Testing the model

Importing Libraries

In [1]:

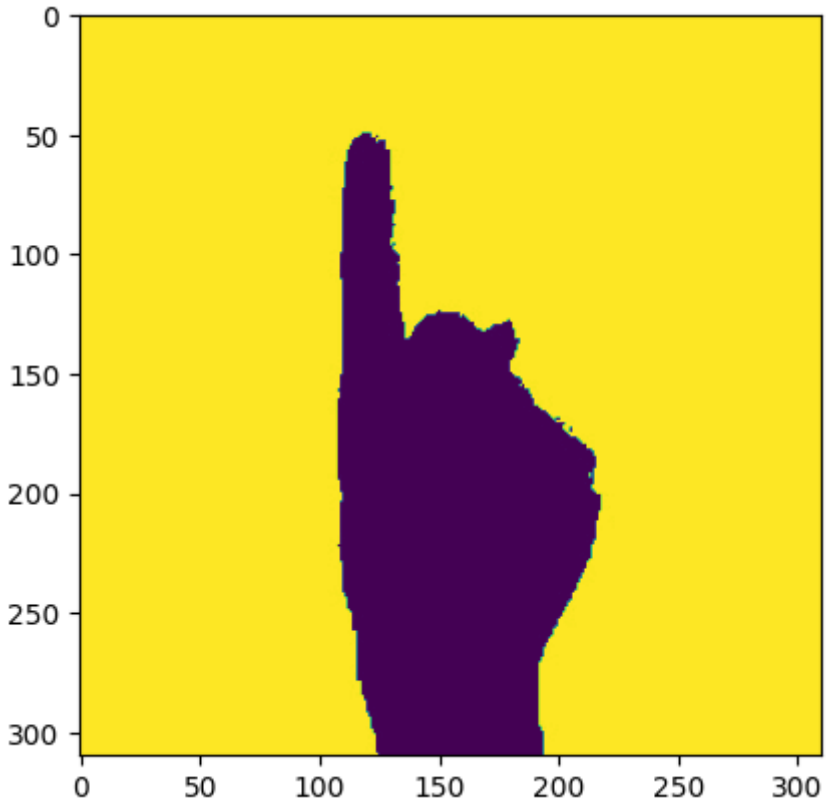
```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model = load_model("D:\\Hand.h5")
path = "D:\\Project\\Dataset\\test\\test\\1\\4.jpg"
```

Plotting the image

In [2]:

```
%pylab inline
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
imgs = mpimg.imread(path)
imgplot = plt.imshow(imgs)
plt.show()
```

```
%pylab is deprecated, use %matplotlib inline and import the required librari
es.
Populating the interactive namespace from numpy and matplotlib
```



In [3]:

```
img = image.load_img(path,
                     color_mode='grayscale',
                     target_size= (64,64))
x = image.img_to_array(img)#image to array
x.shape
```

Out[3]:

```
(64, 64, 1)
```

In [4]:

```
type(x)
```

Out[4]:

```
numpy.ndarray
```

In [5]:

```
x = np.expand_dims(x,axis = 0)
```

In [6]:

```
x.shape
```

Out[6]:

```
(1, 64, 64, 1)
```

Predicting our results

In [7]:

```
pred = np.argmax(model.predict(x), axis=-1)
pred
```

```
1/1 [==============================] - 0s 141ms/step
```

Out[7]:

```
array([2], dtype=int64)
```

In [8]:

```
index=['0','1','2','3','4','5']
result=str(index[pred[0]])
result
```

Out[8]:

```
'2'
```

In [9]:

```
import numpy as np
p = []

for i in range(0,6):
    for j in range(0,5):
        path = "D:\\Project\\Dataset\\test\\test\\"+str(i)+"\\"+str(j)+".jpg"
        img = image.load_img(path,color_mode = "grayscale",target_size= (64,64))
        x = image.img_to_array(img)
        x = np.expand_dims(x,axis = 0)
        pred = np.argmax(model.predict(x), axis=-1)
        p.append(pred)

print(p)
```

```
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 27ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 27ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 29ms/step
[array([2], dtype=int64), array([2], dtype=int64), array([2], dtype=int64),
array([2], dtype=int64), array([2], dtype=int64), array([2], dtype=int64), a
rray([2], dtype=int64), array([2], dtype=int64), array([2], dtype=int64), ar
ray([2], dtype=int64), array([2], dtype=int64), array([2], dtype=int64), arr
ay([2], dtype=int64), array([2], dtype=int64), array([2], dtype=int64), arra
y([2], dtype=int64), array([2], dtype=int64), array([2], dtype=int64), array
([2], dtype=int64), array([2], dtype=int64), array([2], dtype=int64), array
([2], dtype=int64), array([2], dtype=int64), array([2], dtype=int64), array
([2], dtype=int64), array([2], dtype=int64), array([2], dtype=int64), array
([2], dtype=int64), array([2], dtype=int64), array([2], dtype=int64)]
```

In [10]:

```
result = []
index=['0','1','2','3','4','5']
for i in p:
    result.append(index[i[0]])

print(result)
```

```
['2', '2', '2', '2', '2', '2', '2', '2', '2', '2', '2', '2', '2', '2', '2',
'2', '2', '2', '2', '2', '2', '2', '2', '2', '2', '2', '2', '2', '2', '2']
```