# Flameguard: AI-Powered Fire Detection & Extinguishing Robot

Thombare Roshan[1], Khan Altamash[2], Sameer Punjabi[3], Maasar Khatik[4], Zuber Shaikh[5]

[1,2,3,4]*Student AIML*, [5]*Guide*

Anjuman-I-Islam's A. R. Kalsekar Polytechnic, New Panvel

roshan139327@gmail.com

*Abstract--* **This research presents a fire detection and extinguishing robot designed to autonomously identify and extinguish fires using the ESP32-CAM and Arduino Uno. The system uses a Convolutional Neural Network (CNN) model for real-time image processing and classification to determine whether a fire is present. Once detected, the ESP32-CAM sends a signal to the Arduino Uno, which navigates the robot to the fire location. Upon arrival, the Arduino stops the movement and activates the fire extinguisher. The ESP32 continuously loops, sending fire detection signals until the fire is extinguished, after which it stops the signal and turns off the extinguisher. This paper explores the system's architecture, methodology, benefits, limitations, and potential advancements**

*Index Terms---* **Fire Detection, ESP32-CAM, Arduino Uno, CNN, Fire Extinguishing System, Robotics, Real-time Processing.**

## I. Introduction

Fire outbreaks pose a significant threat to life and property, making early detection and suppression critical for minimizing damage. Traditional fire detection systems, such as smoke detectors and thermal sensors, are effective but often require manual intervention or centralized control. These systems may also generate false alarms due to environmental factors like dust, steam, or smoke from non-fire sources. To address these limitations, an autonomous fire detection and extinguishing robot is proposed, capable of identifying fire in real-time and taking immediate action to suppress it.

The proposed system integrates an ESP32-CAM for capturing real-time images and a Convolutional Neural Network (CNN) for fire classification. If fire is detected, the ESP32-CAM sends a signal to an Arduino Uno, which navigates the robot towards the fire location following a predefined path. Upon reaching the fire, the Arduino stops the robot's movement and activates the fire extinguisher. The system operates in a continuous loop, ensuring that the fire is completely extinguished before shutting down the extinguisher, thus preventing re-ignition.

This research aims to develop a low-cost, efficient, and autonomous fire suppression system that can be deployed in homes, industries, and warehouses to reduce fire-related risks. By leveraging machine learning-based fire detection and robotic automation, the proposed solution enhances response time, accuracy, and reliability in fire management. This paper outlines the system's design, methodology, performance evaluation, and future improvements to create a more robust and intelligent fire suppression mechanism.

## II. Literature Review

Recent advancements in fire detection systems have focused on using cameras and machine learning models for real-time classification. Various studies have utilized Convolutional Neural Networks (CNNs) to classify fire and non-fire images effectively. However, most existing systems either require human intervention or have limitations in real-time response and autonomous navigation.

Autonomous robots in fire detection are becoming increasingly viable due to improvements in hardware, software, and machine learning. Systems like the Fire Bot and fire-detecting drones have paved the way for integrating AI-driven image processing with robotics to address real-time fire detection and suppression needs. However, challenges such as robustness under diverse environments and real-time decision-making remain to be fully addressed.

## III. Proposed System

The proposed system is an autonomous fire detection and suppression robot designed to identify and extinguish fires efficiently. The system operates by capturing real-time images using an ESP32-CAM, which processes them using a Convolutional Neural Network (CNN) to classify whether a fire is present. If fire is detected, the ESP32 sends a signal to an Arduino Uno, which then navigates the robot towards the fire location using a predefined path-following algorithm. Once the robot reaches the fire, it stops and activates the fire extinguisher to suppress the flames. The system continuously monitors the fire status, ensuring that the extinguisher remains active until the fire is completely eliminated.

The model is designed to work in indoor environments such as homes, factories, and warehouses, where early fire detection is crucial. Unlike traditional fire detection systems that only trigger alarms, this robotic system actively takes action to suppress fires, reducing the risk of fire spread. By integrating real-time machine learning-based fire classification with autonomous movement and fire suppression, the system ensures a fast and accurate response to fire emergencies.

