Emerging Methods for Early Detection of Forest Fires

A PROJECT REPORT

Submitted by

SANTHANA SELVAM CHANDRU

HARISH RAGAVENDAR

VIKNESWAR

SURESH RAJA PERUMAL

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

KINGS ENGINEERING COLLEGE

DEPARTMENT OF INFORMATION TECHNOLOGY

IRUNGATTUKOTTAI-602117

NOV 2022

BONAFIDE CERTIFICATE

Certified that this project report "**Emerging Methods for Early Detection of Forest Fires**" is the bonafide work of **SANTHANA SELVAM CHANDRU, HARISH RAGAVENDAR, VIKNESWAR, SURESH RAJA PERUMAL** who carried out the project work under my supervision.Certified further that the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this on any other candidate.

Signature

Dr.G.MANIKANDAN M.E.,PhD.

HEAD OF THE DEPARTMENT

DEPARTMENT OF INFORMATION TECHNOLOGY

KINGS ENGINEERING COLLEGE

IRUNGATUKOTTAI

CHENNAI – 602117

Signature

INDUSTRY MENTOR SHANTHI

FACULTY MENTOR J BRISHO BECLAY BELL

SUPERVISOR

DEPARTMENT OF INFORMATION TECHNOLOGY

KINGS ENGINEERING COLLEGE

IRUNGATTUKOTTAI

CHENNAI - 602117

ACKNOWLEDGEMENT

We thank God for his blessings and also for giving as good knowledge and strength in enabling us to finish our project. Our deep gratitude goes to our founder late Dr. D. SELVARAJ, M.A., M.Phil., for his patronage in the completion of our project. We like to take this opportunity to thank our honourable chairperson Dr.S. NALINI SELVARAJ, M.COM., MPhil., Ph.D. and honourable director, MR.S. AMIRTHARAJ, M.Tech., M.B.A for their support given to us to finish our project successfully. We wish to express our sincere thanks to our beloved principal. Dr.T. JOHN ORAL BASKAR., M.E., Ph.D for his kind encouragement and his interest towards us.

We are extremely grateful and thanks to our professor Dr.G.MANIKANDAN, head of the Information Technology department, Kings Engineering College, for his valuable suggestion, guidance and encouragement. We wish to express our sense of gratitude to our project Mentor Mrs. K. SARANYA, Assistant Professor of Information Technology Department, Kings Engineering College and our project Evaluator Mr. B. MUTHAZAGHAN, Associate Professor of Information Technology Department, Kings Engineering College, with their guidance and direction made our project a grand success. Also we thank our industrial Mentor Mr. X. YYYYYYYYYY, for tuning this project as remarkable one. We Also Thank our College Experimental Project Based Learning SPOC Dr. Mrs. M. PARAMESHWARI for managing this initiative in correct direction. We express our sincere thanks to our parents, friends and staff members, who have helped and encouraged us during the entire course of completing this project work successfully.

1. **INTRODUCTION**
    1. Project Overview
    2. Purpose
2. **LITERATURE SURVEY**
    1. Existing problem
    2. References
    3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    1. Empathy Map Canvas
    2. Ideation & Brainstorming
    3. Proposed Solution
    4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    1. Functional requirement
    2. Non-Functional requirements
5. **PROJECT DESIGN**
    1. Data Flow Diagrams
    2. Solution & Technical Architecture
    3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
    1. Sprint Planning & Estimation
    2. Sprint Delivery Schedule
    3. Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    1. Feature 1
    2. Feature 2
    3. Database Schema (if Applicable)
8. **TESTING**
    1. Test Cases
    2. User Acceptance Testing
9. **RESULTS**
    1. Performance Metrics

10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

    GitHub & Project Demo Link

1. INTRODUCTION

1. Project Overview

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 Fires in a sparsely populated forest area. It is more complicated if the prediction is done using ground-based methods like a Camera or Video-Based approach. Satellites can be an important source of data prior to and during the Fire due to their reliability and efficiency. The various real-time forest fire detection and prediction approach, with the goal of informing the local fire authorities.

2. Purpose

We connect our application to a satellite then the satellite camera captures the forest if the camera detects any firing symptoms it allots the server to find the location. We detect the firing symptoms accurately. It will give a solution for forest firing damages. We respond to satellite alerts immediately to take action on a forest fire.

2. LITERATURE SURVEY

1. Existing problem

Forest and urban fires have been and still are serious problems for many countries in the world. Currently, there are many different solutions to fight forest fires. These solutions mainly aim to mitigate the damage caused by the fires, using methods for their early detection. In this paper, we discuss a new approach to fire detection and control, in which modern technologies are used.

