# **PROJECT REPORT**

On

# **Fire Detection**

Submitted to Kurukshetra University in partial fulfilment of the requirement for the

award of the degree of

**BACHELOR OF TECHNOLOGY (2019-2023)**

In

**COMPUTER SCIENCE AND ENGINEERING**

Submitted By

**Tarun Kumar**

**Roll No. 86200107**

**Under the supervision of**

**Mrs. Manisha Mam**

**Assistant Professor , CSE Department**

**APIIT SD INDIA, PANIPAT**

|  |  |
| --- | --- |
|  |  |
|  |  |

**KURUKSHETRA UNIVERSITY, KURUKSHETRA**

**December, 2022**

**Declaration**

I, Ankit Kumar the undersigned, hereby declare that the Summer Training Report Fire and Smoke Detection Submitted by me to the APIIT SD INDIA in partial fulfilment of the requirement for the award of degree of Bachelor of Information Technology under the guidance of Mrs. Manisha Mam, is my original work and the conclusions drawn therein are based on the material collected by myself. The Report submitted is my own work and has not been duplicated from any other source. I shall be responsible for any unpleasant moment/situation.

Place: Name: Ankit Kumar

Date: Roll No.: 8619101

**Acknowledgement**

A successful Summer Training Report is the result of teamwork and coordination that includes not only the group of developers who put forth the ideas, logic and efforts but also those who guide them. So, at the completion of the Summer Training Report, I feel obliged to extent my gratitude towards all those who made valuable contributions throughout my training period. I am thankful for all the knowledge, guidance and support imparted by Dr. / Prof. Prateek Mishra Sir to me who gave me invaluable knowledge during the IT period.

In addition, I wish to convey a deep sense of gratitude towards Prof. Anshu Sharma Mam at any time I need.

At the end just as significantly, I would like to express my sincere thanks to Prof. Manisha Mam and all the other staff members who have provided me excellent knowledge and support throughout my Graduation.

**Certificate**



|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Topic | Page No | Signature |
| 1 | Introduction | 4 |  |
| 2 | summary | 5-11 |  |
| 3 | working | 12-13 |  |
| 4 | Screenshots | 14-26 |  |
| 5 | Conclusion | 27 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 6 | References | 28 |  |

**Introduction**

Fire detection is the main objective of this project besides surveillance. The aim of the project is to early detection of fire apart from preventive measures to reduce the losses due to hazardous fire. The project mainly is based on image processing and arduino serial communication. In this project, at the user end, the fire images will be feeded in the form of video frames.

These images will be further processed by using the software, MATLAB. The proposed system uses RGB and YCbCr color space. The advantage of using YCbCr color space is that it can separate the luminance from the chrominance more effectively than RGB color space. along with this smoke, motion, area detection is also performed using its color characteristics. The proposed system consist of hardware such as arduino, DHT 11 to monitor the Humidity and Temperature. There is a camera for the surveillance. This camera will give a real-time video output to the user on the laptop or computer via a small GUI-graphic user interface which is to be built in MATLAB. Thus the fire will be detected using this model.

This project can also be served for security and surveillance applications.

Fires represent a constant threat to ecological systems, infrastructure and human lives. Past has witnessed multiple instances of fires. With the faster and faster urbanization process, more and more high-rise buildings appear around us. This also can make the frequency of fire increase and bring great losses to people’s lives and property. In areas where fire would pose an unreasonable threat to property, human life or important biological communities, efforts should be made to reduce dangers of fire. As the damage caused by fires is so tremendous that the early fire detection is becoming more and more important. Recently, some fire detectors have been used in many places, they used the smoke, temperature and photosensitive characteristics to detect fires. But they are too worse to meet the needs in a large space, harsh environment or the outdoor environment etc.

**Objective**

Fire-detection systems play a [pivotal role](https://www.sciencedirect.com/topics/engineering/pivotal-role" \o "Learn more about pivotal role from ScienceDirect's AI-generated Topic Pages) in green buildings. By detecting a fire quickly and accurately (i.e., by not sacrificing speed or causing false alarms) and providing early warning notification, a fire-detection system can limit the emission of toxic products created by combustion, as well as global-warming gases produced by the fire itself.

These environmental effects often are overlooked, but undoubtedly occur in all fire scenarios. Therefore, reducing the likelihood of a fire is an important part of designing the green building.

Automatic fire detection usually senses smoke or heat and it can be difficult to set this to avoid false alarms in the immediate welding environment. However, other areas of the workshop may be covered by such systems. Automatic systems can be set to trigger sprinklers and/or to call the fire service automatically.

The welder will not generally be aware of a fire while [arc welding](https://www.sciencedirect.com/topics/engineering/electric-arc-welding" \o "Learn more about arc welding from ScienceDirect's AI-generated Topic Pages), as it is not visible through the filter glass. While welding in an area which is purpose-built for welding, there should be no risk, since the area will be free from combustible materials. However, where welding is carried out in an area that is not purpose-built for welding, a fire watcher should be on duty. This will be one of the stipulations of the hot work permit. If a fire breaks out the watcher must interrupt the welder and take appropriate action. The watcher should be trained in the use of fire extinguishing equipment and in the means for raising the alarm.

**TOOLS USED IN PROJECT**

* Minimum Software Requirements:

This project is basically need some tools:-

1. Jupyter Notebook

* Minimum Hardware Requirements:

To run this project basic hardware requirements are:-

1. 64-bit versions of Microsoft Windows 11,10, 8
2. 4 GB RAM minimum
3. 1.5 GB hard disk space + at least 1 GB for caches.

