A PROJECT REPORT ON

FOREST COMBUSTION RECOGNITION USING AI

Using Convolution Neural Networks (CNN)

Done by

Team - 2

K. Dhanith Krishna

V. Subash

B. Kusula Kumari

K. Jahnavi

Submitted to

SmartBridge Educational Services Pvt Ltd.

SMARTBRIDGE
Let's Bridge the Gap

CONTENTS:

1.INTRODUCTION

1.1Overview

Forest fires are a major environmental issue, creating economic and ecological damage while endangering human lives. There are typically about 100,000 wildfires in the United States every year. Over 9 million acres of land have been destroyed due to treacherous wildfires. It is difficult to predict and detect Forest Fire in a sparsely populated forest area and it is more difficult if the prediction is done using ground based methods like Camera or Video Based approach. Satellites can be an important source of data prior to and also during the Fire due to its reliability and efficiency.Forest fires destroy a total area of 3.5 to 4.5 million km.Here we are going to train the model in such a way that if you give an image/picture of the forest it will detect whether the forest is with fire or without fire. And if it recognizes the fire picture it puts on an alarm sound. This helps us to find out the forest combustion easily.

1.2 Purpose

Forest is one of the major wealth of our country. Forests provide enormous material goods and environmental services. They are useful for industry as well as rural economic growth. At the same time, when the forest is under fire it emits lot of carbon dioxide leads to climate change and global warming. So the forest fire has to be detected at earlier stage. For example , recently we have seen forest fires in Australia. Approximately 18,636,079 hectares (46,050,750 acres) off forest area had undergone combustion. So, due to isolation, inaccessibility, tough weather, shortage of frontier staff, the early finding of forest fire is a difficult task we have developed a model to detect forest fire combustion. Anyways to do it manually we need lot of man power so to overcome that we are implementing CNN (Convolutional Neural Network) Model, which is a deep learning technique to predict the fire in forest.

2. LITERATURE SURVEY

2.1 Existing problem

Now days there are lot of existing systems for forest combustion recognition, one system differ from another systems, few of them doesn’t detect properly whether fire is present or not . Some systems are developed based on IOT Applications, we have to keep that hardware system in forest. Due to weather conditions, like when rainfalls few systems may be destroyed. So, then it will be difficult to detect. Then our model comes into existence, works irrespective of the weather conditions.

2.2 Proposed solution

The proposed Forest Fire Prediction model is based neural networks(CNN) incorporated to an alert system. The developmental approach of the proposed system includes two modules:

- Forest Fire Identification: Identification of fire affected areas.
- Forest Fire Management: Remedial measures for Forest Fire is all about detecting where the fire initiated in the forest.

In our proposed system we made an effort to classify the images of forest with fire and forest without fire for further using convolution neural networks (CNNs).Since the CNNs are capable of handling a large amount of data and can estimate the features automatically, they have been utilized for the task of forest combustion recognition. The forest dataset of train and test has been selected as the working GUI for this approach. Here we select a sample of few images (approx..100) with fire and without fire segregate them into train and test and model predicts them perfectly depending upon accuracy.

3. THEORITICAL ANALYSIS

3.1 Block diagram

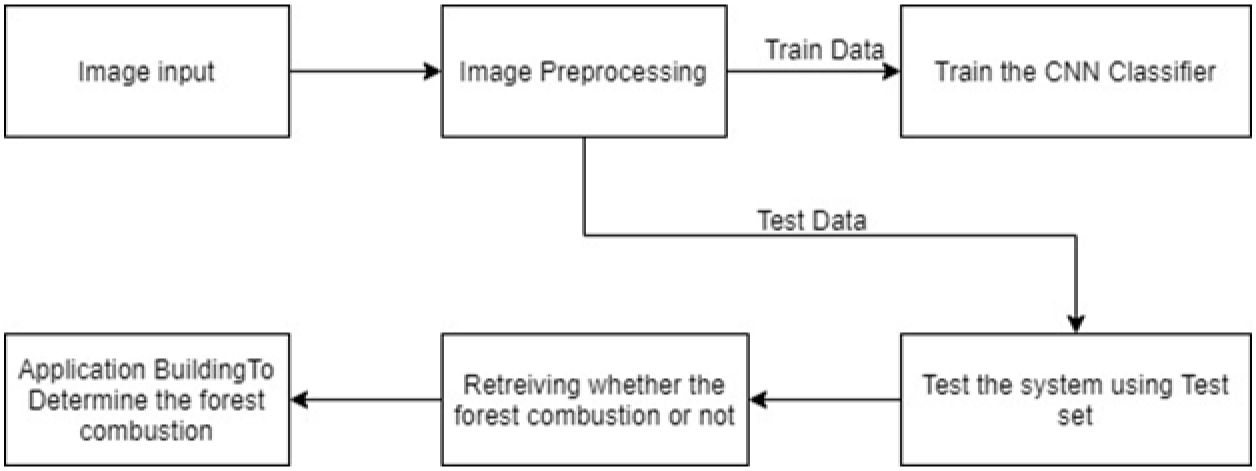


Image input
Image Preprocessing
Train Data
Train the CNN Classifier
Test Data
Test the system using Test set
Retreiving whether the forest combustion or not
Application BuildingTo Determine the forest combustion

3.2 Hardware/Software designing

Hardware

1) A laptop / computer
2) 8-GB RAM
3) 32-GB ROM
4) 64 Bit

<u>Software</u>

Front End:

1. HTML

Back End:

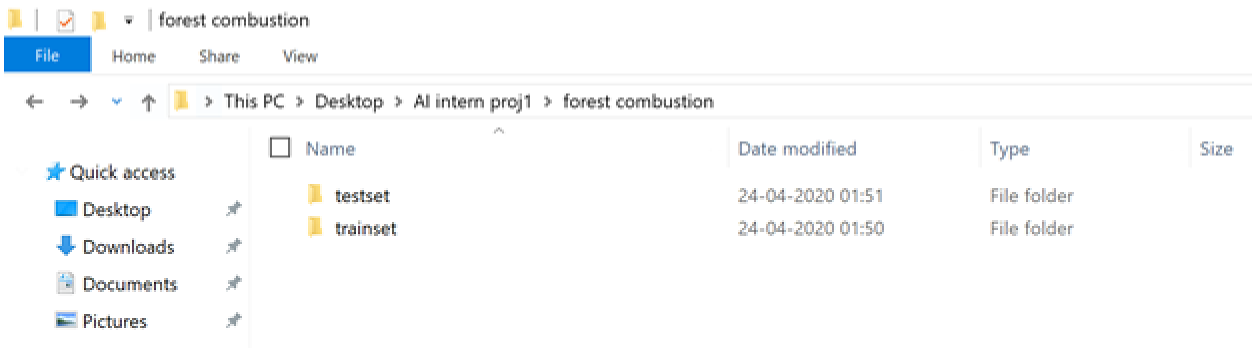
Python – We use following GUI's

1. (Anaconda Navigator -> Jupyter Notebook, Spyder) it is a desktop GUI
2. Anaconda Prompt

4. EXPERIMENTAL INVESTIGATION

Screenshots of accuracy and what steps we have done during the model building include:

Data Collection – Create Train and Test Folders



Name	Date modified	Type	Size
testset	24-04-2020 01:51	File folder	
trainset	24-04-2020 01:50	File folder	

Flask folder -

Steps to be followed:

