

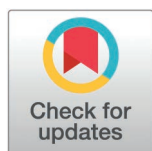
RESEARCH ARTICLE

Indoor fire and smoke detection based on optimized YOLOv5

Md. Shafak Shahriar Sozol^{1*}, M. Rubaiyat Hossain Mondal¹, Achmad Husni Thamrin²

1 Institute of Information and Communication Technology, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, **2** Keio University, Japan.

* shahriarsajal05405@gmail.com



Abstract

Ensuring safety and safeguarding indoor properties require reliable fire detection methods. Traditional detection techniques that use smoke, heat, or fire sensors often fail due to false positives and slow response time. Existing deep learning-based object detectors fall short of improved accuracy in indoor settings and real-time tracking, considering the dynamic nature of fire and smoke. This study aimed to address these challenges in fire and smoke detection in indoor settings. It presents a hyperparameter-optimized YOLOv5 (HPO-YOLOv5) model optimized by a genetic algorithm. To cover all prospective scenarios, we created a novel dataset comprising indoor fire and smoke images. There are 5,000 images in the dataset, split into training, validation, and testing samples at a ratio of 80:10:10. It also used the Grad-CAM technique to provide visual explanations for model predictions, ensuring interpretability and transparency. This research combined YOLOv5 with DeepSORT (which uses deep learning features to improve the tracking of objects over time) to provide real-time monitoring of fire progression. Thus, it allows for the notification of actual fire hazards. With a mean average precision (mAP@0.5) of 92.1%, the HPO-YOLOv5 model outperformed state-of-the-art models, including Faster R-CNN, YOLOv5, YOLOv7 and YOLOv8. The proposed model achieved a 2.4% improvement in mAP@0.5 over the original YOLOv5 baseline model. The research has laid the foundation for future developments in fire hazard detection technology, a system that is dependable and effective in indoor scenarios.

OPEN ACCESS

Citation: Sozol MSS, Mondal MRH, Thamrin AH (2025) Indoor Fire and Smoke detection based on Optimized YOLOv5. PLoS One 20(4): e0322052. <https://doi.org/10.1371/journal.pone.0322052>

Editor: Himadri Majumder, G H Raisoni College of Engineering and Management, Pune, INDIA

Received: October 19, 2024

Accepted: March 15, 2025

Published: April 29, 2025

Copyright: © 2025 Sozol et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data availability statement: Data are available on a public repository given below <https://huggingface.co/datasets/shahriar-5/IFireSmoke/tree/main>

Funding: The author(s) received no specific funding for this work.

1 Introduction

Fire hazards in indoor environments are a serious safety concern because they can cause severe loss of life and property [1–3]. It is important to reduce these risks through early detection and rapid action. Traditional fire detection methods primarily depend on various sensors that detect smoke, heat, or fire [4, 5]. Smoke detectors, often using photoelectric or ionization technologies, are widely deployed in residential and commercial buildings [6]. Heat detectors, which sense a rise in

Competing interests: The authors have declared that no competing interests exist.

temperature, and fire detectors, which detect the light emitted by fires, are also common [7]. Although these methods can be useful in certain scenarios, they often result in high false positive rates and slower response times, especially in larger or more complex environments [8–10]. Additionally, traditional sensors are limited by their reliance on physical proximity to the fire source, making them less effective in large or obstructed spaces. They also struggle to distinguish between actual fire incidents and false alarms triggered by non-fire-related phenomena, such as steam or dust [9,10]. This highlights the pressing need for more refined and efficient fire detection systems. To address this, researchers have started to investigate image-based fire detection approaches, applying advances in computer vision (CV) and cutting-edge image processing techniques. Early approaches used simple color-based algorithms to identify fire by detecting regions with characteristic fire colors such as red, yellow, and orange. However, these methods often struggled with false positives caused by objects of similar colors [11–13]. The introduction of more sophisticated image analysis techniques, including motion detection and texture analysis, improved the reliability of fire detection. These methods analyze the dynamic and structural characteristics of fire, such as the flickering motion of fire and the smooth texture of smoke [14–16]. Despite these improvements, image-based detection approach still faces challenges under varying lighting conditions and complex backgrounds [17]. The potential of CV and AI lies in their ability to address complex, real-world challenges through automation and intelligent decision-making across various fields, including fire safety monitoring [9], construction [18], and agriculture [19, 20]. AI-based deep learning (DL) models for object detection provide a promising approach to tackle these issues in detection systems [21, 22, 23]. There is a lack of indoor fire and smoke detection-based DL models [9,15,21]. Moreover, most available image datasets are outdoor-based or mixed, presenting challenges for indoor-specific model development. These existing datasets emphasized outdoor conditions, which differ significantly from indoor environments in terms of lighting, background complexity, and other variables [21,24–26]. Consequently, they may not provide the well-annotated data required for reliable indoor detection. To address this gap, a novel dataset has been created specifically for indoor fire and smoke detection.

This study makes significant contributions by introducing a novel dataset specifically for indoor fire and smoke detection. An optimized YOLOv5 (HPO-YOLOv5) model was developed through a genetic algorithm to improve detection performance. It ensures higher accuracy and less inference time in indoor environments. The Grad-CAM method is applied to produce visual explanations for the predictions via model, hence enhancing transparency and interpretability. The HPO-YOLOv5 model is evaluated towards other state-of-the-art models, exhibiting better performance. By integrating YOLOv5 with the real-time tracking algorithm DeepSORT enables real-time surveillance of fire progression, providing actual fire hazard alerts.

The key findings of this study are as follows:

- Introduction of a new indoor fire and smoke dataset for accurate detection of fire and smoke in household settings.

- Development of an enhanced YOLOv5 model with optimized hyperparameters using a genetic algorithm, boosting fire and smoke detection performance. Comparative analysis shows our model outperforms Faster R-CNN, YOLOv5s, YOLOv7, and YOLOv8n in indoor fire and smoke detection.
- Application of Grad-CAM technique as explainable AI to generate visual explanations for the model's predictions, ensuring transparency and interpretability.
- Integration of YOLOv5 and DeepSORT enables real-time surveillance of fire development and effective fire hazard alerts.

This paper is structured as follows: Section 2 explores related work on fire and smoke-based object detection. Section 3 outlines the materials and methods, incorporating dataset preparation, and hyperparameter optimization on the model. Section 4 presents the experimental results, along with a detailed discussion of the findings. Lastly, Section 5 concludes the study, summarizing the key findings and future research directions.

2 Related works

Convolutional neural networks (CNNs) have become central to modern object detection, capable of learning complex patterns and features from large datasets [27]. Among the various CNN architectures, including YOLO (You Only Look Once), Faster R-CNN [28], SSD (Single Shot Detector) [29], and EfficientNet [30], YOLO emerged as a highly effective model for real-time fire and smoke detection. YOLO models, including YOLOv3 [31], YOLOv4 [32], YOLOv5 [33], YOLOv6 [34], YOLOv7 [35], and the latest YOLOv8 [36], have been successfully applied to fire detection, showing significant improvements in speed and accuracy over traditional methods. Some of these works can be grouped by the aspect of “explainable AI”, real-time tracking, methods based on YOLO algorithm or based on region-based methods, the use of optimization techniques, etc.

In order to improve fire detection tasks, Chaoxia et al. [37] proposed a Faster R-CNN color-guided anchoring technique to enhance fire detection by using fire color features for anchor placement. The accuracy of fire detection was significantly increased when Wan et al. [38] proposed SSD, which combines feature maps from multiple convolution layers to retain detailed low-level features and abstract high-level semantic information. Dewangan and Gupta [39] developed a soft-voting based deep ensemble model for indoor fire and smoke detection, utilizing four transfer learning models: MobileNetV2, ResNet50V2, EfficientNetB0, and DenseNet121.

Beyond all other object detection models, the YOLO-based model performed significantly better than other SOTA models in many cases [40–45]. Qin et al. [40] applied YOLOv3's multi-scale detection capability, achieving through up-sampling and splicing in certain layers of network prediction. Hu et al. [41] introduced the DS-YOLO model, enhancing YOLOv7 with the DP-ELAN module and SlimNeck to expand the receptive field, improve feature representation, and balance computational efficiency with accuracy for smoke and fire detection. Kong et al. [42] enhanced the fire and smoke real-time detection algorithm for coal mines using an improved YOLOv8s model, focusing on reducing computational complexity and integrating attention mechanisms to achieve a mAP of 91.0%. YOLOv5 [43], a popular successor in the YOLO series, improved on its predecessors with architectural innovations and optimized training, still making it more suitable for fire detection than the newer YOLOv8 in some cases [44]. Yang et al. [46] developed a lightweight YOLOv5s-based network with C3Ghost, Ghost modules, and a CA module for improved fire and smoke detection, enhancing accuracy in complex environments.

Even with the progressions in YOLO models, challenges remained in optimizing their performance for specific tasks such as indoor fire and smoke detection. Hyperparameter optimization is key to enhancing DL model accuracy and robustness. Parameters like learning rate, batch size, and momentum must be carefully tuned for optimal results [47]. Falkner et al. [48] emphasized the necessity of hyperparameter optimization in DL models by showing the strengths of Bayesian Optimization and Hyperband tuner. Manual tuning is often time-consuming and may not yield the best results.

