1.Import Nessasry Libraries.

```
from keras.layers import Flatten,Dense,AveragePooling2D,Conv2D,MaxPooling2D
from keras.models import Sequential
from keras.models import Model
from keras.layers import Dense, Conv2D, MaxPool2D , Flatten
import keras,os
from keras.preprocessing.image import ImageDataGenerator
```

```
train_data=ImageDataGenerator()
test_data=ImageDataGenerator()
```

```
train_path=train_data.flow_from_directory(r'/content/drive/MyDrive/data/train', target_siz
test_path=train_data.flow_from_directory(r'/content/drive/MyDrive/data/test',target_size=(
validation_path=validation_data.flow_from_directory(r'/content/drive/MyDrive/data/validati
```

```
Found 27 images belonging to 3 classes.
Found 10 images belonging to 3 classes.
Found 12 images belonging to 3 classes.
```

1.VGG16

```
from keras.applications.vgg16 import VGG16
vgg16_model=VGG16(input_shape=(224,224,3),include_top=False, weights='imagenet')

for layer in vgg16_model.layers:
    layer.trainable=False
```

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

```
x=Flatten()(vgg16_model.output)
prediction=Dense(3,activation='softmax')(x)
```

```
model=Model(inputs=vgg16_model.input,outputs=prediction)
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics='accuracy')
```

```
model1=model.fit(x=train_path, epochs=10, verbose=2,validation_data=test_path)
```

```
Epoch 1/10
1/1 - 24s - loss: 20.5274 - accuracy: 0.2222 - val_loss: 1.5042 - val_accuracy: 0.80
```

```
Epoch 2/10
1/1 - 1s - loss: 5.4038e-06 - accuracy: 1.0000 - val_loss: 2.8510 - val_accuracy: 0.
Epoch 3/10
1/1 - 1s - loss: 7.9421e-04 - accuracy: 1.0000 - val_loss: 4.0455 - val_accuracy: 0.
Epoch 4/10
1/1 - 1s - loss: 0.0054 - accuracy: 1.0000 - val_loss: 4.9643 - val_accuracy: 0.9000
Epoch 5/10
1/1 - 1s - loss: 2.6491e-07 - accuracy: 1.0000 - val_loss: 5.7558 - val_accuracy: 0.
Epoch 6/10
1/1 - 1s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 6.4381 - val_accuracy: 0.
Epoch 7/10
1/1 - 1s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 7.0338 - val_accuracy: 0.
Epoch 8/10
1/1 - 1s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 7.5591 - val_accuracy: 0.
Epoch 9/10
1/1 - 1s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 8.0253 - val_accuracy: 0.
Epoch 10/10
1/1 - 1s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 8.4415 - val_accuracy: 0.
```

```
model.save('model1.h5')
```

```
from keras.models import load_model
import numpy as np
from keras.preprocessing import image
```

```
from keras.models import load_model
import numpy as np
from keras.preprocessing import image

import numpy as np
from keras.preprocessing import image

test_image = image.load_img(r'/content/drive/MyDrive/data/test/khushbu/IMG-20210823-WA0005
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis = 0)

# load model
model = load_model('model1.h5')
result = model.predict(test_image)
if result[0][0] == 1:
    print('anal')

elif result[0][1]==1:
  print('khushbu')

elif result[0][2]==1:
  print('rinku')
```

```
khushbu
```

