

PROJECT REPORT

Emerging Methods for Early Detection of Forest Fire

TEAM ID : PNT2022TMID43256

Faculty mentor(s) : D.MADESWARAN

Industry Mentor(s) : SANTHI

TEAM SIZE : 4

TEAM LEADER : RAMESHKUMAR.A – 715319104024

TEAM MEMBER : SABARI.M – 715319104025

TEAM MEMBER : MANIKANDAN.R – 715319104014

TEAM MEMBER : PRAVEENKUMAR.N – 715319104022

TABLE OF CONTENT

1. INTRODUCTION

- 1.1 Project Overview
- 1.2 Purpose

2. LITERATURE SURVEY

- 2.1 Existing problem
- 2.2 References
- 2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

- 3.1 Empathy Map Canvas
- 3.2 Ideation & Brainstorming
- 3.3 Proposed Solution
- 3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS

- 4.1 Functional requirement
- 4.2 Non-Functional requirements

5. PROJECT DESIGN

- 5.1 Data Flow Diagrams
- 5.2 Solution & Technical Architecture
- 5.3 User Stories

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 added 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

Source Code

GitHub & Project Demo Link

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 is not enough to monitor all the potential fire prone places. The only way to prevent a fire is to be cautious at 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 which was upgraded 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 $128 \times 128 \times 3$. 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 localization

[2] This paper the authors propose a system that mimics fire the human detection 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 paper 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.

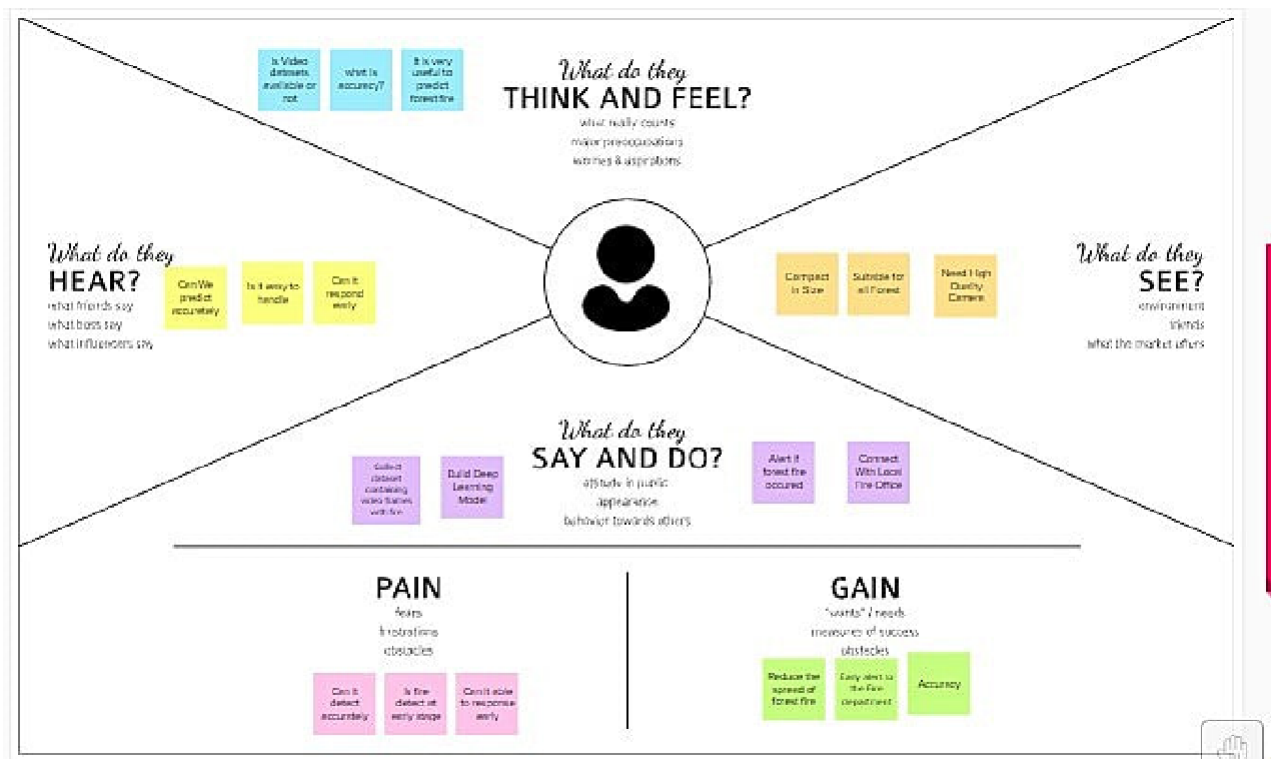
2.3 PROBLEM STATEMENT DEFINITION

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


PROBLEM

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.




