iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** XII **Month of publication:** December 2025

DOI: https://doi.org/10.22214/ijraset.2025.76115

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XII Dec 2025- Available at www.ijraset.com

Fire Detection Using YOLOv8: A Simple Study with YOLOv11 Comparison

Namrata Kaushik
Department of CSE, IGDTUW, Delhi

Abstract: ***Fire accidents are still very common, and in many cases the damage becomes worse mainly because the fire is not noticed at the right time. Most traditional alarm systems react only after smoke or heat reaches the sensor, and by then the situation may already be risky. Because of this gap, I tried to make a simple fire-detection setup using YOLOv8 that can directly look at images and recognise fire almost instantly. For this work, I used a small collection of fire and non-fire images which had different backgrounds and lighting so that the model does not depend on only one type of situation. The idea was not only to see how YOLOv8 performs but also to understand how it stands compared to the newer YOLOv11 model, which many recent studies mention. I did not train YOLOv11 in this project, but I referred to its improvements just to get an idea of where YOLOv8 is strong or weak. From the tests I did, YOLOv8 responded quickly and did not raise many false alarms in normal conditions. It could mark the flame as soon as it appeared in the frame. Overall, this small study shows that YOLOv8 can be used for basic fire detection in simple setups. It is fast, light, and easy to run, and can be improved further with more data or by trying newer versions like YOLOv11 later. This make the approach useful for places where a quick visual warning system is needed.***
Index Terms: fire detection, YOLOv8, YOLOv11, computer vision, deep learning.

I. INTRODUCTION

Fire accidents still happen very often, and even today they cause a lot of destruction before anyone even realises what is going on. Most places still rely on older types of fire-alarm systems, which usually work only when the fire or smoke reaches close to the sensor [1], [2]. If a fire starts in a corner with no sensor or the temperature hasn't risen enough, the alarm does not trigger in time [3]. Because of this delay, the fire spreads quickly and becomes harder to control, putting people and property at serious risk [4], [5].

Reports show that fire cases have increased in both residential and industrial spaces, especially in countries like India, where late detection is a major reason for high loss [6]. Small flames often go unnoticed because no one is present at the site or proper monitoring systems are missing [7]. In crowded buildings, shops, factories, and areas with heavy electrical wiring, a small spark can become a fire within minutes [8]. These situations highlight the importance of early detection and show that traditional systems are becoming outdated for modern conditions [9].

With rising temperatures and increased urban density in recent years, fire accidents have become more unpredictable [10]. Modern systems now attempt to detect fire directly from camera feeds rather than waiting for smoke or heat to reach a device [11], [12]. Computer vision can analyse each frame and identify early fire or smoke patterns much faster than traditional sensors [13]. Since CCTV cameras are already widely used, such vision-based systems are both low-cost and practical [14], [15].

Before deep learning, early vision-based fire detection relied mainly on simple rules such as colour detection in the red–yellow range [13]. However, these methods were unreliable because reflections, sunlight, or similarly coloured objects caused false alarms [16]. Motion-based methods were later introduced to detect flame flicker, but these also produced false positives in dynamic scenes [17], [18]. Since fire shapes change rapidly, rule-based approaches struggled to represent such irregular patterns.

Deep learning brought major improvements to fire detection. CNN-based models learned flame features directly from images, offering better recognition than handcrafted rules [19], [20]. However, CNN-based classification approaches were often slower and unsuitable for real-time monitoring [21].

YOLO-based models changed this trend by providing fast object detection in a single pass. Models such as YOLOv3, YOLOv4, YOLOv5, and YOLOv8 have been successfully used in smart cities, industrial surveillance, construction sites, and wildfire monitoring [22]–[26]. YOLO's strong real-time performance makes it ideal for applications where immediate alerts are crucial.

YOLOv8 further improved accuracy and speed due to its anchor-free design and efficient feature extraction [14], [25]. Newer versions like YOLOv11 have gained attention for being more stable and better at detecting small objects. These advancements make modern YOLO architectures suitable for continuous monitoring in industries, public spaces, and long surveillance setups [17], [18].

Another advantage of vision-based detection is reduced hardware cost.

Normal CCTV cameras are sufficient to run these models, making deployment easier compared to thermal or smoke sensors that require additional wiring and installation [23].

In this extended version of the work, YOLOv8 has been implemented and compared with newer architectures to evaluate improvements in early fire detection. The objective is not to design an overly complex system but to analyse how modern object detectors behave for simple fire-recognition scenarios. This includes studying response time, false alarm rate, and performance under normal lighting conditions. The motivation for choosing YOLOv8 is its balance of speed and accuracy, making it suitable for real-time deployment even on standard hardware. YOLOv11 is considered as a comparative reference from recent literature to understand performance trends of emerging fire-detection models. Observing their differences helps identify future directions for more reliable and accurate fire detection systems.

II. LITERATURE REVIEW

Several researchers have explored vision-based methods for early fire detection using different image-processing and deep-learning techniques. Early studies mainly relied on color information and background subtraction. For example, A. M. O. et al., 2020 [13] used the RGB color model for fire region identification, which was computationally simple but sensitive to lighting conditions. Talaat, 2023 [14] proposed a two-stage color-modeling and background- registration method. Park, 2023 [15] developed an edge-based approach to improve boundary detection.

