# Conversation Engine For Deaf And Dumb

## ABSTRACT

In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. About nine billion people in the world are deaf and dumb. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language. In emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.

The  project aims to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as converting speech into understandable sign language for the deaf and dumb. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.

# 1. INTRODUCTION

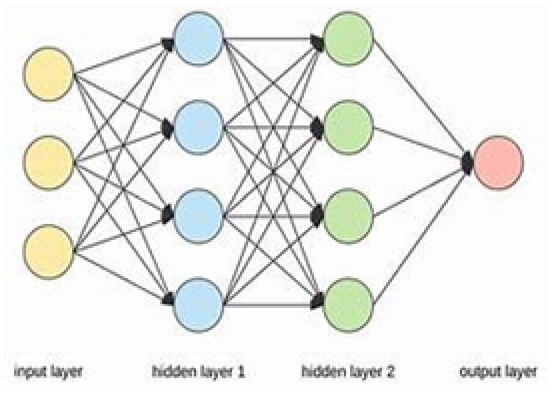
## 1.1. Introduction

### What is sign language?

Deaf and dumb people use sign language as their primary means to express their thoughts and ideas to the people around them with different hand and body gestures. Although sign languages have emerged naturally in deaf communities alongside or among spoken languages, they are unrelated to spoken languages and have different grammatical structures at their core.

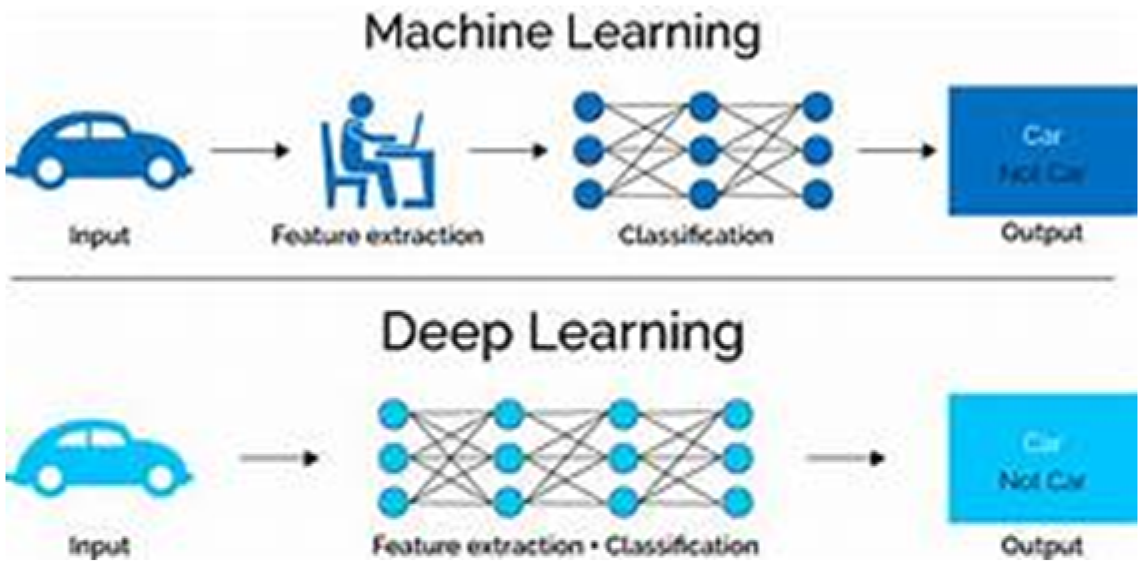
### What is CNN?

Convolution Neural Networks is a part or subset of deep neural technology and it is based on neurons which take image as input in the apperance of pixels and it has many layers like input layer, hidden layers and output layer. Each neuron in one layer connects to other neurons in another layer and takes weight as sum and produces the output finally. CNN is very popular in the study or interest of computer domains and it is mainly used for image classification, detection of objects and image segmentation etc. weights are important in terms of output and it influences the image input on the output, similarly bias are constants and they are extra terms added to subsequent layer.