Key rules of brainstorming


To run an smooth and productive session




Stay in topic.




Encourage wild ideas.




Defer judgment.



Listen to others.



Go for volume.



If possible, be visual.

SATELITE AND OPTICAL SYSTEM

URGECY OF SITUATION

WIRELESS SENSORS

FINDING THE EXACT LOCATION

veeraniranjana V

RENUGA B

TEMPORAL AND WEATHER VARIABLES

HAVE GREAT IMPACT ON ENVIRONMENT

LIGHTNING HUMAN CARELESS EXTREME HEAT

FIRE LOOKOUT TOWERS

UNMANNED AERIAL VEHICL

DETECTION USING LORIMAN SENSOR

TEMPERATURE DATA DETECTION

OBJECT DETECTION USING AI

monica T

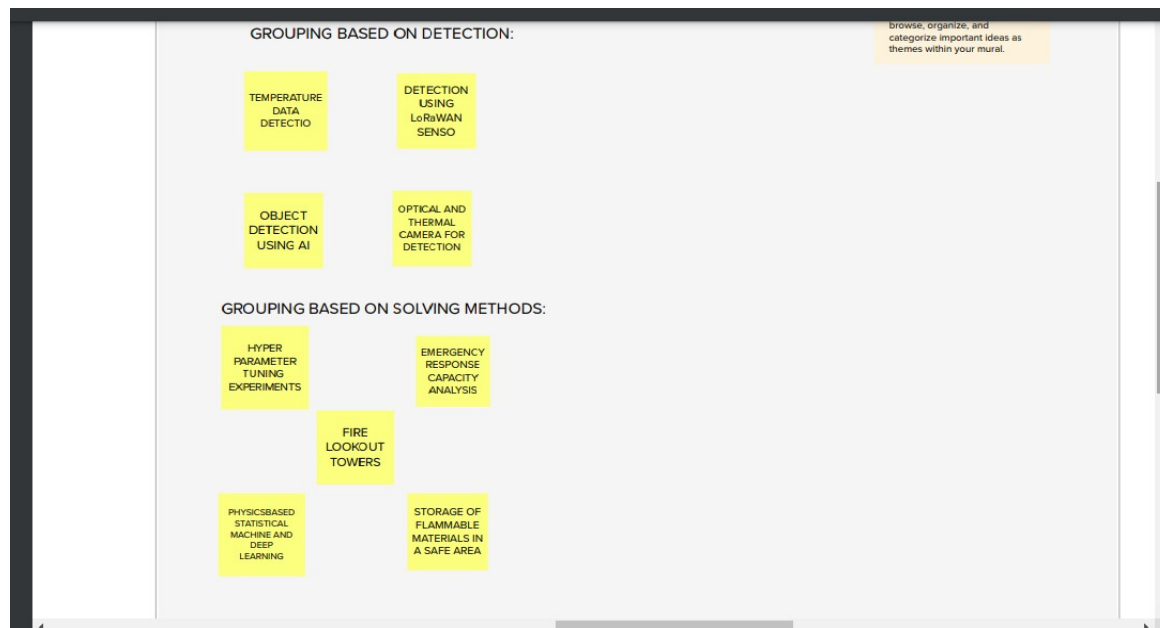
shanmugapriya T

HIGH RESOLUTION CAMERA

HARD TO MAINTAION

FIRE FIGHTERS

OPTICAL AND THERMAL CAMERA FOR DETECTION



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

Define CS, fit into CC	<div>1.CUSTOMER SERVICE</div> <div>CS</div> <div>1.Tribal people and forest department officers living in forest.</div> <div>2.Animals , birds and other living things in the forest.</div>	<div>6. CUSTOMER CONSTRAINTS</div> <div>CC</div> <div>1.Solar power cameras can be used as a power source</div> <div>2. Waterproof cameras.</div> <div>3.Seamless connection.</div>	<div>5. AVAILABLE SOLUTIONS</div> <div>AS</div> <div>1.Notification is sent via messages.</div> <div>2.Fire alarm is activated to nearby stations.</div>	EXplore AS, differentiate
	<div>2. JOBS-TO-BE-DONE / PROBLEMS</div> <div>J&P</div> <div>1.Detecting small fire sparks in forest becomes difficult.</div> <div>2.Camera should always be in motion</div>	<div>9. PROBLEM ROOT CAUSE</div> <div>RC</div> <div>1.Special analysis system can be used.</div> <div>2. Wireless mobile network via SIM can be used transfer alert message throughout areas.</div>	<div>7. BEHAVIOUR</div> <div>BE</div> <div>1.Climate change should be monitored.</div> <div>2. Hot areas should be monitored clearly.</div>	
