

Article

A Fire Detection Method for Aircraft Cargo Compartments Utilizing Radio Frequency Identification Technology and an Improved YOLO Model

Kai Wang, Wei Zhang * and Xiaosong Song

School of Microelectronics, Tianjin University; Tianjin 300072, China; tjdxwk@tju.edu.cn (K.W.); 1023232012@tju.edu.cn (X.S.)

* Correspondence: tjuzhangwei@tju.edu.cn; Tel.: +86-138-2146-5375

Abstract: During flight, aircraft cargo compartments are in a confined state. If a fire occurs, it will seriously affect flight safety. Therefore, fire detection systems must issue alarms within seconds of a fire breaking out, necessitating high real-time performance for aviation fire detection systems. In addressing the issue of fire target detection, the YOLO series models demonstrate superior performance in striking a balance between computational efficiency and recognition accuracy when compared with alternative models. Consequently, this paper opts to optimize the YOLO model. An enhanced version of the FDY-YOLO object detection algorithm is introduced in this paper for the purpose of instantaneous fire detection. Firstly, the FaB-C3 module, modified based on the FasterNet backbone network, replaces the C3 component in the YOLOv5 framework, significantly decreasing the computational burden of the algorithm. Secondly, the DySample module is used to replace the upsampling module and optimize the model's ability to extract the features of small-scale flames or smoke in the early stages of a fire. We introduce RFID technology to manage the cameras that are capturing images. Finally, the model's loss function is changed to the MPDIoU loss function, improving the model's localization accuracy. Based on our self-constructed dataset, compared with the YOLOv5 model, FDY-YOLO achieves a 0.8% increase in mean average precision (mAP) while reducing the computational load by 40%.



Academic Editor: Pawel Strumillo

Received: 8 November 2024

Revised: 26 December 2024

Accepted: 27 December 2024

Published: 30 December 2024

Citation: Wang, K.; Zhang, W.; Song, X. A Fire Detection Method for Aircraft Cargo Compartments Utilizing Radio Frequency Identification Technology and an Improved YOLO Model.

Electronics **2025**, *14*, 106. <https://doi.org/10.3390/electronics14010106>

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: RFID; fire detection; aircraft cargo compartment fire; YOLO

1. Introduction

Aircraft cargo compartments are enclosed spaces. In the event of a fire, the flames can spread rapidly, posing a serious threat to flight safety [1]. Therefore, it is of great significance to detect fires in aircraft cargo compartments quickly and accurately. The current airworthiness regulation FAR 25.858 stipulates that smoke and fire detectors used in aircraft cargo compartments must issue alarm signals within 60 s after the onset of a fire [2]. Most current aircraft cargo compartments employ photoelectric smoke detectors for fire detection, which operate on the principle of incident light scattering by fire smoke particles [3]. When smoke particles enter the detector, a receiver situated at a specific angle with respect to the incident light is able to capture a portion of the scattered light [4]. If the intensity of this scattered light exceeds a specific threshold, the detector determines that a fire has occurred. Nonetheless, in aircraft cargo compartments, these devices are prone to a significant rate of false alarms, attributed to the diversity of cargo and the intricate environments (including fluctuations in air pressure and airflow) that frequently produce

dust particles. When dust particles enter the detector, they may cause the detector to receive scattered light even in the absence of a fire, potentially leading to the smoke detector falsely sounding a fire alarm. In recent years, studies on fire detection utilizing image-processing techniques and artificial intelligence technology have emerged as a hot topic. Kruell et al. [5] introduced an approach to detect fires in aircraft cargo compartments by using low-cost near-infrared CCD cameras to directly detect flames and hotspots, followed by analyzing the captured images to detect smoke. Yu et al. [6] suggested an approach based on the fusion of domain-adversarial neural networks and graph convolutional networks for extracting flame characteristics, aiming to enhance recognition performance in fire scenarios. Buriboev et al. [7] proposed a method that integrates contour analysis with deep convolutional neural networks to detect fires by utilizing flame color and contour features. This approach utilizes cameras to capture image information from the target area and employs deep learning networks to identify whether the images contain flames or smoke, thereby determining if a fire has occurred. Compared with photoelectric smoke detectors, this method boasts advantages such as a lower false alarm rate, superior visualization, and faster detection speed. It can detect small-scale flame and smoke targets during the initial stages of a fire, enabling early fire alarm functionality. However, most existing deep learning models for fire recognition are large-scale, structurally complex, and computationally intensive. The onboard hardware systems of aircraft have limited computational resources and cannot directly deploy existing fire recognition algorithm models. Therefore, this paper puts forward a fire detection algorithm model tailored for aircraft cargo compartments utilizing optimizations to the YOLOv5 model.