2. References

1. FOREST FIRE DETECTION USING MACHINE LEARNING, Georgie Vadakkadathu Rajan MSc. Data Science and Artificial Intelligence Bournemouth University, Poole, England, s5333155@bournemouth.ac.uk

2. EARLY FOREST FIRE DETECTION USING DRONES AND ARTIFICIAL INTELLIGENCE, Diyana Kinaneva, Georgi Hristov, Jordan Raychev and Plamen Zahariev

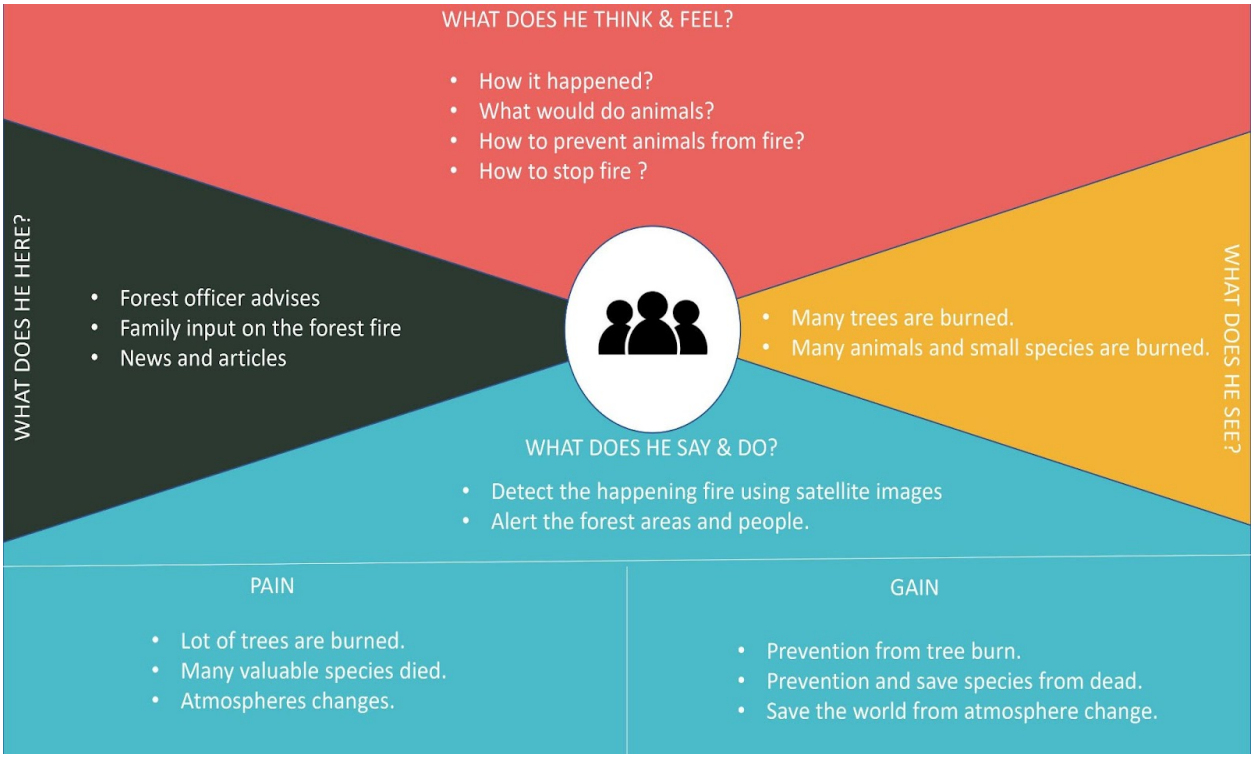
University of Ruse, Department of Telecommunications, Ruse, Bulgaria {dkyuchukova, ghristov, jraychev, pzahariev}@uni-ruse.bg

3. Problem Statement Definition

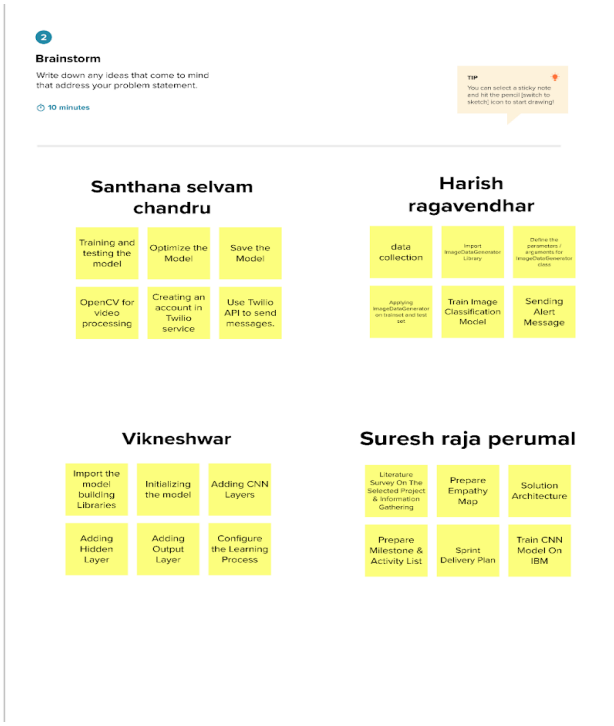
. The Customer Problem Statement template helps you focus on what matters to create experiences people will love. A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

