Emerging Methods for Early Detection of Forest Fires

PROJECT REPORT

Submitted by

TEAM ID : **PNT2022TMID37523**

TEAM MEMBERS

TEAM LEAD	**:**	**LAVANYA.A(312019104031)**
TEAM MEMBER 1	**:**	**GURUVARSHNI.G(312019104020)**
TEAM MEMBER 2	**:**	**GIRIJA.V(312019104018)**
TEAM MEMBER 3	**:**	**RUBIGA.S(312019104054)**

TABLE OF CONTENTS

TABLE OF CONTENTS

6. **PROJECT PLANNING & SCHEDULING**

   6.1 Sprint Planning & Estimation
   6.2 Sprint Delivery Schedule
   6.3 Reports from JIRA

7. **CODING & SOLUTIONING(Explain the features add in the project along with code)**

   7.1 Feature 1
   7.2 Feature 2
   7.3 Database Schema (if Applicable)

8. **TESTING**

   8.1 Test Cases
   8.2 User Acceptance Testing

9. **RESULTS**

   9.1 Performance Metrics

10. **ADVANTAGES & DISADVANTAGES**

11. **CONCLUSION**

12. **FUTURE SCOPE**

13. **APPENDIX**

INTRODUCTION

1.1 PROJECT OVERVIEW

Forest fires are occurring throughout the year with an increasing strength in the summer and autumn periods.These events are mainly caused by the actions of humans, but different nature and biota phenomena, like lightning strikes or ad-lib combustion of dried leafs or sawdust, can also be credited for their occurrence. Regardless of the reasons for the ignition of the forest fires, they normally cause withering damage to both nature and humans.

Forest fires are also considered as a main contributor to the air pollution, due to the fact that during every fire huge amounts of gases and particle mater are released in the atmosphere. To fight forest fires, different solutions were employed throughout the years. They ware primary aimed at the early detection of the fires. The simplest of these solutions is the formation of a network of observance posts - both cheap and easy to accomplish, but also time-consuming for the involved people.

The constant evolution of the info and communication applied science has led to the debut of a new generation of solutions for early detection and even prevention of forest fires. ICT-based networks of cameras and sensors and even satellite-based solutions were developed and used in the last decades. These solutions have greatly decreased the direct involvement of humans in the forest fire detection process, but have also proven to be expensive and hard to maintain.

1.2 PURPOSE

Detection of forest fire and smoke in wild land areas is done through remote sensing-based methods such as satellites, high-resolution static cameras fixed on the ground, and unmanned aerial vehicles Optical/thermal cameras deployed on the observation towers together with the other sensors such as smoke, temperature, and humidity sensors might detect the hazards in the closed environment rather than in the open environment as these sensors need vicinity to the fire or smoke.

The information obtained through these sensors is not appropriate. Distance covered by these methods could be limited, and to cover a large area, more sensors have to be deployed that might incur expenses. Through the deployment of UAV, large areas could be covered, and the images with high spatial and temporal resolutions could be captured properly.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

Over the last few years, climate change and human-caused factors have a significant impact on the environment. Some of these events include heat waves,droughts, dust storms, floods, hurricanes, and wildfires. Wildfires have extreme consequences on local and global ecosystems and cause serious damages to infrastructure, injuries, and losses in human lives; therefore, fire detection and the accurate monitoring of the disturbance type, size, and impact over large areas is becoming increasingly important. To this end, strong efforts have been made to avoid or mitigate such consequences by early fire detection or fire risk mapping.

Traditionally, forest fires were mainly detected by human observation from fire lookout towers and involved only primitive tools, such as the Osborne fire Finder however, this approach is inefficient, as it is prone to human error and fatigue.

On the other hand, conventional sensors for the detection of heat, smoke, flame, and gas typically take time for the particles to reach the point of sensors and activate them. In addition, the range of such sensors is relatively small, hence, a large number of sensors need to be installed to cover large areas

2.2 REFERENCES

1. Astana, M.A.; Apion the, C.; Mermoz, S.; Bouvet, A.; Le Toan, T.; Heuristic, M. Detection of windrows and insect outbreaks by L-band SAR: A case study in the Bavarian Forest National Park. Remote Send. Environ. 2018, 209, 700– 711.

2. Pradon, B.; Suliman, M.D.H.B.; Sawing, M.A.B. Forest fire susceptibility and risk mapping using remote sensing and geographical information systems (GIS). Disaster Prey. Manag. Into. J. 2007, 16.

3. Gaur, A.; Singh, A.; Kumar, A.; Kuliakan, K.S.; Lola, S.; Kapoor, K.; Srivastava, V.; Kumar, A.; Madhyapradesa, S.C. Fire sensing technologies: A review. IEEE Send. J. 2019, 19, 3191–3202.

