EMERGING METHOD FOR EARLY DETECTION OF FOREST FIRE

Submitted by

PREETHI R(821919104015)

KAVITHA M(821919104011)

SRIMATHI G(821919104026)

ANUSHTI K(821919104004)

PRIYA M(821919104017)

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

ST.JOSEPH'S COLLEGE OF ENGINEERING AND TECHNOLOGY

THANJAVUR

ANNA UNIVERSITY: CHENNAI 600 025

TEAM ID: PNT2022TMID46933

TABLE OF CONTENTS

1. INTRODUCTION

1.1 PROJECT OVERVIEW

From sprawling urbans to dense jungles, fire accidents pose a major threat to the world. These could be prevented by deploying fire detection systems, but the prohibitive cost, false alarms, need for dedicated infrastructure, and the overall lack of robustness of the present hardware and software-based detection systems have served as roadblocks in this direction. In this work, we endeavor to make a stride towards detection of fire in videos using Deep learning. Deep learning is an emerging concept based on artificial neural networks and has achieved exceptional results in various fields including computer vision. We plan to overcome the shortcomings of the present systems and provide an accurate and precise system to detect fires as early as possible and capable of working in various environments thereby saving innumerable lives and resources.

1.2 PURPOSE

The proposed framework utilizes the advantages of a convolutional neural network. The CNN receives input, it is preprocessed and pools them using region of proposals. Then the region-based object detection algorithm in CNN classifies those proposals into fire and non-fire in the region of interest (ROI) with the help of convolutional layers.

2. LITERATURE SURVEY

2.1 Existing 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 a fire is to be cautious al 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.

2.2 References

[1]. Janku P., Kominkova Oplatkova Z., Dulik T., Snopek P. and Liba J. 2018. “Fire Detection in Video Stream by Using Simple Artificial Neural”. Network. MENDEL. 24, 2 (Dec. 2018), 55–60.

[2]. Shen, D., Chen, X., Nguyen, M., & Yan, W. Q. (2018). “Flame detection using deep learning”. 2018 4th International Conference on Control, Automation and Robotics (ICCAR).

[3]. Li, C., & Bai, Y. (2018). “Fire Flame Image Detection Based on Transfer Learning”. 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS).

[4]. K. Muhammad, J. Ahmad, I. Mehmood, S. Rho and S. W. Baik, “Convolutional Neural Networks Based Fire Detection in Surveillance Videos," in IEEE Access, vol. 6, pp. 18174-18183, 2018.

[5]. K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang and S. W. Baik, “Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications,” in IEEE Transactions on Systems,Man, and Cybernetics: Systems, vol. 49, no. 7, pp. 1419-1434, July 2019.

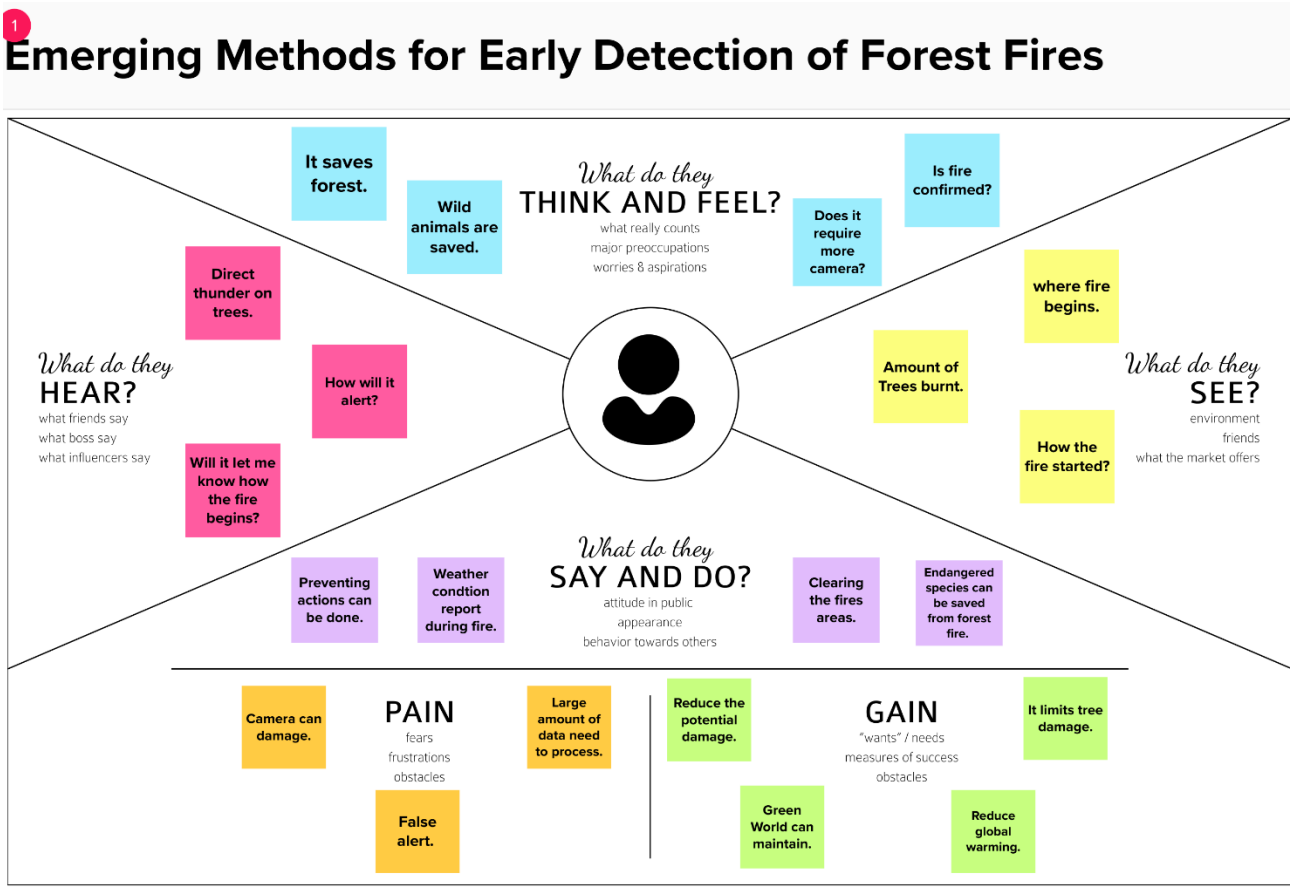
2.3 Problem Statement Definition

QUESTION	DESCRIPTION
What is the problem?	loss of biodiversity and extinction of plants and animals.
what is problem occurs?	Nature causes - Many forest fires start from natural causes such as lighting which set trees on fire.
How to control forest fires?	Which can be done by creating firebreaks in the shape of small clearings of ditches in the forests.
When issue will be identified?	Existing detection methods such as satellite and optical systems can cover large areas.
Why we need to fix problem?	For preventing and controlling a forest fires.