**INTRODUCTION OF TOOLS**

**Python:-**

It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

Python is a dynamic, interpreted (bytecode-compiled) language. There are no type declarations of variables, parameters, functions, or methods in source code. This makes the code short and flexible, and you lose the compile-time type checking of the source code.

It supports multiple programming paradigms beyond object-oriented programming, such as procedural and functional programming.

**Object-Oriented Programming:-**

* Classes.
* Regular Expressions.
* Iteration and Iterators.
* Using Python's Decimal module.
* Interfacing with Unix.
* Special Methods.
* Custom Data Types.
* Exceptions Are Classes.

**Applications of Python:-**

Python is often used as a support language for software developers, for build control and management, testing, and in many other ways. SCons for build control. Buildbot and Apache Gump for automated continuous compilation and testing. Roundup or Trac for bug tracking and project management.

# Web and Internet Development

# Scientific and Numeric

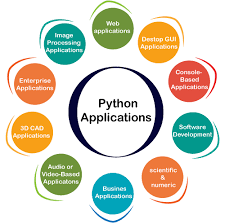
# Education

# Desktop GUIs

# Software Development

# Business Applications

* 3D CAD Applications.
* Console-based Application



**Python Features and Advantages:-**

* Easy to Code. Python is a very high-level programming language, yet it is effortless to learn. ...
* Easy to Read. Python code looks like simple English words. ...
* Free and Open-Source. ...
* Robust Standard Library. ...
* Interpreted. ...
* Portable. ...
* Object-Oriented and Procedure-Oriented. ...
* Extensible.

**Machine Learning:-**

Machine learning (ML) is a branch of artificial intelligence (AI) that enables computers to “self-learn” from training data and improve over time, without being explicitly programmed. Machine learning algorithms are able to detect patterns in data and learn from them, in order to make their own predictions. In short, machine learning algorithms and models learn through experience.

In traditional programming, a computer engineer writes a series of directions that instruct a computer how to transform input data into a desired output. Instructions are mostly based on an IF-THEN structure: when certain conditions are met, the program executes a specific action.

Machine learning, on the other hand, is an automated process that enables machines to solve problems with little or no human input, and take actions based on past observations.

While [artificial intelligence and machine learning](https://monkeylearn.com/blog/nlp-ai/" \t "https://monkeylearn.com/machine-learning/_blank) are often used interchangeably, they are two different concepts. AI is the broader concept – machines making decisions, learning new skills, and solving problems in a similar way to humans – whereas machine learning is a subset of AI that enables intelligent systems to autonomously learn new things from data.

Instead of programming machine learning algorithms to perform tasks, you can feed them examples of labeled data (known as [training data](https://monkeylearn.com/blog/training-data/" \t "https://monkeylearn.com/machine-learning/_blank)), which helps them make calculations, process data, and identify patterns automatically.

**Types of Machine Learning:-**

### Supervised Learning

[Supervised learning algorithms](https://en.wikipedia.org/wiki/Supervised_learning" \t "https://monkeylearn.com/machine-learning/_blank) and supervised learning models make predictions based on labeled training data. Each training sample includes an input and a desired output. A supervised learning algorithm analyzes this sample data and makes an inference – basically, an educated guess when determining the labels for unseen data.

This is the most common and popular approach to machine learning. It’s “supervised” because these models need to be fed manually tagged sample data to learn from. Data is labeled to tell the machine what patterns (similar words and images, data categories, etc.) it should be looking for and recognize connections with.

For example, if you want to automatically detect spam, you would need to feed a machine learning algorithm examples of emails that you want classified as spam and others that are important, and should not be considered spam.

### **Unsupervised Learning**

[Unsupervised learning algorithms](https://en.wikipedia.org/wiki/Unsupervised_learning" \t "https://monkeylearn.com/machine-learning/_blank) uncover insights and relationships in unlabeled data. In this case, models are fed input data but the desired outcomes are unknown, so they have to make inferences based on circumstantial evidence, without any guidance or training. The models are not trained with the “right answer,” so they must find patterns on their own.

One of the most common types of unsupervised learning is clustering, which consists of grouping similar data. This method is mostly used for exploratory analysis and can help you detect hidden patterns or trends.

For example, the marketing team of an e-commerce company could [use clustering to improve customer segmentation](https://bdtechtalks.com/2020/12/28/machine-learning-customer-segmentation/" \t "https://monkeylearn.com/machine-learning/_blank). Given a set of income and spending data, a machine learning model can identify groups of customers with similar behaviors.

Segmentation allows marketers to tailor strategies for each key market. They might offer promotions and discounts for low-income customers that are high spenders on the site, as a way to reward loyalty and improve retention.

### **Semi-Supervised Learning**

In [semi-supervised learning](https://en.wikipedia.org/wiki/Semi-supervised_learning" \t "https://monkeylearn.com/machine-learning/_blank), training data is split into two. A small amount of labeled data and a larger set of unlabeled data.

In this case, the model uses labeled data as an input to make inferences about the unlabeled data, providing more accurate results than regular supervised-learning models.

This approach is gaining popularity, especially for tasks involving large datasets such as image classification. Semi-supervised learning doesn’t require a large number of labeled data, so it’s faster to set up, more cost-effective than supervised learning methods, and ideal for businesses that receive huge amounts of data.