4. Kabuki, R.; Frizz i, S.; Bouchouicha, M.; Chennai, F.; Moreau, E. Video smoke detection review: State of the art of smoke detection in visible and IR range. In Proceedings of the 2017 International Conference on Smart, Monitored and Controlled Cities (SM2C), Kerkennah-Sfax, Tunisia, 17 February 2017; pp. 81–86.

5. Allison, R.S.; Johnston, J.M.; Craig, G.; Jennings, S. Airborne optical and thermal remote sensing for wildfire detection and monitoring. Sensors 2016, 16, 1310.

6. Cappellini, Y.; Mattii, L.; Me-cocci, A. An Intelligent System for Automatic Fire Detection in Forests. In Recent Issues in Pattern Analysis and Recognition; Springer: Berlin/Heidelberg, Germany, 1989; pp. 563–570.

7. Chen, T.H.; Wu, P.H.; Chiou, Y.C. An early fire-detection method based on image processing. In Proceedings of the 2004 International Conference on Image Processing (ICIP 04), Singapore, 24–27 October 2004; Volume 3, pp. 1707–1710.

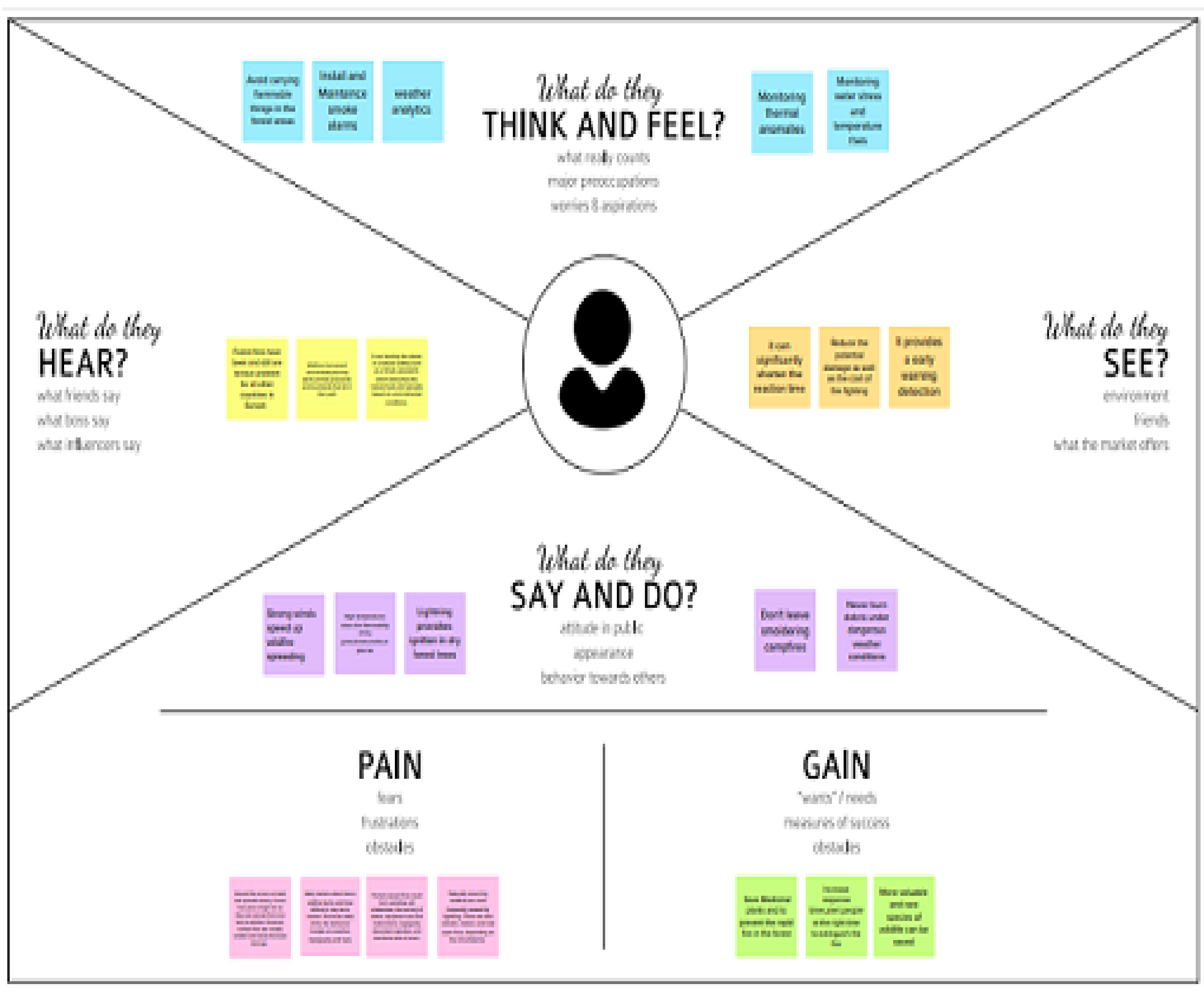