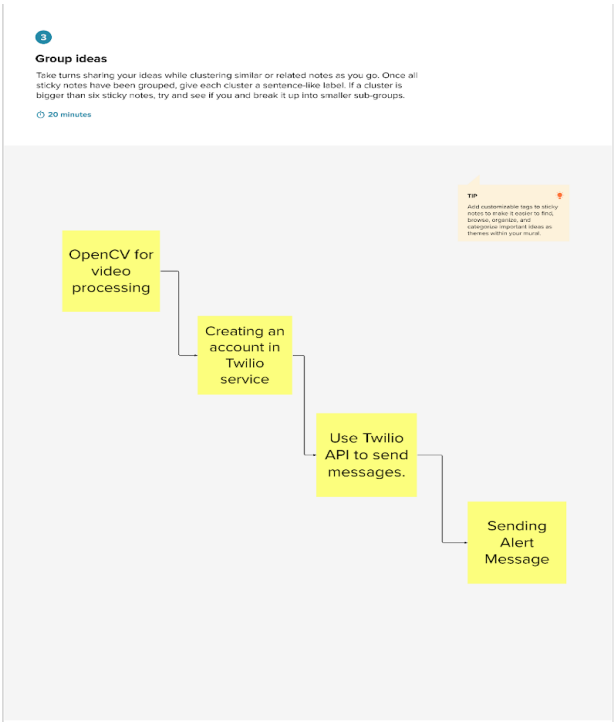
3. IDEATION & PROPOSED SOLUTION

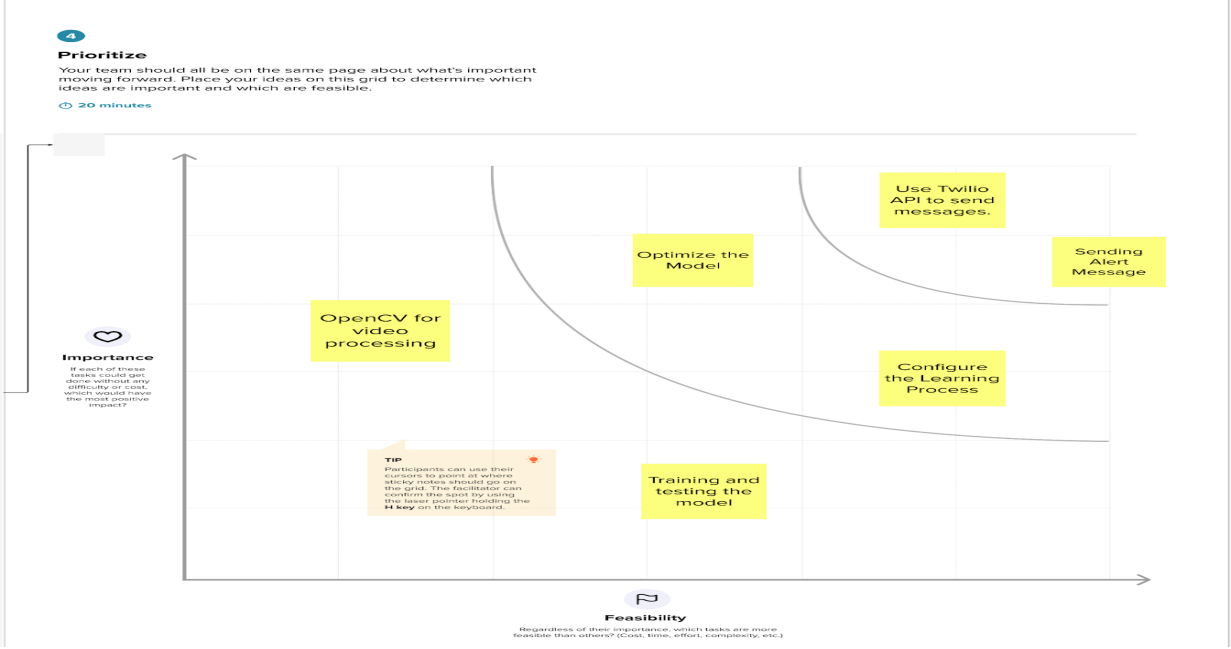
1. Empathy Map Canvas



2. Ideation & Brainstorming







3. Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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.
2.	Idea / Solution description	We connect our application in to satellite then satellite camera capturing forest if camera detects any firing symptoms it allots the server

		to find the location.
3.	Novelty / Uniqueness	We detect the firing symptoms accurately.
4.	Social Impact / Customer Satisfaction	It will give solution for forest firing damages
5.	Scalability of the Solution	We response the satellite alerts immediately to take action on forest fire.