### **Reinforcement Learning**

[Reinforcement learning (RL)](https://en.wikipedia.org/wiki/Reinforcement_learning" \t "https://monkeylearn.com/machine-learning/_blank) is concerned with how a software agent (or computer program) ought to act in a situation to maximize the reward. In short, reinforced machine learning models attempt to determine the best possible path they should take in a given situation. They do this through trial and error. Since there is no training data, machines learn from their own mistakes and choose the actions that lead to the best solution or maximum reward.

**Coding Of Project**

import tensorflow as tf

import numpy as np

from tensorflow import keras

import os

import cv2

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.preprocessing import image

import matplotlib.pyplot as plt

# its important to split the training and testing - Data set is stored in the path specified.

train = ImageDataGenerator(rescale=1/255)

test = ImageDataGenerator(rescale=1/255)

train\_dataset = train.flow\_from\_directory("C:\\Users\\Ak\\Desktop\\forest\_fire\\Training and Validation\\",

target\_size=(150,150),

batch\_size = 32,

class\_mode = 'binary')

test\_dataset = test.flow\_from\_directory("C:\\Users\\Ak\\Desktop\\forest\_fire\\Testing\\",

target\_size=(150,150),

batch\_size =32,

class\_mode = 'binary')

test\_dataset.class\_indices

# We shall build the model here!

# Simple CNN shall do the task, You can try other tech as well.

# Try with other activation functions also.

model = keras.Sequential()

model.add(keras.layers.Conv2D(32,(3,3),activation='relu',input\_shape=(150,150,3)))

model.add(keras.layers.MaxPool2D(2,2))

model.add(keras.layers.Conv2D(64,(3,3),activation='relu'))

model.add(keras.layers.MaxPool2D(2,2))

model.add(keras.layers.Conv2D(128,(3,3),activation='relu'))

model.add(keras.layers.MaxPool2D(2,2))

model.add(keras.layers.Conv2D(128,(3,3),activation='relu'))

model.add(keras.layers.MaxPool2D(2,2))

model.add(keras.layers.Flatten())

model.add(keras.layers.Dense(512,activation='relu'))

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

# It is time to compile the model, let us compile.

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

# let's get the model fit.

r = model.fit(train\_dataset, epochs = 5, validation\_data = test\_dataset)

# Epochs you can vary!

# Can we work on the testing dataset, the predictions happen here.

predictions = model.predict(test\_dataset)

predictions = np.round(predictions)

Predictions

print(len(predictions))

# it's time to plot it! Lets plot loss.

plt.plot(r.history['loss'], label='loss')

plt.plot(r.history['val\_loss'], label='val\_loss')

plt.legend()

# How much accurate it is?

plt.plot(r.history['accuracy'], label='acc')

plt.plot(r.history['val\_accuracy'], label='val\_acc')

plt.legend()

# This helps in taking individual images from the Dataset, load and check results.

def predictImage(filename):

img1 = image.load\_img(filename,target\_size=(150,150))

plt.imshow(img1)

Y = image.img\_to\_array(img1)

X = np.expand\_dims(Y,axis=0)

val = model.predict(X)

print(val)

if val == 1:

plt.xlabel("No Fire",fontsize=30)

elif val == 0:

plt.xlabel("Fire",fontsize=30)

predictImage("C:\\Users\\Ak\\Desktop\\forest\_fire\\Testing\\fire\\abc172.jpg")

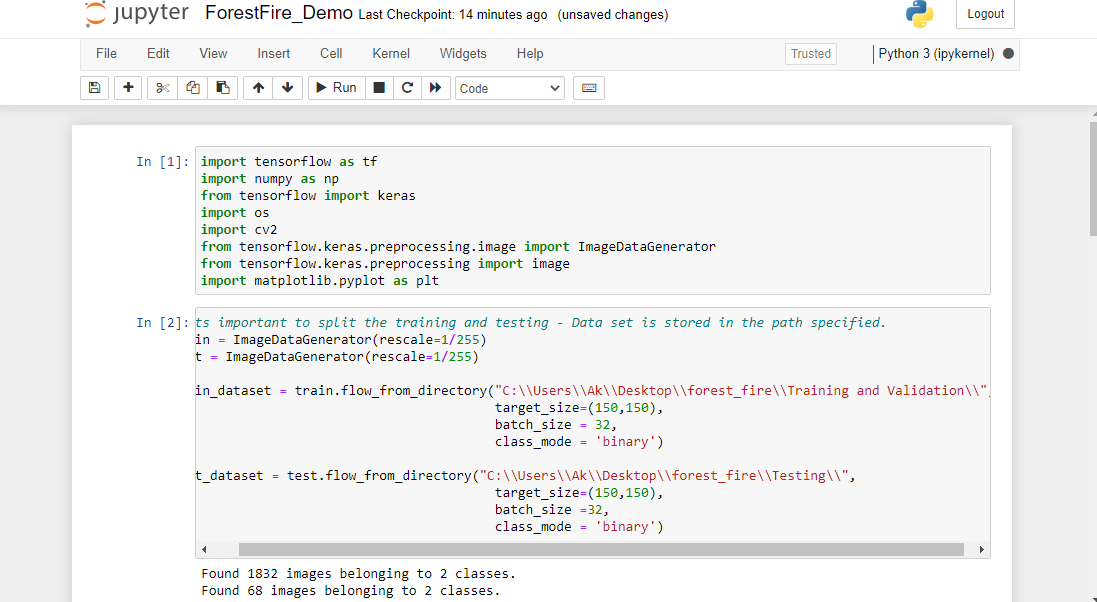
predictImage("C:\\Users\\Ak\\Desktop\\forest\_fire\\Testing\\fire\\abc178.jpg")

predictImage('C:\\Users\\Ak\\Desktop\\forest\_fire\\Testing\\nofire\\abc347.jpg')