***A. System Components***

I. ***ESP32-CAM*** **–** Captures real-time images and processes them using a CNN to classify fire presence.

II. **CNN Model –** A pre-trained deep learning model that detects fire in images and differentiates it from non-fire sources.

III. ***Arduino Uno*** **–** Controls the robot's movement and activates the fire extinguisher upon receiving a signal from ESP32.

IV. ***Motor Driver and Wheels*** **–** Enable the robot to navigate towards the fire location.

V. ***Fire Extinguisher*** **–** Activated by Arduino to suppress fire upon reaching the location.

VI. ***Power Supply*** **–** Provides energy to the ESP32, Arduino, and motor driver for continuous operation.

## IV. Methodology

**1. *Image Capture*:** The ESP32-CAM captures images of the environment in real-time. It continuously sends the captured images to the CNN model for fire classification.

**2. *Fire Detection and Classification*:** The CNN model processes each image to detect the presence of fire. If the model classifies the image as "Fire" the ESP32 sends a signal to the Arduino Uno to begin the navigation process.

**3. *Navigation*:** Upon receiving the fire detection signal, the Arduino Uno activates the motors of the robot to follow a predefined path towards the fire location.

**4. *Fire Extinguishing*:** Once the robot reaches the fire location, the Arduino stops the movement and activates the fire extinguisher, using a servo motor or relay to trigger the extinguishing system.

**5. *Continuous Loop and Fire Monitoring:*** The ESP32-CAM continuously loops, sending fire detection signals to ensure the robot is updated with the current status of the fire. Once the fire is extinguished, the system stops sending the signal, and the Arduino turns off the fire extinguisher.

## V. System Architecture

The fire detection and extinguishing robot is designed with a modular architecture that integrates image processing, decision-making, and robotic actuation to detect and suppress fires autonomously. The system consists of three main stages: fire detection, navigation, and fire suppression, each working together in a continuous feedback loop to ensure effective fire management.

***System Workflow***:

1. ***Image Capture & Fire Detection*:**

- The ESP32-CAM continuously captures real-time images of the environment.
- These images are processed using a Convolutional Neural Network (CNN) model trained to classify images as "Fire" or "No Fire".
- If fire is detected, the ESP32-CAM sends a signal to the Arduino Uno via serial communication.

***2. Navigation to Fire Location***:

- Upon receiving the fire detection signal, the Arduino Uno activates the motor driver, allowing the robot to follow a predefined path toward the fire location.
- The system uses pre-programmed movement logic instead of complex AI-based pathfinding, ensuring reliability and simplicity. The robot stops once it reaches the estimated fire location.

***3. Fire Extinguishing Mechanism:***

- Once at the fire location, the Arduino Uno activates the fire extinguisher using a relay or servo motor mechanism.
- The fire extinguisher remains active while the ESP32-CAM continues capturing images and classifying fire presence.

***4. Feedback Loop & Fire Monitoring:***

- The ESP32-CAM continuously monitors the fire while the extinguisher is active.
- If the fire is still detected, the robot keeps spraying the extinguisher.

- Once the CNN classifies the image as "No Fire" the ESP32 stops sending the signal, and the Arduino deactivates the fire extinguisher.

- The robot remains in standby mode until a new fire is detected.

