International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Conference Proceedings: Volume: 05 ACME – 2026 DOI: 10.55041/ISJEM.ACME117
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

# Automated Fire Fighting Robot using HAAR Cascade Algorithm

***Ms. k. Manisha***
*Assistant Professor*
*Dept of ECE*
*Annamacharya Institute of Technology and Sciences*
*Tirupati, India*
*manishakasiralla.12@gmail.com*

***Kuruva Mahesh***
*B Tech Student*
*Dept of ECE*
*Annamacharya Institute of Technology and Sciences*
*Tirupati, India*
*kuruvamahesh77@gmail.com*

***Pangala Manju***
*B Tech Student*
*Dept of ECE*
*Annamacharya Institute of Technology and Sciences*
*Tirupati, India*
*pangalamanju@gmail.com*

***Allam Pragathi***
*B Tech Student*
*Dept of ECE*
*Annamacharya Institute of Technology and Sciences*
*Tirupati, India*
*pragathiallam@gmail.com*

***Y Koushik Kumar***
*B Tech Student*
*Dept of ECE*
*Annamacharya Institute of Technology and Sciences*
*Tirupati, India*
*koushikkumary14@gmail.com*

***Abstract***: Fire accidents cause severe damage to residential and industrial environments, requiring rapid detection and timely response. This paper presents a vision-based fire detection and robotic suppression system that combines stationary surveillance with a mobile firefighting unit. A fixed-position camera continuously monitors the area and performs real-time flame detection using a Haar Cascade classifier. To reduce false alarms, a temperature sensor validates abnormal heat rise before suppression is activated. Once confirmed, the fire alert is transmitted wirelessly to a mobile robot controlled by an Arduino Uno. The robot navigates toward the affected location using an onboard camera for obstacle avoidance and safe movement. An ESP8266 module enables wireless communication and IoT-based control via the Thing Speak platform. Upon reaching a safe distance, the robot stops and activates a water pump for suppression. Images and short video clips are simultaneously sent to a Telegram bot for real-time alerts and remote monitoring. The proposed system enhances detection reliability, operational safety, and response efficiency.

Keywords: Fire detection, Haar Cascade, Arduino Uno, ESP8266, IoT, Robotic suppression, Vision system, Temperature verification.

## I. INTRODUCTION

Fire incidents continue to pose serious challenges in both domestic and industrial settings, often resulting in loss of life and property. Conventional firefighting practices rely primarily on manual intervention, which exposes firefighters to hazardous environments such as extreme heat, toxic gases, and structural instability. The development of robotic systems has opened new possibilities for minimizing human involvement in such dangerous operations. Autonomous robots equipped with sensing and detection capabilities can assist in identifying and suppressing fire at an early stage.

Traditional robotic firefighting systems are mostly dependent on temperature sensors, smoke detectors, or infrared flame sensors. While these approaches are effective in certain conditions, they often suffer from limited detection range and delayed response. Vision-based systems, on the other hand, provide the advantage of detecting fire based on its visual characteristics before physical parameters such as temperature or smoke reach critical levels. This work introduces a compact and efficient fire-fighting robot built around the ESP32-CAM module. The system employs image processing techniques using the HAAR Cascade algorithm to detect fire in real time and initiate appropriate suppression mechanisms.

## II. MOTIVATION

Fire incidents often escalate rapidly, especially in residential environments where immediate professional response may not be available. Although conventional fire detection systems use smoke or heat sensors, they may produce delayed detection or false triggering under varying environmental conditions. Furthermore, relying entirely on human intervention in hazardous situations increases the risk of injury and operational delays. There is a growing need for an intelligent system that can detect fire at an early stage while minimizing direct human exposure.

Vision-based detection techniques offer wider monitoring coverage and improved reliability compared to isolated sensor-based methods. Additionally, integrating wireless communication and IoT platforms enables remote monitoring and faster response coordination. The motivation behind this project is to develop a distributed fire detection and suppression system that separates monitoring and intervention tasks. By combining stationary vision-based surveillance with a mobile robotic suppression unit, the proposed system aims to enhance detection accuracy, operational safety, and real-time response efficiency.

