International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Fire, Smoke and Flame Only Detection Based on Artificial Intelligence Techniques

A. Faris Al-khafagi[1], A. Subri[2]

[1]Department of Computer and Information Engineering, Faculty of Software Engineering Division, Islamic University, Babylon, Iraq
[2]Department of Applied Mathematics, Islamic University, Babylon, Iraq

Abstract
A unique tracking machine for fire and smoke detection is critical to ensure human protection and safety. Although cutting-edge fire alarm systems offer realistic solutions, there may nevertheless be an urgent need for greater accurate detection techniques. Fires can cause extensive harm; consequently, early detection is important. Convolutional Neural Networks (CNNs) are deep learning techniques that have been advanced to understand smoke and fire in images and video frames. The YOLO (You Only Look Once) model has shown incredible ability, especially the advanced YOLOv8 version. YOLOv8 offers faster and greater accurate detection of smoke and fire capabilities. In this examine, we advise using YOLOv8 for flame detection and check its performance in comparison to standard shallow learning models based on fuzzy common sense, color, motion, and shape. The tough Smoke and fire dataset, which incorporates a wide range of actual-world images, was used to assess the fashions. The outcomes exhibit that YOLOv8 outperforms conventional techniques in terms of model length, accuracy, and detection pace. With an average common Precision (mAP) of 95.3%, it gives an effective solution for smart flame and smoke detection.

Keywords: Artificial Intelligence, Convolutional Neural Network, Deep Learning, Image Processing, Fire, YOLOV8.

1. INTRODUCTION

Fire can start everywhere and pass fast, so its heat, smoke, and glow may also hide in the back of shadows, trees, or fences that everyday eyes bypass. Conventional detectors, which concentrate on heat, smell, or tiny flame sparkles, may also leave out the early tips in wide-open or windy spaces. Whilst warning comes late, injuries increase, evacuation expenses soar, and insurance payouts climb, so catching smoke and sparks the second they appear stays key, and taking into consideration short emergency movement. With expanded industrial interest and urban growth, fire threats have turned out to be more common and dangerous, necessitating more dependable and smart surveillance structures. Traditional fire alarm structures regularly rely upon sensors or guide tracking, which may lead to delayed responses and false signals. Current upgrades in laptop imaginative and prescient and synthetic intelligence (AI) have created new prospects for developing more dependable and accurate fire detection technology [1][2]. Studies and development of more dependable and efficient fire detection gadgets are therefore desperately wanted. Networked sensors ought to capture fire flames [3]. Fires account for as much as 50% of all occurrences, which might be pronounced, with devastating outcomes in terms of both harm and fatalities. Therefore, constraints that still apply to conventional sensors have to be addressed to increase fire detection generation [4]. The part mitigation plays in lessening these negative impacts is a critical limitation. Moreover, this paper highlights a basic factor of fire dynamics, particularly that the production of smoke is regularly a precursor to imminent calamity

Despite the availability of many fire detection systems, many of them have limitations in terms of effectively distinguishing between fire, smoke, and other ambient noises such as fog, illumination changes, or moving objects [5]. These flaws may lead to false positives or missed detections, which can have serious repercussions. As a result, the challenge resides in the necessity for a real-time, automated, and highly accurate fire and smoke detection system that can work efficiently in a variety of circumstances and environments [6]. The market for fire detection and alarm systems is now dominated by three primary strategies: deep learning techniques, image processing techniques, and sensor-based techniques [7]. To improve early warning and combat strategies, this paper investigates state-of-the-art fire detection technologies and methods that surpass the capabilities of conventional sensors. By doing this, we hope to lessen the detrimental effects of fire events and safeguard people's lives and property in our environment, which is becoming more and more vulnerable to fire. Firefighting apparatus could only detect smoke precisely when it was contained, and early smoke detectors were unable to locate smoke precisely.

To solve this challenge, we suggest the use of the YOLOv8 (You Only Look Once version 8)[8] deep learning model for flame and smoke identification in pictures and video streams. YOLOv8 is a cutting-edge object identification framework noted for its speed, lightweight architecture, and excellent detection accuracy. In this paper, we use YOLOv8 on the difficult Smoke and Fire dataset and compare its performance to classic shallow learning models that use heuristic-based techniques, including color, motion, shape analysis, and fuzzy logic. Our results show that YOLOv8 surpasses standard methods in terms of accuracy, detection speed, and model efficiency, making it an ideal alternative for intelligent fire and smoke monitoring applications. This paper makes the following key contributions:

- We suggest the use of YOLOv8, a modern and highly refined object detection model, for identifying fires and smoke in real-time from pictures and video streams. To the best of our knowledge, this is one of the first attempts to use YOLOv8 for fire safety monitoring.
- Experiments compare the YOLOv8 model to traditional fire detection methods, including color analysis, motion tracking, shape features, and fuzzy logic. This emphasizes the limits of traditional methodologies while demonstrating the advantages of newer deep learning models.

