PROJECT REPORT

Emerging Methods for Early Detection of Forest Fire

TEAM ID : PNT2022TMID48835

Faculty mentor(s) : Aruljothi

Industry Mentor(s): Shanthi

TEAM SIZE : 4

Team Leader **;** Veeranirajana.V-921019104053

Team member **:**Renuga.B-921019104041

Team member **:**Shanmugapriya.T-921019104047

Team member **:**Monica.T-921019104032

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Machine learning and deep learning play an important role in computer technology and artificial intelligence. With the use of deep learning and machine learning, human effort can be reduced in recognizing, learning, predictions and in many more areas

Forest fire detection is the ability of computer systems to recognise fire from various region of forest , such as fire, smoke, and so on. This project aims to let users take advantage of machine learning to reduce manual tasks in Detecting the forest fire.

1.2 PURPOSE

The main aim of our project is detection and monitoring the forest fire To minimize the effect of fire breakout by controlling in its early stage also to protect Domestic by informing about the breakout to the respective forest department as early as possible . We have implemented the AI technology to achieve our objective.

CHAPTER 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 isnot 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].In this paper, the author uses CNN-convolutional neural networks to detect fire with the help of live video footage through anti-fire surveillance systems. The paper proposes

YOLOv2 convolutional neural network is one of the best solutions for detecting fire and smoke both indoor and outdoor environment. You only look once (YOLO) is a deep learning model for object detection, YOLOv2 is the next version whbeeupgraded to rectify the setbacks of YOLO namely the inaccuracy to locate and mark the region of interest in the images and the lower recall rate compared to other region-oriented algorithms. Thus, increasing the efficiency of the architecture. They started with an input image of size 128x128x3. They used convolutional layers to map the features on the input image. The features extracted

then given as input to YOLOv2 object detection subnetwork. YOLOv2 Transform layer is implemented to improve network stability for object localiza

[2] This paper the authors propose a system that mimics fire the humadetection system. It uses Faster R-CNN which is a region-based algorithm to detect suspicious Point of interest. After marking the region of interest, the features extracted from the bounding boxes are passed to LSTM Long Short-Term Memory to classify if there is fire or not in short interval of time. Faster R-CNN exploits the features of CNN and introduce a region proposal network which is used to map the features in the input image. It extracts features through the ROI pooling operation and then classifies according to the class scores of the object position.

[3] This research paper, the authors propose a cost-effective fire detection using CNN from surveillance videos. This papers critically analyses the statistics of deaths due to fire. So, their focus is to propose a system that is home friendly and commercial. This paper gives us an insight of how to carefully select the data properly, how to analyse the computational complexity and detection accuracy. They use a model called GoogleNet for extracting the features from the images. For reducing the complexity of larger patches, they reduce dimensionality. The model is tested with two different datasets for validation purposes and results are compared. They achieved an accuracy of 93.5% on the first dataset and an 86% on the next dataset.

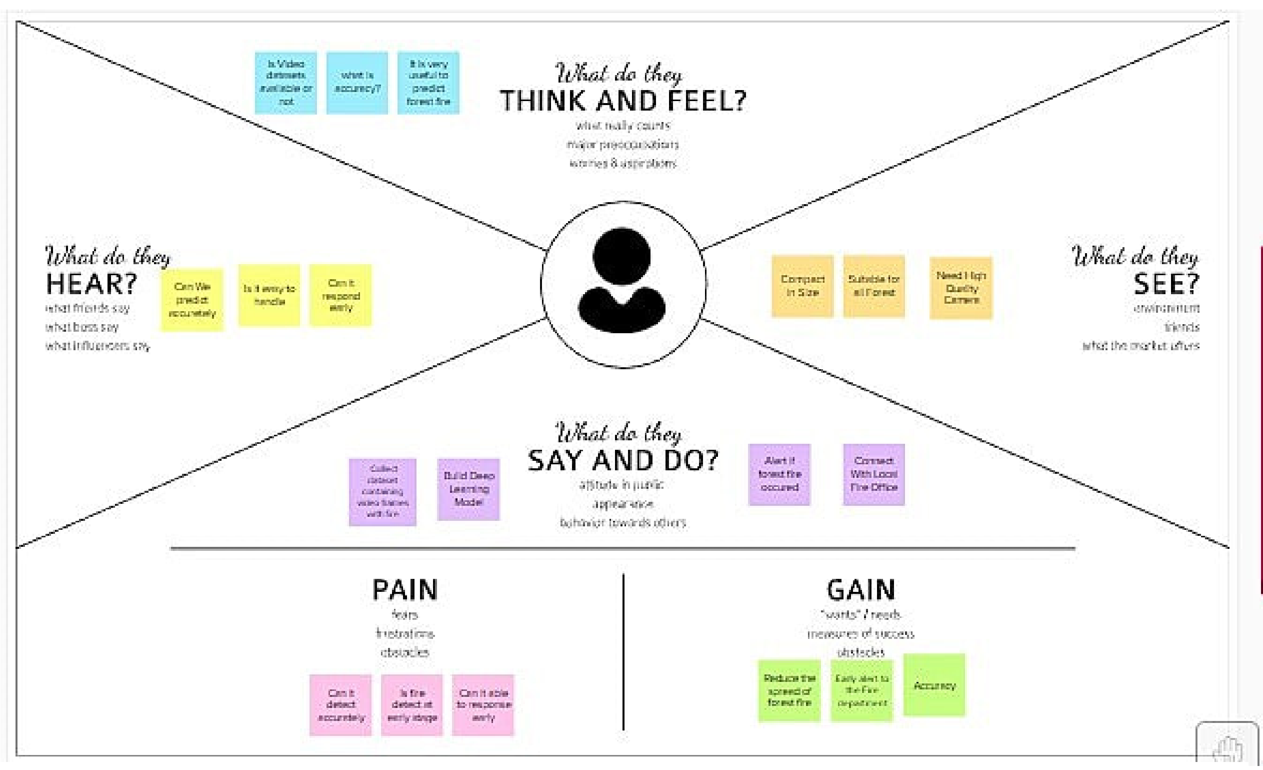
2.3 PROBLEM STATEMENT DEFINITION

Forest fires lead to destruction of forest wealth and not only that it also destroys the flora and fauna which causes harm to biodiversity. Forest is great resources and to preserve them is a major challenge. As, they are irreparable damage to the ecosystem, so forest fire and prevention is utmost important and best way to tackle this problem. But the forest fire early detection and prevention is another major challenge faced all over the world. Several methods for controlling and monitoring of fires have been proposed. In earlier days, manned observation towers were used but this technique was inefficient and failed. After that satellite and camera imaging technologies were tried but this also proved inefficient and ineffective. For example, cameras were installed at different sites in forest but these provide only line of sight pictures. For a very large areas alert system is required as it is really tedious task to monitor all the images.