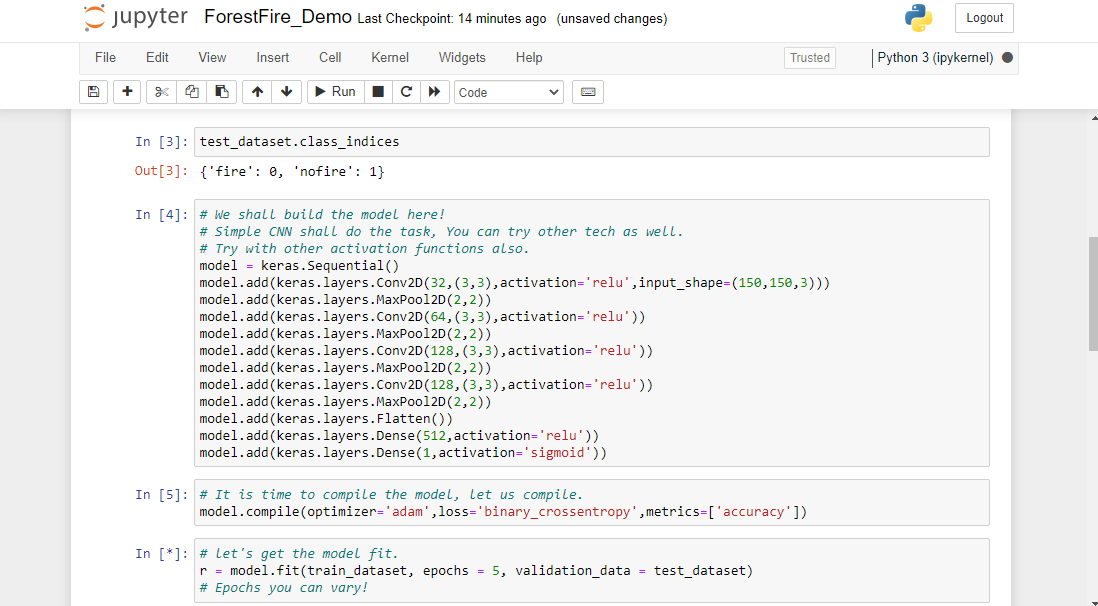
predictImage('C:\\Users\\Ak\\Desktop\\forest\_fire\\Testing\\nofire\\abc367.jpg')

**Screenshots of the Project**

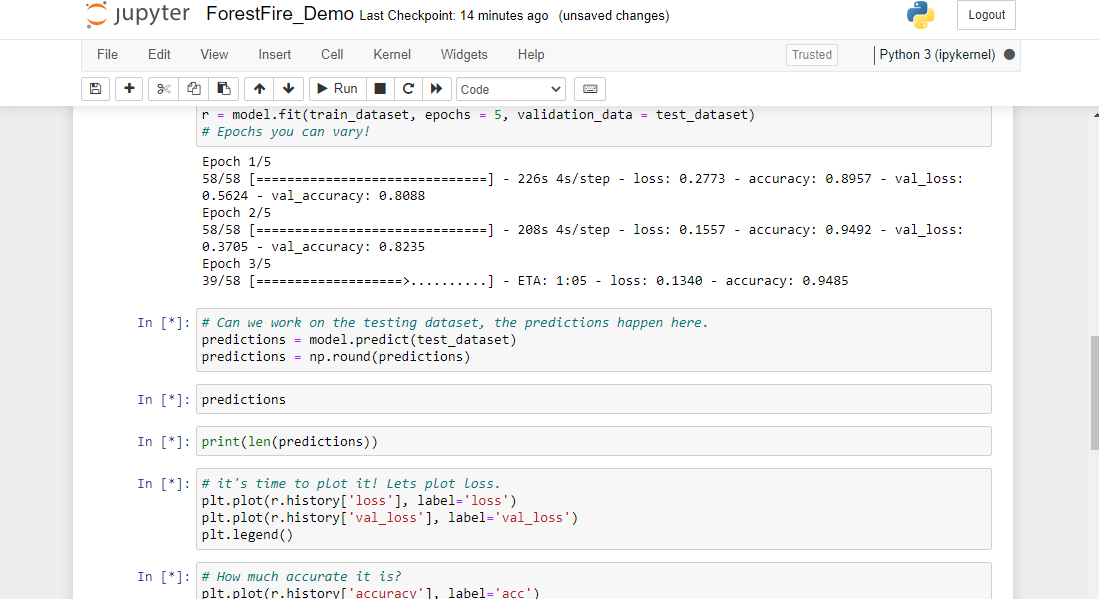
First of all we imported modules we needed.