2.2 PROBLEM STATEMENT DEFENITION

Forest fires are a major environmental issue, creating economic and ecological damage while endangering human lives. There are typically about 100,000 wildfires in the United States every year. Over 9 million acres of land have been destroyed due to treacherous wildfires. It is difficult to predict and detect Forest Fire in a sparsely populated forest area and 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 before and also during the Fire due to its reliability and efficiency. The various real-time forest fire detection and prediction approaches, with the goal of informing the local fire authorities

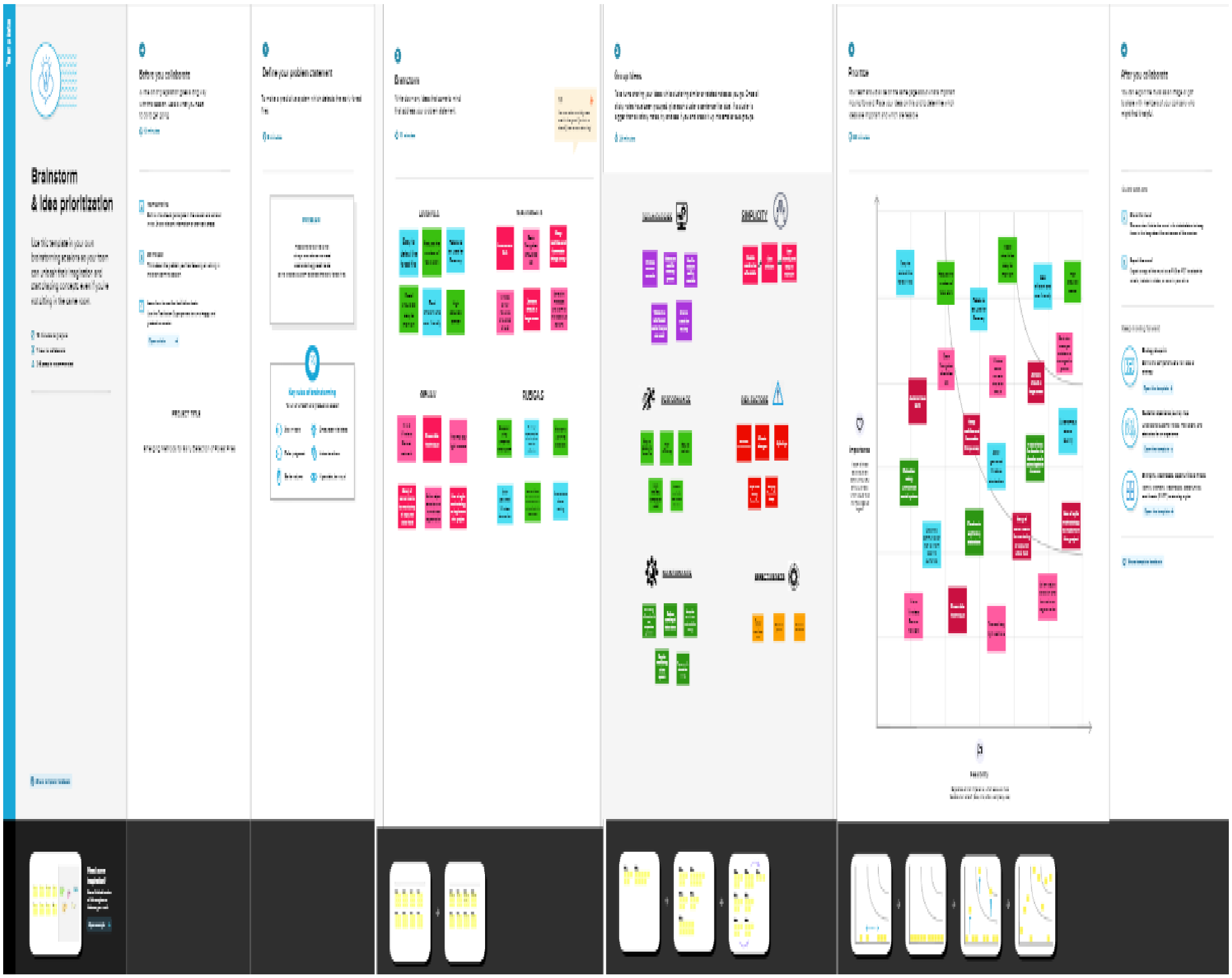
3.IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

An **empathy map** is a collaborative visualization used to articulate what we know about a particular type of user. It externalizes knowledge about users in order to create a shared understanding of user needs, and aid in decision making.