Therefore, automated hyperparameter optimization techniques have gained popularity [49]. Wu et al. [50] highlighted the importance of automated hyperparameter optimization like Bayesian optimization, genetic algorithms, and random search for DL-based side-channel analysis, given the complexity of neural networks' large hyperparameter space. One of the key techniques, Genetic algorithms (GA), are inspired by the process of natural selection and are used to optimize complex problems by iteratively improving a population of candidate solutions [51]. In the context of DL, GA can effectively explore the hyperparameter space, identifying optimal configurations that enhance model performance. Studies [52–55] showed that GA-optimized models outperform those tuned through manual or grid search methods. GA has been applied to optimize hyperparameters in various object detection models, including YOLO. Suhail et al. [56] highlighted their key contribution in using the Evolutionary Genetic Algorithm (EGA) for hyperparameter optimization in YOLOv5, improving the detection of six categories of urine sediment particles from microscopic images. So, GA-based hyperparameter optimization may be highly effective for complex tasks like fire and smoke detection to improve model robustness and reliability. However, current optimized GA models for indoor fire and smoke detection still fall short in recent scenarios.

Moreover, the black-box nature of DL models poses challenges for their adoption in safety-critical applications. Explainable AI (XAI) addresses this issue by providing insights into the model's decision-making process, ensuring transparency and trust [57]. Selvaraju et al. [58] introduced the Gradient-weighted Class Activation Mapping (Grad-CAM) method, which offers visual explanations for the decisions rendered by CNNs. This technique helps in understanding which regions of an image contribute most to the prediction results, thereby enhancing the transparency and trustworthiness of the model. This is particularly useful in fire and smoke detection, where understanding the model's focus can validate its predictions and ensure it responds to actual fire characteristics. However, there is a lack of transparency-checking methods in safety-critical DL models like fire and smoke detection [59].

While recent and previous DL-based object detectors have been successful in detecting fire and smoke instances in images and videos, there is a noticeable gap in research and literature regarding practical approaches for monitoring fire progression in a video to detect actual fire hazard scenarios [60–65]. A review by Jin et al. [60] and Chitram et al. [61] emphasized improvements in image and/or video-based fire detection methods using DL, highlighting the need for improved strategies to monitor and predict fire behavior in real-time. In many cases, the integration of YOLOv5 with real-time tracking algorithms such as DeepSORT [62] has proven highly effective for continuous events monitoring. Zhang Y et al. [63] used YOLOv5 for real-time detection of surgical instruments and then employed DeepSORT to precisely track the movements of those instruments across multiple video frames. Researchers also applied this combined approach to athlete tracking [64], as well as vehicle analysis on expressways [65]. Still, there is no effective approach to tracking fire progression over time in videos to detect and manage fire hazards better.

Despite improvements in DL for fire and smoke detection, several challenges exist. Achieving high detection accuracy and reducing false positives are critical issues, especially in complex indoor environments with varying lighting conditions, occlusions, background noise, and the need to detect actual fire hazards. The availability of diverse and well-annotated indoor datasets is essential for developing effective DL models that can perform reliably across different scenarios. Furthermore, an optimized deep-learning model specifically for indoor fire and smoke detection still needs to be readily available. Additionally, ensuring model transparency and interpretability using techniques like Grad-CAM remains a challenge, as it requires making the decision-making process of the model understandable and accessible to users.

3 Materials and methods

3.1 Dataset creation method

The Indoor Fire and Smoke (Indoor FS) Dataset was designed to aid the development and evaluation of fire and smoke detection systems. To ensure its usefulness in various real-world scenarios, we compiled a diverse collection of images. The images were sourced locally using a Canon PowerShot G16 and an iPhone XS camera, as well as from online platforms (YouTube, Google Images), including CCTV footage. To develop a more well-rounded dataset, real scenes were

combined with synthetic images. Images were maintained in a 60:40 ratio, with 60% captured locally and 40% sourced online and synthesized. This approach showed a diverse range of image quality and perspectives, which is vital for training effective detection models. Annotation and preprocessing of the dataset were conducted using an online tool named Roboflow. Annotations were done carefully, ensuring that bounding boxes were closely enclosed around each object within the images. Each image file was paired with a corresponding.txt file containing annotated information, i.e., object class, coordinates, height, and width.

The Indoor FS dataset follows a hierarchical organization with three main folders: train (for model training), test (for final evaluation), and valid (for interim validation during training). Within each of these folders, there are two subfolders: one for images and another for labels, where the image files (.jpg) are stored separately from their corresponding annotation files (.txt). This structure is illustrated in Fig 1.

Auto-Orient and Resize were the preprocessing steps that were specifically applied for indoor fire and smoke detection dataset. In order to maximize the training process and improve the detection system's performance, these transformations were implemented across all images. This dataset created a solid foundation and resource for developing and evaluating indoor fire and smoke detection systems in the future.

3.2 The proposed optimized YOLOv5

The traditional YOLOv5 architecture and its description have been documented in the literature [43]. This section describes the hyperparameter optimization of the YOLOv5 network structure via GA. The process of optimizing the baseline model is shown in the flowchart (Fig 2). The process starts by generating an initial population of network configurations. Each configuration has a unique set of thirty-two hyperparameters, randomly sampled within predefined ranges based on a seed network setup. These hyperparameters include learning rate, momentum, weight decay, batch size, augmentation settings, etc. After generating the initial population, each model was trained for 150 epochs to ensure stable performance evaluation [66]. Post-training, the models were evaluated using fitness scores where $mAP@0.5$ is the primary evaluation metric. Then, the GA refined the models through selection, crossover, and mutation across multiple generations. The initial population consisted of 20 candidate models ($N=20$) and was selected accordingly. This number was chosen based on empirical studies to balance efficiency and performance. The top 50% of models from each generation were ranked by fitness scores. They were selected for the next generation. This ensured that well-performing models were retained, while underperforming models were discarded. "Crossover" was applied with a probability of 0.8 ($P_c = 0.8$), using a one-point crossover method to combine hyperparameters from two parent models. It created new offspring while maintaining key characteristics of high-performing models. Then, "Mutation" operation was applied with a probability of 0.2 ($P_m = 0.2$) introducing slight random changes in selected hyperparameters. This helped in preventing premature

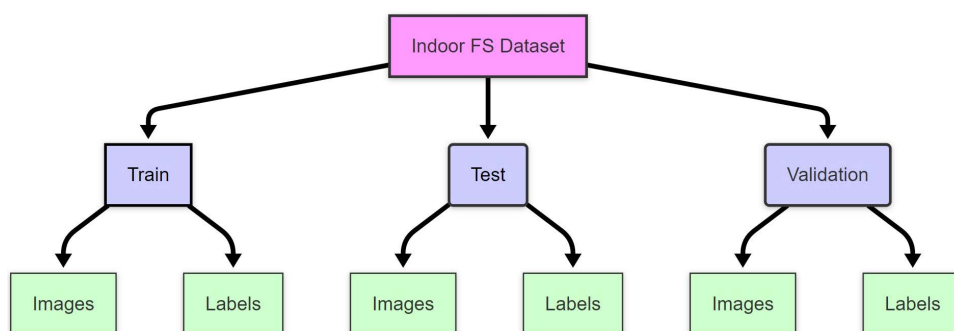


Fig 1. Dataset folder structure.