4. Problem Solution fit

4. **REQUIREMENT ANALYSIS**

1. Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Gmail Registration through Google authentication
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User can demo the application if fire detects or not using	Demonstrate the application

	sample video	
FR-4	User can monitor the forest using CCTV camera	Access camera using CCTV
FR-5	User can receive fire alert	If the fire is detected alert through Twilio API

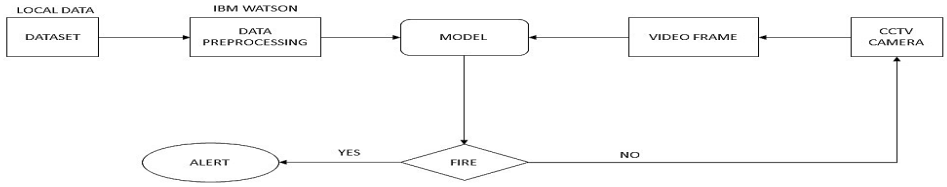
2. Non-Functional requirements

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	Easy to use and feature are easy to use
NFR-2	**Security**	Must protect the user data
NFR-3	**Reliability**	Able to produce accurate results
NFR-4	**Performance**	Immediately sends the alert message without delay
NFR-5	**Availability**	Application runs 24/7 time
NFR-6	**Scalability**	Application is easily modifiable and increase the accuracy and reduce time

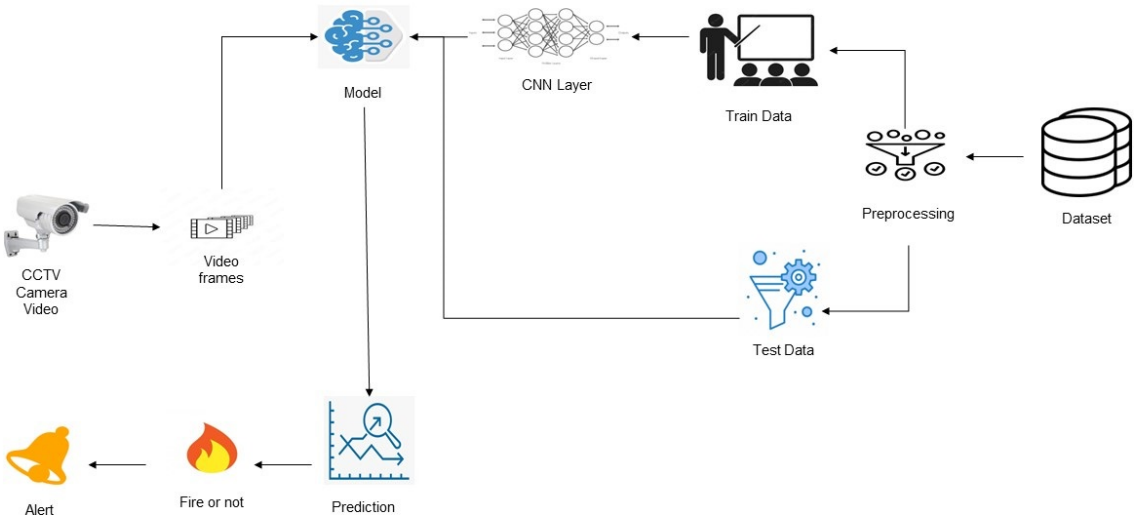
5. PROJECT DESIGN

1. Data Flow Diagrams

DATAFLOW DIAGRAM



2. Solution & Technical Architecture



3. User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release

Customer (Forest Officer, Tourister, and Peoples)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account/dashboard	High	Sprint-1
Customer (Forest Officer, Tourister, and Peoples)	Registration	USN-2	As a user, I will receive a confirmation email once I have registered for the application	I can receive a confirmation email & click confirm	High	Sprint-1
Customer (Forest Officer, Tourister, and Peoples)	Registration	USN-3	As a user, I can register for the application through Google	I can register & access the dashboard with Google authentication Login	Low	Sprint-2
Customer (Forest Officer, Tourister, and Peoples)	Login	USN-5	As a user, I can log into the application by entering email & password	I can log in directly through email and password	High	Sprint-1
Customer (Forest Officer, Tourister, and Peoples)	Login	USN-6	As a user, I can log into the application by using google authentication	I can log in now through google authentication	High	Sprint-2
Customer (Forest Officer, Tourister, and Peoples)	Dashboard	USN-7	As a user, I can monitor the forest using CCTV camera	I can access camera using CCTV	High	Sprint-3

