EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

A PROJECT REPORT

Submitted by

TEAM ID : PNT2022TMID03630

MEKALA BHARGAV (212219060161)

N. PAVAN (212219060195)

MUNJURU BHARADWAJA (212219060174)

NARIBOYINA PAVAN SAI (212219060183)

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

SAVEETHA ENGINEERRING COLLEGE, CHENNAI

INDEX

1. INTRODUCTION

Forest fires pose a serious threat to the environment because they harm the economy, the ecosystem, and put people in danger. In the United States, there are about 100,000 wildfires per year. Dangerous flames have burned more than 9 million acres (about half the area of South Carolina) of land. In a sparsely inhabited forest area, it is challenging to anticipate and detect forest fires, and it is even more challenging if the prediction is made using ground-based techniques like camera or video-based approaches. Due to their dependability and effectiveness, satellites can be a valuable source of data both before and during the Fire. the many methods for predicting and detecting forest fires in real-time, with the aim of informing the local fire authority.

Forests, which are diverse hotspots for flora and fauna and produce one-third of the world's oxygen, are susceptible to both natural and artificial forest fires. Many animals and the ecosystem can be saved by taking the precaution of preventing such a big, destructive flare. An effective way to return Mother Nature's unending gift is to protect forests before they are damaged.

One of the worst tragedies facing our civilization today, wildfires cause irreparable harm. These forest fires may have been started intentionally or accidentally by Mother Nature because of stormy winds or other weather conditions. These flames destroy numerous homes and pieces of property in addition to harming the environment.

1.1 Project Overview

Over the years, various strategies have been used to fight forest fires. They were primarily made to help in early fire detection. The simplest of these ideas is the creation of a network of observation posts, which is time-consuming for those engaged but is also inexpensive and simple to do. A new generation of techniques for early detection and even prevention of forest fires has been made possible by the ongoing development of information and communication technologies. In recent decades, networks of cameras and sensors based on ICT as well as satellite-based solutions were developed and used. Although the direct human involvement in the forest fire detection process has been significantly reduced by these solutions, they have also been shown to be expensive and difficult to maintain.

Throughout the year, there are forest fires, with the intensity peaking in the summer and fall. Although numerous natural and environmental occurrences, such as lightning strikes or the spontaneous combustion of dried leaves or sawdust, can also be blamed for these catastrophes, people are the primary cause of them. No matter what causes a forest fire to start, it usually results in terrible harm to both humans and the environment. Because each fire releases significant quantities of gases and particle matter into the sky, forest fires are also recognized as a major source of air pollution. Over the years, various strategies have been used to fight forest fires.

1.2 Purpose

The hazard posed by forest fires has increased globally, having a harmful impact on both human habitats and forest ecosystems. One result of such devastation is the greenhouse effect and changes to the climate. Interestingly, A greater proportion of forest fires are caused by human activity.

The project objective is to create a system for detecting forest fires in their earliest stages.

2. LITERATURE SURVEY

2.1 Existing problem

An estimated 340,000 early fatalities from respiratory and cardiovascular conditions occur each year and are linked to wildfire smoke.

Global biodiversity is under threat from wildfires because of their increased severity and frequency. Due to fires, people, businesses, and government entities all suffer significant financial losses. We must identify and put out forest fires at an early stage to reduce all of these.

Among the options now available to address this issue are:

Technology

Systems for detecting smoke and particles are available today and are frequently utilized in homes and businesses. These systems can decide if the environment is safe or whether an alarm must be set off by detecting moisture in an area. like how a fire alarm operates, which involves water being sprayed all over space to douse fires.

Fire Fighters

Highly skilled people are employed to deal with fire issues. In all weather conditions, firefighters put skills and trucks to use to put out forest fires.

A firefighter's top priority is to keep people safe and lessen the number of people who are hurt or killed by fire. Property damage and fighting fires come in second and third, respectively.

2.2 References

[1] Hristov, Georgi, et al. "Emerging methods for early detection of forest fires using unmanned aerial vehicles and lorawan sensor networks." *2018 28th EAEEIE annual conference (EAEEIE)*. IEEE, 2018.

[2] Guede-Fernández, Federico, et al. "A deep learning based object identification system for forest fire detection." *Fire* 4.4 (2021): 75.

[3] Wu, Di, et al. "Forest Fire Recognition Based on Feature Extraction from Multi-View Images." *Traitement du Signal* 38.3 (2021).

[4] Sun, Xiaofang, Liping Sun, and Yinglai Huang. "Forest fire smoke recognition based on convolutional neural network." *Journal of Forestry Research* 32.5 (2021): 1921-1927.

[5] Permana, Silvester Dian Handy, et al. "Classification of bird sounds as an early warning method of forest fires using Convolutional Neural Network (CNN) algorithm." *Journal of King Saud University-Computer and Information Sciences* 34.7 (2022): 4345-4357.

[6] Divya, A., T. Kavithanjali, and P. Dharshini. "IoT enabled forest fire detection and early warning system." *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*. IEEE, 2019.

[7] Son, Byungrak, Yong-sork Her, and J. Kim. "A design and implementation of forest-fires surveillance system based on wireless sensor networks for South Korea mountains." *International Journal of Computer Science and Network Security (IJCSNS)* 6.9 (2006): 124-130.

