LOYOLA INSTITUTE OF TECHNOLOGY,CHENNAI

ANNA UNIVERSITY::CHENNAI – 600025

PROJECT TITLE : EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

A NALAIYA THIRAN PROJECT REPORT

SUBMITTED BY

TEAM SIZE : 5

TEAM LEADER : YUVASREE S (210919104052)

TEAM MEMBER : SUBASRI A (210919104044)

TEAM MEMBER : JEAN CATHERINE R (210919104017)

TEAM MEMBER : THAMIZHSELVI V (210919104046)

TEAM MEMBER : ROSEMA RANI R C (210919104036)

TEAM ID : PNT2022TMID25599

MENTOR NAME: Mrs . Savija j **EVALUATOR NAME : Mr.Noor Mohammed**

TABLE OF CONTENT

ABSTRACT

The world is burning. As global warming continues to display a statistical rise in global average temperatures and various environmental factors continue to contribute to the rise in forest fires,

the need for a wireless detection system to recognize these fire hazards and that can successfully alert the necessary first responders is becoming more and more apparent.

Such a detection and alert system would be able to potentially save billions of dollars in property, infrastructure, and environmental costs and damages, preserve wildlife habitats and ecosystems that are directly affected by forest fires, and prevent the displacement of countless families from their homes that neighbour forested areas and regions.

Therefore, we have come together as an engineering team to propose and develop a prototype solution to these issues using our acquired technical knowledge as senior computer science engineering students for our senior design project this semester. Our project idea entitled, “Forrest Fire Detection System,” will be comprised of multiple systems working in tandem

LIST OF KEYWORDS :

Forest fire detection, YOLO algorithm , global warming ,python ,IBM Cloud, IBM Watson Studio ,Deep Learning, Python-Flask

CHAPTER 1

1.1.INTRODUCTION

- Fire accidents are one of the most common form of accidents plaguing the society at large.
- Fire accidents are also the cause of huge loss to lives and property. Such a accident need a good detection system for reducing the loss of lives and property.
- There are a quite a few such existing systems ,However they are not as much effective as the situation demands to need. So we have proposed a new system to detect fire from infrared images.
- We are making use of computer vision and machine learning techniques to make it efficient and reliable. Also the system uses brightness classification along with image processing and histogram based segmentation.
- All these are made to increase the accuracy and also makes the system more suitable for real time implementation.
- Thus the proposed system not only solves the existing problems but also provides a new and efficient approach.

CHAPTER 2

LITERATURE SURVEY

1.Project Title : Optimization of Geographic Information Systems for Forest Fire Risk Assessment, 2020 21st International Symposium on Electrical Apparatus & Technologies (SIELA)

Author: Martina PETKOVIĆ; Ivan GARVANOV; Dragan KNEŽEVIĆ; Slavoljub ALEKSIĆ

TECHNIQUES USED: Geographic information system

ADVANTAGES : The performed geographic information systems optimization enables easier and faster forecasting of critical points that could be threatened by fire

DISADVANTAGES : GIS are not effective in all cases.

2. Project Title: Evaluation of Random Forest model for forest fire prediction based on climatology over Borneo, 2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA)

Author: Hanh Dang-Ngoc; Hieu Nguyen-Trung

TECHNIQUES USED: Random forest algorithm

ADVANTAGES: Random forest algorithm is considered as one of the best prediction algorithms

DISADVANTAGES: Though random forest is best for prediction , it couldn't handle large data

3. Project Title: Aerial Forest Fire Surveillance - Evaluation of Forest Fire Detection Model using Aerial Videos, 2019 International Conference on Advanced Technologies for Communications (ATC)

Author: Martina PETKOVIĆ; Ivan GARVANOV; Dragan KNEŽEVIĆ; Slavoljub ALEKSIĆ

TECHNIQUES USED: motion feature, smoke detection

ADVANTAGES : Smoke detection rate is improved to

91.48 %

DISADVANTAGES: Since hardware devices are used for video detection , it seems to be costly

4. Project Title: Speculation of Forest Fire Using Spatial and Video Data, 2019 International Conference on Advanced Technologies for Communications (ATC)

AUTHOR: T.L. Divya; M.N. Vijayalakshmi; Anupama Kumar S.

TECHNIQUES USED: Image processing

ADVANTAGES: The histograms represent the fireplaces along with intensity based on fire severity in the forest the fire video with frame numbers

DISADVANTAGES: A histogram can present data that is misleading

5. Project Title: Early Detection of Forest Fire Based on Unmaned Aerial Vehicle Platform

AUTHOR: Xingsha Yang; Linbo Tang; Hongshuo Wang; Xinxin He

TECHNIQUES USED: UAV

ADVANTAGES: Compared with the previous single source fire detection method, this method uses optical and infrared data to conduct fire discrimination, which enhances the robustness of the discrimination result

DISADVANTAGES: finally, although there are more public data sets, the lack of evaluation criteria and evaluation platforms for unified video smoke detection, especially the smoke area labeling in the video, remains unresolved

CHAPTER 3

3.1 PROBLEM STATEMENT

- The existing system is based on CNN
- The existing system uses outdated machine learning techniques that are not updated and ineffective to the modern world
- Also the existing system does not work in a stable manner and it is said to be unstable during high data inputs.