3.IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas

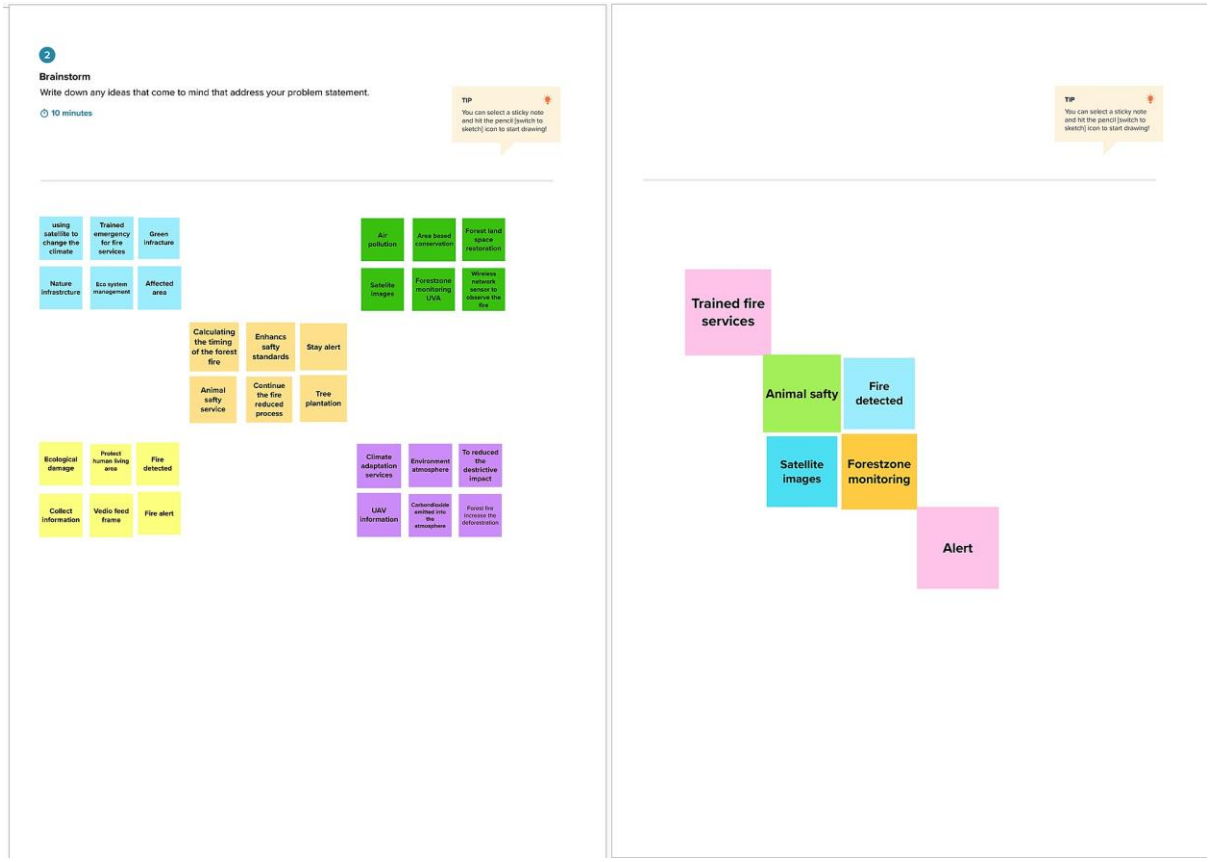
This map is created with view of the project in user's perspective, to find pain & gain points and to summarize it with a list of problem statements.