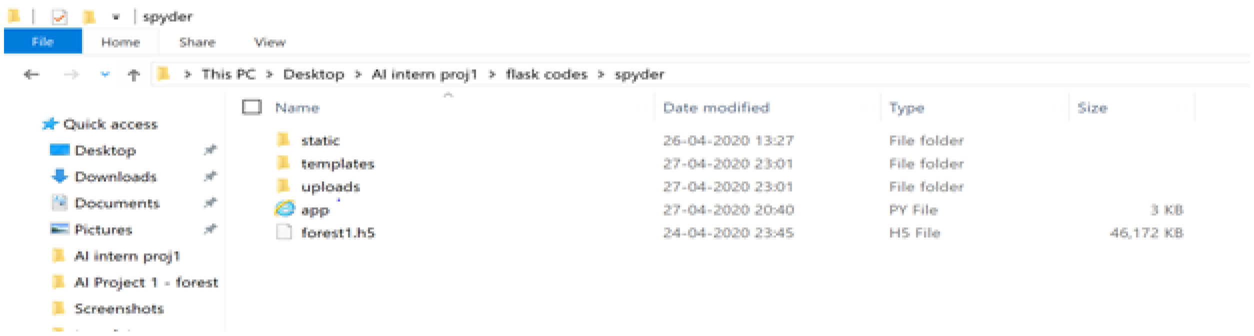
Create a main folder (spyder)

In that main folder create template folder and in that place .html file (base.html)

And create two folders static (contain css and jss) and uploads (images of prediction)

.py file (app.py) should be saved in that main folder

.h5 file (forest1.h5) which is downloaded from jupyter notebook should also be saved in main folder.



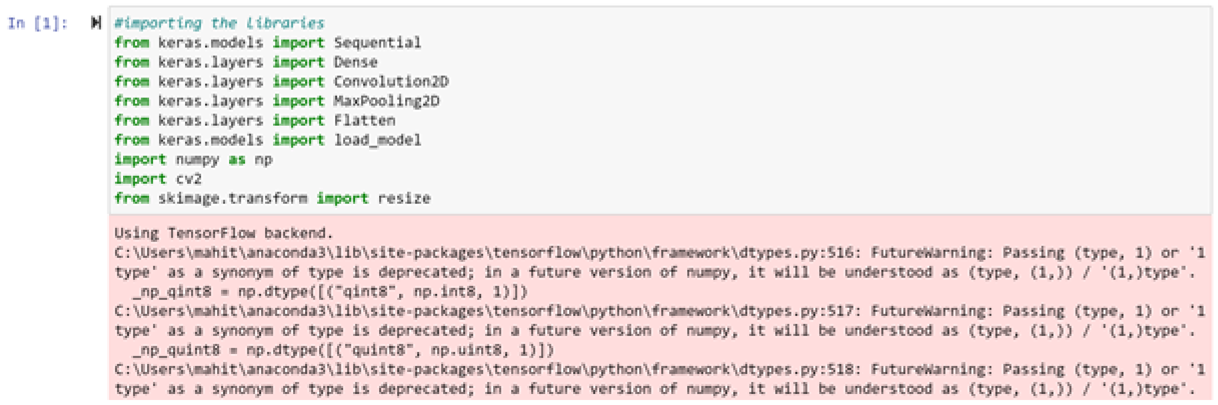
Name	Date modified	Type	Size
static	26-04-2020 13:27	File folder	
templates	27-04-2020 23:01	File folder	
uploads	27-04-2020 23:01	File folder	
app	27-04-2020 20:40	PY File	3 KB
forest1.h5	24-04-2020 23:45	H5 File	46,172 KB

