A project report

on

FIRE DETECTION USING CASCADE CLASSIFIER

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

by

Batch No: A-9

C.Pavithra	**184G1A0556**
T.Jyothirmai	**184G1A0528**
D.Ajith	**184G1A0501**
R.Gnaneswar	**184G1A0518**

Under the Guidance of

Dr.C.Sasikala M.Tech., Ph.D.

Associate Professor

Department of Computer Science & Engineering

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(Affiliated to JNTUA & Approved by AICTE)

(Accredited by NAAC with 'A' Grade & Accredited by NBA(EEE, ECE & CSE))
Rotarypuram Village, B K Samudram Mandal, Ananthapuramu-515701.

2021-2022

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(Affiliated to JNTUA & Approved by AICTE)
(Accredited by NAAC with 'A' Grade & Accredited by NBA (EEE, ECE & CSE))
Rotarypuram Village, B K Samudram Mandal, Ananthapuramu-515701.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Certificate

This is to certify that the project report entitled FIRE DETECTION USING CASCADE CLASSIFIER is the bonafide work carried out by **C. Pavithra** bearing Roll Number **184G1A0556, T. Jyothirmai** bearing Roll Number **184G1A0528, D. Ajith** bearing Roll Number **184G1A0501, R. Gnaneswar** bearing Roll Number **184G1A0518** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2021-2022.

Signature of the Guide

Dr. C. Sasikala M.Tech., Ph.D.
Associate Professor

Head of the Department

Mr. P. Veera Prakash M.Tech. (Ph.D)
Assistant Professor & HOD

Date:
Place: Rotarypuram

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express our gratitude for all of them.

It is with immense pleasure that we would like to express our indebted gratitude to our Guide **Dr. C. Sasikala** M.Tech., Ph.D**, Associate Professor**, **Computer Science & Engineering Department**, who has guided us a lot and encouraged us in every step of the project work. We thank her for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deep felt gratitude to **Mr. K. Venkatesh** M.Tech., **Assistant Professor** project coordinator valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We are very much thankful to **Mr. P. Veera Prakash** M.Tech. (Ph.D)**, Assistant Professor & Head Of the Department, Computer Science & Engineering,** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to **Dr. G. Bala Krishna Ph.D, Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, we thank all other faculty and non-teaching staff, and our friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities**.**

Finally, we wish to convey our gratitude to our family who fostered all the requirements and facilities that we need.

Project Associates

DECLARATION

We, Ms. C. Pavithra bearing reg no: 184G1A0556, Ms. T. Jyothirmai bearing reg no: 184G1A0528, Mr. D. Ajith bearing reg no: 184G1A0501, Mr. R. Gnaneswar bearing reg no: 184G1A0518, students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, Rotarypuram, hereby declare that the dissertation entitled "FIRE DETECTION USING CASCADE CLASSIFIER" embodies the report of our project work carried out by us during IV Year Bachelor of Technology under the guidance of Dr. C. Sasikala M.Tech., Ph.D, Department of CSE and this work has been submitted for the partial fulfillment of the requirements for the award of Bachelor of Technology degree.

The results embodied in this project report have not been submitted to any other Universities of Institute for the award of Degree.

C. PAVITHRA	Reg no: 184G1A0556
T. JYOTHIRMAI	Reg no: 184G1A0528
D. AJITH	Reg no: 184G1A0501
R. GNANESWAR	Reg no: 184G1A0518

CONTENTS **Page No**

LIST OF FIGURES

LIST OF ABBREVIATIONS

OpenCV	Open Computer Vision
CCTV	Closed Circuit Television
RGB	Red, Green and Blue
IOT	Internet Of Things
GSM	Global System for Mobile communication
LCD	Liquid Crystal Display
SMS	Short Message Service
IDE	Integrated Development Environment
CPU	Central Processing Unit
SMTP	Simple Mail Transfer Protocol
ESMTP	Extended Simple Mail Transfer Protocol
XML	Extensible Markup Language
URL	Uniform Resource Locator
WAV	Waveform Audio File Format
MP3	MPEG Audio Layer-3
IP	Internet Protocol
RFC	Request For Comments
UML	Unified Modeling Language
OMG	Object Management Group
OO	Object Oriented
DFD	Data Flow Diagram

ABSTRACT

Accidents by fire can cause damages to the property. Fire gives no time as it spreads rapidly and destroys all the properties that belong to the people. The fire accidents occur due to the fault in wiring, fire flame left by the smoking people and car fire accidents cause severe damage to the victims. Fire and alert systems provide a key aspect called fire detection. There are many different types of fire detection system to warn the people through audio or visual. There are many numeral advantages in using Fire detection system. Mostly the fire detection systems use sensors to detect the fire. The fire is detected when the temperature reaches threshold value. The threshold is set at a particular temperature. They have some disadvantages like false fire alarm. These sensors are not suitable for large open areas. Also, sensors can only detect fire when the fire becomes very large.

We can overcome these problems by using object detection in Image Processing which detects fire in video frames. It compares frames with the data provided in the xml file to recognize fire. The whole process involves: 1) Fire pixel detection 2) Detection of moving pixels 3) Send notifications. When a fire is detected, it rings the alarm and sends an email to the fire department mentioning location of the building in the email. We need to have continuous power facility however it is not a major problem now-a-days. This software can be used in hospitals, educational institutions, factories, etc. to detect fire faster and reduce the damage.

Keywords: *Monitoring, Image Processing, Fire detection, Alarm, E-mail, Location*

CHAPTER – 1

INTRODUCTION

Fire is very dangerous that brings great loss of life and properties. Yearly thousand of accidents related to fire happen all over the world due to power failure, accidental fire, natural lightning. So to control fire, various system is developed and being developed. The existing systems are the smoke sensors types and sprinkler type systems that detect fire from smoke and designed to activate after reaching the threshold set temperature. Still, with this kind of system, there are many disadvantages like a false alarm, space coverage, signal transmission, and also the delay in a fire alarm. As smoke detectors are placed in the ceiling, smokes take time to reach up to the ceiling, which results in time delay. And another problem of the existed system is tough to implement in the open environment and large infrastructures like stadiums, aircraft hangers due to the vast area covered by these infrastructures.

To overcome all the above mentioned problems, we are using image processing technology to detect the fire. Image processing is the process of manipulating images using computers and operates on images to extract the data or the information and execute some tasks from the related images. Using the image processing technology in fire detection opens many possibilities. The technology can implement in hazardous areas where the heat and temperature are very high, and there is always a chance of getting fire. Those places should be monitored continuously because of the high-risk zone with this technology. The system can give the right information about the site.

Computer vision is a process by which we can understand the images and videos how they are stored and how we can manipulate and retrieve data from them. OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, python is capable of processing the OpenCV array structure for

analysis. To identify image pattern and its various features we use vector space and perform mathematical operations on these features.

So, in our project we are using image processing technology and different libraries like OpenCV, smtplib, threading, playsound, imghdr to detect fire, ring the alarm and send an email with the photo of the scene as a proof. The importance of the proposed thesis is to make a reliable, safe, and smart system to reduce limitations and faults like false alarms, which cause panic among the people and even the loss of money with the use of new technology. And make the places safe from the hazardous fire.

1.1 Objectives and Scope of the Project

In this project, a novel method is developed to overcome the problem of detecting fire in real time. The main target of the project is to develop an intelligent system that can localize fire and based on this finding, it can inform that whether there is any fire or not.

1.1.1 Objectives

The objective of this project is to create a system which would be able to detect fire using images from a feed, such as a video with negligible response time. There is no point to having detected that a fire has occurred only after the fire has grown too large, as response time is needed in order to react to the fire. The system must not detect fire where there is none. This means that if there is no fire, another object or objects that look like fire should not be considered as fire, since this would mean that the system is not working correctly and giving false responses.

The main objective of the project is to develop a fire detection technique in real time. The main points of the project objectives are given below:

- To detect fire while they are still small and have not grown too large.
- To develop a low cost fire detection system in real time

1.1.2 Scope

The fire detection covers a very large field, and it would be impossible to cover all aspects in one project. The focus in this project is detection using a low cost camera. A low cost camera is easily available in the market. This would mean that the program does not only work with expensive technology such as infrared cameras or other such cameras. The cameras that are required to at least work with this program are the CCTV cameras, such as those in shopping complexes or malls. One factor that needs to be taken into consideration is that, unlike other fire detectors, this system is not a point type detector. It should be able to detect fire in large open spaces, so that the whole scenario must be considered, and not just a single point on the image from the video feed.

With this, the scope has been narrowed down to a manageable level for this project. The intention and aim for this project is now very clear.

CHAPTER – 2

BACKGROUND STUDY

2.1 Introduction

In this chapter, the background knowledge related to the project work will be discussed. At first the basic concepts related to computer vision will be elaborated. Then the theory and concepts related to the proposed system will be discussed.

2.2 Computer Vision

Computer vision is the science and technology of machines for seeing. The field of computer vision works on the theory that extract information from images. The image may take variety of forms, such as sequences of meaningful frames that constitute a video, views from multiple cameras, static images captured by camera etc. The applications of computer vision include: controlling processes, detecting events, organizing information, modeling objects or information, interaction etc [2].

Computer vision is the inverse of computer graphics. While computer graphics produces image data from 3D models, computer vision often produces 3D models from image data. Related fields of computer vision are: artificial intelligence, pattern recognition, physics, neurobiology, signal processing, image processing.

Typical functions which are found in many computer vision systems are: image acquisition, pre-processing, feature extraction, segmentation, detection, morphological operation, motion tracking, high-level processing.

2.3 Machine Learning

Machine learning is the study of the algorithms and the statistical models that computer system uses for the specific task without the use of instruction or relying on some patterns and interference instead [2]. A machine learning algorithm is created by collecting the data and then represent in the mathematical model, which helps to make the predictions and make the decision referencing from the mathematical model. Machine Learning is used in a wide variety of field like in object detection, sorting of packages, filtering of documents, predicting the patterns which help perform the task very fast and more precisely than human. Machine learning is the science of making

computers learn and act like humans by feeding the data and information without being explicitly programmed.