3.1.2 DISADVANTAGE

- The system also is inaccurate with around 76% accuracy and also draws huge amount of power.
- The system is slow and time consuming too which makes it difficult for real time implementation

3.2 PROPOSED SYSTEM

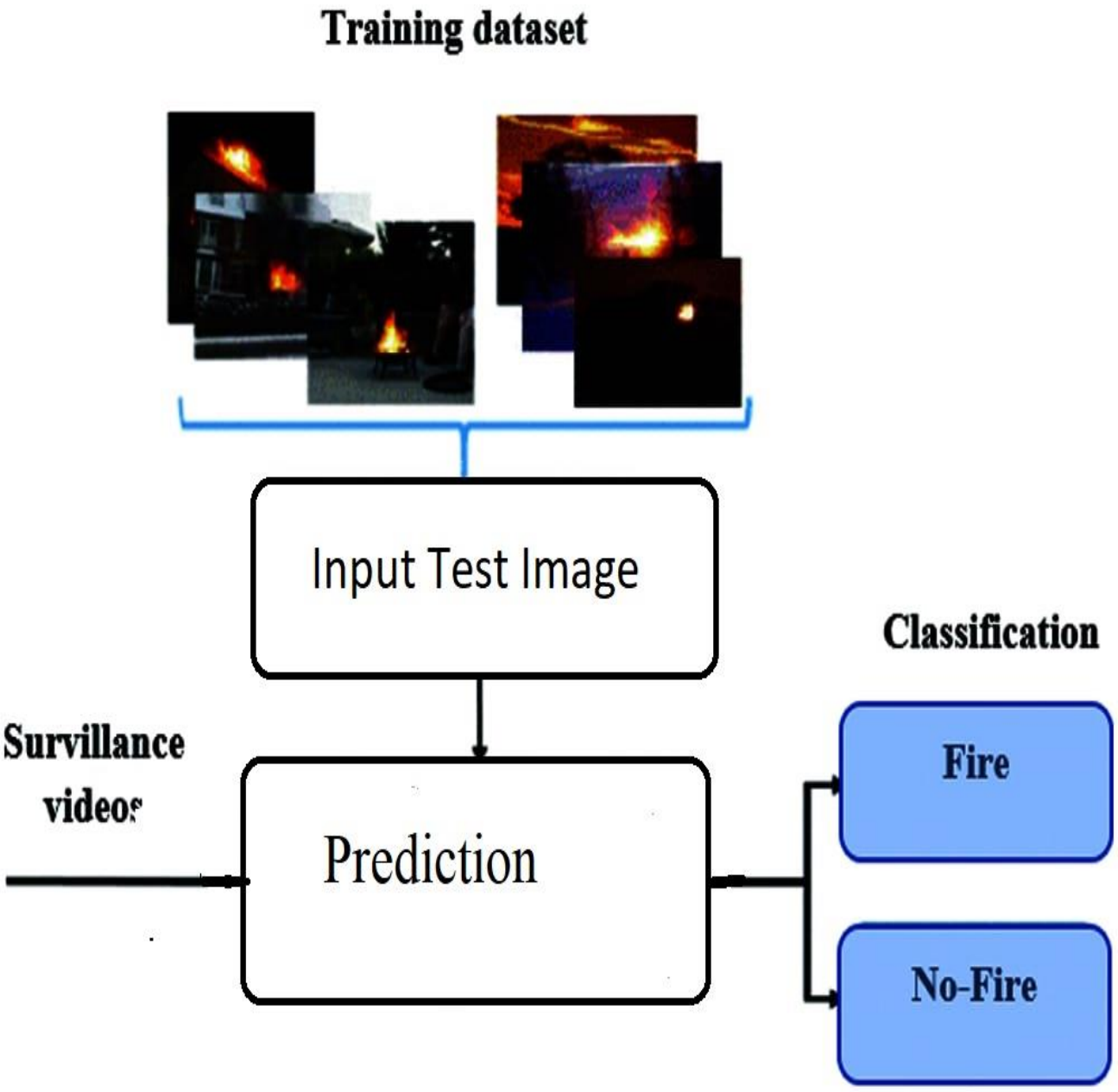
- The proposed system works on a efficient AI based approach which uses YOLOv3
- The system is highly user friendly and offers a 3-4 fold increase in computational performance.
- The system is comparatively faster as well and consumes less space too

3.2.1 ADVANTAGE

- The proposed system also has a accuracy of 92% which is comparatively higher
- The proposed system is power efficient