[8] Cao, Yichao, et al. "An attention enhanced bidirectional LSTM for early forest fire smoke recognition." *IEEE Access* 7 (2019): 154732-154742.

[9] Zhao, Yaqin, Zhong Zhou, and Mingming Xu. "Forest fire smoke video detection using spatiotemporal and dynamic texture features." *Journal of Electrical and Computer Engineering* 2015 (2015).

2.3 Problem Statement Definition

The technology to detect fire before it spreads into a large flame of destruction is described in the following paper:

1. To create a system that uses image processing to identify forest fires.

2. To detect fire more quickly by overcoming the physical and chemical dynamics.

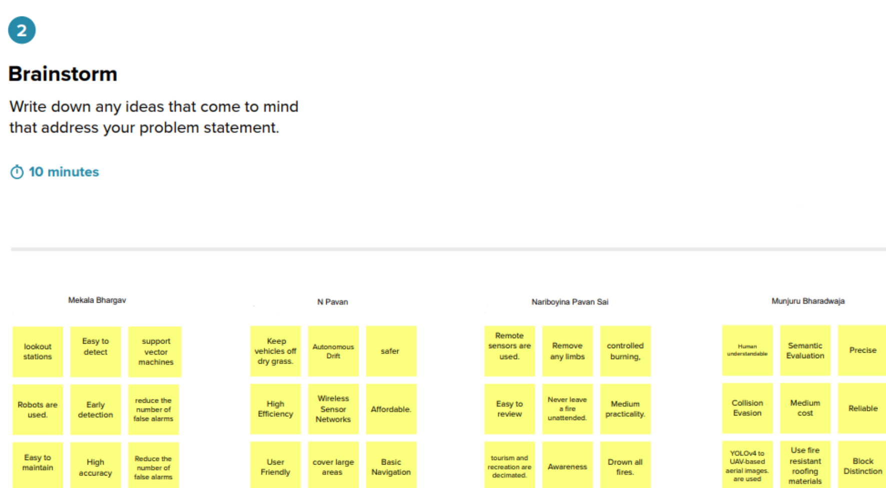
3. A model that will be trained using a variety of photos is being created using a typical neural network. To stop widespread destruction, this device will aid in the early detection of fire.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

Link:[Empathy map.pdf]

3.2 Ideation & Brainstorming



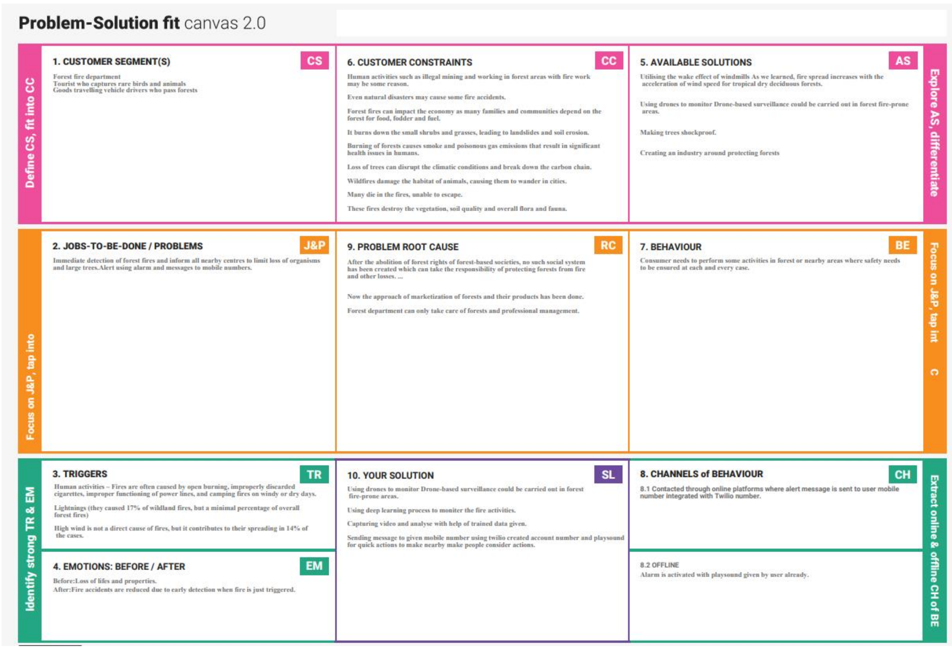
Link: [Brain Storming.pdf]

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Loss of valuable timber resources, Degradation of catchment areas, Loss of biodiversity and extinction of plants and animals, Loss of wildlife habitat and depletion of wildlife, Loss of natural regeneration and reduction in forest cover, Global warming,
2.	Idea / Solution description	Using cameras to monitor camera-based surveillance could be carried out in forest fire-prone areas. Utilising the wake effect of windmills As we learned, fire spread increases with the acceleration of wind speed for tropical dry deciduous forests.
3.	Novelty / Uniqueness	Use of latest algorithms and latest technologies to reduce and predict early detection of fires in forest.
4.	Social Impact / Customer Satisfaction	Even an small fire can cause large impact so evan an small action must be noticed to make customer satisfy.
5.	Business Model (Revenue Model)	The Fire Management Business Model underpins a **risk management model**. The model is used to calculate the probability of ignition and spread of fires across a landscape. This outcome allows for a better understanding of how changes in one aspect of management can affect other aspects of management.
6.	Scalability of the Solution	Capacity increased with increased training data and image pixel density and test datas. Increased camera capturing pixels makes better capture. Noticing small actions even.

