PROJECT REPORT

EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

TEAM ID: PNT2022TMID17691

TEAM SIZE: 4

TEAM LEADER: PRIYANKA.B

TEAM MEMBER: AJAYKUMAR.M

TEAM MEMBER: DHARSHAN.K

TEAM MEMBER: SANJAY.A

1. INTRODUCTION

1.1 PROJECT OVERVIEW

The main objective of a project is to detect the fires occurring in the forest. For this we have created a convolution neural network that predicts the fire. This project is an attempt to use convolutional neural networks (CNN) to detect the presence or the start of a forest fire in an image. The model could be process in open cv and applied in real- time to low-framerate surveillance video (with fires not moving that much fast, this assumption is somewhat sound) and give alert in case of fire.

1.2 PURPOSE

Forest fire is an alarm in condition which occurs in boreal, tropical as well as temporal forests due to natural causes such as volcanic eruption, high atmospheric temperature and dryness (low humidity) and manmade causes such as cigarettes discarded with flame, campus left unattended, burning of debris equipment use and malfunctions, leftovers of ignited flammable products which not only causes harmful effects on the ecosystem but also for human beings who are living in the forest(tribes) as well as the surrounding areas. So it is necessary to prevent fires that are occurring in the forest. Which saves not only the natural resources but also has a good impact in uplifting the quality of environment.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

Aerial monitoring of forest fire using drone Cameras operated in remote locations

- ✓ Use of various sensors such as smoke, flame, gas etc…to sense and detect fire
- ✓ Human surveillance for forest
- ✓ Thermal imaging of forest
- ✓ Use of satellite images to detect fire

2.2 REFERENCES

- ✓ [1] Zhentian Jiao1, Youmin Zhang, Jing Xin, Lingxia Mu, Yingmin Yi, Han Liu and Ding Liu, ”A Deep Learning based forest fire detection approach using UAV and YOLOv3,” 1st International Conference on Industrial Artificial Intelligence (IAI), 2019.
- ✓ [2] Qingjie Zhang, Jiaolong Xu, Liang Xu and Haifeng Guo, ”Deep Convolutional Neural Networks for forest fire detection,” in Proceed ings of the 2016 International Forum on Management, Education and Information Technology Application, Atlantis Press, 2016.
- ✓ [3] Qi-xing ZHANG, Gao-hua LIN, Yong-ming ZHANG, Gao XU, Jin-jun WANG, ”Wildland forest fire smoke detection on Faster R-CNN using synthetic smoke images,” 8th International Conference on Fire Science and Fire Protection Engineering(on the Development of Performance based Fire Code), 2017.
- ✓ [4] Genovese, Angelo and Labati, Ruggero and Piuri, Vincenzo and Scotti, Fabio, ”Wildfire smoke detection using computational intelligence techniques,” IEEE International Conference on Computational Intelligence for Measurement Systems and Applications Proceedings, 2011.

- [5] C. Yuan, Z. X. Liu, and Y. M. Zhang, “UAV-based forest fire detection 5 and tracking using image processing techniques,” in 2015 International Conference on Unmanned Aircraft Systems. IEEE, 2015, pp. 639–643.
- [6] C. Yuan, Z. X. Liu, and Y. M. Zhang, “Aerial images based forest fire detection for firefighting using optical remote sensing techniques and unmanned aerial vehicles,” Journal of Intelligent & Robotic Systems, vol. 88,no. 2-4, pp. 635–654, 2017.
- [7] X. Z. Chunyu Yu, Zhibin Mei, “A real-time video fire flame and smoke detection algorithm,” in Asia-Oceania Symposium on Fire Science and Technology, 2013.
- [8] YongMin Liu, YiMing Liu, HongLei Xu, Kok Lay Teo, ”Forest fire monitoring, detection and decision making systems by wireless sensor network,” IEEE Chinese Control And Decision Conference (CCDC), 2018.
- [9] Pulkit Chugh, Eric Tom Mathews, G. Barath Kumar, ”Forest fire detection through UAV imagery using CNNs,” unpublished.
- [10]A. Koubaa and B. Qureshi, “Dronetrack: Cloud-based real-time object then testing our model by providing the data that is allocated for testing. We can use our saved model file for video analysis of forest fire and use it in real time surveillance camera for real time prediction.

2.3 PROBLEM STATEMENT DEFINITION

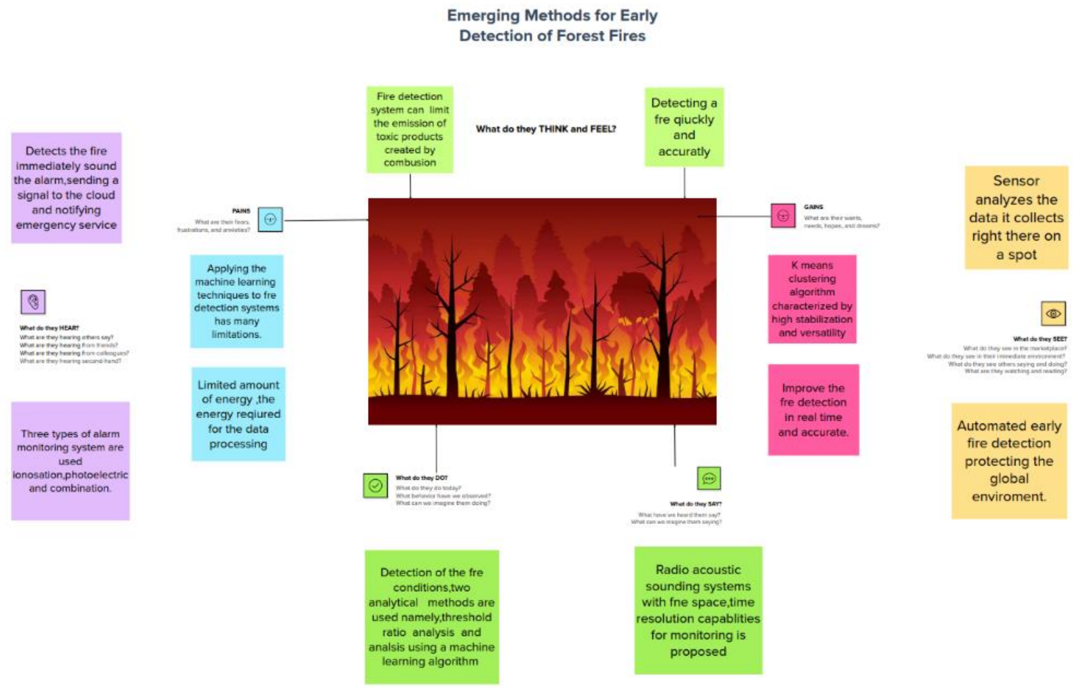
Forest fires result in a wide range of negative effects, including the destruction of
wildlife habitat, the extinction of plants and animals, the destruction of nutrien t-rich top
soil, the reduction of forest cover, the loss of valuable timber resources, the oz one layer
being destroyed, the loss of livelihood for tribal and poor people, the accelerati on of
global warming, the increase in atmospheric carbon dioxide concentration, the degradation of catchment areas, the loss of biodiversity, the spread of disease, etc.