Focus on J&P, into BE, understand RC				Focus on J&P, tap into BE, understand RC
Identify strong TR & EM	<div>3. TRIGGERS</div> <div>TR</div> <div>1. Correct detection.</div> <div>2. Alarm alert</div> <div>3.Follow correct algorithm</div>	<div>10. YOUR SOLUTION</div> <div>SL</div> <div>1.Mobile application can be developed for specific areas.</div> <div>2.Forest can be monitored by several cameras.</div> <div>3.This can be used in wild life sanctuaries.</div>	<div>8.CHANNELS of BEHAVIOUR</div> <div>CH</div> <div>ONLINE</div> <div>Connected directly to the user via Internet.</div> <div>OFFLINE</div> <div>Alerts can be sent via Offline messages and an alarm system is activated.</div>	Identify strong TR & EM
	<div>4. EMOTIONS: BEFORE / AFTER</div> <div>EM</div> <div>BEFORE</div> <div>1. Unable to detect small sparks.</div> <div>2. camera should always be in motion,</div> <div>AFTER</div> <div>1. Able to detect small sparks.</div> <div>2. 360 view of camera is used.</div>			

Focus on J&P, into BE, understand RC

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

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-FUNCTIONAL REQUIREMENTS:

Following are the non-functional requirements of the proposed solution

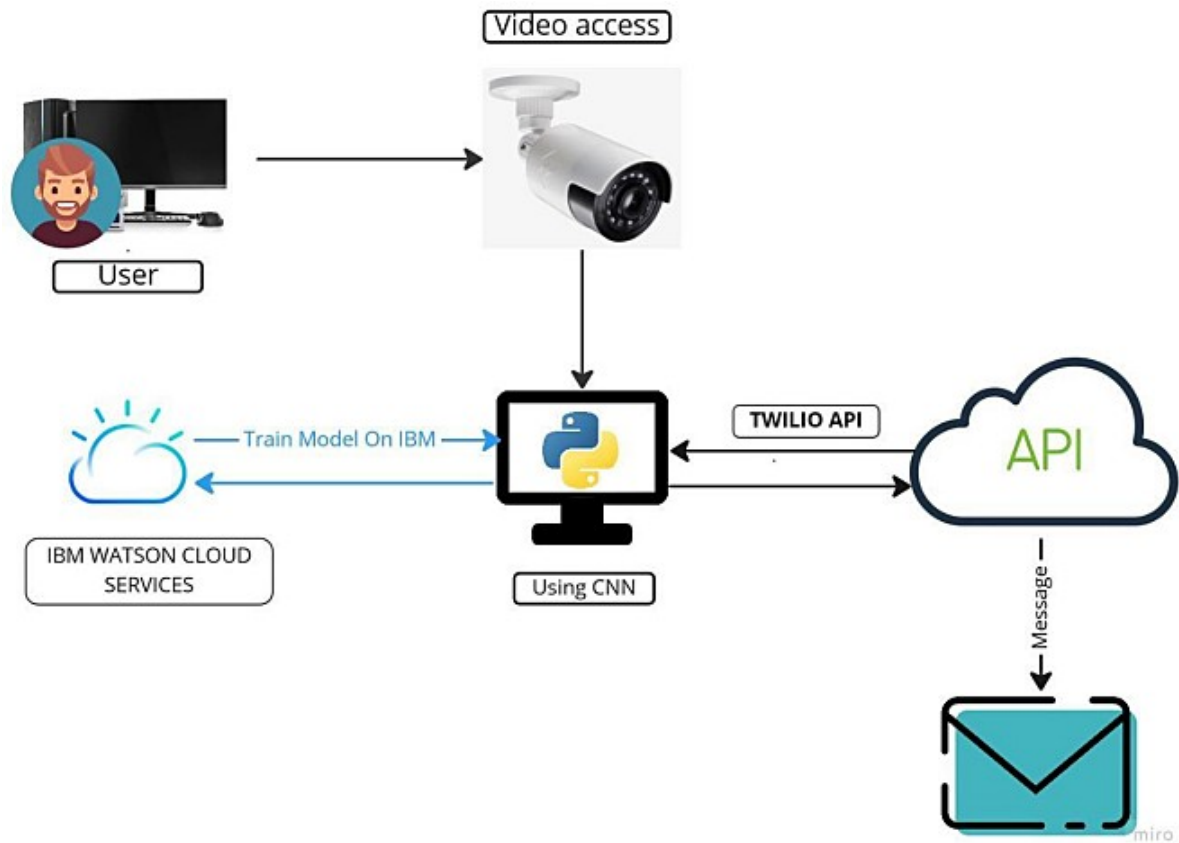
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Model is user friendly to use and very effective.
NFR-2	Security	More secure environment
NFR-3	Reliability	Model is safe to install
NFR-4	Performance	Model will achieve high accuracy
NFR-5	Availability	Build model is available in all the time
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