### How Computer Reads an Image?

If we observe any images there is very small boxes that forms the entire image and those little square boxes are called by us as pixels. Each pixel has rgb colors that varying from 0 to 255. Therefore system understands the given image in the well-known form of 0's and 1's that is nothing but binary language which the system can understand easily. There were different number of channels in each pixel and the image considered as bunch of pixels.



## 1.2. Objectives

The main goals from this research are:

- to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people
- To develop an app that enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.

## 1.3 Methodology:

1. Data Collection.
2. Data Exploration and analysis.
3. Data Preprocessing.
4. Data Splitting.
5. Training the model.
6. Testing the model.

# 2. IMPLEMENTATION

## 2.1. Dataset

TABLE 2.1: DATASET INFORMATION

| **Dataset** | **Number of Images** |
|---|---|
| **Testing Dataset** | 2250 |
| **Training Dataset** | 15750 |

Initially all the images in the dataset are of different sizes and the problem is all the images used to train, validate and test the CNN model should have same size. So the simple solution to this problem is to resize all the images to same size in the dataset. Here, the size is taken as 64 x 64 for all images in the dataset. This will help us to decrease the training time as well as computational power to train the model.

## 2.2. Layers in CNN

The convolution neural network is constructed with the help of small units which are called as nodes or neurons and every node has some weight and they are changed in the training time using various optimization techniques. The model has two parts *i.e* the first phase is feature extraction and it is done using pooling and convolution layers and the second part is whatever the features extracted in the first part is given as input to dense layer which classifies the input and produces the output. Some of the layers in CNN are:

1. **Convolution Layer:** the convolution layer is used to extract features from the image which is given as input. The convolution operation is performed like this, we take a filter that is capable to identify the features and the filter is passed over the image and at each pass the values in the filter are multiplied with the values overlapped on the image and it performs sum of all the values and this outputs an image with reduced size showing the extracted features. Here for filter we use stride that is moving the one square box at a time in any direction if stride is taken as one.

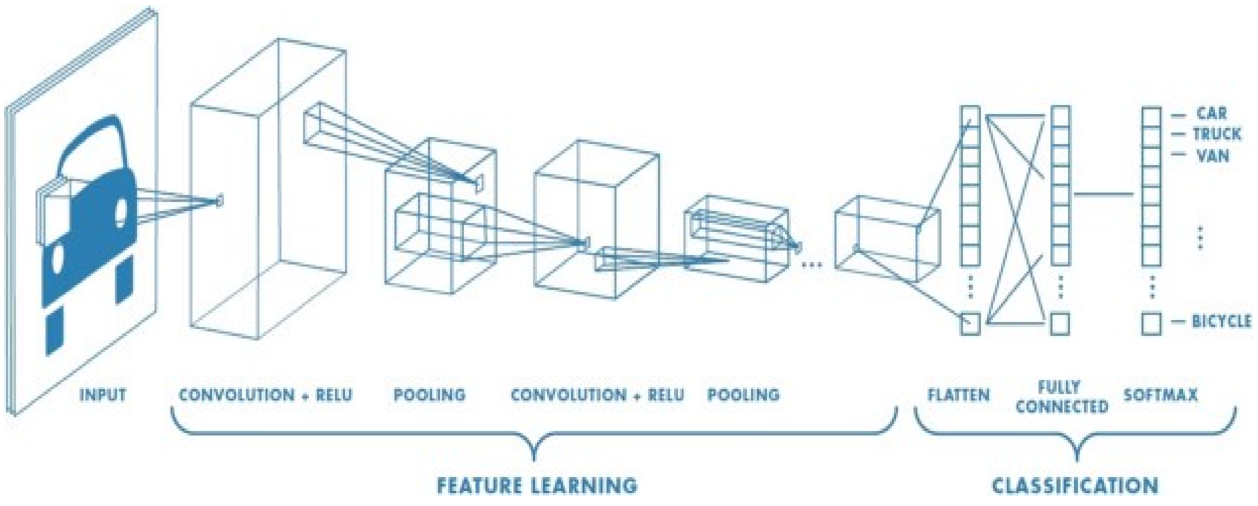
2. **Batch Normalization:** this is used at the end of each convolution layer to maximize the rate of learning and it generally normalizes the image which is given to the CNN model like input.

