInt)*, Kyoto, Japan, 2017, pp. 19-24. doi:10.1109/NICOInt.2017.9
10. Byeongkeun Kang, Subarna Tripathi, 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)
11. Number System Recognition (https://github.com/chasinginfinity/number-sign-recognition)

12. https://opencv.org/
13. https://en.wikipedia.org/wiki/TensorFlow
14. https://en.wikipedia.org/wiki/Convolutional_neural_nework
15. http://hunspell.github.io/

- **Appendix:**
- **OpenCV:**

OpenCV-Python could be a library of Python bindings designed to unravel laptop vision issues. Python could be a general purpose programing language .openCV(Open-Source laptop VisionLibrary) is discharged below a BSD license and therefore it's free for each tutorial and industrial use. It has Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and humanoid. OpenCV was designed for machine potency and with a powerful concentrate on period applications.Written in optimized the library will profit of multi-core process. Enabled with OpenCL, it will profit of the hardware acceleration of the underlying heterogeneous cipher platform. Adopted all round the world, OpenCV has over fortyseven thousand folks of user community of and calculable range of downloads olympian fourteen million. Usage ranges from interactive art, to mines review, sewing maps on the net or through advanced AI.

- **Convolutional Neural Network:**

CNNs use a variation of multilayer perceptron's designed to need stripped pre-processing. they're additionally called shift invariant or area invariant artificial neural networks(SIANN), supported shared-weights design and translation unchangeableness characteristics.

Convolutional networks were impressed by biological processes in this the property pattern between neurons resembles the organization of the animal visual area. Individual plant tissue neurons reply to stimuli solely during a restricted region of the sight view called the receptive field. The receptive fields of various neurons part overlap such they cowl the complete sight view.

CNNs use comparatively very little pre-processing compared to different image classification algorithms. this suggests that the network learns the filters that in ancient algorithms were hand-engineered. This independence from previous information and human effort in feature style could be a major advantage.

They have applications in image and video recognition, recommender systems, image classification, medical image analysis, and tongue process.

- **TensorFlow:**

TensorFlow is associate degree ASCII text file computer code library for dataflow programming across a variety of tasks. it's a symbolic science library, and is additionally used for machine learning applications like neural networks.

It is used for each analysis and production at Google.

TensorFlow was developed by the Google brain team for internal Google use. it absolutely was free beneath the Apache two.0 ASCII text file library on Gregorian calendar month nine, 2015.

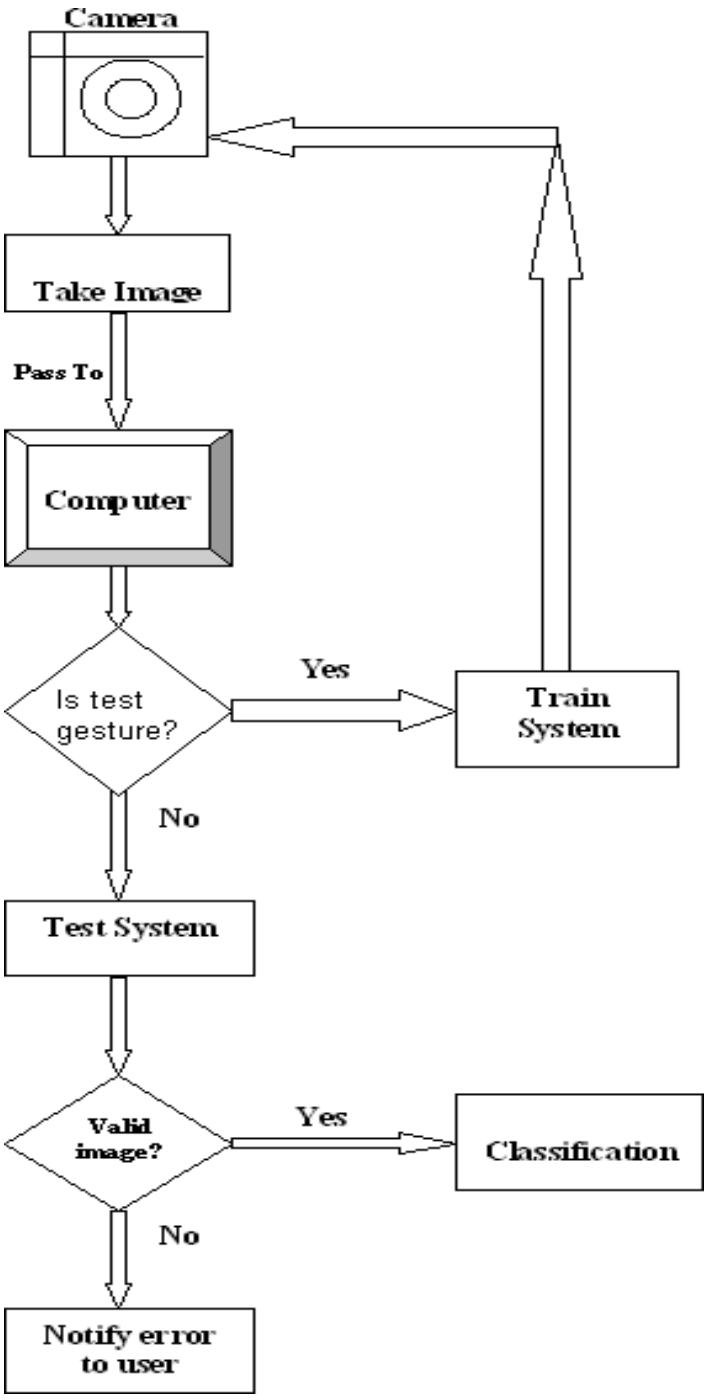
TensorFlow is Google Brain's second-generation system. Version 1.0.0 was free on February eleven, 2017. whereas the reference implementation runs on single devices, TensorFlow will run on multiple CPUs and GPUs (with elective CUDA and SYCL extensions for general computing on graphics process units).