Thus, Develop a system to detect forest fires at the earliest stage as possible us ing the latest technologies.

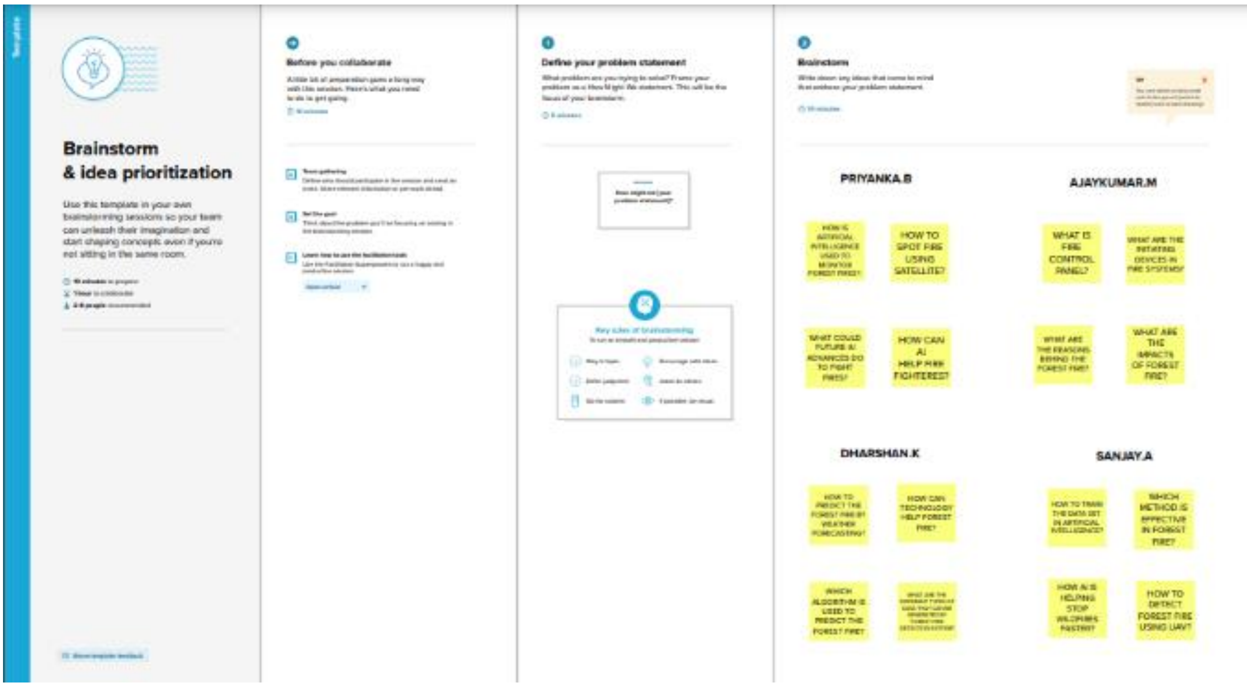
Problem Statement (PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	Local people	Detect the Forest Fire	Unable to detect	Some fires are originating very deep inside the forest	Frightened
PS-2	Fire Authority or Forest management	Detect Forest fires earlier to prevent disaster	Unable to detect earlier	Lack of high level Detection system with no alarming feature	Distraught

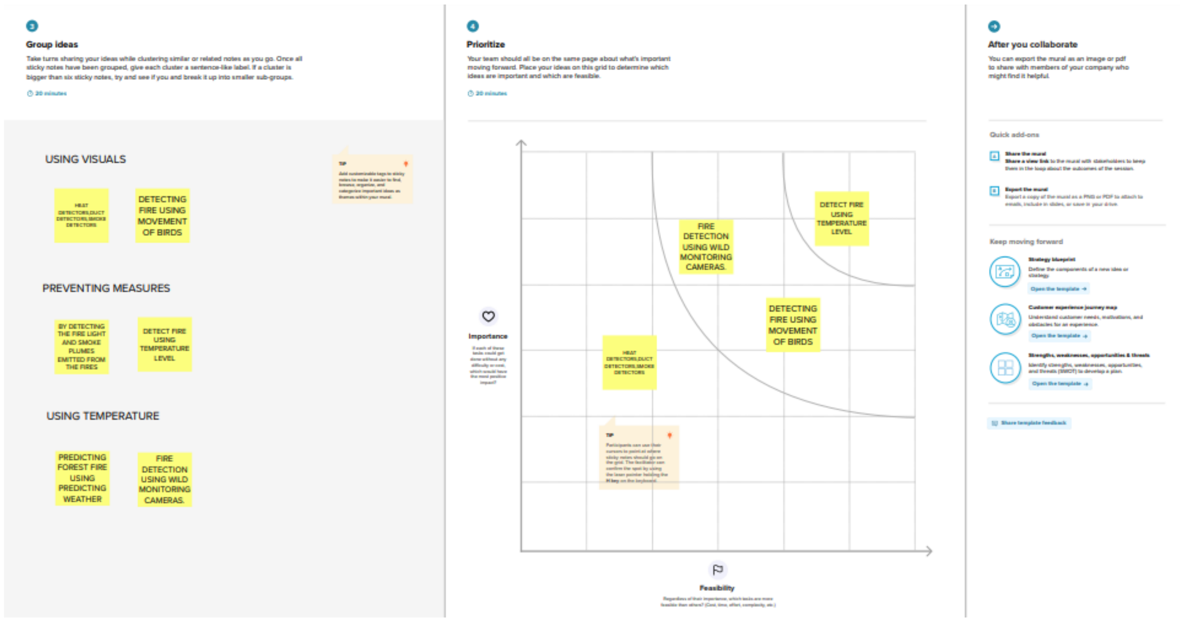
3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTROMING

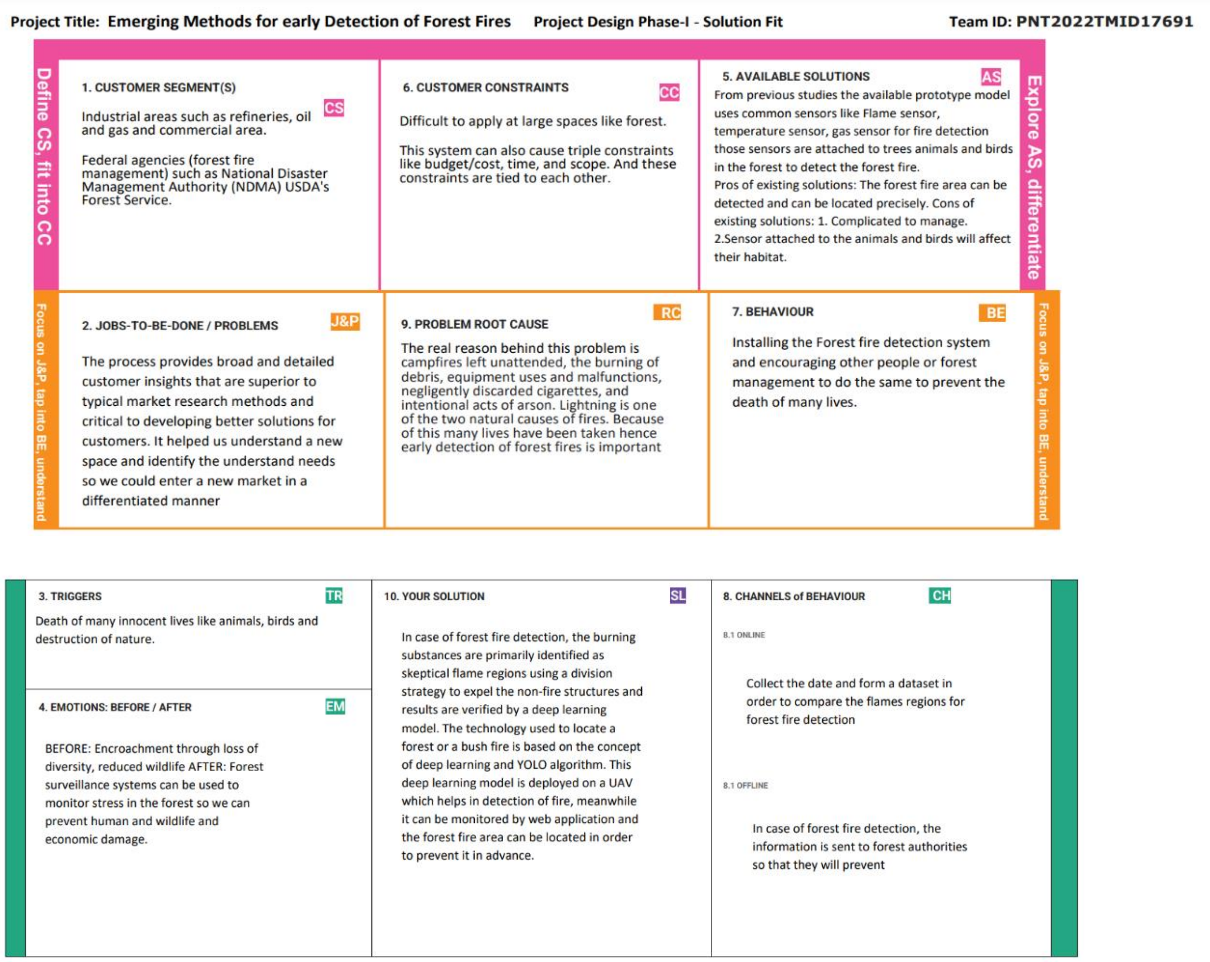




3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Detecting the forest fire at the earliest possible time to reduce its impact in our environment in all possible ways
2.	Idea / Solution description	We implement pre-processing steps to eliminate the noises in images. And also implements feature extraction to extract the colour features and segment of the fire regions.
3.	Novelty / Uniqueness	We classify the pixels using CNN algorithm with an efficient mobile alert system that sends messages to corresponding authorities.
4.	Social Impact / Customer Satisfaction	Saves the environment and its related resources from great loss. Prevent damaging of flora, fauna and some of the important endangered species. Large amounts of CO_2 emissions are avoided.
5.	Business Model (Revenue Model)	Forest ranger's lives will be saved. Prevents global warming. Prevents damaging of the electrical wires and optical fibres which may cost the government. Reports to the forest department and nearest police station quickly so as to take faster actions
6.	Scalability of the Solution	Automated analysis of fire detection. Improved accuracy rate. Reduced time for computational complexity.