3. **Pooling Layer:** it is similar to the above layer and it is also very helpful to extract main or dominant features of the image and it is very helpful to minimize the training time and it is also used to decrease the time for computing. There are 2 types in this layer one is max pooling which takes maximum value in a particular stride and other one is average pooling which takes average value of all the values present in a stride.

4. **Activation :** while constructing the CNN model people generally use the activation function and the mostly used one is Relu called as Rectified Linear Unit and what it does is it removes all negative values and replace them with zero and no effect on positive values and it is helpful to fire a neuron. Another activation function used is dense sigmoid function which is used at end of the model.

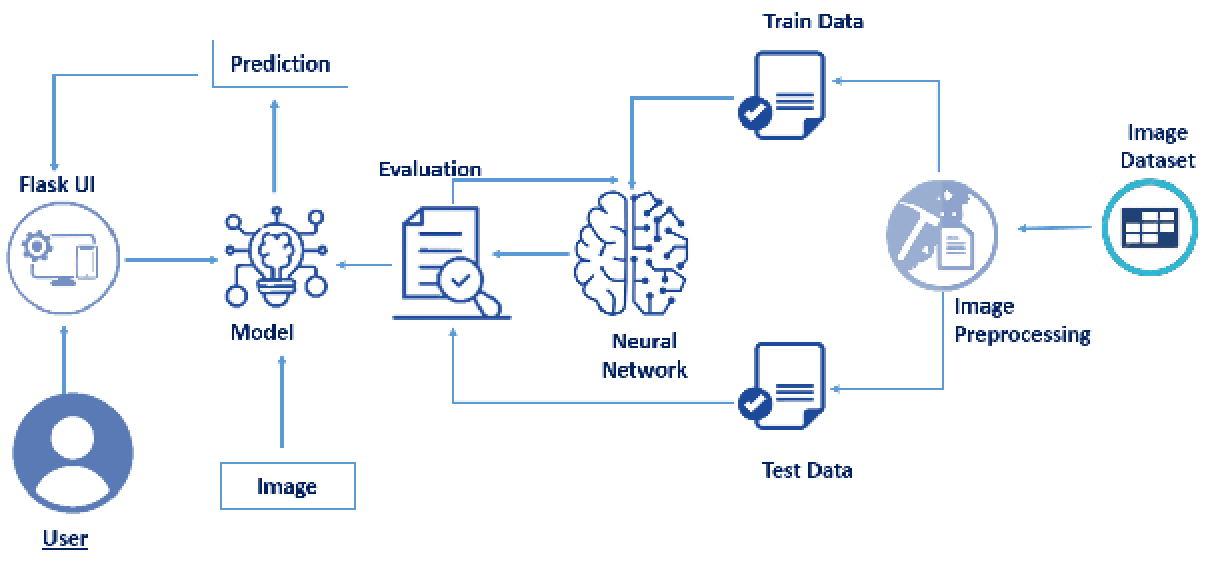
5. **Dropout:** The dropout layer is used to inactive some neurons at a particular epoch and may be active at some time and this used to decrease one thing that is overfitting. In this research we used value 0.5 for dropout.

6. **Dense Layers:** Normally we use 2 dense or fully connected layers in the network and these dense taken the output of pooling and convolved layers and classification is done. The name dense is why because it is the layer which connects to all neural units or nodes in previous layer because it is fully connected. The input is changed in the form of one dimensional as dense layer accepts vector as input and it doesn’t accepts other than string format.



