A Project Report on

ENHANCED FOREST FIRE DETECTION USING DEEP LEARNING

Submitted in partial fulfilment for the award of

Bachelor of Technology

Degree

In

Computer Science & Engineering

By

T.Devi(Y20CS2702)

SK.Yasmeen(Y20CS2696)

M.Naga Lakshmi(Y20CS2666)

Under the Guidance of

Mrs.M.CHAITANYA KUMARI M.Tech, (Ph.D)
Asst. Professor

Department of Computer Science & Engineering
Bapatla Women's Engineering College
An ISO 9001 - 2015 certified Institution
(*Affiliated to Acharya Nagarjuna University, Guntur*)
BAPATLA-522101, Andhra Pradesh
2023-2024

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BAPATLA WOMEN'S ENGINEERING COLLEGE

An ISO 9001-2015 Certified Institution

(APPROVED BY A.I.C.T.E.)

(AFFILIATED TO ACHARYA NAGARJUNA UNIVERSITY, GUNTUR)

BAPATLA-522101

BONAFIDE CERTIFICATE

This is to certify that the project entitled "**ENHANCED FOREST FIRE DETECTION USING DEEP LEARNING** ", is the bonafide work done by **T.DEVI (Y20CS2702), SK.YASMEEN (Y20CS2696), M.NAGA LAKSHMI (Y20CS2666),** under the guidance of **Mrs.M.CHAITANYA KUMARI M.Tech, (Ph.D)** Asst. Professor, Department of **CSE, BWEC** submitted in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science & Engineering under Acharya Nagarjuna University during the academic year 2023-2024.

GUIDE	**HEAD OF DEPARTMENT**
Mrs.M.CHAITANYA KUMARI	Mrs.G.VENKATESWARI
M.Tech, (Ph.D)	**M.Tech, (Ph.D)**
Asst.Professor	**Asst.Professor&HoD**
Dept of C.S.E	**Dept of C.S.E**

ACKNOWLEDGEMENT

We sincerely thank the following distinguished personalities who have given their advice and support for successful completion of the work.

We would like to gratefully acknowledge the guidance and encouragement given by our guide **Mrs.M.Chaitanya Kumari**, **M.Tech, (Ph.D)** Asst. Professor, Department of C.S.E. who boosted our morale all the time. Our sincere thanks to her for evincing keen interest in our endeavor and for being very amicable.

We extend our sincere thanks to **Mrs.G.Venkateswari**, **M.Tech, (Ph.D)** Head of the Dept. for extending her cooperation and providing the required resources.

We would like to thank **Dr.G.Srinivasa Rao**, principal, Bapatla Women's Engineering College, Bapatla for providing the online resources and allowing us to carry out this work.

We would like to express our sincere thanks to the project coordinator **Mrs.MD. Jareena Begum**, Asst. Professor, Department of CSE for providing the suggestions and sources to carry out this work.

We extend our sincere thanks to all other teaching faculty and non-teaching staff of the department, who helped directly or indirectly for their cooperation and encouragement.

T.Devi (Y20CS2702)
Sk.Yasmeen (Y20CS2696)
M.Naga Lakshmi (Y20CS2666)

ABSTRACT

The goal of the project is to use image processing technology to detect fires and notify people. Numerous automatic fire alarm systems already exist, such as the sensor method, which has limitations. The project is put into action by connecting the webcam as hardware and using the PyCharm IDE. Using the open CV library for image processing, the entire code is written in Python. The webcam is used as an input source, capturing the surrounding video and feeding it into the system for analysis. The project mainly emphasis on computer vision, machine learning, image processing, the color model, and the project's working fire detection algorithm. We plan to overcome the shortcomings of the present systems and provide an accurate and precise system to detect fires as early as possible and capable of working in various environments thereby saving innumerable lives and resources.

TABLE OF CONTENTS

LIST OF THE FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	DESCRIPTION
CNN	Convolutional Neural Network
IoT	Internet of Things
CCD	Charge-Coupled Device
RGB	Red Green Blue

1. INTRODUCTION

1.1 INTRODUCTION

Generally, fire accidents cause economic and ecological damage as well as endangering people's lives. To avoid the fire's disasters, many early fire-detection techniques have been explored and most of them are based on particle sampling, temperature sampling, relative humidity sampling, air transparency testing, smoke analysis, in addition to the traditional ultraviolet and infrared fire detectors. However, those detectors either must be set in the proximity of a fire or can't provide the additional information about the process of burning, such as fire location, size, growing rate, and so on. Thus, they are not always reliable because energy emission of non-fires. The paper presents an early fire-alarm raising method based on video processing. The basic idea of the proposed of fire-detection is to adopt a new approach for fire detection which is based on some computer vision techniques. Based on the problems, we think that visual sensors such as video data acquired from CCTV or digital camera became an alternative way in conducting fire detection sensor, which was estimated that by using video data, the detection results can be more quickly, accurately, and effectively from the side coverage area and can be applied to monitor the environment indoor and out-door. a RGB (red, green, blue) model based chromatic and disorder measurement for extracting fire-pixels. The decision function of fire-pixel is mainly deduced by the intensity and saturation of R component. The extracted fire-pixels will be verified if it is a real fire by both dynamics of growth and disorder.

1.2 PROBLEM STATEMENT

The system is essentially useless if the batteries aren't charged, since it won't work properly. There is a bit of a burden to homeowners or business owners to always remember to keep the batteries fresh so the system operates properly when you need it most.

A couple other disadvantages fire alarm system inspectors point out is wireless systems have limited range and don't have centralized monitoring. Range can be a problem for large offices or homes, since a weak wireless connection may cause the system to not operate reliably. Wireless fire alarm systems also don't connect directly to the telephone lines, which are linked to the fire departments, so the response to an emergency could be slower as a result.

1.3 OBJECTIVE

The early detection of a fire is crucial for effective intervention and mitigation efforts. By analyzing real-time data and identifying distinguishing features such as motion, color spectrum, and textural structure, a fire can be detected and separated from other natural components in the sky and forest using various filtering, edge detection, and color recognition tools. However, once a fire becomes visible and is able to spread rapidly, it can be challenging to contain and extinguish. Therefore, proactive measures such as proper land management and fire safety practices, as well as early detection systems such as smoke detectors and surveillance cameras, are important to prevent fires from starting and to detect them in their early stages for prompt intervention.

Once the clues of the fire is inspected in a real time process, it has obvious distinguishing features both in terms of motion, color spectrum and textural structure. In this way, it can be easily separated from other natural components in the sky and forest by means of various filtering, edge detection and color recognition tools. However, it is too late to intervene when the fire and flame become vision.

2.LITERATURE SURVEY

[1] Image processing-based forest fire detection, International Journal of Emerging Technology and Advanced Engineering.

Year of publication: 2022

Description: A clever methodology for woodland fire discovery utilizing the picture handling strategy is proposed.

A standard based variety model for fire pixel grouping is utilized. The proposed calculation utilizes RGB and YC b C r variety space. The ability of YC b C r color space to effectively distinguish luminance from chrominance makes it superior to RGB color space. On two sets of images, one of which contains fire, the proposed algorithm is tested for its effectiveness; The other has areas that look like fire. The algorithm's performance is calculated using standard techniques. The proposed technique has both a higher location rate and a lower deception rate. Since the calculation is modest in calculation, it very well may be utilized for constant woodland fire discovery. Watchwords Backwoods fire discovery, picture handling, rule-based variety model, picture division.