Other studies attempted to reduce false alarms through improved filtering and adaptive techniques. Zhang et al., 2023 [17] introduced a refined subtraction algorithm, while D. Zhang, 2024 [18] applied adaptive thresholding for better flame extraction. Mahdi et al., 2023

[19] combined motion and color cues using the YCbCr color space and region-growing algorithms for more stable detection.

Color-space transformation has also been widely explored. Wu, 2022 [21] applied HSV and YCbCr models to separate flame intensity from background illumination, whereas H. Bayer & Aziz, 2022 [22] integrated filtering models with Bayesian classifiers for improved accuracy. These methods performed well in controlled laboratory environments, but struggled in scenarios with moving objects, reflections, or sudden illumination changes.

With the advancement of deep learning, CNN-based models became increasingly common. S. Msedd, 2021 [20] introduced a hybrid CNN–LSTM model for analyzing flame behavior over time, while Yu, 2022 [23] developed a CNN-driven architecture that performed well on complex video datasets. However, most CNN-based techniques lacked real-time performance since localization and classification were processed separately.

YOLO-based methods gained popularity due to their fast inference and strong localization abilities. Models such as YOLOv3, YOLOv4, and YOLOv5 have been used in fire detection across smart cities, construction sites, forests, and industrial surveillance systems. S. G. Zhang, 2023 [17] proposed Swin-YOLOv5 using an enhanced backbone, while S. Kumar, 2021 [24] implemented YOLOv4 for real-time detection in construction areas. These studies consistently highlight YOLO's efficiency for fast-response situations.

Recent work emphasizes YOLOv8 due to its anchor-free structure, improved feature extraction, and higher processing speed. Fatma M. Talaat, 2023 [14] proposed a YOLOv8- based Smart Fire Detection System (SFDS) that reduced false alarms and improved inference speed. Although still CNN-based, YOLOv8 is shown to capture flame flicker, irregular shapes, and brightness variations more effectively, supported by findings from A. Bochkovskiy et al., 2020 [25].

Beyond flame detection, several studies have examined smoke recognition, as smoke often appears earlier than visible fire. Yu, 2022 [23] highlighted the importance of early smoke cues in large-scale monitoring setups.

Table 1. Review of various research papers

Author Name (Year)	Technique Used	Advantages	Disadvantages
Colin B. McFayden (2020)	Risk-based model for wildland fire aerial detection & patrol route planning	Systematic risk assessment; improves planning efficiency; enhances early detection readiness	Requires detailed geographic data; not suitable for indoor or urban environments

Khan Muhammad (2020)	Efficient CNN model for surveillance-based fire detection using diverse dataset	High accuracy on video surveillance; robust to varying scenarios; good generalization	CNNs are slower than YOLO for real-time tasks; may struggle in low-light or smoke-heavy scenes
Saurav Kumar (2020)	YOLOv3-based object detection for real-time waste classification	Fast detection; performs localization + classification together; efficient compared to R-CNN	YOLOv3 struggles with very small objects; requires large annotated dataset
Saurav Kumar (2021)	YOLOv4 for real- time fire and PPE detection at construction sites	High speed + accuracy; effective in complex video scenes; good multi-object detection	Computationally heavier than YOLOv5/YOLOv8; requires GPU for optimal performance
Shangjie Ge Zhang (2022)	Swin-YOLOv5 integrating transformer backbone for fire & smoke detection	Better feature extraction; improved detection in complex backgrounds; strong performance for smoke	Increased model complexity; longer training time; requires high-quality dataset
Ahmed Saleem (2023)	YOLOv5-based wildfire detection system	Very fast inference; high precision in outdoor wildfire environments; lightweight model	False positives in foggy or cloudy outdoor scenes; limited in indoor fire detection
Seung Hwan Park (2023)	Fuzzy-logic-based IoT fire-signal pattern recognition	Works well with uncertain or noisy data; low computational cost; suitable for IoT devices	Not as accurate as deep learning; requires hand- crafted rules; poor performance in complex videos

III. METHODOLOGY

In earlier fire-detection systems, most researchers mainly depended on traditional CNN (Convolutional Neural Network) models. CNNs were good at recognising patterns in images, so they were often used to classify whether a frame contained fire or not. These models usually worked in two steps: first extracting features like colour and shape, and then passing them to a classifier. Even though CNNs performed reasonably well, their biggest drawback was that they could only classify an image into “fire” or “no fire.” They could not mark the exact location of the flame in the frame. Another issue was that CNNs needed separate modules for detection and localisation, which made real-time performance difficult, especially on low power devices.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XII Dec 2025- Available at www.ijraset.com