## 2.3. Architecture

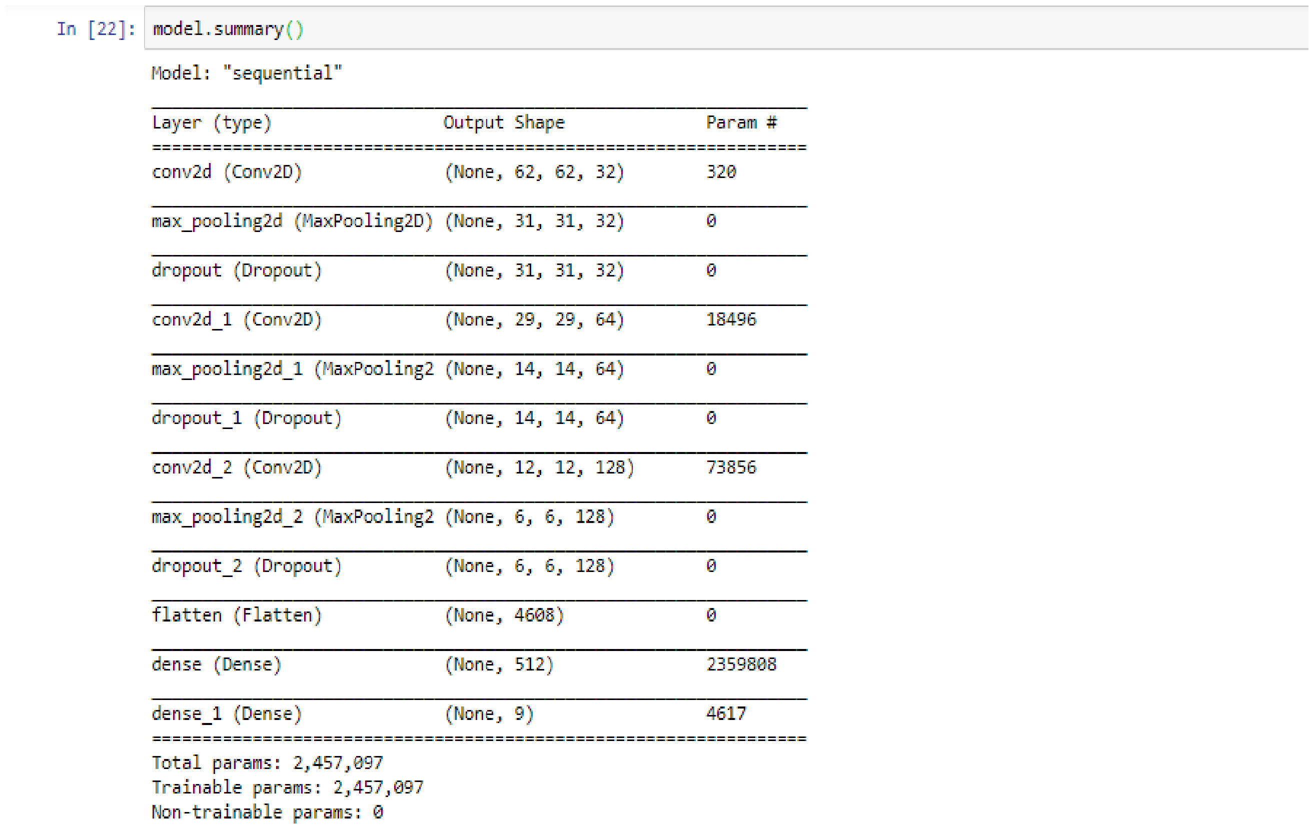
We here were proposing two models of convolution neural networks that is one with the presence of dropout layer and other with the absence of that concept. The common layers present in these two architectures are pooling layer, convolution layer and classification layer. The first part is extraction of features and for this pooling and convolution layers takes part. The job of feature extraction is divided into 2 parts. There are 2 convolution layers in the first part and the size of this is 32 – 32 units and with the convolution layer a max pooling layer (3 x 3) and Relu activator. The next part contains 2 convolution layers and they are of size 64, 128 and max pooling (2 x2) and Relu activator.



The extracted features in the first phase *i.e* output of feature extraction phase are given as input to the fully connected layer and it is also called as dense layer because of the reason that in model each neuron in this layer is connected to every neuron in the past layer. The extracted features are not directly given to the dense layer an introduction to flatten layer is used before passing the features to dense layer because it accepts only vector that is one dimensional array as input. Flatten layer converts the features as one dimensional array and give them to fully connected layer.