Data Preprocessing

Import the ImageDataGenerator Library

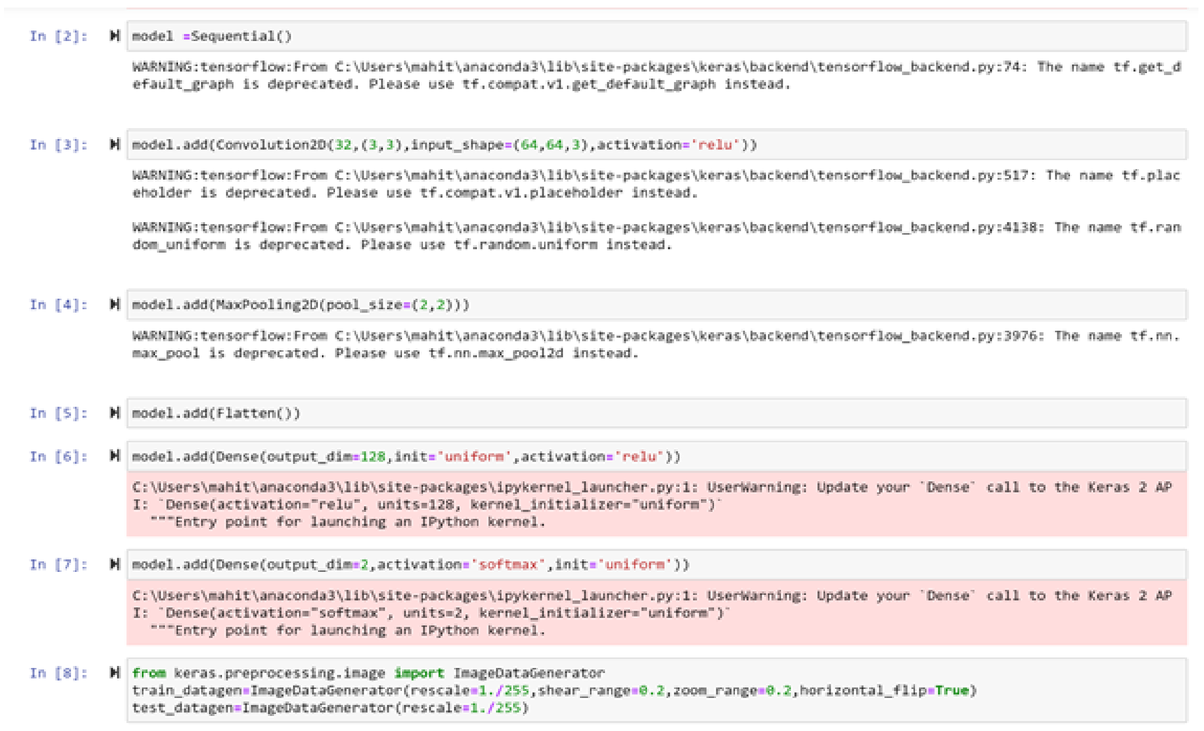
Configure ImageDataGenerator Class

Apply ImageDataGenerator library to Trainset and Testset



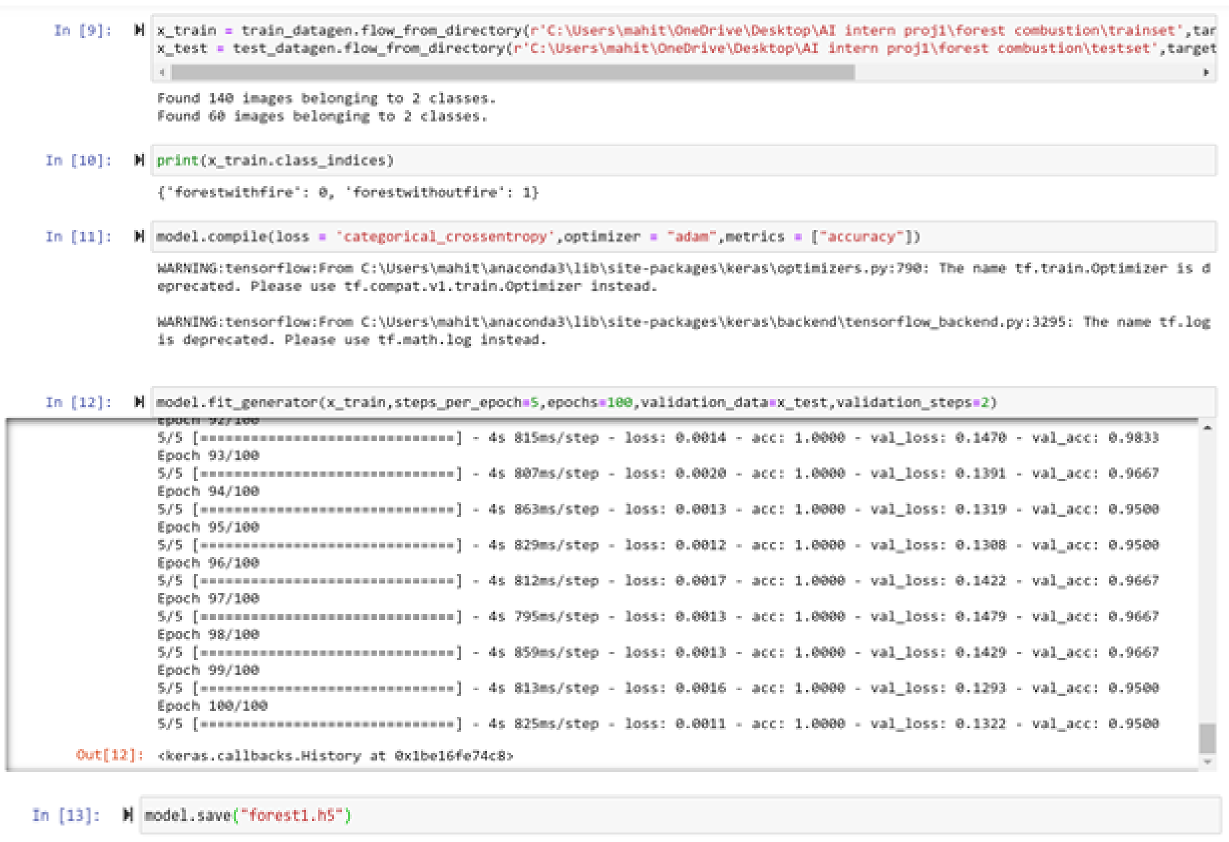
```
In [1]:
#importing the libraries
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.models import load_model
import numpy as np
import cv2
from skimage.transform import resize
```

```
Using TensorFlow backend.
C:\Users\mahit\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:516: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
C:\Users\mahit\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:517: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
C:\Users\mahit\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:518: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
```



```
In [2]:  model =Sequential()

         WARNING:tensorflow:From C:\Users\mahit\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:74: The name tf.get_d
         efault_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.

In [3]:  model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation='relu'))

         WARNING:tensorflow:From C:\Users\mahit\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:517: The name tf.plac
         eholder is deprecated. Please use tf.compat.v1.placeholder instead.

         WARNING:tensorflow:From C:\Users\mahit\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:4138: The name tf.ran
         dom_uniform is deprecated. Please use tf.random.uniform instead.

In [4]:  model.add(MaxPooling2D(pool_size=(2,2)))

         WARNING:tensorflow:From C:\Users\mahit\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:3976: The name tf.nn.
         max_pool is deprecated. Please use tf.nn.max_pool2d instead.

In [5]:  model.add(Flatten())

In [6]:  model.add(Dense(output_dim=128,init='uniform',activation='relu'))

         C:\Users\mahit\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your `Dense` call to the Keras 2 AP
         I: `Dense(activation="relu", units=128, kernel_initializer="uniform")`
           """Entry point for launching an IPython kernel.

In [7]:  model.add(Dense(output_dim=2,activation='softmax',init='uniform'))

         C:\Users\mahit\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your `Dense` call to the Keras 2 AP
         I: `Dense(activation="softmax", units=2, kernel_initializer="uniform")`
           """Entry point for launching an IPython kernel.

In [8]:  from keras.preprocessing.image import ImageDataGenerator
         train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
         test_datagen=ImageDataGenerator(rescale=1./255)
```

Accuracy when we train and test the model we got 95%



```
In [9]:  x_train = train_datagen.flow_from_directory(r'C:\Users\mahit\OneDrive\Desktop\AI intern proj1\forest combustion\trainset',tar
         x_test = test_datagen.flow_from_directory(r'C:\Users\mahit\OneDrive\Desktop\AI intern proj1\forest combustion\testset',target

         Found 140 images belonging to 2 classes.
         Found 60 images belonging to 2 classes.

In [10]: print(x_train.class_indices)

         {'forestwithfire': 0, 'forestwithoutfire': 1}

In [11]: model.compile(loss = 'categorical_crossentropy',optimizer = "adam",metrics = ["accuracy"])

         WARNING:tensorflow:From C:\Users\mahit\anaconda3\lib\site-packages\keras\optimizers.py:790: The name tf.train.Optimizer is d
         eprecated. Please use tf.compat.v1.train.Optimizer instead.

         WARNING:tensorflow:From C:\Users\mahit\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:3295: The name tf.log
         is deprecated. Please use tf.math.log instead.

In [12]: model.fit_generator(x_train,steps_per_epoch=5,epochs=100,validation_data=x_test,validation_steps=2)
         Epoch 92/100
         5/5 [==============================] - 4s 815ms/step - loss: 0.0014 - acc: 1.0000 - val_loss: 0.1470 - val_acc: 0.9833
         Epoch 93/100
         5/5 [==============================] - 4s 807ms/step - loss: 0.0020 - acc: 1.0000 - val_loss: 0.1391 - val_acc: 0.9667
         Epoch 94/100
         5/5 [==============================] - 4s 863ms/step - loss: 0.0013 - acc: 1.0000 - val_loss: 0.1319 - val_acc: 0.9500
         Epoch 95/100
         5/5 [==============================] - 4s 829ms/step - loss: 0.0012 - acc: 1.0000 - val_loss: 0.1308 - val_acc: 0.9500
         Epoch 96/100
         5/5 [==============================] - 4s 812ms/step - loss: 0.0017 - acc: 1.0000 - val_loss: 0.1422 - val_acc: 0.9667
         Epoch 97/100
         5/5 [==============================] - 4s 795ms/step - loss: 0.0013 - acc: 1.0000 - val_loss: 0.1479 - val_acc: 0.9667
         Epoch 98/100
         5/5 [==============================] - 4s 859ms/step - loss: 0.0013 - acc: 1.0000 - val_loss: 0.1429 - val_acc: 0.9667
         Epoch 99/100
         5/5 [==============================] - 4s 813ms/step - loss: 0.0016 - acc: 1.0000 - val_loss: 0.1293 - val_acc: 0.9500
         Epoch 100/100
         5/5 [==============================] - 4s 825ms/step - loss: 0.0011 - acc: 1.0000 - val_loss: 0.1322 - val_acc: 0.9500

Out[12]: <keras.callbacks.History at 0x1be16fe74c8>

In [13]: model.save("forest1.h5")
```