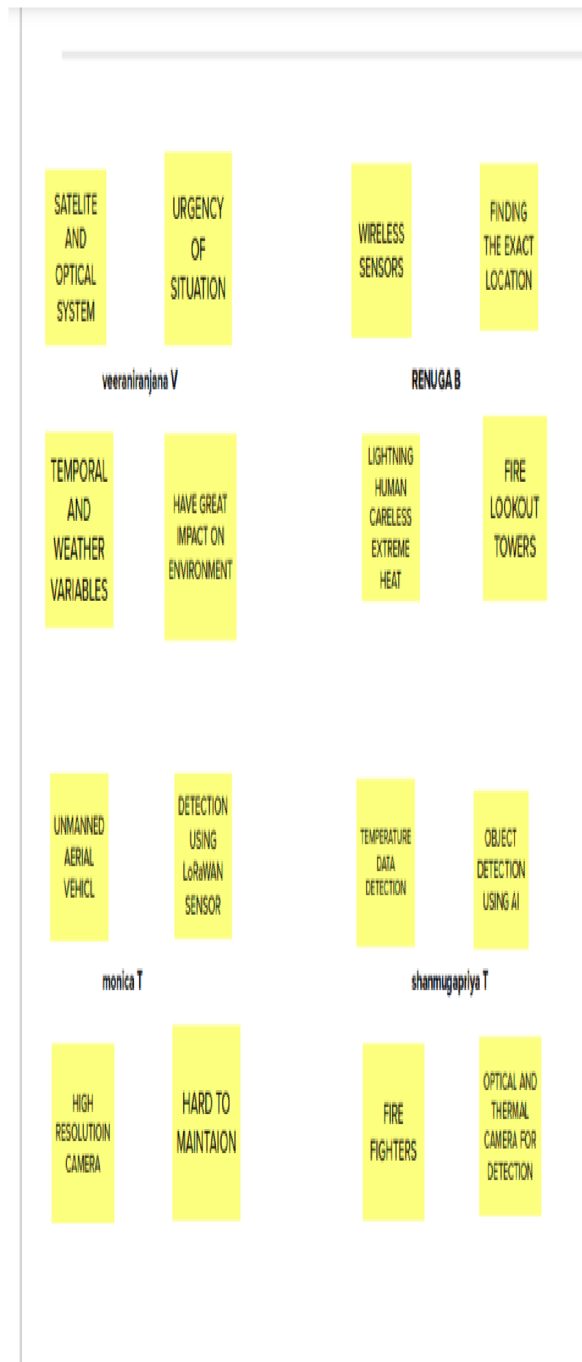
CHAPTER 3

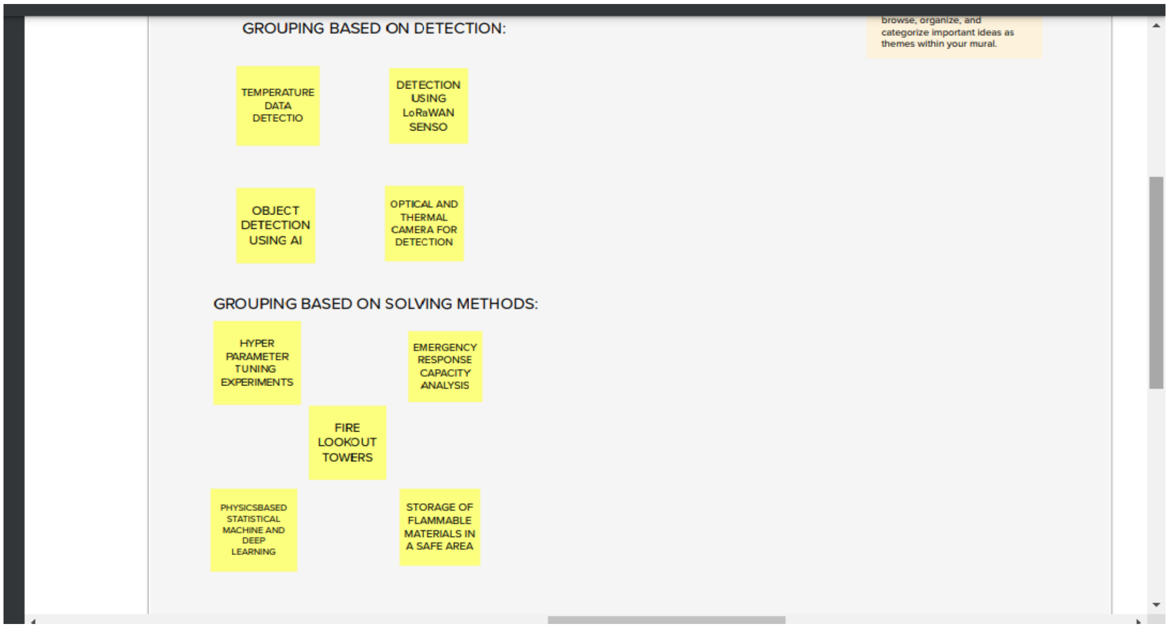
IDEATION AND PROPOSED SOLUTION

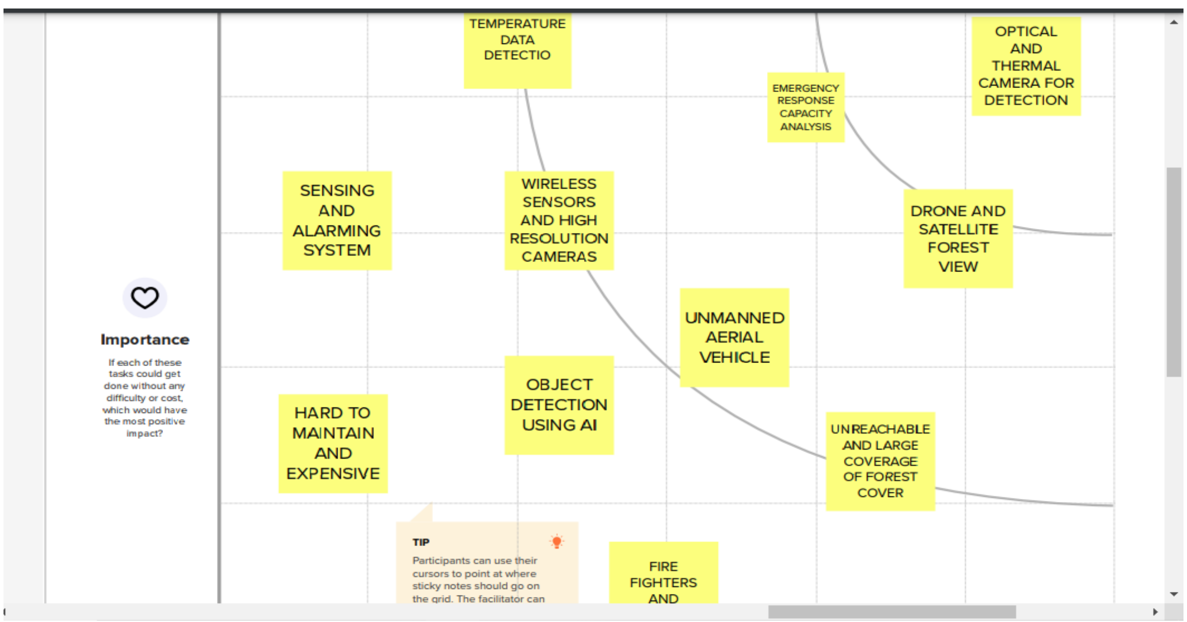
3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTORMING




GROUPING BASED ON DETECTION:
browse, organize, and categorize important ideas as themes within your mural.
TEMPERATURE DATA DETECTIO
DETECTION USING LoRaWAN SENSO
OBJECT DETECTION USING AI
OPTICAL AND THERMAL CAMERA FOR DETECTION
GROUPING BASED ON SOLVING METHODS:
HYPER PARAMETER TUNING EXPERIMENTS
EMERGENCY RESPONSE CAPACITY ANALYSIS
FIRE LOOKOUT TOWERS
PHYSICSBASED STATISTICAL MACHINE AND DEEP LEARNING
STORAGE OF FLAMMABLE MATERIALS IN A SAFE AREA


TEMPERATURE DATA DETECTIO
OPTICAL AND THERMAL CAMERA FOR DETECTION
EMERGENCY RESPONSE CAPACITY ANALYSIS
SENSING AND ALARMING SYSTEM
WIRELESS SENSORS AND HIGH RESOLUTION CAMERAS
DRONE AND SATELLITE FOREST VIEW
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
UNMANNED AERIAL VEHICLE
OBJECT DETECTION USING AI
HARD TO MAINTAIN AND EXPENSIVE
UNREACHABLE AND LARGE COVERAGE OF FOREST COVER
TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can
FIRE FIGHTERS AND

3.3 PROPOSED SOLUTION

	parameter	Description
	Problem Statement (Problem to be solved)	To find emerging methods for early detection of forest fires using artificial intelligence.
	Idea / Solution description	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 a deep learning model.
	Novelty / Uniqueness	Accurate and reliable recognition of sceptical flame regions by means of using YOLO v3 algorithm.
	Social Impact / Customer Satisfaction	1.By using this method we can save environmental damage and lives of living beings. 2. It is fast and accurate method to detect the fire easily and give an alert to the forest fire department simultaneously when the fire is detected.
	Business Model (Revenue Model)	The software platform to provide the fully autonomous processing of data received from the camera of UAV to obtain live feed in web App.