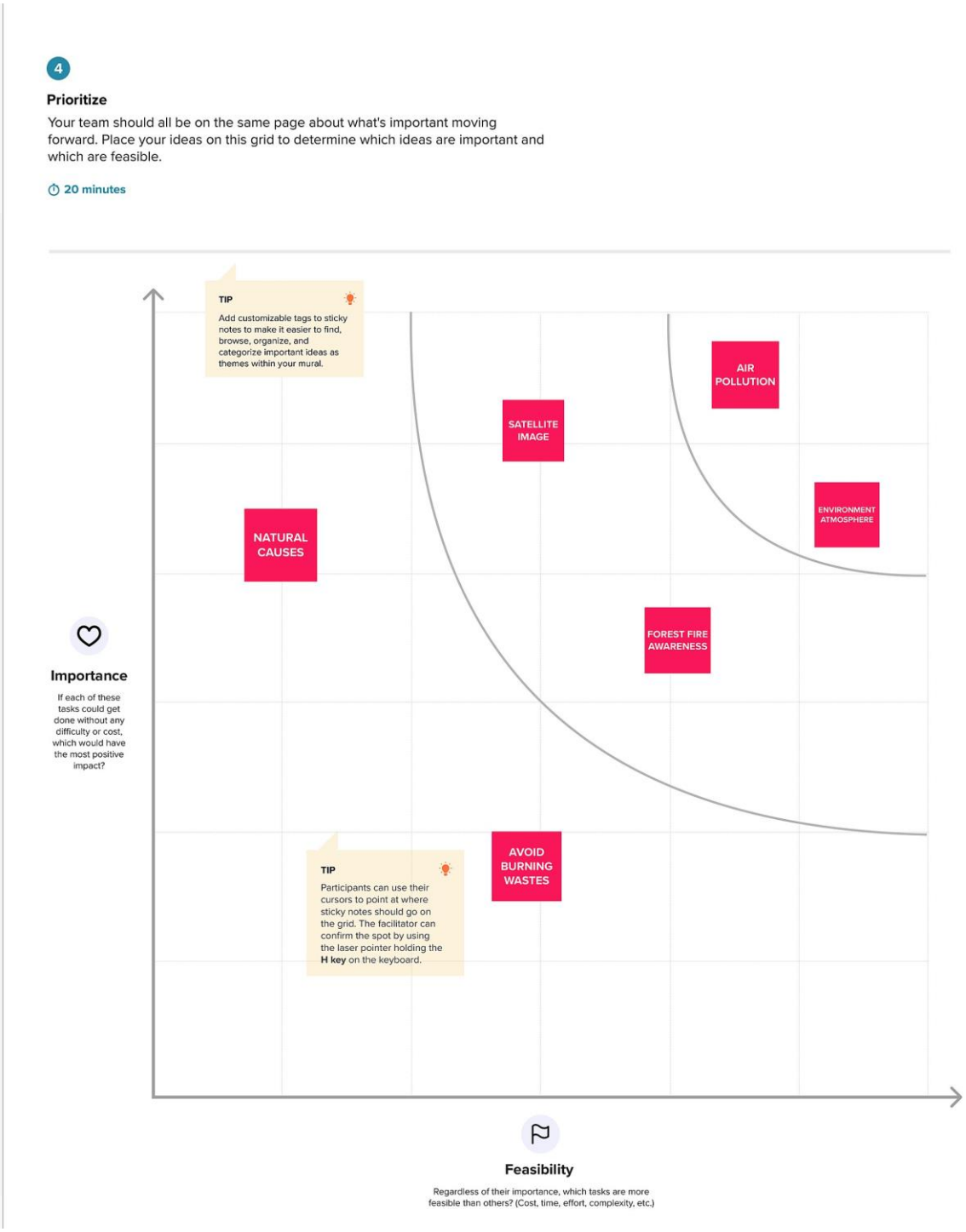
3.2 IDEATION & BRAINSTORMING

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Step-1: Brainstorm,Idea,Listing and Grouping



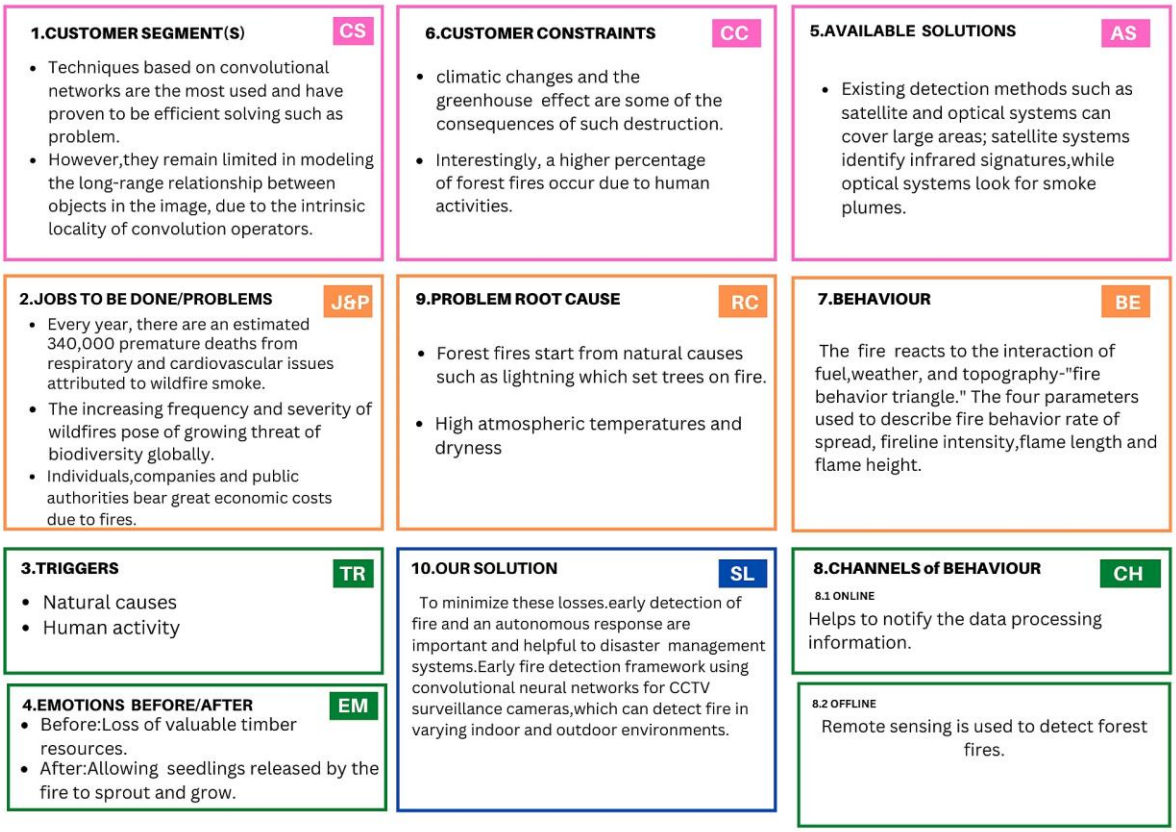
Step-2: Idea Prioritization



3.3 PROPOSED SOLUTION

S.NO	Parameter	Description
1.	Problem Statement(Problem to be solved)	Loss of biodiversity and extinction of plants and animals. Loss of wild life habitat. Loss of natural regeneration and reduction in forest cover,global warming.
2.	Idea /Solution description	Use fire pits in territories protected by the department of natural resources. prepare a bucket of water and a shovle to extinguish the bonfire.
3.	Novelty /Uniqueness	Fire detection systems increase response times, as they are able to alert the correct people in order to extinguish the fire.
4.	Social impact / Customer Satisfaction	Monitoring of the potential risk areas and an early detection of forest fires can significantly shorten the reaction time.
5.	Business Model(Revenue Model)	Due to various shapes,textures and colors of fires,forest fire detection is challenging task.
6.	Scalability of the Solution	Using a coupled multi-physics system to predict the evolution of a forest fires is the ability of capturing the effect of meteorological events .