The dropout layer is used in the model why because the output from the layers of CNN depends on some neurons in other layer and it is dependent precisely only on some of those neural units. Every neuron at some particular epoch in the period of training time with the help of probability it is dropped at that time. This is very helpful to reduce the over fitting in the model and it restricts the model to depend on only some units and with this the model gives accurate results. The neurons which are inactive mean dropped one is not dropped all the times and it may be active in the next subsequent steps. If we considered the probability of dropout as 0.6 then it means 60% of the neural units are dropped. In our model the concept of dropout layer is applied two times and if we observe the figure below it shows that in the time of feature extraction the dropout takes place with a probability value of 0.2 and it is done at the completion of pooling and convolution layers. Here we use the probability only 0.2 because in the feature extraction it is not that much required because it has no that many parameters and due to this there is a less chance of getting the overfitted model here. In the feature extraction part the noises will be not considered because of dropout and it helps to reduce over fitting.

When a dropout of 0.5 used in the model in the classification phase it gives the better performance compared to others. So we introduced the dropout of 0.5 in the classification part and it is shown in the fig below. We used it in the dropout layer in our model to improve the performance and as expected it gives the results with higher accuracy.



```
In [22]: model.summary()

Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 62, 62, 32)        320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 31, 31, 32)        0
_________________________________________________________________
dropout (Dropout)            (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 29, 29, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 14, 14, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 14, 14, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 12, 12, 128)       73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 6, 6, 128)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 6, 6, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 4608)              0
_________________________________________________________________
dense (Dense)                (None, 512)               2359808
_________________________________________________________________
dense_1 (Dense)              (None, 9)                 4617
=================================================================
Total params: 2,457,097
Trainable params: 2,457,097
Non-trainable params: 0
```

## 2.4. Augmentation

If we train the model with less dataset then this leads to overfitting. Consider an example of a model that classifies whether the input is dog or not, if we train the model using dog images which are turned towards right only and when we give new image of dog and it is turning towards left the model is not able to classify the image, this comes under over fitting. To expand the images present in the dataset without collecting new images we required several augmentation techniques are used. Due to this augmentation the CNN model encounters various new images at each epoch in the training time. Some of the augmentation techniques used is rescaling, translation, flipping, rotation, cropping etc. in this model. Some of the techniques are listed in the below table.

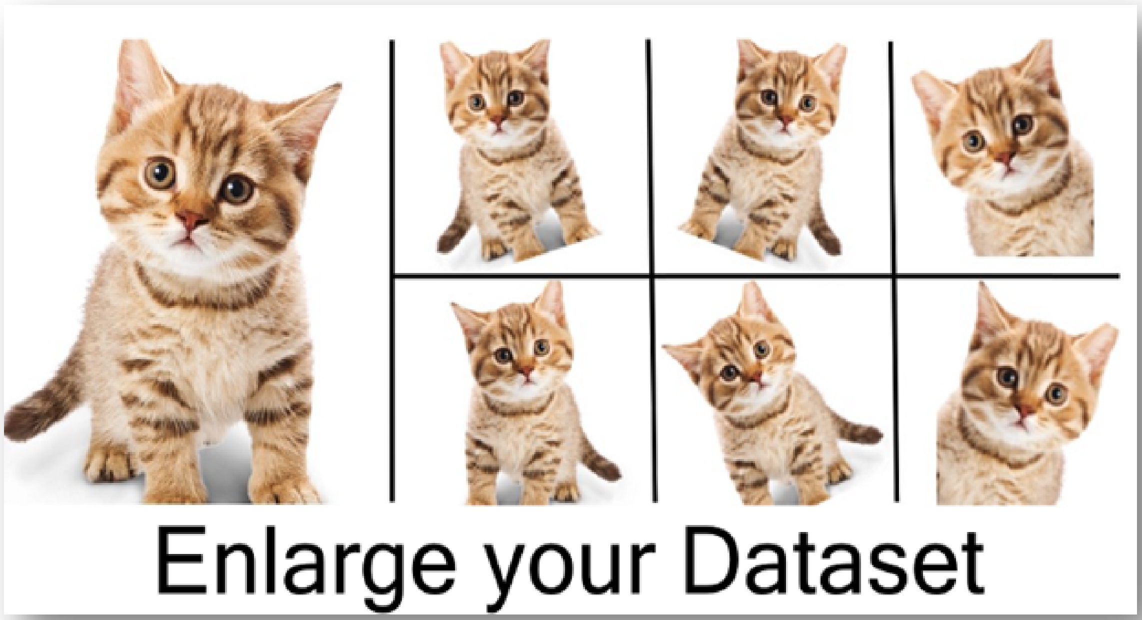
TABLE 2.2: DATA AUGMENTATION TECHNIQUES

| **Augmentation Techniques** | **Values** |
|---|---|
| Rotation Range | 20 |
| Fill Mode | Nearest |
| Rescale | 1.0/255 |
| Zoom Range | 0.2 |
| Shear Range | 0.2 |
| Height Shift Range | 0.2 |
| Horizontal Flip | True |
| Width Shift Range | 0.2 |