## III. OBJECTIVE

The primary objective of this work is to develop a vision-based fire detection and robotic suppression system that integrates stationary monitoring with mobile intervention. The system performs real-time flame detection using a camera and Haar Cascade algorithm, supported by temperature verification to reduce false alarms. A

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Conference Proceedings: Volume: 05 ACME – 2026 DOI: 10.55041/ISJEM.ACME117
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

mobile robot controlled by Arduino Uno is designed to navigate safely toward the detected fire using an onboard camera for obstacle avoidance. Wireless communication through the ESP8266 module enables IoT-based control and remote monitoring via the ThingSpeak platform. The system also activates a water pump for safe-distance fire suppression and transmits real-time alerts, including images and video clips, through a Telegram bot.

## IV. Literature Review

Several studies have proposed robotic systems for fire detection and suppression to reduce human exposure to hazardous environments. Kiran et al. [1] and Sivakumar et al. [2] developed firefighting robots capable of detecting flames and performing directional water spraying. Meshram et al. [3] and Rahman et al. [4] enhanced such systems by incorporating remote and IoT-based control mechanisms. However, many of these approaches rely primarily on onboard sensors, limiting monitoring coverage and detection reliability. Vision-based fire detection techniques have also been explored using image processing methods [5], [8]. The Haar Cascade algorithm, introduced by Viola and Jones [7], [9], has proven effective for real-time object detection due to its computational efficiency, with implementation support provided by OpenCV.

Unlike existing systems that depend mainly on mobile detection, the proposed work introduces a distributed architecture combining stationary vision-based monitoring with robotic suppression, improving coverage, stability, and operational safety.

## V. EXISTING METHOD

The existing fire detection system is based on a sensor-driven robotic platform. In this model, the robot is equipped with temperature sensors, flame sensors, and smoke sensors to monitor environmental parameters associated with fire hazards. The sensors continuously measure variations in heat intensity, infrared radiation, and smoke concentration. When the sensed values exceed predefined threshold limits, the system identifies the presence of fire and triggers an alert or activates a firefighting mechanism. The robot navigates within the environment to detect fire-prone regions using onboard sensors. Although sensor-based robotic systems are widely implemented due to their simplicity and low cost, they suffer from several limitations.

Detection typically occurs only after significant heat or smoke is generated, resulting in delayed response. Furthermore, sensors have limited sensing range and coverage area. Environmental factors such as dust, humidity, and temperature fluctuations may also lead to false alarms. Additionally, the absence of visual monitoring prevents accurate identification of fire location, size, and spread. These limitations highlight the need for an improved vision-based fire detection system capable of early-stage detection and enhanced reliability.

## VI. PROPOSED METHOD

The proposed fire detection system operates using a vision-based approach implemented through a Haar Cascade classifier. The complete working process is illustrated in the flowchart and explained as follows

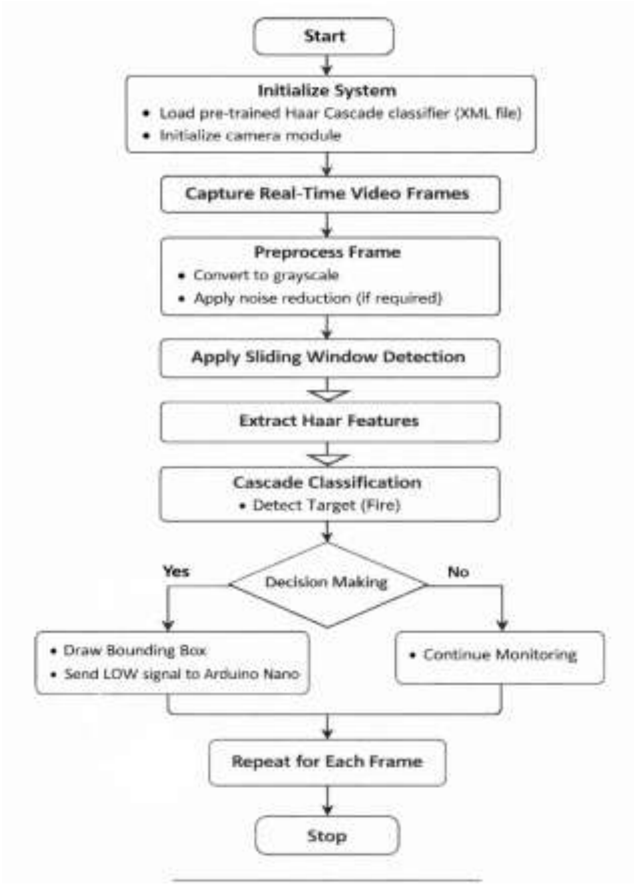



Fig. 1. Proposed System Method

*A. Start*

The system begins execution by powering the hardware module and initializing the software environment required for image processing and fire detection.