Customer (Forest Officer, Tourister, and Peoples)	Dashboard	USN-8	As a user, I can demonstrate the detection fire using sample video	I can check whether an application detects the fire or not	High	Sprint-3
Customer (Forest Officer, Tourister, and Peoples)	Alert	USN-9	If the fire is detected alert through Twilio API	Application notice the fire	High	Sprint-4
Administrator	Proctor	USN-10	As a admin, I can proctor manually and check confirmation of fire alert send to cusotmer	API sends the alert msg and confirmation received by customer	High	Sprint-5

6. PROJECT PLANNING & SCHEDULING

1. Sprint Planning & Estimation

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.	5	High	Harish Ragavendhar
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	4	High	Vikneswar
Sprint-1		USN-4	As a user, I can register for the application through Gmail	4	Medium	Harish Ragavendha
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email &	4	High	Vikneswar

			password			
Sprint-2	Dashboard	USN-7	As a user, I can monitor the forest using CCTV camera	10	Medium	Santhana selvam chandru, Harish Ragavendhar
Sprint-2		USN-8	As a user, I can demonstrate the detection fire using sample video	10	High	Santhana selvam chandru, Harish Ragavendhar
Sprint-3	Alert	USN-9	If the fire is detected alert through Twilio API	15	High	Vikneswar, Suresh raja perumal
Sprint-4	Proctor	USN-10	As a admin, I can proctor manually and check confirmation of fire alert send to cusotmer	20	High	Santhana selvam chandru, Harish Ragavendhar

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	19	6 Days	24 Oct 2022	29 Oct 2022	19	29 Oct 2022
Sprint-2	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-3	15	6 Days	14 Nov 2022	19 Nov 2022	15	19 Nov 2022
Sprint-4	20	6 Days	21 Nov 2022	28 Nov 2022	20	28 Nov 2022