	Scalability of the Solution	It is mainly developed for detecting the forest fire across the world and usefu surveillance the different sections of the forest.

3.4 PROBLEM SOLUTION FIT

Project Title: Emerging Methods For Early Detection Of Forest Fire **Project Design Phase-I Problem Solution-Fit** **Team ID:** PNT2022TMID48835

Define CS, fit into CC

1.CUSTOMER SERVICE CS

1.Tribal people and forest department officers living in forest.
2.Animals , birds and other living things in the forest.

6. CUSTOMER CONSTRAINTS CC

1.Solar power cameras can be used as a power source
2. Waterproof cameras.
3.Seamless connection.

5. AVAILABLE SOLUTIONS AS

1.Notification is sent via messages.
2.Fire alarm is activated to nearby stations.

EXplore A S, differentiate

Focus on J&P, tap into BE, understand RC

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

1.Detecting small fire sparks in forest becomes difficult.
2.Camera should always be in motion

9. PROBLEM ROOT CAUSE RC

1.Special analysis system can be used.
2. Wireless mobile network via SIM can be used transfer alert message throughout areas.

7. BEHAVIOUR BE

1.Climate change should be monitored.
2.Hot areas should be monitored clearly.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

1.Correct detection.
2.Alarm alert
3.Follow correct algorithm

4. EMOTIONS: BEFORE / AFTER EM

BEFORE	AFTER
1.Unable to detect small sparks.	1.Able to detect small sparks.
2.camera should always be in motion.	2. 360 view of camera is used.

10. YOUR SOLUTION SL

1.Mobile application can be developed for specific areas.
2.Forest can be monitored by several cameras.
3.This can be used in wild life sanctuaries.

8.CHANNELS of BEHAVIOUR CH

ONLINE

Connected direc1tly to the user via Internet.

OFFLINE

Alerts can be sent via Offline messages and an alarm system is activated.

