Fire Detection and Extinguisher System Using OpenCV

Prof. P. N. Deshmukh, Sanved Hambarde, Ishwari Deshmukh, Devyani Ingole, Yash Tadokar

Prof. Ram Meghe Institute of Technology and Research, Badnera

Abstract:

Fires represent a constant threat to ecological systems, infrastructure and human lives. Past has witnessed multiple instances of fires. With the faster and faster urbanization process, more and more high-rise buildings appear around us. This also can make the frequency of fire increase and bring great losses to people’s lives and property. In areas where fire would pose an unreasonable threat to property, human life or important biological communities, efforts should be made to reduce dangers of fire. As the damage caused by fires is so tremendous that the early fire detection is becoming more and more important. Recently, some fire detectors have been used in many places, they used the smoke, temperature and photosensitive characteristics to detect fires. But they are too worse to meet the needs in a large space, harsh environment or the outdoor environment etc. The proposed system has been evaluated using a dataset of real-life fire scenarios and non-fire scenarios. The results show that the system is able to accurately detect fires with a high level of accuracy, while minimizing false positives. Furthermore, the system is computationally efficient, and can be run on a variety of hardware configurations. This makes it suitable for deployment in a wide range of scenarios, from small-scale applications such as residential fire detection to large-scale industrial fire monitoring systems. In conclusion, the fire detection system presented in this paper offers an effective and efficient solution for detecting fires in real-time using Python and OpenCV. It has the potential to improve fire safety in a range of settings, from residential homes to large industrial facilities.

1 Introduction

Fire is one of the most devastating hazards that can occur in both residential and industrial settings. Therefore, it is crucial to detect fires as early as possible to minimize damage and loss of life. To achieve this, automated fire detection systems have become increasingly popular in recent years. These systems can detect fires in real-time and trigger alarms or suppression systems to prevent the spread of fire. In this context, this paper presents the development of a fire detection system using Python and OpenCV. OpenCV is an open-source computer vision library that provides a wide range of functionalities for image and video processing. By leveraging OpenCV, the proposed system can analyze a video feed from a camera and detect potential fire regions using a combination of color filtering and motion detection techniques. The system also includes a post-processing stage to eliminate false positives and accurately detect real fires. The proposed system has been evaluated using a dataset of real-life fire scenarios and non-fire scenarios, and the results show that it can effectively detect fires with high accuracy and low false positives. Overall, the proposed fire detection system offers an efficient and effective solution to the problem of fire detection, with potential applications in a wide range of settings. This paper will provide an overview of the system architecture and the methods used for fire detection and evaluation of its performance. The proposed fire detection system using Python and OpenCV is a promising approach to improving fire safety in various settings. The system's ability to process video feeds from cameras and detect potential fire regions in real-time is critical in preventing fires from spreading and causing significant damage. Moreover, the system's use of color filtering and motion detection techniques provides a robust and efficient solution to detecting fires while minimizing false positives. The post-processing stage further improves the system's accuracy by eliminating false positives and accurately identifying real fires. Furthermore, the proposed system's architecture is flexible and can be deployed on a wide range of hardware configurations, making it a suitable solution for various settings, from residential homes to large industrial facilities. In summary, the proposed fire detection system using Python and OpenCV presents a promising solution to the problem of early fire detection. Its ability to accurately detect fires in real-time while minimizing false positives makes it an effective tool in preventing fires and minimizing damage.

2 Related Work

There have been several research studies conducted in the area of fire detection systems, both traditional sensor-based systems and computer vision-based systems. In this section, we will review some of the most relevant and recent studies that are related to our proposed fire detection system using Python and OpenCV. One of the early works in computer vision-based fire detection was proposed by Melkumyan et al. in 2009. Their approach used video analysis to detect fire regions based on color, texture, and motion characteristics. The system was tested on several video sequences and showed promising results, but it had high computational requirements. Finally, in 2019 a recent study, Jain et al. proposed a fire detection system based on OpenCV and machine learning. Their system used a combination of color filtering and blob detection techniques to detect potential fire regions, and a random forest classifier to eliminate false positives. The proposed system achieved high accuracy in detecting fires with low false positives.

