

```

from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_r
ange=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)

from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

!unzip Dataset.zip

unzip: cannot find or open Dataset.zip, Dataset.zip.zip or
Dataset.zip.ZIP.

x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/
Dataset/Dataset/
training_set',target_size=(64,64),batch_size=300,class_mode='categoric
al',color_mode='grayscale')
x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/Datase
t/Dataset/
test_set',target_size=(64,64),batch_size=300,class_mode='categorical',
color_mode='grayscale')

Found 15768 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.

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

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.fit_generator(x_train,steps_per_epoch=24,epochs=10,validation_da
ta=x_test,validation_steps=40)

```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1:
UserWarning: `Model.fit_generator` is deprecated and will be removed
in a future version. Please use `Model.fit`, which supports
generators.
```

```
"""Entry point for launching an IPython kernel.
```

```
Epoch 1/10
```

```
24/24 [=====] - ETA: 0s - loss: 0.0115 -
accuracy: 0.9975
```

```
WARNING:tensorflow:Your input ran out of data; interrupting training.
Make sure that your dataset or generator can generate at least
`steps_per_epoch * epochs` batches (in this case, 40 batches). You may
need to use the repeat() function when building your dataset.
```

```
24/24 [=====] - 54s 2s/step - loss: 0.0115 -
accuracy: 0.9975 - val_loss: 0.1363 - val_accuracy: 0.9778
```

```
Epoch 2/10
```

```
24/24 [=====] - 39s 2s/step - loss: 0.0109 -
accuracy: 0.9975
```

```
Epoch 3/10
```

```
24/24 [=====] - 42s 2s/step - loss: 0.0084 -
accuracy: 0.9983
```

```
Epoch 4/10
```

```
24/24 [=====] - 42s 2s/step - loss: 0.0071 -
accuracy: 0.9990
```

```
Epoch 5/10
```

```
24/24 [=====] - 39s 2s/step - loss: 0.0063 -
accuracy: 0.9993
```

```
Epoch 6/10
```

```
24/24 [=====] - 41s 2s/step - loss: 0.0061 -
accuracy: 0.9989
```

```
Epoch 7/10
```

```
24/24 [=====] - 39s 2s/step - loss: 0.0036 -
accuracy: 0.9996
```

```
Epoch 8/10
```

```
24/24 [=====] - 41s 2s/step - loss: 0.0043 -
accuracy: 0.9992
```

```
Epoch 9/10
```

```
24/24 [=====] - 42s 2s/step - loss: 0.0048 -
accuracy: 0.9992
```

```
Epoch 10/10
```

```
24/24 [=====] - 40s 2s/step - loss: 0.0086 -
accuracy: 0.9976
```

```
<keras.callbacks.History at 0x7fae4f025310>
```

```
model.save('aslpng1.h5')
```

```
from keras.models import load_model
import numpy as np
import cv2

model=load_model('aslpng1.h5')

from skimage.transform import resize

frame=cv2.imread(r'/content/drive/MyDrive/Dataset/Dataset/test_set/I/1.png')
data=detect(frame)
from google.colab.patches import cv2_imshow
cv2_imshow(frame)
cv2.waitKey(0)
cv2.destroyAllWindows()

1/1 [=====] - 0s 31ms/step
THE PREDICTED LETTER IS  I
```