[2] Fire detection: The state-of-the-art NBS Technical Note, US.

Year of publication: 2021

Description: The document provides an overview of the principles of fire detection, types of detectors, and their operational characteristics. It highlights the importance of early detection in preventing fire-related fatalities and property damage. The technical note discusses different types of fire detection methods, including smoke detection, heat detection, and flame detection. It outlines the advantages and disadvantages of each method and emphasizes the need for reliable and accurate fire detection systems.

The document also covers the latest technologies in fire detection, including multi-sensor detectors, which can combine different sensing elements to improve the accuracy and reliability of fire detection. The authors discuss the benefits of integrating fire detection systems with building automation systems and the potential for data analytics to enhance fire safety.

Mainly:

The technical note emphasizes the importance of proper installation, maintenance, and testing of fire detection systems to ensure their effectiveness. It provides guidelines for the placement and installation of fire detectors in various types of buildings and environments.

In conclusion, the NBS Technical Note on **"Fire Detection: The State of the Art"** provides a valuable resource for fire safety professionals, engineers, and researcher.

[3] Convolutional neural network for video fire and smoke detection

Year of publication: 2021

Description: Research on video examination for fire location has turned into a hotly debated issue in PC vision. Be that as it may, the traditional calculations use solely rule-based models and elements vector to characterize regardless of whether an edge is fire. These characteristics are challenging to define and largely dependent on the type of fire observed. The result prompt's low location rate and high misleading problem rate. An alternate methodology for this issue is to utilize a learning calculation to extricate the helpful elements as opposed to utilizing a specialist to fabricate them.For video fire detection, we propose a convolutional neural network (CNN) in this paper. It has been demonstrated that convolutional neural networks perform exceptionally well in the classification of objects.

[4] A probabilistic approach for vision-based fire detection in videos

Year of publication: 2020

Description: Computer vision is currently conducting research on automated fire detection. A novel approach to identifying fire in videos is the subject of our investigation in this paper. PC vision-based fire location calculations are normally applied in shut circuit TV observation situations with controlled foundation. In contrast, the proposed method can be used for both surveillance and automatic video classification in order to retrieve fire disasters from newscast content databases. The changes that occur from frame to frame in particular low-level features that describe potential fire regions are the focus of the proposed method. Color, area size, surface roughness, boundary roughness, and skewness within estimated fire regions are these characteristics.

In view of flashing and arbitrary qualities of fire, these highlights are strong discriminants. For robust fire recognition, the behavioral change of each of these features is evaluated, and the results are combined using a Bayes classifier. Additionally, the classification results are significantly enhanced by using prior knowledge of video fire events. For altered report recordings, the fire locale is typically situated in the focal point of the casings. The probability of fire as a function of position is modeled using this fact. The method's applicability was demonstrated through experiments.

3. SYSTEM REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENTS

The functional requirements of a forest fire alarm system can be categorized into four main areas:

Input :

Cameras (with thermal imaging): Can provide visual confirmation of a fire's presence and location.

Behaviour

Detection:

The system should be able to reliably detect the presence of a forest fire. This may involve using a combination of sensors that can detect smoke, heat, and flame.

Alerting:

Once a fire is detected, the system should promptly generate an alert and transmit it to the appropriate authorities. This may involve sending a signal to a central monitoring station, triggering an alarm siren, or sending text messages or emails to designated personnel.

Reporting:

The system should be able to provide data on the location and size of the fire. This information can be crucial for firefighters in responding to the blaze.

Output:

Once the Fire is identified by the camera the signal send to the central system. The fire is Detected it gives the alarm sound and display a message in command prompt like,

Fire alarm initiated

Fire is detected

3.2 NONFUNCTIONAL REQUIREMENTS

Non-Functional Requirements for a Forest Fire

Alarm System

Non-functional requirements define how the system should behave rather than what it should do. Here are some crucial non-functional requirements for a forest fire alarm system:

Reliability:

The system must be highly reliable with minimal downtime. Early detection of forest fires is critical; a malfunctioning system could have catastrophic consequences.

Availability:

The system needs to be available 24/7/365. This means minimal maintenance periods and quick recovery from any failures.

Scalability:

The system should be scalable to accommodate changes in the size of the forest area being monitored. This could involve adding more sensors or expanding the communication network.

Performance:

The system must have fast response times. Any delays in detecting or reporting a fire could significantly impact firefighting efforts.

Security:

The system needs to be secure against tampering or hacking. Malicious actors could attempt to disable the system or provide false alarms.

Maintainability:

The system should be easy to maintain and repair. This includes having readily available replacement parts and user-friendly diagnostic tools.

Portability for a Forest Fire Alarm System

Portability refers to the ability of the system to function in different environments. In the context of a forest fire alarm system, portability could involve:

Hardware compatibility:

The system should be compatible with a variety of sensor types and communication

protocols. This allows for flexibility in choosing the most appropriate equipment for the specific terrain and forest conditions.

Software independence:

The software should be platform-independent, allowing it to run on different operating systems. This ensures easier integration with existing infrastructure or future upgrades.

Power source flexibility:

The system should be adaptable to different power sources. This could include solar panels, batteries, or even connection to the grid, depending on the remoteness of the location.

Environmental adaptability:

The system's components need to be able to withstand harsh outdoor environments. This includes resistance to extreme temperatures, humidity, and wind.

By considering these non-functional requirements and portability aspects, you can design a forest fire alarm system that is reliable, efficient, and effective in protecting forests from wildfires.

3.3 SOFTWARE REQUIREMENTS

3.3.1 Prerequisites

Before going any further, let's discuss what you need to know in order to follow this tutorial with ease. Firstly, you should have some basic programming knowledge in any language. Secondly, you should know what machine learning is and the basics of how it works, as we will be using some machine learning algorithms for image processing in this article. As a bonus, it would help if you have had any exposure to, or basic knowledge of, Open CV before going on with this tutorial. But this is not required. One thing you should definitely know in order to follow this tutorial is how exactly an image is represented in memory. Each image is represented by a set of pixels i.e., a matrix of pixel values. For a grayscale image, the pixel values range from 0 to 255 and they represent the intensity of that pixel.

an image of 20 x 20 dimensions, it would be represented by a matrix of 20x20 (a total of 400- pixel values).If you are dealing with a colored image, you should know that it would have three channels - Red, Green, and Blue (RGB). Therefore, there would be three such matrices for a single image.

3.3.2 Installation

Note: Since we are going to use OpenCV via Python, it is an implicit requirement that you already have Python (version 3) already installed on your workstation.

Windows

$ pip install opencv-python

MacOS

$ brew install opencv3 --with-contrib --with-python3

Linux

$ sudo apt-get install libopencv-dev python-opencv

To check if your installation was successful or not, run the following command in either a Python shell or your command prompt: import cv2 Some Basics You Should Know Before we move on to using Image Processing in an application, it is important to get an idea of what kind of operations fall into this category, and how to do those operations. These operations, along with others, would be used later on in our applications. So, let's get to it.

3.4 HARDWARE REQUIREMENTS

- System: Pentium i3 Processor.
- Hard Disk: 500 GB.
- Monitor: 15’’ LED
- Input Devices: Keyboard, Mouse
- Ram: 2 GB

Hardware requirements and it's explanation for forest fire alarm system.