Link:Proposed solution.pdf

3.4 Problem Solution fit



Link:[Problem solution fit.pdf]

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

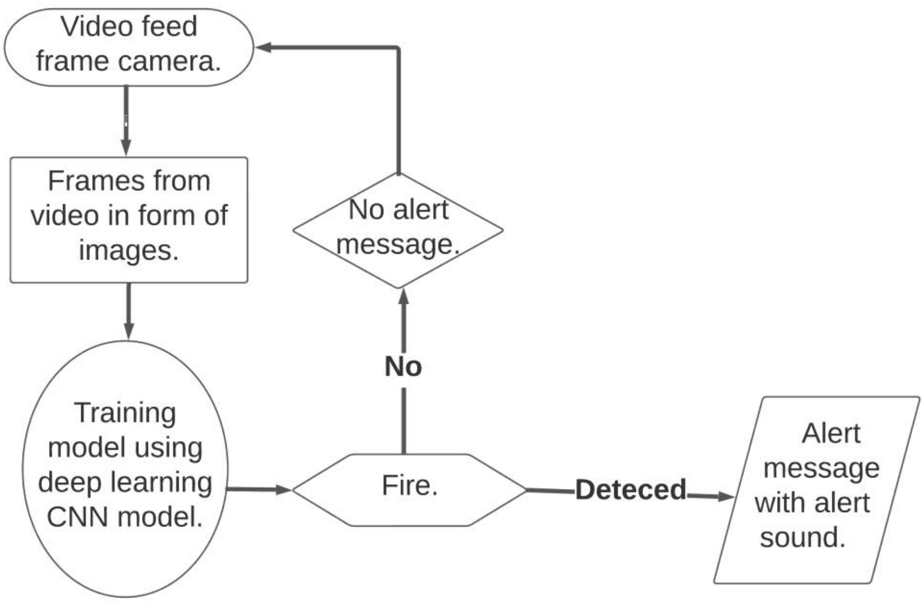
FR No	Functional Requirement	Description
1.	Importing files and software's required.	Import all required files from specific folders or packages.
2.	Image Recognition.	The system should be able to recognize images and analyze them.
3.	Monitoring forest area.	Area surrounded by forest should be observed every second throughout the day.
4.	Detection of danger.	Danger must be detected with perfectly trained from train dataset given.
5.	Alert message.	Giving alert message to the person who is responsible or forest officer immediately after detection.
4.	Operating system.	The system must require some parameters to run all these for better accuracy.

4.2 Non-Functional requirements

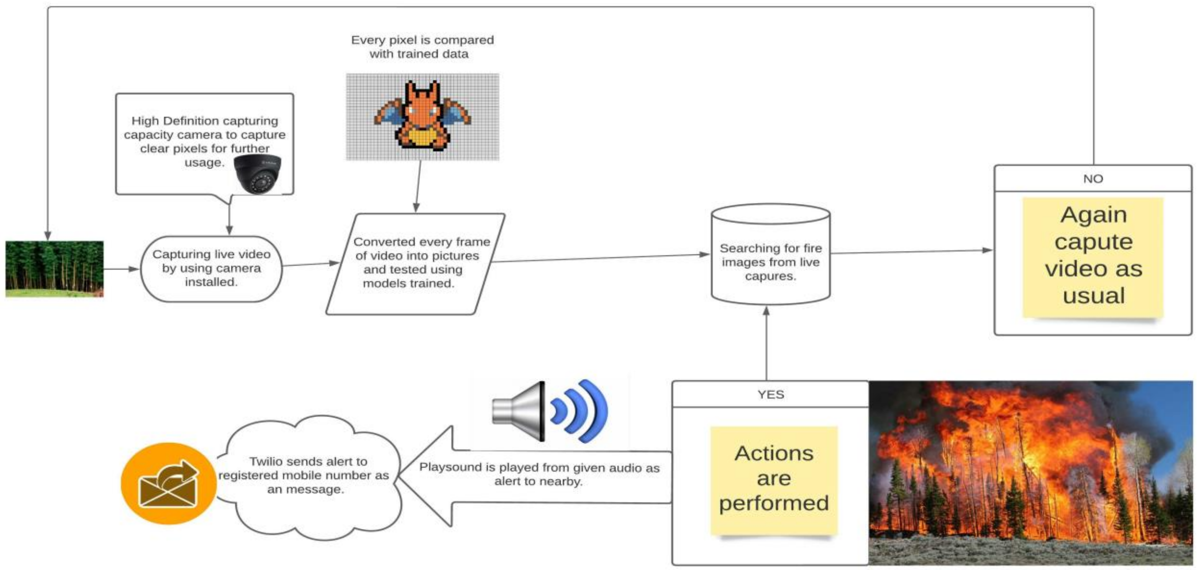
NFR No	Non-Functional Requirement	Description
1.	Usability	Users should find it easy to use and able to operate without any hassle.
2.	Security	A secure environment must be provided for better security.
3.	Reliability	Models should perform safe actions during installation.
4.	Performance	Achievement of high accuracy and efficiency is very important.
5.	Availability	Available all the time to the user.
6.	Scalability	Large data or small data must be processed with the same adaptability.
7.	Testability	More training data for training may help to achieve a more accurate level triggered.