The remainder of the paper is structured as follows: Section 2 provides a comprehensive review of related works in fire and smoke detection, highlighting the limitations of traditional and classical machine learning approaches. Section 3 details the methodology, including the YOLOv8 architecture, dataset description, preprocessing steps, and experimental setup. Section 4 presents and discusses the experimental results, comparing the proposed model with baseline techniques in terms of accuracy, speed, and robustness. Finally, Section 5 concludes the paper and outlines potential directions for future research.

2. Literature Review

Following the presentation of relevant solutions that have been created to address these difficulties, the latest models for smoke and fire flame detection are introduced. Methods for computer vision-based detection. Presentation of relevant solutions created to address these problems. Computer Vision The interdisciplinary field of computer science known as computer vision studies how computers can be designed to accurately and fully comprehend human visual abilities. Regarding digital images or videos

[9]. Algorithms for automatic video fire or smoke detection have been developed for use in tunnels, aircraft hangars, and ships. They usually concentrate on small, crowded spaces. Creation of reliable video. A lot of studies have been done on fire detection systems in large or open areas.

K. Iqbal et al. [10] proposed an AI-powered fire and smoke detection system that combines the YOLO object identification algorithm with a CNN architecture. Their solution works with an Android app, which uses a camera to continuously scan the area and give real-time alerts to users. The COCO dataset is used to train the YOLO model, and the Django framework is used to create the backend of the application, allowing for seamless communication between the deep learning model and the Android UI. Unlike existing fire detection systems, which rely on passive sensors (heat, gas, flame, smoke), the suggested system is low-cost, image-based, and cloud-connected, making it less susceptible to environmental interference and requiring less maintenance. One of the primary benefits of this strategy is its capacity to give early fire detection using visual data, which improves responsiveness. However, one notable drawback is the use of a generic dataset, Common Objects in Context (COCO), rather than a fire-specific dataset, which may have an impact on detection accuracy in real-world fire scenarios. Further upgrades could focus on tailoring the model with site-specific footage and on streamlining the software so it runs smoothly at factory size.

D. Gragnaniello et al. [11] introduced the FLAME agile framework that classifies outdoor video streams for fire. Seeing how earlier systems often scream fire at harmless shadows, they designed an approach that links a modern deep-network detector with a physics-based motion check. By marrying these two ideas, FLAME keeps its fire alert rate high yet slashes false alarms born from waving trees or passing cars. Another strong point is that the whole pipeline runs in real time on lightweight hardware, perfect for cameras and small alert boxes. In head-to-head runs on a large, public fire record, it also clocked better accuracy than rival methods. The catch is that users still need to set motion thresholds carefully, and those same rules might fail when the scenery or weather changes a lot.

C. Zhao et al. [12] introduced SF-YOLO, a streamlined deep-learning network created to spot smoke and flames in real-world scenes. They built on YOLOv11, adding the C3k2 block, which features a dual-path residual attention layer and an attention-laced head that handles occlusions and pulls out clear features even with busy backgrounds. Testing showed SF-YOLO beat strong rivals such as YOLOv8, Gold-YOLO, and Faster-RCNN on both accuracy and stability. The team validated their work on a private dataset, S-Firedata, as well as the public M4SFWD collection, and recorded the same edge over those models. Remarkably, mAP scores at both mAP50 and mAP50-95 rose while the network stayed lightweight. By tackling occlusions head-on, the system offers a solid boost to outdoor fire monitoring.

M. S. Sozol et al. proposed similar challenges by presenting a hyperparameter-tuned YOLOv5, dubbed HPO-YOLOv5, which they optimized with a genetic algorithm. For training, they created a new indoor fire-and-smoke dataset of 5,000 images that captures varied room layouts, light levels, and smoke behaviors, helping HPO-YOLOv5 learn robustly across real scenes. They first applied Grad-CAM so the team could see which image regions drove each prediction. To monitor fires as they spread in real-time, the authors paired YOLOv5 with DeepSORT, a CNN-based tracker. The resulting pipeline logged a mean Average Precision at 0.5 of 92.1%, outperforming top contenders-Faster R-CNN, YOLOv7, and YOLOv8-by at least 2.4% over the baseline YOLOv5. Such numbers promise quicker, steadier alerts in fast-changing indoor scenes and lay solid groundwork for smarter, self-learning fire alarms. Still, real-world tests in scenes far busier than the training set are needed before the system can be declared field-ready.

Lakshmi et al. [14] offered an approach that marries YOLOv8s' near-immediate smoke-and-flame

recognition with XGBoost's crisp final verdicts. Thanks to the more youthful layered characteristic maps, the version flags warm pixels even in cluttered or low-mild rooms. When those proposals reach the booster, it types them into the fireplace or false alarm using guidelines found in many datasets. The mixture runs live, hits top precision, and suits any clever watchtower or campus shield app. Builders at the moment are fine-tuning the hand-off code so the pipeline runs nonstop without lacking a single frame.
A. Aithal et al. [15] provided a new method that pairs a sturdy fire-and-smoke engine with a tweaked YOLOv8 model aimed only at smoke, and they coded it to work on CCTV feeds in real time. By letting both networks share what they see, the setup spots flames and fog with high marks, no matter if the light is bright, dim, indoor, or out. Training ran on two Roboflow collections that alone hold over 20,000 pictures; the team added 1,290 labelled video frames and then stretched every image with common tricks to boost sample size. Bench tests show precision and recall at 0.98 and 0.99, numbers most engineers would gladly claim. Because of this accuracy, the combined toolkit offers watchdogs a sturdier line of defense than many old-school sensor grids. Still, how well the system scales and behaves in messy real-world scenes, such as parking lots or windy rooftops, needs more study.