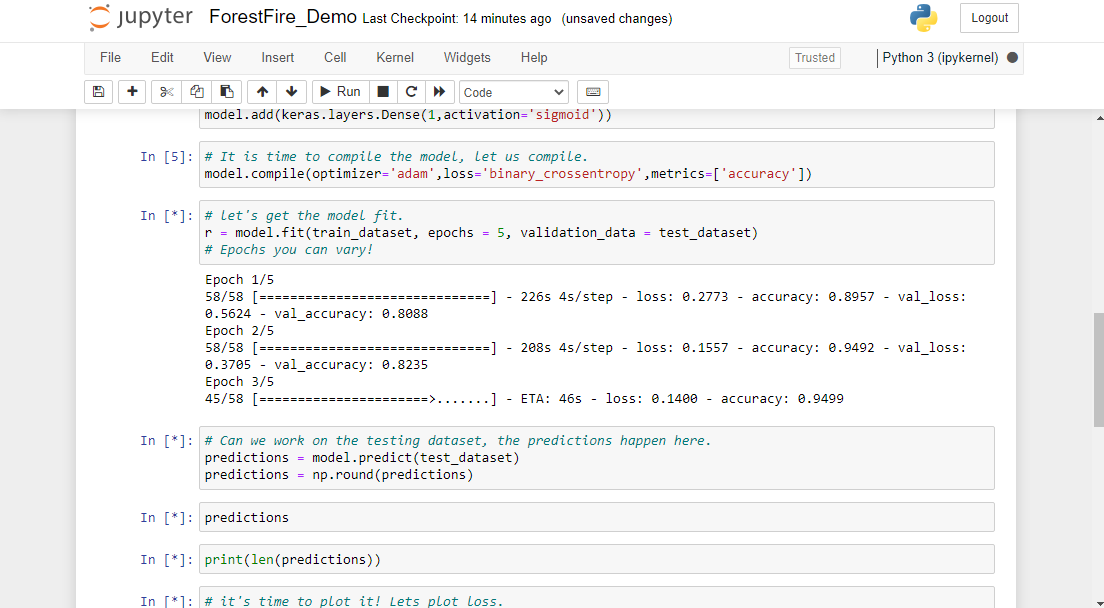
After that we have declared the train data and test data. We make the machine to learn on the basis of training data. And predict on the basis of this data. And we will draw the graph with real and predict data yo check the accuracy between predict and real data.



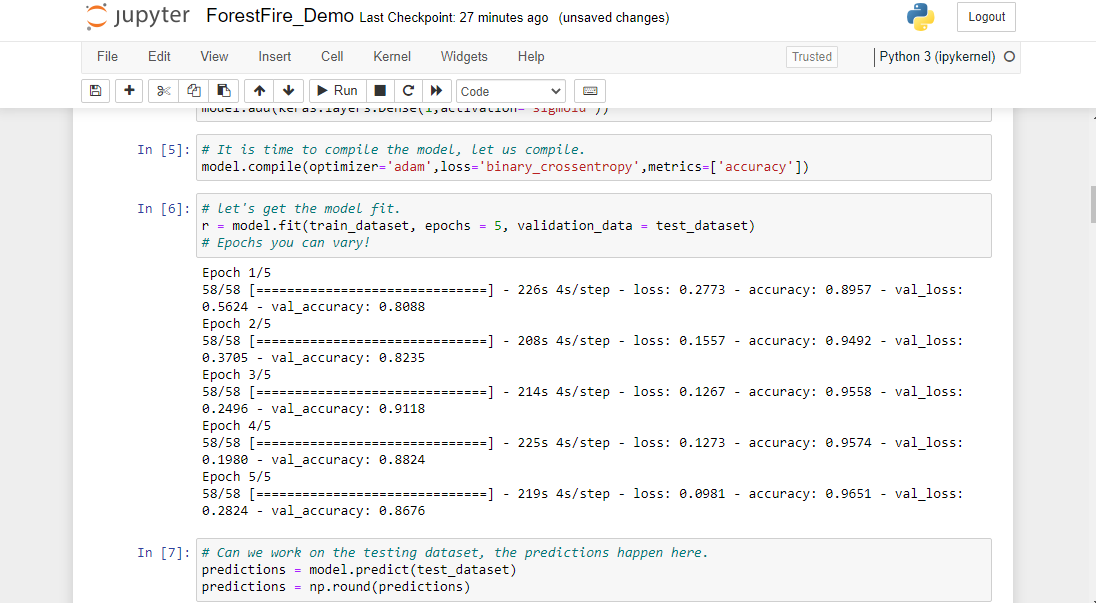
After that we defined some variables that will be used in the programming further.