3.4 PROBLEM SOLUTION FIT



4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR No-1	Video surveillance start	Start surveillance through remote control
FR No-2	Forest monitoring	Continuous monitoring through camera
FR No-3	Detect fire	Fire is detected through CNN model
FR No-4	Alert	Alert the forest officials through message

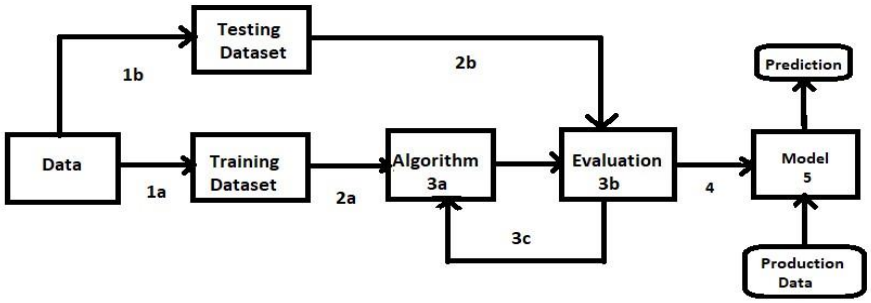
4.2 NON-FUNCTIONAL REQUIREMENTS

NFR No.	Non-Functional Requirement	Description
NFR No-1	Usability	Monitoring of the potential risk areas and an early detection of fire can significantly shorten the reaction time and also reduce the potential damage as well as the cost of fire fighting.
NFR No-2	Security	More secure environment.
NFR No-3	Reliability	Model is safe to install.
NFR No-4	Performance	Model will achieve high accuracy.
NFR No-5	Availability	Build model is available all the time.
NFR No-6	Scalability	The current requirement for a cargo compartment detection system is that a fire has to be detected in 1 minute, and in that time be so small that the fire is not a significant hazard to the airplane. Nuisance alarms also plague the industry, with upwards of 90% of fire alarms being false warnings.

5.PROJECT DESIGN

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.



