EARLY DETECTION OF FOREST FIRE USING DEEP LEARNING

MODEL BUILDING PREDICTIONS

| Team ID | PNT2022TMID52430 |
|--------------|---|
| Project Name | Project-Early detection of forest fire using deep |
| | learning |

PREDICTIONS:

The last and final step is to make use of our saved model to do predictions. For that we have a class in keras called load_model. Load_model is used to load our saved model h5 file (alert.h5).

IMPORT LIBRARIES:

11/7/22, 12:35 AM

Untitled8.ipynb - Colaboratory

Importing Keras libraries

import keras

Importing ImageDataGenerator from Keras

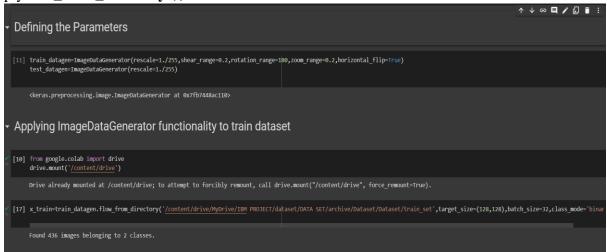
from keras.preprocessing.image import ImageDataGenerator

IMPORT ImageDataGenerator FROM KERAS:

| → Impo | rting Keras libraries |
|--|--|
| | port keras |
| - Impo | rting ImageDataGenerator from Keras |
| | om matplotlib import pyplot as plt om keras.preprocessing.image import ImageDataGenerator |
| → Defin | ing the Parameters |
| | ain_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range=180,zoom_range=0.2,horizontal_flip=True) st_datagen=ImageDataGenerator(rescale=1./255) |
| □, <ke< td=""><td>eras.preprocessing.image.ImageDataGenerator at 0x7fb7448ac110></td></ke<> | eras.preprocessing.image.ImageDataGenerator at 0x7fb7448ac110> |

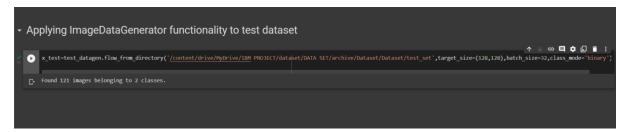
APPLYING ImageDataGenerator to train dataset:

plyflow_from_directory ()methodfor Train folder.



APPLYING ImageDataGenerator to test dataset:

Applying the **flow_from_directory** () methodfortest folder.



IMPORTING MODEL BUILDING LIBRARIES:

11/8/22, 1:16 AM Main code - Colaboratory

Importing Model Building Libraries

```
#to define the linear Initialisation import sequential
from keras.models import Sequential
#to add layers import Dense
from keras.layers import Dense
#to create Convolutional 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:

Initializing the model

```
model=Sequential()
```

ADDING CNN LAYERS:

Adding CNN Layers

```
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
#add maxpooling layers
model.add(MaxPooling2D(pool_size=(2,2)))
#add faltten layer
model.add(Flatten())
```

ADDING DENSE LAYERS:

Add Dense layers

```
#add hidden layers
model.add(Dense(150,activation='relu'))
#add output layer
model.add(Dense(1,activation='sigmoid'))
```

CONFIGURING THE LEARNING PROCESS:

configuring the learning process

```
model.compile(loss='binary_crossentropy',optimizer="adam",metrics=["accuracy"])
```

TRAINING THE MODEL:

Training the model

```
model.fit_generator(x_train,steps_per_epoch=14,epochs=10,validation_data=x_test,validation_
 Epoch 2/10
 Epoch 3/10
 Epoch 4/10
 Epoch 5/10
 Epoch 6/10
 Epoch 7/10
 14/14 [============== ] - 32s 2s/step - loss: 0.1781 - accuracy: 0.928
 Epoch 8/10
 Epoch 9/10
 Epoch 10/10
 <keras.callbacks.History at 0x7fd537101390>
```

SAVE THE MODEL:

Save the model

```
model.save("forest.h5")
```

PREDICTIONS:

→ Predictions

```
#import load model from keras.model
from keras.models import load_model
#import image from keras
from tensorflow.keras.preprocessing import image
import numpy as np
#import cv2
import cv2
#load the saved model
model=load_model('forest.h5')
img=image.load_img('_/content/drive/MyDrive/IBM_ PROJECT/dataset/DATA_SET/archive/Dataset/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/Data_set/D
```