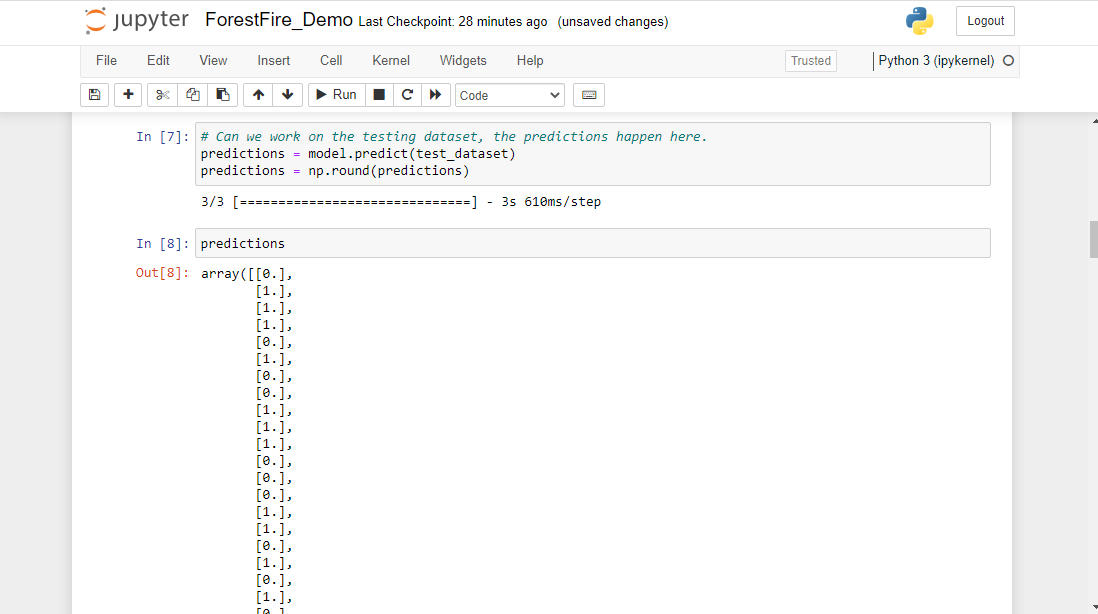


The machine learn on the basis of training data. And predict on test data. This work is done by machine automatically. Algorithms are already declared in python and machine learning.

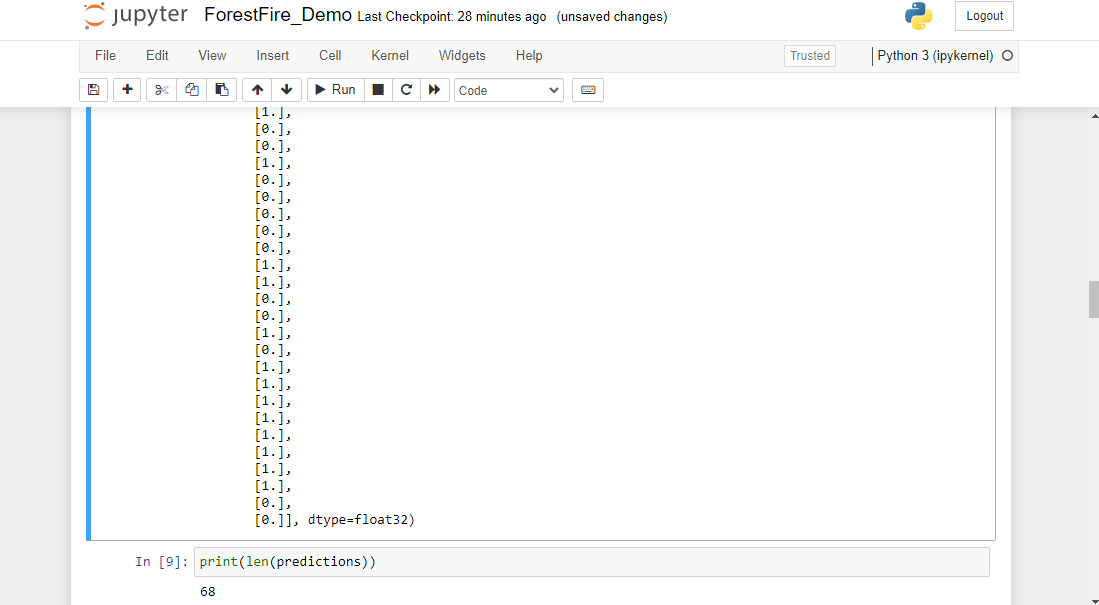


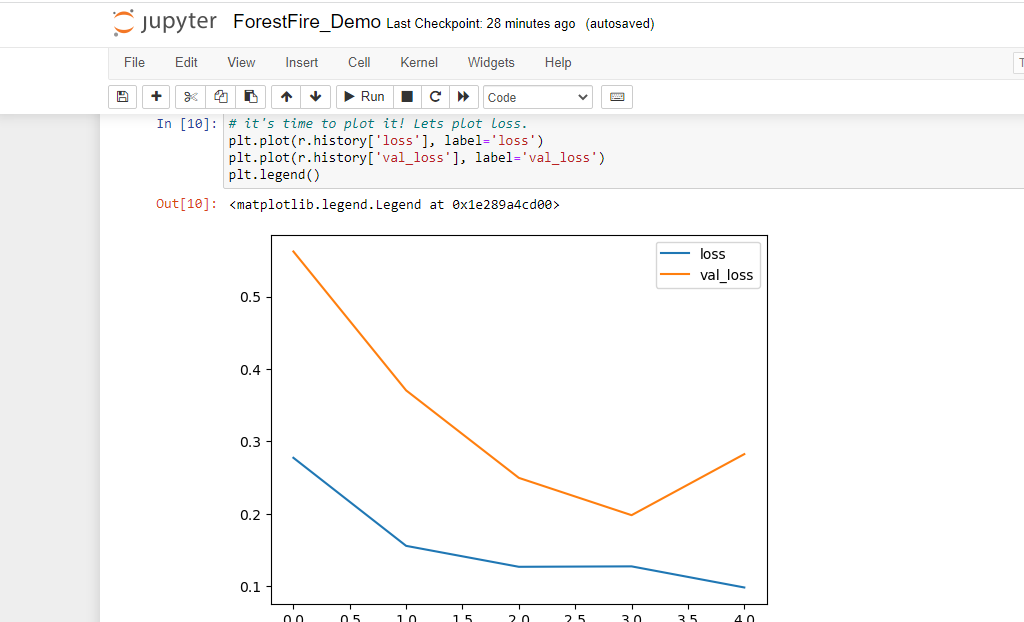
The data is processed and make prediction on the basis of test data. The processing is done on the basis of machine learning algorithms. That are already preregistered in Pyhon and Machine Learning.