*B. Initialize System*

In this stage, the pre-trained Haar Cascade classifier (XML file) is loaded into memory. This classifier contains trained features for detecting fire patterns. Simultaneously, the camera module is initialized to enable real-time video capture.

*C. Capture Real-Time Video Frames*

The camera continuously captures live video frames from the monitoring environment. Each frame is processed individually to detect the presence of fire.

*D. Preprocess Frame*

To improve detection accuracy and reduce computational complexity, preprocessing is performed. The captured frame is converted into grayscale format. Noise reduction techniques are applied if required. This step enhances feature extraction efficiency and minimizes false detection.

*E. Apply Sliding Window Detection*

A sliding window technique is applied across the entire frame. The window scans different regions of the image at multiple scales to ensure that fire objects of varying sizes can be detected.

*F. Extract Haar Features*

Haar-like features are extracted from each scanned region. These features represent intensity differences between adjacent rectangular regions in the image and help in identifying fire-specific patterns.

*G. Cascade Classification*

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Conference Proceedings: Volume: 05 ACME – 2026 DOI: 10.55041/ISJEM.ACME117
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

The extracted features are passed through multiple stages of the cascade classifier. Each stage verifies whether the region corresponds to the trained fire model. If the region satisfies all cascade stages, it is classified as the target (fire).

*H. Decision Making*

At this stage, the system determines whether fire is detected: If fire is detected, the system proceeds to the alert mechanism. If fire is not detected, the system continues monitoring the environment.

*I. Draw Bounding Box and Send LOW Signal*

When fire is detected: A bounding box is drawn around the detected fire region. A LOW signal is transmitted to the Arduino Nano to activate external response mechanisms such as alarms or firefighting modules.

*J. Continue Monitoring*

If no fire is detected, the system continues processing subsequent frames without interruption.

*K. Repeat for Each Frame*

The detection process repeats continuously for every captured frame to ensure real-time monitoring.

*L. Stop*

The system stops operation only when manually terminated or powered off.

## VII. METHODOLOGY

*A. HAAR Cascade Classifiers*

*Stage 1:*

During the learning phase, the differences between dark and light regions in both positive and negative images were analyzed to create the HAAR features (Fig. 1), which are composed of two or three rectangles [8]. To find the specific feature, the HAAR feature continually moves from the upper left of the image to the lower right, even moving pixel by pixel.

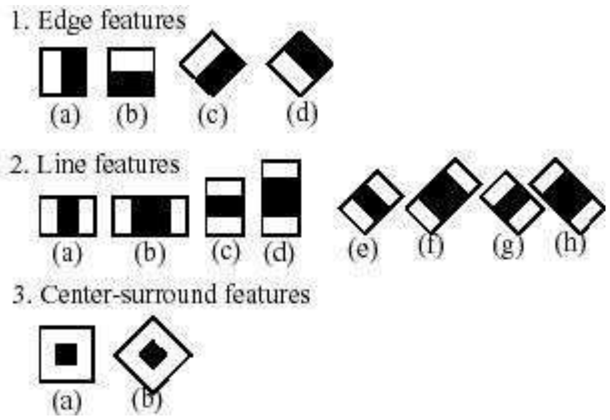



Fig. 2. Haar Cascade Features

The edges can be located vertically, horizontally, or diagonally using the first set of edge attributes. To ascertain whether a lighter area is surrounded by darker areas or vice versa, the second set of attributes is employed. The pixel intensity change across the diagonals is located using the third set of four rectangle features. The total of the white rectangle pixels less the sum of the black rectangle pixels is the feature value.

A more effective feature template to identify the object (fire) is produced when the values of the features from positive and negative photos differ significantly. An example of a 24x24 window generates 180,000 features.

*Stage 2*:

A method for rapidly calculating the sum of the pixel values in a rectangular area of an image is called an integral picture. It minimizes duplicate computations in recurring regions and expedites the feature extraction process. The total of all the pixels to its left and top is stored in each pixel.