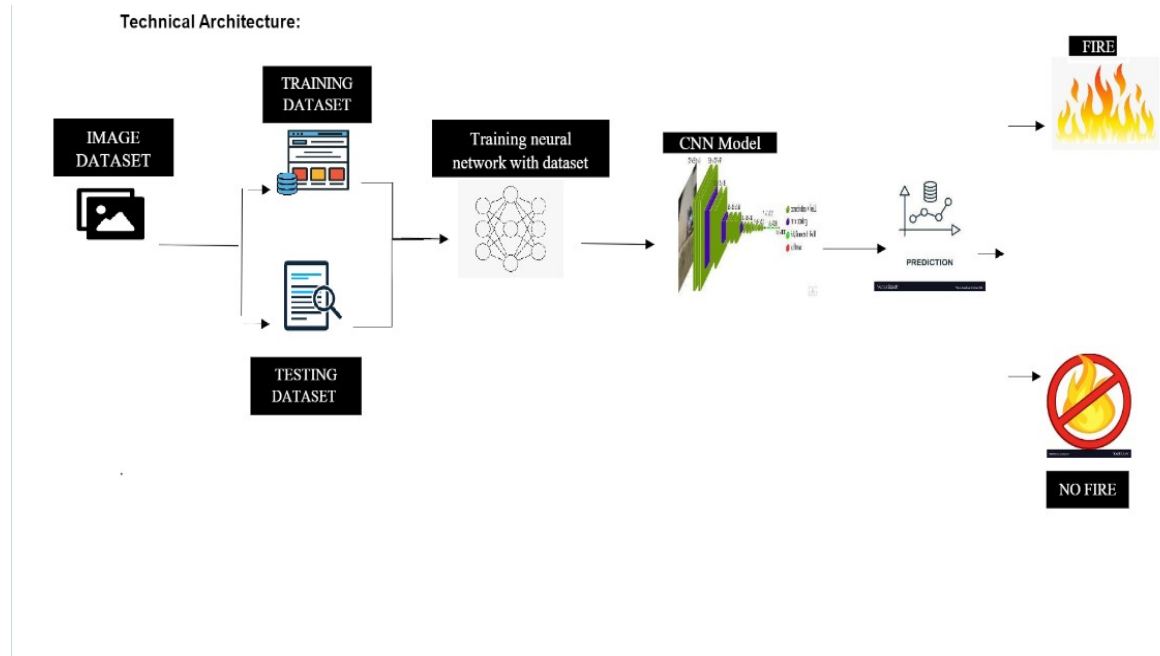
PROJECT

5.1.Data Flow

DESIGN



5.2 SOLUTION & TECHNICAL ARCHITECTURE:



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Environmental ist	Collect the data .	USN-1	As an Environment alist, 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 Requirement(Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Download data set	US N-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	US N-1	<p>In Image processing technique the first step is usually importing the libraries that will be needed in the program.</p> <p>Import Keras library from that library and import the ImageDataGenerator Library to your Python script.</p> <p>The next step is defining the arguments for the ImageDataGenerator. Here the arguments which we are given inside the image data generator class are,</p> <p>rescale, shear_range, rotation range of image, and zoom range that we can consider for images.</p> <p>The next step is applying the ImageDataGenerator arguments to the train and test dataset.</p>	10	High	B.Renuga V.Veeranirajana
Sprint-2	Training image	USN-2	<p>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.</p>	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 StartDate	Sprint EndDate (Planned)	Story Points Completed (as on Planned End Date)	Sprint ReleaseDate (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 Libraries

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

# Data Augmentation
#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 Analysis

