

▼ Classification_Task_Flower_ Classification_VGG16

Checking System Config

```
%cat /etc/lsb-release

DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=18.04
DISTRIB_CODENAME=bionic
DISTRIB_DESCRIPTION="Ubuntu 18.04.6 LTS"
```

Update Repositories

```
!apt-get update
```

```
%cd /content/drive/My Drive/DL

/content/drive/My Drive/DL
```

```
!unzip Flowers_Manual.zip
```

```
%pwd

'/content/drive/MyDrive/DL'
```

```
%cd /content/drive/MyDrive/DL/Flowers_Manual/

/content/drive/MyDrive/DL/Flowers_Manual
```

▼ Libraries

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.applications.vgg16 import VGG16, decode_predictions, preprocess_input
from tensorflow.keras.models import Model
from keras import layers
```

▼ Mode

```
Train_dataGen = ImageDataGenerator(rescale=1/255,rotation_range=20,horizontal_flip=True,width
    height_shift_range=0.2)
Test_dataGen =ImageDataGenerator(rescale=1/255)
```

```
model=VGG16(include_top=False,input_shape=(224,224,3))
```

```
for layer in model.layers:
    layer.trainable=False
```

```
flatten_layer= layers.Flatten()(model.output)
```

```
fC_layer= layers.Dense(512,activation='relu')(flatten_layer)
fC_softmax_layer= layers.Dense(5,activation='softmax')(fC_layer)
```

```
model= Model(inputs=model.input,outputs=fC_softmax_layer)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16
58892288/58889256 [=====] - 1s 0us/step
58900480/58889256 [=====] - 1s 0us/step
```

```
train_itt =Train_dataGen.flow_from_directory('Train',
    batch_size=64,target_size=(224,224))
```

```
test_itt = Test_dataGen.flow_from_directory('Test',
    batch_size=64,target_size=(224,224))
```

```
Found 3735 images belonging to 5 classes.
Found 1199 images belonging to 5 classes.
```

```
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
model.fit(train_itt,validation_data=test_itt,epochs=8)
```

```
Epoch 1/8
59/59 [=====] - 92s 1s/step - loss: 2.0542 - accuracy: 0.5933 .
Epoch 2/8
59/59 [=====] - 73s 1s/step - loss: 0.6287 - accuracy: 0.7778 .
Epoch 3/8
59/59 [=====] - 85s 1s/step - loss: 0.5166 - accuracy: 0.8201 .
Epoch 4/8
59/59 [=====] - 72s 1s/step - loss: 0.4494 - accuracy: 0.8412 .
Epoch 5/8
59/59 [=====] - 71s 1s/step - loss: 0.4313 - accuracy: 0.8428 .
Epoch 6/8
59/59 [=====] - 71s 1s/step - loss: 0.4050 - accuracy: 0.8546 .
Epoch 7/8
59/59 [=====] - 71s 1s/step - loss: 0.4015 - accuracy: 0.8455 .
```

Epoch 8/8

59/59 [=====] - 72s 1s/step - loss: 0.3786 - accuracy: 0.8581 -

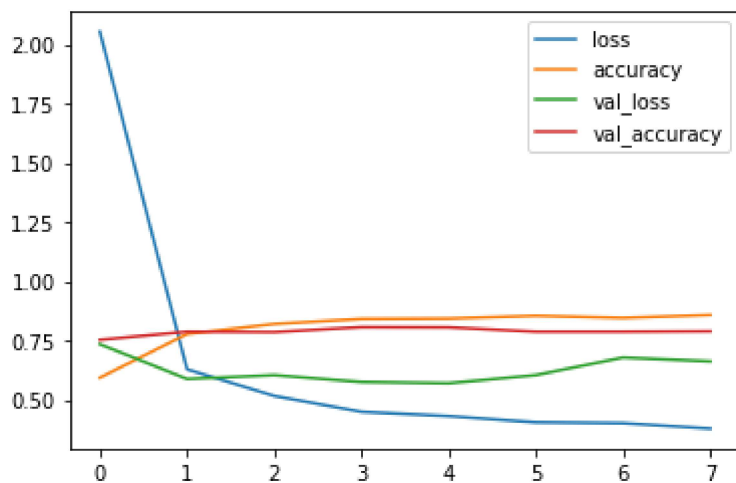
<keras.callbacks.History at 0x7f6f7abbd9d0>



losses = pd.DataFrame(model.history.history)

losses.plot()

<matplotlib.axes._subplots.AxesSubplot at 0x7f6ee80e4610>



plt.plot(losses['loss'])

plt.plot(losses['val_loss'])

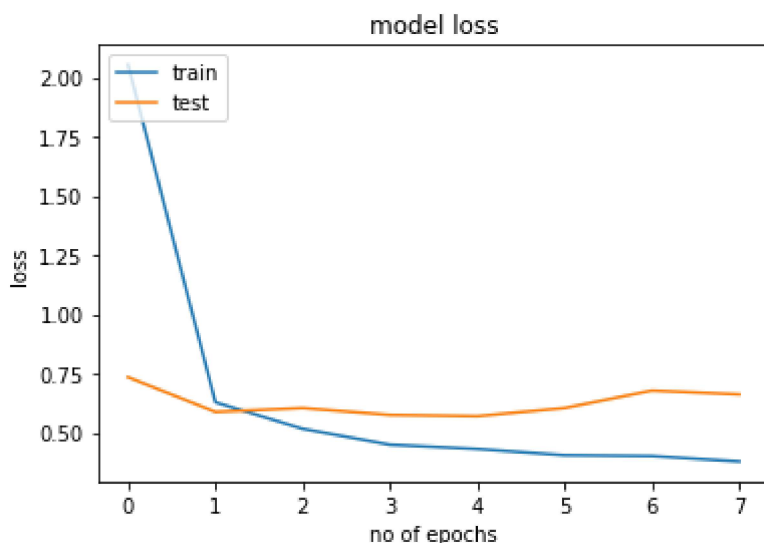
plt.title('model loss')

plt.ylabel('loss')

plt.xlabel('no of epochs')

plt.legend(['train', 'test'], loc='upper left')

plt.show()

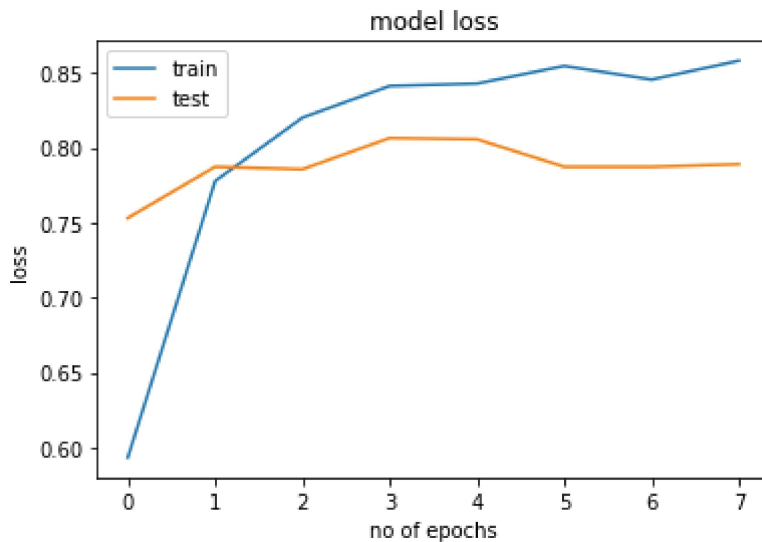


plt.plot(losses['accuracy'])

plt.plot(losses['val_accuracy'])

plt.title('model loss')

```
plt.ylabel('loss')
plt.xlabel('no of epochs')
plt.legend(['train','test'],loc='upper left')
plt.show()
```



```
model.save('/content/drive/MyDrive/DL/VGG16_trained_model.h5')
```

▼ Predicting Mode

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import load_img,img_to_array
import matplotlib.pyplot as plt
import numpy as np
```

```
model= load_model('/content/drive/MyDrive/DL/VGG16_trained_model.h5')
```

```
image = load_img('/content/drive/MyDrive/DL/Images/tulip.jpg',target_size=(224,224))
plt.imshow(image)
plt.show()
image= img_to_array(image)
image=image.reshape(1,224,224,3)
```





```
result=model.predict(image)
class_label={'daisy': 0, 'dandelion': 1, 'rose': 2, 'sunflower': 3, 'tulip': 4}
result=np.argmax(result)

print([key for key in class_label][result])

tulip
```

✓ 0s completed at 4:20 PM