3.4 Problem Solution fit



Project Title: Emerging Methods for early Detection of Forest Fires   Project Design Phase-I - Solution Fit   Team ID: PNT2022TMID17691

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Industrial areas such as refineries, oil and gas and commercial area.

Federal agencies (forest fire management) such as National Disaster Management Authority (NDMA) USDA's Forest Service.

6. CUSTOMER CONSTRAINTS CC

Difficult to apply at large spaces like forest.

This system can also cause triple constraints like budget/cost, time, and scope. And these constraints are tied to each other.

5. AVAILABLE SOLUTIONS AS

From previous studies the available prototype model uses common sensors like Flame sensor, temperature sensor, gas sensor for fire detection those sensors are attached to trees animals and birds in the forest to detect the forest fire.
Pros of existing solutions: The forest fire area can be detected and can be located precisely. Cons of existing solutions: 1. Complicated to manage.
2.Sensor attached to the animals and birds will affect their habitat.

Explore AS, differentiate

Focus on J&P, tap into BE, understand

2. JOBS-TO-BE-DONE / PROBLEMS J&P

The process provides broad and detailed customer insights that are superior to typical market research methods and critical to developing better solutions for customers. It helped us understand a new space and identify the understand needs so we could enter a new market in a differentiated manner

9. PROBLEM ROOT CAUSE RC

The real reason behind this problem is campfires left unattended, the burning of debris, equipment uses and malfunctions, negligently discarded cigarettes, and intentional acts of arson. Lightning is one of the two natural causes of fires. Because of this many lives have been taken hence early detection of forest fires is important

7. BEHAVIOUR BE

Installing the Forest fire detection system and encouraging other people or forest management to do the same to prevent the death of many lives.

Focus on J&P, tap into BE, understand

3. TRIGGERS TR

Death of many innocent lives like animals, birds and destruction of nature.

4. EMOTIONS: BEFORE / AFTER EM

BEFORE: Encroachment through loss of diversity, reduced wildlife AFTER: Forest surveillance systems can be used to monitor stress in the forest so we can prevent human and wildlife and economic damage.

10. YOUR SOLUTION SL

In case of forest fire detection, the burning substances are primarily identified as skeptical flame regions using a division strategy to expel the non-fire structures and results are verified by a deep learning model. The technology used to locate a forest or a bush fire is based on the concept of deep learning and YOLO algorithm. This deep learning model is deployed on a UAV which helps in detection of fire, meanwhile it can be monitored by web application and the forest fire area can be located in order to prevent it in advance.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

Collect the date and form a dataset in order to compare the flames regions for forest fire detection

8.1 OFFLINE

In case of forest fire detection, the information is sent to forest authorities so that they will prevent

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through wildfire portal.
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Data Prediction	Scientists create computer models to predict wildfire potential under a range of potential climate futures. Using different projections of temperature and precipitation, scientists predict where and when wildfires are most likely to occur.
FR-4	Using Sensors	This Bosch environment sensors installed in the forest fire detection system using artificial intelligence deployed as early wildfire warming tool.

4.2 Non-Functional requirements

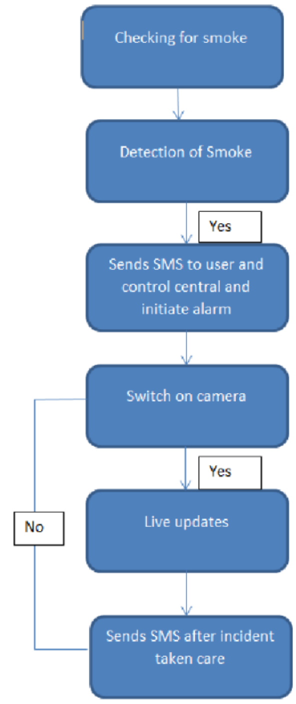
Non-functional Requirements:

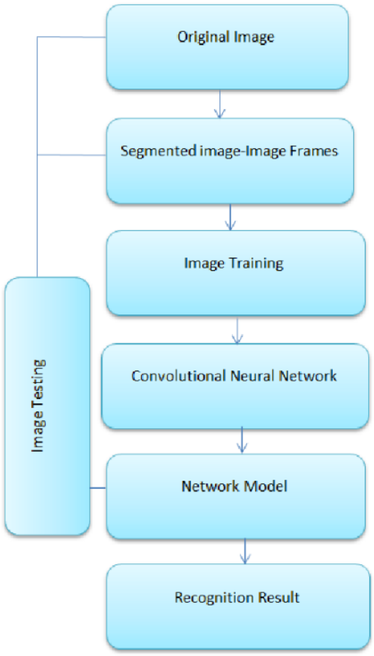
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Many methods have been proposed to detect forest fires, such as camera-based systems, WSN-based systems, and machine learning application-based systems, with both positive and negative aspects and performance figures of detection.
NFR-2	**Security**	More secure environment.
NFR-3	**Reliability**	It has achieved 1.24 seconds of classification time with an accuracy of 91% and F1 score of 0.91.
NFR-4	**Performance**	In the event of a fire, the primary objective of using drones is to gather situational awareness, which can be used to direct the efforts of the firefighters in locating and controlling hot spots. Just like urban fires, forest fires to require monitoring so that firefighters know what they are dealing with.
NFR-5	**Availability**	Forest fires (wildfires) are common hazards in forests, particularly in remote or unmanaged areas. It is possible to detect forest fires, elevated CO2, and temperature levels using AI
NFR-6	**Scalability**	A widely used measure of fire intensity is Fireline intensity, which is the rate of heat transfer per unit length of the fire line (measured in kW m−1) and represents the radiant energy release in the flaming front.