3. METHODOLOGY

3.1.YOLO Architecture

YOLO is a real-time object detection system that was first released in 2016. To identify objects, classify them, and determine their bounding box coordinates, the YOLO technique uses regression. A YOLO architecture typically has three parts: the head, neck, and backbone. They play the following roles: a network intended to retrieve informational characteristics from pictures for later use on the network. The Neck, which sits between the Head and the Backbone, is essential for feature fusion and allows for the better utilization of features that the Backbone has extracted. The Head uses the object's extracted features. The YOLO algorithm's original Backbone network topology had two completely linked layers after 24 CNN layers. While the next CNN layers perform feature extraction while the fully connected layers predict bounding box coordinates and output probabilities. The backbone network architecture experiences major changes due to the technological advancements of YOLO. The system also estimates the probability of an object being present in a specified bounding box. A grid cell, as depicted in the image, is responsible for object detection if its center falls inside the grid cell. Each grid will have many bounding boxes. During training, as in the case of a real-time model, four descriptors can define each, giving YOLO a clear advantage [16]. YOLO achieved 45 fps with no drop in performance, unparalleled by others. Meanwhile, for flame-only, smoke-only, and fire detection tasks, we aim to simplify the YOLO network to improve its efficiency and deployment on low-resource devices. The YOLO model employs a unique feature extraction method that enables it to distinguish between fire, smoke, and flame patterns. Fig.1 illustrates the detection results for each category. To generate the final prediction bounding boxes, the YOLO network computes the center, width, and height of each detected object, assigning it to its corresponding category. Fig.1 includes multiple examples demonstrating the network’s predictions for the three fire-related classes.

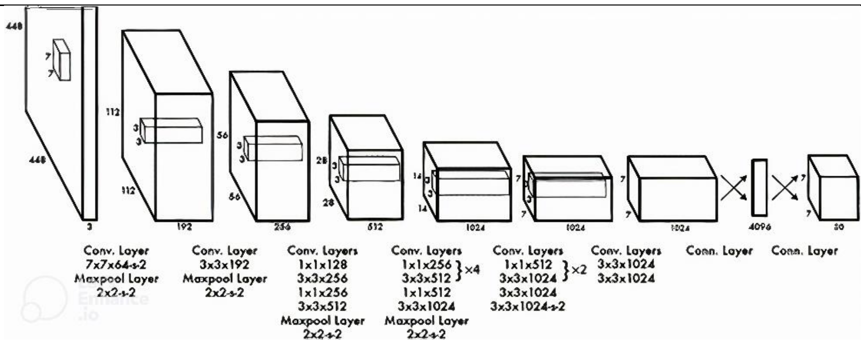

Fig.1. YOLO architecture for object detection and localization [17]

3.2.YOLOV8

Several studies focused on the traditional feature extraction method for smoke and fire detection. The time needed to calculate these feature extractions was the main problem with such systems. This resulted in slow and subpar real-time fire and smoke detection. These methods also resulted in several background detection failures and false positives. In response to current developments in deep learning algorithms, we suggested using the YOLOv8 model for smoke and fire detection. The imaging processing for smoke and fire, flame detection, is finished faster. In this case, YOLOv8 is the best approach to deal with the detection of these things. A neural network was then used to generate the model by labeling the practice photos. YOLOv8 layers as well. We also trained, validated, and evaluated the YOLOv8 technique to assess its accuracy and efficacy. Despite the lack of formal publication, the available repository and discussions offer insights into the remarkable strides enabled by YOLOv8. The backbone and neck, and head of the YOLO system constitute its three main architectural features, which remain the same in YOLOv8. It is clear from Fig. 2 that these components coordinate backbone extracts visual features, neck integrates and refines these features, while the head is responsible for final object detection and recognition. Convolutional operations, like the core building blocks of neural networks, remain central in YOLOv8. Notably, YOLOv8 introduces a more efficient and flexible design by replacing the original 6×6 convolutions in the stem with 3×3 convolutions, and by substituting the C3 module with a more streamlined C2f module [18]. Fig. 2 provides a visual overview of this architectural structure.