Model Building

Importing the building libraries

Initializing the model

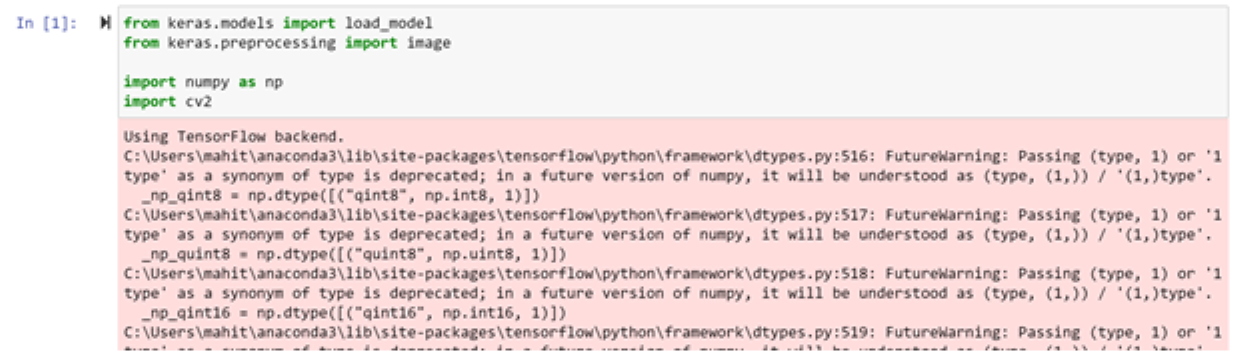
Loading the preprocessed data

Adding CNN and Dense Layers

Configure the learning process

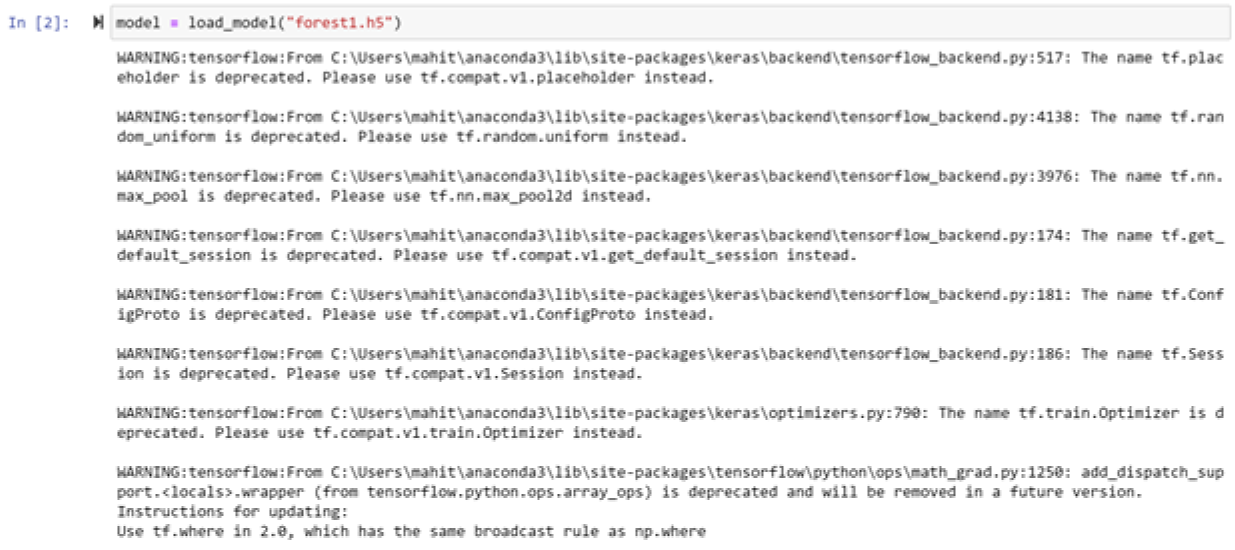
Train and Test Model

Optimize and Save Model



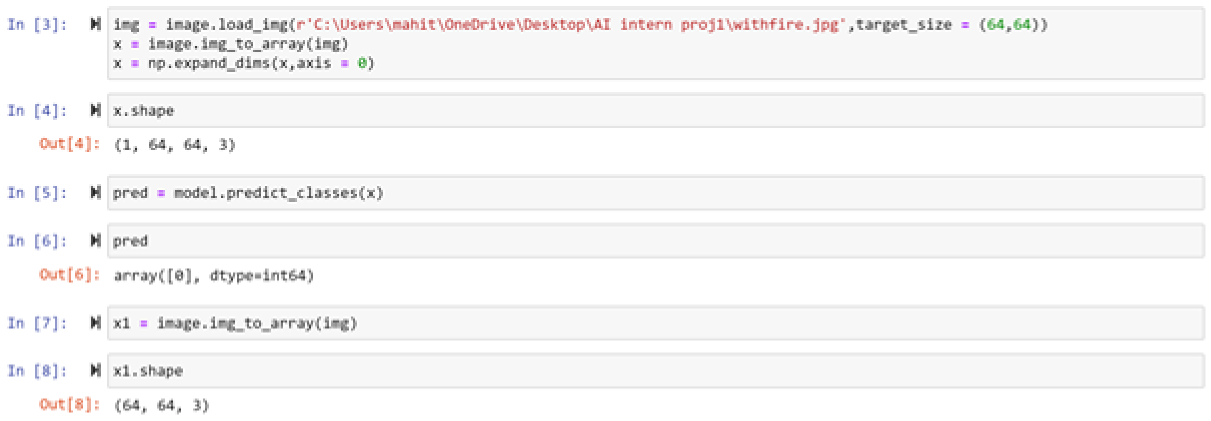
```
In [1]:  from keras.models import load_model
         from keras.preprocessing import image

         import numpy as np
         import cv2
```

```
Using TensorFlow backend.
C:\Users\mahit\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:516: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
C:\Users\mahit\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:517: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
C:\Users\mahit\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:518: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
C:\Users\mahit\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:519: FutureWarning: Passing (type, 1) or '1
```