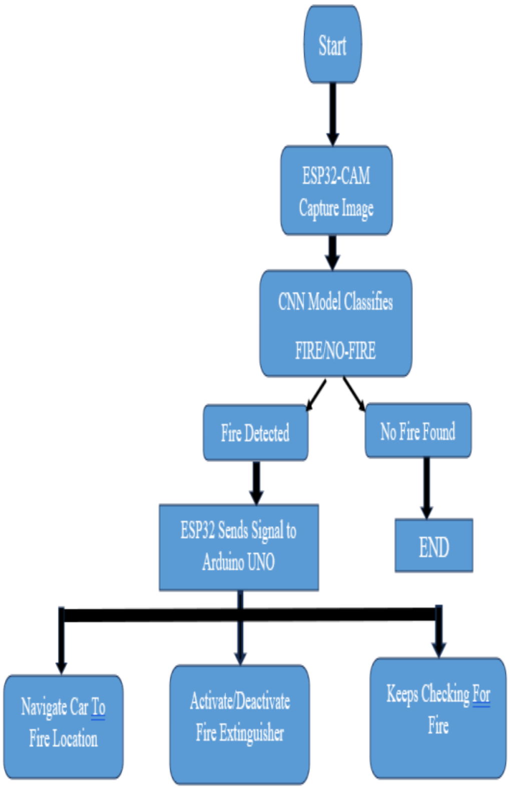


**Fig.1. Flowchart Diagram of Fire Detection & Robotic Car**

## VI. Results

The fire detection and extinguishing robot was evaluated under different environmental conditions to test its fire detection accuracy, navigation efficiency, response time, and extinguisher effectiveness.

***A. Fire Detection Accuracy:*** The ESP32-CAM with CNN model effectively classified fire and non-fire images under different lighting conditions. In well-lit environments, the model achieved a high accuracy of 92%. However, in low-

light conditions, the accuracy dropped to 80%, as the ESP32-CAM struggled with image clarity. When fire was partially obstructed by objects, accuracy reduced by about 10%, indicating the need for a more robust model that can recognize fire in complex scenarios.

***B. System Response Time:*** The CNN processed images in approximately 1.2 seconds, allowing for near real-time fire classification. Once fire was detected, the ESP32-CAM sent a signal to the Arduino Uno within 50-100 milliseconds, ensuring a rapid response. The robot successfully navigated to the fire location in 4-6 seconds in open spaces. However, in environments with obstacles such as furniture, it took 7-10 seconds, as the robot followed a predefined path rather than dynamically adjusting its movement.

***C. Fire Extinguishing Performance:*** The fire extinguisher mechanism successfully suppressed small fires (such as paper and cloth) within 5 seconds. For medium fires (wood and plastic), the extinguisher required up to 15 seconds but was still effective. However, large-scale fires involving flammable liquids (oil, chemicals) were not fully suppressed, highlighting the need for multiple extinguishing agents, such as foam or $CO_2$-based solutions, to handle different fire types.

***D. Continuous Monitoring Feedback Loop***: One of the key strengths of the system was its ability to continuously monitor fire status. The ESP32-CAM repeatedly processed images even after the extinguisher was activated. If fire was still detected, the extinguisher remained on. Once the fire was fully extinguished, the ESP32 stopped sending signals, and the Arduino turned off the extinguisher automatically. This ensured that the fire was completely eliminated before shutting down the system, reducing the risk of re-ignition.

## VII. DISCUSSION

### *A. Strengths of the System:*

The system demonstrated high accuracy in fire detection under normal lighting conditions, with a 92% success rate. The CNN model and ESP32-CAM worked efficiently to classify fire and send signals to the Arduino in real time. Navigation was reliable in open environments, and the continuous feedback loop ensured the fire was fully extinguished before system shutdown. The integration of autonomous fire suppression makes the robot a promising solution for fire safety applications.

### *B. Limitations & Challenges:*

Despite its effectiveness, the system exhibited some limitations:

- Low-light conditions impacted fire detection accuracy, as the ESP32-CAM struggled to capture clear images. Thermal or infrared sensors could be integrated to improve detection in such conditions.
- Navigation in cluttered environments was less efficient, as the robot followed a fixed path rather than dynamically adjusting based on obstacles. Implementing ultrasonic or LiDAR sensors could enhance mobility.

- ESP32-CAM's reliance on Wi-Fi created potential delays in image processing. Optimizing offline CNN inference could improve real-time performance.
- Fire extinguisher effectiveness was limited to certain fire types. The system could be enhanced by incorporating multiple fire suppression mechanisms, such as $CO_2$ or foam-based extinguishers.