2.4 Image

To understand how a computer understands and stores the image first, we need to know how an image is made. Image is the collection of small pixels where a pixel is the smallest single element of the digital image, also known as a picture element. The images on the computer are just the combinations of binary numbers 1 and 0's. To elaborate, let's take an example.

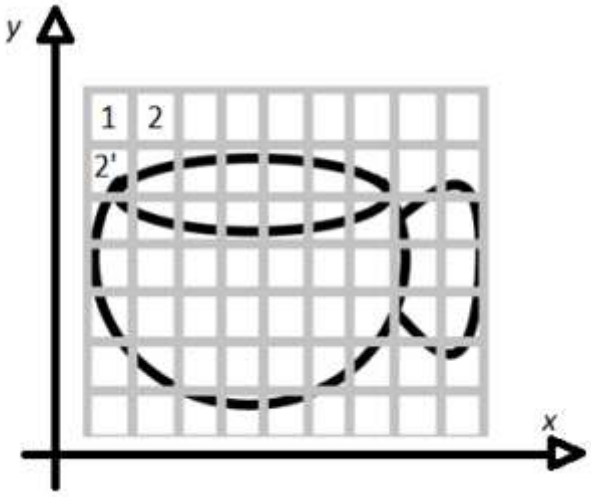

Fig 2.1: picture of a cup in a grid

Figure 2.1 shows the picture of a cup that makes it easy to understand, an image, for a computer. The x-axis represents the rows, and the y-axis represents the column if we considered row and column as a pixel then there are 7*9= 63 pixels. A digital image can be regarded as a matrix having row and column with the function f[x,y] of two continuous variables x and y in which the computer identifies a point from the image. The corresponding matrix value determines the value of darkness at the location. In figure 2.1 the image of a cup is considered to be in grayscale. For computers, the grayscale image has only one value of either black or white, but if the image is colored, then it has three channels of Red, Green, and Blue (RGB) color. So each pixel contains a particular value, the computer gives value according to the level of darkness. Here in Figure 2.1, the image is in grayscale, the first row of the picture.

Pixel 1 is color white so that the computer gives some value for that individual pixel, and similarly, the second column pixel 2' of the image is black again; it gives a certain value for that pixel. As a result, for computer picture is a combination of numbers that are further processed and provides result with in the form of images.

2.5 Image Processing

Image processing is the process of manipulating images using computers and operates on images to extract the data or the information and execute some tasks from the related images. A digital image is an array of the real and complex numbers represented by a finite number of bits. Digital image processing can be referred to as a numerical representation of the object to perform a series of operations by using different algorithms to get the desired output. With the development of image processing technology, it is used in various core research and development programs such as in engineering, medical, transportation, forensic examination, photography, weather forecasting, and other mobile technologies and many more areas.

These are the fundamental steps involved in image processing.

2.5.1 Image Acquisition

Image Acquisition is the first step, and the process of digital image processing, an image first captured by a photo sensor and send into digitizer for further processing to give the output. There are processes like scaling, conversion to the grayscale from RGB, preprocessing to give the final output of image.

2.5.2 Image Enhancement

Image enhancement is the process of adjusting the digital image by changing its brightness and contrasts, removing noise, and sharpen the image. The primary aim of an image enhancement technique is to bring out details of an image so that the image is more suitable for display and analysis as shown in Figure 2.2.

Fig 2.2: image enhancement

2.5.3 Object Recognition

Object recognition is the process of identifying the object using computer vision technology and assigns a label to an object for the description.

2.5.4 Knowledge Base

Knowledge is the simple detailing regions of an image from where the information of interest is located, thus limiting the search which is conducted for seeking information. While it may be complex, such as an interrelated list of all significant defects in the materials inspection problem or an image with many databases such as high-resolution images of a region.

CHAPTER – 3

LITERATURE SURVEY

R Raja John Naveen, R Santhosh, M Vallarasu, S Jayanthi, "Home Fire Detection using Image Processing". They described a system in which they have used camera for flame and smoke detection as the initial fire comes out from smoke. They have used preprocessing techniques where hundreds of fire and smoke images are processed. This preprocessing is used to detect fire pixels using its color information, detection of the moving pixels and analysis of fire pixels in frames. Whereas in other methodologies sensors are used for sensing fire which can get easily damaged or defective. The method used must be provided which requires a huge amount of current. The accuracy of fire detection cannot be provided in large places like stadium's etc. in old or existing method [1].

Bibek Shrestha, "Fire Detection Using Image Processing''. In this, the project aimed to detect fire with a different approach rather than using an existing system. Currently, systems like a smoke detector and sprinkler water discharge systems are used, which are very useful and work at its best. But there are certain limitations to these systems. So, they used image processing technology for detecting the fire, these limitations can be reduced because in this system camera acts like a human eye, as it detects a fire, the video is captured, and the image is processed using the software alert user. They used Raspberry Pi, because of the size, cost-effectiveness, simplicity, and portability, it can be used everywhere. The prototype successfully detects fire [2].

FengjuBu and Mohammad Samadi Gharajeh, "Intelligent and vision-based fire detection systems: A survey". This paper presented overall view and main features of the intelligent and vision-based fire detection systems. Initially, it represented some types of the environments that should be considered by detection systems. Accordingly, fundamental properties of the buildings, forests, and mines are discussed to give a clear view about problems and challenges in these environments. Afterwards, two categorizes of the intelligent and vision-based fire detection systems are discussed with more details, including intelligent detection systems for forest fires and intelligent fire detection systems for all of the environments. The paper discussed

about effects of various intelligent techniques such as convolutional neural network (CNN), color models, fuzzy logic, and particle swarm optimization on fire detection processes in various environments. These techniques can considerably enhance performance of the fire detection systems, in the most real scenarios [3].

Vishesh Goel, Sahil Singhal, Tarun Jain and Silica Kole, "Specific Color Detection in Images using RGB Modelling in MATLAB". This paper gives an approach to recognize colors in a two dimensional image using color thresh-holding technique in MATLAB with the help of RGB color model to detect a selected color by a user in an image. The methods involved for the detection of color in images are conversion of three dimensional RGB image into gray scale image. Further the color of the pixels is recognized by analyzing the RGB values for each pixel present in the image. The algorithm is implemented using image processing toolbox in MATLAB [4].

Faming Gong, Chuantao Li, Wenjuan Gong, Xin Li, Xiangbing Yuan, Yuhui Ma, and Tao Song, "A Real-Time Fire Detection Method from Video with Multifeature Fusion". To address the problem of a high false alarm rate in traditional fire detection, an innovative detection method based on multifeature fusion of flame is proposed. First, they combined the motion detection and color detection of the flame as the fire preprocessing stage. This method saves a lot of computation time in screening the fire candidate pixels. Second, although the flame is irregular, it has a certain similarity in the sequence of the image. According to this feature, a novel algorithm of flame centroid stabilization based on spatiotemporal relation is proposed, and they calculated the centroid of the flame region of each frame of the image and added the temporal information to obtain the spatiotemporal information of the flame centroid. Then, they extracted features including spatial variability, shape variability, and area variability of the flame to improve the accuracy of recognition [5].

M. Li, W. Xu, K. Xu, J. Fan and D. Hou, "Review of fire detection technologies based on video image". This paper reviewed the fire detection methods based on video images in recent years. Through the review, it is clear to see that video based fire detection technology can be divided into two main areas: the characteristics detection of flame and smoke. This fire detection method can improve the accuracy of

the fire alarm, real-time and robustness. If the optimal algorithms can be adopted for each part of detecting motion area and extracting fire characteristics, the system performance will be further improved [6].

YAN Yunyang, GAO Shangbing, Wang Hongyan and GUO Zhibo, "Contour extraction of flame for fire detection". Color and contour are both the important features of a flame image. The method to extract the contour feature of a flame image is developed based on threshold of flame area. The edges of the burning flames jitter continuously, but their contour are similar each other. The method to detect flames in video sequences is proposed here based on flame's dynamic contour. Many experiments show that the system is able to work well and get high detection rate with a low false positive rate [7].

K. Sravan Kumar, K. Balaram Varma, L. Sujihelen, S. Jancy, R. Aishwarya, and R. Yogitha, "Computer Vision-Based Early Fire Detection Using Machine Learning". By using image processing technology, a fire could be detected early and people would be alerted. The sensor method detects fires through smoke, but it has limited applications, and is only suitable for certain areas. In this proposed work, limitations will be reduced and the technology will be optimized. In this proposed system, a Haar Cascade classifier is used for fire detection. Pycharm IDE is used for implementing this work. The system uses the webcam as a source of input for capturing the video feed from the surrounding environment. Detection of the fire will be exact and accurate without any delay with the proposed work [8].

CHAPTER – 4

PROPOSED SYSTEM

Early detection of fire is one of the humanities oldest and most important problems. Till date, several approaches and various methods have been proposed for implementing fire detection system. Many researches are still being carried out to overcome the limitation of the existing systems. Fire detection methods widely differ from each other. Video based fire detection systems can be useful for detecting fire in covered areas including auditoriums, tunnels, atriums, etc., in which conventional chemical fire sensors cannot provide quick responses to fire.

4.1 Overview

In a vision based fire detection system, at first the image is captured. Since the project focus on a low cost solution, it suggests to use normal video surveillance camera to capture the image. Then the captured image is converted to suitable color space. Noise reduction is performed in the image. Then based on the detection algorithm various information of the image is analyzed. Intensity and motion information of the image is analyzed in almost every fire detection system. Different fire detection algorithms are invented based on these information [2]. In some of them several transformations on the image like Fourier transform, wavelet transform are performed. Some of the algorithm uses neural network, hidden markov model etc to classify a image containing fire or not.