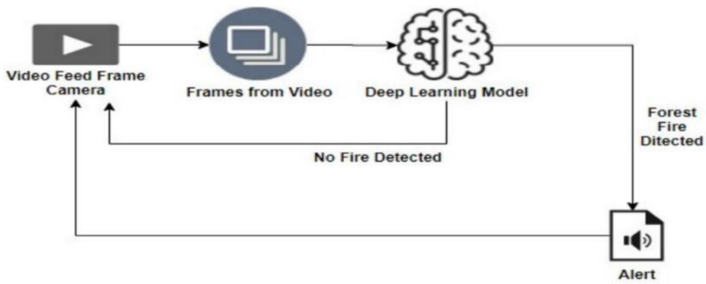
5. PROJECT DESIGN

5.1 Data Flow Diagrams





5.2 Solution & Technical Architecture



5.3 User Stories

User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Forest Authority	Collect the data	USN 1	As an Environmentalist, it is necessary to collect the data of forest which includes data like temperature, humidity, wind and rain	I can analyse the data collected	High	Sprint-1
		USN-2	Identify algorithms that can be used for prediction	collect the algorithm to identify the accuracy level of each algorithms	Medium	Sprint-2
	Identifying the algorithm	USN-3	Identify the accuracy of each algorithms	Accuracy of each algorithm calculated so that it is easy to obtain the most accurate algorithm	High	Sprint-2
	Evaluate Accuracy of Algorithm	USN-4	Identify accuracy, precision, recall of each algorithms	These values are important for obtaining the right output	High	Sprint-3

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

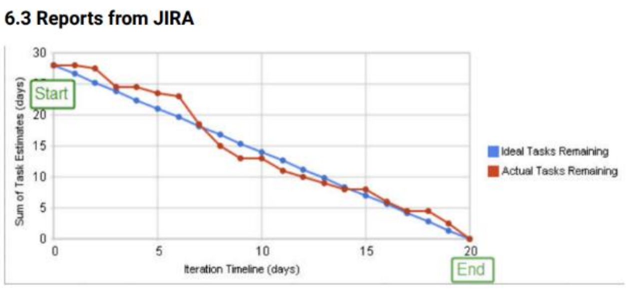
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Download data set	USN-1	The data is downloaded from the Kaggle website and then the data set is classified into training and testing images.	10	High	Priyanka.B
Sprint-1	Image pre-processing	USN-1	In Image processing technique the first step is usually importing the libraries that will be needed in the program. Import Keras library from that library and import the ImageDataGenerator Library to your Python script. The next step is definig the arguments for the ImageDataGenerator . Here the arguments which we are given inside the image data generator class are, rescale, shear_range, rotation range of image, and zoom range that we can consider for images. The next step is applying the ImageDataGenerator arguments to the train and test dataset.	10	High	Priyanka.B Ajaykumar.M Dharshan.K Sanjay.A
Sprint-2	Training image	USN-2	In this training phase the ImageDataGenerator arguments is applied to the training images and the model is tested with several images and the model is saved.	20	High	Priyanka.B Ajaykumar.M Dharshan.K Sanjay.A
Sprint-3	Testing image	USN-3	In this testing phase the Image processing techniques is applied to the testing images and executed for prediction.	20	High	Priyanka.B Ajaykumar.M Dharshan.K Sanjay.A
Sprint-4	Evaluation metrics and accuracy	USN-4	In this phase the result, prediction, accuracy, and performance of the project are tested.	20	High	Priyanka.B Ajaykumar.M Dharshan.K Sanjay.A

6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 Reports from JIRA



7. CODING & SOLUTIONING

7.1 FEATURE 1

1. Preprocessing the dataset which consists of two classes of data(fire,no fire).

Image Preprocessing

#1.Importing the ImageDataGenerator Library

```
import numpy as np

import keras

from sklearn.model_selection import train_test_split

from keras.models import Sequential, load_model

from keras.preprocessing.image import ImageDataGenerator

from keras.callbacks import ModelCheckpoint, EarlyStopping, TensorBoard

from keras.callbacks import ReduceLROnPlateau

from keras.layers import Conv2D, Dropout, Dense, Flatten, MaxPooling2D,
SeparableConv2D, Activation, BatchNormalization

import matplotlib.pyplot as plt

import time

import os

import tensorflow as tf
```

#2.Define parameters for ImageDataGenerator Class

```
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)
```

#3.Applying ImageDataGenerator Functionality to Trainset and Testset

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)

x_dataset =train_datagen.flow_from_directory(r"/content/drive/MyDrive/Forest-Dataset/forest_fire",target_size = (128,128), class_mode = "binary",batch_size = 32)

x_train =train_datagen.flow_from_directory(r"/content/drive/MyDrive/Forest-Dataset/forest_fire/Training and Validation",target_size = (128,128), class_mode = "binary",batch_size = 32)

x_test =test_datagen.flow_from_directory(r"/content/drive/MyDrive/Forest-Dataset/forest_fire/Testing",target_size = (128,128), class_mode = "binary",batch_size = 32)

x_train.class_indices

a. For Dataset

```
In [136]: x_dataset =train_datagen.flow_from_directory(r"/content/drive/MyDrive/Forest-Dataset/forest_fire",target_size = (128,128), class_
Found 1900 images belonging to 2 classes.
```

b. For Trainset

```
In [137]: x_train =train_datagen.flow_from_directory(r"/content/drive/MyDrive/Forest-Dataset/forest_fire/Training and Validation",target_si
Found 1832 images belonging to 2 classes.
```

c. For Testset

```
In [138]: x_test =test_datagen.flow_from_directory(r"/content/drive/MyDrive/Forest-Dataset/forest_fire/Testing",target_size = (128,128), cl
Found 68 images belonging to 2 classes.

In [139]: x_train.class_indices
Out[139]: {'fire': 0, 'nofire': 1}
```

2.Building up a sequential model to train the dataset.