2.VGG19

```
from keras.applications.vgg19 import VGG19


vgg19_model=VGG19(include_top=False, weights='imagenet', input_shape=(224,224,3))
for layer in vgg19_model.layers:
    layer.trainable=False
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/v
80142336/80134624 [==============================] - 1s 0us/step
80150528/80134624 [==============================] - 1s 0us/step
```

```
x=Flatten()(vgg19_model.input)
prediction=Dense(3,activation='softmax')(x)
model2=Model(inputs=vgg19_model.input,outputs=prediction)
model2.compile(optimizer='adam', loss='categorical_crossentropy', metrics='accuracy')


model2.fit(x=train_path, epochs=10, verbose=2,validation_data=test_path)
```

```
Epoch 1/10
1/1 - 2s - loss: 68.9835 - accuracy: 0.4074 - val_loss: 10169.7354 - val_accuracy: 0
Epoch 2/10
1/1 - 1s - loss: 4488.3408 - accuracy: 0.5185 - val_loss: 8914.2520 - val_accuracy:
Epoch 3/10
1/1 - 1s - loss: 4993.8960 - accuracy: 0.4815 - val_loss: 3483.5386 - val_accuracy:
Epoch 4/10
1/1 - 1s - loss: 2361.2964 - accuracy: 0.5556 - val_loss: 1266.5925 - val_accuracy:
Epoch 5/10
1/1 - 1s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 4556.6235 - val_accuracy:
Epoch 6/10
1/1 - 1s - loss: 2042.0189 - accuracy: 0.7407 - val_loss: 3955.6055 - val_accuracy:
Epoch 7/10
1/1 - 1s - loss: 1100.5179 - accuracy: 0.7778 - val_loss: 1927.4576 - val_accuracy:
Epoch 8/10
1/1 - 1s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 2042.9778 - val_accuracy:
Epoch 9/10
1/1 - 1s - loss: 299.2634 - accuracy: 0.9630 - val_loss: 2527.4419 - val_accuracy: 0
Epoch 10/10
1/1 - 1s - loss: 631.9435 - accuracy: 0.9630 - val_loss: 2836.5208 - val_accuracy: 0
<keras.callbacks.History at 0x7f6b45bf92d0>
```

```
model2.save('vgg19model.h5')


from keras.models import load_model
import numpy as np
from keras.preprocessing import image


test_image1=image.load_img('/content/drive/MyDrive/data/test/rinku/IMG-20210829-WA0012.jpg
test_image1=np.array(test_image1)
test_image1=np.expand_dims(test_image1,axis=0)
```

```
vggmodel=load_model('vgg19model.h5')
result=vggmodel.predict(test_image1)

if result[0][0]==1:
  print('anal')
elif result[0][1]==1:
  print('khushbu')
elif result[0][2]==1:
  print('rinku')
```

```
rinku
```

3.InceptionV3

```
from keras.applications.inception_v3 import  InceptionV3


inception_model=InceptionV3(include_top=False, weights='imagenet', input_shape=(224,224,3)
for  layer in inception_model.layers:
    layer.trainable=False


x=Flatten()(inception_model.output)
prediction=Dense(3,activation='softmax')(x)


model3=Model(inputs=inception_model.input,outputs=prediction)
model3.compile(optimizer='adam', loss='categorical_crossentropy', metrics='accuracy')



model3.fit(x=train_path,epochs=10 ,verbose=2,validation_data=test_path)
```

```
Epoch 1/10
1/1 - 2s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 176.0731 - val_accuracy:
Epoch 2/10
1/1 - 1s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 177.1261 - val_accuracy:
Epoch 3/10
1/1 - 1s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 178.0847 - val_accuracy:
Epoch 4/10
1/1 - 1s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 178.9578 - val_accuracy:
Epoch 5/10
1/1 - 1s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 179.9655 - val_accuracy:
Epoch 6/10
1/1 - 1s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 182.3326 - val_accuracy:
Epoch 7/10
1/1 - 1s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 184.5006 - val_accuracy:
Epoch 8/10
1/1 - 1s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 186.4762 - val_accuracy:
Epoch 9/10
1/1 - 1s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 188.2760 - val_accuracy:
Epoch 10/10
1/1 - 1s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 189.9158 - val_accuracy:
<keras.callbacks.History at 0x7f6acf4cfdd0>
```

```
model3.save('inceptionmodel.h5')
```

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

test_image3=image.load_img('/content/drive/MyDrive/data/test/anal/IMG-20211130-WA0073.jpg'
test_image3=np.array(test_image3)
test_image3=np.expand_dims(test_image3,axis=0)

modelintercept=load_model('inceptionmodel.h5')
result=modelintercept.predict(test_image3)

if result[0][0]==1:
  print('anal')
elif result[0][1]==1:
  print('khushbu')
elif result[0][2]==1:
  print('rinku')
```

```
anal
```

4. ResNet 50

```
from keras.applications.resnet import ResNet50
restnetmodel=ResNet50(include_top=False, weights='imagenet', input_shape=(224,224,3))
for layer in restnetmodel.layers:
    layer.trainable=False
```

```
x=Flatten()(restnetmodel.input)
prediction=Dense(3,activation='softmax')(x)
model4=Model(inputs=restnetmodel.input,outputs=prediction)
model4.compile(optimizer='adam', loss='categorical_crossentropy', metrics='accuracy')
```

