EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

IBM-Project-51482-1660979898

NALAIYA THIRAN PROJECT BASED LEARNING ON PROFESSIONAL READLNESS FOR INNOVATION, EMPLOYNMENT AND ENTERPRENEURSHIP

A PROJECT REPORT

SUBMITTED BY

SNEGA T

BAVADHARINI E

PRIYA DHARSHINI S

VENMATHI G

TEAM ID **:** PNT2022TMID18694

INDUSTRY MENTOR : Shanthi

FACULTY MENTOR : Kavishree R

COMPUTER SCIENCE AND ENGINEERING IN

MUTHAYAMMAL ENGINEERING COLLEGE

RASIPURAM – 637408

PROJECT REPORT

1. INTRODUCTION

1.1 Project overview:

Fire can make major hazards in this hectic world. All buildings and vehicles used in public transportation have fire prevention and fire protection systems due to the accelerated number in the fire incidents. Also, many of the firms conduct a mock fire drill in every occurrence of months to protect their employees from the fire. This would help them to understand what to do or what not to do when a fire situation happens. Forests are one of the main factors in balancing the ecology. It is very harmful when a fire occurs in a forest. But most of the time, the detection of forest fire happens when it spread over a wide region. Sometimes, it could not be possible to stop the fire. As a result, the damage of the environment is higher than predictable. The emission of large amount of carbon dioxide (CO_2) from the forest fire damages the environment. As well as it would lead to complete disappearance of rare species in the world. Also, it can make an impact on the weather, and this make major issues like earthquakes, heavy rains, floods and so on.
A research study shows an automatic fire detection can be divided into three groups: aerial, ground and borne detection. The ground-based systems use several staring black and white video cameras are used in fire detection which detect the smoke and compares it with the natural smoke. The main benefit of using this system is high temporal resolution and spatial resolution. So that, the detection is easier.2But these mechanisms still have some drawbacks in detecting the early stage of the fire. So that, it is highly important to introduce a system to detect the fire early as possible.

Moreover, information regarding the seat of the hearth is invaluable for the rapid deployment of fire- fighters. Therefore, early detection, containment at the primary stages and extinguishment of a fireplace before it spreads are crucial for wildfire Management.

1.2 Purpose

Forest fires as of late have been annihilating both for normal biological system, biodiversity and woodland economy. With expanding populace weight and change in worldwide atmosphere situation, there is an expansion in level of fires that are a significant reason for declining Indian woodlands. As indicated by woodland study report of India, 50 % of backwoods regions in nation are fire inclined (going from 50 to 90 % in certain conditions of nation). Around 6 % of the woods are inclined to extreme fire harms. The reason for this planned framework is to

manufacture a dependable fire location framework so as to know dynamic status of backwoods temperature in specific conditions. It is about the sensors and dynamic checking framework to dodge a significant fire and genuine harm to woods.

2. LITERATURE SURVEY

2.1Existing problem

The existing system for detecting fire are smoke alarms and heat alarms. The main disadvantage of the smoke sensor alarm and heat sensor alarms are that just one module is not enough to monitor all the potential fire prone places. The only way to prevent fire is to cautious at the time. Even if they are installed in every nook and corner, it just is not sufficient for an efficient output consistently. As the number of smoke sensor requirement increase the cost will also increase to its multiple. The proposed system can produce consistent and highly accurate alerts within seconds of accident of the fire. It reduces cost because only one software is enough to power the entire network of surveillance. Research is active on this field by data scientists and machine learning researchers. The real challenge is to minimize the error in detection of fire and sending alerts at the right time.The idea of this research is to fabricate a system through IoT sensors, which is arbitrarily spread in the forest and to make a self-sorted out powerful system between the sensors to cover all the enormous territories in the forest that will used to maintain a strategic distance from the fire harm whenever. The capacity of the sensor is to identify fire in the inclusion region between the time intermission of each 5-10 minutes. At the point when the fire is recognized the entirety of the sensor in the region will be dynamic and order to stop the normal assignment. The concept is to build early fire detector using Arduino which is connected with different IoT sensors. Putting all efforts to develop a smarter system by connecting it to a webpage and monitoring the developed system statistics controlled by the Arduino programming. The use of latest technology can help to prevent the catastrophic accidents in forests. The aim is to early detect the fireplace in forest by considering the several factor like smoke, temperature, humidity, flame and based on the data we get from this programming, the forest department will be able to take an appropriate decision and the rescue team will be able to arrive on time at exact location. Consider, if it is a large region and it produces more carbon monoxide than the ordinary vehicle traffic. Surveillance of the danger areas and an early detection of fireplace can appreciably shorten the response time and additionally decrease the practicable injury as nicely as the fee of firefighting. Known rule applies here: 1 minute – 1 cup of water, 2 minutes - 100 liters of water, 10 minutes - 1000 liters of water. The goal is to notice the fireplace as quicker as possible, its actual localization and early notification to the fire devices. When fire starts then the flammable texture may likewise issues fuel to the hearth focal spot. The spot at that point will expand and more extensive. The first phase of start is alluded as "surface fire" stage. This may feed on abutting bushes and the fire will turn into higher and transforming into "crown fire". Generally, at this stage the hearth transforms into wild and injury which end up being extreme that could stay for quite long time while depending on atmosphere conditions and the territory. Forest fire detection using optimized solar–powered ZigBee wireless sensor networks- In this paper, they have developed system for Forest Fire Detection which overcomes the demerits of the Existing technologies of Forest Fire Detection. It can be ensured that the system developed can be implemented on a large scale with its promising results. The system is provided with low-power elements, higher versions of Zigbee, Maximum power point tracking Algorithm is used in order to make the system run for longer periods efficiently. Forest fires are a very serious problem in many countries, and global warming may contribute to make this problem worse. Experts agree that, in order to prevent these tragedies from happening, it is necessary to invest in new technologies and equipment that enable a multifaceted approach. This paper describes a WSN for early detection of forest fires. This network can be easily deployed at areas of special interest or risk. There are two types of nodes from the physical structure point of view: SNs, to collect data from the environment, and CNs, to gather data from the SNs and transmit the information to a Control

Centre. The nodes also can be in different functioning modes. This enables a proper and seamless configuration of the network, provides redundancy, and ensures there will be full temporal and geographical coverage in the deployment zone. The information gathered is related not only to early detection purposes but also to environment monitoring to maximize the WSN usage. This environmental data can also be employed to firefighting preventive tasks such as vegetation modelling, microclimate studies, and propagation model parametrization.

Characteristics	Flaming	Smoldering
Emissions	Light gases Particles high in EC	Hydrocarbons, PAH's, mercaptans, partially oxidized gases, particles lower in EC
Flames	visible	not visible
Extent of reaction (combustion efficiency)	Reactions tend to go to completion (90-95%)	Incomplete combustion reactio (60-90%)
O_2 concentrations	>= 15%	>= 5%
Temperature	>300°C (peak of 1800°K)	< 300°C
Combustion efficiency (1)	About 90-95%	About 60-90%

TABLE 1. Comparison of different techniques

In this paper, a forest fire detection algorithm is proposed. The algorithm uses YCbCr color space since it effectively separates luminance from chrominance and is able to separate high temperature fire center pixels because the fire at the high temperature center region is white. The final results show that the proposed system has good detection rates and fewer false alarms, which are the main crucial problems of the most existing algorithms. The presences of fire in video streams are indicated by semantic events. Most of the existing systems can only be used for the videos obtained from stationary cameras and videos obtained from the controlled lightening conditions. These existing automatic fire detection systems cannot be used for video streams obtained from mobile phones or any hand held devices. It was decried as a global tragedy. Lit by farmers, the fires raged through villages, destroyed ecosystems and pumped climate-warming pollution into the atmosphere.