Some of the augmentation techniques are:

- **Flipping –** We can flip the images that we used in this experiment in horizontal as well as in vertical direction. Rotating the image by 180 degrees is considered as vertical flip.
- **Rotation** – In this we can rotate the image in clockwise or anticlockwise by 90 degrees and the image size may change if we rotate it by finer angles.
- **Translation** – In this technique the image is moved along horizontal axis or vertical axis or we can move the object in image in both directions.
- **Scaling** – if we perform scaling technique on the image the image size varies and if we do outward scaling the image size will grow.

- **Crop** – in this we select a part of image and this part can e resized to normal size of image and there is a difference between scaling and cropping.
- **Zoom** – The images in the collection can be enlarged by 2 or 3 times than original images in the dataset or it can be either zoom in or zoom out.
- **Shearing** – In this technique some minor part of picture is shifted in the shape of parallelogram and it is one type of bounding box technique.



Images of cat using Augmentation Techniques

If we observe the image above, in the left we have original image of cat and the six images which are on the right side are produced from the image on the left without collecting new images of the cat and by using various augmentation techniques described above. Most of the convolution neural networks models used this type of augmentation techniques to enlarge their dataset and this enhancement in the dataset result in better models with high accuracy in the role of classifying the image.

# 3.REQUIREMENTS

## 3.1 SOFTWARE REQUIREMENT SPECIFICATIONS:

A document that specifies the nature of a software model or project is commonly referred to as Software Requirement Specifications (SRS). This can be stated as manual of project and is prepared before proceeding onto the project. We have to follow some important guidelines in preparing an efficient SRS document which can be understood easily. This consists of scope, purpose, functional and non-functional requirements of software application as well as requirements of software and hardware. This also consists of details regarding conditions of environment, safety, security, quality attributes etc.

**Hardware Requirements:**

| | |
|---|---|
| Processor | : minimum core i3 processor |
| RAM | : minimum 2GB or 8GB maximum |
| Hard Disk Space | : more than 50GB |

**Software Requirements:**

| | |
|---|---|
| Operating System | : Windows 8 or later versions of Windows |
| Presentation Tier | : HTML, CSS |
| Anaconda Software | |

## 3.2. TECHNOLOGY DESCRIPTION:

**Python:**

1. Python is an oop language and everyone can understand it easily.
2. It is interpreted, high-level, interactive language which is used worldwide in many applications.
3. The indentation feature of python improves its code readability.

*Syntax and Semantics:*

a. Python doesn't use any curly braces like other languages and it doesn't uses semicolon at end.

b. Python statements includes assignment statement which is like this '='.

c. If statement can be used to execute conditional statements and we can even use elif or else along with that.

d. Using while we can execute statements in the loop until a specified condition is true.

e. for is also used to execute a statement required number of times.