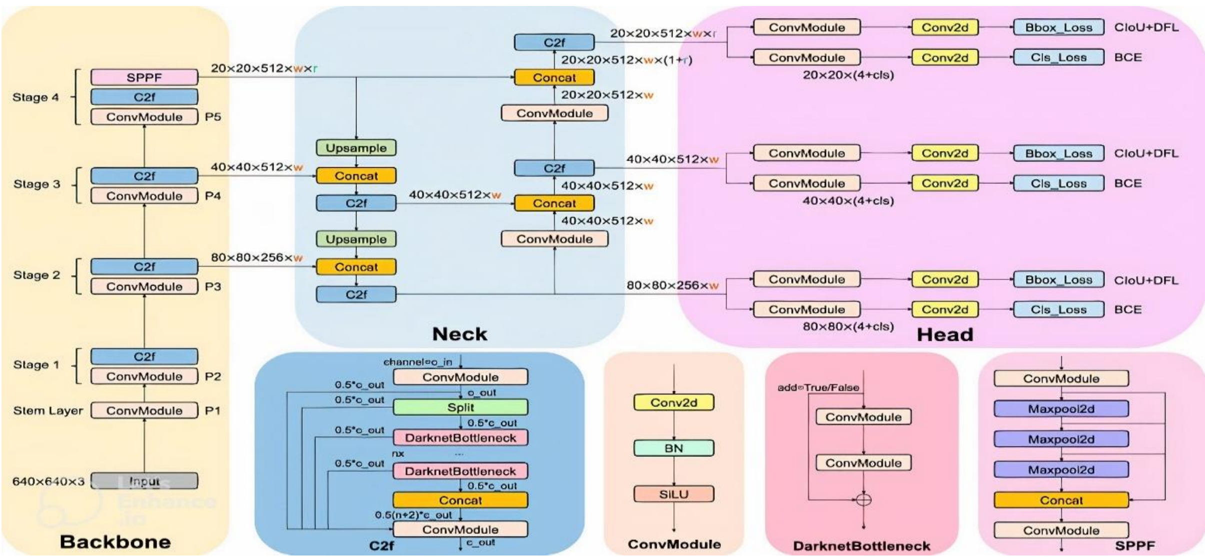

Fig.2.YOLO v8 architecture [19]

The Range King-created YOLO v8 architecture on GitHub is shown in Fig.2. Here are several ways that it is different from YOLO v5.

- The C3 module has been upgraded to the C2f module.
- On the spinal cord, replace the initial 6 9 6 transform while adding a 3 9 3 transformation on the spine. Removal of conversions is required on the YOLOv5 setup.
- Adapt the original 1 9 1 transformation of the bottleneck into a 3 9 3 transformation.
- Eliminate the object using the split head method

C2f has been swapped out for C3 as the fundamental building block, and the stem's original transformation from 6 9 6 to 3 9 3 has been altered. The component block CBS conv, BatchNorm, and SiLU, which are positioned lower in Fig. 2, illustrating that the unit is also CBS, identifies the block features. f outlines the total features counted, while e, the defining rate, indicates the growth in features. The bottleneck is still the same as in YOLOv5but the original convoy size's core has been changed from 1 9 1 to 3 9 3. The significant update that the YOLO series received is shown in the YOLOv8 architecture [20]. A noteworthy development in the YOLO series establishes the benchmark for future efforts in computer vision. YOLOv8 is becoming a more attractive alternative in the real-time object detection market because of its high accuracy rates, innovative architectural modifications, and increased developer possibilities. YOLOv8 is available in a variety of sizes, including nano (n) and small (s), just like its predecessors. These five models are also referred to as medium (m), large (l), and extra-large (x) due to their differing adaptability to changing application needs.

3.3. Dataset

3.3.1 Dataset Description

Since any object detection model requires a dataset to be trained, the dataset is an essential component of the model. Mistakenly, most people assume a dataset is merely a compilation of pictures that can be randomly sourced. This assumption will unavoidably result in a below-average object detection model. Depending on the model being used, the data set may consist of either numbers or pictures. For our model, we chose an image dataset, which Robflow [21] also used. The images were manually annotated with the terms fire, flame, and smoke; the gathered dataset contains 9899 images. We found that there was not enough data to generate reliable detection results. The quantity of images can be increased by employing techniques such as picture augmentation.

3.3.2 Image Pre-Processing

The accuracy rate of our network is only about 95.7%, as shown by the experimental results in the section above. One of the factors contributing to subpar network performance is the overfitting issue. On test data with just 9899 images, it performs poorly. We use additional datasets to train the network to solve this problem. Open access makes it possible to use several data augmentation techniques to add more pictures. This idea is advantageous to our network. To raise one's level of performance. A common technique for boosting the diversity and variety of data collection to boost deep learning's robustness and performance is data augmentation. Artificial intelligence deep learning models are also available. When doing our real data collection process, we used a few photos using augmentation techniques to add realism to the dataset. As can be seen below, we applied several data augmentation techniques to photographs. In this case, the output images for every training example, 15% of the images were in grayscale, with rotations ranging from 15° to +15°. Temperature range for color: -20° to +20°. They used mosaic. Four images are combined into one using mosaic zooms. As a result, everything in the picture is connected. They are smaller than the original picture. In terms of detection, this type of increase is advantageous. For the first time, YOLOv4 [22]. Introduced enhanced precision in identifying tiny objects by incorporating them into the training process using advanced image augmentation techniques.