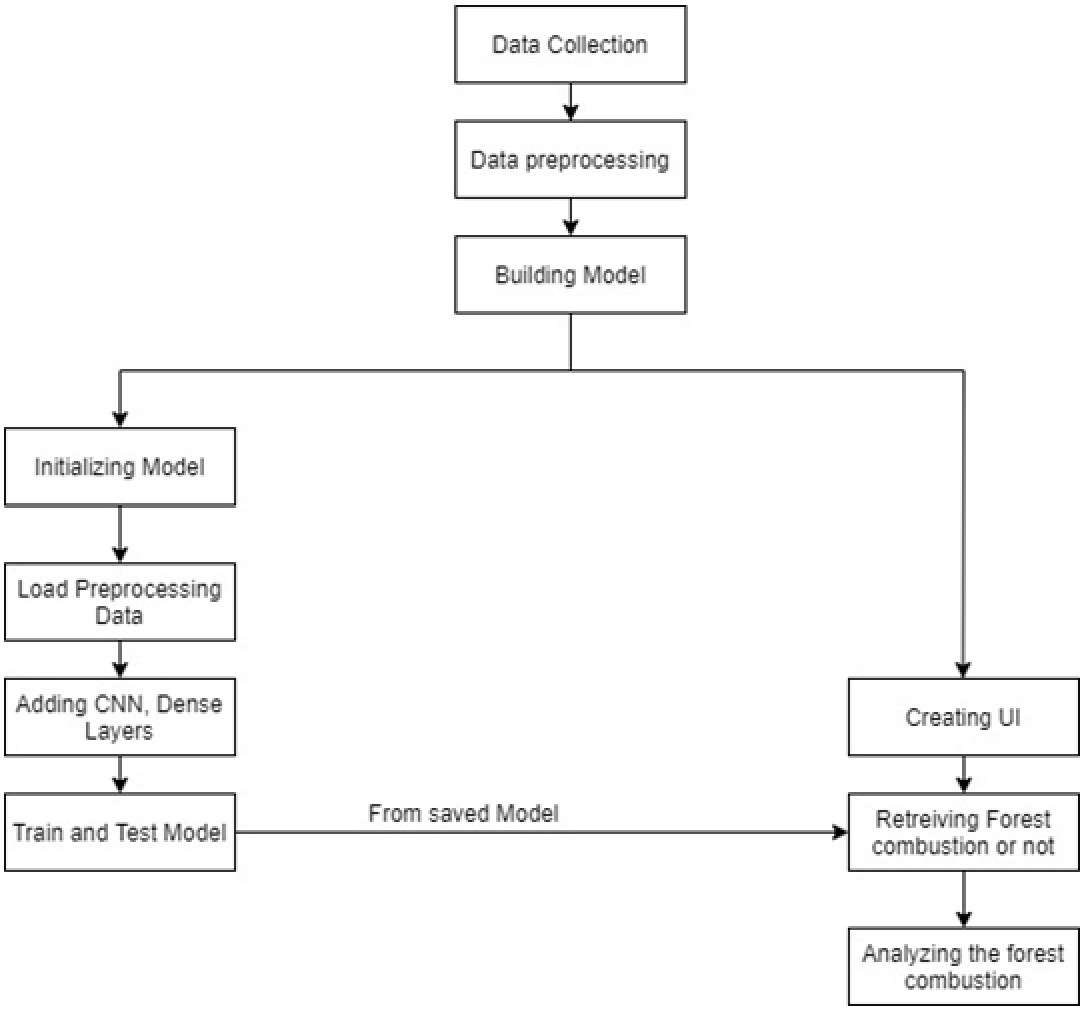
```
In [2]:  model = load_model("forest1.h5")
```

```
WARNING:tensorflow:From C:\Users\mahit\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:517: The name tf.plac
eholder is deprecated. Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From C:\Users\mahit\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:4138: The name tf.ran
dom_uniform is deprecated. Please use tf.random.uniform instead.

WARNING:tensorflow:From C:\Users\mahit\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:3976: The name tf.nn.
max_pool is deprecated. Please use tf.nn.max_pool2d instead.

WARNING:tensorflow:From C:\Users\mahit\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:174: The name tf.get_
default_session is deprecated. Please use tf.compat.v1.get_default_session instead.

WARNING:tensorflow:From C:\Users\mahit\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:181: The name tf.Conf
igProto is deprecated. Please use tf.compat.v1.ConfigProto instead.

WARNING:tensorflow:From C:\Users\mahit\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:186: The name tf.Sess
ion is deprecated. Please use tf.compat.v1.Session instead.

WARNING:tensorflow:From C:\Users\mahit\anaconda3\lib\site-packages\keras\optimizers.py:790: The name tf.train.Optimizer is d
eprecated. Please use tf.compat.v1.train.Optimizer instead.

WARNING:tensorflow:From C:\Users\mahit\anaconda3\lib\site-packages\tensorflow\python\ops\math_grad.py:1250: add_dispatch_sup
port.<locals>.wrapper (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
```



```
In [3]:  img = image.load_img(r'C:\Users\mahit\OneDrive\Desktop\AI intern proj1\withfire.jpg',target_size = (64,64))
         x = image.img_to_array(img)
         x = np.expand_dims(x,axis = 0)

In [4]:  x.shape

Out[4]: (1, 64, 64, 3)

In [5]:  pred = model.predict_classes(x)

In [6]:  pred

Out[6]: array([0], dtype=int64)

In [7]:  x1 = image.img_to_array(img)

In [8]:  x1.shape

Out[8]: (64, 64, 3)
```