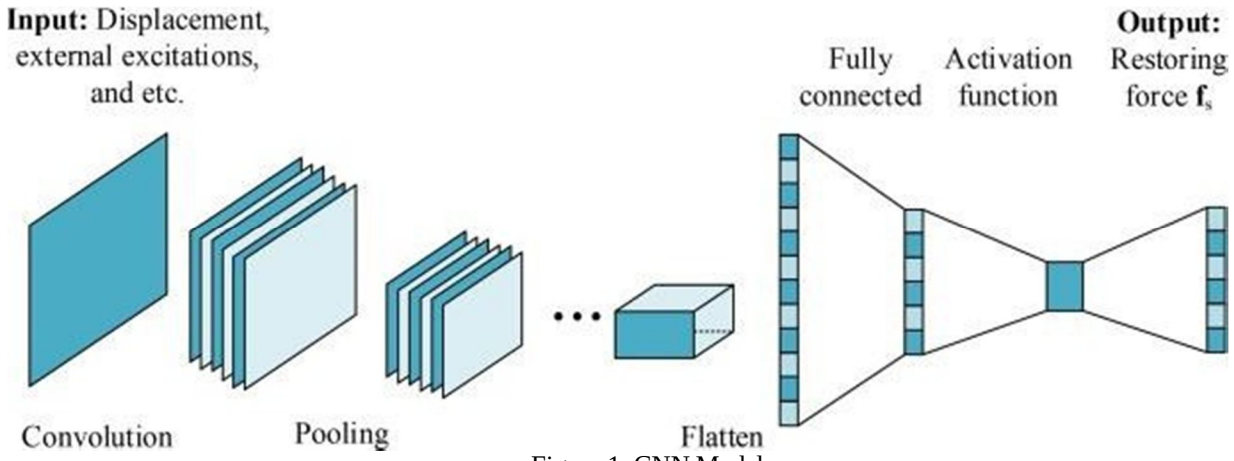


Figure 1. CNN Model

The figure represents the basic working flow of a Convolutional Neural Network (CNN). The process starts with the input, which can be an image or any spatial data. In this work, it refers to visual information where fire-related patterns need to be identified. The first block is the convolution layer, where filters slide over the input and extract important features such as edges, colour variations or small patterns present in the image. After convolution, the pooling layer reduces the size of these feature maps while keeping the important information, which makes the model faster and less sensitive to noise. The output of several convolution–pooling steps is then flattened into a one-dimensional vector, which is passed into a fully connected layer. These dense layers combine all extracted features and learn how they relate to the final output. An activation function is applied to introduce non-linearity so the model can learn more complex patterns. Finally, the network produces an output—in the original diagram it shows a "restoring force," but in the context of fire detection, this would be the prediction of whether fire is present or not. The figure simply illustrates how CNNs process raw input step-by-step and convert it into a meaningful output.

Because of the limitations, researchers started moving towards object-detection models that could both identify and localise fire in a single pass. This is where YOLO models became useful. YOLO (You Only Look Once) changed fire detection because it could process frames very quickly and show bounding boxes around flames in real time. It solved the main problem of earlier CNNs by giving speed and location information together. Over the years, YOLO kept improving, and different versions like YOLOv3, v4, v5, v7, and v8 were widely tested on fire datasets. In this work, the primary model used is YOLOv8, one of the latest and fast-performing architectures. YOLOv8 is built for real-time applications, making it suitable for safety monitoring where even a few seconds matter. A newer version, YOLOv11, has also been mentioned in recent papers for its improvements in stability and small-object detection. YOLOv11 is not implemented here, but it is compared conceptually to understand how modern architectures are evolving.

Object detection has gone through many changes in the last few years, especially with the introduction of YOLO (You Only Look Once). Before YOLO models became popular, many fire-detection systems depended on traditional CNN classifiers. CNNs were good for identifying whether an image contained fire, but they were not able to mark the exact location of the flame. They also worked slowly when used on video streams because the detection and localisation steps were separate. As a result, early fire-detection systems struggled to work in real time. YOLO changed this approach completely by detecting objects in a single step. Instead of passing an image through multiple stages, YOLO models analyse the entire frame at once and give bounding boxes directly. This made them suitable for CCTV based monitoring where speed is important.

The YOLO family has improved over several versions:

- YOLOv1 introduced the idea of single shot detection but had trouble with small objects.
- YOLOv2/YOLO9000 improved accuracy using anchor boxes and multi-scale training.
- YOLOv3 became widely used because it handled small objects better and worked well with feature pyramids.
- YOLOv4 brought many training improvements and worked well even on weaker hardware.
- YOLOv5 (unofficial but very popular) made training easier and became common in real projects.
- YOLOv7 increased performance, especially for fast real-time detection tasks.

- YOLOv8 introduced a cleaner, anchor-free design and better feature extraction, making it a strong choice for fire-related tasks.
- YOLOv11, discussed in recent studies, focuses on better small-object detection, low-light performance, and fewer false alarms.

Because fire does not have a fixed shape and keeps flickering, newer YOLO versions perform better than basic CNNs. In this work, YOLOv8 is used for practical testing, while YOLOv11 is discussed to understand how newer models might improve fire detection further.

A. YOLOV8 Model

YOLOv8 is a real-time object detection model that processes an image in a single forward pass. Unlike older CNN methods or multi-stage pipelines, YOLOv8 directly predicts bounding boxes and class scores, making it extremely fast and practical for fire monitoring through CCTV.