2.2References

• V.Parthipan, D.Dhanasekaran,“ Preventing and Monitoring of Framework for Forest Fire Detection and Data Analysis Using Internet of Things (IoT)“, IJEAT, ISSN: 2249 – 8958, Volume-8, Issue-3S February 2019.

• Shijo Joseph, K. Anitha, M.S.R. Murthy,” Forest fire in India: a review of the knowledge base”,J For Res(2009) 14:127-134.

• Ranjth E, Padmabalaji D, Sibisubramanian S, Radhika S,” An IOT Based Forest Fire Detection and Prevention System using Raspberry PI 3“, IRJET, eISSN: 2395- 0056, p-ISSN: 2395-0072, Volume: 06 Issue: 03, Mar 2019.

• M. Trinath Basu, Ragipati Karthik, J. Mahitha, V. Lokesh Reddy,“IoT Based Forest Fire Detection System“, IJET, 7(2.7) (2018) 124-126.

• T. Saikumar, P. Sriramya, „“IoT Enabled Forest Fire Detection and Alerting the Authorities“, IJRTE, ISSN: 2277-3878, Volume-7, Issue-6S4, April 2019. 6. Wolfgang KRÜLL, Robert TOBERA, Ingolf WILLMS, , Helmut ESSEN, Nora von WAHL,“ Early forest fire detection and verification using optical smoke, gas and microwave sensors“, / Procedia Engineering 45 (2012) 584 – 594.

• Stanmir Zivanovic, Darko Zigar, Dejan Krstic, “The Role of Early Detection of Forest Fire in Environmental Protection“, Safety Engineering, Vol 3, N2 (2013) 93-97. • Antonio Molina-Pico, David Cuesta-Frau, Alvaro Araujo, Javier Alejandre, Alba Rozas,“Forest Monitoring and Wildland Early Fire Detection by a Hierarchical Wireless Sensor Network“, Volume 2016 |Article ID 8325845 | 8 pages, Source: Hindawi.

• Ahmad A. A. Alkhatib, “A Review on Forest Fire Detection Techniques“, Volume 2014, Article ID 597368,12 Pages, Source: Hindawi.

• Yasar Guneri Sahin, Turker Ince,“Early Forest Fire Detection Using RadioAcoustic Sounding System“, Sensors 2009, 9(3), 1485-1498, Source: MDPI.

• Jijitha R., Shabin P.“ A Review on Forest Fire Detection“, HBRP Publication Page 1- 8 2019, Volume 2 Issue

• R. Chandrasekharan, Ashiq M.I, Dr. V. Prakash,“ Forest Fire Detection using Temperature Sensors Powered by Tree and Auto Alarming using GSM“, IJRSI, Volume II, Issue III, March 2015, ISSN 2321 – 2705.

• U. Arun Ganesh, M. Anand, S. Arun, M. Dinesh, P. Gunaseelan and R. Karthik,“ Forest Fire Detection Using Optimized Solar – Powered Zigbee Wireless Sensor Networks“, IJSER, Volume 4, Issue 6, June-2013, ISSN 22295518.

D.Vignesh Kirubaharan, A.John Clement Sunder, S.M.Ramesh,

P.Dhinakar,“ Forest Fire Prediction and Alert System Using Wireless Sensor Network“, IJARECE,Volume 3, Issue 11, November 2014.Vinay Chowdary, Mukul Kumar Gupta,“Automatic Forest Fire Detection and Monitoring Techniques: A Survey“, Source: researchgate.net.

2.3 Problem Statement Definition

- In earlier times fires were detected with the help of watching towers or using satellite images.
- Satellites collect images and send it to the monitoring authority which will decide by seeing images that it is a fire or not.
- But this approach was very slow as the fire may have spread in the large areas and caused so much damage before the rescue team came.
- In the watching tower method, there was a man always standing on the tower who would monitor the area and inform if there was fire.
- This method was also slow because before the man got to know about the fire it may have spread in the inner parts of forest, also it always requires a man who must be present there.
- Since, we know that some areas, especially forest areas are large so it is practically impossible to put a man in every part of forest from where they can monitor the forest area.
- So, both these approaches of watching towers and satellite images failed to detect fire as early as possible to reduce the damage done by fire Problems in fire detection:
- There were mainly two problems in fire detection as discussed:

- (a). Judging criteria for the fire: Edge is set, on the off chance that the worth is more noteworthy than edge, it is a fire, else not.
- So, this problem was removed by using machine learning techniques by many researchers.
- (b). Connection of nodes: Traditional systems used cables to connect alarm with the detectors.
- Cable was mainly of copper. But copper wire may be costly or it can suffer from fault in the mid-way.
- So, this problem was removed using wireless sensor networks.
- So, with the advancement in technology researchers find an efficient method to detect forest fire with the help of Wireless Sensor Network.
- Fire can be identified by conveying sensor hubs in timberland regions by which they illuminate about fire.
- Conveying sensor hubs in the timberland regions means placing sensors in every part of the forest and mostly in the prone areas where risk of 9 catching fire is more. With the use of wireless sensor networks, now it is easy to detect the fire in large areas as soon as possible.