3. Reports from JIRA

7. CODING & SOLUTION

app.py

```
# Import Necessary libraries
from datetime import timedelta
from flask import Flask, render_template, request, redirect, session, flash,
Response
from flask_sqlalchemy import SQLAlchemy
from flask_session import Session
from werkzeug.utils import secure_filename
import os
import cv2
from tensorflow.keras.preprocessing import image
import numpy as np
from keras.models import load_model
from twilio.rest import Client
from pygame import mixer

app = Flask(__name__)
UPLOAD_FOLDER = os.path.join('static', 'uploads')
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
app.secret_key = 'admin'

# Database connection
```

```
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///user.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)

# user cache
app.permanent_session_lifetime = timedelta(minutes=5)
app.config['SESSION_TYPE'] = 'filesystem'
Session(app)

with app.app_context():
    db.create_all()

# global camera object
camera = None

# Database Model
class Registration(db.Model):
    _id = db.Column('id', db.Integer, primary_key=True)
    name = db.Column(db.String(25), nullable=False)
    email = db.Column(db.String(120), nullable=False)
    password = db.Column(db.String(16), nullable=False)
    def __repr__(self):
        return f"Name : {self.name} Mail :{self.email}"

# Home Page
@app.route('/', methods=['GET'])
def home():
    if camera:
        camera.release()
    if session.get('name'):
        return render_template('home.html', link_image_video = True)
```

```
    else:
        return redirect('/signup')
# signup page
@app.route('/signup', methods=['POST', 'GET'])
def signup():
    if request.method == 'POST':
        name = request.form['uname']
        email = request.form['email']
        pwd = request.form['pwd']
        user = Registration.query.filter_by(name=name).first()
        if user is not None:
            return render_template('index.html', msg='Already Registered')
        user = Registration(name=name, email=email, password=pwd)
        db.session.add(user)
        db.session.commit()
        return render_template('index.html', msg='Successfully Registered')
    return render_template('index.html')

# login page
@app.route('/login', methods=['POST'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        pwd = request.form['pwd']
        user = Registration.query.filter_by(email=email).first()
        if user is None:
            return render_template('index.html', msg="Invalid user!")
        if user.password != pwd:
            return render_template('index.html', msg="Invalid Password!")
        session['name'] = user.name
```

```
        return redirect('/')
    return redirect('/signup')

# Logout Page
@app.route('/logout')
def logout():
    session['name'] = None
    return redirect('/')


# About Page
@app.route('/about')
def about():
    if camera:
        camera.release()
    return render_template('about.html')

def detect(pred):
    li = ['Not Fire', 'Fire']
    return li[pred]

model = load_model(r'./static/model/ForestDetectionModel.h5')

def image_prediction(img_file):
    img = image.load_img(img_file, target_size=(150, 150))
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
    pred = model.predict(x)
    pred = detect(int(pred))
    return pred
```

```
def video_prediction(video_file):
    cap = cv2.VideoCapture(video_file)
    WIDTH = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    HEIGHT = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
    FPS = 1
    fourcc = 0x00000021
     output = cv2.VideoWriter("./static/uploads/videos/output.mp4", fourcc,
FPS, (WIDTH, HEIGHT))
    def getFrame(sec):
        cap.set(cv2.CAP_PROP_POS_MSEC, sec * 1000)
        hasFrames, frame = cap.read()
        if hasFrames:
            cv2.imwrite('./static/uploads/images/pic.jpg', frame)
                    img = image.load_img('./static/uploads/images/pic.jpg',
target_size=(150, 150))
            img = image.img_to_array(img)
            img = np.expand_dims(img, axis=0)
            pred = model.predict(img)
            if pred[0] == 1:
                cv2.circle(frame, (480, 55), 10, (0, 0, 255), -1)
                                cv2.putText(frame, 'Fire Alert', (500, 60),
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2, cv2.LINE_AA)
            output.write(frame)
        return hasFrames
    sec = 0
    frameRate = 1
    count = 1
    success = getFrame(sec)
    while success:
```

```
        count = count + 1
        sec = sec + frameRate
        sec = round(sec, 2)
        success = getFrame(sec)
    output.release()
    cap.release()
    return 1

# Check the image and video extensions
def allowed_file(filename):
    image_extensions = {'jpg', 'jpeg', 'png', 'gif', 'webp'}
    video_extensions = {'webm', 'mp4', 'mov', 'avi', 'mkv'}
    ex = filename.rsplit('.', 1)[1].lower()
    if ex in image_extensions:
        return 'images'
    if ex in video_extensions:
        return 'videos'
    return None

# Predict Forest Fire using images and videos
@app.route('/predict', methods=['POST', 'GET'])
def predict():
    if request.method == 'POST':
        file = request.files['Pred_file']
        if not file:
            flash('No File Selected', 'image_video')
            return redirect('/')
        if not allowed_file(file.filename):
            flash('Invalid File Format', 'image_video')
            return redirect('/')
```

```
        if file and allowed_file(file.filename) == 'images':
            img_filename = secure_filename(file.filename)
                path = os.path.join(app.config['UPLOAD_FOLDER'], 'images',
img_filename)
            file.save(path)
            imageprediction = image_prediction(path)
                    return render_template('home.html', img=img_filename,
img_prediction=imageprediction, link_image_video = True)
        if file and allowed_file(file.filename) == 'videos':
            vdo_filename = secure_filename(file.filename)
                path = os.path.join(app.config['UPLOAD_FOLDER'], 'videos',
vdo_filename)
            file.save(path)
            videoprediction = video_prediction(path)
                                    return render_template('home.html',
video_prediction=videoprediction, link_image_video = True)
    return redirect('/')

# send a message to alert fire
def alert_messge():
    account_sid = 'AC68305e6170b2901e2d784239fb9fca5c'
    auth_token = '55e44f3c92c8e9bc70f77f5345afe7b5'
    client = Client(account_sid, auth_token)
    message = client.messages.create(
        body='Fire Alert Please Be Safe',
        from_='+14246881439',
        to='+919025937756',
    )
    return True
```

```
# Generate and predict using  each frames in realtime camera
def gen_frames():
    send = False
    count = 0
    while True:
        success, frame = camera.read()
        if success:
            try:
                cv2.imwrite('./static/uploads/videos/pic.jpg', frame)
                     img = image.load_img('./static/uploads/videos/pic.jpg',
target_size=(150, 150))
                img = image.img_to_array(img)
                img = np.expand_dims(img, axis=0)
                pred = model.predict(img)
                if pred[0][0] == 1.0:
                    cv2.circle(frame, (480, 55), 10, (0, 0, 255), -1)
                                cv2.putText(frame, 'Fire Alert', (500, 60),
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2, cv2.LINE_AA)
                    mixer.init()
                    sound = mixer.Sound('./static/sound/alert1.ogg')
                    sound.play()
                    if not send:
                        send = alert_messge()
                ret, buffer = cv2.imencode('.jpg', frame)
                frame = buffer.tobytes()
                yield (b'--frame\r\n'
                            b'Content-Type: image/jpeg\r\n\r\n' + frame +
b'\r\n')
            except:
```

```
            pass

# render to cv camera to html file
@app.route('/video_feed')
def video_feed():
       return  Response(gen_frames(),  mimetype='multipart/x-mixed-replace;
boundary=frame')

# Realtime camera page
@app.route('/camera_pred')
def camera_pred():
   global camera
   camera = cv2.VideoCapture(0)
         return   render_template('/home.html',   camera_start   =   True,
camera_prediction=True)


# camera stop method
@app.route('/camera_stop')
def camera_stop():
   camera.release()
         return   render_template('/home.html',   camera_start   =   False,
camera_prediction=True)

if __name__ == '__main__':
   app.run(debug=True)
```