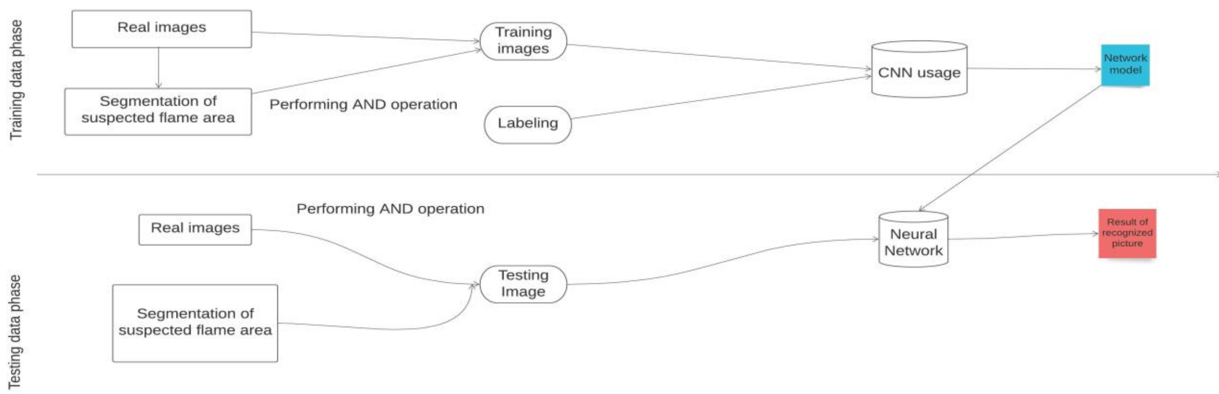
5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



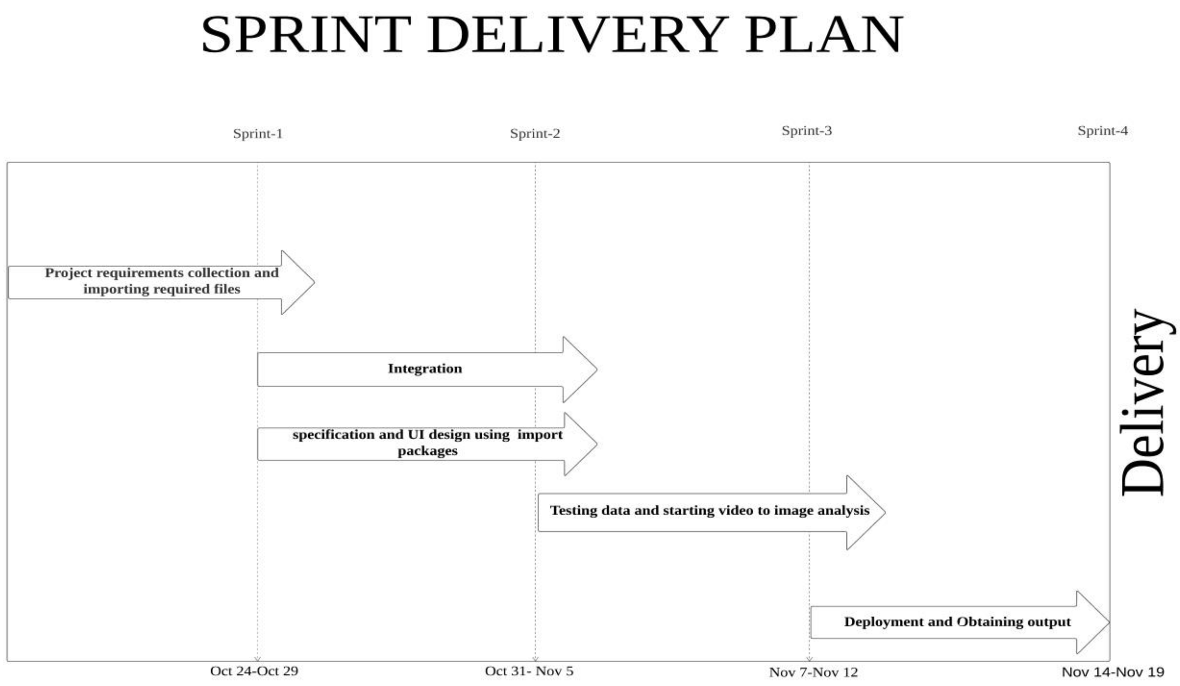


5.3 User Stories

User Type	Functional Requirement	User Story Number	User Story I Task	Acceptance criteria	Priority	Release
Environmenta list	Collect the data	USER-1	As an Environmentalist it is necessary to collect data of the forest which includes the temperature,humidity,wind and rain of the forest.	It is necessary to collect the right data else the prediction maybecome wrong.	High	Sprint-1
		USER-2	Identify algorithms that can be used for prediction	To collect the algorithm to identify the accuracy level of each algorithms	Medium	Sprint-2
	Implement Algorithm	USER-3	Identify the accuracy of each algorithms	Accuracy of each algorithm-calculated so that it is easy to obtain the most accurate output	High	Sprint-2
		USER-4	Evaluate the Dataset	Data is evaluated before processing	Medium	Sprint-1
	Evaluate Accuracy of Algorithm	USER-5	Identify accuracy,precision,recall of each algorithms	These values are important for obtaining the riaht output	High	Sprint-3
	Display Results	USER-6	Outputs from each algorithm are obtained	It is highly used to predict the effect and to High Sprint-4 take precautionary measures	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation



6.2 Sprint Delivery Schedule

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 October 2022	31 October 2022	20	30 October 2022
Sprint - 2	20	6 days	01 November 2022	10 November 2022	20	05 November 2022
Sprint - 3	20	6 days	11 November 2022	13 November 2022	20	10 November 2022
Sprint - 4	20	6 days	14 November 2022	19 November 2022	20	12 November 2022

MILESTONE LIST

Milestone Name	Milestone Number	Description	Mandatory
Project Objectives	M-01	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-02	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-03	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-04	One should have knowledge on the Supervised Learning ,CNN and Regression Classification and Clustering, ANN	Yes
Data collection	M-05	We can collect dataset from different open sources like kaggle.com, UCI machine learning etc	Yes
Image Preprocessing	M-06	Importing the ImageDataGenerator libraries, Define Parameters/Arguments for ImageDataGenerator class, Applying Image Data Generator Functionality to trainset and testset	Yes
Model Building	M-07	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-08	Opencv for video processing, creating an account in twilio service and sending alert message	Yes
Train CNN model	M-09	Register for IBM Cloud and train Image Classification Model	Yes
Ideation Phase	M-10	Prepare Literature Survey on the selected Project and Information Gathering, empathy map and ideation	Yes
Project Design Phase-I	M-11	Prepare Proposed solution , problem-solution fit and Solution Architecture	Yes
Project Design Phase-II	M-12	Prepare Customer journey ,functional requirements,Data flow diagram and Technology Architecture	Yes
Project Planning Phase	M-13	Prepare Milestone list , Activity list and Sprint Delivery Plan	Yes

Project DevelopmentPhase	M-14	Project Development delivery of Sprint 1, Sprint 2, Sprint 3, Sprint 4	Yes

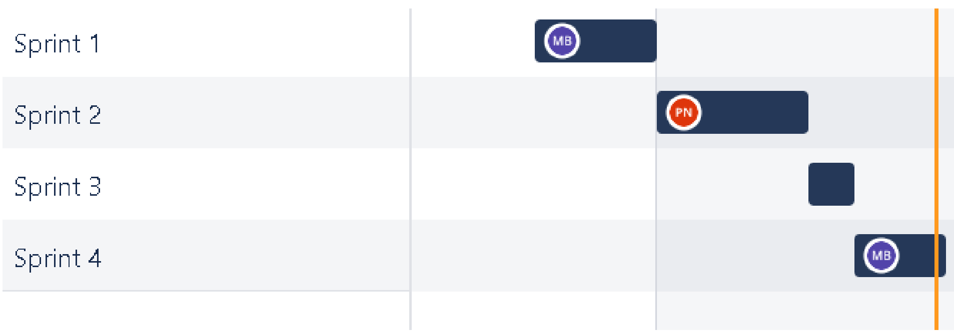
ACTIVITY LIST

Activity Number	Activity	Sub Activity	Assigned To	Status
1.	*PROJECT OBJECTIVES*		All Members	Completed
2.	*PROJECT FLOW*		All Members	Completed
3.	*PRE-REQUISITES*		All Members	Completed
4.	*DATA COLLECTION*	4.1 Download the Dataset	Mekala Bhargav	Completed
5.	*IMAGE PREPROCESSING*	5.1 Import the ImageDataGenerator Library. 5.2 Define the Parameters/Arguments for ImageDataGenerator class. 5.3 Applying ImageDataGenerator Functionality to trainset and testset.	Mekala Bhargav	Completed
6.	*MODEL BUILDING*	6.1 Importing the model building libraries. 6.2 Initializing the model.	Mekala Bhargav	Completed