The hardware requirements for a forest fire alarm system can vary depending on the complexity and desired features of the system.

Here's a breakdown of some common components:

Cameras:

Can be used with visual recognition software to detect flames.

Microcontroller unit (MCU):

Processes data from the sensors and triggers alarms if fire is detected.

Communication module:

Enables transmission of fire alerts to a central monitoring station or emergency responders. This can be done via:

Cellular network

Satellite network

Low-power wide-area network (LPWAN)

Power source: Forest fire alarms typically rely on batteries with a solar panel for recharging to ensure continuous operation.

profile picture

Hardware requirements for forest fire alarm system using cameras

Show drafts

A camera-based forest fire alarm system utilizes cameras as its primary fire detection method. Here are the key hardware components involved:

High-definition cameras:

Equipped with megapixel resolution for capturing clear images even over long distances.

Computer vision processing unit: Analyzes camera footage in real-time using image processing algorithms to detect fire signatures like flickering flames, smoke plumes, and unusual temperature variations.

Additional components:

Microcontroller unit (MCU):

Manages the system's overall operation, coordinates data processing, and triggers alerts.

Communication module:

Transmits fire alerts to designated recipients. (e.g., cellular network, satellite network)

Power source:

Ensures continuous system operation. (e.g., batteries with solar panels).

4. SYSTEM ANALYSIS

4.1 EXISTING SYSTEM

Using Sensors as a Data Acquisition Center. The microcontroller will receive either the flag or the data from these sensors. In order to keep an eye on the entire forest area, a number of end sensors should be placed at specific distances in order to be used in emergency situations. In existing system, the most common hazard in forest fires is accident as the forest themselves destroys the forests and can be great threat to wild life and peoples. The Internet of Things (IoT) is the physical device which is used to connect, store and enable the objects to collect information for exchanging the data through the internet-based system.

4.2 PROPOSED SYSTEM

In this proposed system an image-based fire detection system was proposed, which is based on computer vision-based techniques. We have collected a number of sequential frames from original video, which consists of fire and non-fire images.

The proposed method consists three main stages:

1. fire pixel detection using RGB and YCbCr color model.
2. moving pixel detection and analyzing shape of fire.
3. colored pixels in frames to detect fire pixel in image.

The proposed method is applied on video sequences and then fire is detected.

5. SYSTEM DESIGN

5.1 BLOCK DIAGRAM

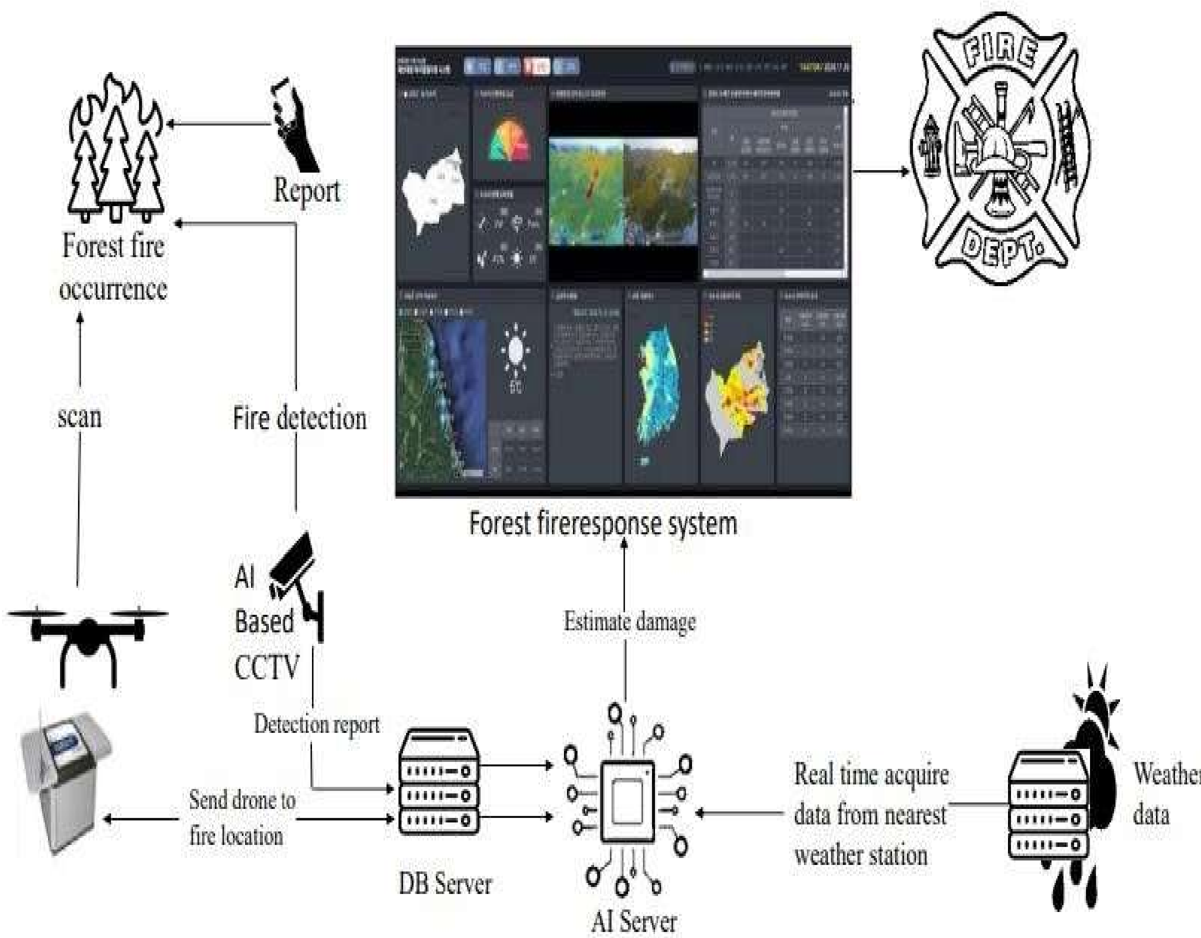

Figure 5.1: block diagram of deep learning- based forest fire response system

Figure 5.1 shows a block diagram of a deep learning-based forest fire response system, which includes a data acquisition module, a deep learning model for fire detection, a decision-making module, and a response module for initiating appropriate actions in response to detected fires.