CHAPTER 4

4.1 STATEMENT ARCHITECTURE



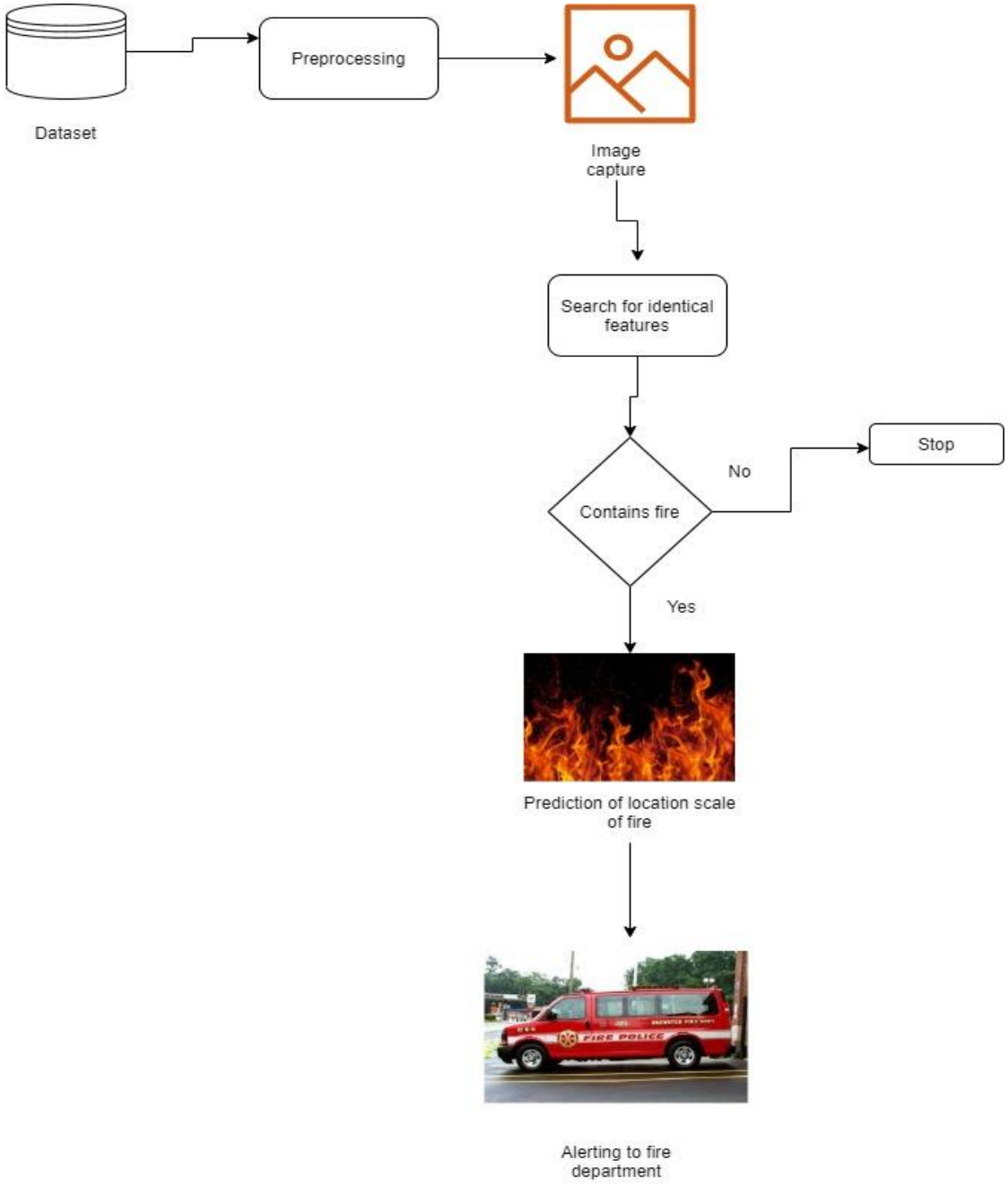
4.2. DATA FLOW DIAGRAM

4.2.1. MODULE

- Data Input GUI
    - This module is where the user interacts to know all the needs and constraints from the user.
- Data Augmentation Module
    - This module is where the data is preprocessed and readied to be feeded to the machine learning model**.**
- Data Process Model
    - This module is where the the model takes in all the various parameters and finds if the fire is present in the image or not.
- Output module
    - This module decides on the intensity and position of the fire in image and gives it as a output in the image.

CHAPTER 5

5.1. System Architecture



CHAPTER 6

6.1 ALGORITHM

- IT is an algorithm based on regression, instead of selecting the interesting part of an Image, it predicts classes and bounding boxes for the whole image in one run of the Algorithm.
- Ultimately, we aim to predict a class of an object and the bounding box specifying object location

CHAPTER 7

7.1. SYSTEM REQUIREMENT

SOFTWARE REQUIREMENT:	
Operating System	Windows 7 , 8, 10, 11 (64 bit)
Software	Anaconda
Tools	Python 3.6 and Higher
Supported Devices	All Device

HARDWARE REQUIREMENT:	
Hard Disk	500GB and Above
RAM	4GB and Above
Processor	I3 and Above

CHAPTER 8

8.1 Source Code:

8.2 App.py

```
import numpy as np
import os
import glob
import imutils
import numpy as np
from keras import models
from tensorflow.keras import preprocessing, layers, models, callbacks
from sklearn.metrics import pairwise
import time
import os  # Operating system functionality
import random  # Random number generator
import pandas as pd  # Data analysis & manipulation
import numpy as np  # Array-processing
import seaborn as sns  # Data visualization
import matplotlib.pyplot as plt  # Data visualization
from tensorflow.keras import preprocessing, layers, models, callbacks  # Neural networks
from sklearn import metrics  # Model evaluation
global loadedModel
size = 300
from PIL import Image, ImageTk
from tkinter import *

import numpy as np
import cv2
from keras.preprocessing import image
from tensorflow.keras.preprocessing.image import img_to_array
import numpy as np
#from keras import models
from tensorflow.keras.models import load_model
# loading Python Imaging Library
from PIL import ImageTk, Image
# To get the dialog box to open when required
from tkinter import filedialog
import tkinter as tk

loadedModel = models.load_model('modified_yolo_fire_new_new.h5')

model = load_model('modified_yolo_fire_new_new.h5')
print('Model Loaded Sucessfully')



#real
```

```
def real_time(current_frame,h,w):
    img = current_frame
    #print(img.shape)
    img = cv2.resize(img, (300, 300))

    img = img/255 #rescalinng
    pred_img =img.copy()
    prob = loadedModel.predict(img.reshape(-1,300,300,1))
    prob=np.argmax(prob,axis=1)
    if (prob>0.5):

        return 0,prob

    else:

        return 1,prob



def yes():



    #camera = cv2.VideoCapture(0)         PLEASE ONLY FOR REAL TIME CAPTURING

    #camera = cv2.VideoCapture("test_2.mp4")

    camera = cv2.VideoCapture(0)

    show_pred = False

    while (True):

        (grabbed,frame) = camera.read()

        frame = imutils.resize(frame,width = 700)


        clone = frame.copy()

        (height,width) = frame.shape[:2]

        grayClone = cv2.cvtColor(clone,cv2.COLOR_BGR2GRAY)



        p,prob = real_time(grayClone,height,width)



        keypress_toshow = cv2.waitKey(1)
```

```
        if (prob>0.5):
            cv2.putText(clone, "ALERT FIRE IS DETECTED", (30, 30), cv2.FONT_HERSHEY_SIMPLEX,
1.4, (0, 0, 255), 5)



        else:
            cv2.putText(clone, "NO FIRE DETECTED ", (30, 30), cv2.FONT_HERSHEY_SIMPLEX, 1.4,
(0, 255, 0), 5)


        cv2.imshow("Real Time Detection",clone)

        keypress = cv2.waitKey(1) & 0xFF

        if(keypress == (ord("q") or ord("Q"))):
            break
    camera.release()

    cv2.destroyAllWindows()

# main function
if __name__ == '__main__':


# Create a window
    root = Tk()

    bg = ImageTk.PhotoImage(file = "22.jpg")

# Create Canvas
    canvas1 = Canvas( root, width = 400,
                 height = 4400)

    canvas1.pack(fill = "both", expand = True)

# Display image
    canvas1.create_image( 0, 0, image = bg,anchor = "nw")
# Set Title as Image Loader
    root.title(" Fire Detection and Alert")
# Set the resolution of window
    root.geometry("700x450")
# Do't Allow Window to be resizable
    root.resizable(width = True, height = True)

    root.configure(bg='white')
    # Create text widget and specify size.
    T = Text(root, height = 5, width = 52)



    btn_exit = Button(root, text ='EXIT', bg='#0052cc', fg='#ffffff',width=10,
height=2,command = root.destroy).place(
                                        x = 400, y= 400)
```

```
#real
    btn_real = Button(root, text ='REAL_TIME', bg='#0052cc', fg='#ffffff',width=10,
height=2,command = yes).place(
                                        x = 600, y= 400)


    root.mainloop()
```

8.3 Jupyter Notebook

```
#Jupyter File

!pip install twilio

!pip install playsound==1.2.2

pip install pygobject

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

def video_test(vid_path):
    cap = cv2.VideoCapture(vid_path)
    if (cap.isOpened()== False):
      print("Error opening video stream or file")
    while(cap.isOpened()):
      ret, frame = cap.read()
      if ret == True:
        x=image.img_to_array(frame)
        res=cv2.resize(x,dsize=(128,128),interpolation=cv2.INTER_CUBIC)
        x=np.expand_dims(res,axis=0)
        model=load_model("/content/forest1.h5")
        cv2_imshow(frame)
        pred=model.predict(x)
        pred = int(pred[0][0])
        pred
        int(pred)
        if pred==0:
          print('ALERT!!!! FOREST FIRE DETECTED')
          break
        else:
          print("NO FOREST FIRE")
          break
```

8.4 Index.html:

```
<html>

<head>
  <title>Forest fire detection Web Application</title>

  <meta name="viewport" content="width=device-width">
  <!-- GoogleFont -->
  <link href="https://fonts.googleapis.com/css2?family=Prompt:wght@600&display=swap"
rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Varela+Round&display=swap"
rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap
" rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css?family=Calistoga|Josefin+Sans:400,700|Pacific
o&display=swap" rel="stylesheet">
  <!-- bootstrap -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
  <link rel="stylesheet" type= "text/css" href= "{{
url_for('static',filename='css/style.css') }}">
  <!-- fontawesome -->
  <script src="https://kit.fontawesome.com/b3aed9cb07.js"
crossorigin="anonymous"></script>

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>

</head>

<script>
  function preview() {
    frame.src=URL.createObjectURL(event.target.files[0]);
```

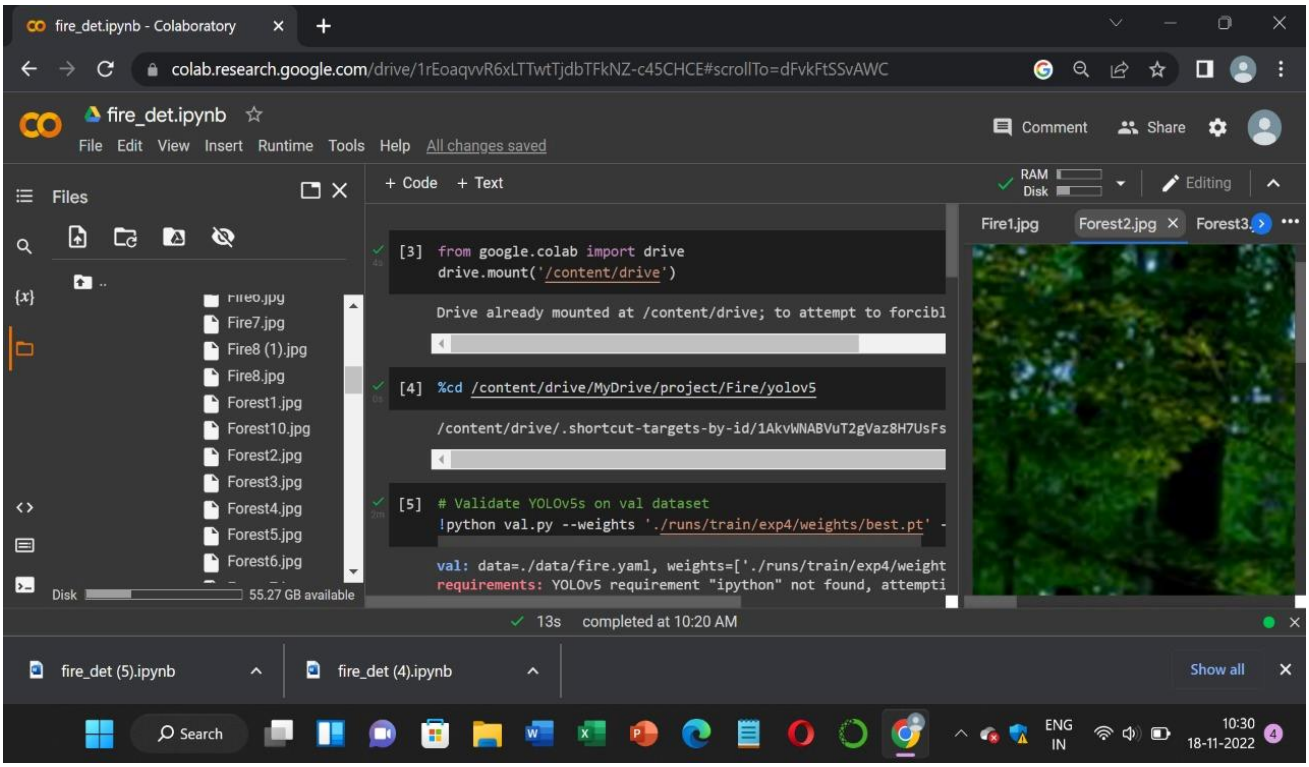
```
}

    $(document).ready(function() {
          $('#clear_button').on('click', function() {
              $('#image').val('');
              $('#frame').attr('src',"");
            });
        });

</script>

<body>

  <h1 class="welcome">IBM PROJECT-A Novel Method For Handwritten Digit Recognition
  <div id="team_id">TEAM ID : PNT2022TMID25747</div>
  </h1>
  <section id="title">
    <h4 class="heading">Handwritten Digit Recognition Website</h4>
    <br><br>
      <p>
        Here Select the Image to be Predicted and Press the button 'Read & Predict'
        to presdict the Digit that is to be recognized
      </p>

  </section>

  <section id="content">

        <div class="leftside">
        <form action="/predict" method="POST" enctype="multipart/form-data">
        <label>Select a image:</label>
        <input id="image" type="file" name="image" accept="image/png, image/jpeg"
onchange="preview()"><br><br>
          <img id="frame" src="" width="100px" height="100px"/>
          <div class="buttons_div">
            <button type="submit" class="btn btn-dark" id="predict_button">Read &
Predict</button>
            <button type="button" class="btn btn-dark" id="clear_button">  Clear
 </button>
          </div>
        </form>
        </div>
  </section>

</body>

</html>
```

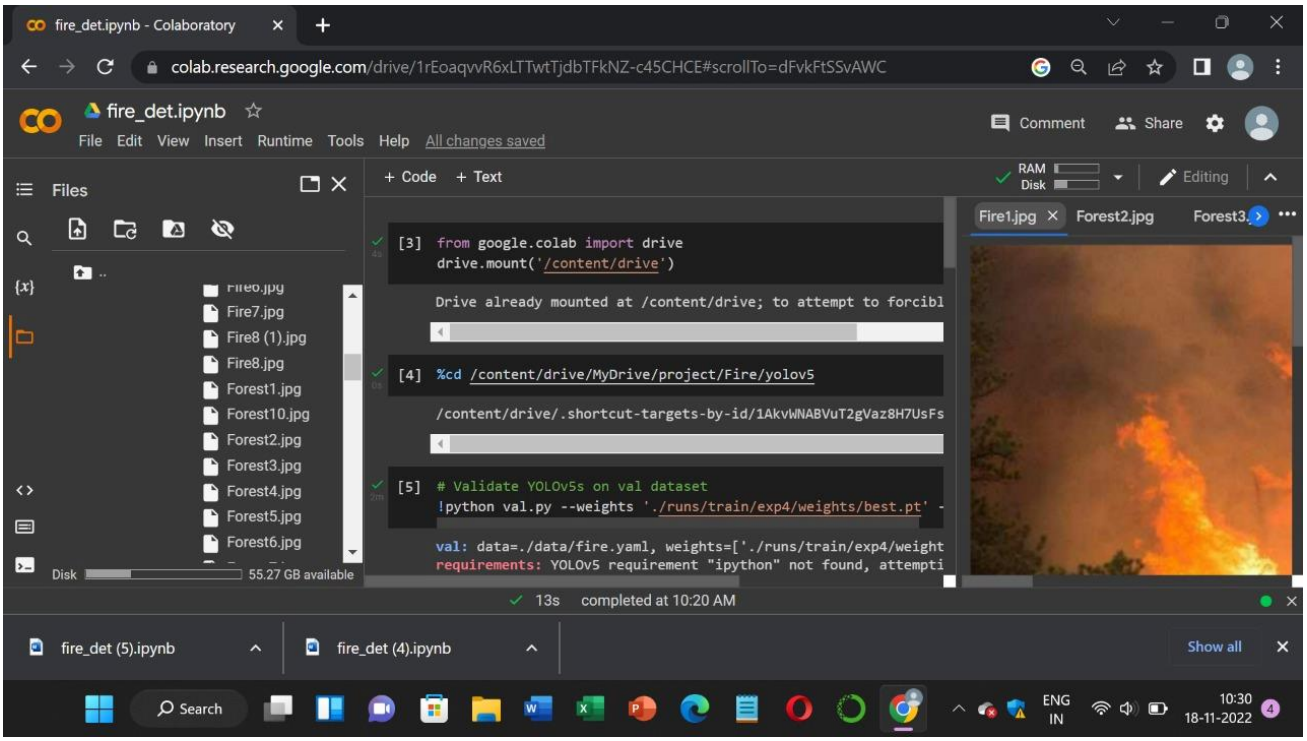
8.5 Predict.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Recognized Digit</title>
</head>

<style>
    body{
     background-image: url('static/images/Tech_Number.png');
     background-repeat: no-repeat;
     background-size: cover;
    }

    #rectangle{
     width:500px;
     height:200px;
     background-color: #202de4;
     border-radius: 25px;
     position:absolute;
     top:25%;
     left:50%;
     opacity: 0.8;
     transform:translate(-50%,-50%);
    }

    #ans{
  text-align: center;
  font-size: 40px;
  margin: 0 auto;
  padding: 3% 5%;
  padding-top: 15%;
  color: white;
    }