<https://doi.org/10.1371/journal.pone.0322052.g001>

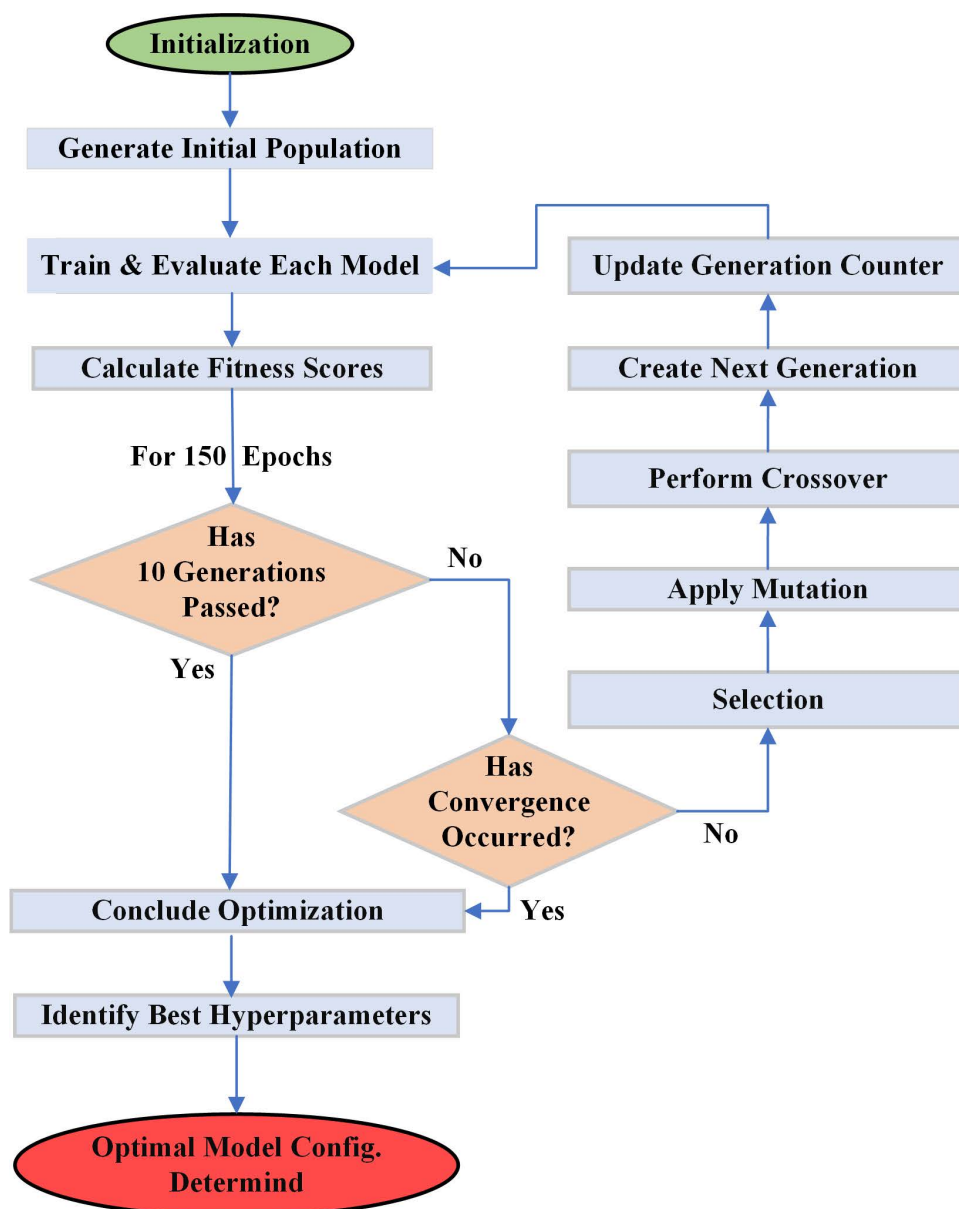


Fig 2. Genetic algorithm for hyperparameter optimization.

<https://doi.org/10.1371/journal.pone.0322052.g002>

convergence and enhanced model diversity. The optimization process went through for a maximum of 10 generations or until convergence, defined as no significant improvement ($\leq 0.5\%$ **mAP@0.5** gain) for three consecutive generations. To validate the GA settings, multiple experiments were conducted with varying population sizes ($N = 10, 20, 30$), crossover probabilities (0.5, 0.8, 1.0), and mutation probabilities ($P_c = 0.1, 0.2, 0.3$). The chosen values ($N = 20, P_c = 0.8, P_m = 0.2$) provided the best tradeoff between convergence speed and accuracy. Increasing the population size beyond 20 resulted in diminishing returns in accuracy while significantly increasing computation time. Similarly, mutation probabilities higher than 0.2 introduced more randomness, leading to unstable training. This systematic hyperparameter tuning approach

allowed the genetic algorithm to effectively explore the hyperparameter space. The result is significant improvements to the detection performance of YOLOv5 model specifically for indoor fire and smoke scenarios.

3.3 Evaluation metrics

To objectively assess the performance of the models, this research employs precision, recall, $mAP@0.5$, and the F1 score. These performance metrics are significant for providing a thorough evaluation of the model's outcomes in various aspects of prediction accuracy and reliability. We adopted the evaluation metrics used in PASCAL VOC [67] to assess our results. The formulas are written as follows (1–5).

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$AP = \int_0^1 P(r) dr \quad (3)$$

$$mAP = \frac{\sum_{i=1}^k AP_i}{k} \quad (4)$$

$$F1 = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \quad (5)$$

Precision (P) measures how accurately a model predicts positive instances, calculated as the ratio of true positives (TP) to the total predicted positives (TP + FP), expressed as in Eqn. (1). Recall (R) assesses the model's ability to detect all relevant positive instances, calculated as the ratio of true positives to the total actual positives (TP + FN) (Eqn. (2)). Average Precision (AP) is the area under the Precision-Recall curve for a specific class, summarizing the trade-off between precision and Recall. Mean Average Precision (mAP) extends AP to multiple classes by averaging AP values across classes (Eqn. (3)). Specifically, $mAP@0.5$ refers to the mean average precision calculated with an Intersection over Union (IoU) threshold of 0.5, meaning a predicted bounding box is considered a true positive if it overlaps with the ground truth by at least 50%. A higher IoU threshold (e.g., 0.75) requires more precise predictions to be considered correct, while a lower threshold (e.g., 0.5) allows for more flexibility. A more lenient threshold (e.g., 0.5) was sufficient for broader detection tasks, where detecting the presence of a fire or smoke was more important than pinpointing the exact location.

The F1-score balances precision and Recall, particularly useful for imbalanced classes (Eqn. (5)). In these formulas, TP (true positives) are correctly predicted positive samples, TN (true negatives) is correctly predicted negative samples, FP (false positives) are negatives misclassified as positives, and FN (false negatives) are positives misclassified as negatives. The Precision-Recall curve (P(r)) showed the trade-off between precision and recall at different thresholds, and k represents the number of classes. By utilizing these metrics, this study ensures a thorough and balanced evaluation of the performance of the model.

3.4 Explainable AI Methods for Model Transparency: Grad-CAM

Explainable artificial intelligence (XAI) seeks to make the internal structure of DL models transparent and accessible to humans. Grad-CAM is an increasingly used XAI technique for visualizing and interpreting DL models, particularly CNNs. This study used the Grad-CAM method to guarantee the transparency and interpretability of the HPO-YOLOv5

model. Visualizing and understanding the decisions taken by CNNs becomes much easier using this approach. Using the gradient information that reached the last convolutional layer of the CNN, this approach gives each neuron importance ratings for a specific decision of interest. This clarified which areas of an image were crucial for predicting an object via a model. Fig 3. shows the Grad-CAM technique in detail by block. The penultimate convolutional layer (layer = -2) of the model explains the decisions made at the output layer. It starts with an input image that the model is supposed to classify or analyze. For instance, the image might depict a scene of fire, which the model needs to detect. The input image is then passed through the backbone of a CNN, which consists of multiple layers of convolutional filters. These layers transform the input image into high-level feature maps, capturing various aspects and patterns present in the image. The convolutional layers produce activation maps (A_k), which are intermediate outputs that represent different features detected in the input image. Each activation map highlights specific features such as edges, textures, or shapes. Grad-CAM requires the computation of gradients of the target class score with respect to the activation maps. This is done through backpropagation from a particular layer (often the last convolutional layer) to compute the gradients, $(\frac{dy^c}{dA_k})$ where Y^c is the score for class c . The gradients are globally averaged over the spatial dimensions (width and height) to obtain the important weights (a_k^c) these weights indicate the importance of each feature map in the final decision for the target class [58].

$$a_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_k^j} \quad (6)$$

The importance weight (a_k^c), is given by Eqn. (6), are used to weigh the corresponding activation maps, where Z is the number of pixels in the activation map and are the spatial locations. These weighted activation maps are then summed to produce a coarse localization map, $L_{Grad-CAM}^c$, that highlights the regions in the image relevant to the target class [58].

$$L_{Grad-CAM}^c = ReLU (\sum_k a_k^c A_k) \quad (7)$$

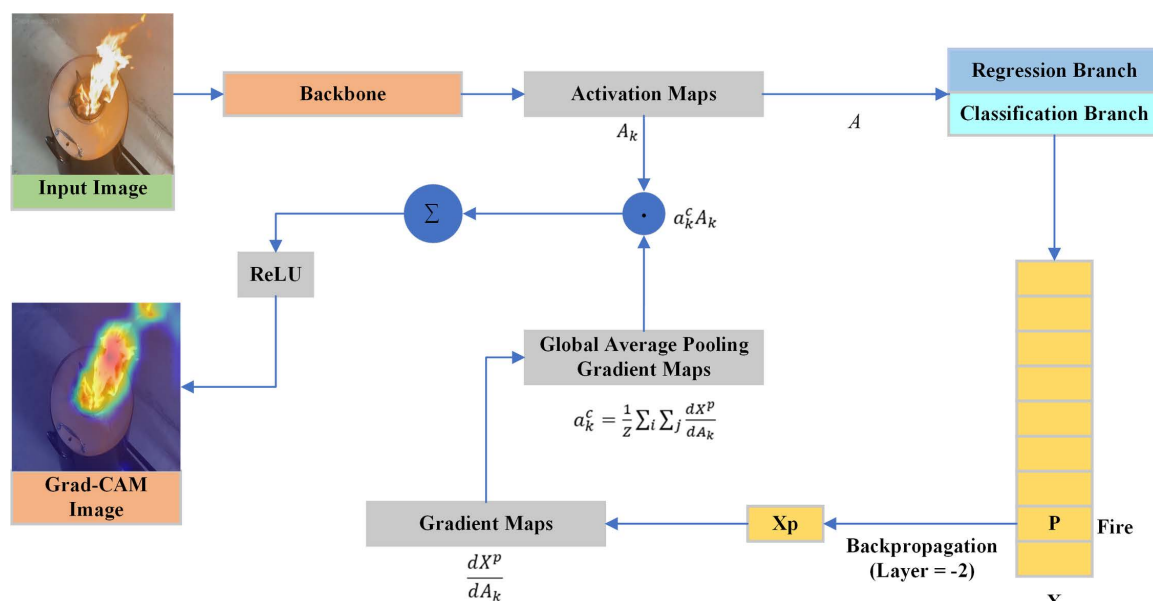


Fig 3. XAI (Grad-Cam) architecture.