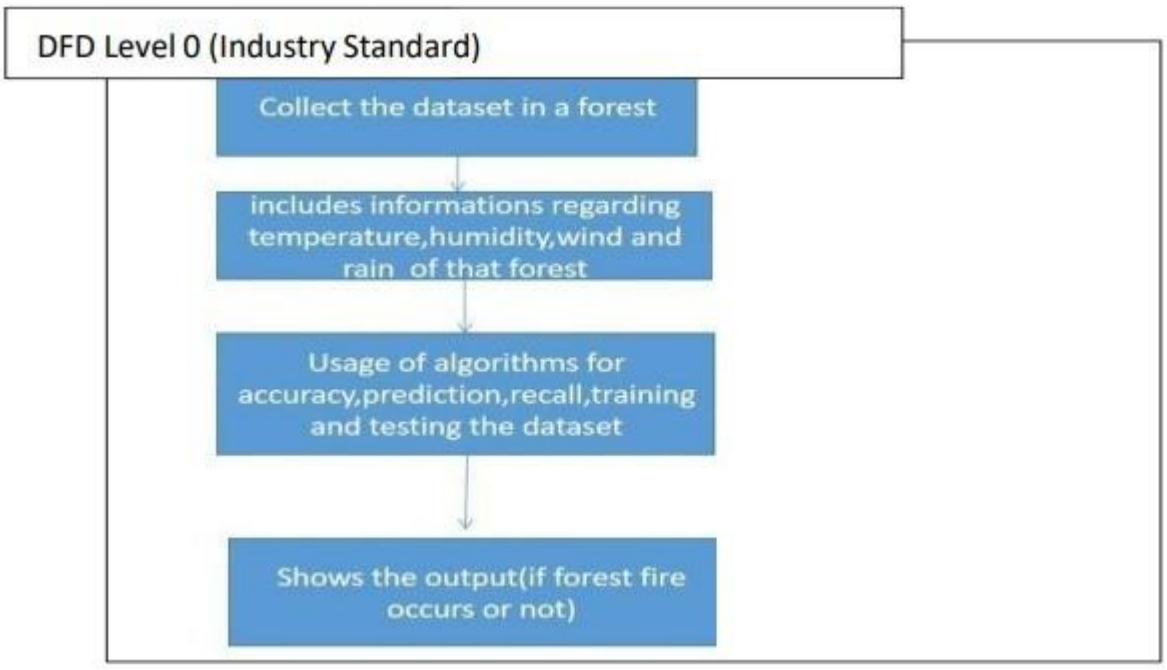
1.Data set

2.Evaluate Dataset

3.Implement Algorithm

4.Evaluate accuracy of algorithms

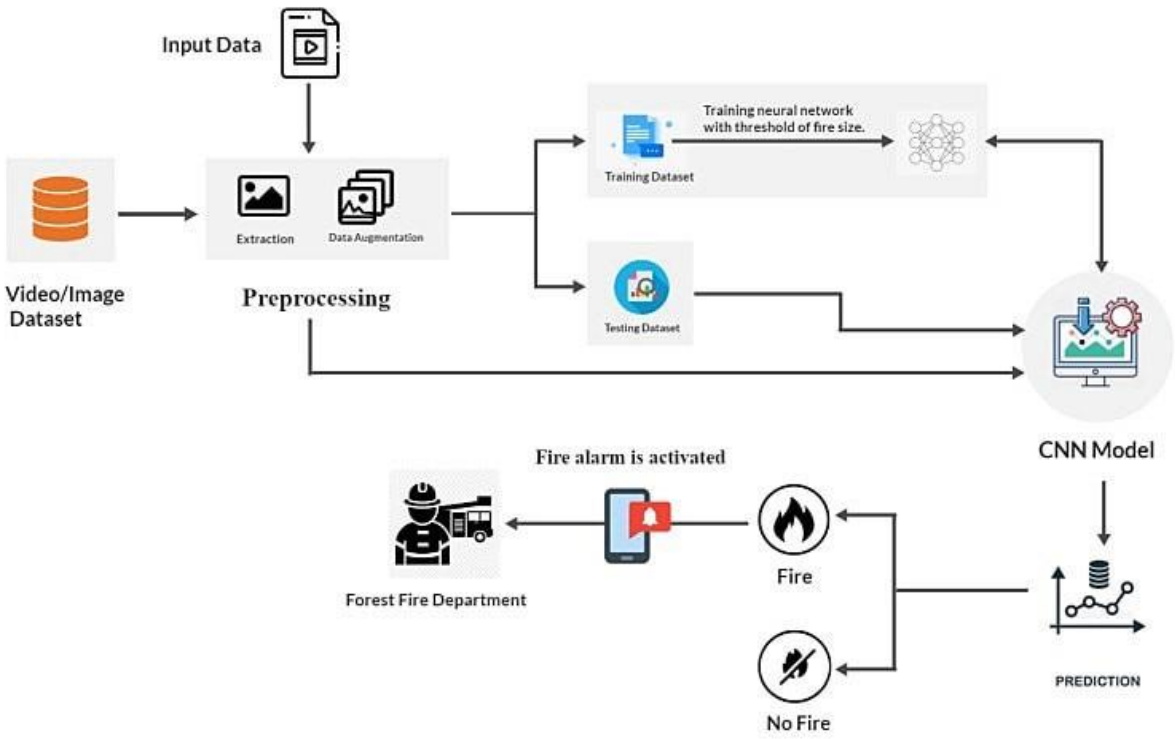
5. Display Results



5.2 SOLUTION & TECHNICAL ARCHITECTURE

Solution Architecture is a complex process-with many sub-process-that bridges the gap between business problem and technologies solutions; It goals are to.

- Find the best tech solutions to solve existing business problems.
- Describe the structure, characteristics, behavior and the aspects of the software to project stakeholders.
- Define the features, development phases and solution requirements.
- Provide specifications according to which the solution is defined, managed and delivered.



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
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 wrong	High	Sprint-1
		USN-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	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	Sprint-2
		USN-4	Evaluate the Dataset	Data is evaluated before processing	Medium	Sprint-1
	Evaluate Accuracy of Algorithm	USN-5	Identify accuracy,precision,recall of each algorithms	These values are important for obtaining the right output	High	Sprint-3
	Display Results	USN-6	Outputs from each algorithm are obtained	It is highly used to predict the effect and to take precautionary measures.	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

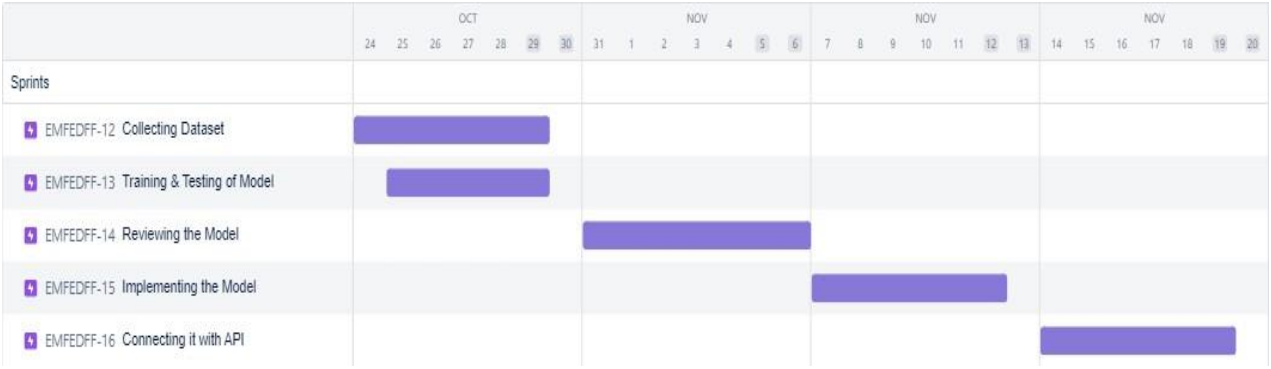
6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	3	Medium	Preethi R, Kavitha M Priya M, Srimathi G
Sprint-2		USN-2	As a user, I will receive confirmation email once I have registered for the application	2	Low	Preethi R, Kavitha M Anushti K, Srimathi G
Sprint-3		USN-3	As a user, I can register for the application through Facebook	2	Low	Preethi R, Kavitha M
Sprint-3		USN-4	As a user, I can register for the application through Gmail	3	Medium	Preethi R, Kavitha M Priya M
Sprint-2	Login	USN-5	As a user, I can log into the application by entering email & password	3	Medium	Anushti K, Srimathi G,Preethi R
Sprint -1	Dataset	USN-6	The dataset is collected and pre-processed and split for training and testing.	5	High	Preethi R, Kavitha M Anushti K, Srimathi G Priya M
Sprint -1		USN-7	The model is created and trained using test and train dataset.	5	High	Preethi R, Kavitha M