3. IDEATION & PROPOSED SOLUTION

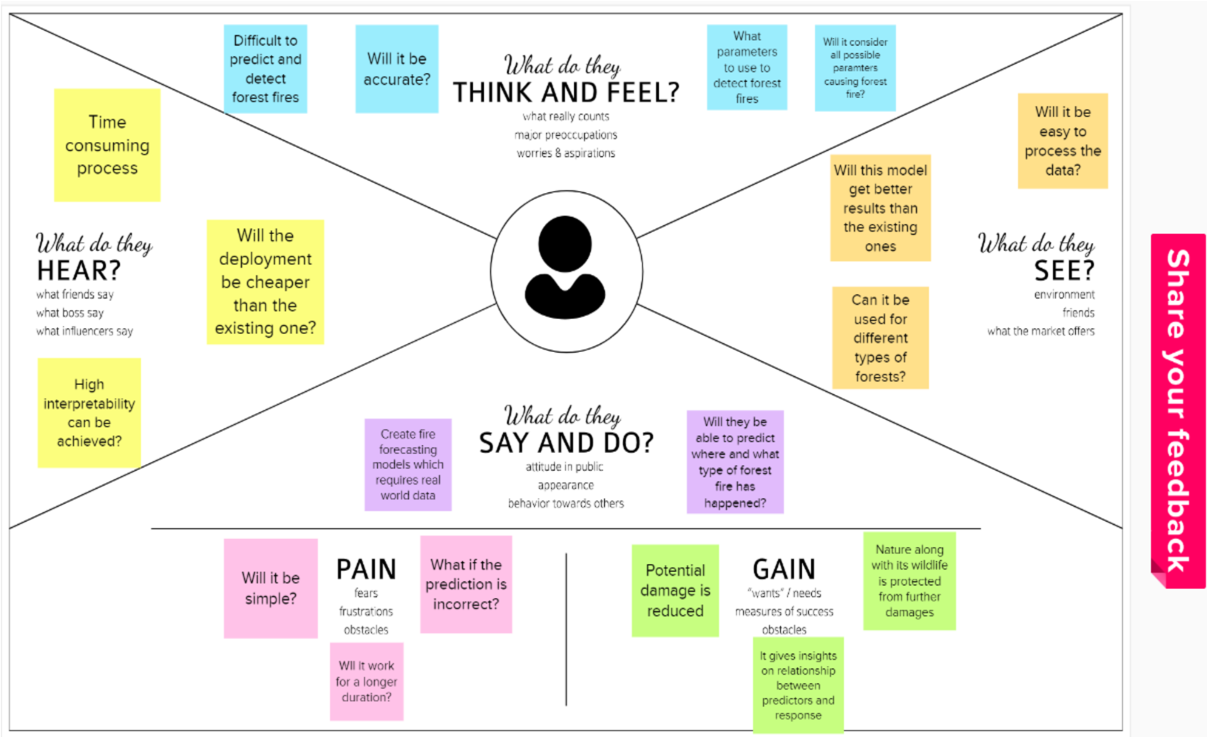
3.1 Empathy Map Canvas

1. An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.

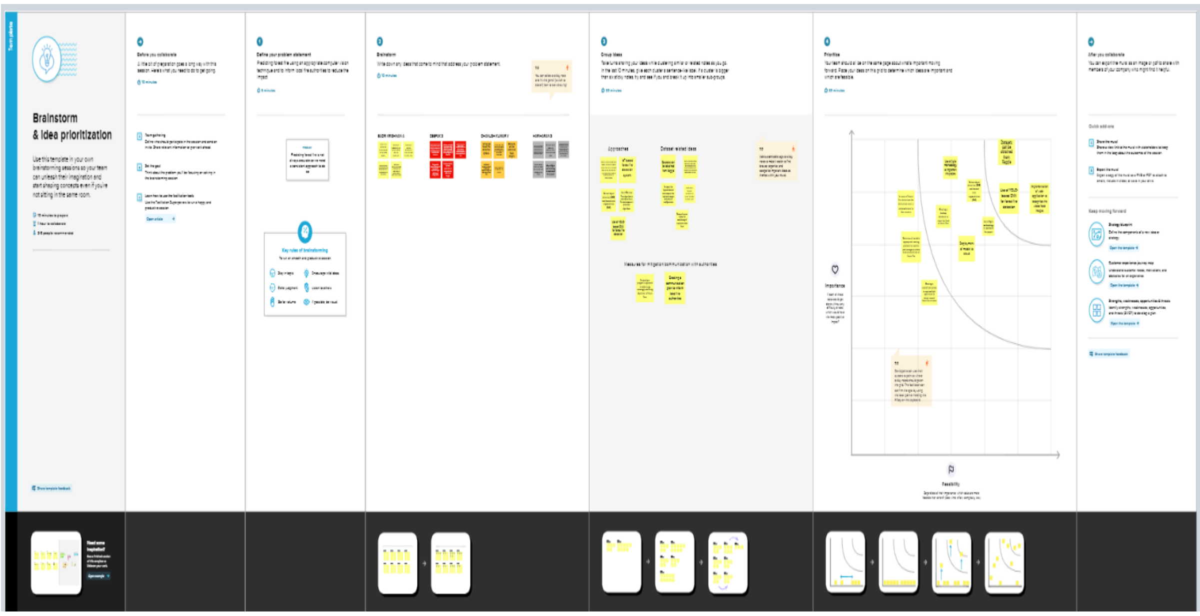
2. An empathy map is an effective visualization template that helps analyze the behavior and emotions of customers and users. Empathy maps not only detect the behaviors but highlight possible mediums for brands to communicate with their customers in a better way

3. Empathy maps can also be used to collect data directly from the users. Used alongside user interviews, survey answers, etc., you can also have a user fill in an empathy map themselves. This often reveals aspects of the user that may have remained unsaid or not thought of.

4. Each of the four quadrants comprises a category that helps us delve into the mind of the user. The four empathy map quadrants look at what the user says, thinks, feels, and does.



Ideation & Brainstorming



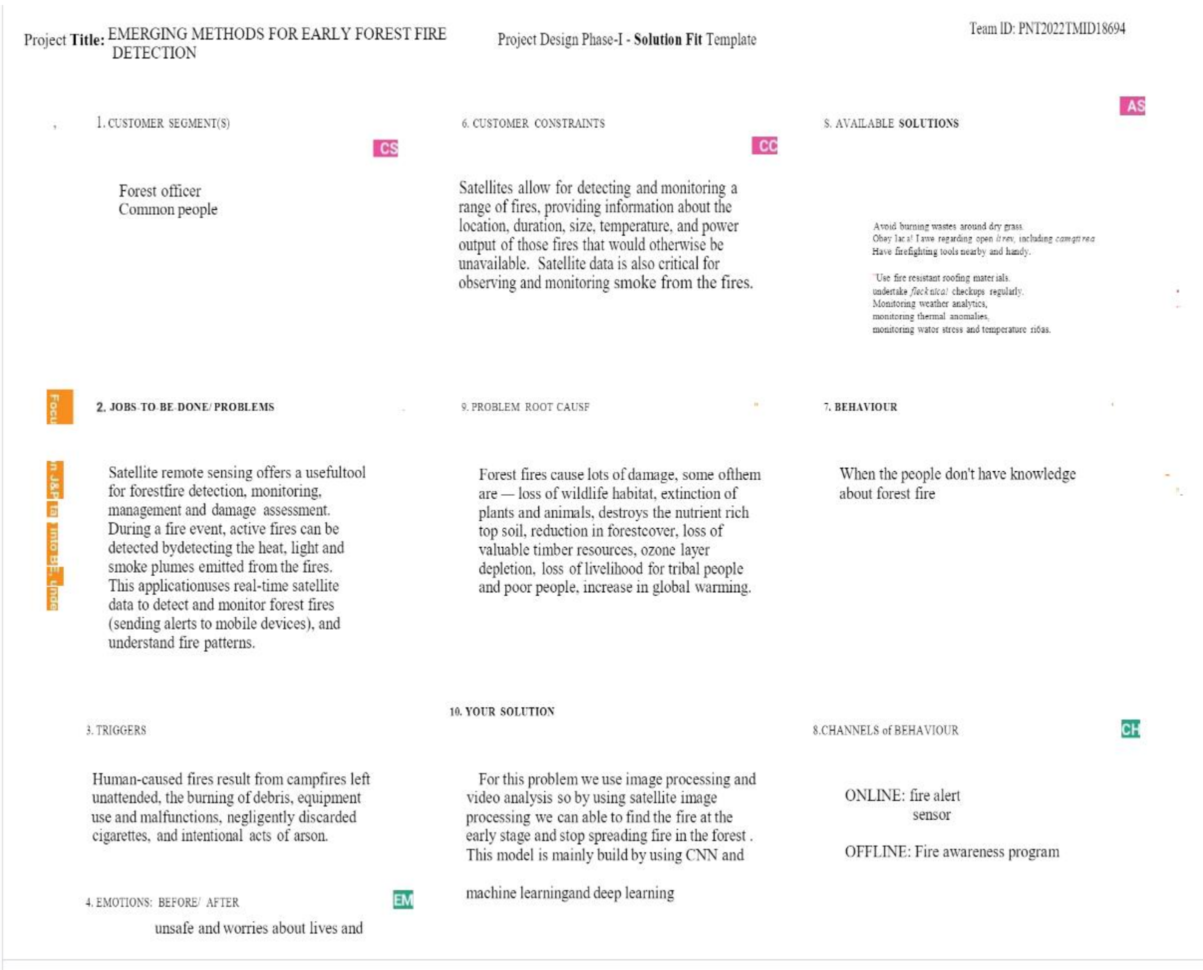
3.1 Proposed Solution

Project team shall fill the following information in proposed solution template.

	Parameter	Description
	Problem Statement (Problem to be solved)	A forest fire risk prediction algorithm, based on support vector machines, is presented. The algorithm depends on previous weather conditions in order to predict the fire hazard level of a day.
	Idea / Solution description	Use computer vision methods for recognition and detection of smoke or fire, based on the still images or the video input from the drone cameras.
	Novelty / Uniqueness	Real time computer program detect forest fire in earliest before it spread to larger area.
	Impact on society	Blocked roads and railway lines, electricity, mobile and land telephone lines cut, destruction of homes and industries.
	Business Model (Revenue Model)	The proposed method was implemented using the Python programming language on a Core i3 or greater (CPU and 4GB RAM.)