1.Importing the Model Building Libraries

#Importing model libraries

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from tensorflow.keras.layers import Convolution2D

from tensorflow.keras.layers import MaxPooling2D

from tensorflow.keras.layers import Flatten

import warnings

warnings.filterwarnings('ignore')

2. Initializing the model

model=Sequential()

model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

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

model.add(Flatten())

Model Summary

model.summary()

```
In [145]: #Model Summary
          model.summary()

          Model: "sequential_1"
          _________________________________________________________________
           Layer (type)                Output Shape              Param #
          =================================================================
           conv2d_1 (Conv2D)           (None, 126, 126, 32)      896

           max_pooling2d_1 (MaxPooling  (None, 63, 63, 32)       0
           2D)

           flatten_1 (Flatten)         (None, 127008)            0

          =================================================================
          Total params: 896
          Trainable params: 896
          Non-trainable params: 0
          _________________________________________________________________
```

3. Prediction of data.

pred = model.predict(x)

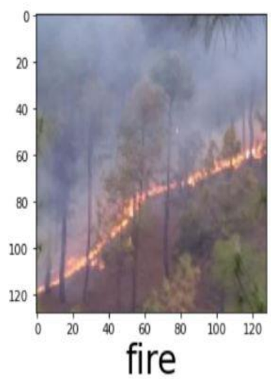
pred = np.round(pred)

pred

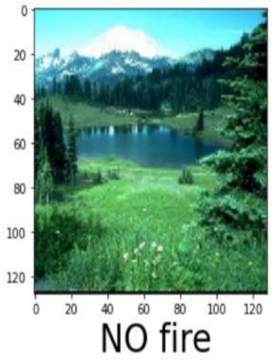
def predictImage(filename):

```
 img1=image.load_img(filename,target_size=(128,128))
 plt.imshow(img1)
 y=image.img_to_array(img1)
 x=np.expand_dims(y,axis=0)
 val=model.predict(x)
 print(val)
 if val==0:
   plt.xlabel(" No Fire",fontsize=30)
 elif val==1:
   plt.xlabel("Fire",fontsize=30)
predictImage("/content/drive/MyDrive/Forest-Dataset/forest_fire/Testing/fire/abc173.jpg")
plt.xlabel("fire",fontsize=30)
predictImage('/content/drive/MyDrive/Forest-Dataset/forest_fire/Testing/nofire/abc377.jpg')
plt.xlabel(" NO fire",fontsize=30)
```

```
In [175]: predictImage("/content/drive/MyDrive/Forest-Dataset/forest_fire/Testing/fire/abc173.jpg")
          plt.xlabel("fire",fontsize=30)

          1/1 [==============================] - 0s 46ms/step
          [[1.]]

Out[175]: Text(0.5, 0, 'fire')
```



```
In [176]: predictImage('/content/drive/MyDrive/Forest-Dataset/forest_fire/Testing/nofire/abc377.jpg')
          plt.xlabel(" NO fire",fontsize=30)

          1/1 [==============================] - 0s 36ms/step
          [[1.]]

Out[176]: Text(0.5, 0, ' NO fire')
```



7.2 Feature 2

1.Creation of twilio account

To send an outgoing SMS message from your Twilio account you’ll need to mak e an HTTP POST to Twilio's Message resource.

Twilio's Python library helps you to create a new instance of the Message reso urce, specifying the To, From, and Body parameters of your message.

Replace the placeholder values for account_sid and auth_token with your uniq ue values. You can find these in your Twilio console.

You’ll tell Twilio which phone number to use to send this message by replacing the from_ number with the Twilio phone number you purchased earlier.

Next, specify yourself as the message recipient by replacing the to number wit h your
mobile phone number. Both of these parameters must use E.164 formatting (+ and a country code, e.g., +16175551212) We also include the body parameter, which contains the content of the SMS we’re going to send.

8. TESTING

8.1 Test Cases

Panel switches and keypads : TEST the operation of each control.

Visual indicators : TEST the operation of each visual indicator and alphanumeric displays.

Battery: MEASURE system quiescent and maximum alarm currents in accordance with
Appendix . Calculate the required battery capacity and CHECK the nominal capacity of the installed batteries is not less than the calculated capacity.
Verify that the measured currents are the same as recorded in the baseline data.

8.2 User Acceptance Testing

1. Purpose of Document The purpose of this document is to briefly explain the test coverage and open issues of the project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis
This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	9	5	1	2	17
Duplicate	1	0	2	0	3
External	3	3	0	1	7
Fixed	10	2	3	20	35
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	4	2	1	7
Totals	13	15	10	25	72

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	53	0	0	53
Security	2	0	0	2
Outsource Shipping	4	0	0	4
Exception Reporting	7	0	0	7
Final Report Output	3	0	0	3
Version Control	1	0	0	1