In this system, In the first approach, intensity and motion information are computed from video sequences to detect fire. According to grayscale color information of pixels, high intensity pixels are detected. High intensity pixels are possible fire regions. To ensure about fire, number of fire pixel variation in subsequent frames are calculated. If variation is above some level, fire is detected.

In the second approach, the first step is to determine the motion in the video. This was done by subtracting the image with a previous image, which results in the differences in the two images, which are the parts that have changed. Then intensity check is done on this moving region. Contour detection is performed on the resultant

image. Then the boundary of the moving region is found and the amount of fire pixels in this area is calculated. Since fire region varies in a great extend, based on these system can find whether there is really fire in the moving fire colored region or not.

4.2 Architecture

The architecture of a system describes its major components, their relationships (structures), and how they interact with each other. Software architecture and design includes several contributory factors such as Business strategy, quality attributes, human dynamics, design, and IT environment.

Architecture serves as a blueprint for a system. It provides an abstraction to manage the system complexity and establish a communication and coordination mechanism among components. It defines a structured solution to meet all the technical and operational requirements, while optimizing the common quality attributes like performance and security.

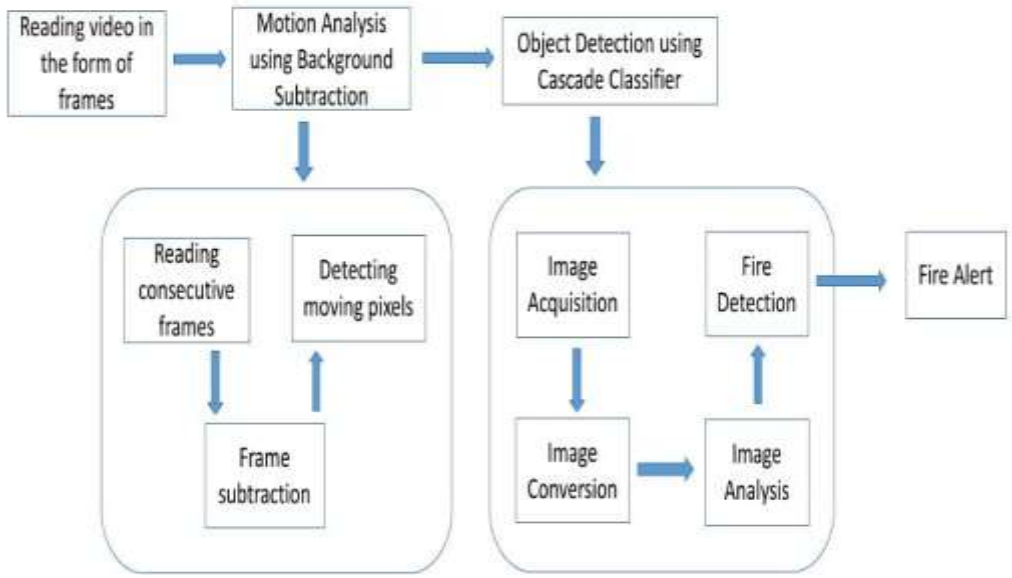

Fig 4.1: architecture of fire detection system

Figure 4.1 shows the complete architecture of the project. The live footage is taken first in a vision-based fire detection system [1]. A video is made up of a series of images known as frames. It appears as a video when we see these frames

all at once. Each frame is made up of a two- dimensional pixel grid. As a capturing live video is nothing more than continuously capturing photos.

After reading frames, the next step is to figure out the motion. The approach for detecting motion here is called 'Background Subtraction.' This was accomplished by subtracting the image from a previous image, yielding the differences between the two photos, which represent the parts that have changed. Changed parts are the pixels that have shifted in comparison to the preceding frame, implying motion. The generated image is subjected to contour detection. The moving region's boundary is then determined, and the number of fire pixels in this area is calculated.

Image processing is used to detect fire in the moving area. Our study focuses on fire detection applying the "Cascade Classifier" object detection method and OpenCV, a library that includes image processing tools. The classifier was trained on a dataset that included hundreds of fire and non- fire images. An XML file is used to import the hundreds of fire and non-fire photos.

A colour image is read and converted to grayscale, after which it is compared against hundreds of preprocessed images. The image is then analysed to determine whether it is a fire image or not [1]. If the image is of a fire, the fire alarm goes off, and the system sends an email to the fire department with the building's location and a snapshot of the fire accident as proof.

CHAPTER – 5

METHODOLOGY AND ALGORITHMS

In this project, we are using image processing approach to detect fire. This machine learning approach for visual object detection is capable of processing images extremely rapidly and achieving high detection rates. In the proposed system, we are using Haar Cascade classifier which is an effective way for object detection. Opencv, an open-source library is used to create HAAR Cascade classifier.

5.1 Training the Classifier

It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. To train the classifier, we use a dataset which consists of hundreds of fire and non-fire images imported in an xml file. This xml file is used in training the classifier making the classifier to recognize and detect fire. The dataset consists of over a hundreds of fire images and non-fire images as shown in Figure 5.1. The negative and positive images are of same size. The hundreds of fire images and non-fire images are imported in an XML file [1].

Fig 5.1: dataset containing fire images

As shown in Figure 5.2, the cascade classifier learns the features of fire from the images in the form of haar features. The classifier uses a learning algorithm called "AdaBoost" to learn the features of fire from the dataset. After training the classifier is able to detect fire in the video frames.

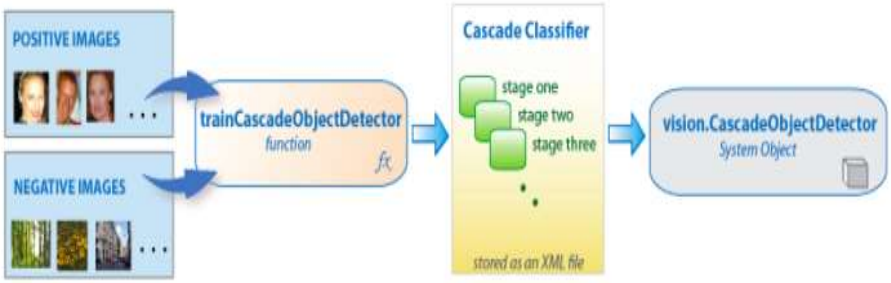

Fig 5.2: training the classifier with dataset

5.1.1 AdaBoost

Yoav Freund and Robert Schapire proposed AdaBoost, or Adaptive Boosting, as one of the ensemble boosting classifiers in 1996. It combines numerous weak classifiers to improve classifier accuracy. AdaBoost is a method for creating iterative ensembles. The AdaBoost classifier creates a powerful classifier by combining several low- performing classifiers, resulting in a high-accuracy classifier. Adaboost's core principle is to establish the weights of classifiers and train the data sample in each iteration so that reliable predictions of uncommon observations may be made.

It works in the following manner.

Model1

Suppose the first model gives the following result, where it is able to classify two blue points on the left side and all red points correctly. But the model also miss-classify the three blue points here in Figure 5.3.

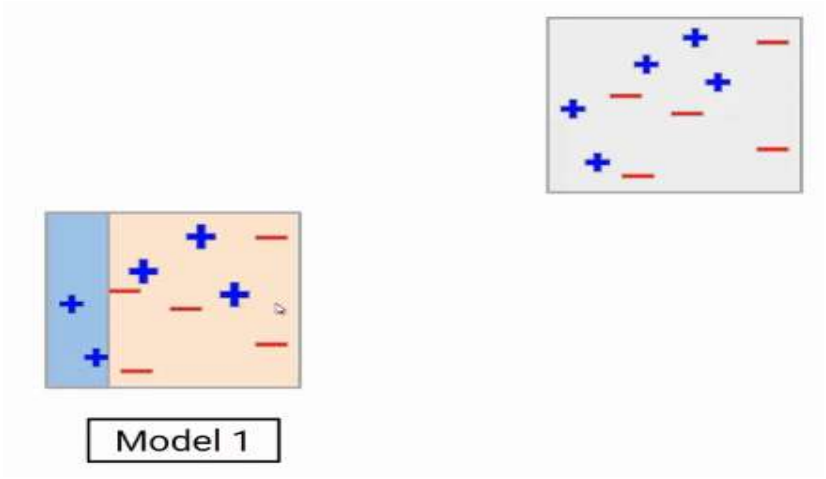

Fig 5.3: Model 1

Model 2

Now, these miss- classified data points will be given higher weight. So these three blue positive points will be given higher weights in the next iteration. For representation, the points with higher weight are bigger than the others in the image. Giving higher weights to these points means my model is going to focus more on these values. Now we will build a new model.

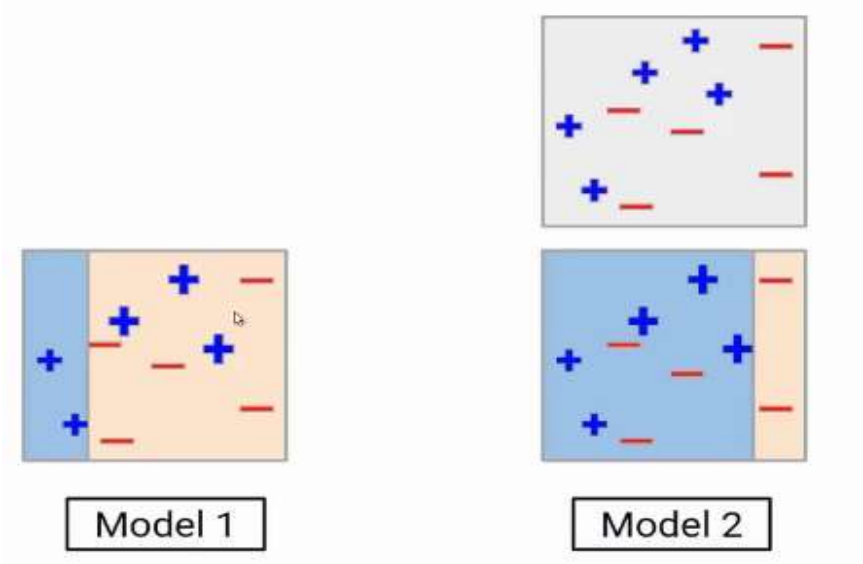

Fig 5.4: Comparing models 1 and 2

In the second model in Figure 5.4 you will see, the model boundary has been shifted to the right side in order to correctly classify the higher weighted points. Still, it's not a perfect model. You will notice three red negatives are miss-classified by model 2.