TensorFlow is obtainable on 64-bit UNIX, macOS, Windows, and mobile computing platforms together with mechanical man and iOS.

Its versatile design permits for the straightforward readying of computation across a spread of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.

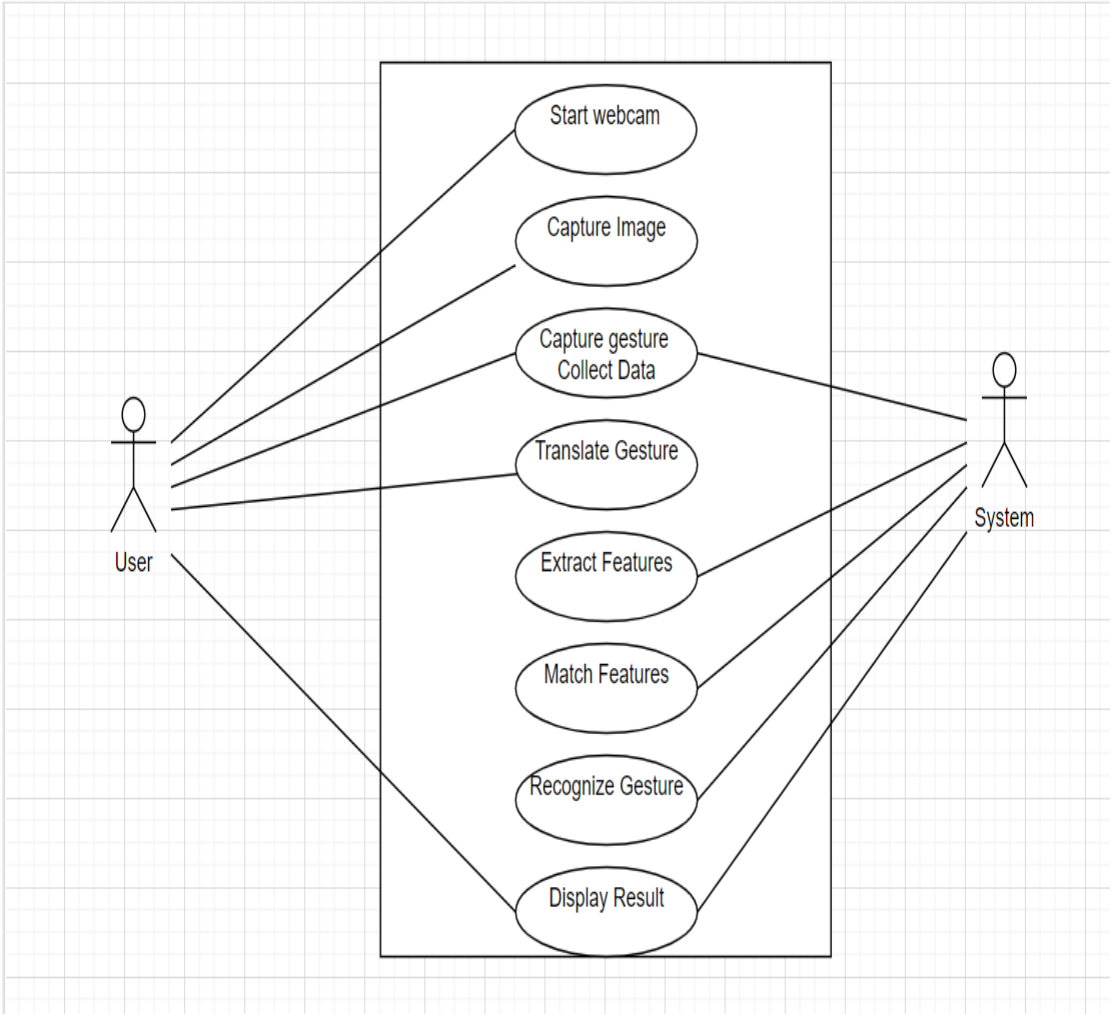
Appendix

Data-Flow Diagram (DFD)

