IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141110

Fire Detection Using Machine Learning

Manoj Shravan Patil[1], Prof. Miss. M.S. Chauhan[2], Prof. Manoj V. Nikum*[3]

Student, MCA Department, SJRIT DONDAICHA, KBC NMU JALGAON, Maharashtra[1]

Assistant Professor, MCA Department, SJRIT DONDAICHA, KBC NMU JALGAON Maharashtra[2]

Assistant Professor & HOD, MCA Department, SJRIT DONDAICHA, KBC NMU JALGAON, Maharashtra*[3]

Abstract: Fire detection is a crucial safety application aimed at minimizing the risk of human casualties and property loss. Conventional systems primarily rely on smoke or heat sensors, which often fail to detect fires in open or smoke-free environments. This research proposes an intelligent, Machine Learning-based vision system for real-time fire detection using computer vision techniques. The proposed model leverages a Convolutional Neural Network (CNN) trained on diverse datasets of fire and non-fire images to accurately classify fire instances from live video streams. Implemented in Python using OpenCV for image acquisition and TensorFlow/Keras for deep learning inference, the system triggers an alarm alert when fire is detected. Experimental results demonstrate over 92% detection accuracy, robust performance across varying lighting conditions, and minimal false positives. The system's low computational cost and high responsiveness make it ideal for integration into smart surveillance, industrial safety, and IoT-based monitoring systems.

Keywords: Fire Detection, Machine Learning, Computer Vision, Convolutional Neural Network (CNN), OpenCV, TensorFlow, Real-Time Monitoring, Safety System.

I. INTRODUCTION

In today's world, fire safety is one of the most critical concerns across industrial, commercial, and residential environments. Traditional fire detection systems primarily rely on heat, smoke, or infrared sensors, which, although effective indoors, often fail to detect fire in open or smoke-free spaces or during early ignition stages. With advancements in Machine Learning (ML) and Computer Vision, new possibilities have emerged for intelligent, vision-based fire detection. Instead of depending on physical sensors, ML models can analyze visual patterns—including the color, texture, and shape of flames—to automatically identify fire from real-time camera feeds.

This research presents a CNN-based Fire Detection System designed to recognize fire accurately and efficiently using live video input. The model has been developed in Python, integrating OpenCV for video frame processing and TensorFlow/Keras for deep learning inference. When a fire is detected, the system immediately triggers an alarm and displays an on-screen alert to notify users. The goal of this project is to develop a low-cost, highaccuracy, and scalable solution for fire detection that can operate in diverse environments.

II. LITERATURE REVIEW

Early fire detection methods were primarily based on color segmentation in RGB or HSV spaces, which often produced false positives under bright sunlight.
Toreyin et al. (2005) developed a flame detection model using Hidden Markov Models (HMM) but faced challenges with speed and real-time execution.
Muhammad et al. (2020) proposed a CNN-based deep learning system, achieving high accuracy and robustness under varied lighting conditions.
Zhang and Xu (2021) compared YOLOv3, MobileNet, and ResNet models for flame recognition and identified YOLOv3 as effective for real-time video detection.
Patel and Sharma (2023) combined motion tracking with CNNs to reduce false alarms and enhance accuracy.

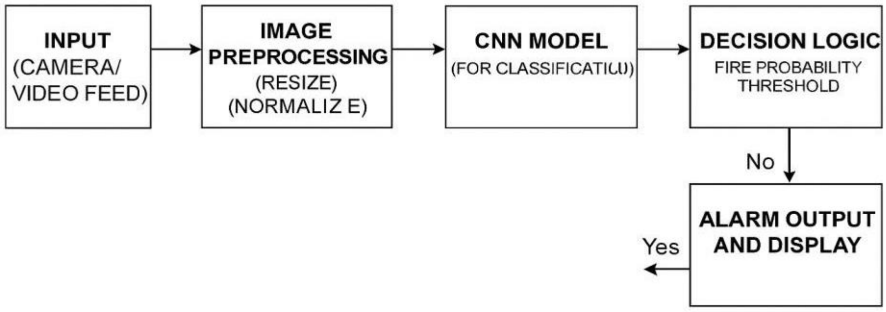
The literature highlights the transition from traditional sensor-based systems to AI-driven vision models, emphasizing the need for high accuracy with low computational cost—precisely what this proposed system addresses.