Model 3

Now, these miss-classified red points will get a higher weight. Again we will build another model and do the predictions. The task of the third model is to focus on these three red negative points. So the decision boundary will be something as shown here.

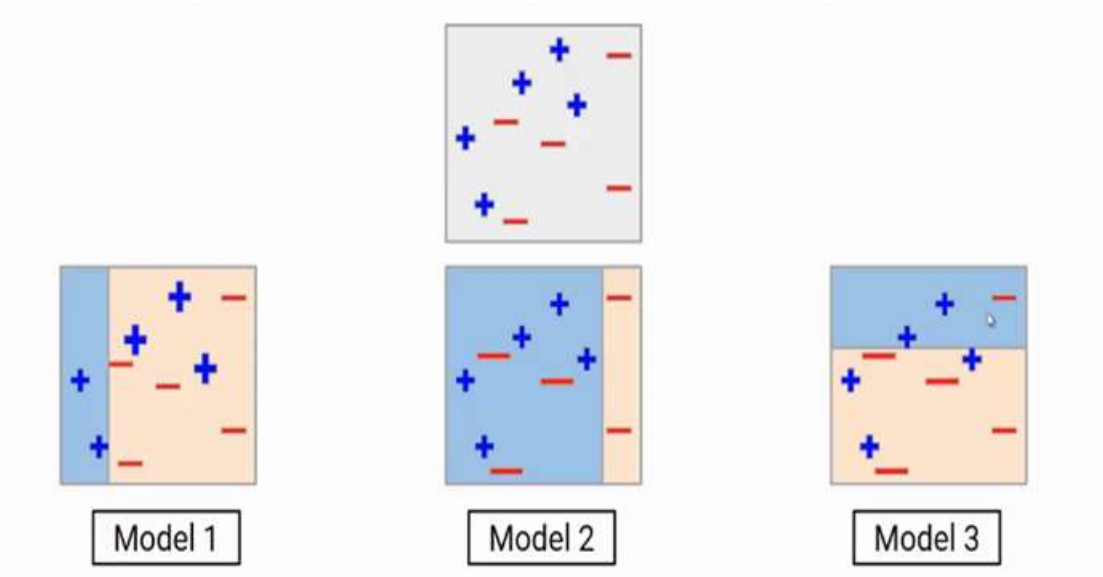

Fig 5.5: Comparing models 1,2 and 3

This new model in Figure 5.5 again incorrectly predicted some data points. At this point, we can say all these individual models are not strong enough to classify the points correctly and are often called weak learners.

Ensemble

And guess what should be our next step. Well, we have to aggregate these models. One of the ways could be taking the weighted average of the individual weak learners. So our final model will be the weighted mean of individual models.

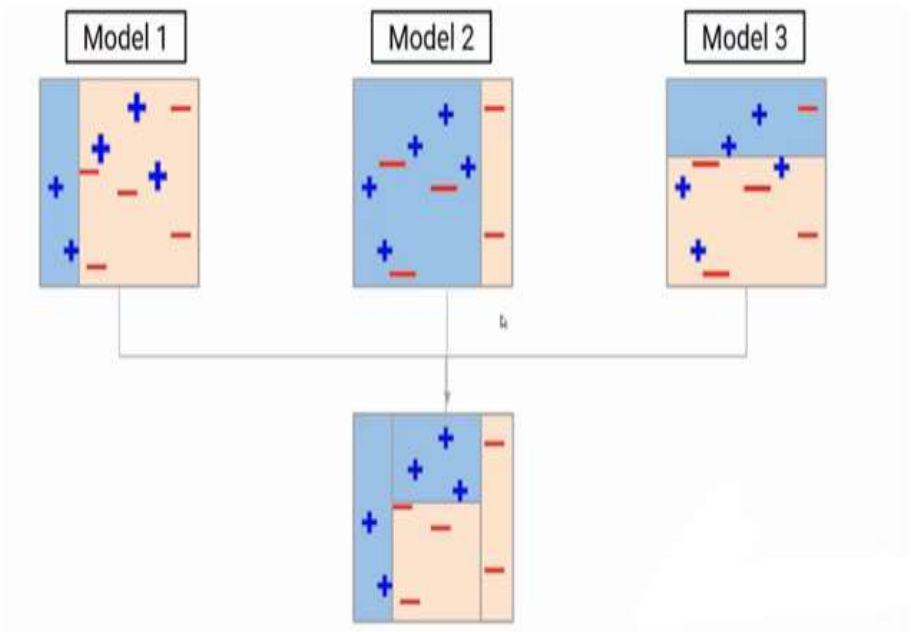

Fig 5.6: ensembling

After multiple iterations, we will be able to create the right decision boundary with the help of all the previous weak learners. As you can see in the Figure 5.6, the final model is able to classify all the points correctly. This final model is known as a strong learner.

With the help of all the preceding weak learners, we will be able to establish the correct decision boundary after numerous iterations. As seen in Figure 5.6, the final model is capable of accurately classifying all of the points. This last model is referred to be a strong learner.

5.2 Capturing Live Video

Python provides various libraries for image and video processing. One of them is OpenCV. OpenCV is a vast library that helps in providing various functions for image and video operations. With OpenCV, we can capture a video from the camera. It lets you create a video capture object which is helpful to capture videos through webcam and then you may perform desired operations on that video.

We need to create a **VideoCapture** object to capture a video. It accepts either the device index or the name of a video file. A number which is specifying to the

camera is called device index. We can select the camera by passing the O or 1 as an argument. After that we can capture the video frame-by-frame.

```
cap = cv2.VideoCapture(0)
```

When we pass the argument as '0', it uses the built-in camera. If the argument is '1', we need to connect an external camera for live capturing.

5.3 Motion Detection

Using motion as a feature in fire detection, helps us to eliminate the false alarms. In many applications based on machine vision, motion detection is used [6]. For example, when we want to count the people who pass by a certain place or how many cars have passed through a toll. In all these cases, the first thing we have to do is extract the people or vehicles that are at the scene.

There are different techniques, methods, or algorithms that enable motion detection. But here we are using a computer vision method called "Background Subtraction" [2].

Background Subtraction

Background subtraction consists of taking an image of the scene without movement and subtracting the successive frames that we are obtaining from a video as shown in Figure 5.7. The image without movement is called the background. The frame that we are going to analyze would be the foreground. Therefore, we have a background from which we are subtracting the different frames.

Fig 5.7: Background Subtraction

Now we will see what are the phases that we must follow to create an algorithm that allows us to detect movement with OpenCV. The process will perform various tasks as follows.

5.3.1 Reading Frames

At first, we need to read the two successive video frames from the live capturing. Then, we are going to start the actual process on these frames.

5.3.2 Determine Motion (change compared to the previous frame)

In this part, we'll do the actual motion detection. We'll compare the previous frame with the current one by examining the pixel values [8]. Since the background is stationary we will be using background subtraction to subtract the unnecessary, here stationary, information from two sequential images. The function that OpenCV provides for this purpose is **absdiff**.

cv2.absdiff is a function which helps in finding the absolute difference between the pixels of the two image arrays. By using this we will be able to extract just the pixels of the objects that are moving.

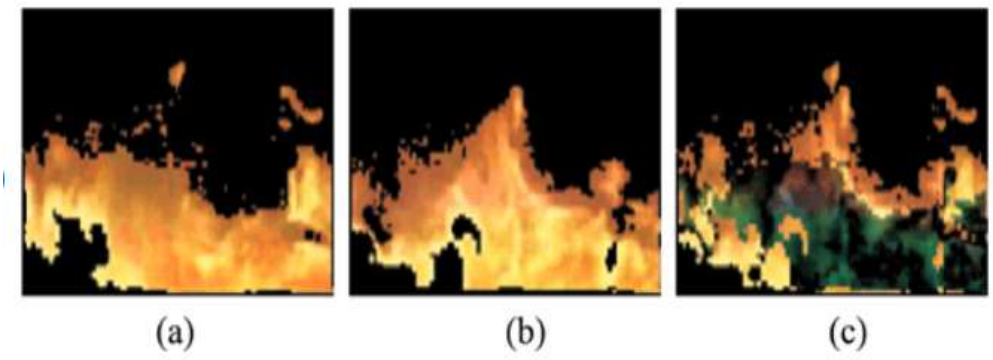

Fig 5.8: frame subtraction

In the Figure 5.8 (a) is the first image and (b) is the second image. When (b) is subtracted from (a) using absdiff method, we obtained a difference frame i.e., (c) which shows only moving pixels.

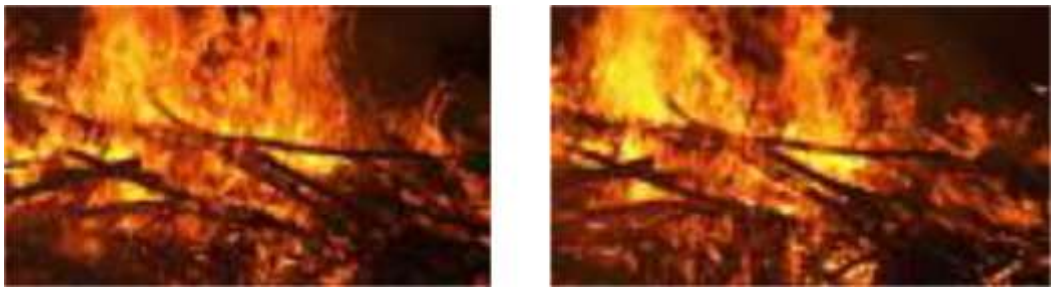

Fig 5.9: images of frame1 and frame2

Fig 5.10: image after motion detection

In Figure 5.9 frame 1 and frame 2 are sequential images and after mapping the corresponding pixels in both of the frames, motion detector compares R, G, and B value of corresponding pixels and give the resultant output to the combination of operator as shown in Figure 5.10.