Identify strong TR & EM

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FUNCTIONAL REQUIREMENTS:

Following are the functional requirements of the proposed solution

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	Registration through Form
FR-3	Image recognition	The system shall be able to take real inputs of satellites images and determine whether image contains fire or not.
FR-4	Forest Monitoring	Forest are monitored 24/7 through
FR-5	Alert	The system will send notification to the user when fire is detected
FR-6	Detection	The system shall take training sets of fire and checks for fire or no fire or smoke
FR-7	Operating system	The system can run as a service on Windows or Linux operating system.

NON-FUNCTIONALREQUIREMENTS:

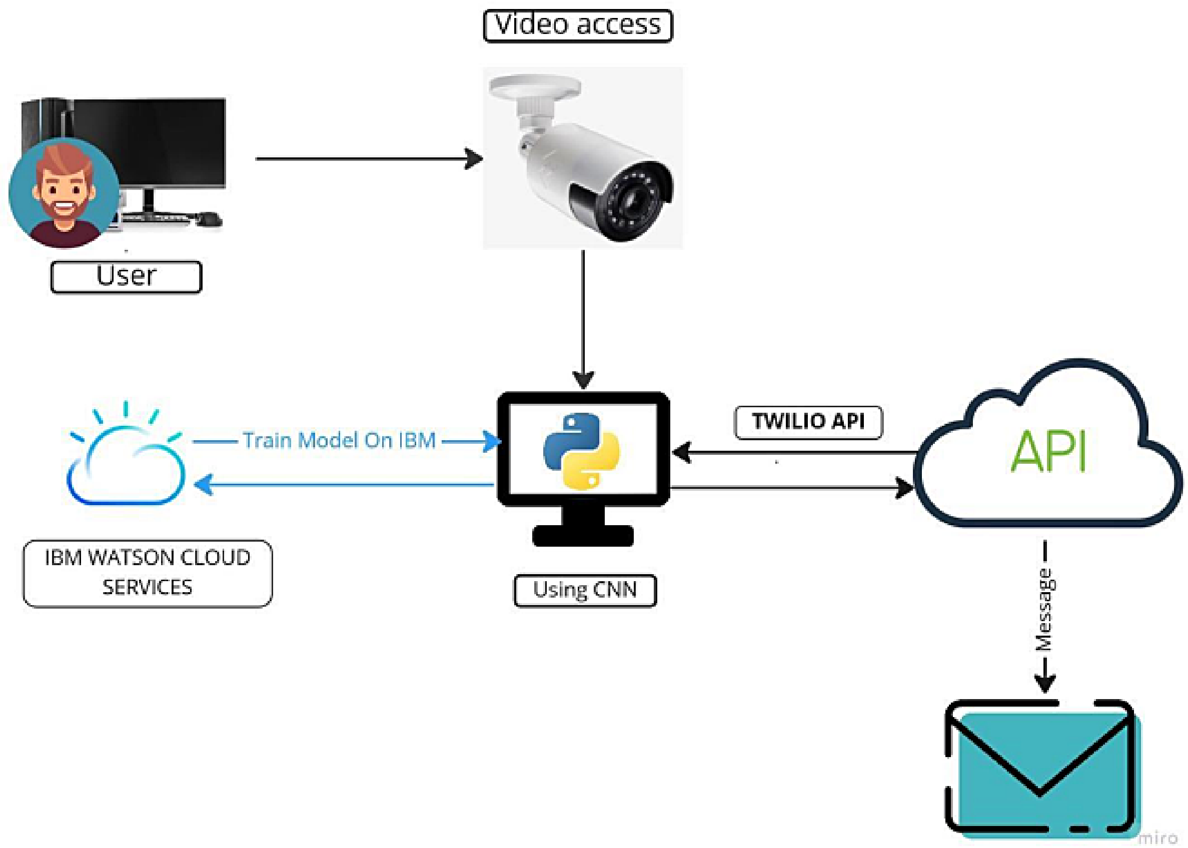
Following are the non-functional requirements of the proposed solution

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Model is user friendly to use and very effective.
NFR-2	Security	More secure environment
NFR-3	Reliability	Model is safe to install
NFR-4	Performance	Model will achieve high accuracy
NFR-5	Availability	Build model is available in all the time
NFR-6	Scalability	Model can handle large amount of data and can easily adapt to every environment.
NFR-7	Testability	Putting in more training data into the model can improve the accuracy level of the system.