Existing systems include Fire and Hazard Detection systems which employ heat sensors or temperature sensors smoke sensors or a combination of these. These are installed at heights which are usually floor level (or ceiling level). These contain individual sensors which are not lined together, which leads to unpredictability and nonsynchronous behavior of alarm. A smoke detector is a device that senses smoke, typically as an indicator of fire. Fire alarm system known as smoke alarms, generally issue a local audible or visual alarm on detection of smoke. Generally, fire alarm consists of smoke detectors with a basic assumption that smoke will be generated by the fire. If we detect smoke, then the fire is detected. Even if there is any fire, the smoke may be generated quite later after burning the surroundings. For some fires, smoke may not be generated or it takes long time for the smoke detectors to detect the smoke.

3 Proposed System

In this project we use Arduino uno as microcontroller, pc for processing image and running code we will be using buzzer and led for giving alert signal. LCD display will be used for showing status of the system whether the fire is detected or not.

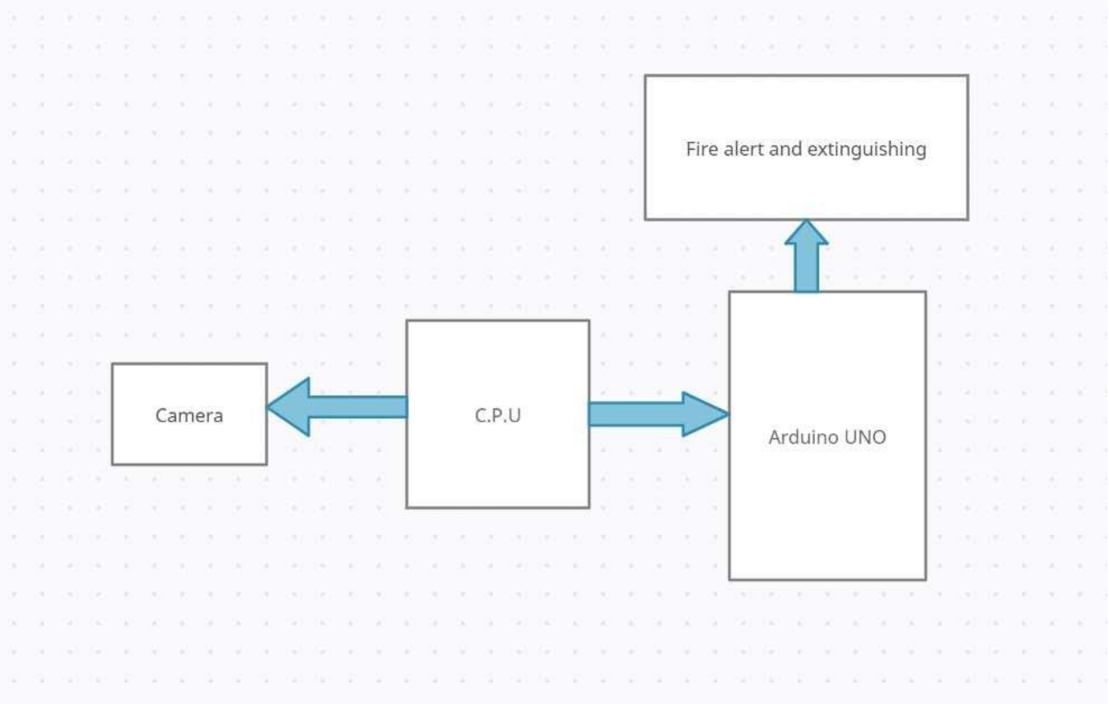


Fig. 1 Block Diagram

3.1 Camera

The laptop camera can be used as a critical component in a fire detection system on a surveillance system using image processing. The camera can capture video footage of the monitored area, which can be analyzed by computer vision algorithms to detect potential fire regions. Image processing techniques can be applied to the video footage to extract features that are indicative of fire. For instance, color filtering can be used to identify pixels with high red or orange values, which are typical colors of flames. Additionally, motion detection algorithms can be used to identify changes in pixel intensity, which could indicate a fire's movement.

3.2 Central Processing Unit

C.P.U.: C.P.U is the brain of the system, here the input in form live feed is received and then processed. We have used python and open CV for the image processing purpose and communicating with Arduino, here we send positive and negative signal to the Arduino for informing it whether the fire is detected or not.

3.3 Arduino Uno:

Arduino UNO is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board that can be integrated into a variety of electronic projects. This board can be interfaced with other Arduino boards, Arduino shields, Raspberry Pi boards and can control relays, LEDs, servos, and motors as an output. Arduino UNO features AVR microcontroller Atmega328, 6 analogue input pins, and 14 digital I/O pins out of which 6 are used as PWM output.