```
|: 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:
            print('Forest fire')
```



```

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!

```

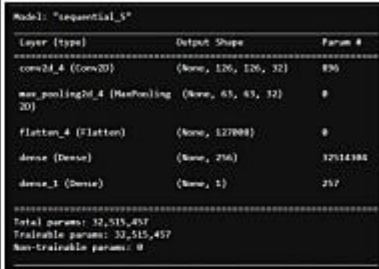
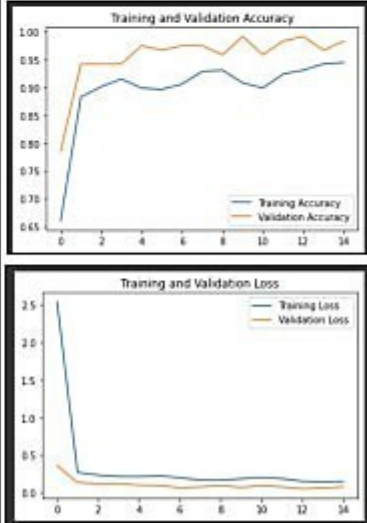
CHAPTER 8

TESTIN

Performance

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	 <pre> 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 </pre>
2.	Accuracy	Training Accuracy – 94.50% Validation Accuracy – 98.35%	 <p>The top graph, titled 'Training and Validation Accuracy', shows accuracy on the y-axis (0.65 to 1.00) against epochs on the x-axis (0 to 14). The Training Accuracy (blue line) starts at approximately 0.65 and rises to about 0.945 by epoch 14. The Validation Accuracy (orange line) starts at approximately 0.75 and rises to about 0.9835 by epoch 14.</p> <p>The bottom graph, titled 'Training and Validation Loss', shows loss on the y-axis (0.0 to 2.5) against epochs on the x-axis (0 to 14). The Training Loss (blue line) starts at approximately 2.5 and drops sharply to about 0.1 by epoch 14. The Validation Loss (orange line) starts at approximately 0.3 and drops to about 0.1 by epoch 14.</p>

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

Page 12 of 12

3. Test Case Analysis

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

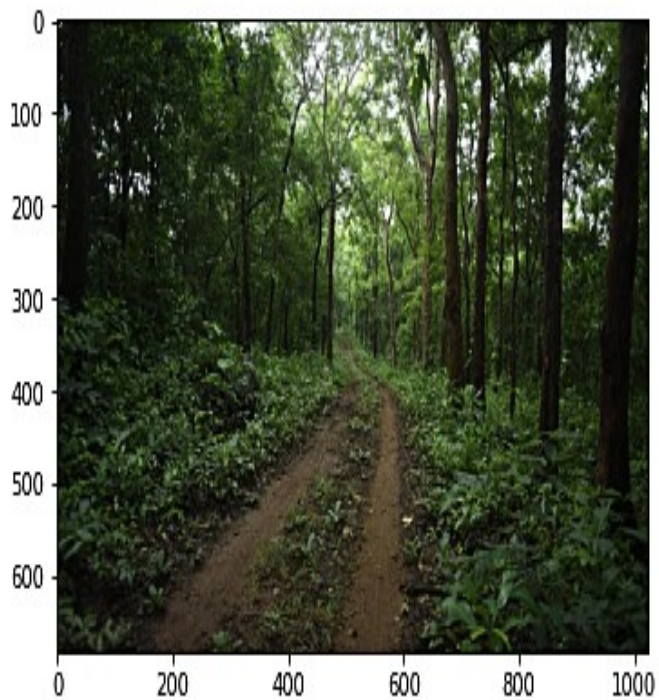
Section	Total Cases	Not Tested	Fail	Pass
Performance	5	0	0	5
UI	1	0	0	1
Security	3	0	0	3