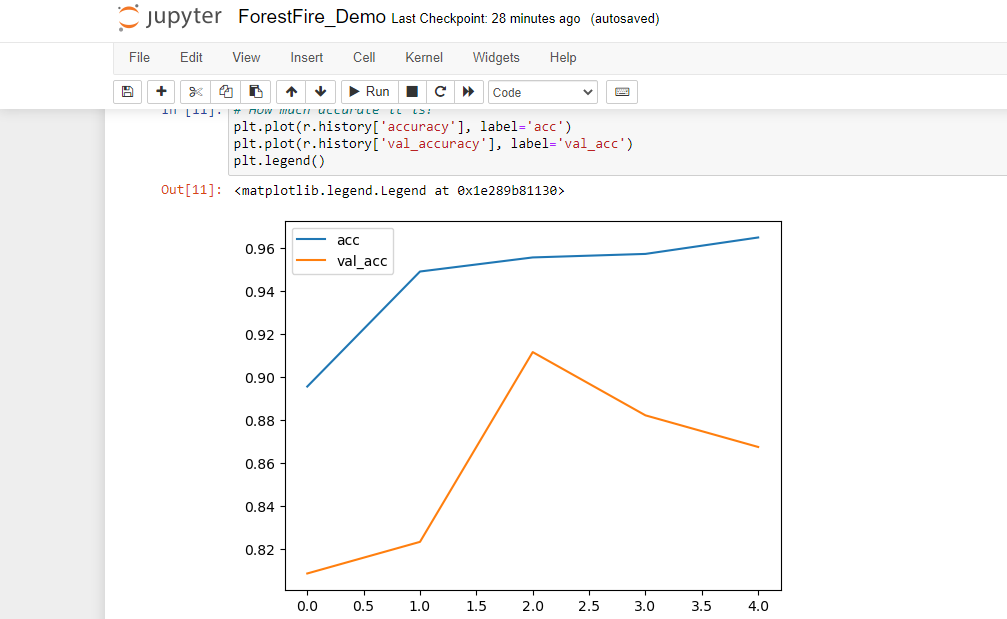


The prediction is done on the 68 data that are in the form of images or jpg format.

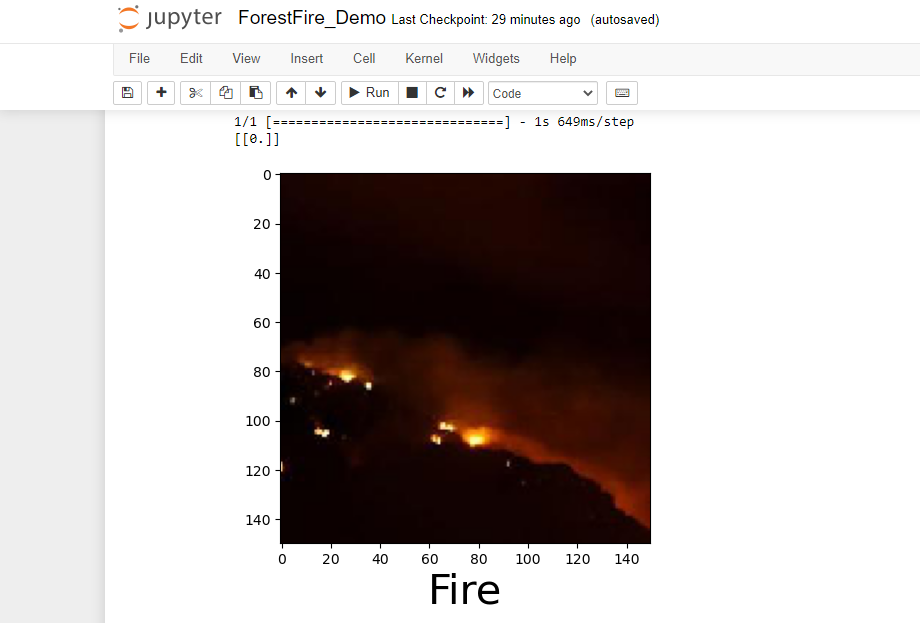




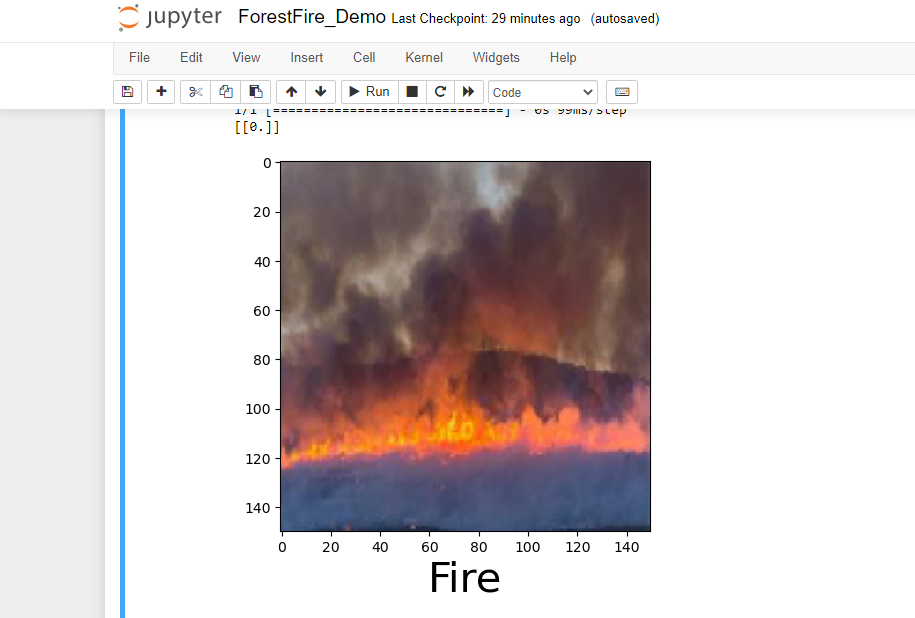
After the prediction the graph is plotted between real and predicted data to check the accuracy.