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 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

```
[5]  #Importing the ImageDataGenerator Library
```

```
[6]  from google.colab import drive
     drive.mount('/content/drive')
```

```
#Data Collection
!unzip '/content/drive/MyDrive/archive.zip'
```

```
[8]  import keras
     from keras.preprocessing.image import ImageDataGenerator
```

```
[9]  #Define the parameters/arguments for ImageDataGenerator class
```

```
[10] train_datagen = ImageDataGenerator(
         rotation_range=180,
         width_shift_range=0.2,
         height_shift_range=0.2,
         rescale=1./255,
         shear_range=0.2,
         zoom_range=0.2,
         horizontal_flip=True,)
     test_datagen = ImageDataGenerator(rescale=1./255)
```

```
#Applying ImageDataGenerator functionality to trainset
```

```
[12] x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/Dataset/Dataset/train_set',
     target_size=(128,128),batch_size=32,class_mode='binary')
```

```
Found 436 images belonging to 2 classes.
```

```
[13] #Applying ImageDataGenerator functionality to testset
```

```
[14] x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/Dataset/Dataset/test_set',
     target_size=(128,128),batch_size=32,class_mode='binary')
```

```
Found 118 images belonging to 2 classes.
```

```
[15] #importing the model building libraries
```

```
[16] #To define linear intialisation import Sequential
     from keras.models import Sequential
     #To add layers import Dense
     from keras.layers import Dense
     #To creat Convolution kernal 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')
```

```
[17] #initializing the model
     model=Sequential()
```

```
[18] #Adding CNN layers
```

```
[19] #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())
```

```
[20] #Adding Dense Layers
```

```
[21] #add hidden layer
     model.add(Dense(150,activation='relu'))
     #add output Layer
     model.add(Dense(1,activation='sigmoid'))
```

```
[22] #configuring the learning process
     model.compile(loss='binary_crossentropy',optimizer="adam",metrics=["accuracy"])
```

```
#Training the model
model.fit_generator(x_train,steps_per_epoch=14,epochs=10,validation_data=x_test,validation_steps=4)
```

```
Epoch 1/10
14/14 [==============================] - 146s 10s/step - loss: 2.5047 - accuracy: 0.6651 - val_loss: 0.1616 - val_accuracy: 0.9237
Epoch 2/10
14/14 [==============================] - 24s 2s/step - loss: 0.3441 - accuracy: 0.8532 - val_loss: 0.1535 - val_accuracy: 0.9237
Epoch 3/10
14/14 [==============================] - 24s 2s/step - loss: 0.2516 - accuracy: 0.8922 - val_loss: 0.1019 - val_accuracy: 0.9746
Epoch 4/10
14/14 [==============================] - 24s 2s/step - loss: 0.2102 - accuracy: 0.9014 - val_loss: 0.1432 - val_accuracy: 0.9576
Epoch 5/10
14/14 [==============================] - 24s 2s/step - loss: 0.2267 - accuracy: 0.9083 - val_loss: 0.0876 - val_accuracy: 0.9746
Epoch 6/10
14/14 [==============================] - 24s 2s/step - loss: 0.2030 - accuracy: 0.9174 - val_loss: 0.0914 - val_accuracy: 0.9576
Epoch 7/10
14/14 [==============================] - 24s 2s/step - loss: 0.2092 - accuracy: 0.9083 - val_loss: 0.1128 - val_accuracy: 0.9322
Epoch 8/10
14/14 [==============================] - 25s 2s/step - loss: 0.1865 - accuracy: 0.9266 - val_loss: 0.0819 - val_accuracy: 0.9661
Epoch 9/10
14/14 [==============================] - 24s 2s/step - loss: 0.1992 - accuracy: 0.8991 - val_loss: 0.0709 - val_accuracy: 0.9746
Epoch 10/10
14/14 [==============================] - 24s 2s/step - loss: 0.1833 - accuracy: 0.9312 - val_loss: 0.0630 - val_accuracy: 0.9915
<keras.callbacks.History at 0x7f1fef024e10>
```

```
[24] #save the model
     model.save("forest1.h5")
```

```
[25] #predictions
```

```
#import load_model from keras.model
from keras.models import load_model
#import image class from keras
from tensorflow.keras.preprocessing import image
#import numpy
import numpy as np
#import cv2
import cv2
```

```
[27] #load the saved model
     model = load_model("forest1.h5")
```

```
[28] img=image.load_img('/content/drive/MyDrive/Dataset/Dataset/test_set/with fire/180802_CarrFire_010_large_700x467.jpg')
     x=image.img_to_array(img)
     res = cv2.resize(x, dsize=(128,128 ), interpolation=cv2.INTER_CUBIC)
     #expand the image shape
     x=np.expand_dims(res,axis=0)
```

```
pred=model.predict(x)
```

```
1/1 [==============================] - 0s 101ms/step
```

```
[30] pred
```

```
array([[1.]], dtype=float32)
```

7.2 Feature

```
[31] #Opencv for video processing
```

```
pip install twilio
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting twilio
  Downloading twilio-7.15.3-py2.py3-none-any.whl (1.4 MB)
     |████████████████████████████████| 1.4 MB 7.5 MB/s
Requirement already satisfied: requests>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from twilio) (2.23.0)
Collecting PyJWT<3.0.0,>=2.0.0
  Downloading PyJWT-2.6.0-py3-none-any.whl (20 kB)
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from twilio) (2022.6)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2022.9.24)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (1.24.3)
Installing collected packages: PyJWT, twilio
Successfully installed PyJWT-2.6.0 twilio-7.15.3
```

```
[33] pip install playsound
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting playsound
  Downloading playsound-1.3.0.tar.gz (7.7 kB)
Building wheels for collected packages: playsound
  Building wheel for playsound (setup.py) ... done
  Created wheel for playsound: filename=playsound-1.3.0-py3-none-any.whl size=7038 sha256=76d0a81ba4283ad317e276bb2c86bba7e5a93cc513dc5db66ba5f39b5f672f2b
  Stored in directory: /root/.cache/pip/wheels/ba/f8/bb/ea57c0146b664dca3a0ada4199b0ecb5f9dfcb7b7e22b65ba2
Successfully built playsound
Installing collected packages: playsound
Successfully installed playsound-1.3.0
```

```
[34] pip install pygobject
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: pygobject in /usr/lib/python3/dist-packages (3.26.1)
```

```
[35] import cv2
     import numpy as np
     from google.colab.patches import cv2_imshow
     from matplotlib import pyplot as plt
     import librosa
     from tensorflow.keras.preprocessing import image
     from keras.models import load_model
```

```
# Create a VideoCapture object and read from input file
# If the input is the camera, pass 0 instead of the video file name
cap = cv2.VideoCapture('/content/drive/MyDrive/Forest with fire.mp4')

# Check if camera opened successfully
if (cap.isOpened()== False):
  print("Error opening video stream or file")
```

```
# Read until video is completed
while(cap.isOpened()):
  # Capture frame-by-frame
  ret, frame = cap.read()
  if ret == True:
    x=image.img_to_array(frame)
    res=cv2.resize(x,dsize=(128,128),interpolation=cv2.INTER_CUBIC)
    #expand the image shape
    x=np.expand_dims(res,axis=0)
    model=load_model("/content/forest1.h5")
    cv2_imshow(frame)
    pred=model.predict(x)
    pred = int(pred[0][0])
    pred
    int(pred)
    if pred==0:
      print('Forest fire')
      break
    else:
      print("danger")
      break

# When everything done, release the video capture object
cap.release()

# Closes all the frames
cv2.destroyAllWindows()
```