5.3.3 Grayscale Conversion

Before performing any operation on images, it is a good idea to convert to grayscale [4]. It is less complex and more optimal to work with these types of images. Now we will convert an color image into a grayscale image. OpenCV provides the method called **cvtColor** that allow to convert frames to grayscale. There are two methods to convert it. Both has their own merits and demerits. The methods are:

- Average method
- Weighted method or luminosity method

Average method

Average method is the most simple one. You just have to take the average of three colors. Since its an RGB image, so it means that you have add r with g with b and then divide it by 3 to get your desired grayscale image.

It's done in this way.

Grayscale = (R + G + B / 3)

For example

Fig 5.11: colour image before average method

If you have an color image like Figure 5.11 and you want to convert it into grayscale using average method. The result would appear as in Figure 5.12.

Fig 5.12: gray image after average method

Explanation

There is one thing to be sure, that something happens to the original works. It means that our average method works. But the results were not as expected. We wanted to convert the image into a grayscale, but this turned out to be a rather black image.

Problem

This problem arise due to the fact, that we take average of the three colors. Since the three different colors have three different wavelength and have their own contribution in the formation of image, so we have to take average according to their contribution, not done it averagely using average method. Right now what we are doing is this,

33% of Red, 33% of Green, 33% of Blue We are taking 33% of each, that means, each of the portion has same contribution in the image. But in reality that's not the case. The solution to this has been given by luminosity method.

Weighted method or luminosity method

You have seen the problem that occurs in the average method. Weighted method has a solution to that problem. Since red color has more wavelength of all the three colors, and green is the color that has not only less wavelength then red color but also green is the color that gives more soothing effect to the eyes.

It means that we have to decrease the contribution of red color, and increase the contribution of the green color, and put blue color contribution in between these two.

So the new equation that form is:

New grayscale image = ((0.3 * R) + (0.59 * G) + (0.11 * B)).

According to this equation, Red has contribute 30%, Green has contributed 59% which is greater in all three colors and Blue has contributed 11%. Applying this equation to the image, we get this

Original Image

Fig 5.13: colour image before luminosity method

Grayscale Image

Fig 5.14: gray image after luminosity method

Explanation

As you can see here, that the image in Figure 5.14 has now been properly converted to grayscale using luminosity method on Figure 5.13. As compare to the result of average method, this image is more brighter.

5.3.4 Gaussian Smoothing

Smoothing is often used to reduce noise within an image or to produce a less pixelated image. Most smoothing methods are based on low pass filters. Smoothing is also usually based on a single value representing the image, such as the average value of the image or the middle (median) value. Image smoothing is often used for removal of small noise in image for subsequent image analysis . It's also used for elimination of in homogeneity of background [8].

A Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise and reduce detail.

Fig 5.15 : image before smoothing.

Fig 5.16: image after smoothing.

The visual effect of this blurring technique is a smooth blur resembling that of viewing the image through a translucent screen, distinctly different from the effect produced by an out-of-focus lens or the shadow of an object under usual illumination. Gaussian smoothing is also used as a pre-processing stage in computer vision algorithms in order to enhance image structures at different scales see scale-space representation and scale-space implementation [3].

As you can see here, that the image in Figure 5.16 has now been properly smoothed using gaussian smoothing on Figure 5.15.

5.3.5 Threshold application

Thresholding is one of the simplest and most widely used image segmentation techniques. The goal of thresholding is to segment an image into regions of interest and to remove all other regions which are not essential. Thresholding functions are used mainly for two purposes: One is for masking out some pixels that do not belong to a certain range, for example, to extract blobs of certain brightness or color from the image.

Fig 5.17: image before thresholding

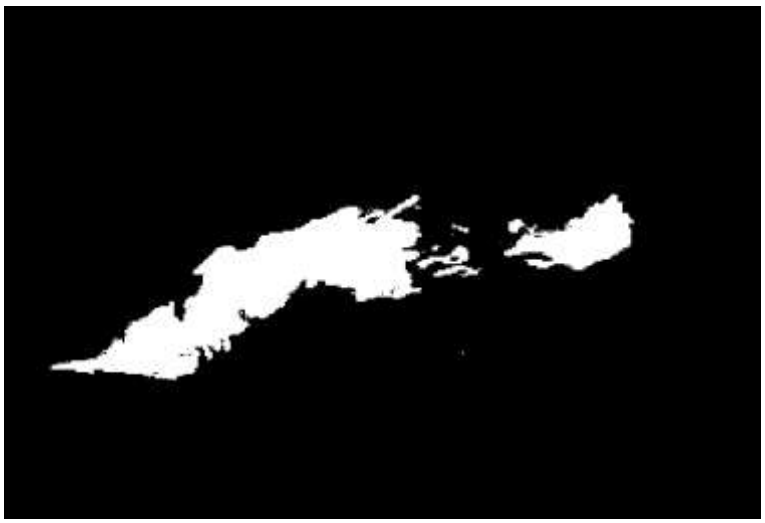

Fig 5.18: image after thresholding

Another is for converting gray scale image to bi-level or black-and-white image. Usually, the resultant image is used as a mask or as a source for extracting higher level topological information, e.g. lines, skeletons contours, etc. The image shown in Figure 5.18 is the image after thresholding is performed in image of Figure 5.17. During the thresholding process, individual pixels in an image are marked as object pixels if their value is greater than some threshold value (assuming an object to be brighter than the background) and as background pixels otherwise. This convention is known as threshold above [5].

Variants include threshold below, which is opposite of threshold above; threshold inside, where a pixel is labeled object if its value is between two thresholds; and threshold outside, which is the opposite of threshold inside. Typically, an object pixel is given a value of 1 while a background pixel is given a value of 0. Finally, a binary image is created by coloring each pixel white or black, depending on a pixel's labels.

5.3.6 Dilation

Dilation is applied to binary images. The main effect of the dilation on a binary image is to continuously increase the boundaries of regions of foreground pixels (for example, white pixels, typically). Thus areas of foreground pixels expand in size while holes within those regions become smaller. The method used for dilation is **dilate**. Dilation helps join broken parts of an object [2].

The effects of dilation include addition of pixels on object boundaries and filling the holes in the foreground and enlarge foreground objects. Having these properties, Dilation can repair breaks and missing pixels in foreground objects.

5.3.7 Contour Detection

Although algorithms like the Canny edge detector can be used to find the edge pixels that separate different segments in an image, they do not tell anything about those edges as entities in themselves. The next step is to be able to assemble those edge pixels into contours. Contours are sequences of points defining a line or curve in an image [7]. Contour matching can be used to classify image objects.

Contour detection in real images is a fundamental problem in many computer vision tasks. Contours are distinguished from edges as follows. Edges are variations in intensity level in a gray level image whereas contours are salient coarse edges that belong to objects and region boundaries in the image. The key to extracting contours appears, from the ground truth, to be the ability to assess what is relevant and what is not in a local neighborhood. The image shown in Figure 5.20 is the image after contour detection is performed in image of Figure 5.19.

Fig 5.19: image before contour detection.

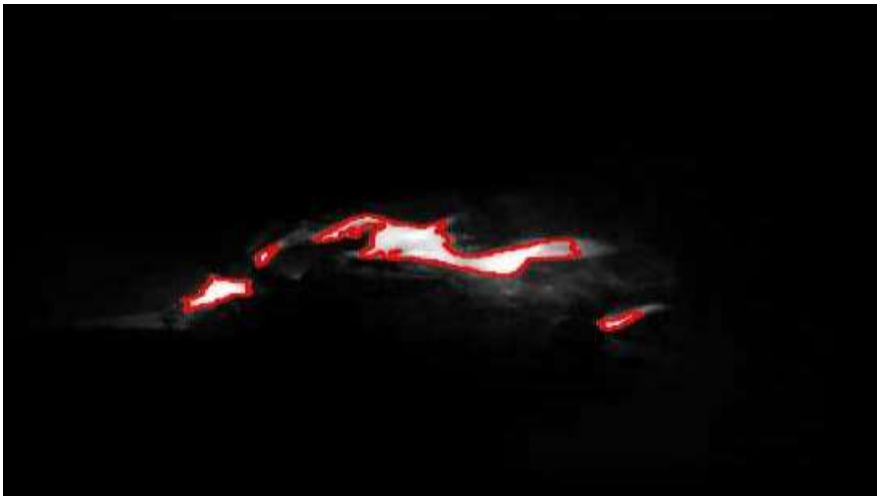

Fig 5.20: image after contour detection.

5.4 Object Detection

After detecting moving pixels , we need to find whether there is fire or not in the frame. So to detect fire, we are using “Cascade Classifier”.

5.4.1 Cascade Classifier

It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. The algorithm outlines a box and searches for the fire in the image mainly, the box is searching for Haar-like features as shown in Figure 5.21.

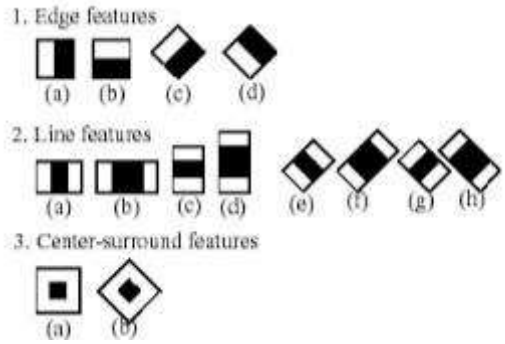

Fig 5.21: haar features

Cascade classifier is used for the accuracy of identification. It is composed of several stages consisting of a strong classifier. Those strong classifiers are passed by so all the features are grouped in several stages where each stage has a certain number of features. The use of these several stages is used to determine whether the given input sub window has features of fire or not if there are no features of fire [8], then the given sub window is discarded and fails to go for other stages. Figure 5.22 shows a schematic description of cascade detection.