8. TESTING

1. Test Cases

Date	3-Nov-22
Team ID	PNT2022TMID25532
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
Register	Functional	Register Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	Mail Id,password	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	http://127.0.0.1:5000/signup	Login/Signup popup should display	Working as expected	Pass	Steps are follow properly			
RegisterPage	UI	Register Page	Verify the UI elements in Login/Signup popup	Mail Id,password	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link e.Last password? Recovery password link	http://127.0.0.1:5000/signup	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link	Working as expected	Fail	Steps are not clear to follow		BUG-1	
LoginPage_TC_OO3	Functional	Login page	Verify user is able to log into application with Valid credentials	Mail Id,password	1.Enter URL(http://127.0.0.1:5000/signup) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: chalam@gmail.com password: Testing123	User should navigate to user account homepage	Working as expected	Pass	Steps are follow properly			
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials	Mail Id,password	1.Enter URL(http://127.0.0.1:5000/signup) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: chalam@gmail password: Testing123	Application should show "Incorrect email or password ' validation message.	Working as expected	Fail	Steps are not clear to follow		BUG-2	
Login	UI	Login page	Verify user is able to log into application with InValid credentials	Mail Id,password	1.Enter URL(http://127.0.0.1:5000/signup) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: chalam@gmail.com password: Testing1236786867868876876	Application should show "Incorrect email or password ' validation message.	Working as expected	Fail	Steps are not clear to follow		BUG-3	
LoginPage	Functional	Login page	Verify user is able to log into application with InValid credentials	Mail Id,password	1.Enter URL(http://127.0.0.1:5000/signup) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: chalam password: Testing1236786867868876876	Application should show "Incorrect email or password ' validation message.	Working as expected	Fail	Steps are not clear to follow			
Home	Functional	Dashboard	verify a user is able to predict and monitor the forest fire detection	web camera, videos, images	1.Enter upload image 2.click predict 3.show the prediction of forest fire	Forest with fire image	detect the fire	Working as expected	Pass	Steps are follow properly			
Home	Functional	Dashboard	verify a user is able to predict and monitor the forest fire detection	web camera, videos, images	1.Enter upload image 2.click predict 3.show the prediction of forest fire	Forest with fire image	detect the fire	Working as expected	Pass	Steps are follow properly			
Home	Functional	Dashboard	verify a user is able to predict and monitor the forest fire detection	web camera, videos, images	1.Enter upload video 2.click predict 3.show the prediction of forest fire	Forest with fire video	detect the fire	Working as expected	Pass	Steps are follow properly			
Home	Functional	Dashboard	verify a user is able to predict and monitor the forest fire detection	web camera, videos, images	1.Enter upload video 2.click predict 3.show the prediction of forest fire	Forest without fire video	detect the fire	Working as expected	Pass	Steps are follow properly			
Home	Functional	Dashboard	verify a user is able to predict and monitor the forest fire detection	web camera, videos, images	1.click camera 2.show the prediction of forest fire using web camera	Forest with fire camera feed	detect the fire	Working as expected	Pass	Steps are follow properly			
Home	Functional	Dashboard	verify a user is able to predict and monitor the forest fire detection	web camera, videos, images	1.click camera 2.show the prediction of forest fire using web camera	Forest without fire camera feed	detect the fire	Working as expected	Pass	Steps are follow properly			

2. User Acceptance Testing\

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	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

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
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. **RESULTS**

    1. Performance Metrics

S.No.	Parameter	Values	Screenshot
1.	Model Summary	**Total params: 1,212,513** **Trainable params: 1,212,513** **Non-trainable params: 0**	https://drive.google.com/file/d/10c7E3OTC8oFVfCmspz2FxcdQfymPOrOm/view?usp=sharing
2.	Accuracy	Training Accuracy - 91.74% Validation Accuracy - 91.74%	https://drive.google.com/file/d/1dJu7y-dz869de3X5nAhx_SuQheejO4gP/view?usp=sharing

```
In [16]: model.summary()
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 148, 148, 32)      896

 activation (Activation)     (None, 148, 148, 32)      0

 max_pooling2d (MaxPooling2D  (None, 74, 74, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 72, 72, 32)        9248

 activation_1 (Activation)   (None, 72, 72, 32)        0

 max_pooling2d_1 (MaxPooling  (None, 36, 36, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 34, 34, 64)        18496

 activation_2 (Activation)   (None, 34, 34, 64)        0

 max_pooling2d_2 (MaxPooling  (None, 17, 17, 64)       0
 2D)

 flatten (Flatten)           (None, 18496)             0

 dense (Dense)               (None, 64)                1183808

 activation_3 (Activation)   (None, 64)                0

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 1)                 65

 activation_4 (Activation)   (None, 1)                 0

=================================================================
Total params: 1,212,513
Trainable params: 1,212,513
Non-trainable params: 0
_________________________________________________________________
```