One of the most effective methods is the mosaic augmentation technique, which combines four different images into a single training sample, thereby increasing both the context and diversity. As illustrated in Fig.3, this method enhances detection performance, particularly for small objects, by enriching the input data distribution. Furthermore, Fig.4 demonstrates the effectiveness of a YOLO-based detection dataset and its significance.

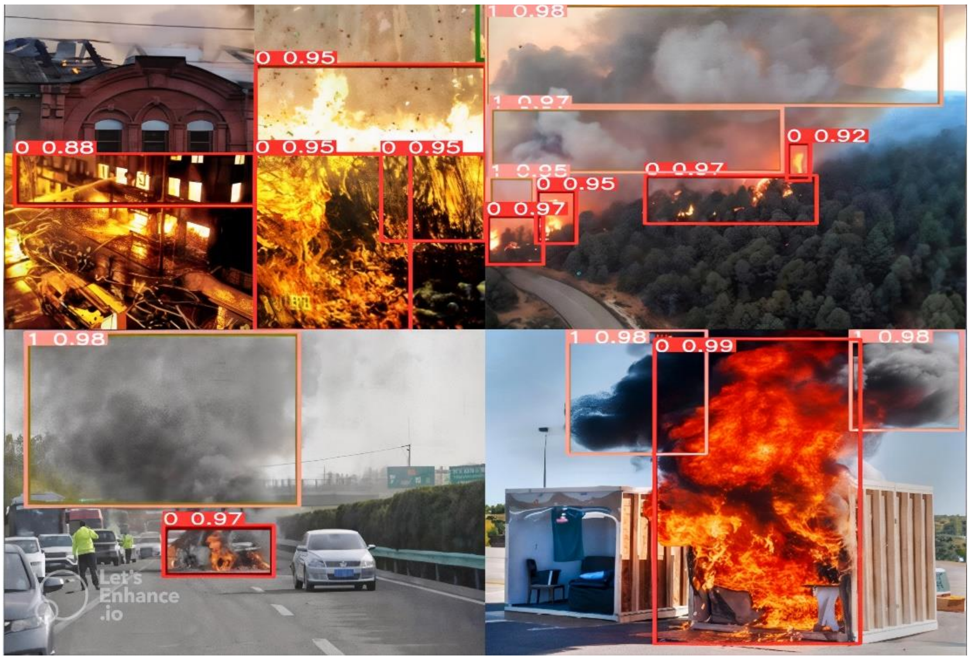

Fig.3. Object Detection Scenarios Using YOLOv8 for Fire and Smoke Recognition

Fig.4. Examples of detection with Yolov8v

3.3.3 Data Splitting

While avoiding overfitting, data splitting enhances the usefulness of the model and its capacity to generalize to new data. Before using the algorithm, it is recommended to perform a data inspection. Existing data was divided into two or three subgroups using this technique. Training, validation, and test sets are the names given to these subsets. It is employed to generate feature sets that can be utilized to train the model. This SpitNN technique tackles the problem of training the model on data items. What is a big data collection? This data is easily usable for training, enabling our model to pick up increasingly intricate and varied characteristics. Data can be divided into training and testing sets using a variety of techniques. Using random sampling is the most widely used technique. Entirely random selection. This approach ensures frequency distribution and is easy to implement. The outcomes are quite equivalent between the training and test sets [23]. To guarantee the accuracy of the model in real time, a distinct test suite is maintained. As a result, the present Instead of being separated, the dataset was divided according to the ratio into training and validation sets, and then into test, validation, and training sets. Both the model architecture and the type of data that was gathered have a big impact on the data split ratio. In tasks involving large image and video datasets, a substantial portion of the data is typically allocated for training to ensure sufficient model learning. However, when the model has numerous hyperparameters, it becomes increasingly important to retain a larger portion of the data for validation to fine-tune these parameters effectively. As illustrated in Fig.5, the dataset is split accordingly to balance learning and generalization, depending on the complexity of the task and the model configuration [24] [25].

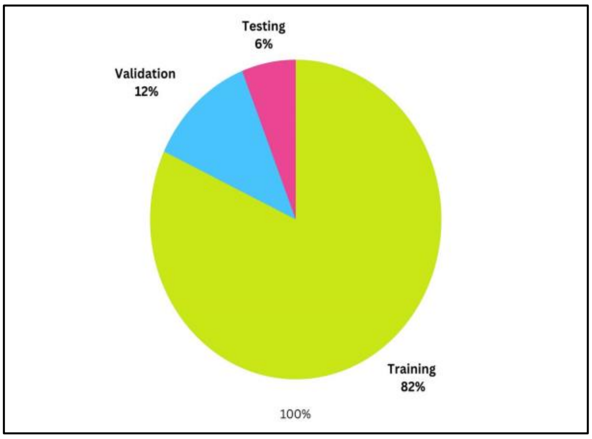

Fig.5. Analyze the data set splitting F

4. Results And Discussions

Evaluating the coach model's performance is just as important as collecting the data and training the machine learning model, which are essential initial phases in the deep learning process. Determining whether the model can be used to solve the problem and how well it generalizes to unseen data are critical. We assess the YOLO models using the metrics of mAP, recall, and accuracy. The performance of the proposed YOLOv8-based method was assessed in this study utilizing a range of evaluation metrics. The performance of object identification models is evaluated using a statistic known as mAP. The mAP value is generated by averaging the Average Precision (AP) across all classes. Precision and Recall: Precision assesses the accuracy of your predictions by calculating the proportion of correctly positive classifications. The definition of precision is as follows:

$\frac{Ture\ postive}{Ture\ postive+Fals\ postive}$	(1)

Recall tests your ability to discover every positive and displays the proportion of positives that the classifiers accurately identified. This is the formula for recall.

$\text{Recall} = \frac{T\,p}{Tp+FN}$	(2)

The F1 score, which is the harmonic mean of Precision and Recall, aims to strike a balance between these two attributes. This could be read as.

$\text{F1 score} = 2 * \frac{R*P}{R+P}$	(3)

Based on the evaluation metrics presented, after training, our model obtained a remarkable F1 score of 0.94, as shown in Table 1. The evaluation metrics presented indicate that the object detection model performs with a high degree of accuracy and reliability across all three target classes: fire, smoke, and flame. Overall, the model achieved a precision of 0.943 and a recall of 0.902, which shows that it is both accurate in its predictions and capable of identifying the most relevant instances. The version also suggested sturdy detection satisfactory with aF mAP @0.5 of 0.953 and a more stringent mAP @0.5:0.95 of 0.912, confirming its robustness throughout various tiers of intersection-over-union (IoU) thresholds. Among the character lessons, the version executed excellently on the "hearth" class, reaching a precision of 0.962 and a don't forget of 0.960. Its mAP scores (0.981 @0.5 and 0.951 @0.5:0.95) advise near-perfect performance in localizing and classifying heart times. For the "smoke" magnificence, precision remained excessive at 0.931, but recall, it dropped to 0.854, implying that a few smoke times have been overlooked, likely due to their visible similarity to the heritage or versions in lighting.
The mAP scores for smoke (0.932 @0.5 and 0.873 @0.5:0.95) were still respectable, but lower than those for fire. The "flame" class showed balanced metrics with a precision of 0.925 and a recall of 0.885, and corresponding mAP scores of 0.941 and 0.896. While the model demonstrates solid performance on flames, there is slight room for improvement, potentially due to the visual overlap with fire or inconsistent flame patterns in the dataset.

Table 1. Evaluation of the Performance of the YOLOv8 Training Step

Parameters	**P**	**R**	**mAP 0.5**	**mAP 50.95**
all Class	0.943	0.902	0.953	0.912
Fire	0.962	0.96	0.981	0.951
Smoke	0.931	0.854	0.932	0.873
Flam	0.925	0.885	0.941	0.896

To further illustrate the learning progress, Fig.6 displays the accuracy over each epoch for both the training and validation datasets. This visual representation highlights the model's convergence behavior and performance consistency throughout the training process.