5.2 FLOW CHART

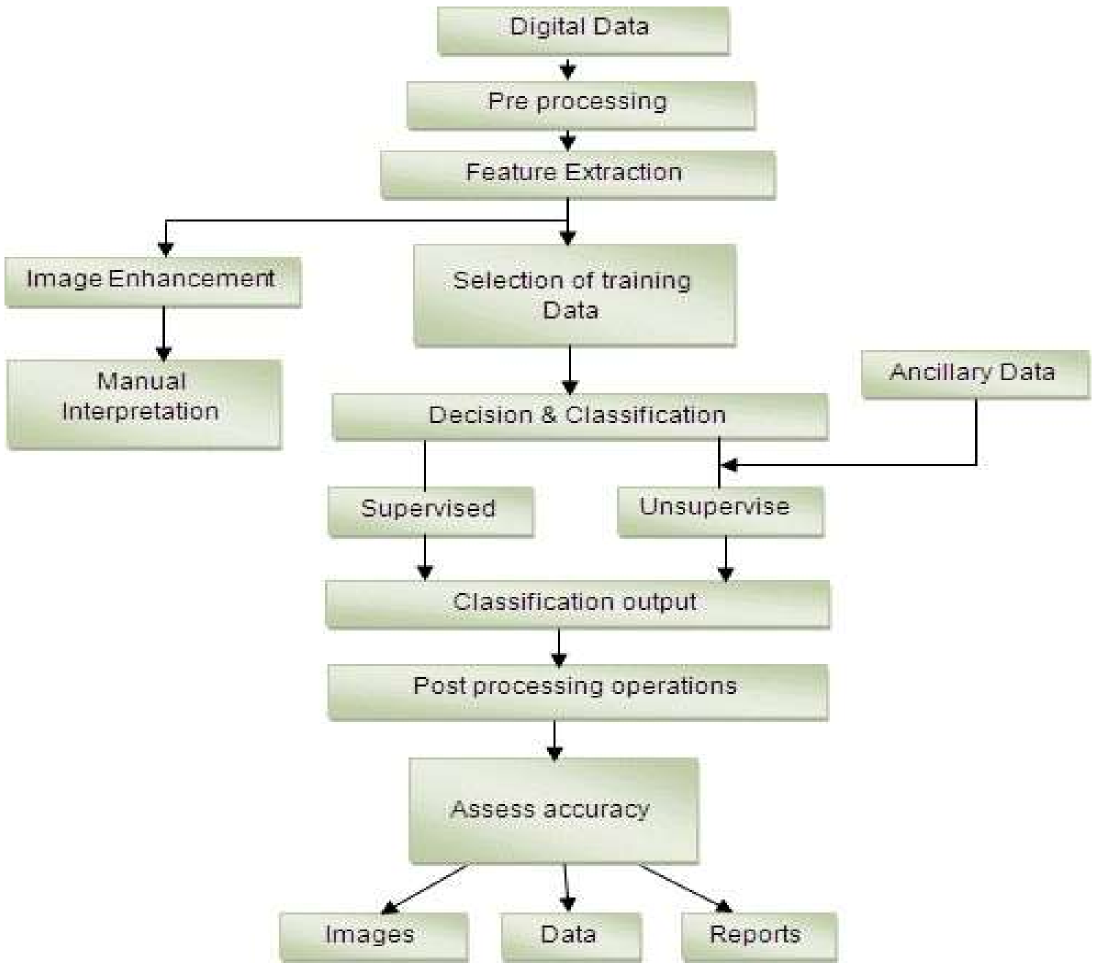

Figure 5.2: flow chart of image processing

Overall, this flow chart provides a high-level overview of the typical steps involved in image processing, and the specific techniques used in each step may vary depending on the application and the type of images being processed.

5.3 ARCHITECTURE DIAGRAM

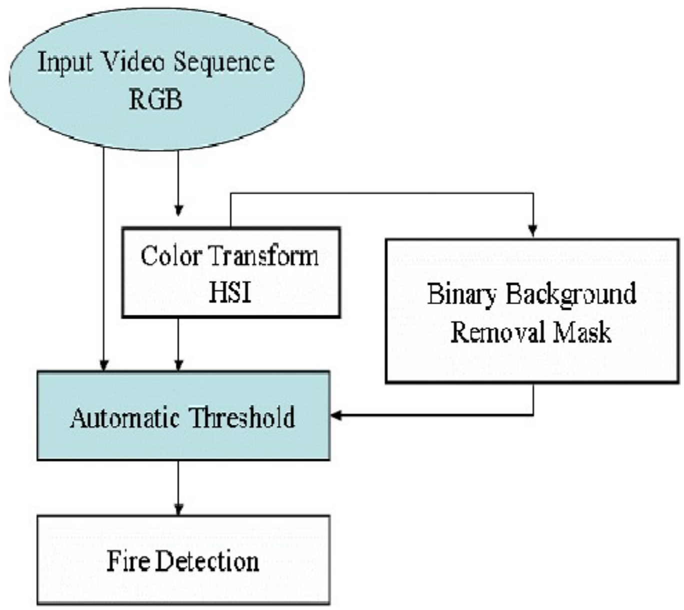

Figure 5.3: Architecture Diagram

- The first step is the training of the classifier, as mentioned early. Training for the highly accurate classifier needs lots of time and processing power, so here we only used a little number of pictures.
- After training the fire cascade classifier, the captured frame from the webcam is converted into grayscale. The reason for converting frame into grayscale is because the frame captured by webcam is in RGB color.
- Since RGB images have three channels of colors, so if the image is converted into grayscale, there will be only one channel, either black or white, which is easy to process.
- After the conversion, the fire classifier is used, which will help to find the features and location of images. Parameters like scale factor and min neighbor are passed.

  These factors are an essential factor in detecting fire. A scale factor is used for creating of scale pyramid because while training the classifier fixed size of the image is trained, so the scale factor will allow rescaling the size of an input frame to detect the fire.

- Another parameter min 24 neighbor will determine the quality of an image here for thesis min neighbor factor.