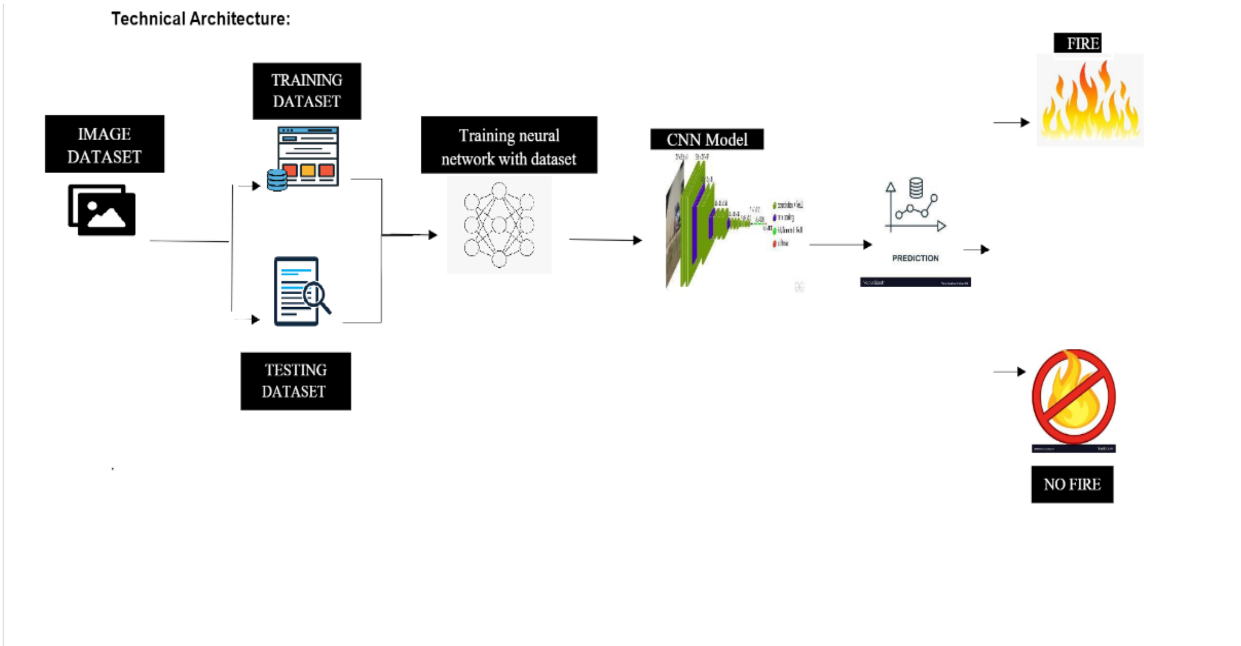
CHAPTER 5

PROJECT DESIGN

5.1.Data Flow Diagram



5.2 SOLUTION & TECHNICAL ARCHITECTURE:



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	It is necessary to collect the right data image feed frame camera.	High	Sprint-1

			rain of the forest			
		USN-2	Identify algorithms that can be used for prediction	To collect the algorithm to identify the accuracy level of each algorithm	Medium	Sprint-2
	Implement Algorithm.	USN-3	Identify the accuracy of each algorithm	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 the accuracy of algorithm.	USN-5	Identify accuracy, precision, recall of each algorithm	These values are important for obtaining the right output	High	Sprint-3
	Display unit.	USN-6	Outputs from each algorithm are obtained	It is highly used to predict the effect and to take precautionary measures	High	Sprint-4

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

User Type	Functional Requiremnt (Epic)	User Story Number	User Story / Task	Story Points	Priorty	Team Members
Sprint-1	Download data set	USN-1	The data is downloaded from theKaggle website and then the data set is classified into training and testing images.	10	High	B.Renuga V.Veeranirajana

Sprint-1	Image pre-processing	USN-1	In Image processing technique the first step is usually importingthe libraries that will be needed in the program. Import Keras libraryfrom that library and import the ImageDataGenerator Library to your Python script. The next step is definig the arguments for the ImageDataGenerator Here the arguments whichwe are given inside the image data generator class are, rescale, shear_range, rotation range of image, and zoom range that we can consider for images. The next step is applying the ImageDataGenerator arguments to the train and test dataset.	10	High	B.Renuga V.Veeranirajana
Sprint-2	Training image	USN-2	In this training phase the ImageDataGenerator arguments is applied to the training images and the model is tested with several images and the model is saved.	20	High	T.Monica B.Renuga T.Shanmugapriya V.Veeranirajana
Sprint-3	Testing	US	In this testing phase the	20	High	T.Monica

	image	N-3	Image processing techniques is applied to the testing images and executed for prediction.			B.Renuga T.Shanmugapriya V.Veeranirajana
Sprint-4	Evaluation metrics and accuracy	US N-4	In this phase the result, prediction, accuracy, and performance of the project are tested.	20	High	T.Monica B.Renuga T.Shanmugapriya V.Veeranirajana

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	07 Nov 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Oct 2022	20	07 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

CHAPTER 7

CODING & SOLUTIONING

SPRINT-1

Image Data PreProcessing

Importing necessary Libreries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
# Data Augumentation
#Define The Parameters /Arguments For ImageDataGenerator Class
```

```
# Data Generator

train_data_generator = ImageDataGenerator(rescale=1./255,
                                          rotation_range=90,
                                          width_shift_range=0.4,
                                          fill_mode='reflect',
                                          vertical_flip=True,
                                          channel_shift_range=150.0,
                                          zoom_range=[0.5, 1.5],
                                          shear_range=45.0,
                                          brightness_range=(0.2, 0.8))

test_data_generator = ImageDataGenerator(rescale=1./255)
```

```
#Applying ImageDataGenerator Functionality To Trainset
```

```
                                    rotation_range=90,
                                    width_shift_range=0.4,
                                    fill_mode='reflect',
                                    vertical_flip=True,
                                    channel_shift_range=150.0,
                                    zoom_range=[0.5, 1.5],
                                    shear_range=45.0,
                                    brightness_range=(0.2, 0.8))

test_data_generator = ImageDataGenerator(rescale=1./255)
```

```
#Applying ImageDataGenerator Functionality To Trainset
```

```
train_data = train_data_generator.flow_from_directory(r"./Dataset/Dataset/train_set",
                                                      target_size=(64,64),
                                                      batch_size=50,
                                                      class_mode='binary',
                                                      shuffle=True,
                                                      color_mode='rgb')
#Applying ImageDataGenerator Functionality To  Testset

test_data = test_data_generator.flow_from_directory(r"./Dataset/Dataset/test_set",
                                                      target_size=(64,64),
                                                      batch_size=50,
                                                      class_mode='binary',
                                                      shuffle=True,
                                                      color_mode='rgb')
```

```
Found 436 images belonging to 2 classes.
Found 121 images belonging to 2 classes.
```

SPRINT-2

CNN MODEL BUILDING

Model Building

[37]:

```
# Initializing the Model
model = Sequential()
# Adding CNN Layers
#convolution and Pooling layer 1
model.add(Conv2D(filters=48,kernel_size=3,activation='relu',input_shape=(64,64,3)))
model.add(MaxPool2D(pool_size=2,strides=2))
model.add(Dropout(0.4))

#convolution and Pooling layer 2
model.add(Conv2D(filters=32,kernel_size=3,activation='relu'))
model.add(MaxPool2D(pool_size=2,strides=2))
model.add(Dropout(0.4))

#Flattening the images
model.add(Flatten())

# Add Dense Layer
model.add(Dense(64,activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(1,activation='sigmoid'))
```

In [43]:

```
#Save the Model
model.save('Forest_Fire.h5')
```

In [44]:

```
#Test Model
saved_model = load_model('forest_Fire.h5')
```

In [12]:

```
#Predict The Model
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
import cv2
```

In [23]:

```
model = load_model("forest_fire.h5")
```

In [27]:

