```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = 0.2, zoom_range = 0.2,
test_datagen = ImageDataGenerator(rescale = 1./255)
```

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten
```

```
import IPython.display as display
from PIL import Image
import pathlib
```

```
!unzip '/content/drive/MyDrive/Dataset/IBM/conversation engine for deaf and dumb.zip'
```

```
Streaming output truncated to the last 5000 lines.
 extracting: Dataset/training_set/G/1223.png
 extracting: Dataset/training_set/G/1224.png
 extracting: Dataset/training_set/G/1225.png
 extracting: Dataset/training_set/G/1226.png
 extracting: Dataset/training_set/G/1227.png
 extracting: Dataset/training_set/G/1228.png
 extracting: Dataset/training_set/G/1229.png
  inflating: Dataset/training_set/G/123.png
 extracting: Dataset/training_set/G/1230.png
 extracting: Dataset/training_set/G/1231.png
 extracting: Dataset/training_set/G/1232.png
  inflating: Dataset/training_set/G/1233.png
  inflating: Dataset/training_set/G/1234.png
  inflating: Dataset/training_set/G/1235.png
  inflating: Dataset/training_set/G/1236.png
  inflating: Dataset/training_set/G/1237.png
  inflating: Dataset/training_set/G/1238.png
  inflating: Dataset/training_set/G/1239.png
  inflating: Dataset/training_set/G/124.png
  inflating: Dataset/training_set/G/1240.png
  inflating: Dataset/training_set/G/1241.png
  inflating: Dataset/training_set/G/1242.png
  inflating: Dataset/training_set/G/1243.png
  inflating: Dataset/training_set/G/1244.png
  inflating: Dataset/training_set/G/1245.png
 extracting: Dataset/training_set/G/1246.png
  inflating: Dataset/training_set/G/1247.png
  inflating: Dataset/training_set/G/1248.png
  inflating: Dataset/training_set/G/1249.png
  inflating: Dataset/training_set/G/125.png
  inflating: Dataset/training_set/G/1250.png
  inflating: Dataset/training_set/G/1251.png
  inflating: Dataset/training_set/G/1252.png
  inflating: Dataset/training_set/G/1253.png
  inflating: Dataset/training_set/G/1254.png
  inflating: Dataset/training_set/G/1255.png
  inflating: Dataset/training_set/G/1256.png  
  inflating: Dataset/training_set/G/1257.png  
  inflating: Dataset/training_set/G/1258.png  
  inflating: Dataset/training_set/G/1259.png  
  inflating: Dataset/training_set/G/126.png  
  inflating: Dataset/training_set/G/1260.png  
  inflating: Dataset/training_set/G/1261.png  
 extracting: Dataset/training_set/G/1262.png  
  inflating: Dataset/training_set/G/1263.png  
  inflating: Dataset/training_set/G/1264.png  
  inflating: Dataset/training_set/G/1265.png  
  inflating: Dataset/training_set/G/1266.png  
  inflating: Dataset/training_set/G/1267.png  
 extracting: Dataset/training_set/G/1268.png  
  inflating: Dataset/training_set/G/1269.png  
  inflating: Dataset/training_set/G/127.png  
  inflating: Dataset/training_set/G/1270.png  
  inflating: Dataset/training_set/G/1271.png  
  inflating: Dataset/training_set/G/1272.png  
  inflating: Dataset/training_set/G/1273.png  
```

```
X_train = train_datagen.flow_from_directory('/content/Dataset/training_set',target_size =
X_test = test_datagen.flow_from_directory('/content/Dataset/test_set', target_size = (64,6
```

```
Found 15750 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.
```

```
model = Sequential()
model.add(Convolution2D(32,(3,3), input_shape = (64,64,1), activation = 'relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Flatten())
model.add(Dense(units = 512, activation = 'relu'))
model.add(Dense(units = 256, activation = 'relu'))
model.add(Dense(units = 128, activation = 'relu'))
model.add(Dense(units = 64, activation = 'relu'))
model.add(Dense(units = 9, activation = 'softmax'))
```

```
model.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'
```

```
model.fit(X_train, steps_per_epoch = len(X_train), epochs = 10, validation_data = X_test,
```

```
Epoch 1/10
53/53 [==============================] - 15s 230ms/step - loss: 0.5748 - accuracy: 0
Epoch 2/10
53/53 [==============================] - 12s 218ms/step - loss: 0.0687 - accuracy: 0
Epoch 3/10
53/53 [==============================] - 11s 213ms/step - loss: 0.0309 - accuracy: 0
Epoch 4/10
53/53 [==============================] - 12s 219ms/step - loss: 0.0201 - accuracy: 0
Epoch 5/10
53/53 [==============================] - 11s 215ms/step - loss: 0.0087 - accuracy: 0
Epoch 6/10
53/53 [==============================] - 11s 214ms/step - loss: 0.0093 - accuracy: 0
Epoch 7/10
53/53 [==============================] - 12s 217ms/step - loss: 0.0105 - accuracy: 0
```

```
Epoch 8/10
53/53 [==============================] - 11s 215ms/step - loss: 0.0058 - accuracy: 0
Epoch 9/10
53/53 [==============================] - 11s 215ms/step - loss: 0.0034 - accuracy: 0
Epoch 10/10
53/53 [==============================] - 11s 214ms/step - loss: 0.0028 - accuracy: 0
<keras.callbacks.History at 0x7f37bb73ded0>
```

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        320       
                                                                 
 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0         
 )                                                               
                                                                 
 flatten (Flatten)           (None, 30752)             0         
                                                                 
 dense (Dense)               (None, 512)               15745536  
                                                                 
 dense_1 (Dense)             (None, 256)               131328    
                                                                 
 dense_2 (Dense)             (None, 128)               32896     
                                                                 
 dense_3 (Dense)             (None, 64)                8256      
                                                                 
 dense_4 (Dense)             (None, 9)                 585       
                                                                 
=================================================================
Total params: 15,918,921
Trainable params: 15,918,921
Non-trainable params: 0
_________________________________________________________________
```

```
model.save('Model_test.h5')
```

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
import cv2
```

```
model = load_model('Model_test.h5')
```

```
img = image.load_img('/content/Dataset/test_set/A/10.png', target_size = (100,100))
```

```
img
```

```
from skimage.transform import resize
def detect(frame):
    img=image.img_to_array(frame)
    img = resize(img,(64,64,1))
    img = np.expand_dims(img,axis=0)
    pred=np.argmax(model.predict(img))
    op=['A','B','C','D','E','F','G','H','I']
    print("The Predicted Letter is ",op[pred])
```

```
img = image.load_img('/content/Dataset/test_set/A/105.png')
detect(img)
```

```
1/1 [==============================] - 0s 101ms/step
The Predicted Letter is  A
```

```
img = image.load_img('/content/Dataset/test_set/E/102.png')
detect(img)
```

```
1/1 [==============================] - 0s 16ms/step
The Predicted Letter is  E
```

```
img = image.load_img('/content/Dataset/test_set/H/100.png')
detect(img)
```

```
1/1 [==============================] - 0s 14ms/step
The Predicted Letter is  G
```

✓ 0s completed at 10:03 PM ● ✕