5.4 ACTIVITY DIAGRAM

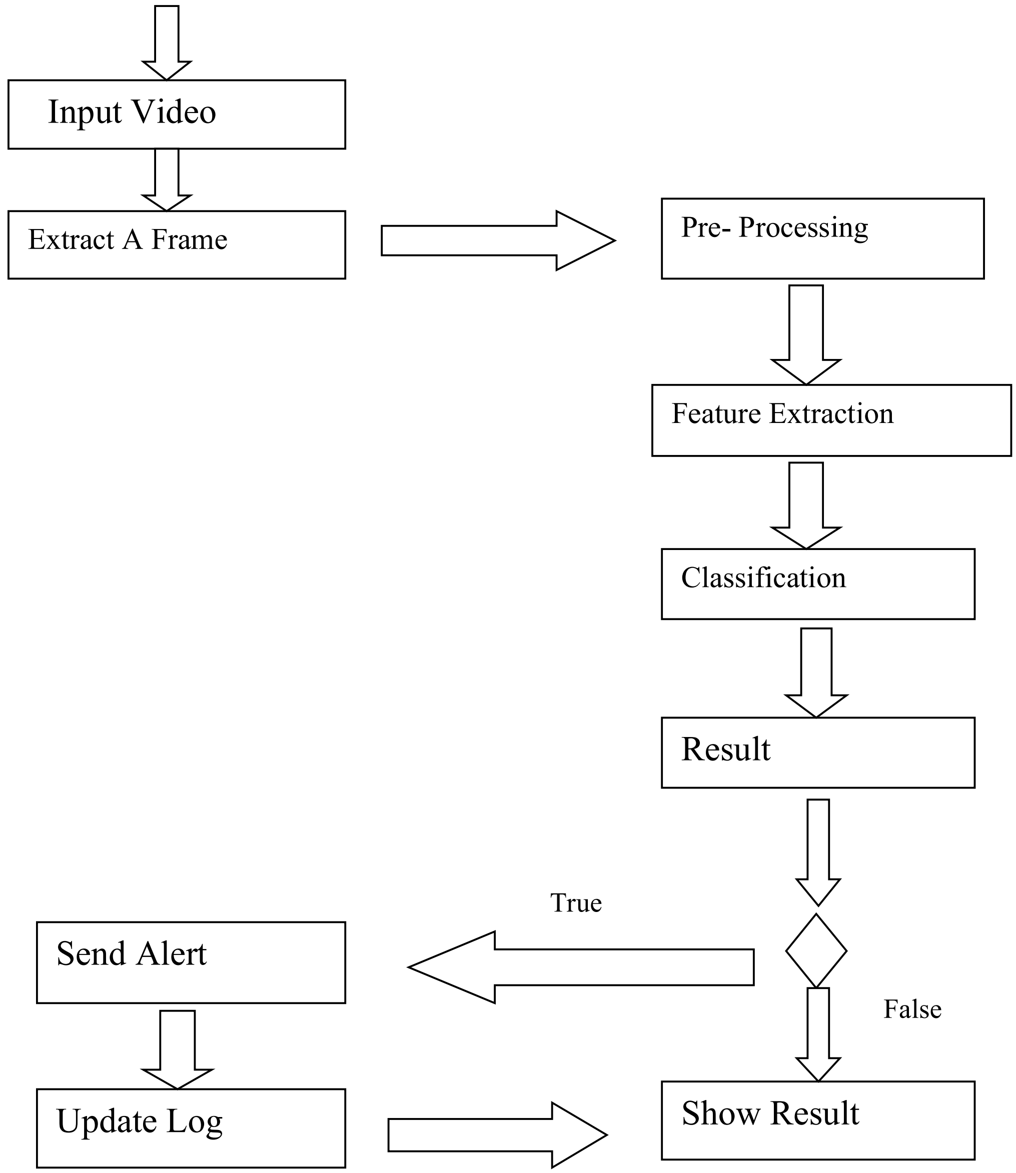

Figure 5.4: Activity Diagram

6.SYSTEM IMPLEMENTATION

6.1 INTRODUCTION OF PYTHON

Below are some facts about Python:

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like Google, Amazon, Facebook, Instagram, Dropbox, Uber... etc.

The biggest strength of Python is huge collection of standard library which can be used for the following

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)

6.2 INSTALLATION OF PYTHON

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative. functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive you can actually sit at a Python prompt and interact

with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking Install Python Step-by-Step in Windows and Mac

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.

Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.

Step 3: Click on Install NOW After the installation is successful. Click on Close.

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

Step 1: Click on Start

Step 2: In the Windows Run Command, type "cmd"

Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type python -V and press Enter.

Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works:

Step 1: Click on Start

Step 2: In the Windows Run command, type "python idle"

Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. Click on File

> Click on Save

Step 5: Name the file and save as type should be Python files. Click on SAVE. Here 1 have named the files as Hey World.

Step 6: Now for e.g. enter print ("Hey World") and Press Enter.

You will see that the command given is launched. With this, we end our tutorial on how to install Python. You have learned how to download python for windows into your respective operating system.

6.3 DEEP LEARNING

Deep learning is a type of artificial intelligence (AI) that imitates the structure and function of the human brain. It uses artificial neural networks, which are complex algorithms inspired by networks of neurons in the brain.

These neural networks are trained on large amounts of data to learn how to identify patterns and relationships.

Deep learning is useful for a variety of tasks, including:

Image recognition: Deep learning can be used to identify objects in images, such as faces, cars, and animals.

Speech recognition: Deep learning can be used to convert spoken language into text.

Natural language processing: Deep learning can be used to understand the meaning of text and generate human-like text.

Machine translation: Deep learning can be used to translate text from one language to another.

Recommendation systems: Deep learning can be used to recommend products or services to users based on their past behavior.

Working of Deep Learning for Forest Fire Detection:

Deep learning is proving to be a valuable tool in forest fire detection. Here's a simplified overview of how it works:

Data Collection: Large datasets of images containing forest fires and non-fires are collected from satellites, drones, or ground-based cameras.

Feature Extraction: Deep learning models, typically convolutional neural networks (CNNs), are used to automatically extract features from the images. These features can include patterns, colors, and textures that are indicative of fire.

Model Training: The CNN is trained on the labeled data (fire or non-fire) to learn the correlation between the extracted features and the presence of fire.

Fire Detection: Once trained, the model can then analyze new images and identify features that match its understanding of fire, thereby detecting forest fires in real-time.

6.4 CONVOLUTIONAL NEURAL NETWORK

In deep learning, a convolutional neural network (CNN) is a specialized architecture particularly well-suited for image recognition and analysis tasks.

CNNs leverage a specific structure inspired by the visual cortex of the human brain, enabling them to efficiently capture spatial relationships and patterns within images.

CNN's decisions are influenced by what it has learned in the past because it recalls the past.

- Simple feed-forward networks also "don't forget" things, but they think about things they learned in training at some point.
- Similar to feedforward neural networks in appearance, a recurrent neural network also has connections that point backwards.
- The CNN receives the inputs x(t) and its own output from the previous time step, y(t-1), at each time step t (also referred to as a frame). It is typically set to 0 because there is no previous output at the first time

step.

- You can easily construct a layer of convolutional neurons. Every neuron receives the input vector x(t) and the output vector y(t-1) from the previous time step at every step t.

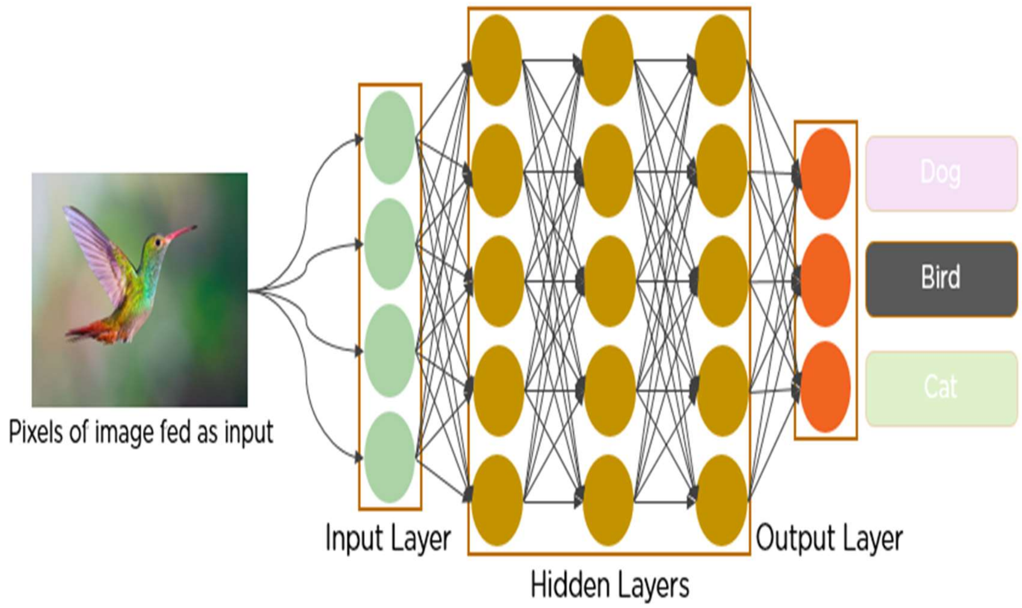

Figure:6.4 CNN (convolutional Neural Network)

6.5 MODULES

6.5.1 Image Acquisition

Image acquisition can be defined as the act of procuring an image from sources.This can be done by hardware system such as cameras and datasets and also some encoders sensors also take place in this process.