B. YOLOV8 Architecture

YOLOv8 is structured around three major components: the backbone, the neck, and the detection head. The backbone is built on a Cross Stage Partial (CSP) architecture, which is responsible for extracting low-level and mid-level features such as colour intensity, flame texture, edges, and other visual cues that help characterize fire. This CSP design reduces computational load without compromising accuracy, making it particularly effective for detecting objects like flames that change shape rapidly. The neck of the model is designed using a PAN-FPN structure, which fuses information from both shallow and deep network layers.

This multi-scale feature aggregation allows the model to detect flames of varied sizes and at different distances, ensuring that both small sparks and larger fire regions are represented accurately. Finally, the anchor-free detection head removes the need for predefined anchor boxes and predicts bounding boxes directly from extracted features. Since fire does not have a fixed geometry or boundary, this anchor-free approach improves flexibility and makes YOLOv8 more reliable for flame detection. For fire-detection tasks, YOLOv8 processes each incoming frame through these stages sequentially. The backbone extracts key flame-related features, which are then refined and combined across multiple scales by the neck. The detection head generates bounding boxes along with confidence scores, after which the system highlights flame regions within the frame. This design enables the model to detect typical fire characteristics such as yellow to orange red colour patterns, irregular and fast-changing shapes, and flickering movement — all of which are essential indicators of flame presence.YOLOv8 offers several advantages that make it suitable for real-time fire detection. Its inference speed is extremely high, allowing it to process multiple frames per second, which is essential for CCTV-based monitoring. The model is lightweight and can run efficiently not only on high-end GPUs but also on regular laptops or compact devices such as the NVIDIA Jetson Nano. It demonstrates improved mean Average Precision (mAP) compared to earlier versions like YOLOv5 and produces fewer false positives due to better feature extraction and architectural enhancements.

However, despite these strengths, YOLOv8 is not without limitations. Its accuracy tends to decrease in environments with dense smoke, as visual features become harder to distinguish. The model may also struggle in very low-light conditions where flame visibility is minimal. Additionally, highly reflective or shiny surfaces sometimes cause the model to misinterpret bright reflections as small fire regions, which can lead to false alarms. These limitations highlight the need for further improvements in future architectures to enhance robustness in diverse real-world scenarios.

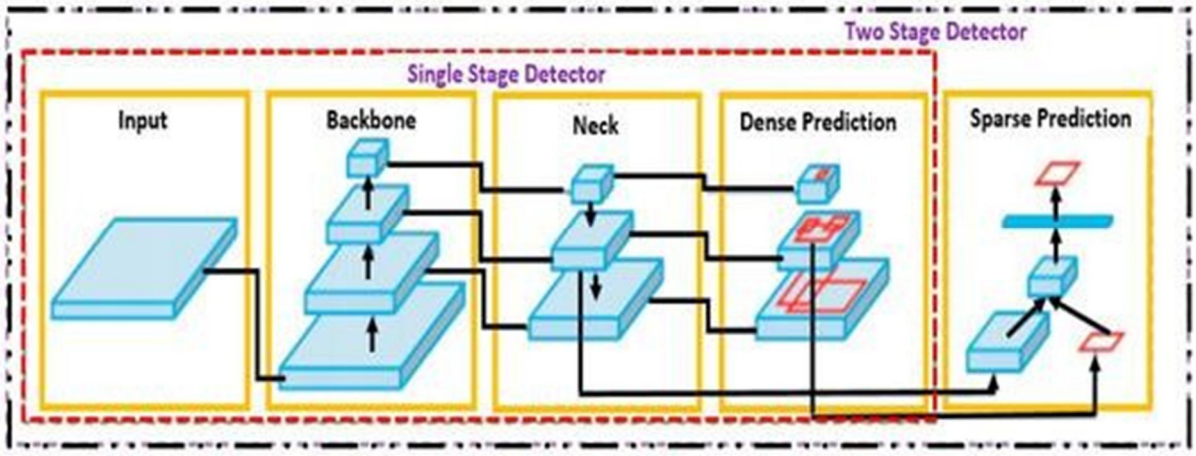


Figure 2. Yolo Model

C. YOLOV11 Model

YOLOv11 is one of the newer versions of the YOLO family, and it came out around 2025 with several changes that mainly focus on stability and better accuracy. The idea behind this version is to catch very small fire regions and reduce unnecessary detections in scenes where there are a lot of things going on. In many real environments, especially in factories or outdoor areas, the background is messy, and earlier models sometimes got confused.

YOLOv11 tries to handle these situations more confidently. One of the big improvements is in its backbone. The enhanced C2f backbone picks up finer details from the image, so even weak or half-visible flames get noticed. This also helps when there is smoke or the video is slightly dark. The next important part is the attention mechanism. Instead of treating the whole frame equally, the model gives more importance to regions that actually look like fire. This is helpful because sometimes metal surfaces, reflections, or random bright spots trick older models. YOLOv11 ignores a lot of that extra noise. The detection head has also been made more stable. When the flame flickers or moves suddenly, the bounding box does not jump around too much. It stays more consistent and marks the fire more clearly. This stability also reduces false alarms, which is a major issue in automated fire-monitoring systems. Things like headlights, sunlight coming from windows, and reflections from glass often caused mistakes earlier, but YOLOv11 handles such situations a bit better. During fire detection, YOLOv11 can identify very small sparks, flames from far away, and fire in dim or uneven lighting. This makes it useful for CCTV footage where lighting is not always perfect. It also performs well in backgrounds that have people, machinery, traffic or other movements, which usually make detection difficult.