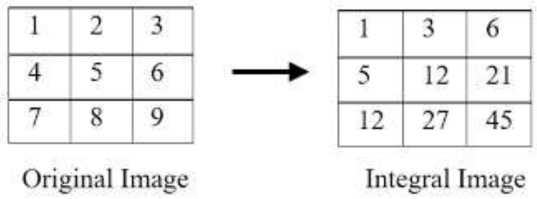


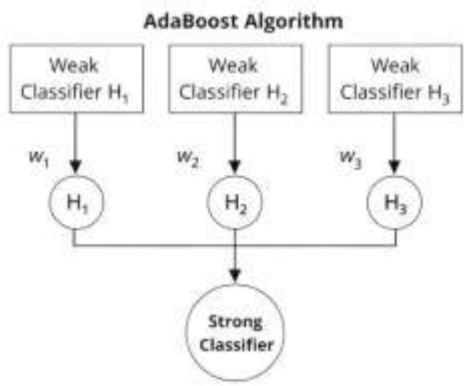

Fig. 3. Integral Image Representation

*Stage 3*:

The AdaBoost algorithm is used to build a strong classifier that can effectively distinguish fire from non-fire objects. It simplifies feature selection by identifying the most important characteristics needed for accurate classification. AdaBoost works by assigning weights to both the training data and the classifiers, allowing the model to focus more on difficult-to-classify samples. This improves overall detection performance. Instead of relying on a single decision rule, AdaBoost combines multiple simple classifiers, known as weak classifiers, to form a powerful strong classifier. During training, this process significantly reduces the number of features required. For example, in a 24×24 detection window, the number of features can be reduced from 180,000 to around 6,000. These selected features are then used to analyze training images and determine whether fire is present. Weak classifiers are generated in each training stage and refined through multiple iterations until either a maximum number of iterations is reached or the error rate becomes sufficiently low.

The cascade classifier operates in multiple stages, where each stage applies increasingly strict checks. As the process progresses, it becomes harder for a candidate region to pass through all stages. A region is classified as fire only if it successfully passes every stage; otherwise, it is rejected at the stage where it fails.

Fig. 4. HAAR Cascade Classifier




Training is carried out using both positive images (containing fire) and negative images (without fire), all standardized to the same size. Once trained, the classifier scans new images and labels regions that resemble fire with "1" and non-fire regions with "0", based on the features learned during training.

The classifier dynamically adjusts the size of the scanning window to enhance both detection speed and accuracy. This adaptive approach allows the algorithm to concentrate only on the most relevant regions of the image, thereby reducing unnecessary computation and improving efficiency.

During the classification process, the model also identifies optimal rectangular regions that closely match the object features within the window. This helps in refining the detection process and improves the overall reliability of the system.

*B. OpenCV*

OpenCV is an open-source computer vision library developed using C and C++. It is compatible with major operating systems such as Linux, Windows, and macOS. The library is designed for high computational efficiency, making it well-suited for real-time applications. OpenCV is widely used by developers and researchers working in the field of computer vision. Computer vision refers to the process of analyzing visual data from images or videos in order to extract meaningful information and make decisions based on that data.

*C. VNC Viewer*

Virtual Network Computing (VNC) is a client-server system that uses a simple and platform-independent display protocol. It was originally developed at the Olivetti Research Laboratory in Cambridge, England.

VNC enables remote access without the need for specialized hardware, making mobile computing more flexible and convenient. It allows a user to view and control one computer's desktop from another system using a simple software interface.

## VIII. EXPERIMENTAL ANALYSIS AND RESULTS

The experiments were conducted on a system equipped with a 13th Gen Intel® Core™ i5-13420H processor operating at 2.10 GHz, with 16 GB RAM (15.7 GB usable), running a 64-bit operating system on an x64-based architecture.

Real-time video frames were captured using the built-in webcam of the laptop. Each captured frame was converted into grayscale format and processed using the Haar Cascade classifier trained for fire detection.