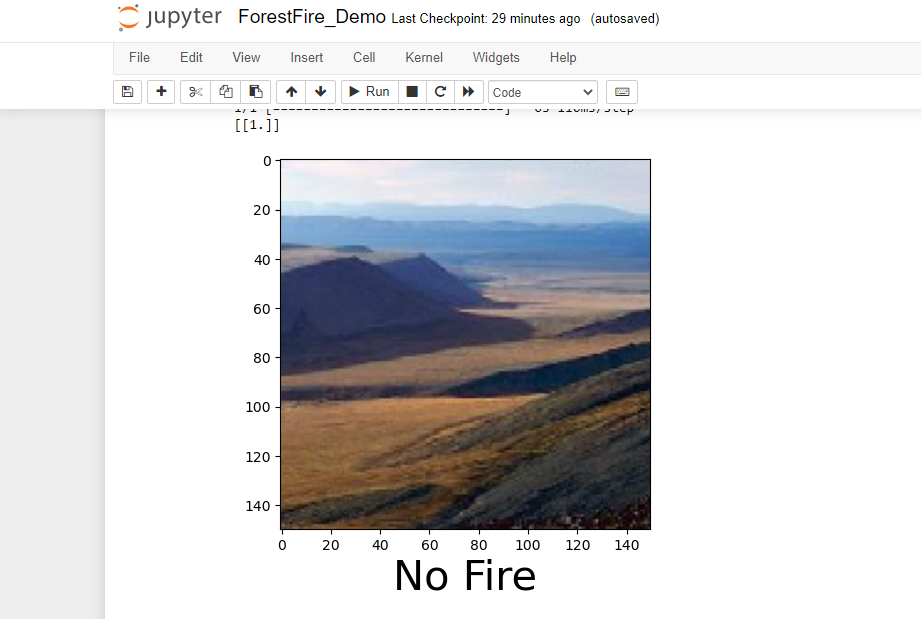
After the prediction the graph is plotted between real and predicted data to check the accuracy.



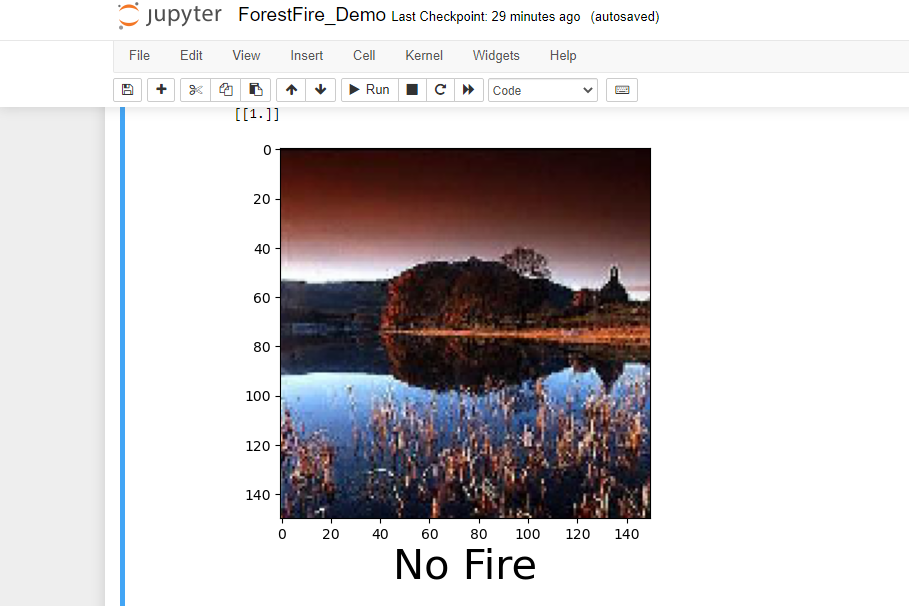
In last we pick one image and check there is fire or no fire.



In last we pick one image and check there is fire or no fire to check weather our project is working accordingly or not



In last we pick one image and check there is fire or no fire and check weather our project is working appropiate or not



**Conclusion**

This project, Fire Detection System has been developed using Image Processing and Matlab software. This system has the ability to apply image processing techniques to detect fire. This system can be used to monitor fire and has achieved 90% accuracy for single webcam. The system works on real time, as it extracts frames in every 2 seconds, it provides continuous monitoring. This system has high efficiency as it has incorporated techniques of Area detection, color detection, Motion detection, and Smoke detection as well as Humidity and Temperature detection. For better performance outcomes use of RGB, HSV and YCbCr color space is made in the detection techniques, as per their suitability, efficiency and properties. The different parameters like threshold value, blind-spots will be handled properly in our future research. Thus application of proposed fire detection system gives us a better system performance in term of less false alarm and thus a higher system performance is achieved References

**References**

* Youtube
* Github
* Google
* Sofcon India pvt. Ltd.(Training Institute)