TABLE 2. COMPARISON OF CNN, YOLOv8 and YOLOv11

Feature	CNN (traditional)	YOLOV8	YOLOV11
Type of model	Image classifier	Real time object detection	Advanced Real Time detector with improved backbone
Speed	Slow for video	Very fast	Fast
False Alarm	High	Lower false alarms	Very low false alarm
Architecture Complexity	Simple	Medium	Higher
Training Time	Short	Moderate	Higher
Real-Time suitability	Not suitable	Excellent	Excellent (slightly heavier)
Stability of Detection	Unstable	Stable	Highly Stable

D. Working Model

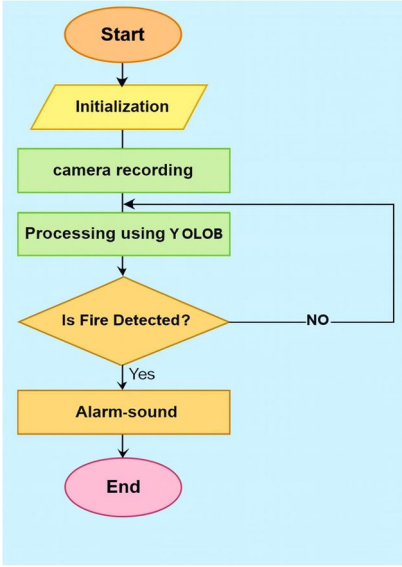


Figure 3: Flow Diagram Of Working Model

The proposed fire detection system processes real- time video input and identifies fire using two deep- learning models: YOLOv8 and YOLOv11. The overall workflow of the system is shown in Figure 3.

First, the system captures continuous frames from a camera. Each frame is pre-processed and passed to the detection module. Both YOLOv8 and YOLOv11 extract visual features such as flame color, motion, and shape directly from the image. YOLOv11 further enhances detection in low-light and complex environments due to its improved backbone and attention mechanism.

After processing, each model outputs predicted bounding boxes and confidence scores. If the confidence score exceeds a defined threshold, the system classifies the frame as “fire detected” and triggers an alert. If the threshold is not met, the frame is marked as “no fire”.

E. Confusion Matrix Evaluation

The performance of the system is evaluated using a confusion matrix, shown in Figure

The Matrix Consists Of Four Outcomes

- True Positive (Hit): Fire is present and correctly detected.
- False Positive (False Alarm): Fire is absent but the system predicts fire.
- False Negative (Miss): Fire is present but the system fails to detect it.
- True Negative (Correct Rejection): Fire is absent and correctly classified.

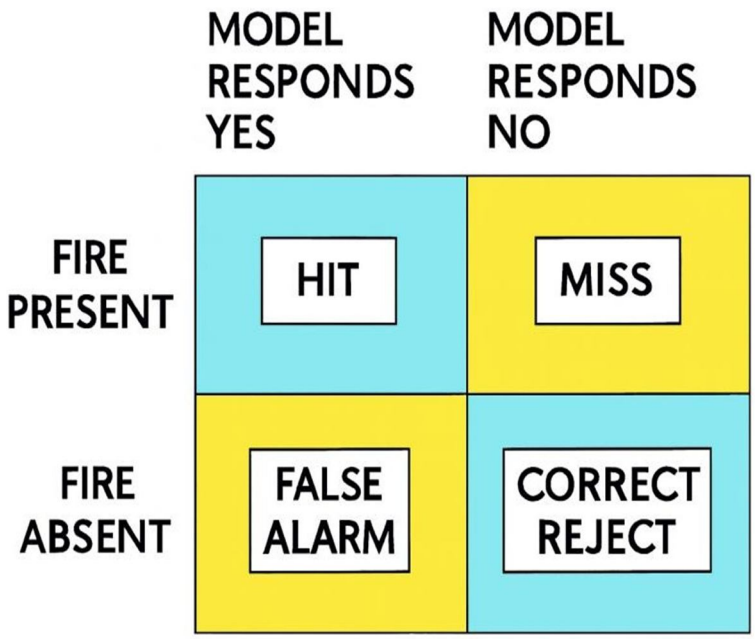


Figure 4. Confusion matrix

IV. RESULTS

The fire detection system was tested using the YOLOv8 model in a real-time environment with a webcam and a candle flame. The purpose of the testing was to observe how accurately the system identifies the presence or absence of fire.