<https://doi.org/10.1371/journal.pone.0322052.g003>

A ReLU (Rectified Linear Unit) activation function is applied to the coarse localization map to retain positive contributions, focusing on regions that influence the target class prediction. The resulting Grad-CAM heatmap highlights key parts of the input image that are important to the model's decision, providing a visual explanation. This is crucial for identifying which image sections the model focuses on when predicting smoke and fire. In the case of fire detection, the Grad-CAM heatmap should indicate the fire regions, indicating that the model's focus is consistent with human intuition. Similarly, for smoke detection, the heatmap should include areas with visible smoke. The adoption of Grad-CAM in our proposed model improves interpretability by providing insights into the decision-making process of the model. This transparency is critical for applying the model in real-world safety and monitoring systems because understanding the foundation of its predictions can increase trust and allow for more effective responses.

3.5 Concept of Fire hazard detection

Detecting fire hazards accurately is crucial because traditional detection methods can sometimes produce false positives, causing false alarms. For instance, objects with similar visual characteristics to fire, such as a yellow t-shirt, might be incorrectly identified as a fire. Similarly, small, controlled fires, like those from a candle, may not represent a significant danger. Therefore, to reliably detect actual fire hazards, it is necessary to monitor the progression of the fire over time. By monitoring how a fire grew or changed over time, the system dynamically assessed and differentiated between non-threatening situations and actual fire hazards. Continuous tracking allows for real-time updates, ensuring that safety measures are activated, when necessary, thus reducing unnecessary alarms. This section outlines an overall approach that utilizes state-of-the-art object detection and tracking algorithms to infer potential fire hazards from video frames. The methodology involves the use of YOLOv5 for object detection and DeepSORT for object tracking, followed by comparisons to identify significant changes in fire size.

In Fig 4, a real-world video with a duration of approximately 4 minutes was used, and a 60-second segment was selected for analysis. This segment was sampled at a rate of 1 frame per second (fps), yielding a total of 60 frames. The original video, recorded at 30 frames per second, was downsampled to 1 fps using OpenCV [66] for further processing. The methodology began by detecting and inferring potential fire hazards in the video captured at 1 fps over the 60-second interval. First, the YOLOv5 model detected objects in each video frame and outputted bounding boxes (B) around the fire instance. We only considered here a single fire instance (T_1). The DeepSORT algorithm tracked T_1 across the frames, keeping a consistent identification of the fire instance. In the first frame, the initial bounding box area (A_1) was calculated, which served as the baseline for further comparisons. A running sum (S_A) is then initialized to track the total area across frames. From frame 2–60, the bounding box area (A_i) is calculated for each frame. This value was added to the running sum (SA). All 60 frames were processed, the average bounding box area ($\bar{A}$) was computed using the formula, $\bar{A} = S_A / t - 1$, to efficiently calculate the average. In the last frame (Frame no. 60), the total bounding box area (B_{total}) was determined. The average area for the last frame (B) was also calculated. The algorithm then compared the initial area (A_1) to $\bar{A}$ and B.



Fig 4. Fire monitoring process.

<https://doi.org/10.1371/journal.pone.0322052.g004>

If either $\bar{A} - A_1$ or $B - A_1$ exceeds a set threshold (U), a fire hazard incident was detected ($H = \text{True}$). If not, no significant fire growth was observed ($H = \text{False}$). Finally, the algorithm outputs the status of the fire hazard inference (H).

Pseudo Algorithm for Fire Hazard Inference (1 fps, Single Instance)

A. Object Detection (YOLOv5):

- **Input:** Video frames, F (1 fps)
- **Output:** Detected bounding boxes, B
- **Process:** $B = \text{YOLOv5}(F)$

B. Object Tracking (DeepSORT):

- **Input:** Detected bounding boxes, B
- **Output:** Tracked fire instance, T (T_1)
- **Process:** $T = \text{DeepSORT}(B)$

C. Initial Area Calculation (Frame 1):

- **Input:** Tracked fire instance, T_1
- **Output:** Initial bounding box area, A_1
- **Process:**
 - $A_1 = \text{Area}(B_{\text{initial}}(T_1))$
 - Initialize $S_A = 0$ (Running sum for instance T_1)

D. Monitoring (Frame 2 to 60):

- **Input:** Tracked fire instance T , Time interval $\Delta t = 60$ s
- **Output:** Average area $\bar{A}$ for T_1
- **Process:**
 - For each frame $t = 2$ to 60 (1 fps):
 - Calculate area for T_1 at time t :
 - $A_t = \text{Area}(T_{1,t})$
 - Update running sum:
 - $S_A = S_A + A_t$
 - After 60 frames, compute the average area:
 - $\bar{A} = S_A / (t - 1)$ (Efficient mean calculation)

E. Last Frame Area (Frame 60):

- **Input:** Last frame F_{last} , Tracked instance T_1
- **Output:** Total area B_{total} and average area $\bar{B}$
- **Process:**

- $B_{total} = \text{Area}(F_{last}(T_1))$
- $B = B_{total}$

F. Comparison:

- **Input:** A_1, A, B, U
- **Output:** H
- **Process:**
 - $H = \begin{cases} \text{True, if } (A - A_1 > U) \text{ or } (B - A_1 > U) \\ \text{False, otherwise} \end{cases}$

G. Final Output:

- Fire hazard inference H .

This approach combines object detection, tracking, and dynamic area calculations to develop a strong framework for real-time fire hazard detection.

4 Experimental results and discussion

4.1 Dataset description

The Indoor Fire and Smoke Dataset is a carefully labelled collection of images with a median resolution of 416 by 416 pixels. The dataset distribution comprises a total of 5,000 images and 5,931 instances, with a split of 80% for training, 10% for validation, and 10% for testing, as elaborated in [Table 1](#). Since an image may have more than one instance, the total image samples (*) have not been calculated.

To analyze whether the dataset utilized in this research corresponded to the distribution of fire and smoke instances in indoor environments, a statistical analysis of the dataset labels was implemented, and the outcomes were presented in [Fig 5](#). The bar chart suggests that the number of 'Fire' instances is greater than that of 'Smoke'. The central region of the image exhibits a higher density in the distribution of bounding box sizes. The scatter plots revealed that most labels were centered, with a concentration of the ratio of horizontal and vertical coordinates in the center of the labels. Furthermore, the label width to image width and label height to image height ratios indicate that the majority of label frames are of comparable dimensions. This broad distribution, which was primarily concentrated in the center of the image, suggested that the dataset contained small to large-sized objects, which were consistent with actual indoor fire and smoke scenarios.

4.2 Experiment platform

The training of the proposed model was conducted using an image size of 416 by 416 pixels, a batch size of 8, and over 100 epochs. For training, the experimental environment settings are described in [Table 2](#).

Table 1. Dataset distribution by class.

Class	Number of Images				Number of Instances			
	Training	Validation	Testing	Total	Training	Validation	Testing	Total
Fire	2009	290	221	2520	2571	299	251	3121
Smoke	2195	236	279	2710	2290	241	279	2810
Total	4204	526	500	*	4861	540	530	5931

<https://doi.org/10.1371/journal.pone.0322052.t001>

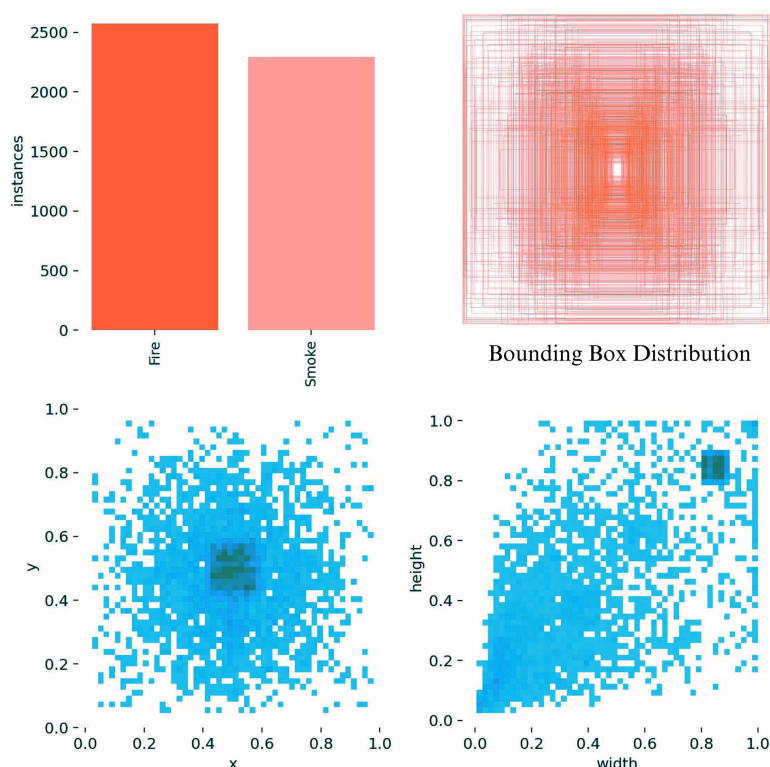


Fig 5. Statistical analysis of indoor fire and smoke dataset labels.

<https://doi.org/10.1371/journal.pone.0322052.g005>

Table 2. The experimental environment settings.