CHAPTER 9 RESULTS

9.1 PERFORMANCE METRICS

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

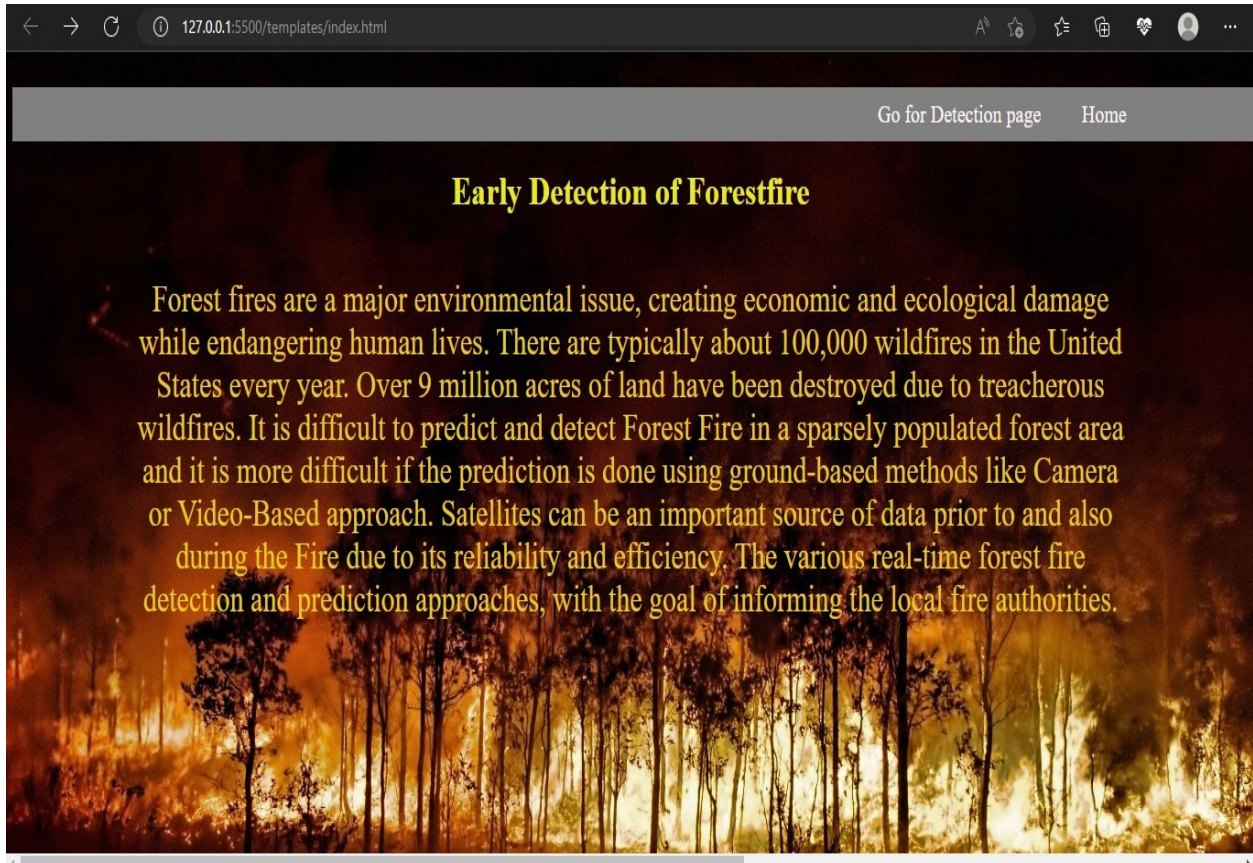
NO FOREST FIRE DETECTED