This project was carried out utilizing a new technique by employing the YOLOV8 model idea YOLOV8 was used to detect the fire for each pixel in the frame and each pixel from the frame was then subjected to the following conditions:

- If confidence > 40
  Then give an alert sound.
- If confidence < 40
  Then do not give an alert sound

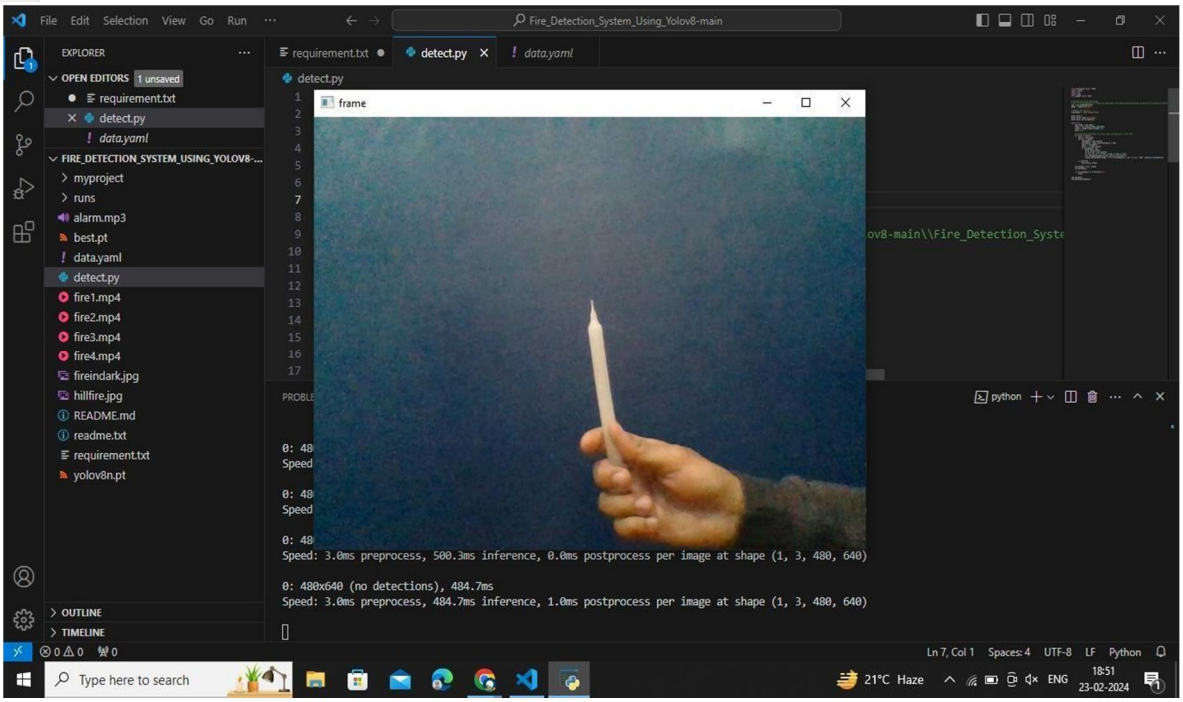


Figure 5. Candle without fire

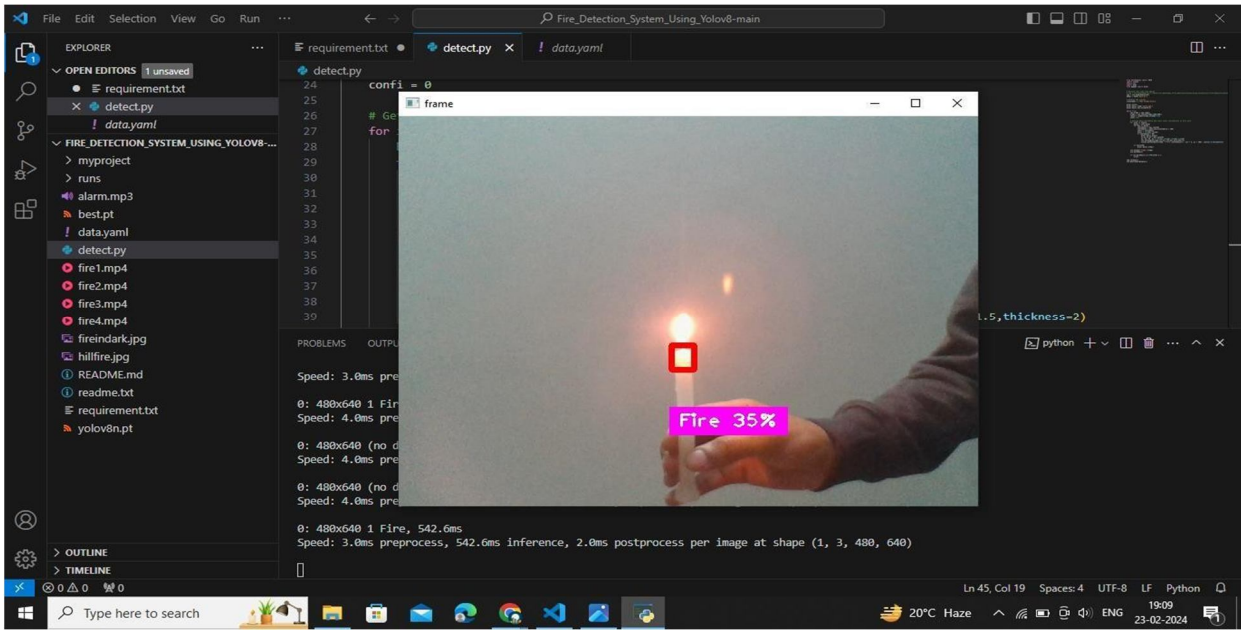


Figure 6. Candle with fire

A. Case 1: No Fire Detected

In Figure 5, the candle is completely unlit. When I tested this frame, the YOLOv8 model basically scanned it like any normal picture and didn't highlight anything. There was no bounding box or any sign of detection. This makes sense because there was nothing bright or moving in the frame that could confuse the model. What I noticed during testing is that YOLOv8 stays calm in these situations. A lot of simple detection systems sometimes react to reflections or random light spots, but here nothing unusual happened. It just showed the plain frame, which actually feels reassuring because false alarms are a big problem in real setups. If a system keeps ringing or marking things when nothing is wrong, people stop trusting it. So this test kind of showed that the model is predictable and doesn't panic when there's nothing to detect.