Items	Type
Operating System	Ubuntu 18.04.3
Deployment environment	Python 3.10.12
Memory	16GB of HBM2
CPU	Intel(R) Xeon(R) CPU @ 2.20GHz
GPU	Tesla V100 GPU
DL framework	PyTorch 2.1.0
Accelerated computing architecture	CUDA 12.1

<https://doi.org/10.1371/journal.pone.0322052.t002>

4.3 Hyperparameter results

In this study, the baseline YOLOv5 model was subjected to significant hyperparameter tuning using a genetic algorithm approach in order to optimize its performance for indoor fire and detection tasks. [Table 3](#) summarizes the key hyperparameters adjusted during our experiments, along with their key identifiers, descriptions, and magnitudes.

4.4 Analysis of HPO-YOLOv5 experimental results

Post-training, the model comprised 157 layers, 7,015,519 parameters, and zero gradients, completing the training process in 2.401 hours. The outcomes showed significant performance improvement, as seen in [Table 4](#).

Table 3. Hyperparameters characteristics.

Parameter	Identifier	Description	Magnitude
<i>Model Hyperparameters</i>			
Initial Learning Rate	lr0	Starting learning rate	0.01
Final Learning Rate	lrf	Ending learning rate	0.01
Momentum	momentum	Gradient acceleration	0.98
Weight Decay	weight_decay	Regularization penalty	0.00051
<i>Training Hyperparameters</i>			
Warmup Epochs	warmup_epochs	Initial LR ramp-up period	2.6604
Warmup Momentum	warmup_momentum	Initial momentum adjustment	0.8
Warmup Bias Learning Rate	warmup_bias_lr	Bias LR during warmup	0.09617
Box Loss Weight	box	Box regression weight	0.04702
Class Loss Weight	cls	Classification loss weight	0.39525
Class Prediction Weight	cls_pw	Classification loss modifier	1.0
Objectness Loss Weight	obj	Objectness loss weight	1.187
Objectness Prediction Weight	obj_pw	Objectness loss modifier	1.0968
IOU Threshold	iou_t	IoU detection threshold	0.2
Anchor Threshold	anchor_t	Anchor matching threshold	4.0
<i>Augmentation Hyperparameters</i>			
Hue Adjustment	hsv_h	Hue shift	0.01717
Saturation Adjustment	hsv_s	Saturation shift	0.55958
Value Adjustment	hsv_v	Brightness shift	0.43654
Translation	translate	Image translation	0.0974
Scaling	scale	Image scaling	0.45796
Flip Left-Right Probability	flip_lr	Horizontal flip chance	0.5
Mosaic	mosaic	Mosaic augmentation	0.973

<https://doi.org/10.1371/journal.pone.0322052.t003>

Table 4. Proposed model performance by class.

Class	P (%)	R (%)	F1 (%)	mAP@0.5 (%)
All	92.7	85.5	88.78	92.1
Fire	88.0	88.1	87.95	90.6
Smoke	97.4	83.0	89.47	93.6

<https://doi.org/10.1371/journal.pone.0322052.t004>

The training process of the HPO-YOLOv5 model is depicted through a series of loss and performance metric curves, which highlight the learning dynamics and efficacy of the model. As shown in Fig 6, the training and validation losses for box regression, objectness, and classification steadily decrease, indicating effective learning and convergence over the epochs. Post-training evaluation metrics display significant improvements. The Precision-Confidence Curve (Fig 7(a)) shows high precision across various confidence levels, with the smoke class achieving higher precision than the fire class. The Recall-Confidence Curve (Fig 7(b)) illustrates that recall remains high across different confidence thresholds, though the fire class has slightly lower recall compared to the smoke class. The F1-Confidence Curve (Fig 7(c)) balances precision and recall, showing the optimal confidence threshold for peak performance. Finally, the Precision-Recall (P-R) Curve (Fig 7(d)) displays the ability of the model to maintain high precision and recall across various thresholds, with the smoke

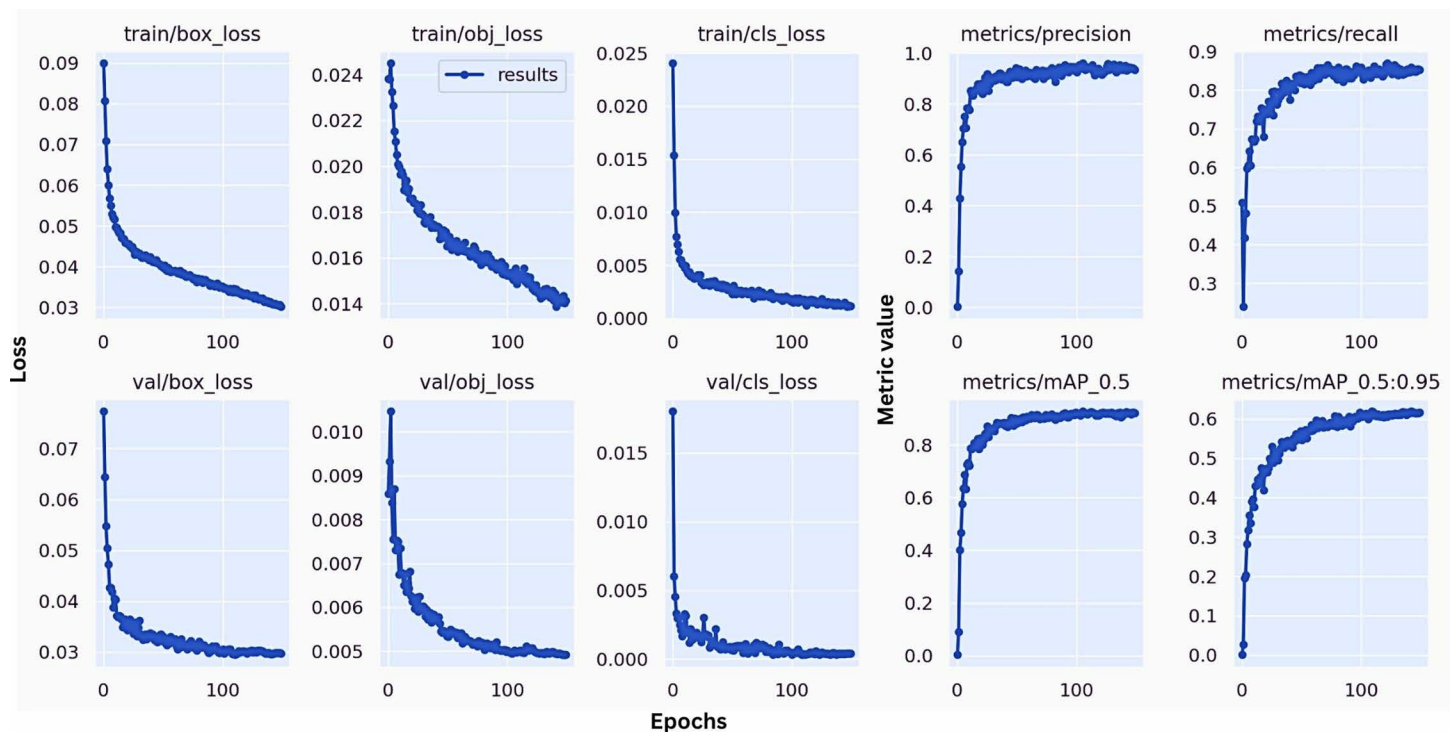


Fig 6. Improved YOLOv5s training results.

<https://doi.org/10.1371/journal.pone.0322052.g006>

class generally outperforming the fire class. These curves collectively validate the robustness and accuracy of our HPO-YOLOv5 model in detecting fire and smoke, highlighting its potential for reliable indoor safety and monitoring applications.

The improved model HPO-YOLOv5 was evaluated on several unseen test images to assess its detection capabilities for fire and smoke in various scenarios (Fig 8). The model successfully detected a bonfire in a dark environment with a high confidence score of 90% (Fig 8(a)). It effectively identified smoke in an indoor setting with a confidence score of 95% (Fig 8(b)), presenting its ability to detect smoke accurately within confined spaces. In a complex scenario involving a fire outbreak, the model demonstrated multiple detections with varying confidence scores for both fire (89%, 78%) and smoke (91%, 84%, 53%) ((Fig 8(c)), highlighting its robustness in multi-object detection scenarios. Additionally, the model identified a small candle fire with a confidence score of 89% ((Fig 8(d)), indicating its sensitivity to detecting small-scale fires. These results indicate the proposed model's capability to identify fire and smoke across different surroundings and conditions.

4.5 Analysis of experimental results of different models

The state-of-the-art (SOTA) comparison experiment was executed to assess and compare the performance of various leading object detection models, including Faster R-CNN, YOLOv5s, YOLOv7, YOLOv8n, and our HPO-YOLOv5. The performance measures analyzed were precision (P), recall (R), F1 score, mean average precision (mAP@0.5) at an IoU threshold of 0.50, and inference time (in milliseconds). The results are summarized in Table 5. To maintain fairness, all networks underwent an identical fine-tuning process during the experiment. We used an image resolution of 416 by 416, trained for 100 epochs with a batch size of 8, employed the SGD optimizer, set a patience value of 100, and a learning rate of 0.01. To reduce the impact of software and hardware on model inference time, the experiments were conducted in a controlled setup detailed in Table 2.