9. RESULTS

9.1 Performance Metrics

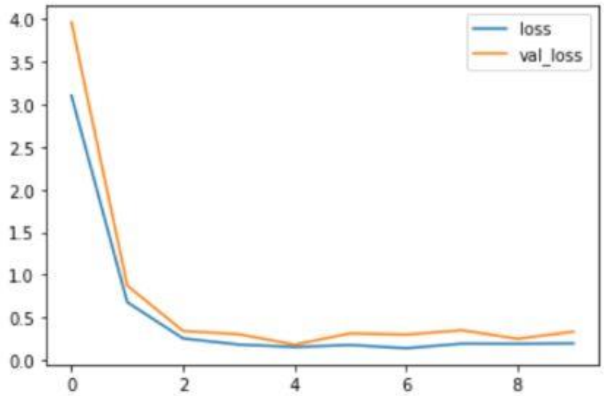
TRAINING

6.Training the Model

```
In [149…
#fit or train the model
r=model.fit_generator(x_train,steps_per_epoch=14,
                    epochs=10,validation_data=x_test,
                    validation_steps=2)

Epoch 1/10
14/14 [==============================] - 10s 676ms/step - loss: 3.1005 - accuracy: 0.7406 - val_loss: 3.9601 - val_accuracy: 0.5156
Epoch 2/10
14/14 [==============================] - 9s 662ms/step - loss: 0.6805 - accuracy: 0.9040 - val_loss: 0.8754 - val_accuracy: 0.8750
Epoch 3/10
14/14 [==============================] - 9s 629ms/step - loss: 0.2560 - accuracy: 0.8973 - val_loss: 0.3434 - val_accuracy: 0.8750
Epoch 4/10
14/14 [==============================] - 9s 656ms/step - loss: 0.1841 - accuracy: 0.9196 - val_loss: 0.3057 - val_accuracy: 0.8750
Epoch 5/10
14/14 [==============================] - 10s 663ms/step - loss: 0.1558 - accuracy: 0.9353 - val_loss: 0.1833 - val_accuracy: 0.9062
Epoch 6/10
14/14 [==============================] - 11s 803ms/step - loss: 0.1790 - accuracy: 0.9397 - val_loss: 0.3160 - val_accuracy: 0.9062
Epoch 7/10
14/14 [==============================] - 10s 665ms/step - loss: 0.1416 - accuracy: 0.9509 - val_loss: 0.3013 - val_accuracy: 0.9062
Epoch 8/10
14/14 [==============================] - 9s 631ms/step - loss: 0.1950 - accuracy: 0.9222 - val_loss: 0.3519 - val_accuracy: 0.8906
Epoch 9/10
14/14 [==============================] - 9s 619ms/step - loss: 0.1927 - accuracy: 0.9330 - val_loss: 0.2520 - val_accuracy: 0.9062
Epoch 10/10
14/14 [==============================] - 9s 638ms/step - loss: 0.1976 - accuracy: 0.9340 - val_loss: 0.3360 - val_accuracy: 0.8906
```

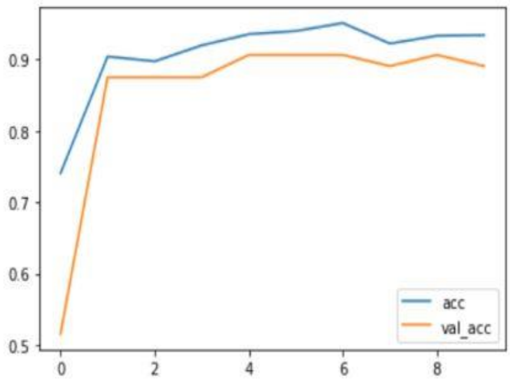
LOSS OR NO LOSS

```
In [150]: import matplotlib.pyplot as plt
          plt.plot(r.history['loss'],label='loss')
          plt.plot(r.history['val_loss'],label='val_loss')
          plt.legend()

Out[150]: <matplotlib.legend.Legend at 0x7f461a7fda10>
```



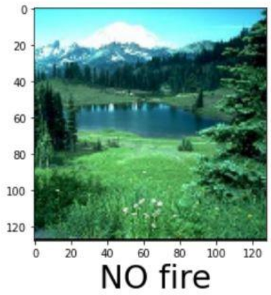
ACCURACY VALUE

```
In [151]: plt.plot(r.history['accuracy'],label='acc')
          plt.plot(r.history['val_accuracy'],label='val_acc')
          plt.legend()

Out[151]: <matplotlib.legend.Legend at 0x7f461a85d890>
```

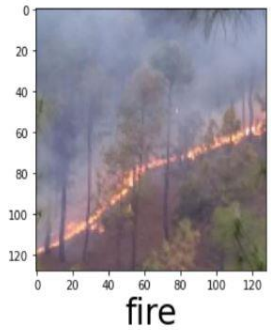


PREDICTIONS

```
In [176]: predictImage('/content/drive/MyDrive/Forest-Dataset/forest_fire/Testing/nofire/abc377.jpg')
          plt.xlabel(" NO fire",fontsize=30)

          1/1 [==============================] - 0s 36ms/step
          [[1.]]

Out[176]: Text(0.5, 0, ' NO fire')
```




```
In [175]: predictImage("/content/drive/MyDrive/Forest-Dataset/forest_fire/Testing/fire/abc173.jpg")
          plt.xlabel("fire",fontsize=30)

          1/1 [==============================] - 0s 46ms/step
          [[1.]]

Out[175]: Text(0.5, 0, 'fire')
```



10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- ✓ The proposed system detects the forest fire at a faster rate compared to existing system. It has enhanced data collection feature.
- ✓ The major aspect is that it reduces false alarm and also has accuracy due to various sensors present.
- ✓ It minimizes the human effort as it works automatically.
- ✓ This is very affordable due to which can be easily accessed.
- ✓ The main objective of our project is to receive an alert message through an app to the respective user.

- ✓ The arrangement is fire-
  proof and can withstand high temperatures, rugged, reliable, cost-effective, and easy to install.
- ✓ It is also easy to decode the data from satellite at the ground station and no
  experts are required to understand or decode the data from the satellite .
- ✓ All the components like temperature sensor and the GPS are easy to interface.
- ✓ The approximate value of temperature and the GPS co-ordinates are obtained.
  Since we are using wireless sensing networks, the attenuation during the transmission of the signal or the data is minimised.
- ✓ It is More Reliable

DISADVANTAGES:

- ✓ The electrical interference diminishes the effectiveness of radio receiver.
- ✓ The main drawback is that it has less coverage range areas.
- ✓ Even a small fault would cause the whole system to fail.

11. CONCLUSION

The proposed system for forest fire detection using wireless sensor networks and machine learning was found to be an effective method for fire detection in forests that provides more accurate results. Here, to obtain a more accurate outcome within the lowest latency, the analysis should take place continuously and camera monitoring should be effectively done. This system is well developed to fit any weather condition, climatic condition, or area, . In the case of node deployment, cameras can be mounted at any place in the forest even with good connectivity and built-in network infrastructure. IR frame sensors are used to enhance efficiency of the system. A unique feature that sends alert messages to the concerned authorities when fire is detected is also added. Thus, By detecting the forest fire we can reduce air pollution, landslides,soil erosion by protecting strong rooted trees,and the emission of CO_2 into the air during fire causing no loss of life and resources.