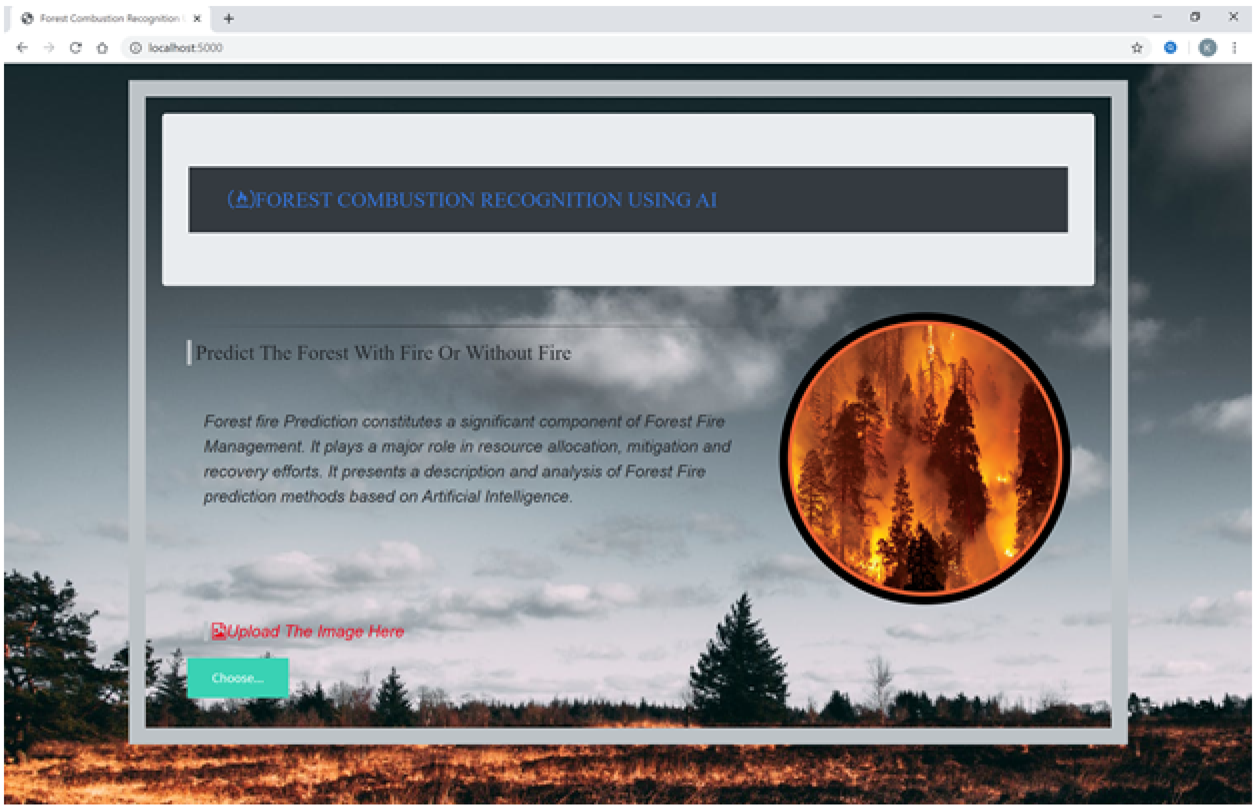
5. FLOWCHART



6. RESULT

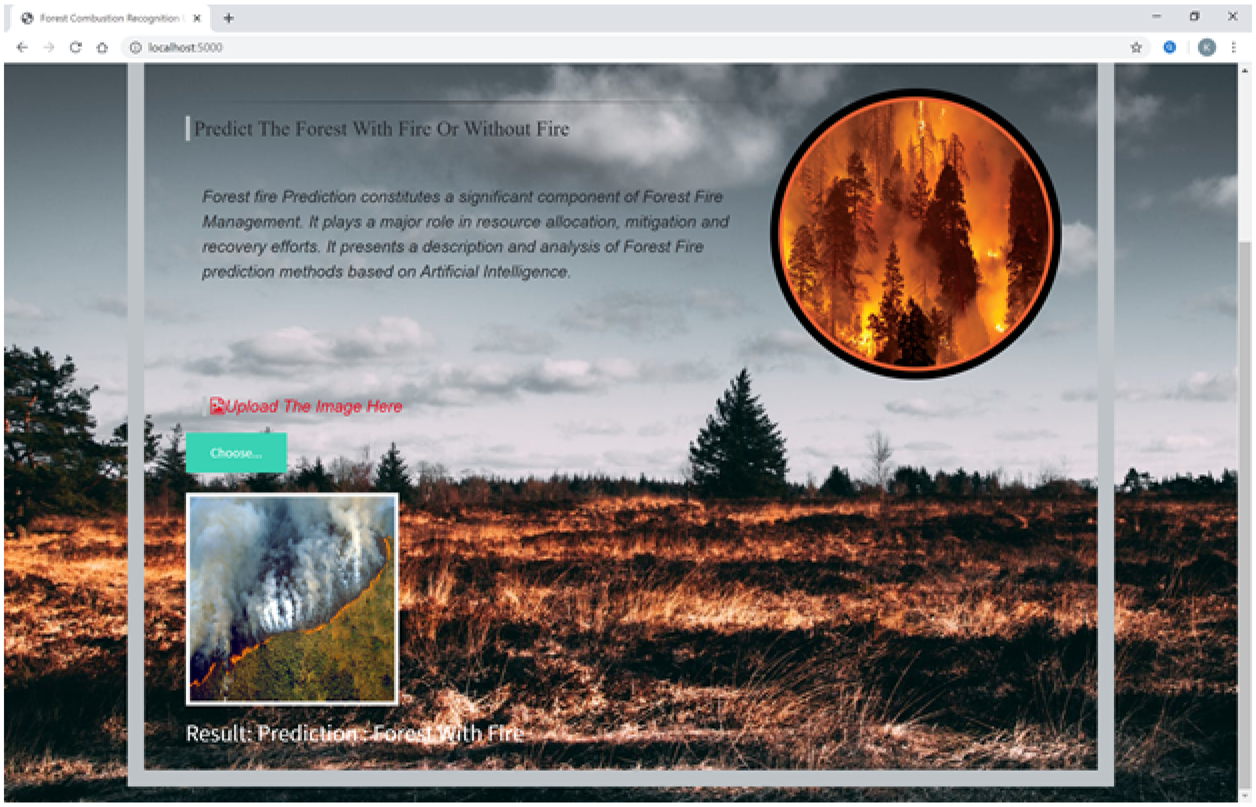
As a result, when we uploaded an image by choosing it we will get the predict button then after clicking that the model will predict whether it is forest with fire or without fire.

First the page looks as such,


Forest Combustion Recognition
localhost:5000
FOREST COMBUSTION RECOGNITION USING AI
Upload The Image Here
Choose...

Predict The Forest With Fire Or Without Fire

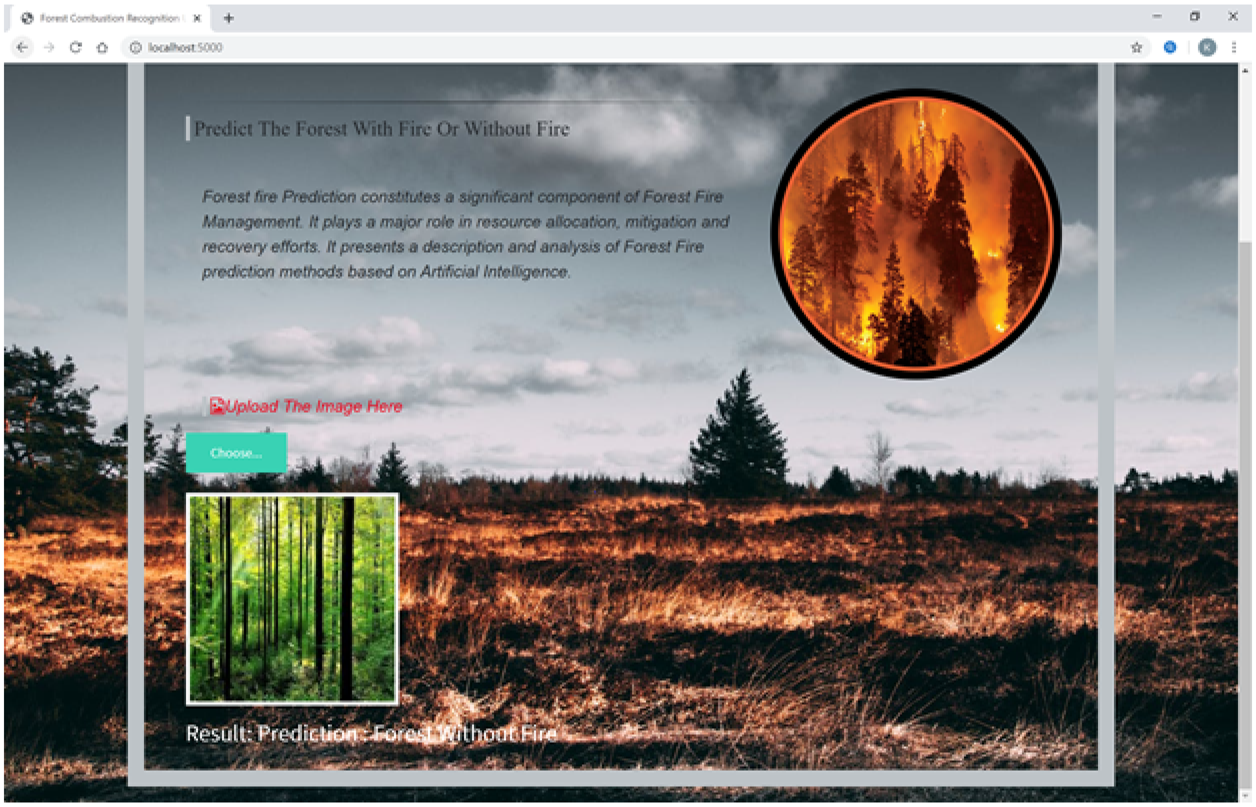
Forest fire Prediction constitutes a significant component of Forest Fire Management. It plays a major role in resource allocation, mitigation and recovery efforts. It presents a description and analysis of Forest Fire prediction methods based on Artificial Intelligence.



Predict The Forest With Fire Or Without Fire

Forest fire Prediction constitutes a significant component of Forest Fire Management. It plays a major role in resource allocation, mitigation and recovery efforts. It presents a description and analysis of Forest Fire prediction methods based on Artificial Intelligence.