	Scalability of the Solution	Computer vision models enable land cover classification and smoke detection from satellite and ground cameras

3.2 Problem Solution fit


Project Title: EMERGING METHODS FOR EARLY FOREST FIRE DETECTION

Project Design Phase-I - **Solution Fit** Template

Team ID: PNT2022TMID18694

1. CUSTOMER SEGMENT(S) CS

Forest officer
Common people

6. CUSTOMER CONSTRAINTS CC

Satellites allow for detecting and monitoring a range of fires, providing information about the location, duration, size, temperature, and power output of those fires that would otherwise be unavailable. Satellite data is also critical for observing and monitoring smoke from the fires.

5. AVAILABLE **SOLUTIONS** AS

Avoid burning wastes around dry grass.
Obey lac a! I awe regarding open ltrev, including camgtt rea
Have firefighting tools nearby and handy.

Use fire resistant roofing mater ials.
undertake fleck nica! checkups regularly.
Monitoring weather analytics,
monitoring thermal anomalies,
monitoring wator stress and temperature riбas.

2. JOBS-TO-BE-DONE/ PROBLEMS

Satellite remote sensing offers a usefultool for forestfire detection, monitoring, management and damage assessment. During a fire event, active fires can be detected bydetecting the heat, light and smoke plumes emitted from the fires. This applicationuses real-time satellite data to detect and monitor forest fires (sending alerts to mobile devices), and understand fire patterns.

9. PROBLEM ROOT CAUSE

Forest fires cause lots of damage, some ofthem are — loss of wildlife habitat, extinction of plants and animals, destroys the nutrient rich top soil, reduction in forestcover, loss of valuable timber resources, ozone layer depletion, loss of livelihood for tribal people and poor people, increase in global warming.

7. BEHAVIOUR

When the people don't have knowledge about forest fire

3. TRIGGERS

Human-caused fires result from campfires left unattended, the burning of debris, equipment use and malfunctions, negligently discarded cigarettes, and intentional acts of arson.

10. YOUR SOLUTION

For this problem we use image processing and video analysis so by using satellite image processing we can able to find the fire at the early stage and stop spreading fire in the forest . This model is mainly build by using CNN and machine learningand deep learning

8.CHANNELS of BEHAVIOUR CH

ONLINE: fire alert sensor

OFFLINE: Fire awareness program

4. EMOTIONS: BEFORE/ AFTER EM

unsafe and worries about lives and

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

4.1.1.High Priority

1. The system shall take training sets of fire images and recognize whether there is a fire or the beginning of a fire (smoke) or if there is no fire
2. The system shall send a notification to the admin when it recognizes a fire in the image given

3. The system shall take real inputs of camera images and determine whether the image contains a fire or not
4. The system shall be able to take images with a variety of sizes and convert it to one fixed image to be used throughout the application.
5. The system shall run as a service on either a Windows or Linux operating system.
6. In the event that the computer on which the system is running shuts down, the system service should start automatically when the computer restarts

4.1.2.Medium Priority

1. The system shall provide following facility that will allow web pages that the user is permitted to access. The system must support the following facility:
    a. Send alert message
    b. Customer data management

4.1.3.Low Priority

1. The system shall allow the user's status to be stored for the next time he returns to the web site. This will save the user x minutes per visit by not having to reenter already supplied data.
2. The system shall provide information about event log of forest.

4.2 Non Functional Requirements

3. Non-Functional Objectives

4.2.1.Reliability

- The system shall be completely operational at least x% of the time.
- Down time after a failure shall not exceed x hours.

4.2.2.Usability

- Customer should be able to use the system in his job for x days .
- A user who already knows what camera he is using should be able to connect and view that page in x seconds.

4.2.3.Performance

- The system should be able to support x simultaneous users.
- The mean time to view a web page over a 56Kbps modem connection shall not exceed x seconds..

4.2.4.Security

- The system shall provide password protected access to web pages that are to be viewed only by users.

4.2.5.Supportability

- The system should be able to accommodate many camera links.
- The system web site shall be viewable from chrome or any browser.

4.2.6.Interfaces

The system must interface with

- The cloudant db for customer and customer log information
- The acquired web site search engine.