```
model4.fit(x=train_path,epochs=10 ,verbose=2,validation_data=test_path)
```

```
Epoch 1/10
1/1 - 3s - loss: 176.1176 - accuracy: 0.3704 - val_loss: 7785.0049 - val_accuracy: 0
Epoch 2/10
1/1 - 2s - loss: 10199.0439 - accuracy: 0.2593 - val_loss: 18904.4355 - val_accuracy
Epoch 3/10
1/1 - 2s - loss: 17549.8906 - accuracy: 0.4815 - val_loss: 17183.8750 - val_accuracy
Epoch 4/10
1/1 - 2s - loss: 16203.6299 - accuracy: 0.4815 - val_loss: 9930.1074 - val_accuracy:
Epoch 5/10
1/1 - 2s - loss: 11053.5166 - accuracy: 0.5185 - val_loss: 6135.6758 - val_accuracy:
Epoch 6/10
1/1 - 2s - loss: 6361.9883 - accuracy: 0.6296 - val_loss: 5573.5586 - val_accuracy:
```

```
Epoch 7/10
1/1 - 2s - loss: 4365.2183 - accuracy: 0.3704 - val_loss: 2375.1362 - val_accuracy:
Epoch 8/10
1/1 - 2s - loss: 132.8501 - accuracy: 0.8519 - val_loss: 1428.1198 - val_accuracy: 0
Epoch 9/10
1/1 - 2s - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 1695.2457 - val_accuracy:
Epoch 10/10
1/1 - 2s - loss: 266.8096 - accuracy: 0.9630 - val_loss: 4074.8274 - val_accuracy: 0
<keras.callbacks.History at 0x7f6acf3d9590>
```

```
model4.save('resnetmodel.h5')
```

```
from keras.models  import load_model

test_image4=image.load_img('/content/drive/MyDrive/data/test/anal/IMG-20211130-WA0075.jpg'
test_image4=np.array(test_image4)
test_image4=np.expand_dims(test_image4,axis=0)

modelresnet=load_model('resnetmodel.h5')
result=modelresnet.predict(test_image4)

if result[0][0]==1:
  print('anal')
elif result[0][1]==1:
  print('khushbu')
elif result[0][2]==1:
  print('rinku')
```

```
anal
```

5.MobileNet

```
from keras.applications.mobilenet import MobileNet
```

```
model5=MobileNet(include_top=False, weights='imagenet', input_shape=(224,224,3))
for layer in  model5.layers:
    layer.trainable=False
```

```
x=Flatten()(model5.input)
prediction=Dense(3,activation='softmax')(x)
```

```
model5=Model(inputs=model5.input,outputs=prediction)
model5.compile(optimizer='adam', loss='categorical_crossentropy', metrics='accuracy')
```

```
model5.fit(x=train_path,epochs=10 ,verbose=2,validation_data=test_path)
```

```
Epoch 1/10
1/1 - 2s - loss: 145.6521 - accuracy: 0.4815 - val_loss: 11990.9004 - val_accuracy:
Epoch 2/10
1/1 - 1s - loss: 19928.1348 - accuracy: 0.4074 - val_loss: 18831.6816 - val_accuracy
Epoch 3/10
1/1 - 1s - loss: 29278.7109 - accuracy: 0.1481 - val_loss: 10787.5625 - val_accuracy
Epoch 4/10
1/1 - 2s - loss: 16164.2939 - accuracy: 0.1481 - val_loss: 2578.1785 - val_accuracy:
Epoch 5/10
1/1 - 2s - loss: 2696.6006 - accuracy: 0.4444 - val_loss: 5716.9536 - val_accuracy:
Epoch 6/10
1/1 - 2s - loss: 2117.3901 - accuracy: 0.8519 - val_loss: 13183.2441 - val_accuracy:
Epoch 7/10
1/1 - 2s - loss: 5455.5845 - accuracy: 0.5556 - val_loss: 17526.5527 - val_accuracy:
Epoch 8/10
1/1 - 1s - loss: 7586.1157 - accuracy: 0.4815 - val_loss: 17882.9258 - val_accuracy:
Epoch 9/10
1/1 - 2s - loss: 7186.1743 - accuracy: 0.6667 - val_loss: 16871.3906 - val_accuracy:
Epoch 10/10
1/1 - 1s - loss: 6280.9561 - accuracy: 0.7778 - val_loss: 15030.0176 - val_accuracy:
<keras.callbacks.History at 0x7f6b438b0b50>
```

```
model5.save("mobilenet.h5")
```

```
test_image5=image.load_img('/content/drive/MyDrive/data/test/khushbu/IMG-20210825-WA0032.j
test_image5=np.array(test_image5)
test_image5=np.expand_dims(test_image5,axis=0)

model5=load_model('mobilenet.h5')
result=model5.predict(test_image5)

if result[0][0]==1:
  print('anal')
elif result[0][1]==1:
  print('khushbu')
elif result[0][2]==1:
  print('rinku')
```

```
khushbu
```

✓ 0s completed at 3:37 PM