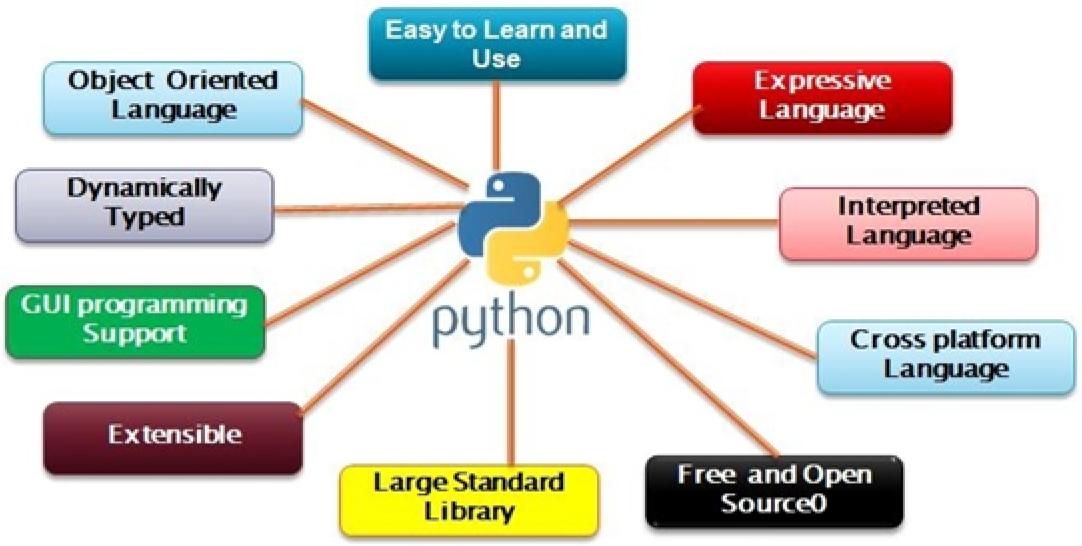
f. Exceptional handling in python can be done using the statements like try and except and we can even use finally block which executes always.

g. def statement in python is used to define methods or functions.

h. Return statement is used to return something from a method or function.

i. class statement is used to define a blueprint to create objects.

j. import statement is useful while importing the modules, functions or classes which are already there.



**Flask web framework:**

In python there is micro web framework which is flask and the template engine used by this framework is jinja.it doesn’t want any particular libraries and it supports extensions which we can add additional features of application.

Features:

1. Extensive documentation.
2. Development server and debugger.
3. Unicode based.
4. Uses jinja templating.
5. Feature enhancement by extensions.

Example program:

```
#first import flask
from flask import Flask
#next instantiate flask applications
app=Flask(__name__)
#using route decorator we can traverse to web pages
@app.route(“/”)
#here it is the root page and the content to be displayed is written in python function
def welcome():
   return “welcome to flask”
```