5. PROJECT DESIGN

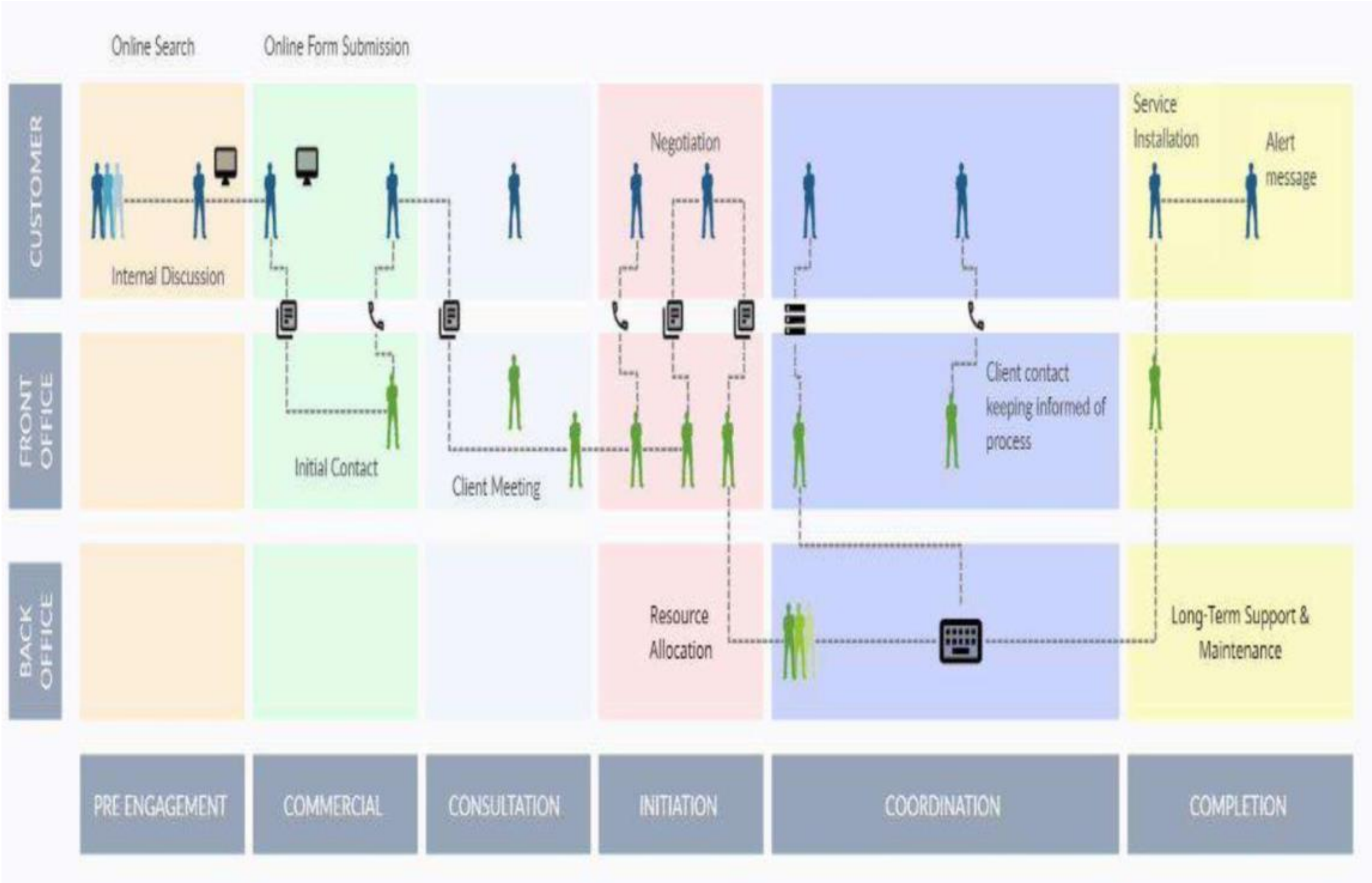


Fig 5.1

5.1 Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

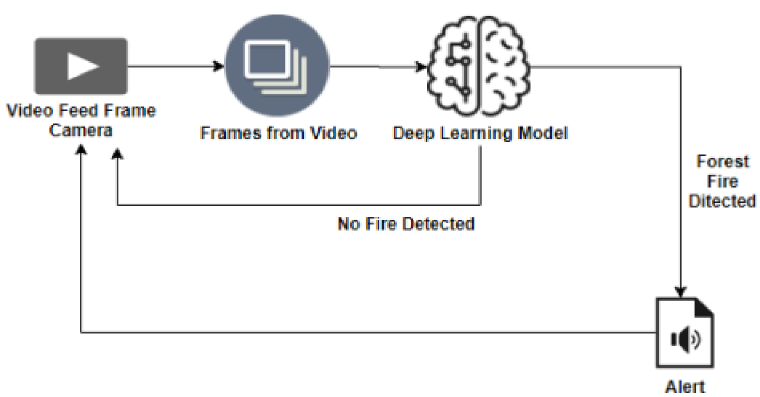


Fig 5.2

5.2 Solution with technical architecture:

- If the fire is not detected ,it will send the result to the frame camera ,if the forest fire will detected the alert will go to the video feed frame camera.
- 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.

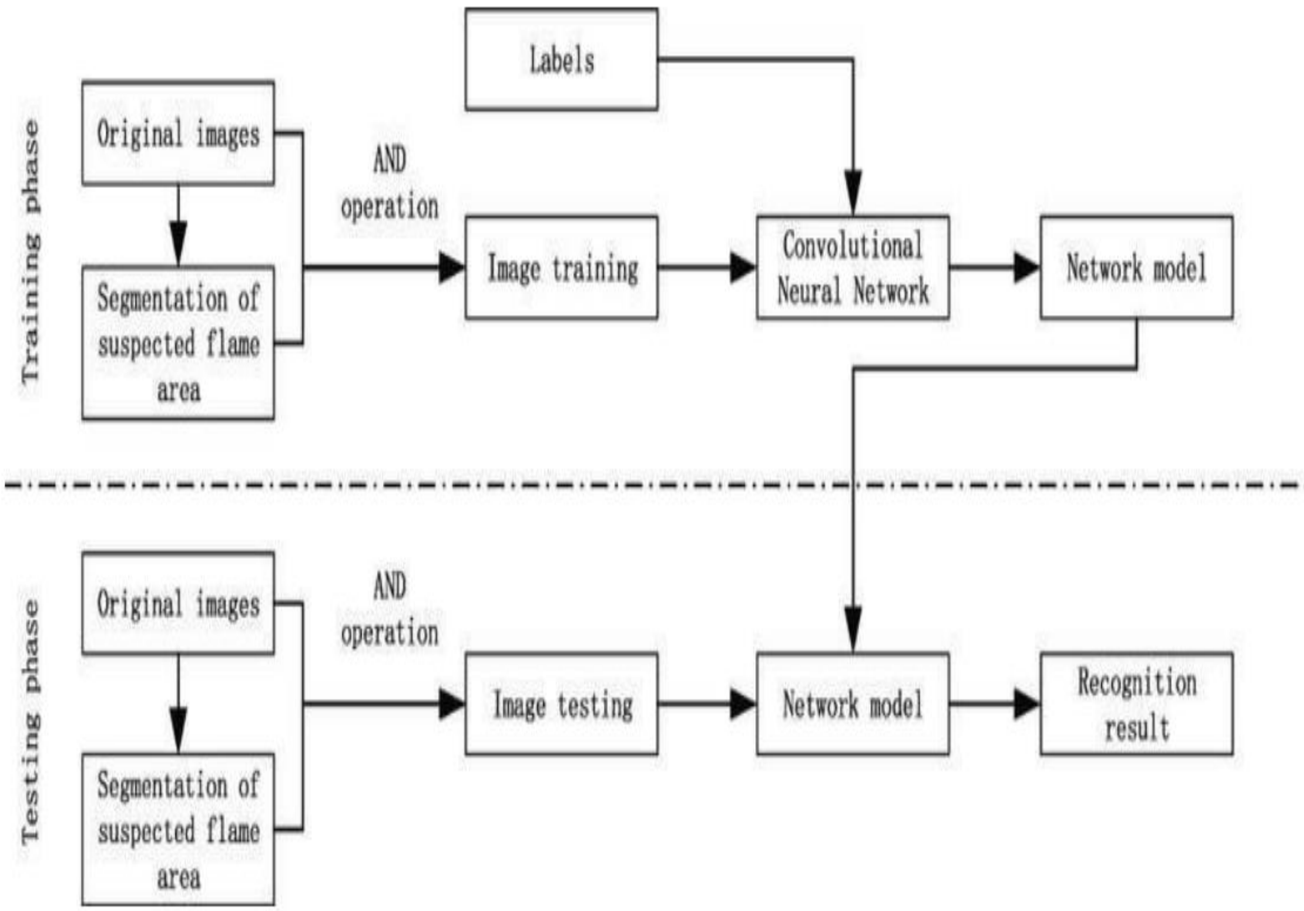


Fig 5.3

- The various real-time forest fire detection and prediction approaches, with the goal of informing the local fire authorities

5.2 User Stories

Collects data set , algorithm and defines how the planning phase is been optimized for the environment , weather and necessary parameters

Training and testing will be based on these matrices.

User Type	Functional Requirement (Epic)	User Story Number	User Story I Task	Acceptance criteria	Priority
Environmentalist	Collect the data	USN-1	As an Environmentalist.it is necessary to collect the data of the forest which includes temperature,humidity,wind and rain of the forest	It is necessary to collect the right data else the prediction may become wrona	High
		USN-2	Identify algorithms that can be used for prediction	To collect the algorithm to identify the accuracy level of each algorithms	Medium
	Implement Algorithm	USN-3	Identify the accuracy of each algorithms	Accuracy of each algorithm-calculated so that it is easy to obtain the most accurate output	High
		USN-4	Evaluate the Dataset	Data is evaluatedbefore processing	Medium
	Evaluate Accuracy of Algorithm	USN-5	Identify accuracy,precision,recall of each algorithms	These values are important for obtaining the riaht output	High