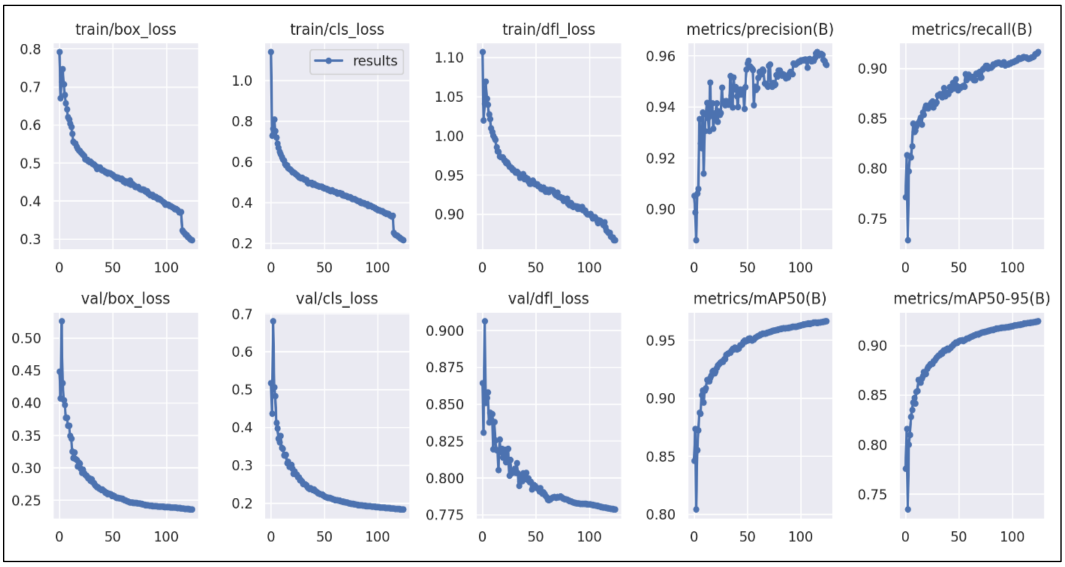

Fig.6. Graph of the Results and Losses

4.1 Model Preference

Although there is undoubtedly space for improvement, the findings reveal that YOLOv8, one of the most sophisticated A high mAP @0.5 using single-stage target detection models for broad fire classification and recognition, including structures. YOLOv8L has a detection accuracy of 0.953. The YOLOv8L training time across 125 epochs is shown in the table. The new Ultralytics. The same data collection and parameter settings were used in this experiment to compare how well it performed. The parameter Earning rate of ValueBatch-size 32 epochs 125 0.001 640 x 640 pixels 5.383 train-time. Images from three different scenes, the interior, an urban road, and a forest, are included in the fire dataset. After much testing, several techniques for data augmentation training have been identified. This case-based method of information provision is more practical. Greatly increases accuracy. The background is displayed while the model is being trained. The picture has weight as well. The basic functionality of the model has been trained, but it is ideal to train it in urban areas or at intersections of buildings. Most models in earlier studies concentrate on model optimization, which includes increased detection speed, reduced weight, and improved detection accuracy. Only pictures of urban highway fires are included in the model. We were able to achieve up-detection on both ends by using both ends. Identifying urban road fires and building fires, and identifying flames within homes. The YOLOv8 L model is used to identify several fire incidents, as shown in Fig.6. Anticipation: The model allows for exceptional alignment and precision, precisely matching the original detailed fire test photos. Remarkably, even when the model is asked to locate ever-tinier pockets of fire throughout multiple fire seasons, its performance remains constant. The model's improved performance in fire detection tasks is confirmed by the fact that despite these challenges still has a high degree of confidence in detecting fire and smoke incidents.

4.2 Comparison with Existing Methods

To assess our fire detection system, we compared it to traditional sensor-based systems and other deep learning models such as CNN and LSTM-based methods. Our YOLOv8-based system outperformed the other approaches in terms of accuracy, mAP, and recall. Our model outperformed traditional systems in real-time fire detection, achieving an mAP of 0.953 and a recall rate of 0.902 while reducing false positives and processing times. Traditional systems had lower detection rates (70%) and slower response times, while deep learning models such as CNNs and LSTMs had detection rates around 85 percent with mAP values in the range of 0.70 to 0.80.

5. Conclusion

Systems for detecting smoke and fire, flames, are essential for saving lives and stopping hazards from spreading. Existing smoke and fire detection equipment is useless in places with lots of open space. External conditions. New methods for solving a variety of problems have been made possible by deep learning, machine learning, and artificial intelligence. It is hard to solve using standard methods. (1) To raise the accuracy of flame detection, we shall endeavor to enhance the YOLO model. Additional CNN models, such as Faster Even though they, could require more time to train, but RCNNs could perform better in terms of accuracy. (2) YOLO and Faster RCNN in conjunction.
Computer vision is the foundation of our thesis, and we use data on fire and smoke from online groups, regardless of whether the data is obtained from security cameras outside or other sources. It transfers learning to lightweight neural networks, leveraging the latest advances in deep learning. This article established this extremely high level. Grouping. Even with limited data, deep CNNs can achieve performance. Weaknesses are caused by overfitting, a problem brought on by a limited collection of visual data during training. Neural network models can address this problem by using a variety of data reinforcement techniques to increase the size of our training datasets.
In the future, we hope to intensify our current efforts. Developing a robust system to identify smoke and flames competent to recognize smoke and fire. To further our understanding of these models in various settings and contexts, it would be advantageous to expand this study to more datasets in the future. This would enhance the results' generalizability and contribute to the expansion of the body of knowledge. Additionally, there is an excellent opportunity to expand on existing discoveries and do new research.

References

1. S. S. Samy, Y. S. Swarup, T. S. Kumar, C. Shankar, S. Reddy, J. S. Sudarsan, and S. Nithiyanantham, "Fire and smoke detection using adoption of machine learning algorithm for improving fire safety and disaster preparedness," *Asian Journal of Civil Engineering*, vol. 26, no. 1, pp. 1–5, May 2025.
2. J. L. Wilk-Jakubowski, V. Loboichenko, M. Divizinyuk, G. Wilk-Jakubowski, R. Shevchenko, S. Ivanov, and V. Strelets, "Acoustic Waves and Their Application in Modern Fire Detection Using Artificial Vision Systems: A Review," *Sensors*, vol. 25, no. 3, p. 935, Feb. 2025, doi: 10.3390/s25030935.
3. A. Haque and H. Soliman, "Smart Wireless Sensor Networks with Virtual Sensors for Forest Fire Evolution Prediction Using Machine Learning," *Electronics*, vol. 14, no. 2, p. 223, Jan. 2025, doi: 10.3390/electronics14020223.mdpi.com+1ouci.dntb.gov.ua+1
4. L. Zolfagharipour and M. H. Kadhim, "A Technique for Efficiently Controlling Centralized Data Congestion in Vehicular Ad Hoc Networks," *International Journal of Computer Networks and Applications (IJCNA)*, vol. 12, no. 2, pp. 267–277, 2025, doi: 10.22247/ijcna/2025/17.
5. L. Zolfagharipour, M. H. Kadhim, and T. H. Mandeel, "Enhance the security of access to IoT-based equipment in fog," in *2023 Al-Sadiq International Conference on Communication and Information Technology (AICCIT)*, 2023, pp. 142–146, doi: 10.1109/AICCIT57614.2023.10218280.ojs.cvut.cz
6. A. G. Medewar, A. D. Sawarkar, U. V. Kshirsagar, A. G. Mr, and U. Kshirsagar, "A review on fire and smoke detection with intelligent control for enhanced safety using machine learning (ML) and Internet of Things (IoT)," *Cureus*, vol. 1, no. 1, Sep. 26, 2024, doi: 10.7759/10