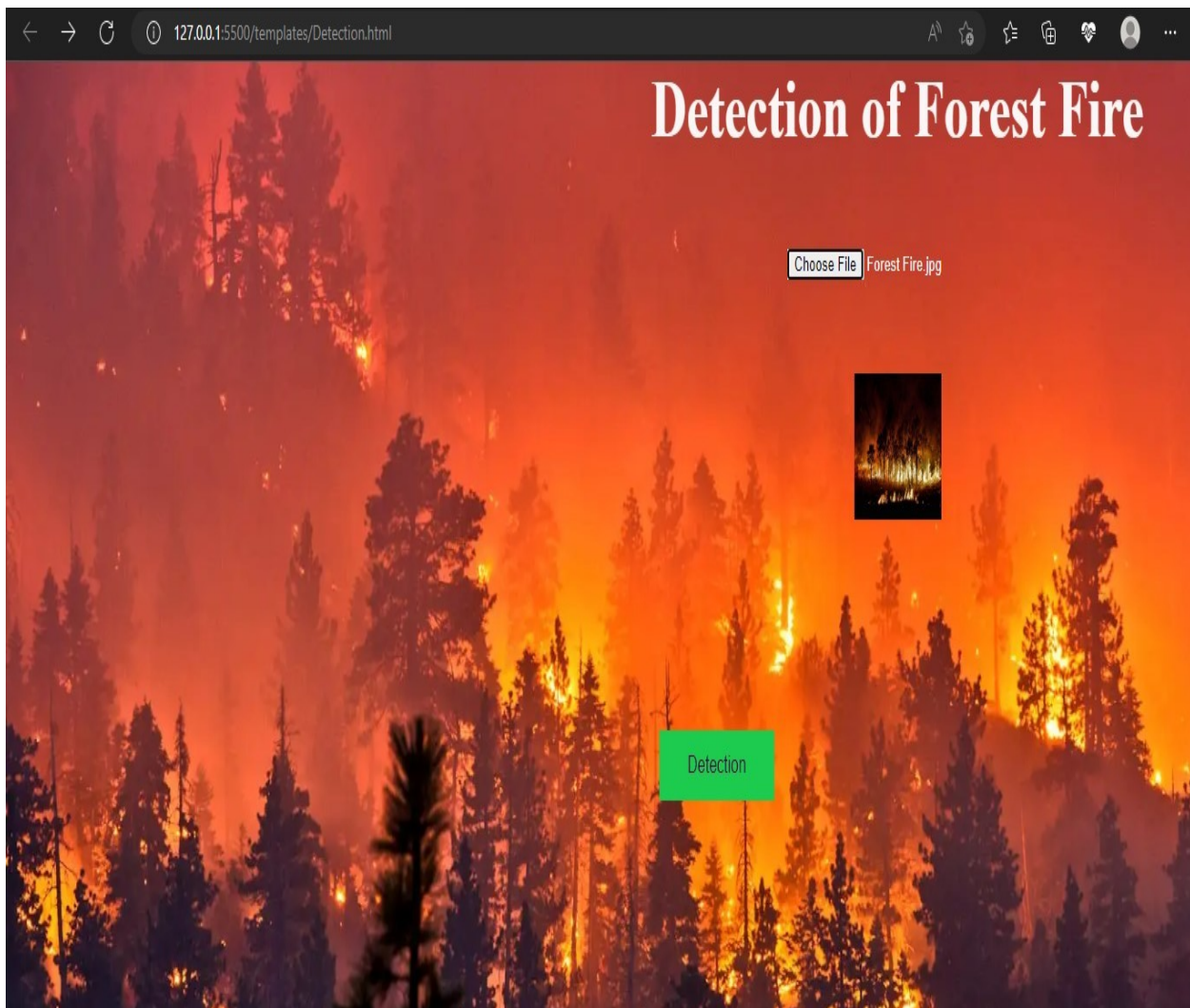


```
In [11]: prediction(r'Dataset/Dataset/test_set/with fire/599857.jpg')
```

FOREST FIRE DETECTED! SMS SENT!