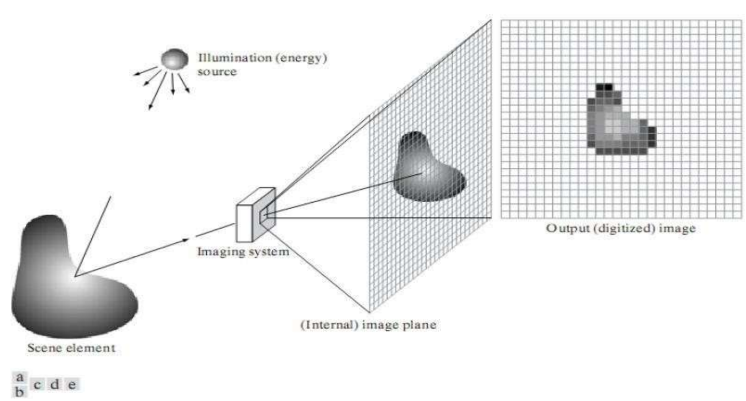

Figure 6.5.1: Image acquisition

6.5.2 Pre-Processing

In this step, the acquired images are pre-processed to enhance image quality and correct for any distortions or noise. This may involve techniques such as image filtering, color correction, and image registration.

Image processing

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually, Image Processing system includes treating images as two-dimensional signals while applying already set signal processing methods to them.

It is among rapidly growing technologies today, with its applications in various aspects of a business. Image Processing forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps.

- Importing the image with optical scanner or by digital photography.
- Analyzing and manipulating the image which includes data compression and image enhancement and spotting patterns that are not to human eyes like satellite photographs.
- Output is the last stage in which result can be altered image or report that is based on image analysis.

Digital Processing techniques help in manipulation of the digital images by using computers.

The three general phases that all types of data have to undergo while using digital technique are Pre- processing, enhancement and display, information extraction.

For this article we'll be using the following image:

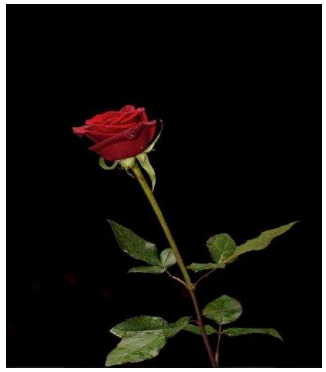

Figure 6.5.2: Image used for image processing

Note: The image has been scaled for the sake of displaying it in this article, but the original size we are using is about 1180x786.

You probably noticed that the image is currently colored, which means it is represented by three color channels i.e., Red, Green, and Blue. We will be converting the image to grayscale, as well as splitting the image into its individual channels using the code below.

6.5.2(1) Finding Image Details

After loading the image with the in read() function, we can then retrieve some simple properties about it, like the number of pixels and dimensions:

```
import cv2
img = cv2.imread('rose.jpg') print("Image Properties")
print("- Number of Pixels: " + str(img.size))
print("- Shape/Dimensions: " + str(img.shape))
```

output:

Image Properties

- Number of Pixels: 2782440

- Shape/Dimensions: (1180, 786, 3)

Now we'll split the image in to its red, green, and blue components using OpenCV and display them:

```
from google.colab.patches import cv2_imshow
blue, green, red = cv2.split(img) # Split the image into its channels
img_gs = cv2.imread('rose.jpg', cv2.IMREAD_GRAYSCALE) # Convert image
to grayscale
cv2_imshow(red) # Display the red channel in the image
cv2_imshow(blue) # Display the red channel in the image
cv2_imshow(green) # Display the red channel in the image
cv2_imshow(img_gs) # Display the grayscale version of image
```

For brevity, we'll just show the grayscale image

Figure 6.5.2(1): Grayscale image

The concept of thresholding is quite simple. As discussed above in the image representation, pixel values can be any value between 0 to 255. Let's say we wish to convert an image into a binary image i.e., assign a pixel either a value of 0 or 1. To do this, we can perform thresholding. For instance, if the Threshold (T) value is 125, then all pixels with values greater than 125 would be assigned a value of 1, and all pixels with values lesser than or equal to that would be assigned a value of 0. Let's do that through code to get a better understanding

Figure: 6.5.2.(2). Image used for Thresholding

```
import cv2
# Read image
img = cv2.imread('image.png', 0)
# Perform binary thresholding on the image with T = 125
r,threshold=cv2.threshold(img,125,255,cv2.THRESH_BINARY)
cv2_imshow(threshold)
```

output:

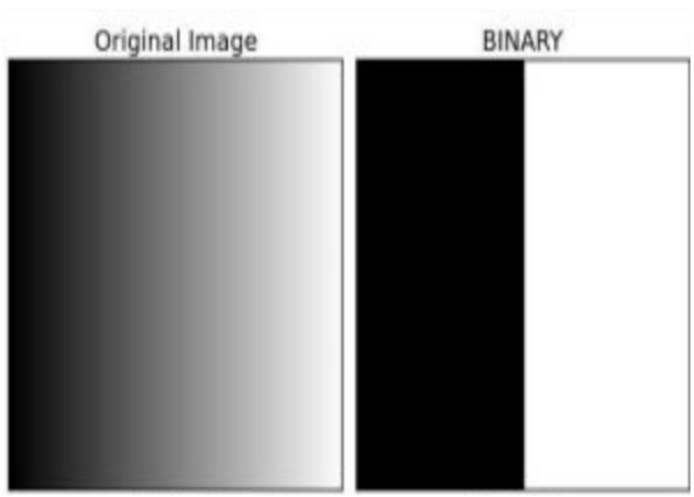

Figure 6.5.2(2): Resultant image of thresholding

As you can see, in the resultant image, two regions have been established, i.e., the black region (pixel value 0) and white region (pixel value 1). Turns out, the threshold we set was right in the middle of the image, which is why the black and white values are divided there.

6.5.3 Image Segmentation

Image segmentation is the process of dividing an image into multiple regions based on similarities in color, texture, or other features. This step is used to identify regions of interest in the image, such as objects or areas with specific characteristics.

6.5.4 Feature Extraction

In this step, key features of the segmented regions are extracted to represent the image in a more compact and meaningful way. These features may include shape, texture, color, or other properties that can be used to identify or classify the image.

6.5.5 Classification

The extracted features are used to classify the image into one or more predefined categories, based on the intended application of the image processing. This may involve using machine learning algorithms, such as neural networks or support vector machines, to train a model to recognize and classify images.

6.5.6 Post Pre-processing

Finally, the classified images may undergo post-processing, which may involve further enhancement, filtering, or compression to prepare them for storage or display.