Fig 5.4

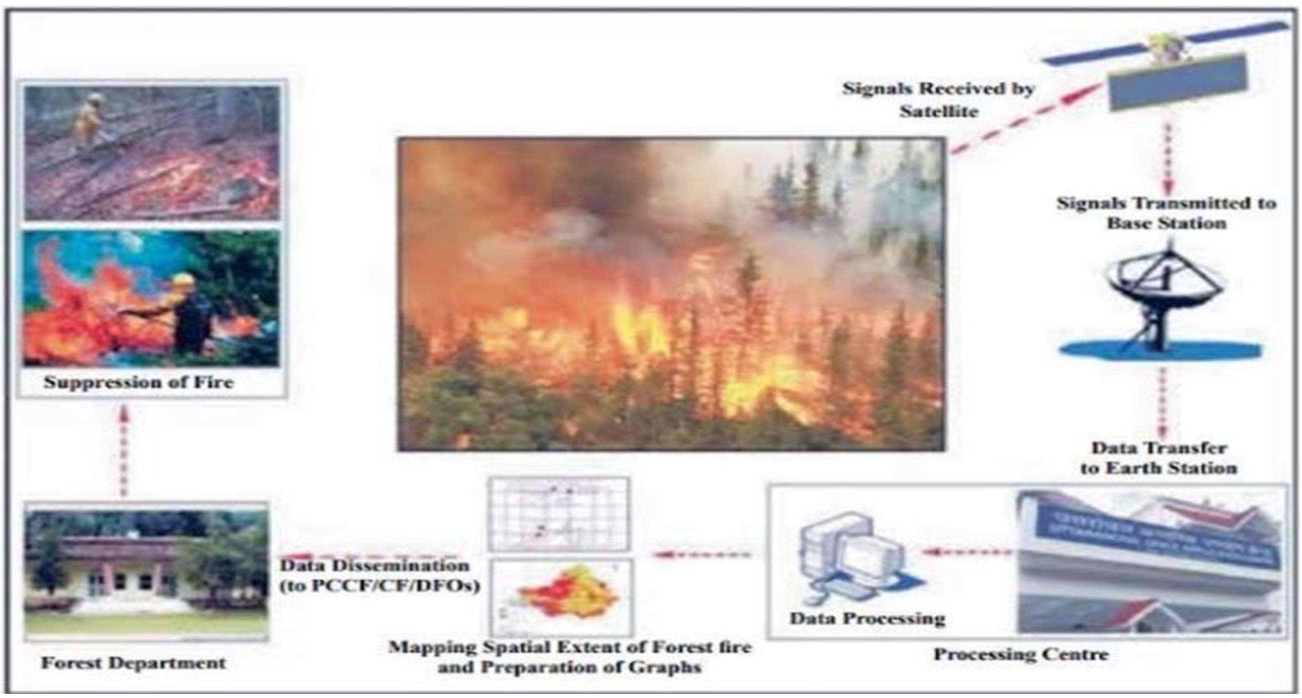


Fig 5.5

6. PROJECT PLANNING & SCHEDULING:

6.1.Sprint Planning & Estimation

Name	Milestone Number	Description	Mandatory	Optional
Project Objectives	M-001	We will be able to learn to prepare dataset, image processing, working with CNN layers, read images using OpenCV and CNN for computer vision AI	Yes	-
Project Flow	M-002	A project management process flowchart is a graphical aid, designed to visualize the sequence of steps to be followed throughout the project management process	Yes	
Pre-Requisites	M-003	To complete this project we should have known following project such as Keras, Tensorflow, Python , Anaconda, OpenCV, Flask, Scikit-learn etc…	Yes	
Prior Knowledge	M-004	One should have knowledge on the Supervised Learning ,CNN and Regression Classification and Clustering, ANN	Yes	
Data collection	M-005	We can collect dataset from different open sources like kaggle.com, UCI machine learning etc	Yes	
Image Preprocessing	M-006	Importing the ImageDataGenerator libraries, Define Parameters/Arguments for ImageDataGenerator class, Applying Image Data Generator Functionality to trainset and testset	Yes	
Model Building	M-007	Importing the model building libraries, Initializing the model, Adding CNN layers, Adding Dense layers, Configuring the learning Process ,Train the model, Save the model, Predictions.	Yes	
Video Analysis	M-008	Open cv for video processing, creating an account in twilio service and Sending alert messages.	Yes	
Train CNN model	M-009	Register for IBM Cloud and train Image Classification Model	Yes	
Ideation Phase	M-010	Prepare Literature Survey on the selected Project and Information Gathering, empathy map and ideation	Yes	
Project Design Phase-I	M-011	Prepare Proposed solution , problem-solution fit and Solution Architecture	Yes	
Project Design Phase-II	M-012	Prepare Customer journey ,functional requirements, Data flow diagram and Technology Architecture	Yes	
Project Planning Phase	M-013	Prepare Milestone list , Activity list and Sprint Delivery Plan	Yes	
Project Development Phase	M-014	Project Development delivery of Sprint 1, Sprint 2, Sprint 3, Sprint 4	Yes	

6.2.1 Activity list

1.1	Access Resources	Access the resources (courses) in project dashboard.	All Members	COMPLETED
1.2	Rocket chat registration	Join the mentoring channel via platform& rocket-chat mobile app.	All Members	COMPLETED
1.3	Access workspace	Access the guided project workspace.	All Members	COMPLETED
1.4	IBM Cloud registration	Register on IBM Academic Initiative &Apply Feature code for IBM Cloud Credits.	All Members	COMPLETED
1.5	Project Repository Creation	Create GitHub account & collaborate with Project Repository in project workspace.	All Members	COMPLETED
1.6	Environment Setup	Set-up the Laptop / Computers based on the pre-requisites for each technology track.	All Members	COMPLETED
2.1	Literature survey	Literature survey on the selected project & Information Gathering.	All Members	COMPLETED
2.2	Technology Training	Attend the technology trainings as per the training Calendar.	All Members	COMPLETED
2.3	Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	All Members	COMPLETED
2.4	Technology Training	Attend the technology trainings as per the training Calendar.	All Members	COMPLETED
2.5	Brainstorming	List the ideas (at least 4 per each team member) by organizing the brainstorm session and prioritize the ideas	All Members	COMPLETED
2.6	Technology Training	Attend the technology trainings as per the training Calendar.	All Members	COMPLETED
3.1	Proposed Solution Document	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	All Members	COMPLETED

3.2	Technology Training	Attend the technology trainings as per the training Calendar.	All Members	COMPLETED
3.3	Problem - Solution fit & Solution Architecture	Prepare problem - solution fit document& Solution Architecture.	All Members	COMPLETED
3.4	Technology Training	Attend the technology trainings as per the training Calendar.	All Members	COMPLETED
4.1	Customer Journey Map	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	All Members	COMPLETED
4.2	Technology Training	Attend the technology trainings as per the training Calendar.	All Members	COMPLETED
4.3	Functional Requirements & Data Flow Diagrams	Prepare the Functional Requirement Document & Data Flow Diagrams.	All Members	COMPLETED
4.4	Technology Architecture	Prepare Technology Architecture of the solution.	All Members	COMPLETED
4.5	Technology Training	Attend the technology trainings as per the training Calendar.	All Members	COMPLETED
5.1	Milestone& Activity List	Prepare Milestone & Activity List.	All Members	COMPLETED
5.2	Sprint Delivery Plan	Prepare Sprint Delivery Plan.	All Members	COMPLETED
6	Data Collection	Collect datasets from different open sources like kaggle.com, data.gov, UCI machine learning repository, etc.	All Members	COMPLETED
7.1	Image Preprocessing	Importing the ImageDataGenerator Library	All Members	COMPLETED
7.2	Image Preprocessing	Define the parameters/arguments for ImageDataGenerator class.	All Members	COMPLETED
7.3	Image Preprocessing	Applying ImageDataGenerator functionality to trainset and test set.	All Members	COMPLETED
8.1	Model Building	Importing the model building libraries.	All Members	COMPLETED
8.2	Model Building	Initializing the model.	All Members	COMPLETED
8.3	Model Building	Adding CNN Layers.	All Members	COMPLETED

8.4	Model Building	Adding Dense Layers	All Members	COMPLETED
8.5	Model Building	Configuring the learning process	All Members	COMPLETED
8.6	Model Building	Training the Model	All Members	COMPLETED
8.7	Model Building	Save the model	All Members	COMPLETED
8.8	Model Building	Predictions	All Members	COMPLETED
9.1	Video Analysis	OpenCV for video processing.	All Members	COMPLETED
9.2	Video Analysis	Creating an account in Twilio service.	All Members	COMPLETED
9.3	Video Analysis	Sending alert message.	All Members	COMPLETED
10.1	Train CNN Model on IBM	Register for IBM Cloud	All Members	COMPLETED
10.2	Train CNN Model on IBM	Train Image Classification Model	All Members	COMPLETED