Here instead of using all the features in a window, the features are a group in different stages of classifiers as shown in Figure 5.22, there are some stages stage 1 and stage 2. Usually, the first few stages will contain very less numbers of features. If the window fails, it is discarded if not apply the second stage of features and continue the process. The window or stage which passes all the features of fire, then it is detected as fire.

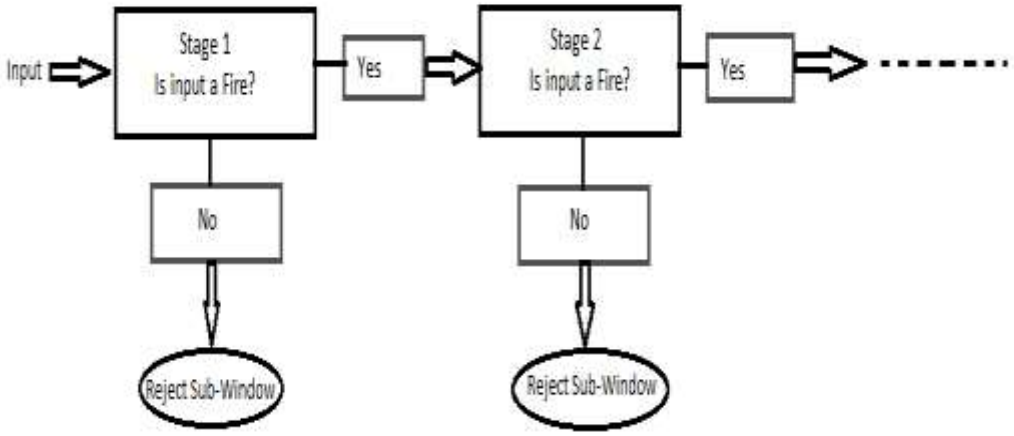

Fig 5.22: schematic description of cascade detection

To detect haar like features, we use an inbuilt function called detectMultiScale() and parameters like scale factor and min neighbor s are passed.

#detectMultiScale(frame, scale _factor, min_ neighbors)

The scale factor will allow rescaling the size of an input frame to detect the fire of any size. The min neighbors parameter directly affects the quality of detected fire.

Fig 5.23: fire pixel detection

While searching for the haar features as in Figure 5.23, if the haar features are found in the same region minimum number of times as min neighbors then the region is said to be contained fire [8].

5.5 Fire Alarm

When the dire is detected, an alarm must ring warning the people to leave out of the buildings. For this purpose, we use playsound module in python.

The playsound module is a cross platform module that can play audio files. This doesn't have any dependencies, simply install with pip in your virtualenv and run!

- The playsound module contains only a single function named **playsound()**.
- It requires one argument: the path to the file with the sound we have to play. It can be a local file, or a URL.
- There's an optional second argument, **block**, which is set to True by default. We can set it to False for making the function run **asynchronously**.
- It works with both **WAV** and **MP3** files.

5.6 Sending Email

Simple Mail Transfer Protocol (SMTP) is a protocol, which handles sending e-mail and routing e-mail between mail servers.

Python provides smtplib module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.

Here is a simple syntax to create one SMTP object, which can later be used to send an e-mail –

```
import smtplib

smtpObj = smtplib.SMTP( [host [, port [, local_hostname]]] )
```

Here is the detail of the parameters −

host − This is the host running your SMTP server. You can specify IP address of the host or a domain name like tutorialspoint.com. This is optional argument.

port − If you are providing host argument, then you need to specify a port, where SMTP server is listening. Usually this port would be 25.

local_hostname − If your SMTP server is running on your local machine, then you can specify just localhost as of this option.

An SMTP object has an instance method called sendmail, which is typically used to do the work of mailing a message. It takes three parameters −

The sender − A string with the address of the sender.

The receivers − A list of strings, one for each recipient.

The message − A message as a string formatted as specified in the various RFCs.

To send the mail you use smtpObj to connect to the SMTP server on the local machine and then use the sendmail method along with the message, the from address, and the destination address as parameters (even though the from and to addresses are within the e-mail itself, these aren't always used to route mail).

CHAPTER - 6

UML DIAGRAMS

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

OMG is continuously making efforts to create a truly industry standard.

- UML stands for **Unified Modeling Language**.
- UML is different from the other common programming languages such as C++, Java, COBOL, etc.
- UML is a pictorial language used to make software blueprints.
- UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system.
- Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc.

UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After some standardization, UML has become an OMG standard.

6.1 Goals of UML

A picture is worth a thousand words, this idiom absolutely fits describing UML. Object-oriented concepts were introduced much earlier than UML. At that point of time, there were no standard methodologies to organize and consolidate the object-oriented development. It was then that UML came into picture.

There are a number of goals for developing UML but the most important is to define some general purpose modeling language, which all modelers can use and it also needs to be made simple to understand and use.

UML diagrams are not only made for developers but also for business users, common people, and anybody interested to understand the system. The system can be a software or non-software system. Thus it must be clear that UML is not a development method rather it accompanies with processes to make it a successful system.

In conclusion, the goal of UML can be defined as a simple modeling mechanism to model all possible practical systems in today's complex environment.

6.2 Object-Oriented Concepts

UML can be described as the successor of object-oriented (OO) analysis and design. An object contains both data and methods that control the data. The data represents the state of the object. A class describes an object and they also form a hierarchy to model the real-world system. The hierarchy is represented as inheritance and the classes can also be associated in different ways as per the requirement.

Objects are the real-world entities that exist around us and the basic concepts such as abstraction, encapsulation, inheritance, and polymorphism all can be represented using UML.

UML is powerful enough to represent all the concepts that exist in object-oriented analysis and design. UML diagrams are representation of object-oriented concepts only. Thus, before learning UML, it becomes important to understand OO concept in detail.

Following are some fundamental concepts of the object-oriented world −

- **Objects** − Objects represent an entity and the basic building block.
- **Class** − Class is the blue print of an object.
- **Abstraction** − Abstraction represents the behavior of an real world entity.
- **Encapsulation** − Encapsulation is the mechanism of binding the data together and hiding them from the outside world.
- **Inheritance** − Inheritance is the mechanism of making new classes from existing ones.
- **Polymorphism** − It defines the mechanism to exists in different forms.

6.3 OO Analysis and Design

OO can be defined as an investigation and to be more specific, it is the investigation of objects. Design means collaboration of identified objects. Thus, it is important to understand the OO analysis and design concepts. The most important purpose of OO analysis is to identify objects of a system to be designed. This analysis is also done for an existing system. Now an efficient analysis is only possible when we are able to start thinking in a way where objects can be identified. After identifying the objects, their relationships are identified and finally the design is produced.

The purpose of OO analysis and design can described as −

- Identifying the objects of a system.
- Identifying their relationships.
- Making a design, which can be converted to executables using OO languages.

6.4 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Activity diagrams are mainly used as a flowchart that consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities. These additional capabilities include branching, parallel flow, swimlane, etc.

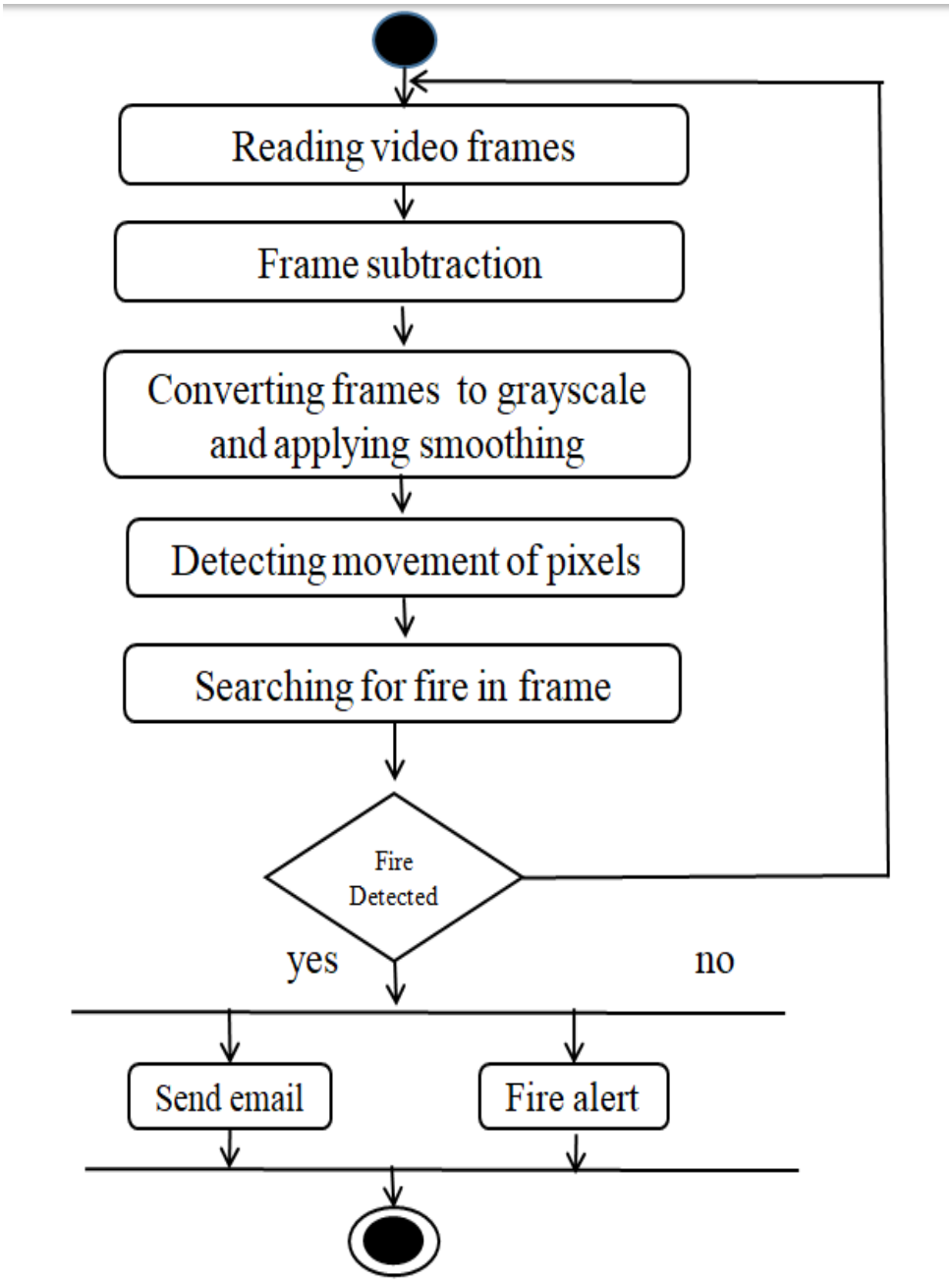

Fig 6.1: activity diagram

As shown in Figure 6.1, at first, we are reading two consecutive frames from the live capturing. Now, in order to detect the moving pixels, we do frame subtraction. After that we covert the frame into grayscale format and apply Gaussian smoothing to reduce the details. Then, contours are drawn around fire to show that there is

motion. Then the frame which contains movement, is searched for fire. If fire is found, then alarm rings and mail is sent to the fire department.