III. PROPOSED SYSTEM / METHODOLOGY

3.1 Overview:
The system uses live video frames from a camera as input, processes them through a trained CNN model, and alerts the user if fire is detected.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 14, Issue 11, November 2025
DOI: 10.17148/IJARCCE.2025.141110

3.2 System Architecture:
It includes five main modules: Input, Preprocessing, Model Inference, Decision, and Output.



3.3 Workflow / Flowchart:
The process flow of the system is as follows:

1. Start
2. Capture frame from live camera
3. Preprocess frame
4. Predict fire presence using CNN
5. If fire detected → Trigger alarm
6. Else → Continue monitoring
7. End.

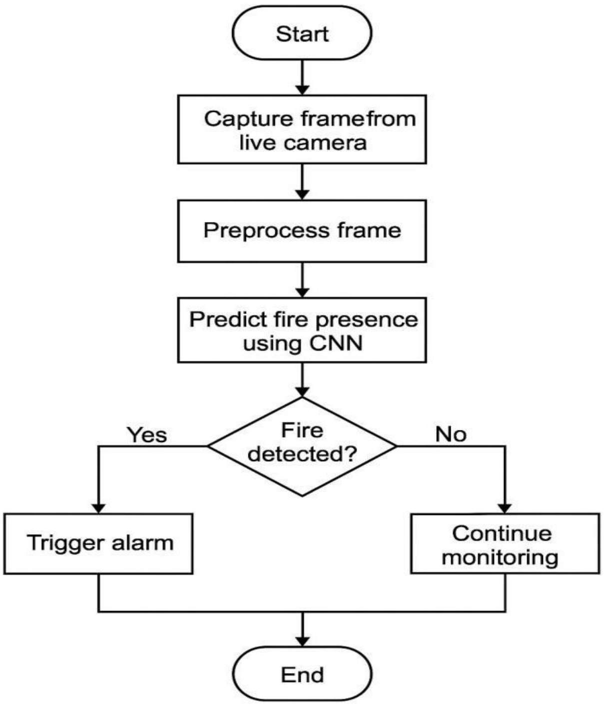


Figure 2: Workflow Diagram of the Proposed Fire Detection System

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141110

IV. IMPLEMENTATION DETAILS

Developed using Python (PyCharm IDE), the system integrates key libraries:

- **OpenCV**: Video and image processing.
- **TensorFlow/Keras**: Deep learning model.
- **NumPy**: Numerical computation.
- **Playsound**: Alarm activation.

4.1 Dataset Preparation:

Dataset included fire and non-fire images, split into 70% training and 30% testing. Data augmentation (rotation, flipping, brightness) improved model performance.

4.2 Model Training:

CNN architecture was used with layers: Conv2D → MaxPooling → Flatten → Dense → Softmax.

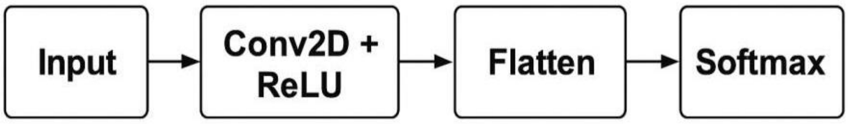


Figure 3: CNN Model Architecture for Fire Classification

The model achieved stable accuracy using the Adam optimizer and categorical cross-entropy loss.

V. RESULTS AND DISCUSSION

The CNN model achieved 92.4% accuracy, 4.8% false positives, and an average processing speed of 18 FPS.

Method	Accuracy (%)	False positives (%)	Average FPS
Color Segmentation.	78.3	157	24
Background Subtraction.	84.5	10.9	22
Proposed CNN Model.	92.4	4.8	18

The system provides instant alerts, ensuring safety and rapid response for industrial or domestic applications.