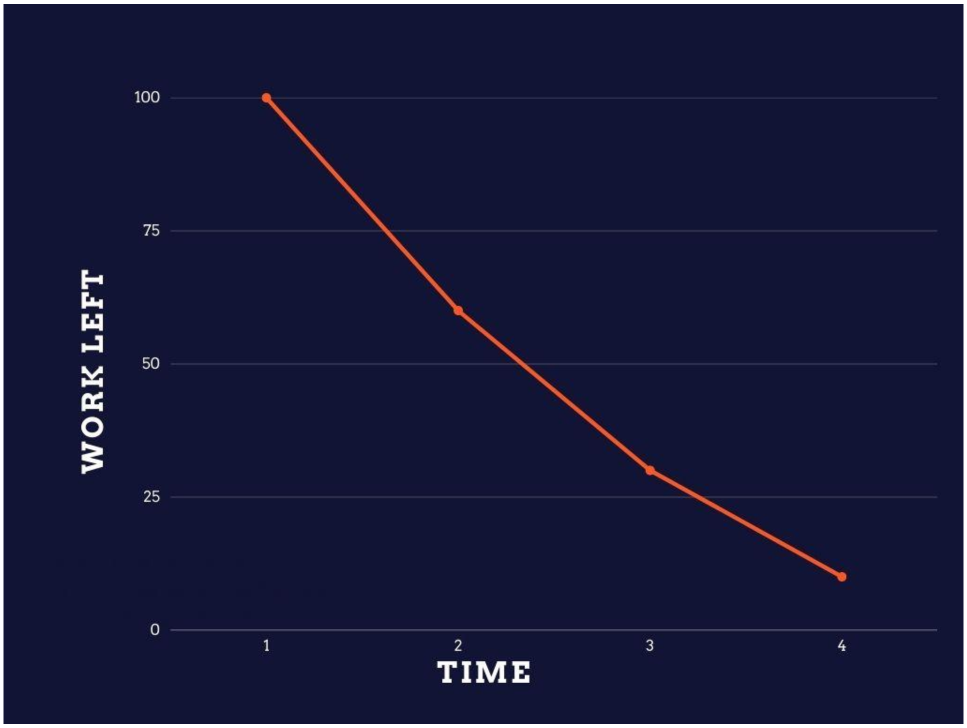
		6.3 Adding CNN layers. 6.4 Adding dense layers. 6.5 Configuring the learning process. 6.6 Training the model. 6.7 Saving the model. 6.8 Predictions.		
7.	*VIDEO ANALYSIS*	7.1 OpenCV for video processing. 7.2 Creating an account in Twilio service. 7.3 Sending alert message.	Mekala Bhargav	Completed
8.	*TRAIN CNN MODEL ON IBM*	8.1 Train image classification model. 8.2 Register for IBM cloud.	Mekala Bhargav	In Progress
9.	*IDEATION PHASE*	9.1 Literature Review. 9.2 Empathy map. 9.3 Ideation.	All Members	Completed
10.	*PROJECT DESIGN PHASE – I*	10.1 Proposed Solution. 10.2 Problem solution fit. 10.3 Solution Architecture.	All Members	Completed
11.	*PROJECT DESIGN PHASE -II*	11.1 Customer journey. 11.2 Functional requirement. 11.3 Data flow Diagrams. 11.4 Technology Architecture.	All Members	Completed

12.	*PROJECT PLANNING PHASE*	12.1 Prepare milestone and activity list. 12.2 Sprint delivery plan.	All Members	Completed
13.	*PROJECT DEVELOPMENT PHASE*	13.1 Project Development-Delivery of Sprint-1. 13.2 Project Development-Delivery of Sprint-2. 13.3 Project Development-Delivery of Sprint-3. 13.4 Project Development-Delivery of Sprint-4.	All Members	Completed

6.3 JIRA



Burn graph


WORK LEFT
100
75
50
25
0
1
2
3
4
TIME

7. CODING & SOLUTIONING

7.1 FEATURE 1

IMAGE DATA GENERATOR

Keras ImageDataGenerator is used for getting the input of the original data and further, it makes the transformation of this data on a random basis and gives the output resultant containing only the data that is newly transformed. It does not add the data.

from keras.preprocessing.image import ImageDataGenerator

SHEAR RANGE

Shear range means that the image will be distorted along an axis, mostly to create or rectify the perception angles. It's usually used to augment images so that computers can see how humans see things from different angles.

ZOOM RANGE

The zoom augmentation method is used to zoom the image. This method randomly zooms the image either by zooming in or it adds some pixels around the image to enlarge the image. This method uses the zoom range argument of the ImageDataGenerator class. We can specify the percentage value of the zooms in a float, range in the form of an array.

ROTATION RANGE

ImageDataGenerator class allows you to randomly rotate images through any degree between 0 and 360 by providing an integer value in the rotation range argument. When the image is rotated, some pixels will move outside the image and leave an empty area that needs to be filled in.

RESCALE

The ImageDataGenerator class can be used to rescale pixel values from the range of 0-255 to the range 0-1 preferred for neural network models. Scaling data to the range of 0-1 is traditionally referred to as normalization.

HORIZONTAL FLIP

The horizontal flip basically flips both rows and columns horizontally. So, for this, we must pass the horizontal flip=True argument in the ImageDataGenerator constructor.

CONVOLUTION NEURAL NETWORK

A CNN is a kind of network architecture for deep learning algorithms and is specifically used for image recognition and tasks that involve the processing of pixel data. There are other types of neural networks in deep learning, but for identifying and recognizing objects, CNNs are the network architecture of choice. The layers used in the CNN algorithm is Convolutional, maxpooling, and flatten layer.

1. Convolutional Layer

A convolutional layer is the main building block of a CNN. It contains a set of filters (or kernels), parameters of which are to be learned throughout the training. The size of the filters is usually smaller than the actual image. Each filter convolves with the image

Convolution layer is used for a image processing to blur and sharpen images, but also to perform other operations.

from keras.layers import Convolution2D

2. Maxpooling Layer:

Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter.

from keras.layers import MaxPooling2D

3. Flatten Layer:

Flattening is used to convert all the resultant 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector. The flattened matrix is fed as input to the fully connected layer to classify the image.

from keras.layers import Flatten

4. DENSE LAYER:

Dense Layer is used to classify image based on output from convolutional layers.