6.5 Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both. It shows how data enters and leaves the system, what changes the information, and where data is stored. The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

6.5.1 Components of DFD

The Data Flow Diagram has 4 components:

- **Process**

  Input to output transformation in a system takes place because of process function. The symbols of a process are rectangular with rounded corners, oval, rectangle or a circle. The process is named a short sentence, in one word or a phrase to express its essence.

- **DataFlow**

  Data flow describes the information transferring between different parts of the systems. The arrow symbol is the symbol of data flow. A relatable name should be given to the flow to determine the information which is being moved. Data flow also represents material along with information that is being moved. Material shifts are modeled in systems that are not merely informative. A given flow should only transfer a single type of information. The direction of flow is represented by the arrow which can also be bi-directional.

- **Warehouse**

  The data is stored in the warehouse for later use. Two horizontal lines represent the symbol of the store. The warehouse is simply not restricted to being a data file rather it can be anything like a folder with documents, an optical disc, a filing cabinet. The data warehouse can be viewed independent of its implementation. When the data flow from the warehouse it is considered as data reading and when data flows to the warehouse it is called data entry or data updation.

- **Terminator**

  The Terminator is an external entity that stands outside of the system and communicates with the system. It can be, for example, organizations like banks, groups of people like customers or different departments of the same organization, which is not a part of the model system and is an external entity. Modeled systems also communicate with terminator.

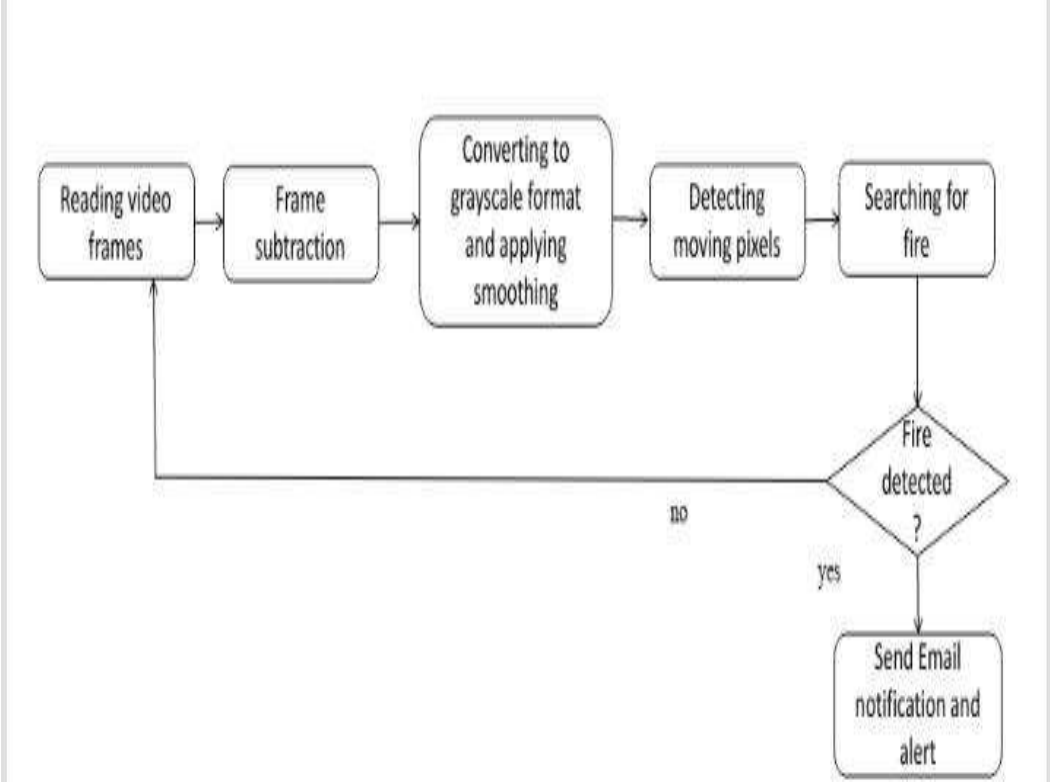

Fig 6.2: dataflow diagram of fire detection system

As shown in Figure 6.2, at first, we are reading two consecutive frames from the live capturing. Now, in order to detect the moving pixels, we do frame subtraction. After that we covert the frame into grayscale format and apply Gaussian smoothing to reduce the details. Then, contours are drawn around fire to show that there is motion. Then the frame which contains movement, is searched for fire. If fire is found, then alarm rings and mail is sent to the fire department.

CHAPTER – 7

IMPLEMENTATION

In this chapter, the implementation of the project work will be discussed. At first software and hardware requirements are specified. As the project is implemented using python in Spyder IDE, the basic knowledge related to these is discussed. As the proposed system uses OpenCV library, basic knowledge related to this library will be focused also.

7.1 Software Specifications

Software	:	Python 2.7 or above versions
Operating System	:	Windows 10
Technology	:	Machine Learning
Programming Language	:	Python

7.2 Hardware Specifications

Camera	:	Logitech C920
Processor	:	Intel i3
RAM	:	4 GB
Hard Disk	:	10 GB

7.3 Python

Python is a deciphered, object-situated, significant level prearranging and programming language. Python was first presented in 1991 by Guido van Rossum, a Dutch PC developer who needed to build up a language that could be utilized by anybody. The primary objective of Python was to diminish the expectation to absorb information by picking a grammar that is justifiable as plain English.

The simple syntax rules of the programming language further make it easier for you to keep the code base readable and application maintainable. There are also a

number of reasons why you should prefer Python to other programming languages. Python is one of the widely used programming languages for image processing. Its amazing libraries and tools help in achieving the task of image processing very efficiently.

7.4 Spyder IDE

Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific of prominent packages in the scientific python stack, including Numpy, Scipy, Matplotlib, Pandas, Ipython, Sympy and as well as some other open-source softwares.

Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community.

Spyder is extensible with first-party and third-party plugins, includes support for interactive tools for data inspection and embeds Python-specific code quality assurance and introspection instruments, such as Pyflakes, Pylint and Rope.

7.5 OpenCV – Open Computer Vision Library

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then (which was later acquired by Intel). The library is cross- platform and free for use under the open-source BSD license.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it

with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

7.6 Modules Used

In Python, Modules are simply files with the ".py" extension containing Python code that can be imported inside another Python Program.

In simple terms, we can consider a module to be the same as a code library or a file that contains a set of functions that you want to include in your application.

With the help of modules, we can organize related functions, classes, or any code block in the same file. So, It is considered a best practice while writing bigger codes for production-level projects in Data Science is to split the large Python code blocks into modules containing up to 300–400 lines of code.

The module contains the following components:

- Definitions and implementation of classes,
- Variables, and
- Functions that can be used inside another program.

To incorporate the module into our program, we will use the import keyword, and to get only a few or specific methods or functions from a module, we use the from keyword.

7.6.1 Threading

Threading allows you to have different parts of your process run concurrently. These different parts are usually individual and have a separate unit of execution belonging to the same process. The process is nothing but a running program that has individual units that can be run concurrently. For example, A web-browser could be a process, an application running multiple cameras simultaneously could be a process; a video game is another example of a process.

Inside a process comes the concept of multiple threading or commonly known as multi-threading, where multiple threads work together to achieve a common goal.

The most crucial benefit of using threads is that it allows you to run the program in parallel.

In Python, the threading module is a built-in module which is known as threading and can be directly imported. Since almost everything in Python is represented as an object, threading also is an object in Python. A thread is capable of holding data, stored in data structures like dictionaries, lists, sets, etc and can be passed as a parameter to a function. A thread can also be executed as a process.

7.6.2 Playsound

Play sound on Python is easy. There are several modules that can play a sound file (.wav).These solutions are cross platform (Windows, Mac, Linux).The main difference is in the ease of use and supported file formats. All of them should work with Python 3. The audio file should be in the same directory as your python program, unless you specify a path.

The playsound module is a cross platform module that can play audio files. This doesn't have any dependencies, simply install with pip in your virtualenv and run!

7.6.3 Smtplib

Simple Mail Transfer Protocol (SMTP) is a protocol, which handles sending e-mail and routing e-mail between mail servers. Python provides smtplib module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.

An SMTP object has an instance method called **sendmail**, which is typically used to do the work of mailing a message. It takes three parameters −

- The sender − A string with the address of the sender.
- The receivers − A list of strings, one for each recipient.
- The message − A message as a string formatted as specified in the various RFCs.