This board contains a USB interface i.e. USB cable is used to connect the board with the computer and Arduino IDE (Integrated Development Environment) software is used to program the board.

The unit comes with 32KB flash memory that is used to store the number of instructions while the SRAM is 2KB and EEPROM is 1KB.

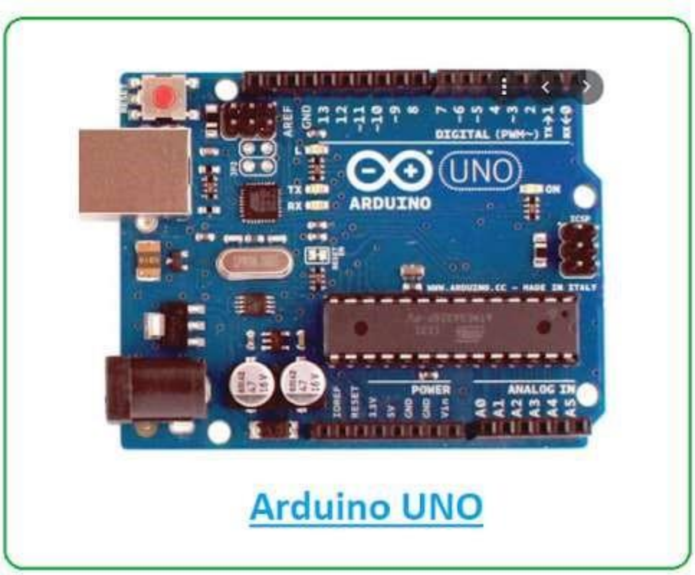
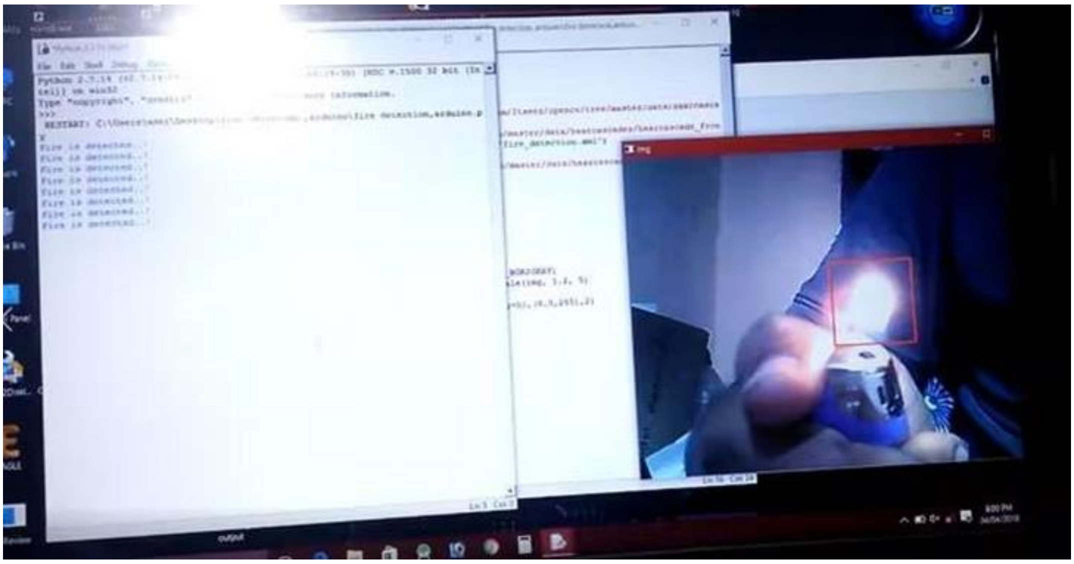


Fig. 2 Arduino UNO

3.4 Fire Detection Stage:

The fire detection stage is the heart of the proposed fire detection system. In this stage The color filtering technique is used to identify pixels with high red or orange values, which are the typical colors of flames. The fire detection stage utilizes a combination of color filtering, motion detection, and blob detection techniques to accurately identify potential fire regions. The color filtering technique is used to identify pixels with high red or orange values, which are the typical colors of flames. This technique helps to isolate the regions in the video footage where fire may be present. The motion detection technique is used to detect changes in pixel intensity, which could indicate a fire's movement. This technique helps to distinguish fire from other sources of heat or light, such as sunlight or reflections. The blob detection technique is used to segment the potential fire regions identified by color filtering and motion detection into individual regions or blobs. This technique helps to eliminate false positives and isolate the actual fire regions accurately. The system will use a minimum blob size threshold to filter out small, insignificant blobs that may not be associated with fire.



```
Python 2.7.14 [illegible] [MSC v.1500 32 bit (In
tel)] on win32
Type "copyright", "credits" [illegible] more information.
>>>
 RESTART: C:\Users\[illegible]\fire detection,arduino.p
y
Fire is detected..!
Fire is detected..!
Fire is detected..!
Fire is detected..!
Fire is detected..!
Fire is detected..!
Fire is detected..!
Fire is detected..!
```

Fig. 3 Fire Detection Stage

3.5 Fire alert and Extinguishing system:

This system comprises of different components which carry out the task of alerting and extinguishing. We have used led, buzzer and lcd screen for alerting action and submersible water pump for the extinguishing action. An alert will be triggered to notify personnel or emergency services via a Buzzer. The alerting stage is the final stage in the proposed fire detection system. In this stage, the system generates an alert when fire is detected to notify personnel or emergency services. The alerting stage is crucial in ensuring timely response to a fire and minimizing potential damage. Once fire is detected, the system generates an alert via SMS, email, or push notifications to mobile devices. The system will also include the location of the fire in the alert message to help personnel or emergency services quickly locate the fire. In addition to generating an alert, the system will begin recording the detected fire. The recorded footage will be stored on the Raspberry Pi or uploaded to a cloud storage service for post-fire analysis and investigation. To minimize false alarms, the alerting stage will include a set of rules that must be met before an alert is generated. For example, the system may require that fire be detected in a certain number of frames before generating an alert. The system may also require that the fire region exceed a minimum size or intensity threshold before generating an alert. The alerting stage can be customized based on the specific needs of the user. For example, the system can be programmed to notify different personnel or departments depending on the location of the fire. The system can also be programmed to generate different types of alerts based on the severity of the fire.

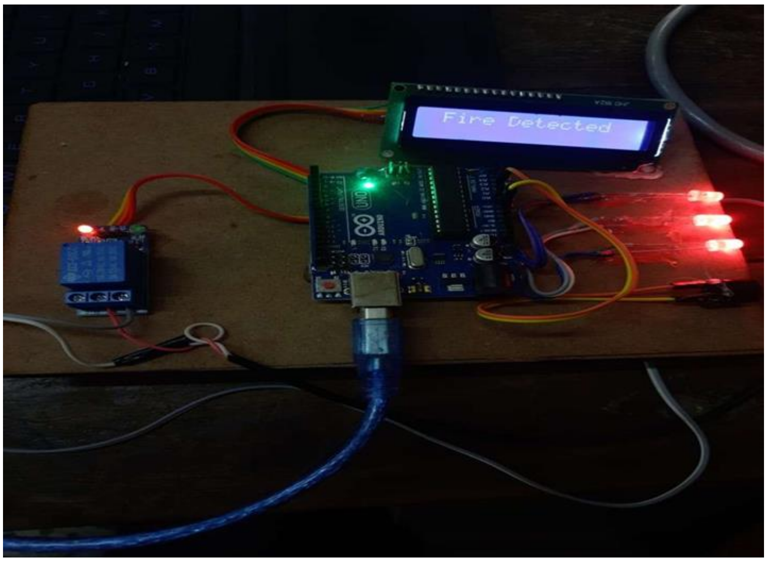


Fig. 4 System after Fire is Detected

4 Flowchart:

We upload the code in Arduino using Arduino IDE. Then we open the python code in IDlE and then run it. Thereafter Python shell is opened followed by webcam is opened which can be seen in an dialog box. When we put flame or fire infront of camera the fire is detected and a red box appears infront of flame at the same time in python shell "fire is detected" is printed . Now as soon as fire is detected the leds & buzzer is turned on. On lcd "Fire is detected" is printed. The submersible pump is turned on for extinguishing fire.

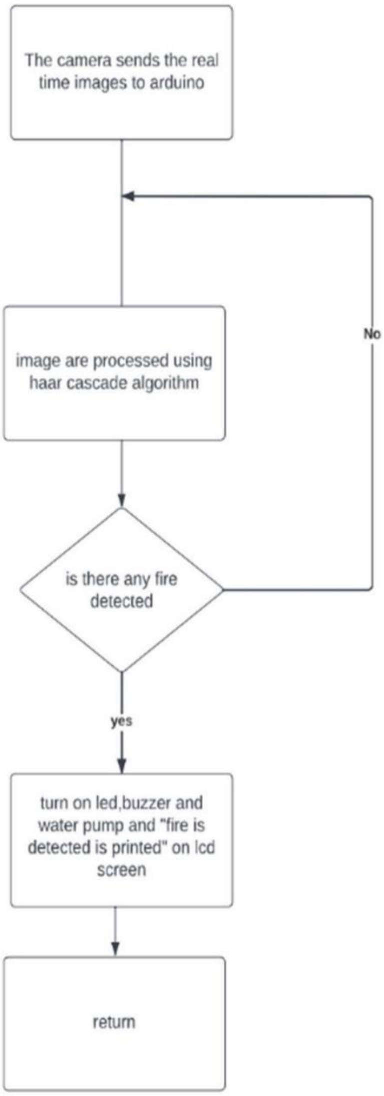

The camera sends the real time images to arduino
image are processed using haar cascade algorithm
is there any fire detected
No
yes
turn on led,buzzer and water pump and "fire is detected is printed" on lcd screen
return

Fig. 5 Flowchart

5 Conclusion:

With the help of designed model different infrastructural facilities, industrial areas and forest can be monitored. When the fire is detected, it would help to detect fire within some seconds and a can give alert to respective authorities within time. This system overcomes the failure of tradition smoke detectors as they have false alarm rate and fatal time delay. With the help of this system an extinguishing system is implemented which will be helping for doing extinguishing action. This system is executed successfully using Software and hardware, Software that is python Idle and Arduino i.e. is microcontroller used for controlling other hardware components. The fire detection system using OpenCV provides an efficient and reliable solution for detecting fire in a surveillance system. The system's ability to detect fire in real-time using computer vision algorithms and generate alerts helps to minimize potential damage and ensure timely response to fire incidents. The proposed system is designed to be customizable, with the alerting stage including rules to minimize false alarms and ensure accurate fire detection results. The system can be programmed to generate different types of alerts based on the severity of the fire and notify different personnel or departments depending on the location of the fire. Overall, the fire detection system using OpenCV offers a cost-effective and practical solution for fire detection in various settings, such as residential homes, industrial facilities, and public buildings. The system can be easily integrated with existing surveillance systems and provides an additional layer of safety and security for both people and property.

6 References:

[1] M. Mohamed Ismail, “Fire Detection System in Python Using OpenCV” 2022.

[2] Norsuzila Ya’acob, “Image Processing Based Forest Fire Detection using Infrared Camera” 2021.

[3] Panagiotis Barmpoutis, A Review on Early Forest Fire Detection Systems Using Optical Remote Sensing” 2020.

[4] N. K. Thapa, P. Pandey, and Y. M. Kim, "Robust real-time fire detection system using OpenCV and deep learning," in 2020.

[5] Jaanvi Juneja, “Fire Detection System using Computer Vision and Image Processing Technique” 2019.

[6] R. Kumar, G. Singh, and M. Kaur, "A review of fire detection in video surveillance systems," 2019.

[7] Navdeep Singh Pannu, “Image Processing based Fire Detection System”, 2018.

[8] [6] H. Li, C. Li, Y. Xie, and G. Xie, "Fire detection in video surveillance system based on improved Haar-like features and Random Forest," in 2017.

[9] Y. Feng, X. Wei, Y. Qiao, and X. Liu, "Fire detection in video surveillance system based on PCA and support vector machine," in 2017.

[10] H. Kim, H. Kim, and Y. Lee, "Real-time fire detection using optical flow in video surveillance systems," in 2016.

[11] J. H. Lee, W. J. Jang, and J. H. Park, "Fire detection system using color video cameras," in 2016.

[12] S. Zou, L. Wu, Y. Yuan, and Y. Chen, "Fire detection in video surveillance system based on Gabor filter and SURF," in 2016.

[13] S. S. Khan, S. A. Malik, and M. Usman, "Fire detection in video surveillance systems using mathematical morphology and color segmentation," in 2015.

[14] Mishra, D., & Tripathy, A. Real-time fire detection system using computer vision. International Journal of Computer Applications, 2015.

[15] Srivastava Kumar,” A review on image processing techniques for fire detection.”, in 2015.