Fit the model

```
In [9]: model.fit(
            train_generator,
            epochs=10,
            validation_data=val_generator
        )
```

```
Epoch 1/10
28/28 [==============================] - 20s 648ms/step - loss: 0.6372 - acc: 0.7317 - val_loss: 0.1436 - val_acc: 0.9917
Epoch 2/10
28/28 [==============================] - 18s 620ms/step - loss: 0.3401 - acc: 0.8394 - val_loss: 0.1146 - val_acc: 0.9669
Epoch 3/10
28/28 [==============================] - 16s 566ms/step - loss: 0.2715 - acc: 0.8991 - val_loss: 0.0463 - val_acc: 0.9752
Epoch 4/10
28/28 [==============================] - 16s 575ms/step - loss: 0.2348 - acc: 0.8991 - val_loss: 0.0917 - val_acc: 0.9587
Epoch 5/10
28/28 [==============================] - 16s 571ms/step - loss: 0.2262 - acc: 0.9083 - val_loss: 0.0452 - val_acc: 0.9835
Epoch 6/10
28/28 [==============================] - 16s 576ms/step - loss: 0.2792 - acc: 0.9014 - val_loss: 0.0337 - val_acc: 0.9917
Epoch 7/10
28/28 [==============================] - 17s 578ms/step - loss: 0.2249 - acc: 0.9220 - val_loss: 0.0946 - val_acc: 0.9587
Epoch 8/10
28/28 [==============================] - 16s 560ms/step - loss: 0.2202 - acc: 0.9083 - val_loss: 0.0614 - val_acc: 0.9669
Epoch 9/10
28/28 [==============================] - 15s 551ms/step - loss: 0.2301 - acc: 0.9151 - val_loss: 0.0441 - val_acc: 0.9917
Epoch 10/10
28/28 [==============================] - 15s 525ms/step - loss: 0.2143 - acc: 0.9174 - val_loss: 0.1344 - val_acc: 0.9669
```

```
Out[9]: <keras.callbacks.History at 0x1686ed388e0>
```

10. ADVANTAGES

1. Localization: all the previous work used a GPS or fixed the nodes in a known place.
2. Coverage: the nodes deployed randomly a full coverage is almost impossible.
3. Network life span: For sensor nodes working on batteries, it is impossible to go back to each node in the forest and recharge it again.
4. Fire detection method: this is the heart of the application; it should be precise and reliable.

DISADVANTAGES

1. Optical sensors or camera systems in general need to be improved in order to reduce the number of false alarms due to various dynamic phenomena, such as wind-tossed trees, cloud shadows, reflections, and human activity.
2. The difficulties of processing landscape images are due to their varying nature and to the large number of dynamic events that may appear under various illumination conditions, depending on weather, distance, time of day, masking objects, and so forth. These events produce dynamic envelopes, which are not always caused by motion, and consist of time-varying gray levels of connected pixels in several image regions.

3. This kind of technology only provides a line of sight vision; where high trees or the hills and mountains can block the vision; plus it might be impossible to provide images for ignition place.
4. Weather condition and night vision reflect on the camera performance.
5. Finally, these systems are very expensive; the camera tower can be worth more than thirty thousand dollars per tower, and there is a need to build these towers and install a communication infrastructure in the remote areas inside the forests

11. CONCLUSION

From this project we came to the conclusion that the sequential model has a remarkable accuracy of 99% in predicting fires in forest areas. This reduces the chances of false alarm to a great extent. Our system is able to differentiate various forest fire scenarios, from the initial case (no fire) to the detection of fire, fairly accurately. It can accurately determine the growth of the fire. This will help in the early stages of fire detection and help to confine the fire to limited areas before much damage occurs. The system will be very effective in preventing the occurrence of false alarms. We aim at monitoring the forests without constant human supervision.

12. FUTURE SCOPE

This project carries a broad prospective for the future. Moreover it is a need for great research to be done in this field in the coming years. In future, our project can be extended towards finding an efficient way of localization of the fire, gravity of fire, direction of spread, area burnt and many more. In our experiment, the process of simulation of forest fire was done by burning the dried leaves directly. We could come up with ways to make this simulation more close to actual forest fires. Moreover, we can include region.

13. APPENDIX

GitHub: https://github.com/IBM-EPBL/IBM-Project-26518-1660028834

Project Demo Link : https://drive.google.com/file/d/1-uII06BJQA9fb0q5tfX09KyKznHtQwHa/view?usp=sharing