7.2 FEATURE 2

IMPORTING ALL REQUIRED LIBRARY FILES

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 keras

from keras.preprocessing.image import ImageDataGenerator

DEFINE THE PARAMETERS TO TRAIN AND TEST AND APPLYING IMAGEDATAGENERATOR FUNCTIONALITY TO TRAIN DATASET

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_train=train_datagen.flow_from_directory(r'C:\Users\green\Downloads\Dataset\Dataset\train_set',target_size=(128,128),batch_size=32,class_mode='binary')

x_test=test_datagen.flow_from_directory(r'C:\Users\green\Downloads\Dataset\Dataset\test_set',target_size=(128,128),batch_size=32,class_mode='binary')

```
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)
```

IMPORTING MODEL BUILDING LIBRARIES

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')

INITIALIZING THE MODEL

model = keras.Sequential()

ADDING CNN LAYERS

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

#add maxpooling layers

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

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

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

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

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

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

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

model.add(Flatten())

#add faltten layer

ADD DENSE LAYERS

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

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

CONFIGURING THE LEARNING PROCESS

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

TRAINING THE MODEL AND PREDICTIONS

```
r = model.fit(x_train, epochs = 10, validation_data = x_train)
predictions = model.predict(x_train)
predictions = np.round(predictions)
Predictions
```

SAVING THE MODEL

```
model.save("forest1.h5")
```

LOADING MODEL TO CAPTURE VIDEO AND CONVERT IT INTO FRAMES

```
import cv2
#import facevec
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
from twilio.rest import Client
# import Twilio library for sending messages.
from playsound import playsound
# playsound for surrounding alert message in form of alarm or given sound by user.
model = load_model(r"forest1.h5")
video=cv2.VideoCapture(0)
```

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



model = load_model(r'forestfire.h5')
video = cv2.VideoCapture(0)
# Cpaturing video from external camera.
name = ['forest','with fire']

while(1):
    success, frame = video.read()
    cv2.imwrite("image.jpg",frame)
    img = image.load_img("image.jpg",target_size = (128,128))
    x  = image.img_to_array(img)
    x = np.expand_dims(x,axis = 0)
    pred = model.predict(x)
    classes_x=np.argmax(pred,axis=1)
    p = pred[0][0]
    print(pred)
    cv2.putText(frame, "predicted  class = "+str(name[p]), (100,100),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,0), 1)
    # Predicting each frame for for fire.

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

    if pred[0]==1:
# Details of Twilio account for sending message.
        account_sid = 'AC8d01fbXXXXXXXXXXXX’
# Exposing account_token will deactivate and change token of current one.
```

```
        auth_token = 'ea25472d4XXXXXXXXXXXX'
        client = Client(account_sid, auth_token)

        message = client.messages \
        .create(
         body='Forest Fire is detected, stay alert',
         from_='+1385XXXXXX', #twilio free number
         to='+919550630271' # User Number)
        print(message.sid)

        print('Fire Detected')
        print ('SMS sent!')
Playsound(r "C:\Users\green\Downloads\fire alarm.mp3")
        break
# If fire found then send message from Twilio account to registered mobile number
    else:
        print("no danger")
       #break
    cv2.imshow("image",frame)
# If no fire found then again searching for fire from upcoming frames