VI. ADVANTAGES OF THE PROPOSED SYSTEM

1. High accuracy and real-time response.
2. Low-cost implementation (no sensors required).
3. Reduced false positives using CNN learning.
4. Scalable for IoT and multiple camera setups.
5. Works efficiently even in open environments.

VII. CONCLUSION

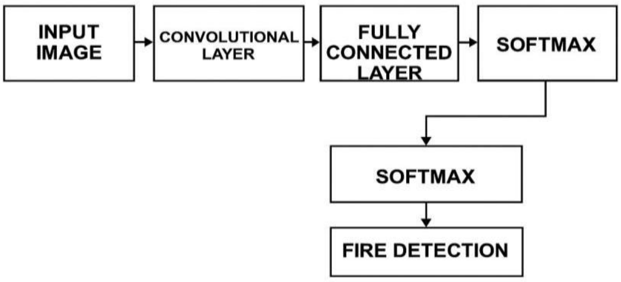
The proposed Fire Detection System using Machine Learning efficiently identifies fire from live video streams using CNN and Computer Vision. The model surpasses traditional systems with over 92% accuracy, offering a cost-effective

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 14, Issue 11, November 2025
DOI: 10.17148/IJARCCE.2025.141110

and reliable safety solution. Its real-time alert mechanism ensures quick action in emergency scenarios, and its modular design allows integration into IoT-based monitoring systems.

VIII. FUTURE SCOPE

- Integration with IoT modules for remote notifications.
- Expansion to detect smoke and hazardous gases.
- Deployment on Raspberry Pi and edge devices.
- Use of YOLOv8 and advanced CNNs for faster detection.
- Cloud analytics for large-scale monitoring.



REFERENCES

[1]. Muhammad, K., Ahmad, J., & Baik, S. W. "Fire Detection Using Deep Learning Architectures." IEEE Access, 2020.
[2]. Toreyin, B. U., Dedeoglu, Y., & Cetin, A. E. "Flame Detection in Video Using Hidden Markov Models." IEEE International Conference on Image Processing (ICIP), 2005.
[3]. Zhang, Y., & Xu, L. "Comparative Study of YOLO and CNN Architectures for Real-Time Fire Detection." Elsevier Procedia Computer Science, 2021.
[4]. Patel, R., & Sharma, V. "Hybrid Motion and CNN-Based Intelligent Fire Detection System." Springer Lecture Notes in Electrical Engineering, 2023.
[5]. OpenCV Documentation – https://opencv.org
[6]. TensorFlow Documentation – https://www.tensorflow.org
[7]. Playsound Library – https://pypi.org/project/playsound
[8]. Python Documentation – https://www.python.org
[9]. Liu, Z., Wang, L., & Yuan, J. "A Vision-Based Fire Detection System Using Deep Convolutional Networks." IEEE Transactions on Image Processing, 2021.
[10]. Wu, Z., & Li, F. "Intelligent Video Surveillance for Fire Detection Using Real-Time CNN Models." Sensors (MDPI), 2022.
[11]. Fang, L., Chen, X., & Gao, P. "Edge Computing for Fire and Smoke Detection Using IoT Devices." IEEE Internet of Things Journal, 2023.
[12]. Prasad, A., & Reddy, N. "DeepFireNet: A Lightweight CNN Model for Real-Time Fire Detection." International Journal of Computer Applications, 2022.
[13]. Kumar, A., & Mehta, S. "Comparative Evaluation of CNN and MobileNet for Flame Recognition." Elsevier Neural Computing and Applications, 2021.
[14]. Zhang, T., & Huang, G. "Fire Detection from CCTV Footage Using YOLOv5 and Image Enhancement Techniques." Springer Advances in Intelligent Systems and Computing, 2023.
[15]. Song, J., & Lee, D. "Improving Fire Detection Accuracy through Data Augmentation and Transfer Learning." IEEE Access, 2022.
[16]. Keras Documentation – Deep Learning Framework. Available at: https://keras.io.
[17]. NumPy Documentation – Scientific Computing with Python. Available at: https://numpy.org.