B. Case 2: Fire Detected

In Figure 6, the candle is lit and the flame is clearly visible. The moment the flame appeared, YOLOv8 caught it almost instantly and drew a red box around it. The confidence score (for example 35%) also popped up on the screen. Even though 35% doesn't sound very high, for a tiny candle flame it is actually normal because the flame keeps changing shape every second.

While checking these frames one by one, I noticed that the box appears immediately and stays around the flame even if the fire flickers or moves slightly. This part is important because flames hardly stay still. The model still manages to follow the shape without jumping around or giving a shaky detection.

Overall, this small test showed that YOLOv8 reacts quickly and doesn't need a big fire to start detecting. Even a small flame was enough for it to recognise the fire. So for early- warning situations, where seconds matter, this behaviour is useful. It may not be perfect in every lighting condition, but in a normal room setup it worked well.

C. YOLOv8 vs YOLOv11

YOLOv11 was **not implemented** in this project. Therefore, its performance is **not experimentally tested**.

The comparison included in this paper is based solely on **recent research studies**, which report that:

- YOLOv11 provides improved small-object detection
- YOLOv11 is more stable in complex environments
- YOLOv11 reduces false alarms due to better feature extraction

This comparison is included to highlight how newer architectures may further improve fire detection in future work.

V. CONCLUSION

In this work, I mainly tried to see how a simple fire detection system can be made using YOLOv8, without any complicated setup. I just used a normal webcam and some basic frames, and surprisingly the model reacted quite well. Whenever the candle flame appeared, YOLOv8 picked it up almost immediately, and when there was no fire, it didn't show anything. This part was actually nice because false alarms are very irritating in real-life situations, and at least in my small test, it didn't behave in a strange way.

While doing the tests, I noticed that YOLOv8 works fine when the lighting is normal. But in slightly darker areas or when the flame was too small, the model hesitated a bit or the confidence score went down. So it's not perfect, but for a basic trial, it was okay. When I checked the confusion-matrix part, the results more or less matched what I saw: good detections in simple scenes and a few misses when the situation was tricky.

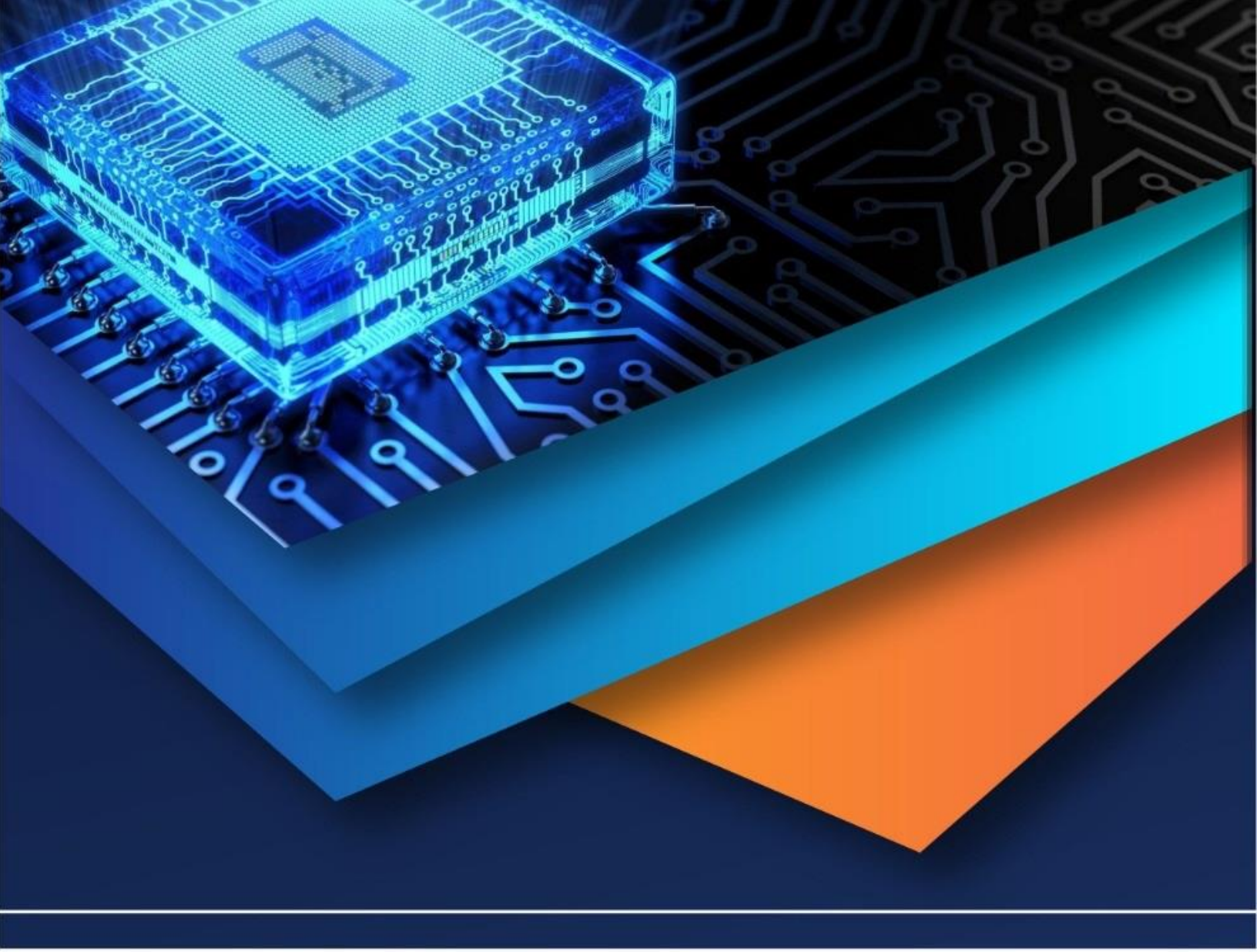
YOLOv11 was not used here, but I still mentioned it because many papers say it performs better with small objects and in difficult conditions. Maybe if this project is continued later, YOLOv11 can be tried to see if the small-flame problem gets solved or not. Right now, I only discussed it to show how newer versions might improve things.