7.6.4 Imghdr

Suppose you are given an image type file and you need to determine the type of that file. In simple words, you need to get the extension of that image type file. This can be used in a project to verify whether the image you have requested for is actually an image and with which extension does it come.

The imghdr module determines the type of image contained in a file or byte stream. The imghdr module defines the following function:

#imghdr.what(filename[, h])

Tests the image data contained in the file named by filename, and returns a string describing the image type. If optional h is provided, the filename is ignored and h is assumed to contain the byte stream to test.

CHAPTER - 8

EXECUTION AND RESULT

The step by step execution of the fire detection is as follows.

Step1: At first the live video is captured as shown in Figure 8.1 and each frame is processed.

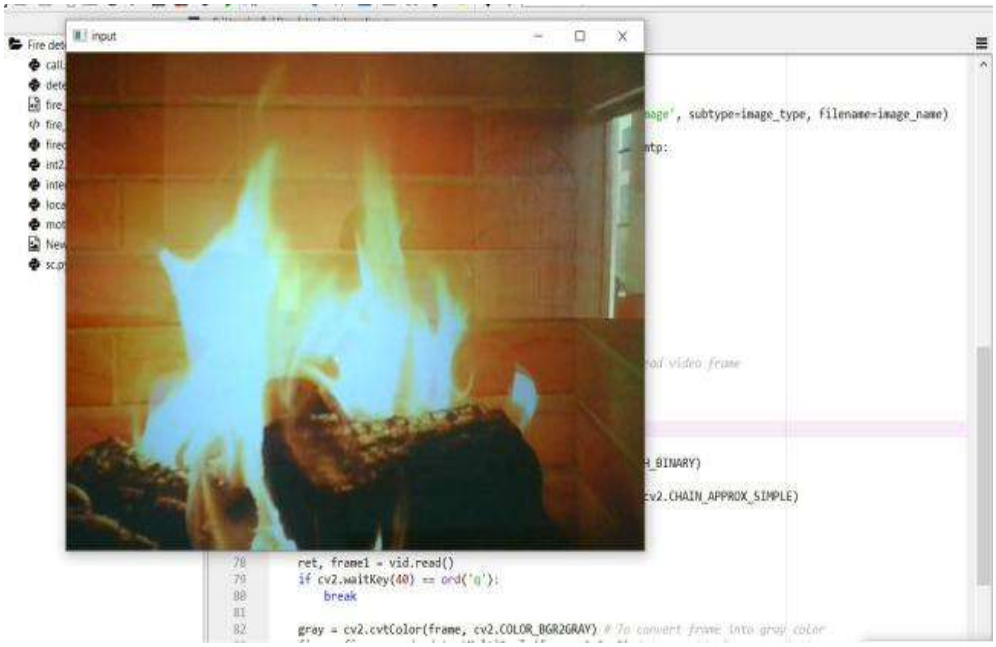

Fig 8.1: video capturing

Step2: The frames are converted to grayscale as in Figure 8.2 and then dilated to sharpen the object in the image.

Fig 8.2: conversion to grayscale

Step3: The contours are drawn along the boundaries of fire as in Figure 8.3 to show it's motion.

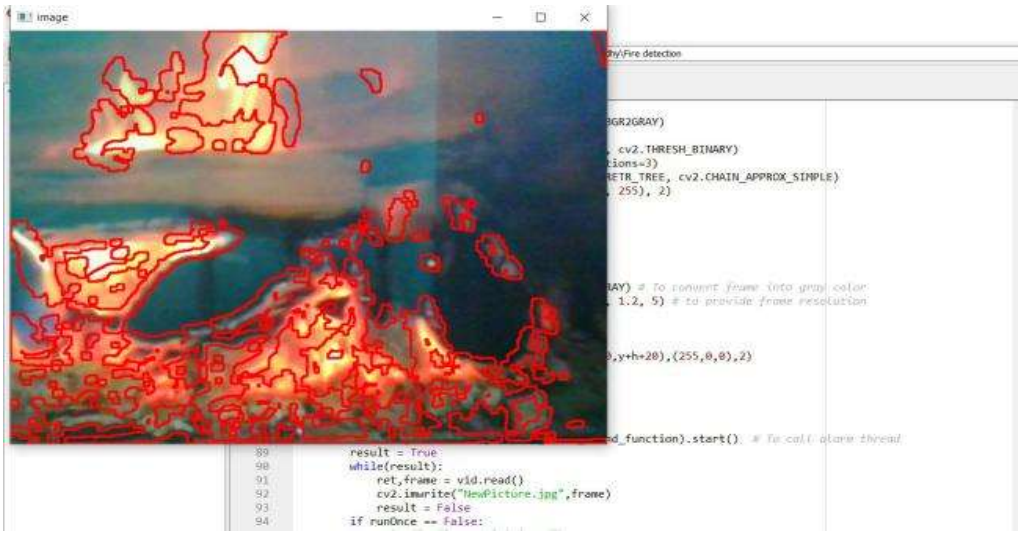

Fig 8.3: contours around fire

Step4: If the pixels in the movement area are of fire, then the fire alarm starts and email send is initiated. It is shown in Figure 8.4. A photo of the scene is captured which is attached to email as a proof .

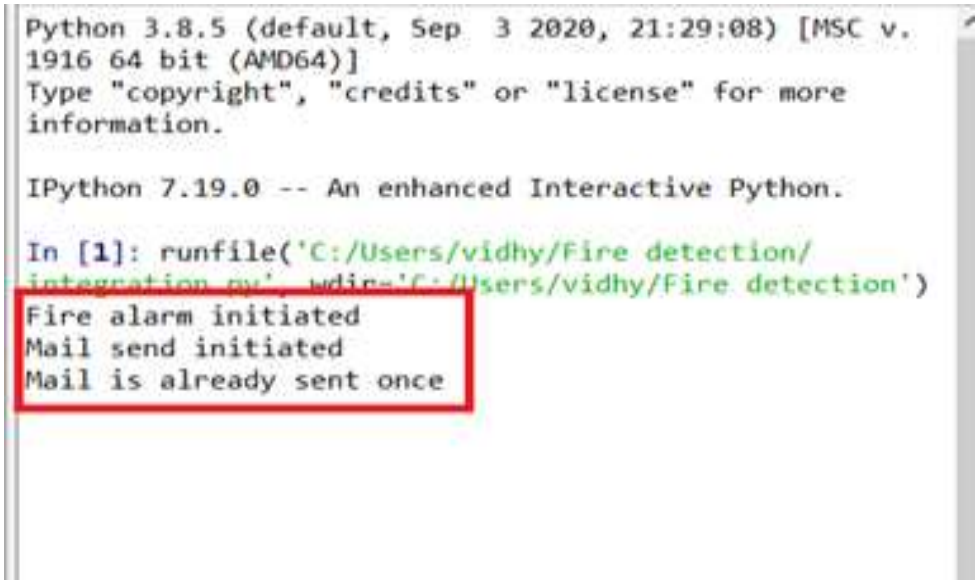

Fig 8.4: alarm and email initiation

Step5: The output is the email that is sent to fire department. Figure 8.5 shows the email sent to the fire department.

Fig 8.5: email

CONCLUSION & FUTURE SCOPE

Conclusion

This project, Fire Detection System has been developed using Image Processing. This system has the ability to apply image processing techniques to detect fire. This system can be used to monitor fire and has achieved 90% accuracy for single webcam. The system works on real time, as it extracts frames in every 2 seconds, it provides continuous monitoring. This system has high efficiency as it has incorporated techniques of Object detection, Motion detection. For better performance outcomes use of RGB is made in the detection techniques, as per their suitability, efficiency and properties. The different parameters like threshold value, blind-spots will be handled properly in our future research. Thus application of proposed fire detection system gives us a better system performance in term of fewer false alarms and thus a higher system performance is achieved.

Future Scope

For further accuracy use of Neural Networks for decision making can be made and GSM module can also be implemented for sending SMS to nearby fire station in case of severe fire. Water sprinklers can also be incorporated. By research and analysis, the efficiency of the proposed Fire detection system can be increased. The margin of false alarms can be reduced even further by developing algorithms to eliminate the false positives. By proper analysis, suitable location height and length for camera installment can be decided, in order to remove blind-spot areas.

REFERENCES

[1] R Raja John Naveen, R Santhosh, M Vallarasu, S Jayanthi, "Home Fire Detection using Image Processing", IJSDR | Volume 6, Issue 3 ISSN: 2455-2631 March 2021.

[2] Bibek Shrestha, "Fire Detection Using Image Processing'', Metropolia University, Feb 2020.

[3] FengjuBu ,Mohammad Samadi Gharajeh, "Intelligent and vision-based fire detection systems: A survey", Image and Vision Computing Volume 91, November 2019.

[4] Vishesh Goel, Sahil Singhal, Tarun Jain, Silica Kole, "Specific Color Detection in Images using RGB Modelling in MATLAB", International Journal of Computer Applications, March 2017.

[5] Faming Gong, Chuantao Li, Wenjuan Gong, Xin Li, Xiangbing Yuan, Yuhui Ma, and Tao Song , "A Real-Time Fire Detection Method from Video with Multifeature Fusion", 14 Jul 2019.

[6] M. Li, W. Xu, K. Xu, J. Fan, and D. Hou, "Review of fire detection technologies based on video image", JATIT, vol. 49, no. 2, pp. 700-707, Mar. 2013.

[7] YAN Yunyang, GAO Shangbing, Wang Hongyan and GUO Zhibo, "Contour extraction of flame for fire detection", Advanced Materials Research, Manufacturing Science and Technology, vol, 383-390, pp. 1106-1110, 2012.

[8] K. Sravan Kumar, K. Balaram Varma, L. Sujihelen, S. Jancy, R. Aishwarya and R.Yogitha, "Computer Vision-Based Early Fire Detection Using Machine Learning", International Conference on Communication, Computing and Internet of Things (IC3IoT), DOI: 10.1109/IC3IOT53935.2022.9767886, 12 May 2022.