7. A. A. AlQahtani, M. Sulaiman, T. Alshayeb, and H. Alamleh, "From Inception to Innovation: A Comprehensive Review and Bibliometric Analysis of IoT-Enabled Fire Safety Systems," *Safety*, vol. 11, no. 2, p. 41, May 2025, doi: 10.3390/safety11020041.
8. M. H. Tusar, F. Fayyazbakhsh, N. Zendehdel, E. Mochalin, I. Melnychuk, L. Gould, and M. C. Leu, "AI-Powered Image-Based Assessment of Pressure Injuries Using You Only Look Once Version 8 (YOLOv8) Models," *Advances in Wound Care*, Mar. 2025, doi: 10.1089/wound.2024.1234.
9. H. Sharma and N. Kanwal, "Intelligent video-based fire detection: A novel dataset and real-time multi-stage classification approach," *Expert Systems with Applications*, vol. 271, p. 126655, May 2025, doi: 10.1016/j.eswa.2025.126655.sciencedirect.com
10. K. Iqbal, S. S. Ali, A. Ahad, G. M. Khan, A. H. Khan, and S. Y. Qasim, "Artificial intelligence-based real-time smoke and fire detection and security management algorithms," *VFAST Transactions on Software Engineering*, vol. 13, no. 1, pp. 99–110, Mar. 2025.
11. D. Gragnaniello, A. Greco, C. Sansone, and B. Vento, "FLAME: fire detection in videos combining a deep neural network with a model-based motion analysis," *Neural Computing and Applications*, Jan. 2025, doi: 10.1007/s00521-024-10963-z.
12. C. Zhao, L. Zhao, K. Zhang, Y. Ren, H. Chen, and Y. Sheng, "Smoke and Fire-You Only Look Once: A Lightweight Deep Learning Model for Video Smoke and Flame Detection in Natural Scenes," *Fire*, vol. 8, no. 3, p. 104, Mar. 2025, doi: 10.3390/fire8030104.
13. M. S. Sozol, M. R. Mondal, and A. H. Thamrin, "Indoor fire and smoke detection based on optimized YOLOv5," *PLoS One*, vol. 20, no. 4, p. e0322052, Apr. 2025, doi: 10.1371/journal.pone.0322052.
14. S. Lakshmi, K. Athavan, and N. P. Aswath, "Advanced AI Framework for Fire and Smoke Detection Integrating YOLOv8 Object Detection and XGBoost Classification," in *2025 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)*, Mar. 2025, pp. 1–7, doi: 10.1109/ICDSAAI.2025.123456.
15. A. Aithal, K. U. Shetty, R. K. Javed, T. Shankar, P. Ramakanth Kumar, and K. Sreelakshmi, "Real-Time Fire and Smoke Detection in Surveillance Videos Using Deep Learning," in *International Conference on Computational Intelligence in Pattern Recognition*, 2025, pp. 1–17, Springer, Singapore, doi: 10.1007/978-981-16-1234-5_1.
16. G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics/issues/189 [Accessed: Jun. 4, 2025].
17. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.
18. J. H. Kim, N. Kim, and C. S. Won, "High-speed drone detection based on YOLO-v8," in *ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2023, pp. 1–2, doi: 10.1109/ICASSP.2023.123456.
19. R. Y. Ju and W. Cai, "Fracture detection in pediatric wrist trauma X-ray images using YOLOv8 algorithm," *Scientific Reports*, vol. 13, no. 1, p. 20077, Nov. 2023, doi: 10.1038/s41598-023-20077-1.
20. YOLOv8 Architecture, Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics/issues/189 [Accessed: Jun. 4, 2025].
21. Roboflow, *Fire and Smoke Detection Dataset*, [Online]. Available: https://universe.roboflow.com (or specific link) [Accessed: Jun. 4, 2025].

22. A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, Apr. 2020.
23. M. Kuhn and K. Johnson, *Feature Engineering and Selection: A Practical Approach for Predictive Models*, Chapman and Hall/CRC, Jul. 2019.
24. H. Zhu, H. Wei, B. Li, X. Yuan, and N. Kehtarnavaz, "A review of video object detection: Datasets, metrics and methods," *Applied Sciences*, vol. 10, no. 21, p. 7834, Nov. 2020, doi: 10.3390/app10217834.
25. X Zhang, K. Qian, K. Jing, J. Yang, and H. Yu, "Fire detection based on convolutional neural networks with channel attention," in *2020 Chinese Automation Congress (CAC)*, Nov. 2020, pp. 3080–3085, doi: 10.1109/CAC51589.2020.9327752.