SOURCE CODE

Import datagenerator to train and test

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale =
1./255,shear_range=0.2,zoom_range=
0.2,horizontal_flip=True,vertical_flip=False)
test_datagen = ImageDataGenerator(rescale = 1./255)
import tensorflow as tf
import os
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout,
MaxPooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt
import IPython.display as display
from PIL import Image
import pathlibfrom keras.preprocessing import image
# and use
# image.ImageDataGenerator()
# image.load_img()
```

Apply ImageDataGenerator Functionality To Train And Test set

```
from google.colab import drive
from tensorflow.keras.preprocessing.image import ImageDataGenerator
print("This dataset has been created and uploaded by IBM-TeamID-IBM-
Project-PNT2022TMID31975")
x_train= train_datagen.flow_from_directory(r"/content/drive/MyDrive/IBM
PROJECT/DATA
COLLECTION/training_set",target_size=(64,64),class_mode="categorical",
batch_size=48)
x_test = test_datagen.flow_from_directory(r"/content/drive/MyDrive/IBM
PROJECT/DATA COLLECTION/test_set",target_size=
(64,64),class_mode= "categorical",batch_size=48)
x_train.class_indices
x_test.class_indices
```

MODEL BUILDING

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten
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=9, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])
model.save('Realtime.h5')
a=len(x_train)
b=len(x_test)
```

Length of training and testing data

```
print(a)
```

print(b)**TEST THE MODEL**

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
import cv2
img = image.load_img('/content/drive/MyDrive/IBM PROJECT/DATA
COLLECTION/test_set/D/101.png',target_size = (500,500))
img
from skimage.transform import resize
arr=image.img_to_array(frame)
arr = resize(arr,(64,64,1))
arr = np.expand_dims(arr,axis=0)
pred=np.argmax(model.predict(arr))
op=['A','B','C','D','E','F','G&#
39;,'H','I']
print("THE PREDICTED LETTER IS ",op[pred])
from skimage.transform import resize
def detect(frame):
img=resize(frame,(64,64,1))
img=np.expand_dims(img,axis=0)
if(np.max(img)>1):
prediction=model.predict(img)
print(prediction)
prediction=model.predict_classes(img)
print(prediction)
arr= image.img_to_array(img)
frame=cv2.imread('/content/drive/MyDrive/IBM PROJECT/DATA
COLLECTION/test_set/F/107.png')
data=detect(frame)
from google.colab.patches import cv2_imshow
cv2_imshow(frame)
cv2.waitKey(0)
cv2.destroyAllWindows()
frame=cv2.imread('/content/drive/MyDrive/IBM PROJECT/DATA
COLLECTION/test_set/A/102.png')
data=detect(frame)
from google.colab.patches import cv2_imshow
cv2_imshow(frame)
cv2.waitKey(0)
cv2.destroyAllWindows()frame=cv2.imread('/content/drive/MyDrive/IBM PROJECT/DATA
COLLECTION/test_set/D/108.png')
data=detect(frame)
from google.colab.patches import cv2_imshow
cv2_imshow(frame)
cv2.waitKey(0)
cv2.destroyAllWindows()
print("THE PREDICTED LETTER IS ",op[pred])
```