12. FUTURE SCOPE

- ✓ Right now we have designed the project for control of two devices but it can be designed for more numbers of devices.
- ✓ It can be further expanded with a voice interactive system facility.
  A feedback system can also be included which provides the state of devi ce to the remote users.

13. APPENDIX

Source Code

#Download the Dataset

pwd

#Load the Image Dataset

from google.colab import drive

drive.mount('/content/drive')

call load_data with allow_pickle implicitly set to true

import numpy as np

data = np.load('/content/drive/My Drive/Forest-Dataset/Dataset.zip', allow_pickle=True)

print('data loaded')

cd //content/drive/MyDrive/Forest-Dataset

#Unzip the Dataset

!unzip Dataset.zip

#Image Preprocessing

#1.Importing the ImageDataGenerator Library

import numpy as np

import keras

from sklearn.model_selection import train_test_split

from keras.models import Sequential, load_model

from keras.preprocessing.image import ImageDataGenerator

from keras.callbacks import ModelCheckpoint, EarlyStopping, TensorBoard

from keras.callbacks import ReduceLROnPlateau

from keras.layers import Conv2D, Dropout, Dense, Flatten, MaxPooling2D, SeparableConv2D, Activation, BatchNormalization

import matplotlib.pyplot as plt

import time

import os

import tensorflow as tf

#2.Define parameters for ImageDataGenerator Class

```
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)
```

#3.Applying ImageDataGenerator Functionality to Trainset and Testset

x_dataset =train_datagen.flow_from_directory(r"/content/drive/MyDrive/Forest-Dataset/forest_fire",target_size = (128,128), class_mode = "binary",batch_size = 32)

x_train =train_datagen.flow_from_directory(r"/content/drive/MyDrive/Forest-Dataset/forest_fire/Training and Validation",target_size = (128,128), class_mode = "binary",batch_size = 32) x_test =test_datagen.flow_from_directory(r"/content/drive/MyDrive/Forest-Dataset/forest_fire/Testing",target_size = (128,128), class_mode = "binary",batch_size = 32)

x_train.class_indices

 # Model Building

 # 1.Importing the Model Building Libraries

#Importing model libraries

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from tensorflow.keras.layers import Convolution2D

from tensorflow.keras.layers import MaxPooling2D

from tensorflow.keras.layers import Flatten

import warnings

warnings.filterwarnings('ignore')

 # 2.Initializing the Model

model=Sequential()

 # 3.Adding CNN Layers

model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

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

model.add(Flatten())

#Model Summary

model.summary()

 # 4.Adding Dense Layers

#model.add(Dense(300,activation='relu'))

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

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

 # 5.Configuring the Learning Process

model.compile(loss='binary_crossentropy',

optimizer='adam',

metrics=['accuracy'])

 # 6.Training the Model

#fit or train the model

r=model.fit_generator(x_train,steps_per_epoch=14,

epochs=10,validation_data=x_test,

validation_steps=2)

import matplotlib.pyplot as plt

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

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

plt.legend()

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

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

plt.legend()

 # 7.Save the Model

model.save("forest1.h5")

ls forest_fire/

 # 8.Test The Model

import numpy as np

from tensorflow.keras.models import load_model

from tensorflow.keras.preprocessing import image

import cv2

#load the model

model=load_model('forest1.h5')

img=image.load_img('/content/drive/MyDrive/Forest-Dataset/forest_fire/Testing/fire/abc169.jpg')

```
img=image.load_img('/content/drive/MyDrive/Forest-Dataset/forest_fire/Testing/fire/abc169.jpg',target_size=(128,128))

img

x=image.img_to_array(img)

x

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

x

y=np.argmax(model.predict(x),axis=1)

y

x_train.class_indices

index=['fire','nofire']

index[y[0]]

img=image.load_img('/content/drive/MyDrive/Forest-Dataset/forest_fire/Testing/fire/abc183.jpg',target_size=(128,128))

x=image.img_to_array(img)

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

index=['fire','nofire']

print('Fire')

img

=image.load_img('/content/drive/MyDrive/Forest-Dataset/forest_fire/Testing/nofire/abc337.jpg',target_size=(128,128))

x=image.img_to_array(img)

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

index=['fire','nofire']

print('No Fire')

img
```

```
=image.load_img('/content/drive/MyDrive/Forest-Dataset/forest_fire/Testing/nofire/abc377.jpg',target_size=(128,128))

x=image.img_to_array(img)

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

index=['fire','nofire']

print('nofire')

img
```

[7:18 pm, 18/11/2022] ⍰:

```
img=image.load_img('/content/drive/MyDrive/Forest-Dataset/forest_fire/Testing/fire/abc173.jpg',target_size=(128,128))

x=image.img_to_array(img)

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

index=['fire','nofire']

print('fire')

img
```

 # 9.Predictions

```
pred = model.predict(x)

pred = np.round(pred)

pred

def predictImage(filename):

 img1=image.load_img(filename,target_size=(128,128))

 plt.imshow(img1)

 y=image.img_to_array(img1)

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

 val=model.predict(x)

 print(val)

 if val==0:
```

```
    plt.xlabel(" No Fire",fontsize=30)
  elif val==1:
    plt.xlabel("Fire",fontsize=30)
predictImage("/content/drive/MyDrive/Forest-Dataset/forest_fire/Testing/fire/abc173.jpg")
plt.xlabel("fire",fontsize=30)
predictImage('/content/drive/MyDrive/Forest-Dataset/forest_fire/Testing/nofire/abc377.jpg')
plt.xlabel(" NO fire",fontsize=30)
```

GitHub & Project Demo Link

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-46165-1660740383

Project Demo Link:

Drive Link: https://drive.google.com/file/d/1QRxLM07572XOT58ZYWTDOhQdKd06-ATu/view?usp=share_link

YouTube link:

https://youtu.be/jTZCYX7wbyk