</style>
<body>
    <div id="rectangle">
        <h1 id="ans">Recognized Digit is : {{num}}</h1>
    </div>
</body>
</html>
```

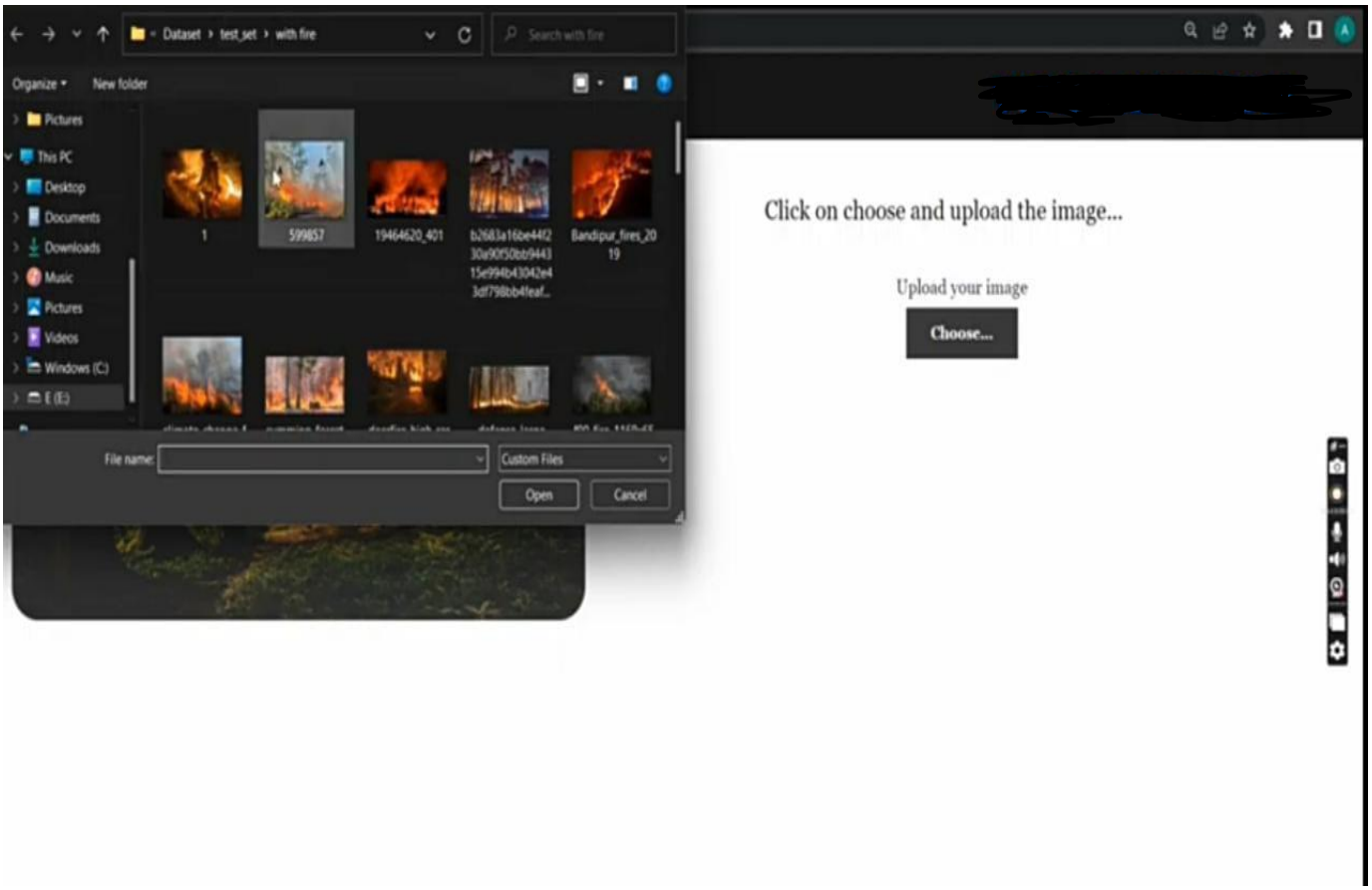
8.5 OUTPUT

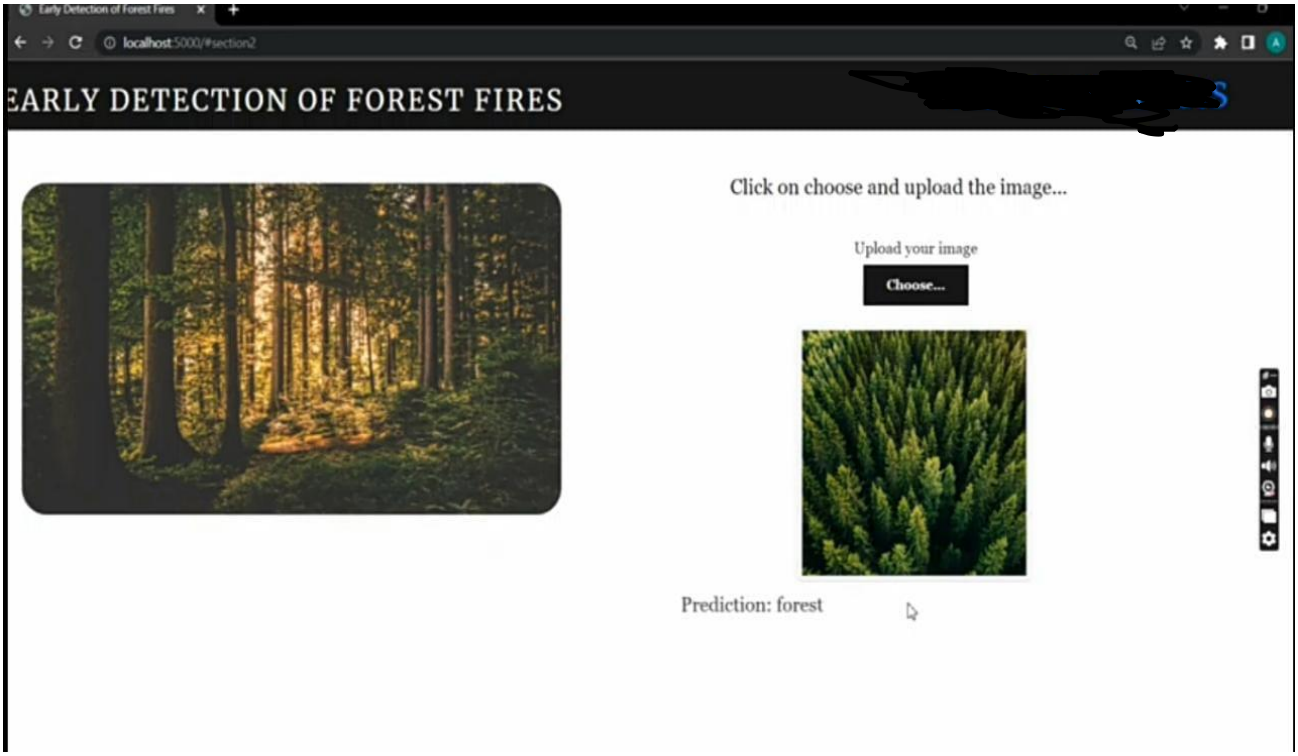
8.6 TRAINING AND TESTING DATASETS

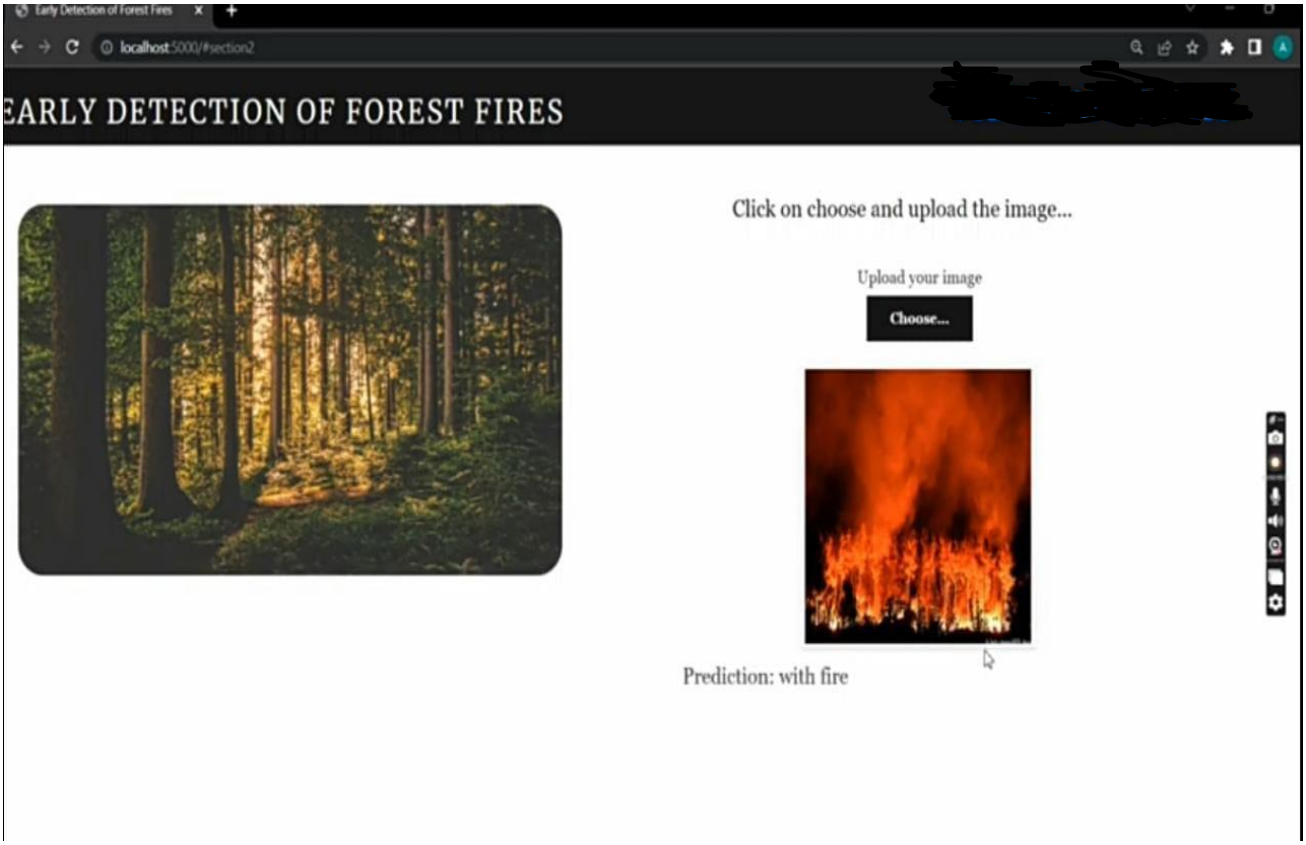




8.7 FLASK WEB APPLICATION




localhost:5000/#section2
EARLY DETECTION OF FOREST FIRES
Click on choose and upload the image...
Upload your image
Choose...
Prediction: forest


localhost:5000/#section2
EARLY DETECTION OF FOREST FIRES
Click on choose and upload the image...
Upload your image
Choose...
Prediction: with fire

CHAPTER 9

9.1 CONCLUSION

This wildfire detection application will predict if there is an image in the fire or not by using YOLO algorithm. It will be very much helpful to the forest departments of in early detection of forest fire and the immediate response can be taken thus saving number of lives and wildlife resources.

REFERENCES

1.A. Aaalkhatib, "Smart and Low Cost Technique for Forest Fire Detection using Wireless Sensor Network", *Int. J. Comput. Appl.*, vol. 81, no. 11, pp. 12-18, 2018.