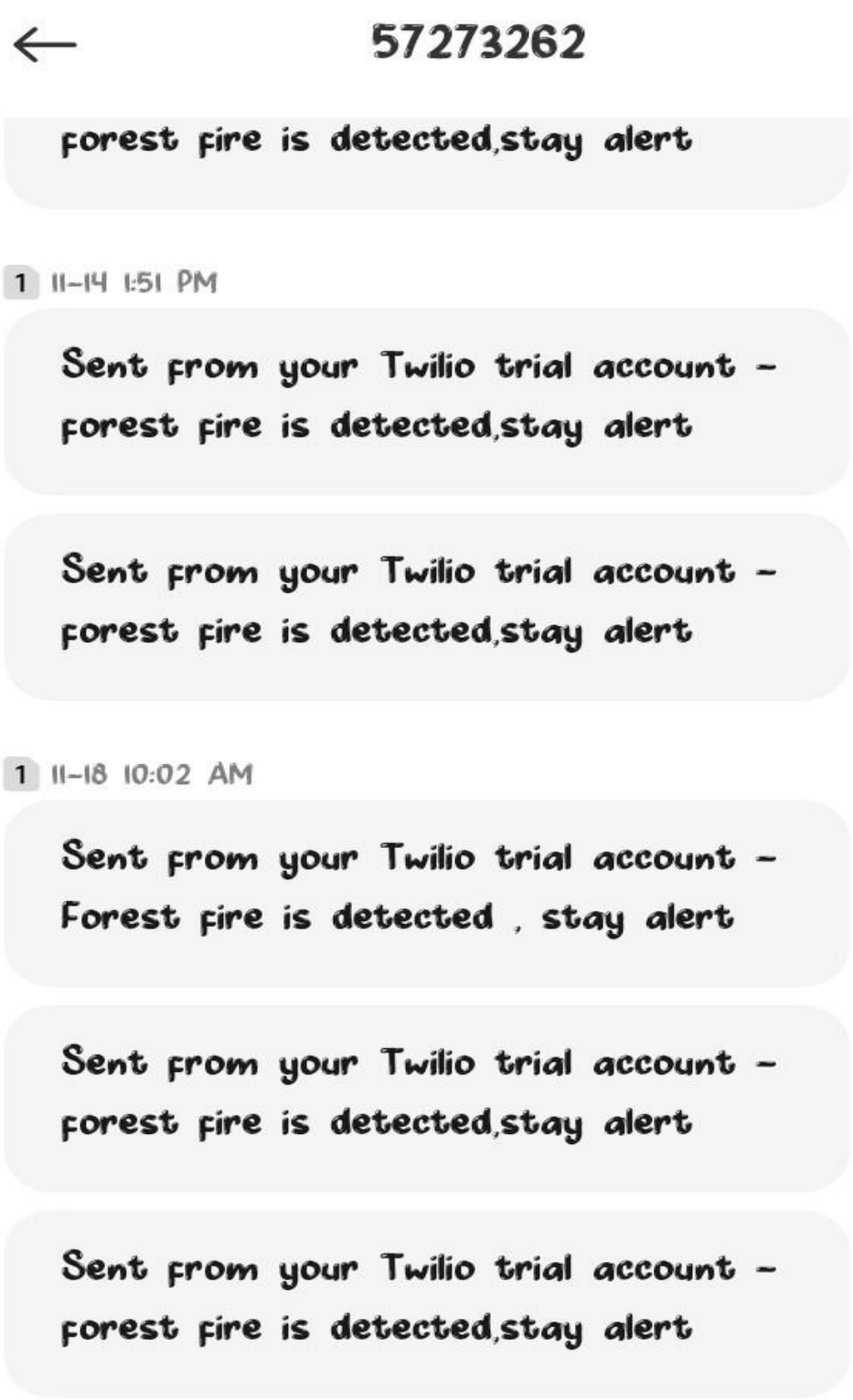
0s

#sending alert message

```
from twilio.rest import Client
from playsound import playsound
if pred==0:
  account_sid='ACff2114503c83032fe8cb614ba790fa6f'
  auth_token='225b921ecec6b05d45a9874ec77aa58c'
  client=Client(account_sid,auth_token)
  message=client.messages \
  .create(
      body='forest fire is detected,stay alert',
      #use twilio free number
      from_='+17088347091',
      #to number
      to='+917094693674')
  print(message.sid)
  print("Fire detected")
  print("SMS Sent!")
```

```
1/1 [==============================] - 0s 83ms/step
Forest fire
SM3989bc5aa4568e46b08e435c017f0df9
Fire detected
SMS Sent!
```

Message output



8.TESTING

8.1 Test cases

				Date	24-Nov-22								
				Team ID	PNT2022TMID46933								
				Project Name	Emerging methods for early detection of forest fires								
				Maximum Marks	4 marks								
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Pre-Requisite**	**Steps To Execute**	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**Commnets**	**TC for Automation(Y/N)**	**BUG ID**	**Executed By**
HomePage_TC_OO1	Functional	Home Page	Verify user is able to see the home page or not.		1.Enter URL and click go 2.Verify Whether the user is able to see the home page	Enter URL and click go	user able to see the home page	Working as expected	Pass	Nill	N	-	Preethi R
HomePage_TC_OO2	UI	Home Page	Verify the UI elements in home page		1.Enter URL and click go 2.Verify the UI elements in home page	Enter URL and click go	Application should show below UI elements	Working as expected	Pass	Nill	N	-	Kavitha M,Preethi
RegisterPage_TC_OO3	Functional	Register page	Register page is able to will input the user data.		1.Enter URL and click go 2.Verify the UI elements in home page 3.Click on signin button	Click in sign up home page	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	Nill	N	-	Priya M,Preethi R
LoginPage_TC_OO4	Functional	Login page	Verify user is able to redirect to detect page or not.		1.Enter URL and click go 2.Click on detect button 3.Verify Whether the user to redirect to detect page or not	Click in sign up home page	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	Nill	N	-	Srimathi G,Preethi R
PredictPage_TC_OO5	UI	Predict page	Verify the UI elements in Predict page.		1.Enter URL and click go 2.Verify the UI elements in Predict page	Click the predict button and redirect to predict page	Application should show below UI elements Dropdown list,detect button	Working as expected	Pass	Nill	N	-	Anushri K,Preethi R
PredictPage_TC_OO6	Functional	Predict page	Verify user is able to select the dropdown value or not.		1.Enter URL and click go 2.Click on detect button 3..Verify Whether the user to redirect to detect page or not 4..Verify user is able to select the dropdown value or not	fire predicted or not	Application should shows detecting video	Working as expected	pass	Nill	N	-	Preethi R, Kavitha M
redictpage_TC_00	Functional	Predict page	Verify the video		1.Enter URL and click go 2.Click on predict button 3..Verify Whether the user to redirect to predict page or not 4..Verify user is able to select the dropdown value or not 5.verify the video	Predicting the video	Application should shows the uploaded video	Working as expected	pass	Nill	N	-	Preethi R,Priya M
redictpage_TC_00	Functional	Predict page	Verify Whether the forest fire is predicted or not.		1.Enter URL and click go 2.Click on predict button 3..Verify Whether the user to redirect to predict page or not 4..Verify user is able to select the dropdown value or not 5.verify whether the video is predicted correctly or not	Click the detect button	Application shows the predicted output	Working as expected	Pass	Nill	N	-	Preethi R,Kavitha M, Priya M,Srimathi G, Anushri K

