

```
In [2]: from tensorflow import keras
```

```
In [3]: from keras.preprocessing.image import ImageDataGenerator
```

```
In [5]: train_data=ImageDataGenerator(rescale=1./255,rotation_range=0.2,shear_range=0.3,zoom_range=0.2,
test_data=ImageDataGenerator(rescale=1./255)
validation_data=ImageDataGenerator(rescale=1./255)
```

```
In [9]: train_path=train_data.flow_from_directory(directory=r'C:\data\train',target_size=(150,150),
color_mode='rgb')
test_path=test_data.flow_from_directory(directory=r'C:\data\test')
validation_path=validation_data.flow_from_directory(directory=r'C:\data\train')
```

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

```
In [35]: train_path.class_indices
```

```
Out[35]: {'anal': 0, 'khushbu': 1, 'rinku': 2}
```

```
In [15]: from keras.layers import Flatten,Dense,MaxPool2D,Conv2D,MaxPooling2D
from keras.models import Sequential
```

```
In [19]: model=Sequential()
model.add(Conv2D(input_shape=(256,256,3),filters=16,strides=1 ,activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2),strides=2))
model.add(Conv2D(strides=1,filters=32,activation='relu',kernel_size=(5,5)))
model.add(MaxPooling2D(pool_size=(2,2),strides=2))
model.add(Flatten())
model.add(Dense(units=4000,activation='relu'))
model.add(Dense(units=1000,activation='relu'))
model.add(Dense(units=400,activation='relu'))
model.add(Dense(units=3,activation='softmax'))
```

```
In [20]: model.compile(optimizer='adam',loss='categorical_crossentropy',metrics='accuracy')
```

```
In [22]: model_training=model.fit(x=train_path, batch_size=32, epochs=10,verbose=2,validat
```

```
Epoch 1/10
1/1 - 60s - loss: 66.1171 - accuracy: 0.4815 - val_loss: 89.8195 - val_accurac
y: 0.3704 - 60s/epoch - 60s/step
Epoch 2/10
1/1 - 105s - loss: 86.5219 - accuracy: 0.3704 - val_loss: 32.5332 - val_accurac
y: 0.1481 - 105s/epoch - 105s/step
Epoch 3/10
1/1 - 109s - loss: 31.2061 - accuracy: 0.1481 - val_loss: 0.7165 - val_accurac
y: 0.8519 - 109s/epoch - 109s/step
Epoch 4/10
1/1 - 109s - loss: 0.7628 - accuracy: 0.8519 - val_loss: 1.8610 - val_accuracy:
0.8519 - 109s/epoch - 109s/step
Epoch 5/10
1/1 - 99s - loss: 1.8222 - accuracy: 0.8519 - val_loss: 1.0483 - val_accuracy:
0.8519 - 99s/epoch - 99s/step
Epoch 6/10
1/1 - 104s - loss: 0.9342 - accuracy: 0.8519 - val_loss: 0.7943 - val_accuracy:
0.6667 - 104s/epoch - 104s/step
Epoch 7/10
1/1 - 103s - loss: 0.6787 - accuracy: 0.8148 - val_loss: 0.5477 - val_accuracy:
0.8519 - 103s/epoch - 103s/step
Epoch 8/10
1/1 - 97s - loss: 0.4928 - accuracy: 0.8519 - val_loss: 0.3689 - val_accuracy:
0.8519 - 97s/epoch - 97s/step
Epoch 9/10
1/1 - 95s - loss: 0.3512 - accuracy: 0.8889 - val_loss: 0.3086 - val_accuracy:
0.9259 - 95s/epoch - 95s/step
Epoch 10/10
1/1 - 95s - loss: 0.3052 - accuracy: 0.9259 - val_loss: 0.3885 - val_accuracy:
0.8519 - 95s/epoch - 95s/step
```

```
In [24]: model.save('fitmodel.h5')
```

```
In [27]: from keras.preprocessing import image
import numpy as np
from keras.models import load_model
```

```
In [37]: test_image=image.load_img(r'C:\data\test\khushbu\k1.jpg',target_size=(256,256))
test_image=np.array(test_image)
test_image=np.expand_dims(test_image,axis=0)
```

In [39]: `result=model.predict((test_image))`

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

khushbu