```
img=image.load_img(r'Dataset\Dataset\test_set\forest\0.48007200_1530881924_final_forest.jpg')
x=image.img_to_array(img)
res = cv2.resize(x, dsize=(64, 64), interpolation=cv2.INTER_CUBIC)
x = np.expand_dims(res, axis=0)
```

In [28]:

```
pred = model.predict(x)
pred = int(pred[0][0])
pred
```

Out[28]: 0

In [31]:

```
img=image.load_img(r'Dataset\Dataset\test_set\with fire\19464620_401.jpg')
x=image.img_to_array(img)
res = cv2.resize(x, dsize=(64, 64), interpolation=cv2.INTER_CUBIC)
x = np.expand_dims(res, axis=0)
```

SPRINT-4

Video Analaysis

```
import  cv2
import numpy as np
from keras.preprocessing import image
from keras.models import load_model
from twilio.rest import Client
from playsound import playsound
```

```
import cv2
import numpy as np

from matplotlib import pyplot as plt

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('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=(64,64),interpolation=cv2.INTER_CUBIC)
    #expand the image shape
    x=np.expand_dims(res,axis=0)
    model=load_model("Forest_Fire.h5")
    cv2.imshow('Forest Fire Detection',frame)
    pred=model.predict(x)
    pred = int(pred[0][0])
    pred
    int(pred)
    if pred==0:
```

```python
    if pred==0:
      print('Forest fire')

      account_sid='ACf68255cbe82460c9266beb672d744602'
      auth_token='c7e7819f9ee7412b408abe7b09083cae'
      client=Client(account_sid,auth_token)
      message=client.messages \
     .create(
      body='forest fire is detected,stay alert',
      #use twilio free number
      from_='+18176705182',
      #to number
      to='+916381614097')
      print(message.sid)
      print("Fire detected")
      print("SMS Sent!")
      playsound(r'Alaram.mp3')

    else:
      print("no danger")
      break
# When everything done, release the video capture object
cap.release()

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

```
Forest fire
SM4c3ed97a65a57ae32b897612f200074d
Fire detected
SMS Sent!
Forest fire
SM5e30af484b03bf86b962f2b65ac66b24
Fire detected
SMS Sent!
Forest fire
SMcd0d8c4a8f6d82c48b852d658ae66af1
Fire detected
SMS Sent!
```

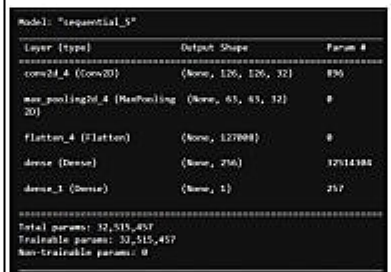
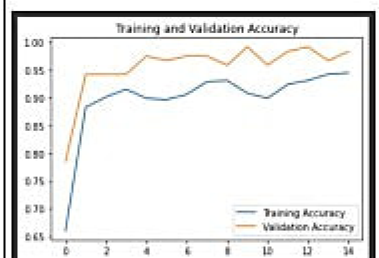
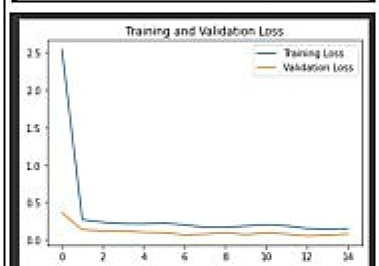
CHAPTER 8

TESTING

Performance Testing

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 : 32,515,457** **Trainable params : 32,515,457** **Non-trainable params : 0**	
2.	Accuracy	**Training Accuracy – 94.50%** **Validation Accuracy – 98.35%**	 

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 126, 126, 32)      896

 max_pooling2d_4 (MaxPooling  (None, 63, 63, 32)       0
 2D)