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 [ProductName] 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	1	0	0	0	1
Duplicate	0	0	0	0	0
External	0	0	0	0	0
Fixed	0	0	0	0	0
Not Reproduced	0	2	0	0	2
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	1	2	0	0	3

3.Test Case Analysis

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

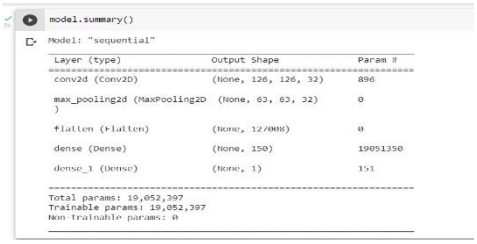
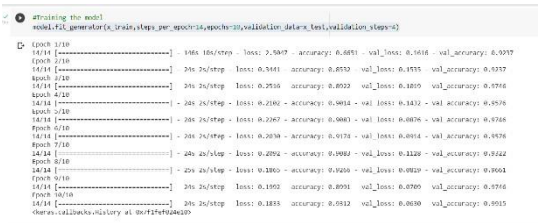
Section	Total Cases	Not Tested	Fail	Pass
Performance	5	0	0	5
UI	1	0	0	1
Security	3	0	0	3

9.Results

9.1 Performance Metrics

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Total params: 19,052,397 Trainable params: 19,052,397 Non-Trainable params: 0	
2.	Accuracy	Training Accuracy – 12.68 Validation Accuracy -13.07	

Model Summary

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 126, 126, 32)      896

 max_pooling2d (MaxPooling2D  (None, 63, 63, 32)       0
 )

 flatten (Flatten)           (None, 127008)            0

 dense (Dense)               (None, 150)               19051350

 dense_1 (Dense)             (None, 1)                 151

=================================================================
Total params: 19,052,397
Trainable params: 19,052,397
Non-trainable params: 0
_________________________________________________________________
```

Accuracy

```
#Training the model
model.fit_generator(x_train,steps_per_epoch=14,epochs=10,validation_data=x_test,validation_steps=4)
```

```
Epoch 1/10
14/14 [==============================] - 146s 10s/step - loss: 2.5047 - accuracy: 0.6651 - val_loss: 0.1616 - val_accuracy: 0.9237
Epoch 2/10
14/14 [==============================] - 24s 2s/step - loss: 0.3441 - accuracy: 0.8532 - val_loss: 0.1535 - val_accuracy: 0.9237
Epoch 3/10
14/14 [==============================] - 24s 2s/step - loss: 0.2516 - accuracy: 0.8922 - val_loss: 0.1019 - val_accuracy: 0.9746
Epoch 4/10
14/14 [==============================] - 24s 2s/step - loss: 0.2102 - accuracy: 0.9014 - val_loss: 0.1432 - val_accuracy: 0.9576
Epoch 5/10
14/14 [==============================] - 24s 2s/step - loss: 0.2267 - accuracy: 0.9083 - val_loss: 0.0876 - val_accuracy: 0.9746
Epoch 6/10
14/14 [==============================] - 24s 2s/step - loss: 0.2030 - accuracy: 0.9174 - val_loss: 0.0914 - val_accuracy: 0.9576
Epoch 7/10
14/14 [==============================] - 24s 2s/step - loss: 0.2092 - accuracy: 0.9083 - val_loss: 0.1128 - val_accuracy: 0.9322
Epoch 8/10
14/14 [==============================] - 25s 2s/step - loss: 0.1865 - accuracy: 0.9266 - val_loss: 0.0819 - val_accuracy: 0.9661
Epoch 9/10
14/14 [==============================] - 24s 2s/step - loss: 0.1992 - accuracy: 0.8991 - val_loss: 0.0709 - val_accuracy: 0.9746
Epoch 10/10
14/14 [==============================] - 24s 2s/step - loss: 0.1833 - accuracy: 0.9312 - val_loss: 0.0630 - val_accuracy: 0.9915
<keras.callbacks.History at 0x7f1fef024e10>
```

10.ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Fire removes low-growing underbrush, cleans the forest floor of debris, opens it up to sunlight, and nourishes the soil.
- Reducing this competition for nutrients allows established trees to grow stronger and healthier.
- History teaches us that hundreds of years ago forests had fewer, yet larger, healthier trees.
- Fire kills diseases and insects that prey on trees and provides valuable nutrients that enrich the soil.

DISADVANTAGES

- Forest fires can create health problems for people.
- Forest fires can trigger mudslides, landslides, and other forms of erosion.
- Forest fires under control can still burn other structures .
- The cutting down of forests leads to a loss in biodiversity.

11. CONCLUSION

A Deep Learning based Convolutional Neural Network (CNN) model is presented to detect a forest fire. The following techniques such as Image Collection, Preprocessing, Image Classification, Model building and video streaming and alerting is done. Initially, the images in the dataset are pre-processed, and fed into the CNN for feature extraction and detection.

12.FUTURE SCOPE

- The scope of using video frames in the detection of fire using CNN 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 systems.
- The proposed system can be developed to more advanced system by integrating wireless sensors for added protection and precision. The algorithm shows great promise in adapting to various environment.
- Future studies may focus on deploying the model into Database and cloud storage and using necessary support packages to detect the real time fire by making challenging and specific scene understanding datasets for fire detection methods and detailed experiments with Large datasets and training models.

13.APPENTEX

GitHub

https://github.com/IBM-EPBL/IBM-Project-45145-1660728510

Project Demo Link

https://youtu.be/J1gV4MpxERg