6.2 Sprint delivery plan

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 ov 2022

6.3Reports from JIRA

JIRA has categorized reports in four levels, which are −

1.6.1.Agile

1.6.2.Issue Analysis

1.6.3.Forecast & Management

1.6.4.Others

VELOCITY: SPRINT - 1

Sprint duration = 5 days

Velocity of team = 20 points

Average Velocity (AV) = $\dfrac{\text{Velocity}}{\text{Sprint duration}}$

AV = 20/5 = 4

Average Velocity = 4

VELOCITY: Sprint 1 - 4

Sprint duration = 20 days

Velocity of team = 80 points

Average Velocity (AV) = $\dfrac{\text{Velocity}}{\text{Sprint duration}}$

AV = 80/20 = 4

Total Average Velocity = 4

7. CODING & SOLUTIONING

7.1 Feature 1

```
!pip install tensorflow

!pip install opencv-python

!pip install opencv-contrib-python

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

train=ImageDataGenerator(rescale=1./255,
                         shear_range=0.2,
                         rotation_range=180,
                         zoom_range=0.2,
                         horizontal_flip=True)

train = ImageDataGenerator(rescale=1/255)

test = ImageDataGenerator(rescale=1/255)

train_dataset = train.flow_from_directory("/content/drive/MyDrive/Dataset/train_set",
                                          target_size=(128,128),
                                          batch_size = 32,
                                          class_mode = 'binary' )

test_dataset = test.flow_from_directory("/content/drive/MyDrive/Dataset/test_set",
                                        target_size=(128,128),
                                        batch_size = 32,
                                        class_mode = 'binary' )

test_dataset.class_indices

#to define linear initialisation import sequential

from keras.models import Sequential

#to add layer 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')
model =Sequential()
#add convolutional layer
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())
model.add(Dense(150,activation='relu'))

model.add(Dense(1,activation='sigmoid'))
model.compile(loss = 'binary_crossentropy',
              optimizer = "adam",
              metrics = ["accuracy"])
model.fit_generator(x_train,steps_per_epoch=14,epochs=5,validation_data=x_test,valida
tion_steps=4)
model.save("/content/drive/MyDrive/archive(1)/forest1.h5")
predictions = model.predict(test_dataset)
predictions = np.round(predictions)
predictions
print(len(predictions))
#import load_model from keras.model
from keras.models import load_model
#import image class from keras
import tensorflow as tf
from tensorflow.keras.preprocessing import image
```

```python
#import numpy

import numpy as np

#import cv2

import cv2

#load the saved model

model = load_model("/content/drive/MyDrive/archive(1)/forest1.h5")

def predictImage(filename):

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

  Y = image.img_to_array(img1)

  X = np.expand_dims(Y,axis=0)

  val = model.predict(X)

  print(val)

  if val == 1:

    print(" fire")

  elif val == 0:

      print("no fire")

predictImage("/content/drive/MyDrive/Dataset/test_set/with fire/19464620_401.jpg")
```

7.2 Feature 2

```
!pip install tensorflow

!pip install opencv-python

!pip install opencv-contrib-python

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

train=ImageDataGenerator(rescale=1./255,

                                  shear_range=0.2,

                                  rotation_range=180,

                                  zoom_range=0.2,

                                  horizontal_flip=True)

train = ImageDataGenerator(rescale=1/255)

test = ImageDataGenerator(rescale=1/255)

train_dataset = train.flow_from_directory("/content/drive/MyDrive/Dataset/train_se
t",

                                          target_size=(128,128),

                                          batch_size = 32,

                                          class_mode = 'binary' )

test_dataset = test.flow_from_directory("/content/drive/MyDrive/Dataset/test_set",

                                          target_size=(128,128),

                                          batch_size = 32,

                                          class_mode = 'binary' )

test_dataset.class_indices

#to define linear initialisation import sequential

from keras.models import Sequential

#to add layer import Dense

from keras.layers import Dense
```

```python
#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')

model =Sequential()

#add convolutional layer

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())

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


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

model.compile(loss = 'binary_crossentropy',

              optimizer = "adam",

              metrics = ["accuracy"])

model.fit_generator(x_train,steps_per_epoch=14,epochs=5,validation_data=x_test,val
idation_steps=4)

model.save("/content/drive/MyDrive/archive(1)/forest1.h5")

predictions = model.predict(test_dataset)

predictions = np.round(predictions)

predictions

print(len(predictions))

#import load_model from keras.model

from keras.models import load_model

#import image class from keras

import tensorflow as tf

from tensorflow.keras.preprocessing import image
```

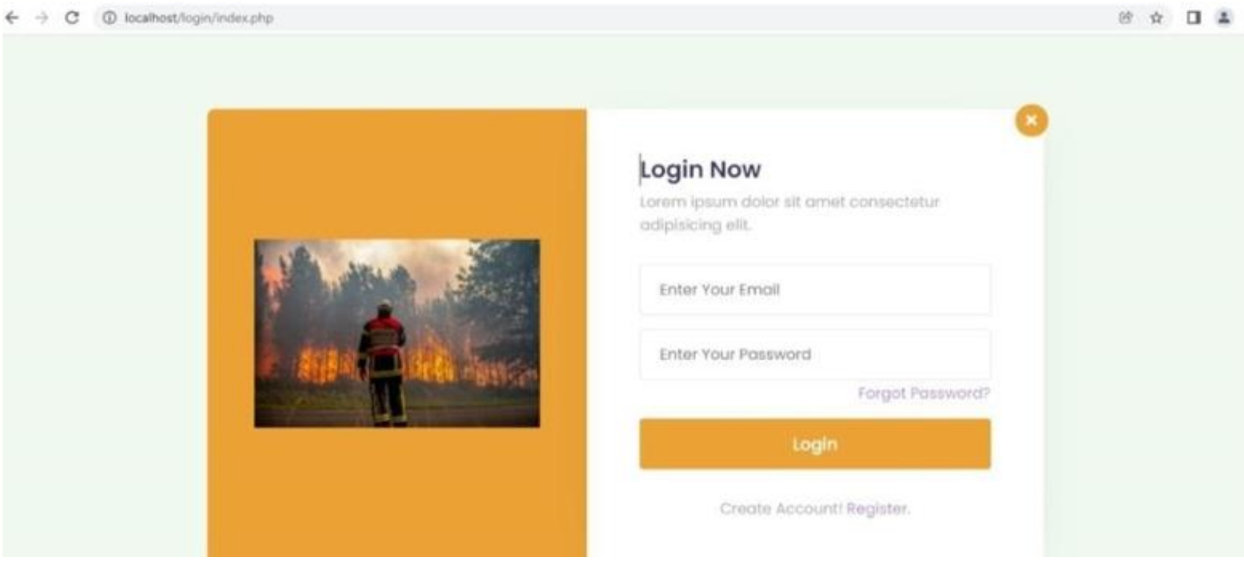
```
    #import numpy
    import numpy as np
    #import cv2
    import cv2
    #load the saved model
    model = load_model("/content/drive/MyDrive/archive(1)/forest1.h5")
    def predictImage(filename):
      img1 = image.load_img(filename,target_size=(128,128))
      Y = image.img_to_array(img1)
      X = np.expand_dims(Y,axis=0)
      val = model.predict(X)
      print(val)
      if val == 1:
        print(" fire")
      elif val == 0:
          print("no fire")
    predictImage("/content/drive/MyDrive/Dataset/test_set/with fire/19464620_401.jpg")
pip install twilio
pip install playsound
#import opencv librariy
import cv2
#import numpy
import numpy as np
#import image function from keras
from keras.preprocessing import image
#import load_model from keras
from keras.models import load_model
#import client from twilio API
from twilio.rest import Client
#imort playsound package
from playsound import playsound
#load the saved model
```

```
model = load_model(r'/content/drive/MyDrive/archive(1)/forest1.h5')

#define video

video = cv2.VideoCapture('/content/Fighting Fire with Fire _ Explained in 30 Seconds.
mp4')

#define the features

name = ['forest','with forest']

account_sid='ACfb4e6d0e7b0d25def63044919f1b96e3'

auth_token='f9ae4fc4a617a527da8672e97eefb2d8'

client=Client(account_sid,auth_token)

message=client.messages \

.create(

      body='Forest Fire is detected, stay alert',

      from_='+1 302 248 4366',

      to='+91 99400 12164'

)

print(message.sid)

pip install pygobject

def message(val):

  if val==1:

    from twilio.rest import Client

    print('Forest fire')

    account_sid='ACfb4e6d0e7b0d25def63044919f1b96e3'

    auth_token='f9ae4fc4a617a527da8672e97eefb2d8'

    client=Client(account_sid,auth_token)

    message=client.messages \

     .create(

        body='forest fire is detected, stay alert',

        #use twilio free number

        from_='+1 302 248 4366',

        #to number

        to='+91 99400 12164')

    print(message.sid)

    print("Fire detected")
```

```
    print("SMS Sent!")
  elif val==0:
    print('No Fire')
from matplotlib import pyplot as plt
#import load model from keras.model
from keras.models import load_model
#import image from keras
from tensorflow.keras.preprocessing import image
img1 = image.load_img('/content/drive/MyDrive/Dataset/test_set/with fire/Wild_fires.jpg',target_size=(128,128))
Y = image.img_to_array(img1)
x = np.expand_dims(Y,axis=0)
val = model.predict(x)
plt.imshow(img1)
plt.show()
message(val)
img2 = image.load_img('/content/drive/MyDrive/Dataset/test_set/forest/1200px_Mountain area.jpg',target_size=(128,128))
Y = image.img_to_array(img2)
x = np.expand_dims(Y,axis=0)
val = model.predict(x)
plt.imshow(img2)
plt.show()
message(val)
```