3.2 IDEATION & BRAINSTORMING



3.3 PROPOSED SOLUTION

S.NO	Paramater	Description
1.	Problem Statement (Problem to be solved)	**Statement:** To find emerging methods for early detection of forest fires using artificial intelligence. **Description**: This technology is to be implemented to locate a forest or a bush fire based on the concept of deep learning and YOLO algorithm. After detecting, authorities are to be alerted immediately to mitigate any damage.
2.	Idea / Solution description	**1.** In case of forest fire detection, the burning substances are primarily identified as sceptical flame regions using a division strategy to expel the non-fire structures and results are verified by deep learning model. **2.** The technology used to locate a forest or a bush fire is based on the concept deep learning andYOLO algorithm.
3.	Novelty / Uniqueness	1. Accurate and reliable recognition of sceptical flame regions by means of using YOLO v3 algorithm. 2. Unlike previous algorithms, the exact location of the origin of the forest fire is also detected and sent to the coating.

3.4 PROPOSED SOLUTION FIT

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

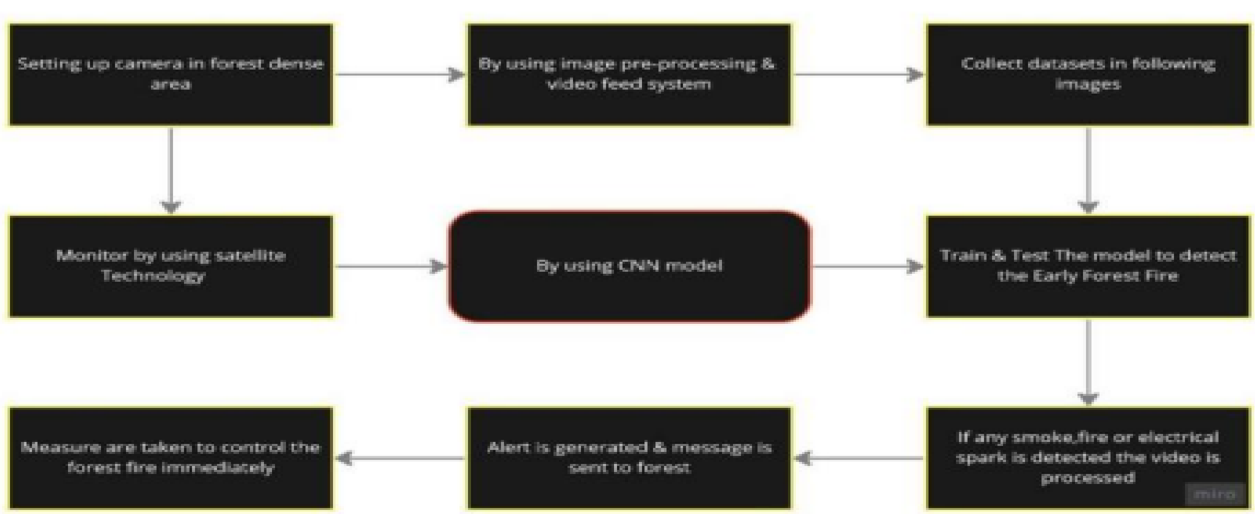
FR. NO.	Functional Requirement	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through wildfire portal.
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Data Prediction	Scientists create computer models to predict wildfire potential under a range of potential climate futures. Using different projections of temperature and downfall, scientists predict where and when wildfires are likely to occur

4.2 NON FUNCTIONAL REQUIREMENTS

FR. NO.	Non-Functional Requirement	Description
NFR-1	Usability	Many methods have been proposed to detect forest fires, such as camera-based systems, WSN-based systems, and machine learning coating-based systems, with both positive and negative aspects and performance figures of detection.
NFR-2	Protection	We have designed this project to secure the forest from wild fires.
NFR-3	Performance	In the event of a fire, the primary objective of using drones is to gather situational consciousness, which can be used to direct the efforts of the firefighters in locating and controlling hot spots. Just like urban fires, forest fires to require monitoring so that firefighters know what they are dealing with.