 flatten_4 (Flatten)         (None, 127008)            0

 dense (Dense)               (None, 256)               32514304

 dense_1 (Dense)             (None, 1)                 257

=================================================================
Total params: 32,515,457
Trainable params: 32,515,457
Non-trainable params: 0
_________________________________________________________________
```

Acceptance Testing UAT Execution & Report Submission

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

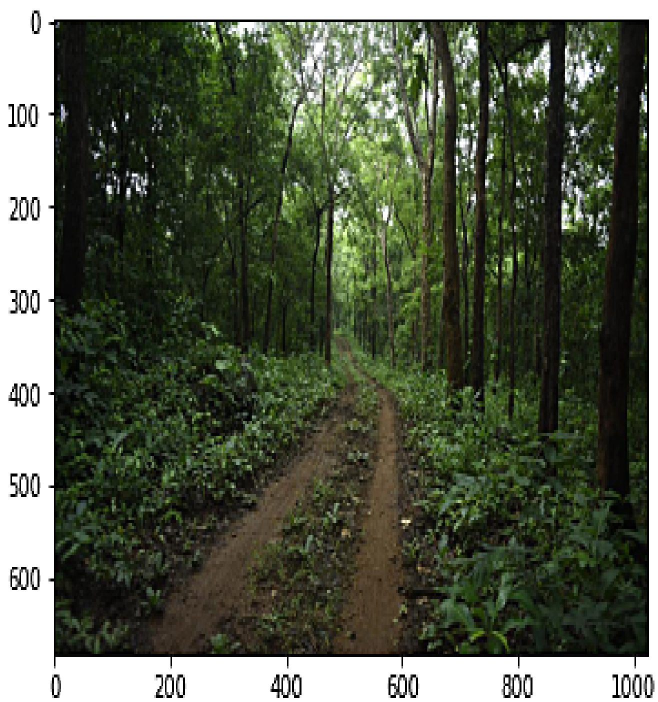
CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS

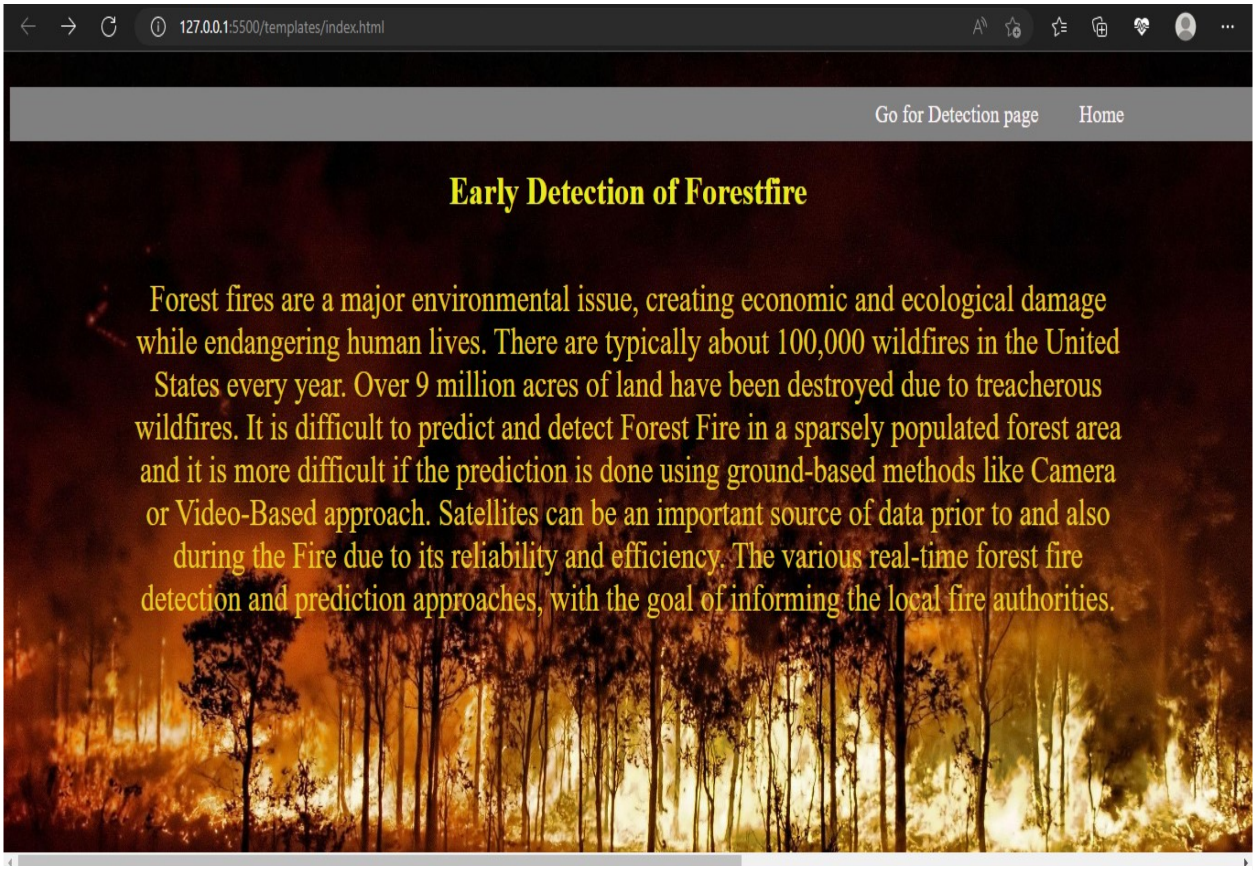
```
prediction(r'Dataset\Dataset\test_set\forest\0.64133000_1519374442_forest_deep.jpg')
```

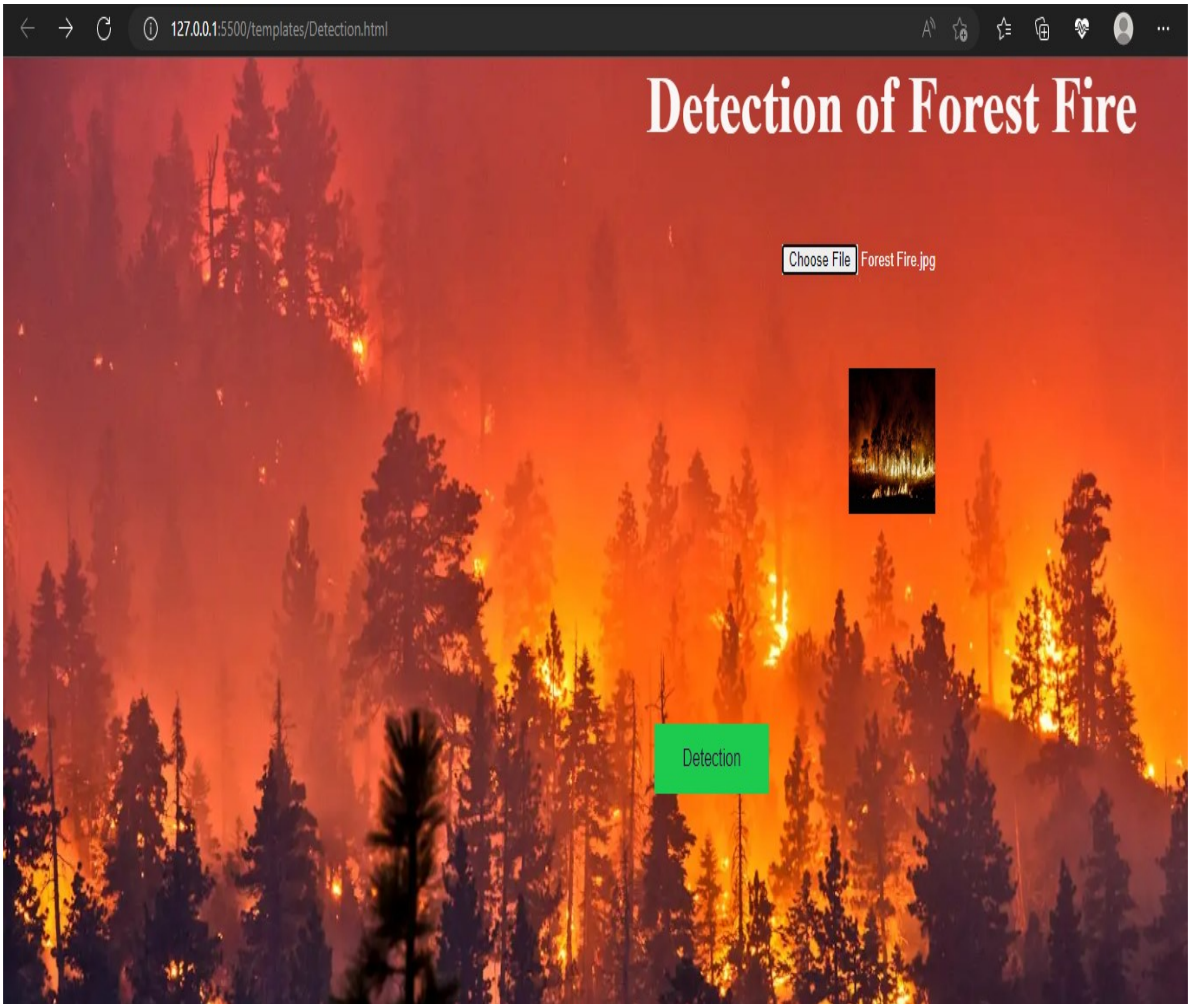
NO FOREST FIRE DETECTED