8. TESTING

8.1. Test Cases



9. RESULTS

9.1. Performance Metrics

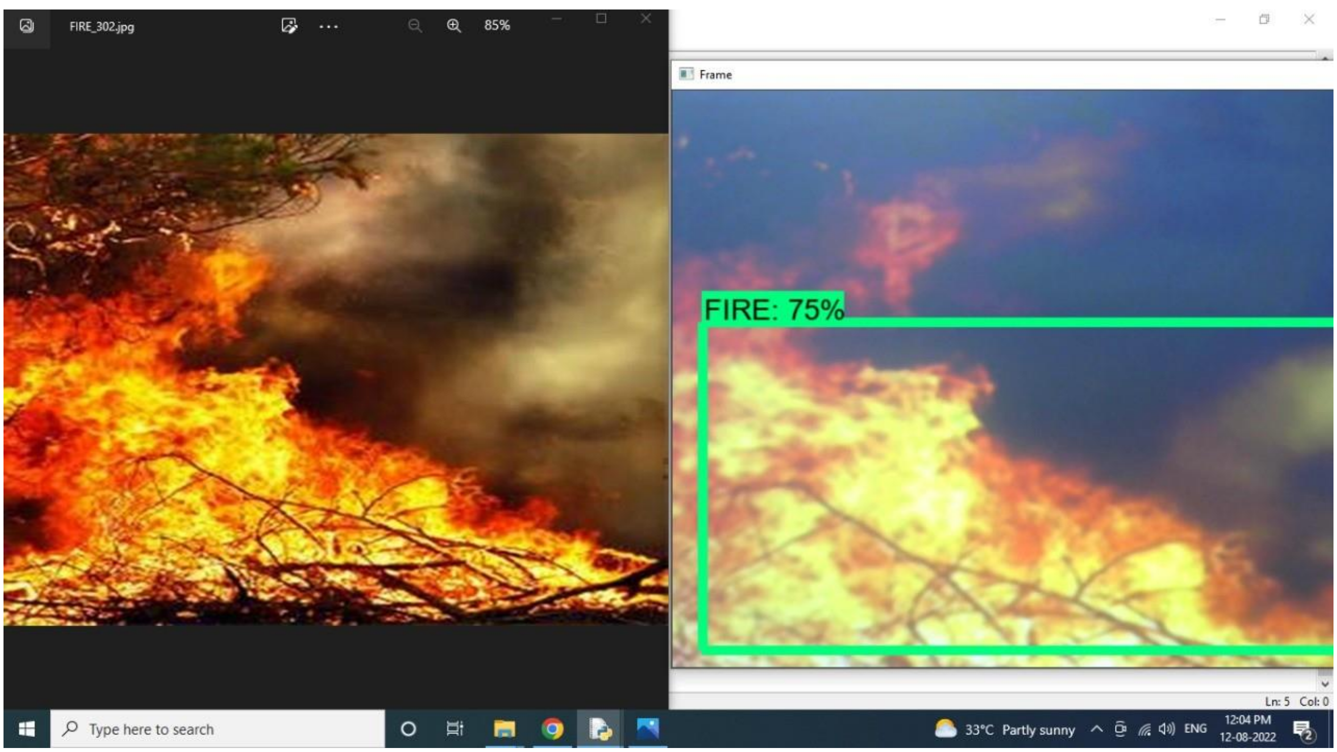


Fig 9.1

Value obtained from three sensor, if any Infrared ray detected, it gives output as IR detected, Sensor activated! Similarly, if there is any temperature change it will show Abnormal temperature and its intensity.

For any smoke detection it output as Smoke detected and sensor value.

Above image is result obtained from the trained ML model showing count for damaged and intact homes.

10.ADVANTAGES & DISADVANTAGES

Advantages:

1. It refreshes the habitat zones: Fire clears out plants and trees to make more natural resources available to the habitat. Fewer trees mean more water becomes available for the remaining plants and animals that call the area their home. New grass and shrubs are food sources for a number of animals as well. A ground cover that comes back after a fire becomes a new micro-habitat. Everything is refreshed with a fire.

2. Low-intensity fires don't usually harm trees**:** The bark of a tree is like an armored shell against fire, pests, and other things that could damage them. Most forest fires burn at low-temperature levels when conditions are optimal and this causes minimal damage to the trees of the forest when it occurs. The end result is a clearing of the ground floor of the forest while the trees are able to continue standing majestically.

3.Decreases the Wastes on Forests: Forests have a lot of waste that ends up building up over time and these wastes can help create wildfires. If a large wildfire breaks out it might take weeks to control it and the damage it can cause is just too extensive to understand for us. Waste such as dead leaves on the ground can be pretty useful for wildfires to feed on and small forest fires just deal with these wastes properly without going out of control.

Disadvantages:

1.A forest fire sets up the potential for soil erosion to occur: Forest fires clear the underbrush away and encourage new growth, but there is a period of time between the fire and the new growth where the forest is vulnerable.

2.Forest fires always bring death in some form**:** Maybe it's just the weak plants of the forest that are killed during a fire, but there is always some sort of death that happens when a fire occurs. Sometimes it is the firefighters who are tasked with stopping the fire. It could be animals or pets.

3.Uncontrolled fires can cause localized air pollution**:** Despite the amount of global development that has occurred, there are many forests that are difficult or nearly impossible to reach. Fires in these areas are left to burn in an uncontrolled fashion and this creates air pollution which can affect the local environment and make it difficult to breathe.

11.CONCLUSION

This project will help in early detection of forest fire and the prevention. It also involves the risk factor of analyzing the drone images of affected areas using machine learning algorithm which overcomes the existing project. This system detects the fire conditions in a short time before any fire accidents spreads over the forest area. The scope of using video frames in the detection of fire using machine learning is challenging as well as innovative. If this system with less error rate can be implemented at a large scale like in big factories, houses, forests, it is possible to prevent damage and loss due to random fire accidents by making use of the Surveillance System.

12.Appendix

Github :

https://github.com/IBM-EPBL/IBM-Project-51482-1660979898