5. PROJECT DESIGN

5.1 DATAFLOW DIAGRAMS



5.2 **SOLUTION& TECHNICAL ARCHITECTURE**

4.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Forest Manageme nt Team	Setting up a camera	USN-1	As a user, the forest management team has to survey the forest by adding camera to the fire prone areas.	The live video captured can be monitored	High	Sprint-1
		USN-2	As a user, the forest management team can get video feed which is used for processing	The camera sends video or image to the forest centre	High	Sprint-2
		USN-3	Along with forest team, the NGO can also get access of the video to take some early measurement of forest fires.	They can also get the view of the live monitoring of forest	Low	Sprint-1
Technical Team	Image Classificatio n	USN-4	By using CNN Model, the images captured by the camera is classified accordingly by testing & training the model	The model should be able to identify the difference between fire and a normal smoke	Medium	Sprint-2
	Using Open CV	USN-5	The recorded video is under monitoring continuously to determine the detection of early video	Therefore, by using CNN we can determine the input layer, classify the hidden layers and send warnings through output layer	High	Sprint-2
Alert Team	Dashboard	USN-6	Thus, after successful detection of fire by processing images. This, API sends the alert by buzzing the alarm and sends messages through chatbot	Thus, the immediate response which is required for earlier determination through sending quick responses	High	Sprint-2

6.PROJECT PLANNING & SCHEDULING

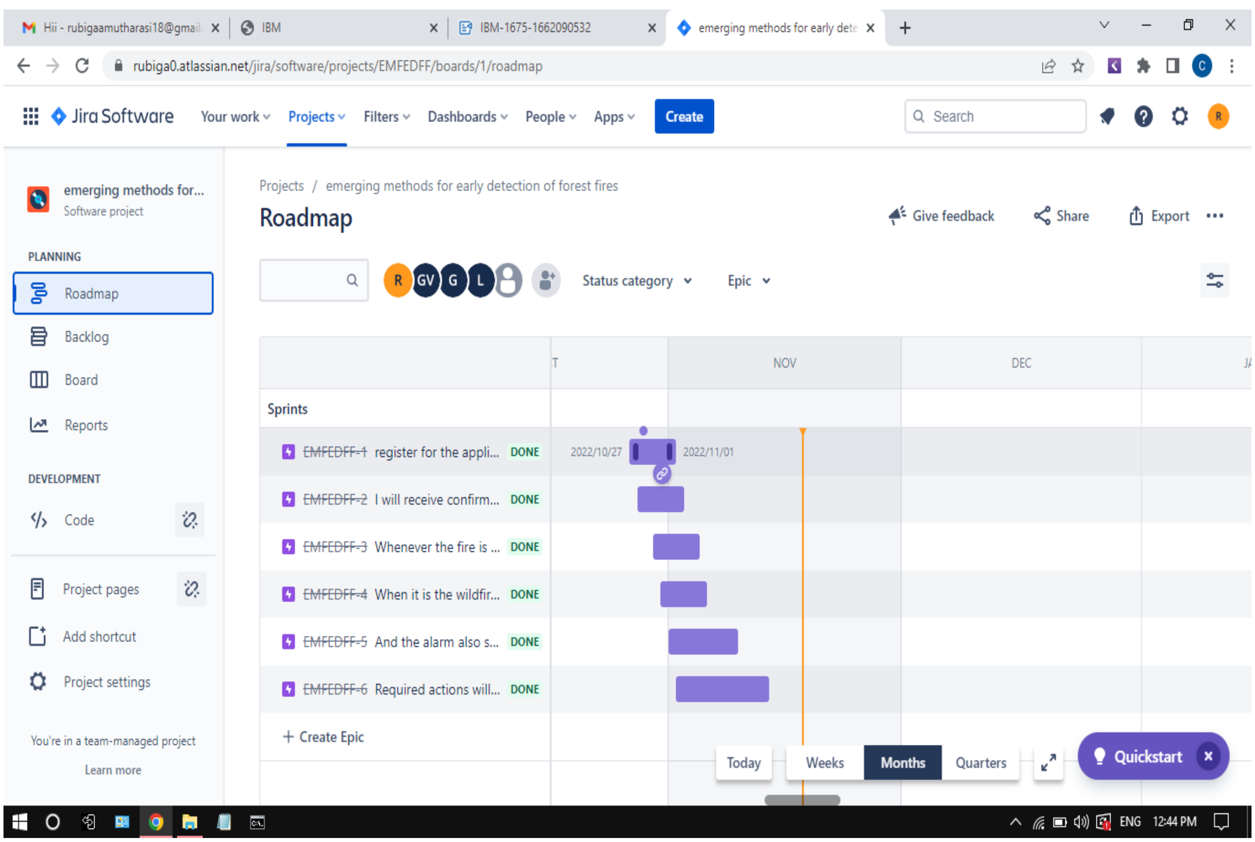
6.1 SPRINT PLANNING&ESTIMATES

Sprint	Functional Requirement (Epic)	User Story Number	User Story/Task	Story points	Priority	Team Members
Sprint 1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	20	High	Lavanya.A Guruvarshni.G Girija.V Rubiga.S
Sprint 2	Input	USN-3	Whenever the fire is detected, the information is given to the database.	20	High	Lavanya.A Guruvarshni.G Girija.V Rubiga.S
Sprint 3	Output	USN-5	And the alarm also sent to the like departments and made them know that the wildfire is erupted.	20	High	Lavanya.A Guruvarshni.G Girija.V Rubiga.S

6.2 SPRINT DELIVERY SCHEDULE

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

6.3 REPORT FROM JIRA



7.CODING & SOLUTIONING

7.1 FEATURE 1

```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
import matplotlib.pyplot as plt

train = ImageDataGenerator(rescale=1/255)
test = ImageDataGenerator(rescale=1/255)
train_dataset = train.flow_from_directory("/content/drive/MyDrive/Dataset/train_set",target_size=(150,150),batch_size = 32,class_mode = 'binary')
test_dataset = test.flow_from_directory("/content/drive/MyDrive/Dataset/test_set",target_size=(150,150),batch_size =32,class_mode = 'binary')
test_dataset.class_indices

model = keras.Sequential()
model.add(keras.layers.Conv2D(32,(3,3),activation='relu',input_shape=(150,150,3)))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Conv2D(64,(3,3),activation='relu'))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Conv2D(128,(3,3),activation='relu'))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Conv2D(128,(3,3),activation='relu'))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Flatten())
model.add(keras.layers.Dense(512,activation='relu'))
model.add(keras.layers.Dense(1,activation='sigmoid'))
model.summary()
```

```python
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
r = model.fit(train_dataset,epochs = 10,validation_data = test_dataset)
model.save("forest1.h5")
predictions = model.predict(test_dataset)
predictions = np.round(predictions)
predictions
print(len(predictions))
import matplotlib.pyplot as plt
plt.plot(r.history['loss'], label='loss')
plt.plot(r.history['val_loss'], label='val_loss')
plt.plot(r.history['accuracy'], label='accuracy')
plt.legend()
plt.plot(r.history['accuracy'], label='accuracy')
plt.plot(r.history['val_accuracy'], label='val_accuracy')
plt.legend()
def predictImage(filename):
    img1 = image.load_img(filename,target_size=(150,150))
    plt.imshow(img1)
    Y = image.img_to_array(img1)
    X = np.expand_dims(Y,axis=0)
    val = model.predict(X)
    print(val)
    if val == 1:
        plt.xlabel("Fire")
    elif val == 0:
        plt.xlabel("No Fire")
predictImage(r"C:\Users\ELCOT\Desktop\Python\new\Dataset\test_set\with fire\599857.jpg")
predictImage(r"C:\Users\ELCOT\Desktop\Python\new\Dataset\test_set\forest\01_NeilBurnell_Mystical_photoverticall.jpg")
predictImage(r"C:\Users\ELCOT\Desktop\Python\new\Dataset\train_set\with fire\with fire (2).jpg")
```

7.2 FEATURE 2

FLASK INTEGRATION

```
from __future__ import division, print_function
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
from flask import Flask, request, render_template
from werkzeug.utils import secure_filename
from twilio.rest import Client

global graph
#graph=tf.get_default_graph()
# Define a flask app
app = Flask(__name__)
model = load_model('forest1.h5')



@app.route('/', methods=['GET'])
def index():
    # Main page
    return render_template('digital.html')


@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['image']

        # Save the file to ./uploads
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)
        img1 = image.load_img(file_path,target_size=(150,150))
```

```python

        y = image.img_to_array(img1)
        x= np.expand_dims(y, axis=0)
        val = model.predict(x)
        print(val)
        if val == 1:
            send_message()
            result ="Fire"
        elif val == 0:
            result = "No Fire"
        return result

def send_message():
       client = Client(account_sid, auth_token)
       message = client.messages.create(
            body="Forest Fire detected , Stay safe!!!",
            )
       print(message.sid)
       print("Fire Detected")
       print("SMS Sent!")

if __name__ == '__main__':
    app.run(threaded = False

```

8.TESTING

8.1.TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	BUG ID	Executed By
HomePage_ TC_O O1	UI	Home Page	Display the Emerging Methods of Forest Fires	Enter URL(http://127.0.0.1:5000)	http://127.0.0.1:5000	Displaying the Home Page	Home Page displayed	**Pass**	**-**	Lavanya.A Guruvarshni.G
HomePage_ TC_O O2	UI	Home Page	Displayed the Prediction Forest Fires	1.Enter URL(http://127.0.0.1:5000) 2.Content of the Home Page should be Displayed	http://127.0.0.1:5000	Displaying the content of Home page	Content of Home page is displayed	**Pass**	**-**	Girija.V Rubiga.S
HomePage_ TC_O O3	Functional	Home	Checks whether the Drop the Image Here! Button is visible	1.Enter URL(http://127.0.0.1:5000) 2.Content of the Home Page should be Displayed 3.Click Drop the Image Here! Button	http://127.0.0.1:5000	Displays the Button	Drop the Image Here! Button is pops up.	Pass	-	Guruvarshni.G Girija.V
PredictiedPage_T C_OO4	Functional	Predicted page	Display the Prediction Page and Choose Image Button	1.Enter URL(http://127.0.0.1:5000) 2.Content of the Home Page should be Displayed 3.Click Drop the Image Here! Button 4.Click Choose Image	http://127.0.0.1:5000	Displays the Prediction Page and the Choose Image Button	Prediction page displayed. Choose image button was clicked.	Pass	-	Lavanya.A Rubiga.S
PredictedPage_TC _OO4	Functional	Predicted page	Select the Image and Click the Predict Button	1.Enter URL(http://127.0.0.1:5000) 2.Content of the Home Page should be Displayed 3.Click Drop the Image Here! Button 4.Click Choose Image 5.Choose the Image and Click Predict	http://127.0.0.1:5000	Displays e the selected Image	Displays the Selected Imag	Pass	**-**	Lavanya.A Guruvarshni.G

8.2.USER ACCEPTANCE TESTING

8.2.1.Defect Analysis

Resolution	severity 1	severity 2	severity 3	badness 4	subtotal
By Design	1	1	2	0	4
Duplicate	0	0	0	0	0
External	0	0	2	1	3
Fixed	4	2	4	1	11
Not Reproduced	0	0	0	0	0
Skipped	0	0	1	1	2
Won't Fix	0	0	0	1	1
Totals	5	3	9	4	21

8.2.2.TEST CASE ANALYSIS

Section	Total Cases	Not tested	Fail	Pass
Client Application	10	0	0	10
Security	2	0	0	2
Performance	2	0	0	2
Exception Reporting	2	0	0	2
Final Report Output	3	0	0	3

9.RESULTS

9.1.PERFORMANCE METRICES

<table>
<tr><th>S. No</th><th>Parameter</th><th>Values</th><th>Screenshot</th></tr>
<tr><td>1.</td><td>Model Summary</td><td>3,453,121</td><td>Layer (type) Output Shape Param #
conv2d (Conv2D) (None, 148, 148, 32) 896
max_pooling2d (MaxPooling2D) (None, 74, 74, 32) 0
conv2d_1 (Conv2D) (None, 72, 72, 64) 18496
max_pooling2d_1 (MaxPooling2 (None, 36, 36, 64) 0
conv2d_2 (Conv2D) (None, 34, 34, 128) 73856
max_pooling2d_2 (MaxPooling2 (None, 17, 17, 128) 0
conv2d_3 (Conv2D) (None, 15, 15, 128) 147584
max_pooling2d_3 (MaxPooling2 (None, 7, 7, 128) 0
flatten (Flatten) (None, 6272) 0
dense (Dense) (None, 512) 3211776
dense_1 (Dense) (None, 1) 513
Total params: 3,453,121
Trainable params: 3,453,121
Non-trainable params: 0</td></tr>
<tr><td>2.</td><td>Accuracy</td><td>Training Accuracy - 0.9665

Validation Accuracy -0.9833</td><td>Epoch 1/10
14/14 [==============================] - 96s 7s/step - loss: 0.5717 - accuracy: 0.6552 - val_loss: 0.2805 - val_accuracy: 0.8750
Epoch 2/10
14/14 [==============================] - 84s 6s/step - loss: 0.3266 - accuracy: 0.8434 - val_loss: 0.1193 - val_accuracy: 0.9667
Epoch 3/10
14/14 [==============================] - 74s 5s/step - loss: 0.2247 - accuracy: 0.9227 - val_loss: 0.1184 - val_accuracy: 0.9500
Epoch 4/10
14/14 [==============================] - 75s 5s/step - loss: 0.1682 - accuracy: 0.9425 - val_loss: 0.0206 - val_accuracy: 1.0000
Epoch 5/10
14/14 [==============================] - 82s 6s/step - loss: 0.1173 - accuracy: 0.9630 - val_loss: 0.0631 - val_accuracy: 0.9667
Epoch 6/10
14/14 [==============================] - 76s 6s/step - loss: 0.0925 - accuracy: 0.9741 - val_loss: 0.0169 - val_accuracy: 1.0000
Epoch 7/10
14/14 [==============================] - 80s 6s/step - loss: 0.0884 - accuracy: 0.9734 - val_loss: 0.0154 - val_accuracy: 1.0000
Epoch 8/10
14/14 [==============================] - 72s 5s/step - loss: 0.1050 - accuracy: 0.9661 - val_loss: 0.0455 - val_accuracy: 0.9833
Epoch 9/10
14/14 [==============================] - 70s 5s/step - loss: 0.1032 - accuracy: 0.9676 - val_loss: 0.0044 - val_accuracy: 1.0000
Epoch 10/10
14/14 [==============================] - 92s 6s/step - loss: 0.1144 - accuracy: 0.9665 - val_loss: 0.0807 - val_accuracy: 0.9833</td></tr>
</table>

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

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

DISADVANTAGE:

- The electrical interference diminishes the potency of radio receiver.
- The main drawback is that it has less coverage range areas.

11.CONCLUSION

This type of system is the first of its kind to ensure no further damage is then to forests when there is fire breakout and instantly a message is sent to the user through the App. Immediate response or early warning to a fire breakout is mostly the only ways to avoid losses and biology, cultural heritage damages to a great extent. Therefore the most important goals in fire surveillance are quick and authentic detection of fire. It is so much easier to suppress fire while it is in its early stages. info about progress of fire is highly valuable for managing fire during all its stages. Based on this data the firefighting staff can be guided on target to block fire before it reaches cultural heritage sites and to suppress it quickly by utilise required firefighting equipment and vehicles. With further research and invention, this project can be implemented in various forest areas so that we can save our forests and maintain great environs.

12.FUTURE SCOPE

This project is far from complete and there is a lot of room for betterment. Some of the betterment that can be made to this project are as follows:

Additional pump can be added so that it automatically sends water when there is a fire breakout. Also industrial sensors can be used for better ranging and accuracy.

➔ This project has endless potential and can always be enhanced to become better. enforce this concept in the real world will benefit several industries and reduce the workload on many workers, enhancing overall work efficiency.

13.APPENDIX

SOURCE CODE

```
import tensorflow as tf
import numpy as np
from tensorflow import keras
import os
import cv2
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt
```

```
train = ImageDataGenerator(rescale=1/255)
test = ImageDataGenerator(rescale=1/255)

train_dataset = train.flow_from_directory("/content/drive/MyDrive/Dataset/train_set",
                                          target_size=(150,150),
                                          batch_size = 32,
                                          class_mode = 'binary')

test_dataset = test.flow_from_directory("/content/drive/MyDrive/Dataset/test_set",
                                          target_size=(150,150),
                                          batch_size =32,
                                          class_mode = 'binary')
```

```
test_dataset.class_indices
```

```
model = keras.Sequential()
model.add(keras.layers.Conv2D(32,(3,3),activation='relu',input_shape=(150,150,3)))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Conv2D(64,(3,3),activation='relu'))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Conv2D(128,(3,3),activation='relu'))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Conv2D(128,(3,3),activation='relu'))
model.add(keras.layers.MaxPool2D(2,2))
model.add(keras.layers.Flatten())
model.add(keras.layers.Dense(512,activation='relu'))
model.add(keras.layers.Dense(1,activation='sigmoid'))
```

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

```
r = model.fit(train_dataset,
          epochs = 10,
          validation_data = test_dataset)
```

```
predictions
```

```
print(len(predictions))
```

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

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

```
plt.plot(r.history['acc'], label='acc')
plt.plot(r.history['val_acc'], label='val_acc')
plt.legend()
```

```
def predictImage(filename):
    img1 = image.load_img(filename,target_size=(150,150))
    plt.imshow(img1)
    Y = image.img_to_array(img1)
    X = np.expand_dims(Y,axis=0)
    val = model.predict(X)
    print(val)
    if val == 1:
        plt.xlabel("Fire",fontsize=30)
    elif val == 0:
        plt.xlabel("No Fire",fontsize=30)
```

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

predictImage('/content/drive/MyDrive/Dataset/test_set/forest/0.48007200_
1530881924_final_forest.jpg')

predictImage('/content/drive/MyDrive/Dataset/train_set/with fire/with
fire (101).jpg')

predictImage('/content/drive/MyDrive/Dataset/test_set/forest/0.48007200_
1530881924_final_forest.jpg')

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

GITHUB LINK

IBM-EPBL/IBM-Project-1675-1658409227

PROJECT DEMO LINK

https://drive.google.com/drive/folders/1UEOt5yZay-2M2Svo-5re0i-drVbUZKQz