7. SOURCE CODE

It includes the development and execution of the python program in the software where we can design and also execute the same program in a single software. The program has been typed in the workspace and executed by using the serial capture program:

CODE

```
import cv2
# Library for openCV

import threading

# Library for threading -- which allows code to run in backend

import playsound

# Library for alarm sound

import smtplib

# Library for email sending

fire_cascade =
cv2.CascadeClassifier('fire_detection_cascade_model.xml')
# To access xml file which includes positive and negative images of
fire. (Trained images)

        # File is also provided with the code.
vid = cv2.VideoCapture(0)
# To start camera this command is used "0" for laptop inbuilt camera
and "1" for USB attached camera for pc

runOnce = False

# created Boolean
def play_alarm_sound_function():
# defined function to play alarm post fire detection using threading
playsound.playsound('Alarm Sound.mp3',True)

# to play alarm
# mp3 audio file is also provided with the code.
```

```
print ("Fire alarm end")
# to print in console
def send_mail_function():
# defined function to send mail post fire detection using threading
recipientmail = "adharshyanamal0492@gmail.com"
# recipients mail recipientmail = recipientmail.lower()
# To lower case mail

server = smtplib.SMTP('smtp.gmail.com',587) server.ehlo()
server.starttls()
server.login("adharshyanamal0492@gmail.com",
'SPEEDISGOOD1')
#Sender's mail ID and password
        server.sendmail('adharshyanamal0492@gmail.com',
recipientmail, "Warning fire accident has been reported")
        # recipients mail with mail message
        print ("Alert mail sent successfully to {}".format(recipientmail))
        # to print in console to whom mail is sent
server.close()
## To close server
except Exception as e:
print(e)
# To print error if any
while (True):
Alarm_Status = False
ret, frame = vid.read()
# Value in ret is True # To read video frame
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    # To convert frame into gray color
fire = fire_cascade.detectMultiScale(frame, 1.2, 5)
```

```
    # to provide frame resolution

    ## to highlight fire with square for (x,y,w,h) in fire:

    cv2.rectangle(frame, (x-20, y-20), (x+w+20, y+h+20), (255,0,0),2)

    roi_gray = gray [y:y+h, x:x+w]

    roi_color = frame [y:y+h, x:x+w]

    print ("Fire alarm initiated")

    threading.Thread(target=play_alarm_sound_function).start()

    # To call alarm thread
if runOnce == False: print ("fire is detected")

    threading.Thread(target=send_mail_function).start()

    # To call alarm thread

    runOnce = True

if runOnce == True: print ("fire is detected")

runOnce = True

    cv2.imshow('frame', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):

        break
```

8. RESULT

The output frame of a deep learning-based forest fire response system typically consists of a visual representation or heatmap indicating the areas of the forest that are identified as being affected by the fire.

When open the command prompt type the file name with extension before open the path of the file and type like Python main.py
Then the camera open immediately

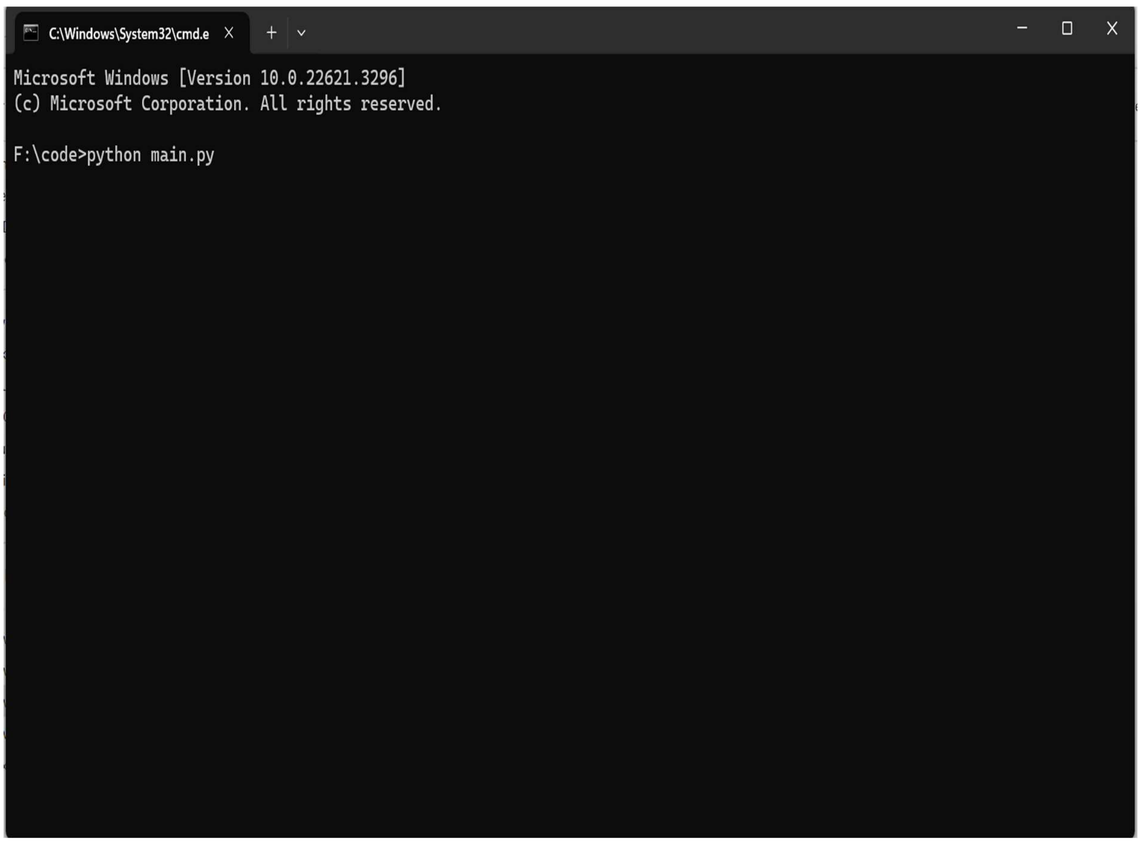

Figure 8.1: Opening Main File

.