2.J. Zhang, W. Li, Z. Yin, S. Liu and X. Guo, "Forest fire detection system based on wireless sensor network", *2019 4th IEEE Conf. Ind. Electron. Appl. ICIEA 2009*, pp. 520-523, 2009.

3.A. A. A. Alkhatib, "A review on forest fire detection techniques", *Int. J. Distrib. Sens. Netw.*, vol. 2019, March 2014.

4.P. Skorput, S. Mandzuka and H. Vojvodic, "The use of Unmanned Aerial Vehicles for forest fire monitoring", *2016 International Symposium ELMAR*, pp. 93-96, 2019.

5.F. Afghah, A. Razi, J. Chakareski and J. Ashdown, Wildfire Monitoring in Remote Areas using Autonomous Unmanned Aerial Vehicles, 2019.

6.Hanh Dang-Ngoc and Hieu Nguyen-Trung, "Evaluation of Forest Fire Detection Model using Video captured by UAVs", *2019 19th International Symposium on Communications and Information Technologies (ISCIT)*, pp. 513-518, 2019.

7.C. Kao and S. Chang, "An Intelligent Real-Time Fire-Detection Method Based on Video Processing", *IEEE 37th Annu. 2018 Int. Carnahan Conf. OnSecurity Technol. 2018 Proc.*, 2018.