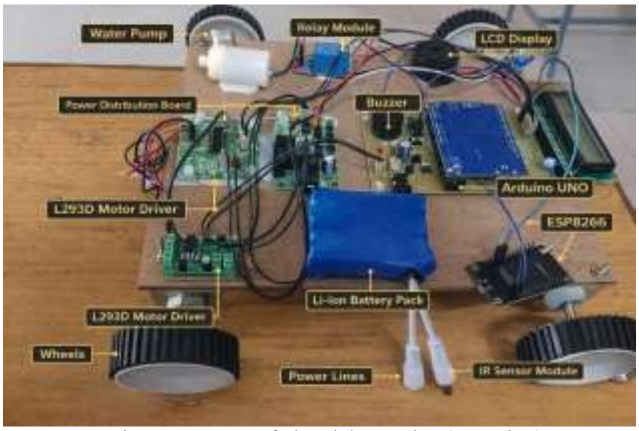



Fig. 5. Prototype of Fire Fighter Robot (Top View)

***Graphical User Interface (GUI):***

Fig. 6. Fire Detection using haar cascade

The proposed system includes a real-time monitoring dashboard called Fireguard AI for fire detection and alert management. The GUI displays live video streaming from the camera, where detected fire regions are highlighted using bounding boxes along with confidence scores. The status panel dynamically updates between "NORMAL" and "FIRE DETECTED", providing immediate visual alerts and evacuation warnings. Upon detection, the system automatically captures an image, records a short video, and sends alerts via Telegram. All system activities are recorded in the log section with timestamps.
Additionally, a manual control panel enables directional movement of the robotic platform for emergency response.

TABLE I. COMPARISON OF ACCURACY IN DIM LIGHT ROOM AND FULLY LIGHT ROOM OF THE PROTOTYPE

| Distance (cm) | Accuracy in Fully Lit Room (%) | Accuracy in Dim Lit Room (%) |
|---|---|---|
| 8 | 98 | 100 |
| 13 | 96 | 100 |
| 18 | 95 | 100 |
| 23 | 93 | 100 |
| 28 | 90 | 100 |
| 33 | 88 | 100 |
| 38 | 85 | 100 |
| 33 | 82 | 100 |
| 48 | 78 | 100 |
| 53 | 75 | 100 |
| 58 | 72 | 100 |
| 64 | 68 | 100 |
| 69 | 64 | 100 |
| 76 | 60 | 100 |
| 80 | 57 | 100 |
| 85 | 55 | 100 |
| 90 | 52 | 100 |
| 95 | 50 | 100 |
| 100 | 48 | 100 |

The table.1 presents a comparative evaluation of detection accuracy in both dim-light and fully illuminated environments for a 2.5 cm (1 inch) flame. Five trials were conducted at each distance, and the average accuracy was computed. This system accuracy is determined by using:

$$Accuracy = (TP+TN+FP+FN)/(TP+TN)$$

where TP, TN, FP, and FN denote True Positive, True Negative, False Positive, and False Negative, respectively. The results show that the system achieves 100% accuracy in dim-light conditions across the entire tested range (8 cm to 100 cm). In fully lit conditions, accuracy decreases progressively with distance. The system records 98% at 8 cm, maintains values above 85% up to 38 cm, reduces to 78% at 48 cm, and gradually declines to 48% at 100 cm.
The recall (sensitivity) of the system is 88.37% up to 80 cm, calculated as:

$$Recall=(TP+FN)/TP$$

The precision (Positive Predictive Value – PPV) is 69.09%, calculated as:

$$Precision=(TP+FP)/TP$$

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Conference Proceedings: Volume: 05 ACME – 2026 DOI: 10.55041/ISJEM.ACME117
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

The prototype demonstrates effective flame detection up to 100 cm, with optimal performance observed under low-light conditions. Compared to conventional flame sensors—which detect a 2.5 cm flame only up to 5 cm—the proposed system offers significantly improved detection range and operational efficiency.

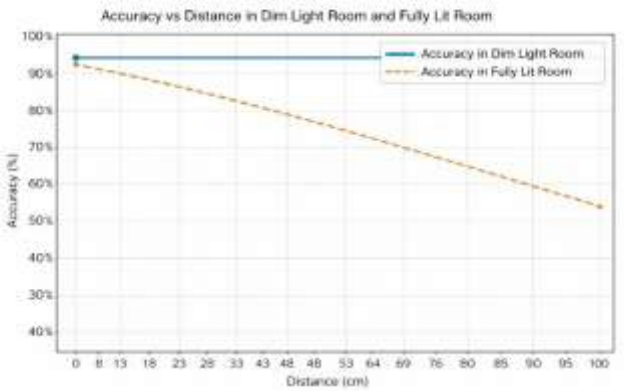



Fig. 7. Accuracy vs Distance graph for dim and fully lit rooms

## IX. DISCUSSION

In the studies titled "Firefighting Robot Based on IoT and Ban Levels Technique" and "Fire Detection and Direction Control of Fire Fighting Robot," the proposed systems primarily rely on infrared flame sensors and smoke sensors for fire detection. However, these sensors typically operate within a limited range and require close proximity to heat or smoke for accurate detection.