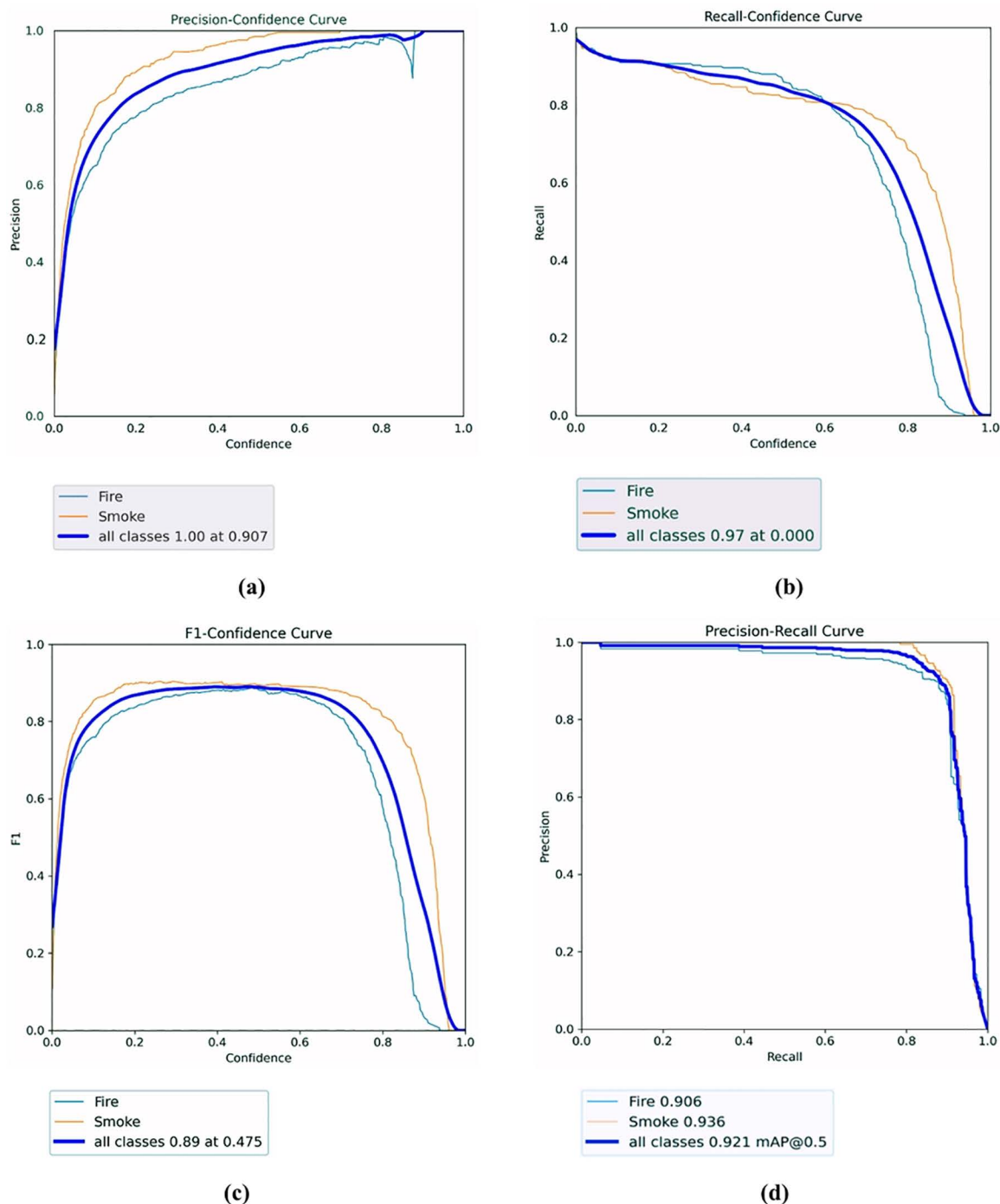


Fig 7. Training progression curves: (a) Precision; (b) Recall; (c) F1; (d) P-R.

<https://doi.org/10.1371/journal.pone.0322052.g007>

Faster R-CNN offered high precision at 91.4%, but it fell short in recall (78.0%) and had a slower inference time of 9.9ms per instance. This highlighted a trade-off between accuracy and speed. On the other hand, YOLOv5s improved the detection with a recall of 82.9%, an F1 score of 87.3%, and an inference time of 6.5ms, making it more efficient for real-time scenarios.

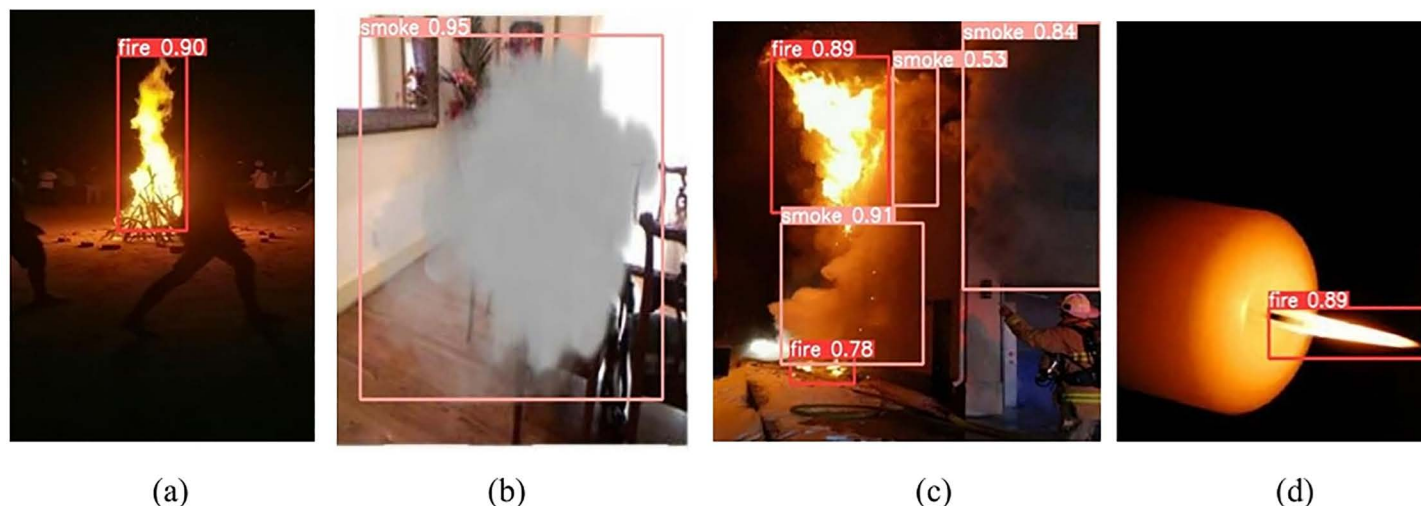


Fig 8. Performance in varied settings: (a) Night environment; (b) Confined spaces; (c) Complex scenario; (d) Dark environment.

<https://doi.org/10.1371/journal.pone.0322052.g008>

Table 5. Comparative Analysis of SOTA object detection model performance metrics.

Methods	P (%)	R (%)	F1 (%)	mAP@0.5 (%)	Inference time (ms)
Faster R-CNN	91.4	78.0	84.1	86.2	9.9
YOLOv5s	92.8	82.9	87.3	89.7	6.5
YOLOv7	89.3	81.6	85.3	88.0	7.3
YOLOv8n	91.9	83.1	87.5	90.1	6.3
Our HPO-YOLOv5	92.7	85.5	88.9	92.1	6.2

<https://doi.org/10.1371/journal.pone.0322052.t005>

YOLOv7 provided balanced performance with a precision of 89.3%, an F1 score of 85.3%, and a moderate inference time of 7.3ms, though it did not quite match its successors. YOLOv8n performed strongly with precision and recall figures of 91.9% and 83.1%, respectively, an F1 score of 87.5%, and a quicker inference time of 6.3ms, showcasing its robust efficiency. However, the standout performer was the HPO-YOLOv5 model. Through hyperparameter optimization, it achieved the highest recall (85.5%), the best F1 score (88.9%), and the top mAP@0.5 at 92.1%, with an inference time of just 6.2ms.

The comparative analysis underscores the proficiency of the HPO-YOLOv5 model in object detection tasks, particularly for indoor fire and smoke detection. Its good performance in precision, recall, F1 score, and mAP, combined with low inference time, positions it as a highly effective solution for real-time safety and monitoring applications.

4.6 Model explainability

To explain the effectiveness of the Grad-CAM method, we present several outputs that highlight the regions in the input images on which the model focused for its predictions.

In the first pair of Fig 9(a), we observe the original input image of a fire and its related Grad-CAM heatmap. The input image shows a visible fire, and the Grad-CAM heatmap is superimposed on the original image, indicating the locations that the model used to make its prediction. The heatmap clearly emphasizes the fire zone, indicating that the model's attention is correctly focused on the relevant features in the image. The highlighted regions in the Grad-CAM heatmap correspond precisely to the fire, providing the model's ability to detect actual fire objects. In the second pair of Fig 9(b), we analyze the original input image of smoke and its related Grad-CAM heatmap. The input image displays visible smoke

in a kitchen environment, and the Grad-CAM heatmap highlights the areas with visible smoke. The heatmap shows the model's focus on the smoke regions, indicating its ability to correctly identify and prioritize these features for prediction. The Grad-CAM heatmap highlights the regions containing smoke objects, validating that the model effectively recognizes and focuses on the smoke regions within the image.

The aforementioned cases show that the HPO-YOLOv5 effectively focuses on relevant features, like fire in fire detection and smoke in smoke detection.

4.7 Fire hazard experimental outputs

The experimental methodology for fire hazard detection starts with the initialization phase, where object detection is carried out using YOLOv5 to identify fire instances in video frames, and object tracking is managed using DeepSORT to provide consistent IDs for these instances across frames. For further frame-based analysis, OpenCV (Open-Source Computer Vision Library) [68] is used here.

The final output, shown in Fig 10, follows the fire hazard detection methodology across video frames. It presents the initial bounding box area from frame 1, the average area over frames 2–60, the last frame's area, and the fire hazard



Fig 9. Grad-CAM heatmap detection: (a) Fire; (b)Smoke.

<https://doi.org/10.1371/journal.pone.0322052.g009>

```

-----
|   Fire Hazard Detection Output   |
-----
| Initial Frame Area: 600          |
| Average Area (60s): 818.67      |
| Last Frame Area: 13064          |
-----
| Comparison:                     |
| A_bar - A_1 = 218.67            |
| B_bar - A_1 = 12464             |
-----
| Fire Hazard Inference: TRUE     |
| Status: Potential fire hazard detected |
-----

```

Fig 10. Fire hazard detection output.

<https://doi.org/10.1371/journal.pone.0322052.g010>

inference. In here, the initial area is 600 square pixels, the average area is 818.67 pixels, and the last frame area is 13,064 pixels. These values are compared to a threshold ($U = 1000$) to assess fire growth. The system calculates the average bounding box area over 59 frames and compares it to the initial area. Although the average difference (218.67) does not exceed the threshold, the difference in the last frame (12,464) far exceeds the threshold, indicating Fire Hazard Inference: True and potential fire hazard detected status.

Threshold (U) We have historical data via experiments showing that typical non-hazardous fire instances have bounding box areas between 500 and 1500 square pixels. Bounding box areas have been observed to increase by 2000–5000 square pixels during hazardous conditions. Primary, a threshold of 1000 is set, as it is above normal fluctuations (0–400) and aligns with significant fire growth patterns. This threshold balances sensitivity (detecting real hazards) and specificity (avoiding false alarms). If the system generates false alarms, the threshold can be increased, and if it misses hazards, it can be lowered later. This ensures the detection system is practical and reliable. The threshold can be adjusted to fine-tune sensitivity and prevent errors.

5 Conclusion and future works

This study focuses on indoor fire and smoke detection by developing an HPO-YOLOv5 model using a genetic algorithm and introducing a novel indoor fire and smoke dataset. It addresses the limitations of traditional fire detection methods by image-based techniques, resulting in high $mAP@0.5$ (92.1%) and recall (85.5%). The integration of the Grad-CAM technique ensures model transparency and interpretability, while the combination of YOLOv5 with DeepSORT assists real-time monitoring of fire progression, providing early fire hazard warnings. Comparative analysis with SOTA models, including Faster R-CNN with Detectron2, YOLOv5s, and YOLOv7, shows that the HPO-YOLOv5 model has more reliable performance than others in terms of accuracy and robustness. Future work may focus on expanding the indoor fire-smoke dataset, exploring advanced model architectures, implementing real-world deployments, enhancing explainability, and predicting fire hazard detection in more real use cases. DeepSORT may struggle with dynamic fire changes, such as rapid alterations in fire shape and size, leading to tracking failures. These challenges can be mitigated by using more sophisticated tracking algorithms or supplementary sensors in the future. These efforts aim to improve the detection system performance, enhance practical use, and increase safety in indoor environments.

Acknowledgments

This research work is part of the MSc (ICT) thesis of the author Md. Shafak Shahriar Sozol under the supervision of the author M. Rubaiyat Hossain Mondal at the Institute of Information and Communication Technology (IICT) of Bangladesh University of Engineering and Technology (BUET), Bangladesh. Hence, the authors would like to thank IICT, BUET for its support.

Author contributions

Conceptualization: Md. Shafak Shahriar Sozol, M. Rubaiyat Hossain Mondal, Achmad Husni Thamrin.

Data curation: Md. Shafak Shahriar Sozol.

Formal analysis: Md. Shafak Shahriar Sozol.

Investigation: Md. Shafak Shahriar Sozol.

Methodology: Md. Shafak Shahriar Sozol.

Project administration: M. Rubaiyat Hossain Mondal, Achmad Husni Thamrin.

Software: Md. Shafak Shahriar Sozol.

Supervision: M. Rubaiyat Hossain Mondal.

Validation: Md. Shafak Shahriar Sozol.

Visualization: Md. Shafak Shahriar Sozol.

Writing – original draft: Md. Shafak Shahriar Sozol.

Writing – review & editing: M. Rubaiyat Hossain Mondal, Achmad Husni Thamrin.

References

- Ivanov M, Chow W. Fire safety in modern indoor and built environment. *Indoor and Built Environment*. 2023;32(1):3–8.
- Kodur V, Kumar P, Rafi M. Fire hazard in buildings: review, assessment and strategies for improving fire safety. *PSU Research Review*. 2020;4(1):1–23.
- Kohl L, Meng M, de Vera J, Bergquist B, Cooke CA, Hustins S, et al. Limited retention of wildfire-derived PAHs and trace elements in indoor environments. *Geophysical Research Letters*. 2019;46(1):383–91.
- Khan F, Xu Z, Sun J, Khan FM, Ahmed A, Zhao Y. Recent Advances in Sensors for Fire Detection. *Sensors (Basel)*. 2022;22(9):3310. <https://doi.org/10.3390/s22093310> PMID: 35590999
- Ghali R, Jmal M, Soudiene Mseddi W, Attia R. Recent advances in fire detection and monitoring systems: A review. *Proceedings of the 8th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT'18)*. Springer International Publishing. 2020;1:332–40.
- Moinuddin K, Bruck D, Shi L. An experimental study on timely activation of smoke alarms and their effective notification in typical residential buildings. *Fire Safety Journal*. 2017;93(1):1–1.
- Mehta BR, Reddy YJ. Fire and gas detection system. In: Mehta BR, Reddy YJ, editors. *Industrial Process Automation Systems*. Butterworth-Heinemann. 2015:217–35. <https://doi.org/10.1016/b978-0-12-800939-0.00005-x>
- Yu M, Yuan H, Li K, Wang J. Research on multi-detector real-time fire alarm technology based on signal similarity. *Fire safety journal*. 2023;136(1):103724. <https://doi.org/10.1016/j.firesaf.2023.103724>
- Bu F, Gharajeh MS. Intelligent and vision-based fire detection systems: A survey. *Image and Vision Computing*. 2019;91:103803. <https://doi.org/10.1016/j.imavis.2019.08.007>
- Gaur A, Singh A, Kumar A, Kulkarni K, Lala S, Kapoor K, et al. Fire sensing technologies: A review. *IEEE Sensors Journal*. 2019;19(9):3191–202.
- Sharma A, Singh PK, Kumar Y. An integrated fire detection system using IoT and image processing technique for smart cities. *Sustainable Cities and Society*. 2020;61:102332. <https://doi.org/10.1016/j.scs.2020.102332>
- Celik T. Fast and efficient method for fire detection using image processing. *ETRI J*. 2010;32(6):881–90. <https://doi.org/10.4218/etrij.10.0109.0695>
- Verma VKr, Jain K, Kumari J, Kumar R, Kumar U. Image processing-based fire detection and protection system using OPENCV. *2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET)*. 2023:296–301. <https://doi.org/10.1109/icseiet58677.2023.10303598>
- Patel P, Tiwari S. Fire detection using image processing techniques. *International Journal of Computer Applications*. 2012;58(18):XX–YY.
- Wu S, Sheng B, Fu G, Zhang D, Jian Y. Multiscale fire image detection method based on CNN and Transformer. *Multimed Tools Appl*. 2023;83(16):49787–811. <https://doi.org/10.1007/s11042-023-17482-4>
- Chunyu Y, Yongming Z, Jun F, Jinjun W. Texture analysis of smoke for real-time fire detection. *2009 Second International Workshop on Computer Science and Engineering*. 2009;2:511–5.
- Prema EC, Vinsley SS, Suresh S. Efficient fire detection based on static and dynamic texture analysis in forest fire detection. *Fire Technology*. 2018;54(1):255–88.
- Hu K, Chen Z, Kang H, Tang Y. 3D vision technologies for a self-developed structural external crack damage recognition robot. *Automation in Construction*. 2024;159:105262. <https://doi.org/10.1016/j.autcon.2023.105262>
- Chen M, Chen Z, Luo L, Tang Y, Cheng J, Wei H, et al. Dynamic visual servo control methods for continuous operation of a fruit harvesting robot working throughout an orchard. *Computers and Electronics in Agriculture*. 2024;219:108774.
- Wang C, Li C, Han Q, Wu F, Zou X. A Performance analysis of a litchi picking robot system for actively removing obstructions, using an artificial intelligence algorithm. *Agronomy*. 2023;13(11):2795. <https://doi.org/10.3390/agronomy13112795>
- Ryu J, Kwak D. A Study on flame and smoke detection algorithm using convolutional neural network based on deep learning. *J Korean Soc Hazard Mitig*. 2020;20(1):223–32. <https://doi.org/10.9798/kosham.2020.20.1.223>
- Balaji G, Mary S, Mantravadi N, Shajin F. Graph CNN-ResNet-CSOA transfer learning architype for an enhanced skin cancer detection and classification scheme in medical image processing. *International Journal on Artificial Intelligence Tools*. 2024;33(02):2350063.
- Venkatesan R, Balaji G. Balancing composite motion optimization using R-ERNN with plant disease. *Applied Soft Computing*. 2024;154:111288.
- Wang M, Jiang L, Yue P, Yu D, Tuo T. FASDD: an open-access 100,000-level fire and smoke detection dataset for deep learning in fire detection. *Earth System Science Data Discussions*. 2023 Mar 15; 2023:1–26.

25. Namozov A, Cho YI. An efficient deep learning algorithm for fire and smoke detection with limited data. *Adv Electr Comp Eng*. 2018;18(4):121–8. <https://doi.org/10.4316/aece.2018.04015>
26. Dewangan A, Pande Y, Braun H-W, Vernon F, Perez I, Altintas I, et al. FlgLib & smokeynet: dataset and deep learning model for real-time wildland fire smoke detection. *Remote Sensing*. 2022;14(4):1007. <https://doi.org/10.3390/rs14041007>
27. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, et al. Recent advances in convolutional neural networks. *Pattern Recognition*. 2018;77:354–77.
28. Girshick R. Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*. 2015:1440–8.
29. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. Ssd: Single shot multibox detector. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I* 14. Springer International Publishing. 2016:21–37.
30. Koonce B. EfficientNet. *Convolutional neural networks with swift for Tensorflow: image recognition and dataset categorization*. 2021:109–23.
31. Redmon J, Farhadi A. Yolo v3: an incremental improvement. *arXiv preprint*. 2018. <https://doi.org/10.48550/arXiv.1804.02767>
32. Bochkovskiy A, Wang C, Liao H. Yolo v4: optimal speed and accuracy of object detection. *arXiv preprint*. 2020. <https://doi.org/10.48550/arXiv.2004.10934>
33. Jiang P, Ergu D, Liu F, Cai Y, Ma B. A review of Yolo algorithm developments. *Procedia Computer Science*. 2022;199:1066–73.
34. Li C, Li L, Jiang H, Weng K, Geng Y, Li L, et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*. 2022 Sep 7.
35. Wang C-Y, Bochkovskiy A, Liao H-YM. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2023:7464–75. <https://doi.org/10.1109/cvpr52729.2023.00721>
36. Sohan M, Sai Ram T, Reddy R, Venkata C. A review on Yolo v8 and its advancements. *Proceedings of the International Conference on Data Intelligence and Cognitive Informatics 2024*. Springer: Singapore. 2024:529–45.
37. Chaoxia C, Shang W, Zhang F. Information-guided fire detection based on faster R-CNN. *IEEE Access*. 2020;8:58923–32.
38. Wan Z, Zhuo Y, Jiang H, Tao J, Qian H, Xiang W, et al. Fire detection from images based on single shot multibox detector. *Proceedings of the 10th International Conference on Computer Engineering and Networks*. Springer: Singapore. 2021:302–13.
39. Dewangan DK, Gupta GP. Indoor Fire and Smoke Detection Using Soft-Voting Based Deep Ensemble Model. 2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT). 2024:692–7. <https://doi.org/10.1109/csnt60213.2024.10545933>
40. Qin Y-Y, Cao J-T, Ji X-F. Fire Detection Method Based on Depthwise Separable Convolution and YOLOv3. *Int J Autom Comput*. 2021;18(2):300–10. <https://doi.org/10.1007/s11633-020-1269-5>
41. Hu J, He Y, Zeng M, Qian Y, Zhang R. Smoke and Fire Detection Based on YOLOv7 With Convolutional Structure Reparameterization and Lightweighting. *IEEE Sensors Letters*. 2024 Jul 1.
42. Kong D, Li Y, Duan M. Fire and smoke real-time detection algorithm for coal mines based on improved YOLOv8s. *PLoS One*. 2024;19(4):e0300502. <https://doi.org/10.1371/journal.pone.0300502> PMID: 38635515
43. Jocher G. YOLOv5 by Ultralytics [software]. Version 7.0. Zenodo; 2020 May 29. Available from: <https://github.com/ultralytics/yolov5>. doi: 10.5281/zenodo.3908559. License: AGPL-3.0.
44. Casas E, Ramos L, Bendek E, Rivas-Echeverria F. YOLOv5 vs. YOLOv8: Performance Benchmarking in Wildfire and Smoke Detection Scenarios. *JOIG*. 2024;12(2):127–36. <https://doi.org/10.18178/joig.12.2.127-136>
45. Talaat FM, ZainEldin H. An improved fire detection approach based on YOLO-v8 for smart cities. *Neural Comput & Applic*. 2023;35(28):20939–54. <https://doi.org/10.1007/s00521-023-08809-1>
46. Yang J, Zhu W, Sun T, Ren X, Liu F. Lightweight forest smoke and fire detection algorithm based on improved YOLOv5. *PLoS One*. 2023;18(9):e0291359. <https://doi.org/10.1371/journal.pone.0291359> PMID: 37683034
47. Feurer M, Hutter F. Automated machine learning: Methods, systems, challenges. 2019:3–3.
48. Falkner S, Klein A, Hutter F. Practical hyperparameter optimization for deep learning. 2018. <https://doi.org/10.48550/DOlorotheridentifiers>
49. Shawki N, Nunez RR, Obeid I, Picone J. On Automating Hyperparameter Optimization for Deep Learning Applications. 2021 IEEE Signal Processing in Medicine and Biology Symposium (SPMB). 2021:1–7. <https://doi.org/10.1109/spmb52430.2021.9672266>
50. Wu L, Perin G, Picek S. I choose you: Automated hyperparameter tuning for deep learning-based side-channel analysis. *IEEE Transactions on Emerging Topics in Computing*. 2022.
51. Mirjalili S, Mirjalili S. Genetic algorithm. *Evolutionary algorithms and neural networks: theory and applications*. 2019:43–55.
52. Sarkar D, Mishra N, Biswas A. Genetic algorithm-based deep learning models: a design perspective. *Proceedings of the Seventh International Conference on Mathematics and Computing: ICMC 2021*. 2022;2021:361–72.
53. Li N, Ma L, Yu G, Xue B, Zhang M, Jin Y. Survey on Evolutionary Deep Learning: Principles, Algorithms, Applications, and Open Issues. *ACM Comput Surv*. 2023;56(2):1–34. <https://doi.org/10.1145/3603704>
54. Shrestha A, Mahmood A. Optimizing deep neural network architecture with enhanced genetic algorithm. 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). 2019:1365–70. <https://doi.org/10.1109/icmla.2019.00222>

55. Aszemi NM, Dominic PDD. Hyperparameter optimization in convolutional neural network using genetic algorithms. *IJACSA*. 2019;10(6). <https://doi.org/10.14569/ijacsa.2019.0100638>
56. Suhail K, Brindha D. Microscopic urinary particle detection by different YOLOv5 models with evolutionary genetic algorithm based hyperparameter optimization. *Comput Biol Med*. 2024;169:107895. <https://doi.org/10.1016/j.combiomed.2023.107895> PMID: 38183704
57. Arrieta A, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, et al. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*. 2020;58(6):82–115.
58. Selvaraju R, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*. 2020;128(2):336–59.
59. Gunning D, Aha DW. DARPA's explainable artificial intelligence program. *AI Magazine*. 2019;40(2):44–58. <https://doi.org/10.1609/aimag.v40i2.2850>
60. Jin C, Wang T, Alhusaini N, Zhao S, Liu H, Xu K, Zhang J. Video fire detection methods based on deep learning: datasets, methods, and future directions. *Fire*. 2023 Aug 14;6(8):315.
61. Chitram S, Kumar S, Thenmalar S. Enhancing fire and smoke detection using deep learning techniques. *Engineering Proceedings*. 2024;62(1):7.
62. Du Y, Zhao Z, Song Y, Zhao Y, Su F, Gong T, et al. Strongsort: make deepsort great again. *IEEE Transactions on Multimedia*. 2023;25(1):8725–37.
63. Zhang Y, Kim M, Jin S. Real-time detection and tracking of surgical instrument based on YOLOv5 and DeepSORT*. 2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN). 2023:1758–63. <https://doi.org/10.1109/ro-man57019.2023.10309495>
64. Zhao G. Research on real-time athlete tracking method based on computer deep learning. 2024 IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT). 2024:1063–8. <https://doi.org/10.1109/iccect60629.2024.10546017>
65. Guo F, Xu Y. Vehicle analysis system based on DeepSORT and Yolov5. *Proceedings of the 2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA)*. 2022:175–9.
66. Géron A. Hands-on machine learning with scikit-learn, keras, and tensorflow. 2nd ed. O'Reilly Media, Inc. 2022:418–418.
67. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A. The pascal Visual Object Classes (VOC) Challenge. *Int J Comput Vis*. 2010;88(2):303–38. <https://doi.org/10.1007/s11263-009-0275-4>
68. Culjak I, Abram D, Pribanic T, Dzapo H, Cifrek M. A brief introduction to OpenCV. *Proceedings of the 35th International Convention MIPRO 2012*. 2012:1725–30.