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

CONCLUSIO

N

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 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. 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+ x
app.py > ...
1  import numpy as np
2  from PIL import Image
3  import os
4  from flask import Flask, request, render_template, url_for, redirect
5  from werkzeug.utils import secure_filename, redirect
6  from gevent.pywsgi import WSGIServer
7  from keras.models import load_model
8  import cv2
9  from keras.preprocessing import image
10 from tensorflow.keras.preprocessing import image
11 from flask import send_from_directory
12
13 FOLDER = 'static/upload'
14 app = Flask(__name__)
15 app.config['UPLOAD_FOLDER'] = FOLDER
16
17 model = load_model("Forest_Fire.h5")
18
19 @app.route('/')
20 def index():
21     return render_template('index.html')
22
23 @app.route('/Detection', methods=['GET', 'POST'])
24 def Detection():
25     if request.method == 'POST':
26         return redirect(url_for('index.html'))
27     return render_template('Detection.html')
28 @app.route('/predict', methods=['GET', 'POST'])
29 def upload():
30     if request.method == "POST":
31         f = request.files["image"]
32         filepath = secure_filename(f.filename)
33         f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))

```

app.py 9+ X

app.py > ...

```
22
23 @app.route('/Detection', methods=['GET', 'POST'])
24 def Detection():
25     if request.method == 'POST':
26         return redirect(url_for('index.html'))
27     return render_template('Detection.html')
28 @app.route('/predict', methods=['GET', 'POST'])
29 def upload():
30     if request.method == "POST":
31         f = request.files["image"]
32         filepath = secure_filename(f.filename)
33         f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))
34         uploading_img = os.path.join(FOLDER, filepath)
35         img = Image.open(uploading_img).convert("L")
36         x=image.img_to_array(img)
37         res=cv2.resize(x,dsize=(64,64),interpolation=cv2.INTER_CUBIC)
38         #expand the image shape
39         x=np.expand_dims(res,axis=0)
40         pred=model.predict(x)
41         pred = int(pred[0][0])
42         pred
43         pred1=int(np.argmax(pred))
44         #if pred==0:
45         |   #print('Forest fire')
46         #elif pred==1:
47         |   # print('No Fire')
48         return render_template('predict.html',pred=pred1)
49
50 if __name__ == '__main__':
51
52     app.run(debug=False)
53
54
```

Ln 54, Col 1 Spaces: 4 UTF-



```
app.py 9+  index.html X
templates > index.html > html
1  m1>
2  <head>
3      <title>Early Detection OF Forestfire</title>
4      <style>
5          .top{
6              display: inline-block;
7              text-decoration: none;
8              text-decoration-line: none;
9              padding-left: 70%;
10             margin-top: 20px;
11             font-size: 18px;
12             width:100%;
13             background-color: gray;
14             padding-top: 10px;
15             padding-bottom: 10px;
16         }
17         .home{
18             padding-right: 40px;
19         }
20         .home, .reg{
21             color: rgb(252, 244, 244);
22             text-decoration: none;
23             font-size: 20px;
24         }
25         .heading{
26             text-align: center;
27             color: rgb(234, 234, 29);
28         }
29         body{
30             background-image: url("https://wallpaperaccess.com/full/437922.jpg");
31             background-size: 100%;
32         }
33     </style>
34 </head>
```

```
app.py 9+  Detection.html X
templates > Detection.html > html > head > style > .button
3  <html>
4  <head>
5  <title>Detection page</title>
6  <style>
7  body {
8  background-image: url('https://i.insider.com/560ae19dd7cc16008bde7c?width=1416');
9  background-size: 100%;
10 margin: 0;
11 font-family: 'Times New Roman', Times, serif, Helvetica, sans-serif;
12 height: 100%;
13 width: 100%;
14 }
15 h1 {
16 display: block;
17 font-size: 3.5em;
18 margin-top: 0em;
19 margin-bottom: 0em;
20 margin-left: 50%;
21 margin-right: 0;
22 font-weight: bold;
23 text-align: center;
24 }
25
26
27 .button {
28 border: 1px solid #e5b9f3;
29 color: rgb(56, 1, 69);
30 padding: 15px 32px;
31 text-align: center;
32 text-decoration: none;
33 display: inline-block;
34 font-size: 16px;
```

Github link:

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

Demo link:

<https://drive.google.com/file/d/1XkRcvO8eZFJyji8oPlZFVSo25E9ujecC/view?usp=sharing>