Result: Prediction : Forest With Fire



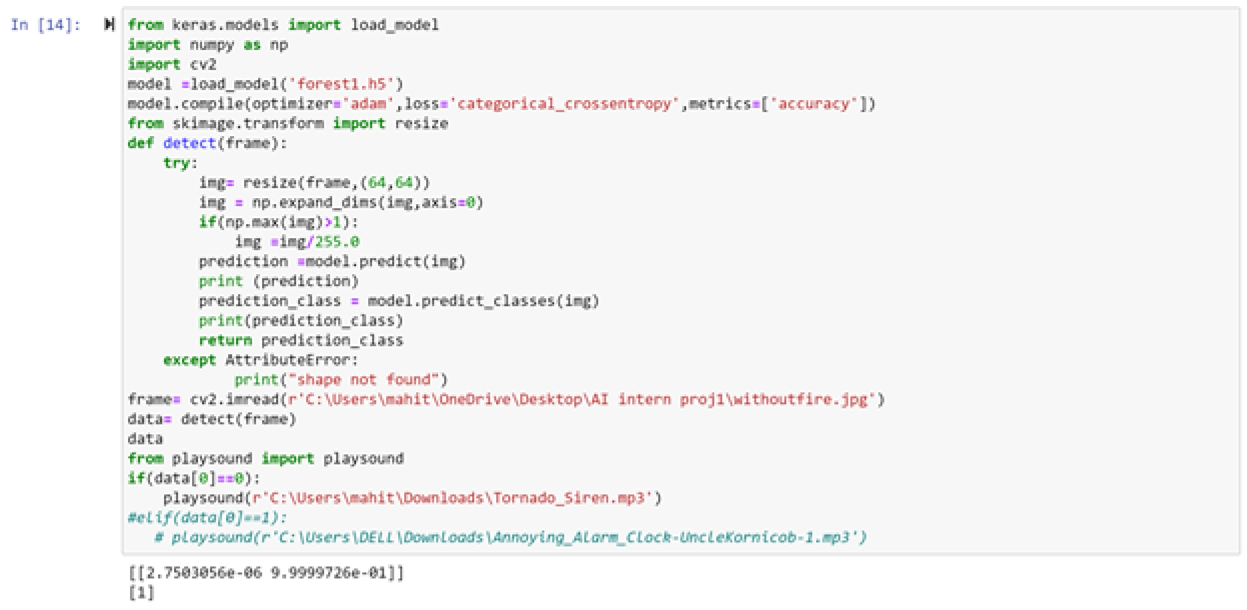
Predict The Forest With Fire Or Without Fire

Forest fire Prediction constitutes a significant component of Forest Fire Management. It plays a major role in resource allocation, mitigation and recovery efforts. It presents a description and analysis of Forest Fire prediction methods based on Artificial Intelligence.

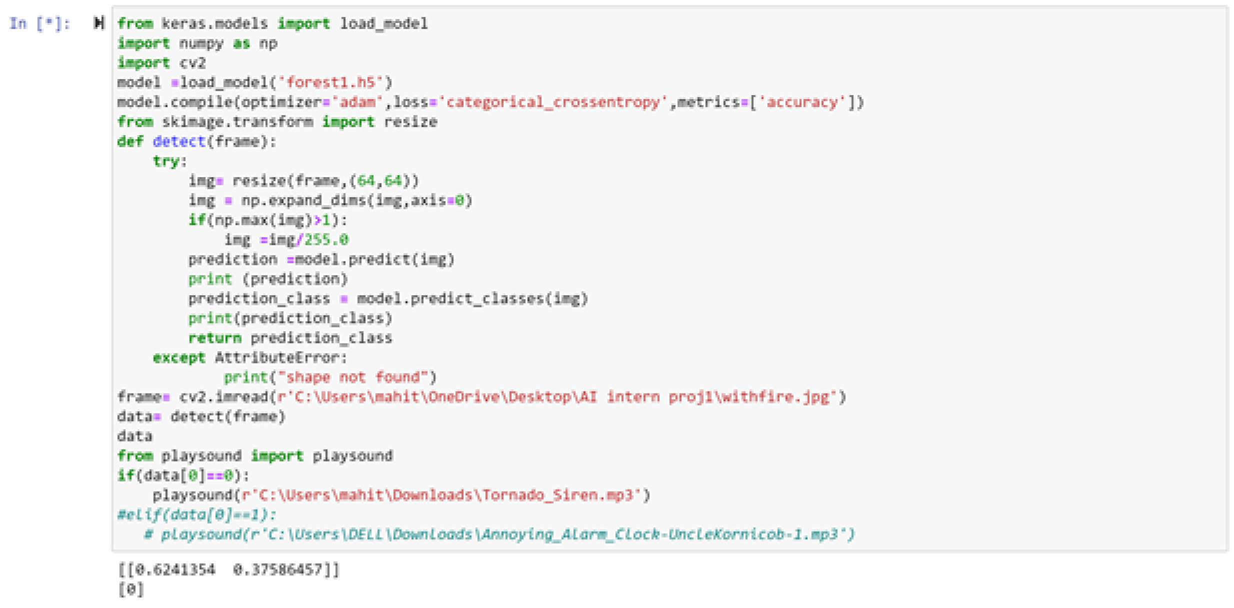
Result: Prediction : Forest Without Fire

Result of alert alarm in jupyter Notebook

It will predict as following different outputs, when we insert an image with fire we will get fire alarm in our system or it doesn't produce any kind of sound in the system.

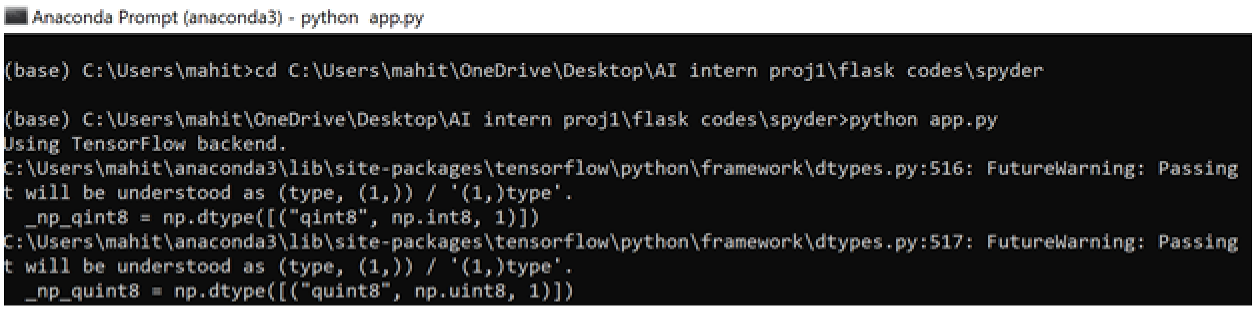


```
In [14]:
from keras.models import load_model
import numpy as np
import cv2
model =load_model('forest1.h5')
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
from skimage.transform import resize
def detect(frame):
    try:
        img= resize(frame,(64,64))
        img = np.expand_dims(img,axis=0)
        if(np.max(img)>1):
            img =img/255.0
        prediction =model.predict(img)
        print (prediction)
        prediction_class = model.predict_classes(img)
        print(prediction_class)
        return prediction_class
    except AttributeError:
            print("shape not found")
frame= cv2.imread(r'C:\Users\mahit\OneDrive\Desktop\AI intern proj1\withoutfire.jpg')
data= detect(frame)
data
from playsound import playsound
if(data[0]==0):
    playsound(r'C:\Users\mahit\Downloads\Tornado_Siren.mp3')
#elif(data[0]==1):
   # playsound(r'C:\Users\DELL\Downloads\Annoying_Alarm_Clock-UncleKornicob-1.mp3')

[[2.7503056e-06 9.9999726e-01]]
[1]
```