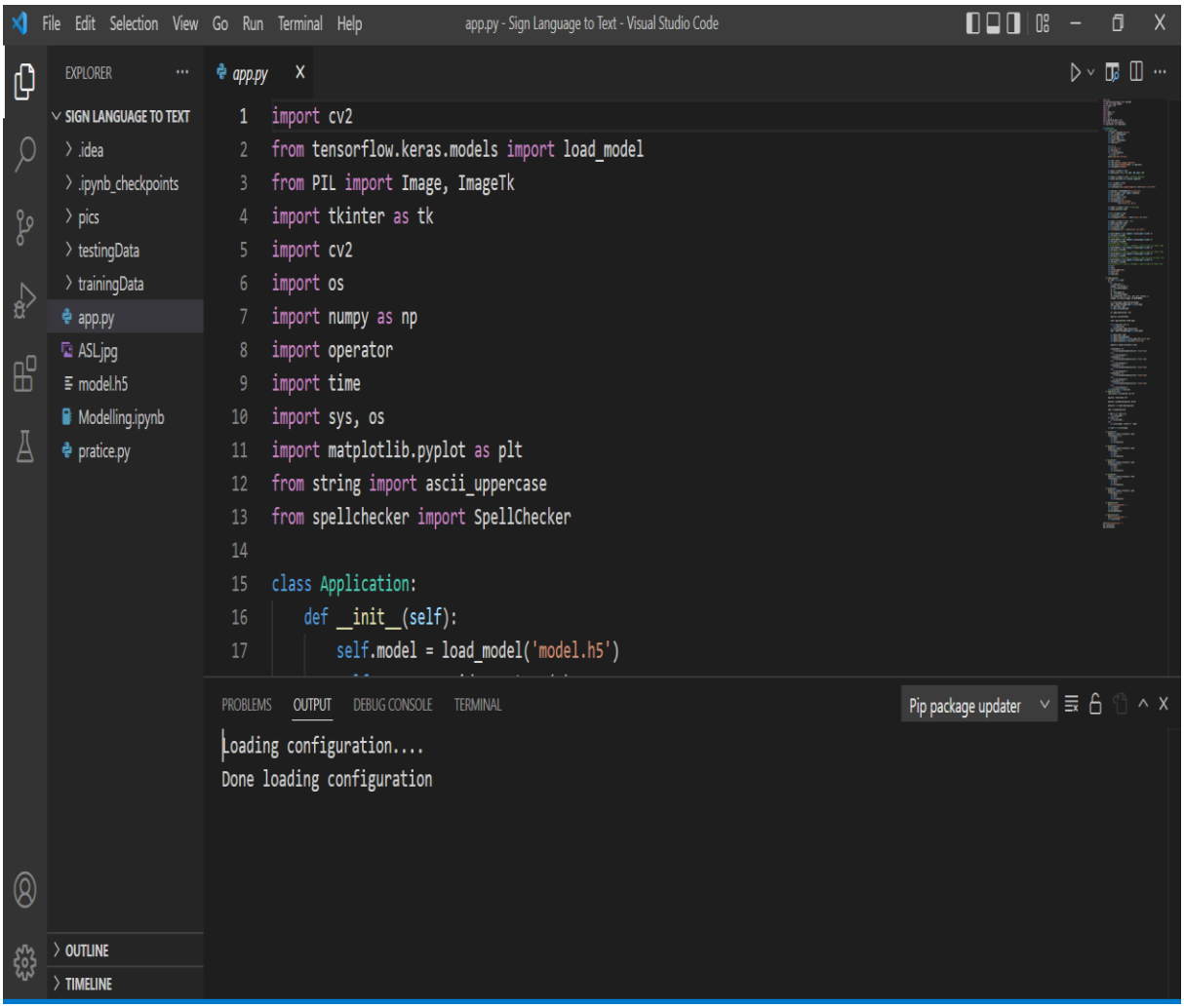


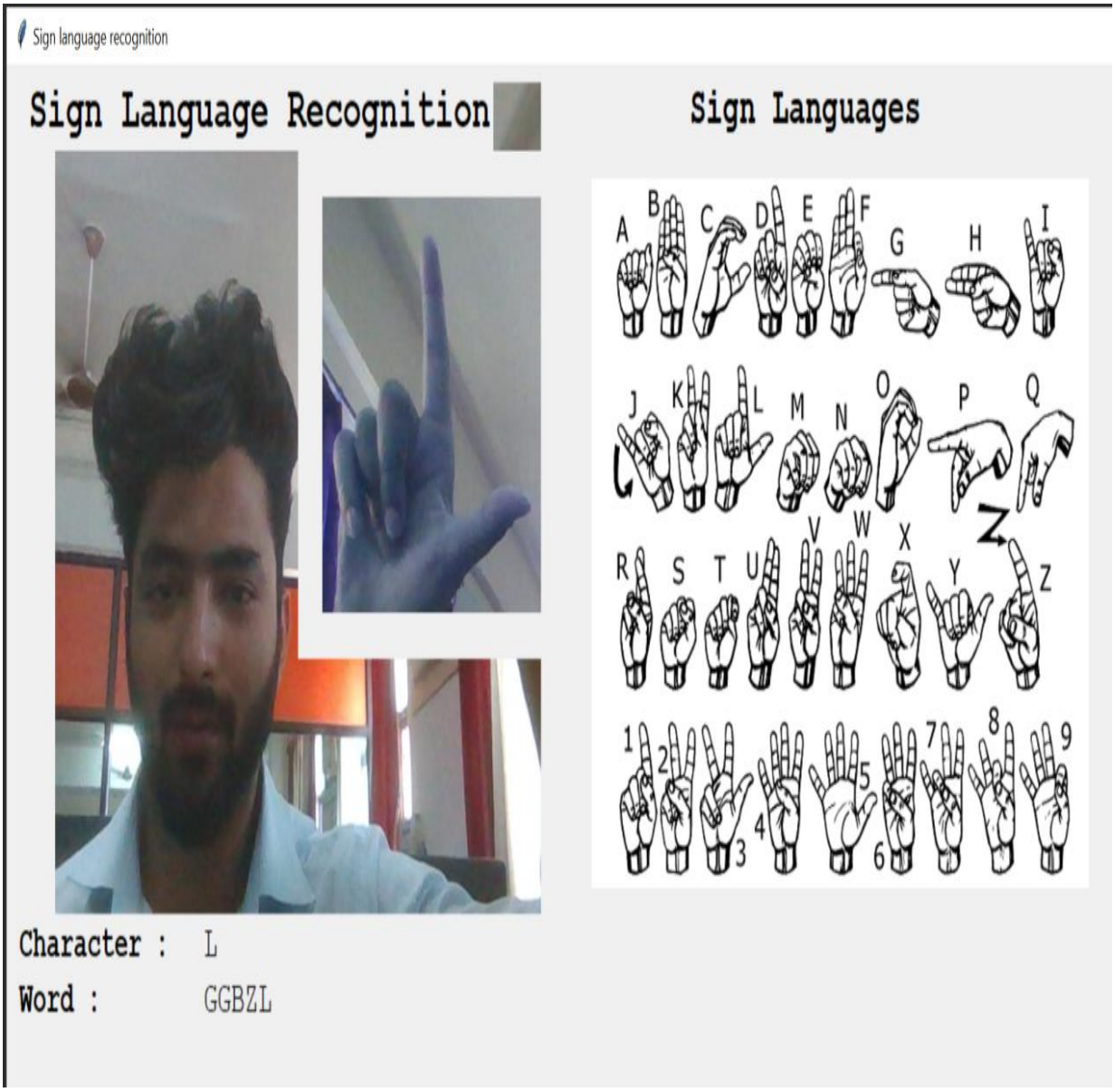
Appendix

Use-case Diagram (UCD)



- **Screenshot**


File Edit Selection View Go Run Terminal Help
app.py - Sign Language to Text - Visual Studio Code
EXPLORER
app.py
SIGN LANGUAGE TO TEXT
.idea
.ipynb_checkpoints
pics
testingData
trainingData
app.py
ASL.jpg
model.h5
Modelling.ipynb
pratice.py
1 import cv2
2 from tensorflow.keras.models import load_model
3 from PIL import Image, ImageTk
4 import tkinter as tk
5 import cv2
6 import os
7 import numpy as np
8 import operator
9 import time
10 import sys, os
11 import matplotlib.pyplot as plt
12 from string import ascii_uppercase
13 from spellchecker import SpellChecker
14
15 class Application:
16 def __init__(self):
17 self.model = load_model('model.h5')
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Pip package updater
Loading configuration....
Done loading configuration
OUTLINE
TIMELINE


Sign language recognition
Sign Language Recognition
Sign Languages
Character : L
Word : GGBZL