**Team ID:PNT2022TMlD50173**

**Importing The ImageDataGenerator Library**

In [ ]:

**import** keras

**from** keras.preprocessing.image **import** ImageDataGenerator

**Define the parameters/arguments for ImageDataGenerator class**

In [ ]:

train\_datagen**=**ImageDataGenerator(rescale**=**1.**/**255,shear\_range**=**0.2,rotation\_range**=**180,zoom\_range**=**0.2, horizontal\_flip**=True**)

test\_datagen**=**ImageDataGenerator(rescale**=**1.**/**255)

**Applying ImageDataGenerator functionality to trainset**

In [ ]:

x\_train**=**train\_datagen**.**flow\_from\_directory(r'/content/drive/MyDrive/Dataset/train\_set',target\_size**=**(128,128),batch\_size**=**32, class\_mode**=**'binary')

Found 436 images belonging to 2 classes.

**Applying ImageDataGenerator functionality to testset**

In [ ]:

x\_test**=**test\_datagen**.**flow\_from\_directory(r'/content/drive/MyDrive/Dataset/test\_set',target\_size**=**(128,128),batch\_size**=**32, class\_mode**=**'binary')

Found 121 images belonging to 2 classes.

**Import model building libraries**

In [ ]:

*#To define Linear initialisation import Sequential*

**from** keras.models **import** Sequential

*#To add layers import Dense*

**from** keras.layers **import** Dense

*#To create Convolution kernel import Convolution2D*

**from** keras.layers **import** Convolution2D

*#import Maxpooling layer*

**from** keras.layers **import** MaxPooling2D

*#import flatten layer*

**from** keras.layers **import** Flatten

**import** warnings

warnings**.**filterwarnings('ignore')

**Initializing the model**

In [ ]:

model**=**Sequential()

**Add CNN Layer**

In [ ]:

model**.**add(Convolution2D(32,(3,3),input\_shape**=**(128,128,3),activation**=**'relu'))

*#add maxpooling layer*

model**.**add(MaxPooling2D(pool\_size**=**(2,2)))

*#add flatten layer*

model**.**add(Flatten())

**Add Hidden Layer**

In [ ]:

*#add hidden layer*

model**.**add(Dense(150,activation**=**'relu'))

*#add output layer*

model**.**add(Dense(1,activation**=**'sigmoid'))

**Configure the learning process**

In [ ]:

model**.**compile(loss**=**'binary\_crossentropy',optimizer**=**"adam",metrics**=**["accuracy"])

**Train the model**

In [ ]:

model**.**fit\_generator(x\_train,steps\_per\_epoch**=**14,epochs**=**10,validation\_data**=**x\_test,validation\_steps**=**4)

Epoch 1/10

14/14 [==============================] - 205s 15s/step - loss: 2.7344 - accuracy: 0.7454 - val\_loss: 0.2016 - val\_accuracy: 0.9256

Epoch 2/10

14/14 [==============================] - 20s 1s/step - loss: 0.3267 - accuracy: 0.8945 - val\_loss: 0.2290 - val\_accuracy: 0.9339

Epoch 3/10

14/14 [==============================] - 20s 1s/step - loss: 0.2991 - accuracy: 0.8922 - val\_loss: 0.0524 - val\_accuracy: 0.9835

Epoch 4/10

14/14 [==============================] - 20s 1s/step - loss: 0.2418 - accuracy: 0.9174 - val\_loss: 0.1570 - val\_accuracy: 0.9421

Epoch 5/10

14/14 [==============================] - 20s 1s/step - loss: 0.1984 - accuracy: 0.9083 - val\_loss: 0.0767 - val\_accuracy: 0.9752

Epoch 6/10

14/14 [==============================] - 20s 1s/step - loss: 0.1643 - accuracy: 0.9335 - val\_loss: 0.0749 - val\_accuracy: 0.9752

Epoch 7/10

14/14 [==============================] - 20s 1s/step - loss: 0.1538 - accuracy: 0.9312 - val\_loss: 0.1264 - val\_accuracy: 0.9421

Epoch 8/10

14/14 [==============================] - 20s 1s/step - loss: 0.1732 - accuracy: 0.9266 - val\_loss: 0.0652 - val\_accuracy: 0.9835

Epoch 9/10

14/14 [==============================] - 20s 1s/step - loss: 0.1514 - accuracy: 0.9358 - val\_loss: 0.0567 - val\_accuracy: 0.9835

Epoch 10/10

14/14 [==============================] - 20s 1s/step - loss: 0.1445 - accuracy: 0.9404 - val\_loss: 0.0448 - val\_accuracy: 0.9917

Out[ ]:

**Save The Model**

In [ ]:

model**.**save("forest1.h5")

**Predictions**

In [ ]:

*#import load\_model from keras.model*

**from** keras.models **import** load\_model

*#import image class from keras*

**from** tensorflow.keras.preprocessing **import** image *#import numpy*

**import** numpy **as** np

*#import cv2*

**import** cv2

In [ ]:

*#load the saved model*

model **=** load\_model("forest1.h5")

In [ ]:

img**=**image**.**load\_img(r'/content/drive/MyDrive/Dataset/test\_set/forest/0.48007200\_1530881924\_final\_forest.jpg')

x**=**image**.**img\_to\_array(img)

res **=** cv2**.**resize(x, dsize**=**(128, 128), interpolation**=**cv2**.**INTER\_CUBIC)

*#expand the image shape*

x**=**np**.**expand\_dims(res,axis**=**0)

In [ ]:

pred**=** model**.**predict(x)

1/1 [==============================] - 0s 94ms/step

In [ ]:

pred

Out[ ]:

array([[0.]], dtype=float32)