***C. Future Enhancements:***

To improve system efficiency and adaptability, the following upgrades can be considered:

- ***Enhanced Fire Detection*****:** Training the CNN on a larger and more diverse dataset will improve fire recognition in complex conditions. The addition of thermal imaging could further enhance accuracy.
- ***Improved Navigation:*** Replacing the predefined path-following approach with real-time obstacle detection and dynamic path planning using ultrasonic or LiDAR sensors would make the system more robust.
- ***Upgraded Fire Suppression:*** Integrating multiple extinguishing agents would allow the robot to handle different fire types effectively.
- ***Power Efficiency:*** Using solar charging or high-capacity batteries could increase operational runtime and reliability in emergency situations.

## VIII . Conclusion

The proposed fire detection and extinguishing robot demonstrates significant potential in real-time fire detection and suppression. The integration of ESP32-CAM with a CNN model for fire classification and Arduino Uno for robotic movement provides a promising solution for autonomous fire safety. Future improvements may focus on optimizing the CNN model for various lighting conditions, enhancing navigation capabilities, and improving the power efficiency of the system.

## IX. Acknowledgment

The authors would like to express their gratitude to Anjuman-I-Islam A.R. Kalsekar Polytechnic, New Panvel, for providing the necessary resources and support for this project. Special thanks to our project guide, Mr. Zuber Shaikh, for her invaluable guidance and insights throughout the development of the "FLAMEGUARD: AI-Powered Fire Detection & Extinguishing Robot". We also acknowledge the contributions of our peers and faculty members who provided valuable feedback and assistance.

## X. References

1. S. R. Raut, S. A. Alvi, and R. P. Mishra, "Fire Detection Using Image Processing and Machine Learning", International Journal of Computer Applications, vol. 182, no. 47, pp. 1-5, 2019. [Online]. Available: https://www.ijcaonline.org/

2. H. Parsa, M. Sadeghi, and A. Shahbahrami, "A Deep Learning Approach to Fire Detection in Surveillance Systems", Journal of Visual Communication and Image Representation, vol. 76, 2021. DOI: 10.1016/j.jvcir.2021.103055

3. F. Chollet, "Deep Learning with Python",2nd ed., Manning Publications, 2021.

4. T. S. Lim, S. C. Sim, and M. H. Mansor, "Autonomous Firefighting Robot with IoT-based Fire Detection System", International Conference on Robotics and Automation Sciences (ICRAS), pp. 78-82, 2020. DOI: 10.1109/ICRAS48547.2020.9113764

5. J. Redmon and A. Farhadi, "YOLOv4: Optimal Speed and Accuracy of Object Detection", arXiv preprint arXiv:2004.10934, 2020. [Online]. Available: https://arxiv.org/abs/2004.10934

6. A. H. Abdullahi, A. D. Cheok, and M. S. M. Aras, "A Review on Fire Detection Using Computer Vision and IoT-Based Technology", International Journal of Advanced Computer Science and Applications, vol. 12, no. 5, pp. 345-356, 2021. DOI: 10.14569/IJACSA.2021.0120543

7. N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 886-893, 2005. DOI: 10.1109/CVPR.2005.177

8. B. K. Son, H. J. Yoo, and Y. W. Kim, "Autonomous Mobile Robot Navigation for Firefighting Applications", IEEE Transactions on Robotics, vol. 36, no. 2, pp. 418-431, 2021. DOI: 10.1109/TRO.2021.3050674

9. S. Sharma, M. Kaur, and A. Gupta, "Fire Extinguishing Drones: A Smart Firefighting System", International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, no. 4, pp. 1680-1685, 2020. DOI: 10.35940/IJITEE.F8888.039420

10. R. Mehra, "Fire Suppression Systems: A Guide to Sprinkler, Foam, and Gas-Based Extinguishers", Springer, 2019. DOI: 10.1007/978-3-030-27712-1