Overall, the study basically shows that YOLO models can be used to make a simple fire-detection idea work, especially when someone wants something fast and not too expensive. If more data, more lighting variations, or even outdoor tests are added later, the system can be improved. So this project is more like a starting point, and there is a lot that can be done in the future to make it more stable and reliable.

REFERENCES

[1] Kumar P. M., Fire Detection using Deep Convolution Neural Networks in Video Streams, 2022.
[2] S. Kamalanathan, Smart Surveillance System for Abnormal Activity Detection using CNN, 2021.
[3] H. Qian, A Fire Monitoring and Alarm System Based on Channel-Wise Pruned YOLOv3, 2022. [4] A. Robert Singh, Fire Detection by Parallel Classification of Fire and Smoke Using Convolutional Neural Networks, 2021.
[4] Y. Zhang, P. Geng, and C. Sivaparthipan, "Big data and artificial intelligence-based early risk warning system of fire hazard for smart cities," 2021.
[5] L. Zhao, Fire-YOLO: A Small Target Object Detection Method for Fire Inspection, 2022.
[6] S. S. Muhammad, Fire Detection Using DenseNet-201 Algorithm and Surveillance Camera Images, 2024.
[7] S. Ke et al., Improved YOLOX Fire Scenario Detection Method, 2022.
[8] D. C. Prabhu T., "Automated Home Security using Facial Recognition with IoT Remote Monitoring," IEEE IPACT, 2021.
[9] L. Lou et al., "Smoke root detection from video sequences based on multi-feature fusion," 2022.
[10] S. Wang, ES-YOLO: A New Lightweight Fire Detection Model, 2023.
[11] C. Li, Research on Fire Detection Algorithm Based on Deep Learning, 2022.
[12] A. M. O. et al., "Model of settlement evacuation based on imitation modeling application," IOP Conf. Ser., 2020.
[13] F. M. Talaat, An Improved Fire Detection Approach Based on YOLO-v8 for Smart Cities, 2023. [15] S. H. Park, Recognition of IoT-based Fire Detection System Fire-Signal Patterns Applying Fuzzy Logic, 2023.
[14] S. Bashambu, Real-Time Fire and Smoke Detection System, 2023.

[15] S. G. Zhang, Swin-YOLOv5: Research and Application of Fire and Smoke Detection Algorithm, 2023.
[16] D. Zhang, A YOLO-Based Approach for Fire and Smoke Detection in IoT Surveillance Systems, 2024.
[17] A. S. Mahdi, Wildfire Detection System Using YOLOv5 Deep Learning Model, 2023.
[18] S. Msedd, Fire Detection and Segmentation Using YOLOv5 and U-Net, 2021.
[19] H. Wu, Ship Fire Detection Based on an Improved YOLO Algorithm with Lightweight CNN Model, 2022.
[20] H. Bayer and A. Aziz, Object Detection of Fire Safety Equipment Using YOLOv5 Neural Network, 2022.
[21] S. Yu, Forest Fire Detection Algorithm Based on Improved YOLOv5, 2022.
[22] S. Kumar, YOLOv4 Algorithm for Real-Time Detection of Fire and PPE at Construction Sites, 2021.
[23] A. Bochkovskiy, C-Y. Wang, and H-Y. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv:2004.10934, 2020.
[24] Y. Kumar, H. Gupta et al., A Novel YOLOv3 Algorithm-Based Deep Learning Approach for Waste Segregation, 2020.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY