ASSIGNMENT 3

Assignment Date	**27/09/2022**
Student Name	**S.Saiusha**
Student Roll No	**960519104070**
Maximum Marks	**2 Marks**

In[1]:

```
importsplitfoldersimport
numpyasnpimport tensorflowastf
fromtensorflow.keras.preprocessing.imageimport
ImageDataGeneratorfromtensorflow.keras.preprocessingimportimage
fromtensorflow.kerasimportlayersfromtensorflow.keras.modelsimport
Sequentialfromtensorflow.keras.modelsimportload_model
fromtensorflow.keras.layersimportDense,Convolution2D,MaxPooling2D,Flattenfrom
tensorflow.keras.applications.resnet50importpreprocess_input,decode_predictifrom
tensorflow.keras.preprocessingimportimageimportmatplotlib.pyplotasplt
```

```
train_datagen= ImageDataGenerator(rescale=1../255,zoom_range=0.2,horizontal_flip=Tru
```

2.ImageAugmentation

In[2]:

In [3]:
```
test_datagen =ImageDataGenerator(rescale=1./25 )
```

In [4]:
```
input_folder = '.\Flowers-Dataset\flowers'
```

In[5]: `splitfolders.ratio(input_folder,output="flowers",ratio=(.8,0,.2),group_prefix=None)`

```
Copyingfiles:4317files[00:03,1292.11files/s]
```

In[6]: `x_train=train_datagen.flow_from_directory(r".\flowers\train",target_size=(64,64),cla`

```
Found3452imagesbelongingto5classes.
```

In[7]: `x_test=test_datagen.flow_from_directory(r".\flowers\test",target_size=(64,64),class_`

```
Found865imagesbelongingto5classes.
```

In[8]:

```
x_train .class_indices
```

{'daisy':0,'dandelion':1,'rose':2,'sunflower':3,'tulip':4}Out[8]:

3.CreateModel

In[9]:

```
model=Sequential()
```

4.AddLayers

4.1.ConvolutionLayer

In[10]:

```
model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation='relu'))
```

4.2.MaxPoolingLayer

In[11]:

```
model.add(MaxPooling2D(pool_size=(2,2)))
```

4.3.FlattenLayer

In[12]:

```
model.add(Flatten ())
```

4.4.DenseLayer

```
In[13]:model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu')) In[14]:
```

```
model.summary()
```

```
Model:"sequential"
_________________________________________________________________
   Layer(type)              OutputShape             Param#
================================================================
             =conv2d(Conv2D) (None,62,62,32) 896                               max_pooling2d(MaxPooling2D
(None,31,31,32) 0
  )                                                                flatten(Flatten) (None,
  30752)             0                                dense(Dense)                (None,300)
  9225900                              dense_1(Dense) (None,150)                  45150
================================================================
=
Totalparams:9,271,946
Trainableparams:9,271,946
Non-trainableparams:0
```

```
model.add(Dense(5,activation ='softmax'))
```

```
model.summary()
```

4.5.OutputLayer

In[15]:

In[16]:

```
Model:"sequential"
_________________________________________________________________
 Layer(type)               OutputShape              Param#
=================================================================
conv2d(Conv2D)  (None,62,62,32)  896  max_pooling2d(MaxPooling2D  (None,31,31,32)  0  )  flatten(Flatten)
(None,30752) 0 dense(Dense) (None,300) 9225900 dense_1(Dense) (None,
 150)            45150                              dense_2(Dense) (None,5)                    755
=================================================================
Totalparams:9,272,701
Trainableparams:9,272,701
Non-trainableparams:0
_________________________________________________________________
```

```
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])len(x_train)
```

5.CompileTheModel

In[17]:

144Out[17]:

```
epo=20history=
model.fit(x_train,steps_per_epoch=len(x_train),validation_data=x_test,vali
```

6.FitTheModel

In[18]:

```
Epoch1/20
144/144[==============================]-29s202ms/step-loss:1.4725-accuracy:
0.4293-val_loss:1.1148-val_accuracy:0.5538Epoch2/20
144/144[==============================]-15s101ms/step-loss:1.0813-accuracy:
0.5640-val_loss:1.0807-val_accuracy:0.5653
Epoch3/20
144/144[==============================]-15s102ms/step-loss:0.9676-accuracy:
0.6185-val_loss:1.0689-val_accuracy:0.5977
Epoch4/20
144/144[==============================]-15s101ms/step-loss:0.9144-accuracy:
0.6411-val_loss:0.9561-val_accuracy:0.6497
Epoch5/20
```

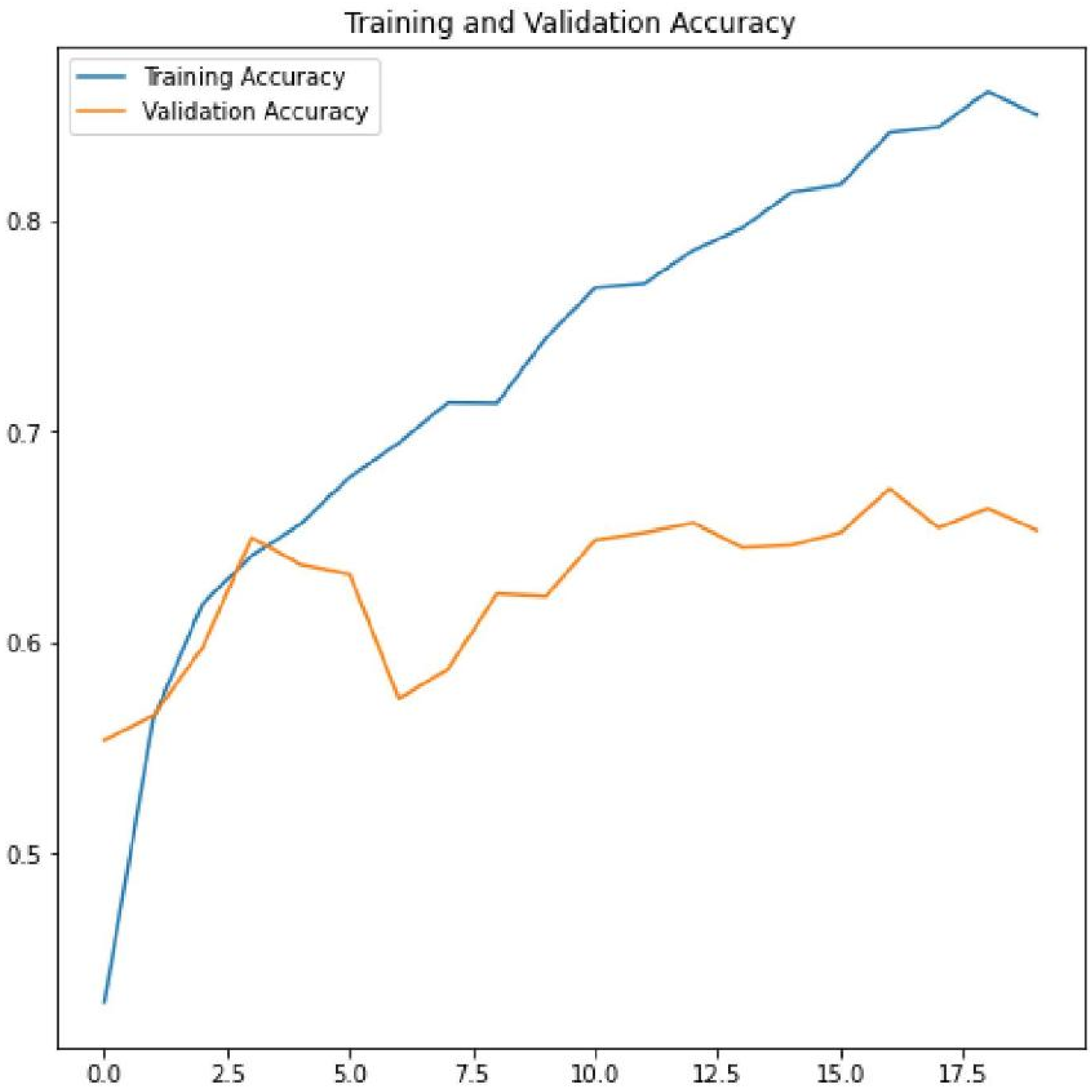
```
144/144[==============================]-17s116ms/step-loss:0.8731-accuracy:
0.6561-val_loss:0.9766-val_accuracy:0.6370Epoch6/20
144/144[==============================]-15s107ms/step-loss:0.8303-accuracy:
0.6784-val_loss:1.0373-val_accuracy:0.6324
Epoch7/20
144/144[==============================]-16s108ms/step-loss:0.7858-accuracy:
0.6947-val_loss:1.1446-val_accuracy:0.5734
Epoch8/20
```

```
144/144[==============================]-15s105ms/step-loss:0.7539-accuracy:
0.7138-val_loss:1.1979-val_accuracy:0.5873
Epoch9/20
144/144[==============================]-15s107ms/step-loss:0.7262-accuracy:
0.7135-val_loss:1.0924-val_accuracy:0.6231Epoch10/20
144/144[==============================]-15s101ms/step-loss:0.6684-accuracy:
0.7445-val_loss:1.1218-val_accuracy:0.6220
Epoch11/20
144/144[==============================]-15s106ms/step-loss:0.6142-accuracy:
0.7683-val_loss:1.0576-val_accuracy:0.6486
Epoch12/20
144/144[==============================]-15s106ms/step-loss:0.6006-accuracy:
0.7703-val_loss:1.0454-val_accuracy:0.6520Epoch13/20
144/144[==============================]-15s105ms/step-loss:0.5584-accuracy:
0.7859-val_loss:1.0735-val_accuracy:0.6566
Epoch14/20
144/144[==============================]-15s102ms/step-loss:0.5387-accuracy:
0.7966-val_loss:1.1083-val_accuracy:0.6451
Epoch15/20
144/144[==============================]-15s103ms/step-loss:0.4935-accuracy:
0.8134-val_loss:1.0815-val_accuracy:0.6462
Epoch16/20
144/144[==============================]-14s100ms/step-loss:0.4961-accuracy:
0.8172-val_loss:1.0991-val_accuracy:0.6520Epoch17/20
144/144[==============================]-15s103ms/step-loss:0.4373-accuracy:
0.8418-val_loss:1.2605-val_accuracy:0.6728
Epoch18/20
144/144[==============================]-15s102ms/step-loss:0.4228-accuracy:
0.8444-val_loss:1.1316-val_accuracy:0.6543
Epoch19/20
144/144[==============================]-15s104ms/step-loss:0.3853-accuracy:
0.8612-val_loss:1.1264-val_accuracy:0.6636
Epoch20/20
144/144[==============================]-14s100ms/step-loss:0.3900-accuracy:
0.8502-val_loss:1.1911-val_accuracy:0.6532
```