    if cv2.waitKey(1) & 0xFF == ord('a'):
        break

video.release()
cv2.destroyAllWindows()
# If fire alert message sent then close the window.
```

8. TESTING

8.1 Test Cases

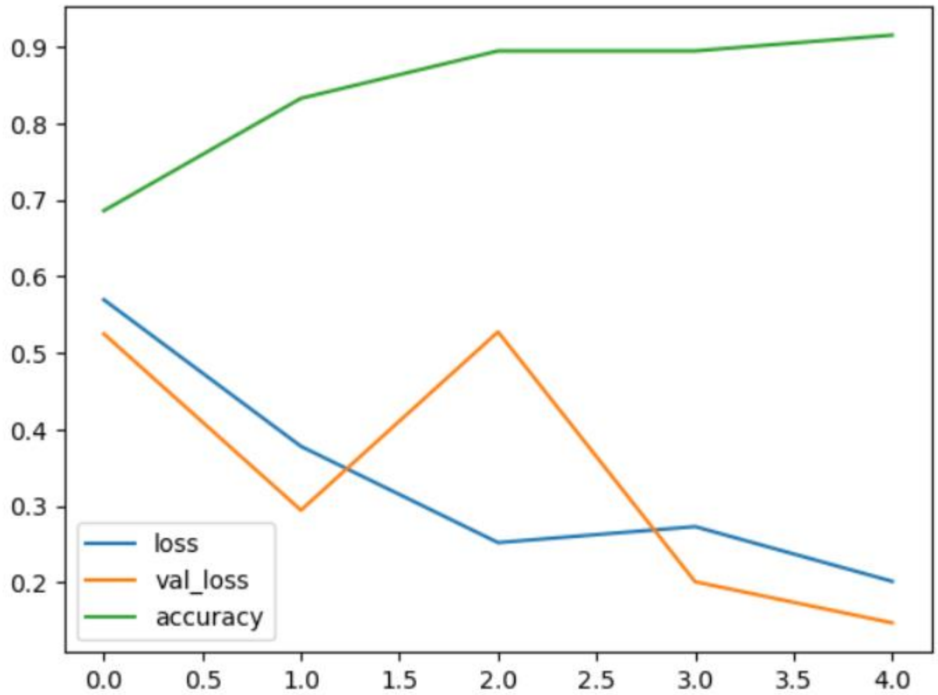
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	30	0	0	30
Security	2	0	0	2
Out source Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

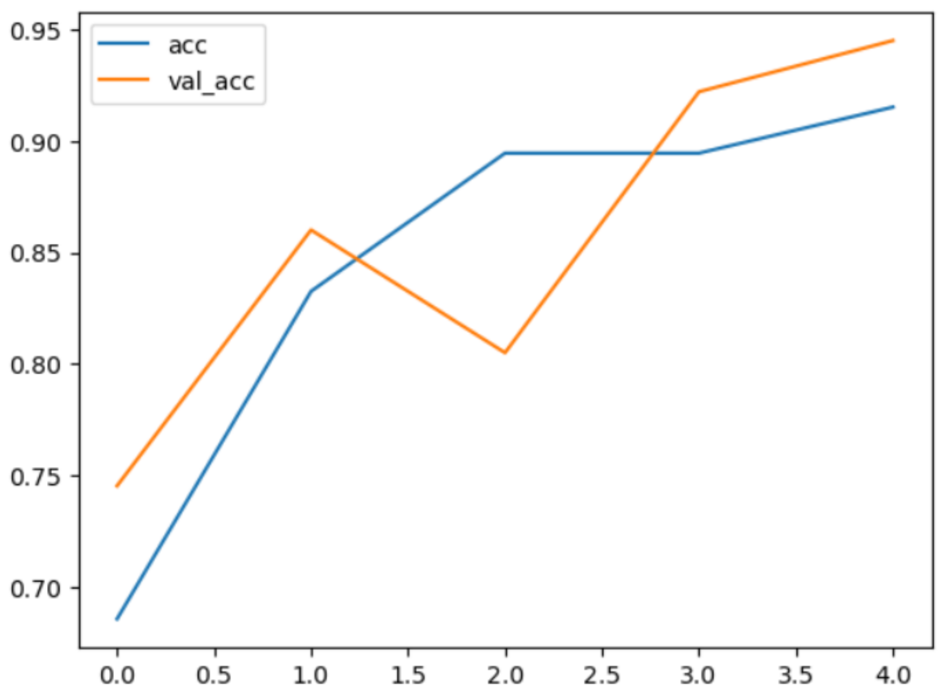
8.2 User Acceptance Testing

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	5	1	1	1	8
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	7	2	4	10	23
Not Reproduced	0	0	0	0	0
Skipped	0	0	1	1	2
Won'tFix	0	3	2	1	6
Totals	15	9	11	14	49

9. RESULTS

9.1 Performance Metrics





S.No.	Parameter	Values

1.	Model Summary	As the threat of forest fire increases due to climate changes, the need for finding a detection system increases. We proposed a Deep Learning-based model for early detection of forest fire. The Proposed model successfully classifies the images into fire and no fire and sends an alert message in case of fire. Thus, the Deep Learning algorithms proved their efficiency in detecting different objects.
2.	Accuracy	Training Accuracy - 91% - 98% Validation Accuracy - 94%

10. ADVANTAGES & DISADVANTAGES

Advantages:

1. Minimal loss.
2. Accuracy increased.
3. Fast alertness when compared to manual observation.
4. No man needed for observation all the time.
5. Fast recovery can be maintained.

Disadvantages:

1. Cost increased.
2. Maintenance required.
3. Wrong detection may require more time to train with high datasets.

11. CONCLUSION

The CNN facility can be used for a variety of purposes. CNN is the foundation of a real-time forest fire monitoring system. For users to log in and access data, a camera module is connected to obtain real-time data. The rate of fire detection is maximized thanks to real-time pollutant monitoring. Due to the low cost of sensors and microcontrollers, implementation costs are low. Future upgrades to the camera could include image processing algorithms that would allow for the creation of a completely automated system for safety and widespread surveillance. In the absence of the authorized user, this makes it possible to generate automatic control. As the threat of forest fire increases due to climate changes, the need for finding a detection system increases. We proposed a Deep Learning-based model for early detection of forest fire. The Proposed model successfully classifies the images into fire and no fire and sends an alert message in case of fire. Thus, the Deep Learning algorithms proved their efficiency in detecting different objects.

12. **FUTURE SCOPE**

- Compile real-time satellite data and handle the flames in real-time.
- To increase speed, increase the temporal complexity of forest fire detection.
- These accidents are more easily under control.
- Forest fires result in the extinction of an abundance of species; by employing this strategy, we can protect both human life and the ecosystem.

13. APPENDIX

Source Code

Link for source code

GitHub & Project Demo Link

Project GitHub link