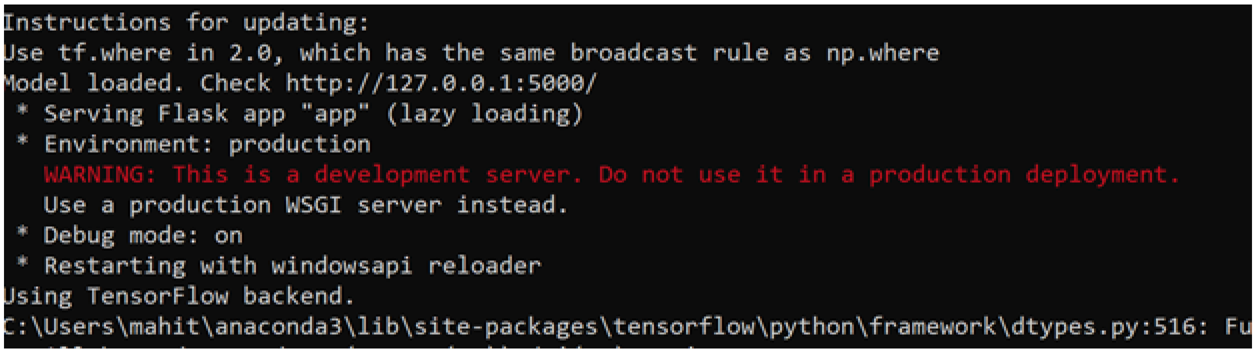


```
In [*]:
from keras.models import load_model
import numpy as np
import cv2
model =load_model('forest1.h5')
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
from skimage.transform import resize
def detect(frame):
    try:
        img= resize(frame,(64,64))
        img = np.expand_dims(img,axis=0)
        if(np.max(img)>1):
            img =img/255.0
        prediction =model.predict(img)
        print (prediction)
        prediction_class = model.predict_classes(img)
        print(prediction_class)
        return prediction_class
    except AttributeError:
            print("shape not found")
frame= cv2.imread(r'C:\Users\mahit\OneDrive\Desktop\AI intern proj1\withfire.jpg')
data= detect(frame)
data
from playsound import playsound
if(data[0]==0):
    playsound(r'C:\Users\mahit\Downloads\Tornado_Siren.mp3')
#elif(data[0]==1):
   # playsound(r'C:\Users\DELL\Downloads\Annoying_Alarm_Clock-UncleKornicob-1.mp3')

[[0.6241354  0.37586457]]
[0]
```

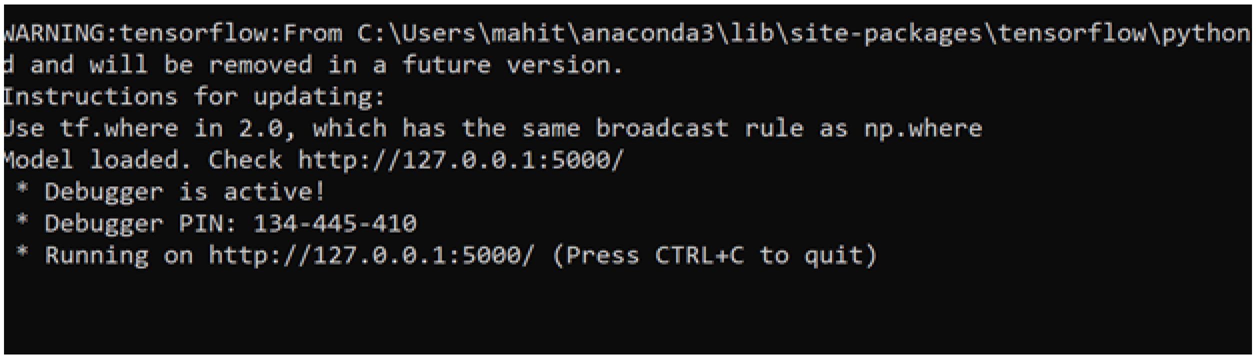
In loading the UI Page Anaconda Prompt is Used. Connects to the Server (localhost:5000)



```
Anaconda Prompt (anaconda3) - python  app.py
(base) C:\Users\mahit>cd C:\Users\mahit\OneDrive\Desktop\AI intern proj1\flask codes\spyder

(base) C:\Users\mahit\OneDrive\Desktop\AI intern proj1\flask codes\spyder>python app.py
Using TensorFlow backend.
C:\Users\mahit\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:516: FutureWarning: Passing
t will be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
C:\Users\mahit\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:517: FutureWarning: Passing
t will be understood as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
```



```
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
Model loaded. Check http://127.0.0.1:5000/
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with windowsapi reloader
Using TensorFlow backend.
C:\Users\mahit\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:516: Fu
```



```
WARNING:tensorflow:From C:\Users\mahit\anaconda3\lib\site-packages\tensorflow\python
d and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
Model loaded. Check http://127.0.0.1:5000/
 * Debugger is active!
 * Debugger PIN: 134-445-410
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

7. ADVANTAGES & DISADVANTAGES

ADVANTAGES

- The main advantage of detecting defects in building is people can able to easily access the software whenever they need.
- Also just by predicting they would able to rectify which type of problem they are facing to their buildings.
- This will save the money to human beings who are in need.

DISADVANTAGES

- This type of detections will cut of the employment in the present world.
- We need lot of training data to predict the accurate one.
- High computational cost.

8. APPLICATIONS

- This system is useful for all the people around the globe who want to detect the fire when occurred in forest areas.
- Our system is user friendly web app which is used to make human life easier just by
  giving the image they can recognize whether it is with fire or not.

9. CONCLUSION

This study proposes an effective forest fire detection method using image processing techniques. The algorithm uses Train and test model. It will train our model and further it test whether fire is present or not. Five fire detection rules are applied to detect the fire. The performance of the proposed algorithm is tested on data set consisting of various pictures of forest collected from Internet, 100 of which were actual fire pictures , while another 100 were without fire .The results show that the proposed algorithm achieves good detection rates. These results indicate that the proposed method is accurate and can be used in automatic forest fire-alarm systems.

10. FUTURE SCOPE

The vital requirements of an autonomous early forest combustion recognition system, its main modules and methods for wildfire management and risk assessment, smoke and combustion detection, it have been done based on images. We have trained and tested the model got the accuracy of 95% and got the alarm sound in Jupyter Notebook. Further we wanted to develop our UI linked to the html page by getting the alarm in our system and make our model to run more rapidly and accurately.

11. BIBILOGRAPHY

https://www.researchgate.net/publication/261272818_Artificial_intelligence_for_forest_fire_prediction

https://www.sciencedirect.com/topics/earth-and-planetary-sciences/forest-fire

APPENDIX

A. Source Code :

```
#importing the libraries

from keras.models import Sequential

from keras.layers import Dense

from keras.layers import Convolution2D

from keras.layers import MaxPooling2D

from keras.layers import Flatten

from keras.models import load_model

import numpy as np

import cv2

from skimage.transform import resize

model =Sequential()

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

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

model.add(Flatten())

model.add(Dense(output_dim=128,init='uniform',activation='relu'))

model.add(Dense(output_dim=2,activation='softmax',init='uniform'))

from keras.preprocessing.image import ImageDataGenerator

train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_fli
p=True)

test_datagen=ImageDataGenerator(rescale=1./255)

x_train = train_datagen.flow_from_directory(r'C:\Users\mahit\OneDrive\Desktop\AI intern
proj1\forest combustion\trainset',target_size = (64,64),batch_size = 32,class_mode = 'categorical')

x_test = test_datagen.flow_from_directory(r'C:\Users\mahit\OneDrive\Desktop\AI intern
proj1\forest combustion\testset',target_size = (64,64),batch_size = 32,class_mode = 'categorical')

print(x_train.class_indices)

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

model.fit_generator(x_train,steps_per_epoch=5,epochs=100,validation_data=x_test,validation_step
s=2)

model.save("forest1.h5")

img = image.load_img(r'C:\Users\mahit\OneDrive\Desktop\AI intern proj1\withfire.jpg',target_size = (64,64))

x = image.img_to_array(img)

x = np.expand_dims(x,axis = 0)

x.shape

pred = model.predict_classes(x)

pred

x1 = image.img_to_array(img)

x1.shape
```