In[19]:

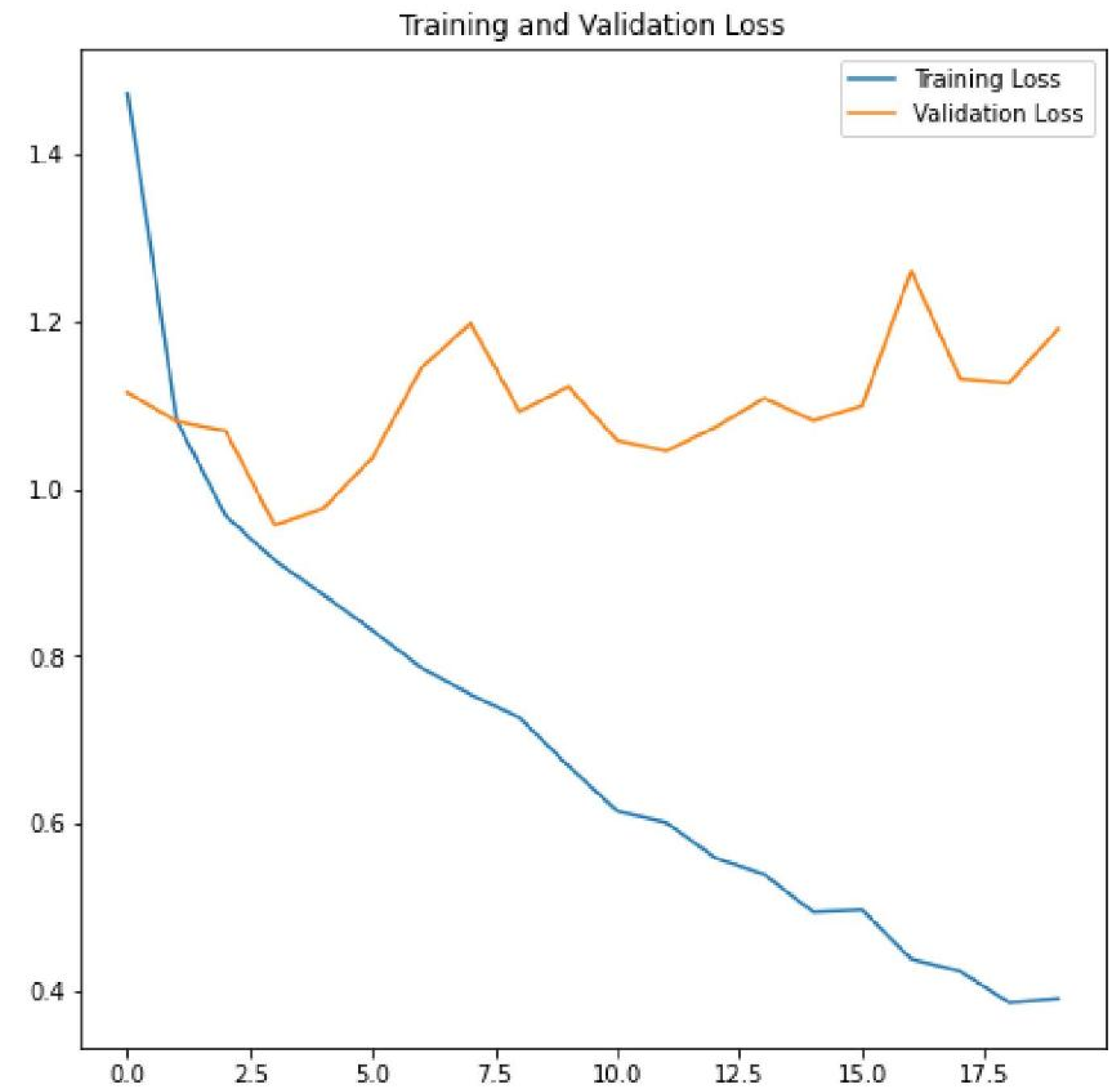
```
epochs_range=range(epo)

plt.figure(figsize=(8,8))
plt.plot(epochs_range,history.history['accuracy'],label='TrainingAccuracy')plt.plot(epochs_range,
history.history['val_accuracy'],label='ValidationAccuracy')plt.legend()
plt.title('TrainingandValidationAccuracy')plt.show()
```



In[20]:

```
plt.figure(figsize=(8,8))plt.plot(epochs_range,history.history['loss'],label='TrainingLoss')
plt.plot(epochs_range,history.history['val_loss'],label='ValidationLoss')plt.legend()
plt.title('TrainingandValidationLoss')plt.show()
```



7.SavetheModel

In[21]:

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

```
img=image.load_img(r".\flowers\test\daisy\3706420943_66f3214862_n.jpg",target_size=(x=image.img
_to_array(img)x=np.expand_dims(x,axis=0)y=np.argmax(model.predict(x),axis=1)x_train.class_indices
index=['daisy','dandellion','rose','sunflower','tulip']index[y[0]]
```

8.TesttheModel

In[22]:

```
1/1[==============================]-0s77ms/step 'daisy'Out[22]:
```

In[23]:2

```
img_url=
 "https://storage.googleapis.com/download.tensorflow.org/example_images/59img_path=
 tf.keras.utils.get_file('Red_sunflower',origin=img_url)

 img=image.load_img(img_path,target_size=(224,224))img_array= image.img_to_array(img)
 img_batch=np.expand_dims(img_array,axis=0)

 img_preprocessed=preprocess_input(img_batch)model=
 tf.keras.applications.resnet50.ResNet50()prediction=
 model.predict(img_preprocessed)
 print(decode_predictions(prediction,top=3)[0])

   score=tf.nn.softmax(prediction[0])
```

```
Downloadingdatafromhttps://storage.googleapis.com/download.tensorflow.org/example
_images/592pxRed_sunflower.jpg
117948/117948[==============================]-0s0us/stepDownloadingdata
fromhttps://storage.googleapis.com/tensorflow/kerasapplications/r
esnet/resnet50_weights_tf_dim_ordering_tf_kernels.h5
102967424/102967424[==============================]-3s0us/step
1/1[==============================]-1s868ms/stepDownloadingdata
fromhttps://storage.googleapis.com/download.tensorflow.org/data/im agenet_class_index.json
35363/35363[==============================]-0s0us/step
[('n11939491','daisy',0.5775759),('n02206856','bee',0.24938338),('n03991062','pot',0.01181931)]
```