Execution:

```
>>set FLASK_APP=filename.py
>>flask run
```

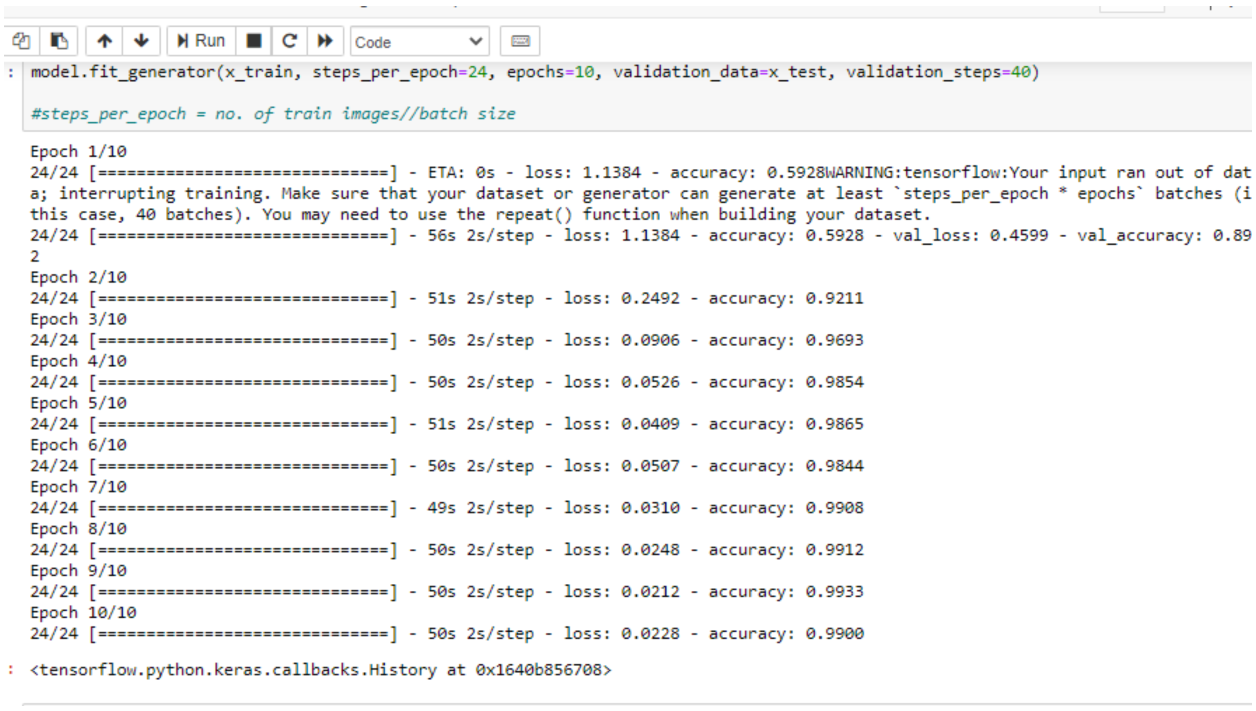
**HTML:**

Hypertext markup language is used as markup language for documents. It is a front-end technology and the web browser receives documents of HTML from server and it defines the structure of a web page. The elements in html are considered as building blocks for web pages.

**<u>CSS:</u>**

Cascading style sheets gives the presentation for the page that is written using HTML. CSS gives presentation, layout, fonts and colors etc. It has many properties regarding styles and the specifications of CSS are maintained by W3C.

## 4. EXPERIMENTAL RESULTS



```
: model.fit_generator(x_train, steps_per_epoch=24, epochs=10, validation_data=x_test, validation_steps=40)

  #steps_per_epoch = no. of train images//batch size

  Epoch 1/10
  24/24 [==============================] - ETA: 0s - loss: 1.1384 - accuracy: 0.5928WARNING:tensorflow:Your input ran out of dat
  a; interrupting training. Make sure that your dataset or generator can generate at least `steps_per_epoch * epochs` batches (i
  this case, 40 batches). You may need to use the repeat() function when building your dataset.
  24/24 [==============================] - 56s 2s/step - loss: 1.1384 - accuracy: 0.5928 - val_loss: 0.4599 - val_accuracy: 0.89
  2
  Epoch 2/10
  24/24 [==============================] - 51s 2s/step - loss: 0.2492 - accuracy: 0.9211
  Epoch 3/10
  24/24 [==============================] - 50s 2s/step - loss: 0.0906 - accuracy: 0.9693
  Epoch 4/10
  24/24 [==============================] - 50s 2s/step - loss: 0.0526 - accuracy: 0.9854
  Epoch 5/10
  24/24 [==============================] - 51s 2s/step - loss: 0.0409 - accuracy: 0.9865
  Epoch 6/10
  24/24 [==============================] - 50s 2s/step - loss: 0.0507 - accuracy: 0.9844
  Epoch 7/10
  24/24 [==============================] - 49s 2s/step - loss: 0.0310 - accuracy: 0.9908
  Epoch 8/10
  24/24 [==============================] - 50s 2s/step - loss: 0.0248 - accuracy: 0.9912
  Epoch 9/10
  24/24 [==============================] - 50s 2s/step - loss: 0.0212 - accuracy: 0.9933
  Epoch 10/10
  24/24 [==============================] - 50s 2s/step - loss: 0.0228 - accuracy: 0.9900

: <tensorflow.python.keras.callbacks.History at 0x1640b856708>
```

## 4.1 Advantages

- This is used to understand sign language easily.
- The results given by the CNN model is accurate.
- It is economical.

## 4.2. Disadvantages

- The over fitting may cause severe damage to humans.
- Wrong predictions by the model may even leads to misunderstanding also.

# 5. CONCLUSIONS

## 5.1. Conclusion

We have developed a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as converting speech into understandable sign language for the deaf and dumb. We have used convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.

## 5.2 Future Scope of Work

In future, instead of using dense layer concept we can use batch normalization and early stopping and we can check how better they restrict the overfitting. To do the job of enhancing accuracy of models we can further use various other augmentation techniques and we can also use different optimizers and this research we used Adam optimizer to increase the classification accuracy.