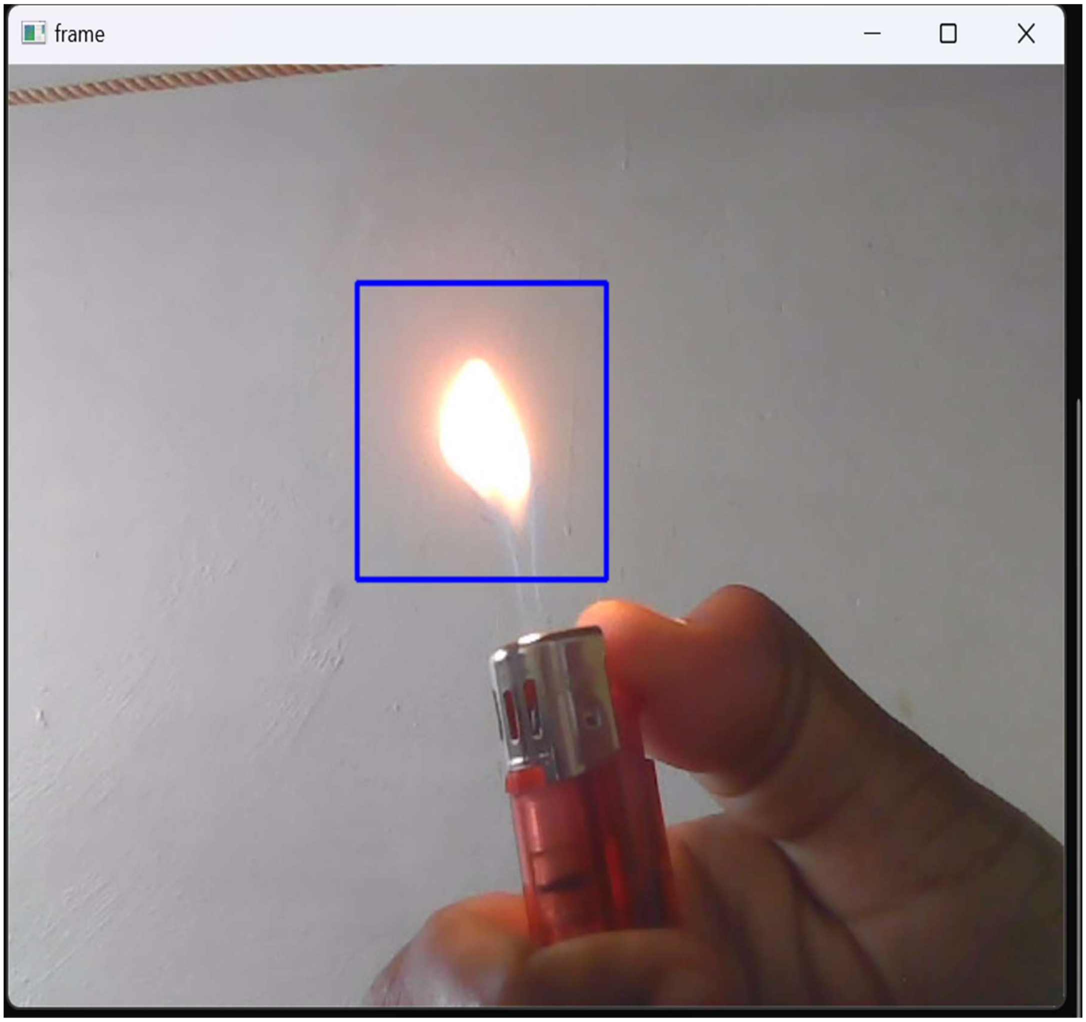

Figure 8.2 Extracting the Fire Image from the Video

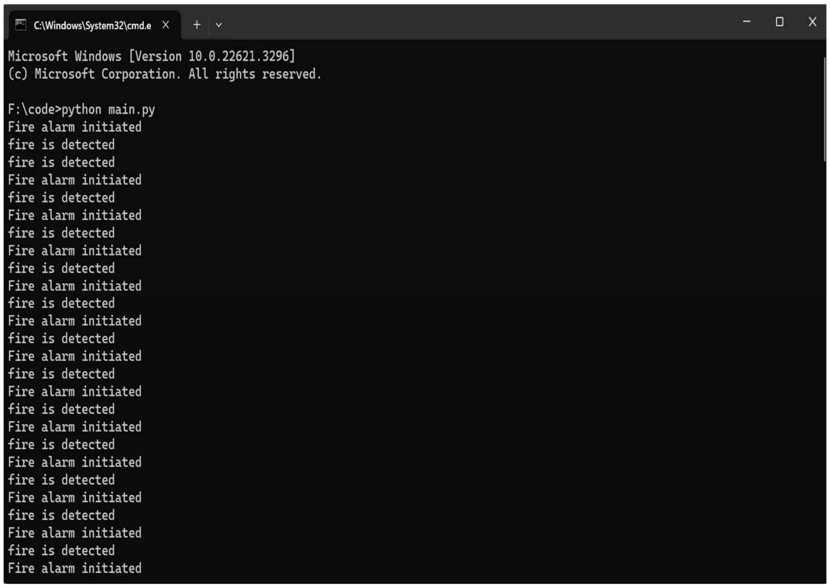

Figure 8.3: Final Output Frame

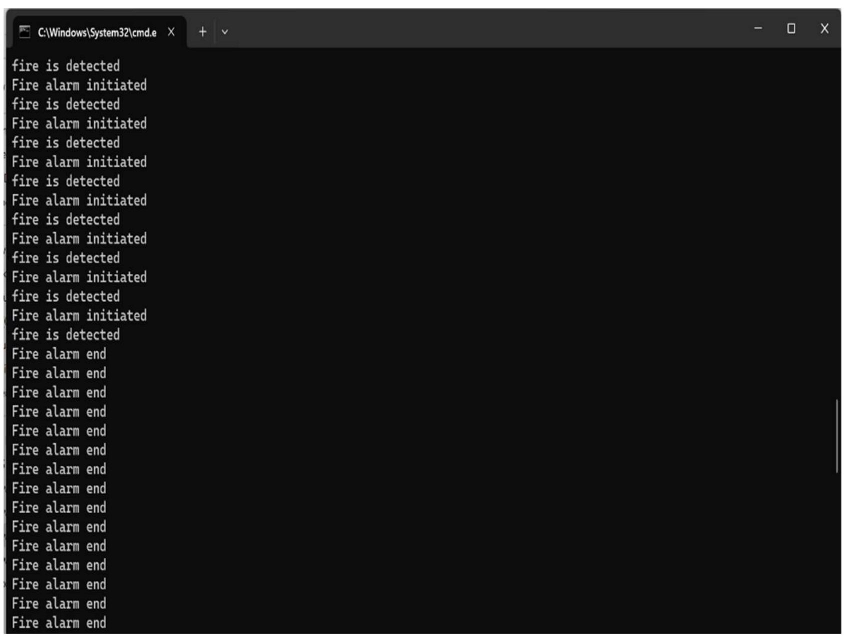

Figure 8.4: Fire Alarm end

9. SYSTEM TESTING

9. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

TYPES OF TESTS

9.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format.

- No duplicate entries should be allowed.
- All links should take the user to the correct page.

9.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

9.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

9.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

9.5 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

9.6 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box,you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

9.7 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

10. CONCLUSION AND FUTURE WORK

10.1 CONCLUSION

The deep learning-based forest fire response system represents a significant advancement in forest fire management. By leveraging deep learning algorithms and real-time monitoring, the system enhances detection accuracy, response time, and resource allocation. This project has the potential to contribute to more effective forest fire response efforts, safeguarding lives, property, and the environment.

10.2 FUTURE WORK

Currently, we not used the systems like a sprinkler water discharge systems, but in future it can be included. Develop mobile applications that allow users to report potential fire incidents, share geotagged images or videos, and receive real-time alerts and safety instructions. Engaging the public in the fire detection and response process can improve early reporting and facilitate a more proactive approach.

11. BIBILOGRAPHY

[1] V.Vipin, "Image processing-based forest fire detection using BN," International Journal of Emerging Technology and Advanced Engineering", 2022 pp.871-950.

[2] A.Enis Cetin Fire detection,"The stateof-the-art NBS Technical Note", US. Dec 2021, Pp.187-843.

[3] Pvk.Borges," A probabilistic approach for vision-based fire detection in videos", Aug 2020, pp.721-73.

[4] S. Kethavath, and M. Dua. "Early Discovery of Disaster Events from Sensor Data Using Fog Computing," International Conference on Intelligent Computing, Information and Control Systems. Springer, Cham, 2019.

[5] S. Kethavath, and M. Dua. "Fog Computing and Deep CNN Based Efficient Approach to Early Forest Fire Detection with Unmanned Aerial Vehicles," International Conference on Inventive Computation Technologies. Springer, Cham, 2019.

[6] M. Khan, J. Ahmad, and S.W. Baik. "Early fire detection using convolutional neural networks during surveillance for effective disaster management," Neurocomputing 288 (2018): pp. 30-42.

[7] A.G. Howard, M. Zhu, B. Chen, D. Kanlenichenko, W. Wang, T. Weyand, M. Andreetto, and H, Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861 (2017).

[8] S. Frizzi, R. Kaabi, M. Bouchouicha, J. M. Ginoux, E. Moreau, and F. Fnaiech, "Convolutional neural network for video fire and smoke detection,"In IECON 2016 42nd Annual Conference of the IEEE Industrial Electronics Society, Oct 2016, pp. 887-882.

[9] K.Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," CoRR, abs/1409.1556, 2014.

[10] V.Vipin, " Image processing-based forest fire detection,"International Journal of Emerging Technology and Advanced Engineering, 2.2 (2012), pp. 87-95.