In contrast, the proposed prototype in this work enables early fire detection from a distance without the need for direct exposure to heat or smoke. This enhances both safety and response time. Furthermore, while the referenced studies recommend incorporating live camera streaming as a future enhancement, the present system already integrates real-time video monitoring, thereby improving functionality and making the prototype more advanced and efficient.

## X. CONCLUSION

An IoT-based automated firefighting robot has been developed using the HAAR Cascade algorithm for fire detection. The integration of image processing techniques with OpenCV improves detection accuracy. Unlike conventional systems that rely on flame or smoke sensors, the proposed robot can identify fire from a greater distance. A live camera feed is provided through Wi-Fi, offering better range and faster data transmission compared to Bluetooth. This enables early fire detection and automatic extinguishing, helping to reduce risks to firefighters.

## XI`. FUTURE RESEARCH

The performance of the prototype varies based on camera quality and environmental conditions, indicating scope for further improvement. The accuracy and response time can be enhanced by training the cascade classifier with a larger dataset. Additionally, using real-time camera frames for continuous training can help the system adapt to changing environments and improve detection reliability. The system can also be upgraded by integrating different types of fire extinguishers, such as chemical, wet chemical, carbon dioxide, foam, and dry powder, to handle various fire sources effectively.

## XII. REFERENCES

[1] Kiran, PhD, Keerthana Krishnan, Meghana M., Nikhita Mallasure, Sindhu S., and Sunil Kumar D.S., PhD, Fire Detection and Direction Control of Fire Fighting Robot, International Journal of Computer Applications, vol. 184, no. 26, pp. 21, Aug. 2022.

[2] M. Sivakumar, P. Kanakarajan, S. Dharun, R. Kirubakaran, and M. Girivasan, Development of an Artificial Intelligent Firefighting Robot and Experiment Investigation on Fire Scene Patrol, E3S Web of Conferences, vol. 547, pp. 02010, 2024.

[3] A. Meshram, O. Lad, S. Dhanawade, N. Shinde, and S. Khodke, Implementation of Fire Fighting Robot which can be controlled by various means like Android application, Remote and Voice inputs, Journal of Emerging Technologies and Innovative Research (JETIR), vol. 6, no. 5, pp. 408, May 2019.

[4] A. A. A. Rahman, M. Masrie, Z. Janin, T. S. Gunawan, R. Sam, and F.
D. A. Rahman, Firefighting Robot Based on IoT and Ban Levels Technique, in Proceedings of the 2022 IEEE 8th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), Shah Alam, Malaysia, 2022, pp. 1–6, doi: 10.1109/ICSIMA55652.2022.9929114.

[5] D. Gowri, S. Shiva Prasad, O. Sai Kiran, and M. Mysaiah, Image Processing Based Fire Detection by Using Raspberry Pi, Journal of Engineering Sciences, vol. 14, no. 6, pp. 1–6, 2023.

[6] G. P., T. T., R. M., J. Santhiya, and V. A. R., Object Detection Using HAAR Cascade Machine Learning Algorithm, International Journal of Creative Research Thoughts (IJCRT), vol. 10, no. 6, pp. 1–10, June 2022, ISSN: 2320-2882.

[7] Viola P, Jones M. Robust real-time face detection [J], International Journal of Computer Visio 57( 2):137-154.

[8] N. S. Pannu, S. Muradkar, and A. Chaurasia, Image Processing Based Fire Detection System, Journal of Engineering Sciences, vol. 1, no. 1,
pp. 1–6, 2023, ISSN: 2581-4419.

[9] Voila P, Jones M J. Rapid object detection using a boosted, cascade of simple features [C], Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (ICCVPR 2001), vol. 1, pp. I-511-I-518, 8-14 Dec. 2001, Kauai, USA.

[10] Shahad Salh Ali, Jamila Harbi Al' Ameri, and Thekra Abbas, Face Detection Using HAAR Cascade Algorithm, 2022 Fifth College of Science International Conference on Recent Trends in Information Technology (CSCTIT), Baghdad, Iraq, 2022, pp. 198–201, doi: 10.1109/CSCTIT56496.2022.9