```
In [11]: prediction(r'Dataset/Dataset/test_set/with fire/599857.jpg')
```

```
FOREST FIRE DETECTED! SMS SENT!
```


127.0.0.1:5500/templates/index.html
Go for Detection page
Home
Early Detection of Forestfire
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 prior to 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.


127.0.0.1:5500/templates/Detection.html
Detection of Forest Fire
Choose File
Forest Fire.jpg
Detection

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

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

DISADVANTAGES

The electrical interference diminishes the effectiveness of radio receiver.

The main drawback is that it has less coverage range areas.

CHAPTER 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 immediately 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 environmental, cultural heritage damages to a great extent. Therefore the most important goals in fire surveillance are quick and reliable detection of fire. It is so much easier to suppress fire while it is in its early stages. Information about progress of fire is highly valuable for managing fire during all its stages. Based on this information the firefighting staff can be guided on target to block fire before it reaches cultural heritage sites and to suppress it quickly by utilizing required firefighting equipment and vehicles. With further research and innovation, this project can be implemented in various forest areas so that we can save our forests and maintain great environment.

CHAPTER 12

FUTURE SCOPE

This project is far from complete and there is a lot of room for improvement. Some of the improvements that can be made to this project are as follows:
Additional pump can be added so that itautomatically 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. Implementing this concept in the real world will benefit several industries and reduce the workload on many workers, enhancing overall work

APPENDIX

SOURCE CODE

app.py 9+ ×

app.py > ...

```python
import numpy as np
from PIL import Image
import os
from flask import Flask, request, render_template, url_for,redirect
from werkzeug.utils import secure_filename, redirect
from gevent.pywsgi import WSGIServer
from keras.models import load_model
import cv2
from keras.preprocessing import image
from tensorflow.keras.preprocessing import image
from flask import send_from_directory

FOLDER ='static/upload'
app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = FOLDER

model = load_model("Forest_Fire.h5")

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/Detection', methods=['GET', 'POST'])
def Detection():
    if request.method == 'POST':
        return redirect(url_for('index.html'))
    return render_template('Detection.html')
@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == "POST":
        f = request.files["image"]
        filepath = secure_filename(f.filename)
```

app.py 9+ ×

app.py > ...

```python

@app.route('/Detection', methods=['GET', 'POST'])
def Detection():
    if request.method == 'POST':
        return redirect(url_for('index.html'))
    return render_template('Detection.html')
@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == "POST":
        f = request.files["image"]
        filepath = secure_filename(f.filename)
        f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))
        uploading_img = os.path.join(FOLDER, filepath)
        img = Image.open(uploading_img).convert("L")
        x=image.img_to_array(img)
        res=cv2.resize(x,dsize=(64,64),interpolation=cv2.INTER_CUBIC)
        #expand the image shape
        x=np.expand_dims(res,axis=0)
        pred=model.predict(x)
        pred = int(pred[0][0])
        pred
        pred1=int(np.argmax(pred))
        #if pred==0:
            #print('Forest fire')
        #elif pred==1:
           # print('No Fire')
        return render_template('predict.html',pred=pred1)

if __name__ == '__main__':

    app.run(debug=False)

```

Ln 54, Col 1 Spaces: 4 UTFapp.py 9+ index.html ×

templates > index.html > html

```
ml>
 <head>
     <title>Early Detection OF Forestfire</title>
     <style>
          .top{
             display: inline-block;
             text-decoration: none;
             text-decoration-line: none;
             padding-left: 70%;
             margin-top: 20px;
             font-size: 18px;
             width:100%;
             background-color: gray;
             padding-top: 10px;
             padding-bottom: 10px;
         }
         .home{
             padding-right: 40px;
         }
         .home, .reg{
             color: rgb(252, 244, 244);
             text-decoration: none;
             font-size: 20px;
         }
         .heading{
             text-align: center;
             color: rgb(234, 234, 29);
         }
         body{
             background-image: url("https://wallpaperaccess.com/full/437922.jpg");
             background-size: 100%;
         }
```

app.py 9+ <> Detection.html ×

templates > <> Detection.html > html > head > style > .button

```
<html>
    <head>
        <title>Detection page</title>
        <style>
            body {
             background-image: url('https://i.insider.com/560aeee19dd7cc16008bde7c?width=1416');
             background-size: 100%;
             margin: 0;
             font-family: 'Times New Roman', Times, serif, Helvetica, sans-serif;
             height: 100%;
             width: 100%;
            }
            h1 {
                display: block;
                font-size: 3.5em;
                margin-top: 0em;
                margin-bottom: 0em;
                margin-left: 50%;
                margin-right: 0;
                font-weight: bold;
                text-align: center;
            }


            .button {
                border: #e5b9f3;
                color: rgb(56, 1, 69);
                padding: 15px 32px;
                text-align: center;
                text-decoration: none;
                display: inline-block;
                font-size: 16px;
```

Github link:

https://github.com/IBM-EPBL/IBM-Project-41567-1660643004

Demo link:

https://drive.google.com/file/d/1XkRcvO8eZFJyji8oPIZFVSo25E9ujecC/view?usp=sharing