The key contributions of this research are as follows:

1. By incorporating the Faster Block from the FasterNet backbone into the C3 module, this paper reduces the number of parameters in the YOLOv5 framework and significantly decreases the model's inference time.
2. To enhance the detection of small-scale smoke and flames during the initial stages of a fire, this paper replaces the upsampling module with DySample, which better preserves feature details and enhances detection accuracy.
3. This paper employs the MPDIoU loss function to enhance the accuracy of identifying small-scale flame and smoke targets during the initial stages of a fire and improve the effectiveness of multi-scale target detection, which facilitates the identification of targets, such as flames and smoke, when they are relatively small during the initial stages of a fire.
4. Multiple cameras are typically used for image acquisition in aircraft cargo hold areas. This paper proposes managing these surveillance cameras in aircraft cargo holds using RFID (radio frequency identification) technology, with the aim of identifying the location of the camera that triggers the fire alarm in the event of a fire.

The organization of this paper is as follows: Section 2 describes the common methods adopted in the existing research for aerial fire detection problems and analyzes the potential for improving algorithms in terms of computational efficiency. Section 3 presents the modifications made to the YOLO (You Only Look Once) model's structure and the enhancement of the loss function, which significantly reduce the computational load without compromising recognition performance. Additionally, efficient management of image detectors is achieved through RFID technology. Section 4 compares the proposed algorithm with several common algorithms using a public fire dataset, highlighting the advantages of our algorithm in terms of computational efficiency and recognition accuracy. Section 5 summarizes the content of this paper.

2. Aviation Fire Detection and Image-Based Fire Detection Methods

At the onset of a fire, a substantial amount of smoke and flames are produced, which exhibit significant image characteristics. Existing photoelectric smoke detectors only utilize the spectral characteristics of flames for detection without leveraging the rich image features of fires [8]. Yongbo et al. [9] proposed a composite fire detection device that combines temperature sensors, CO sensors, and dual-wavelength photoelectric smoke sensors to enhance the alarm accuracy of the detector. Li et al. [10] introduced an optical fire detection system that combines tunable diode laser absorption spectroscopy (TDLAS) technology with laser-ranging technology to achieve fire detection by detecting CO produced during the initial phases of a fire. The early fire detection algorithms utilizing image processing necessitated manual identification of the fire image features that needed to be extracted. For instance, Wu et al. [11] proposed a fire detection method that utilizes flame shape and brightness. Fang et al. [12] introduced a smoke detection method based on video features that utilizes features such as smoke color, motion direction, and texture to distinguish smoke. This method not only detects smoke but also locates the source of the fire simultaneously. However, these algorithms rely on a relatively small number of features for fire decision making.

Currently, detection models represented by the YOLO series have achieved an optimal equilibrium between speed and accuracy, fulfilling the real-time requirements of fire detection. These methods have found widespread application in fire detection in areas such as forests, transportation, and ports [13]. Zheng et al. [14] presented an instantaneous fire detection algorithm based on YOLOv4, in which the CSP Darknet53 module was replaced with the MobileNetV3-large module in the YOLOv4 model, thereby reducing the computational load of the model. Du et al. [15] enhanced the YOLOv5s model by incorporating an attention mechanism, which enables the model to more fully extract multi-scale spatial information of the targets, thereby capturing prominent features of smoke and flames. Wang et al. [16] integrated the attention mechanism of the convolutional block attention module (CBAM) into the YOLOv6 model, optimizing the weights of feature maps in both channel and spatial aspects, thereby enhancing the model's ability to extract fire-related features. Chen et al. [17] introduced a fire detection approach that integrates the ECA (efficient channel attention) mechanism and utilizes the SIoU (symmetric intersection over union) loss function within the YOLOv7 framework. This approach improves the stability of the model's loss function convergence and boosts regression accuracy. Titu et al. [18] introduced an instantaneous algorithm for fire detection by using YOLOv8m as the base model and incorporating knowledge distillation techniques.

Despite the emergence of numerous YOLO series algorithms applied to fire detection in recent years, there is still considerable scope for enhancing fire detection models specifically tailored for aircraft cargo holds. Due to the limitations of onboard hardware computing resources, the fire detection model requires further optimization of its network structure to reduce computational load while ensuring that recognition accuracy does not significantly decrease. This paper presents an optimized design based on the basic network structure of YOLOv5, realizing a fire detection method for aircraft cargo holds.

3. Improved Network

3.1. FDY-YOLO

The YOLOv5 model structure comprises four main components: the input end, backbone, neck, and head, which work together to achieve efficient feature extraction and object detection from input images. The backbone, composed of key components such as Conv modules, C3 modules, SPPF modules, and CSP structures, is tasked with extracting useful feature information from the input images. The neck network, located between the

backbone and head networks, further processes and fuses the features extracted from the backbone through multi-scale feature fusion. The head network, meanwhile, predicts the categories and locations of objects based on the feature information extracted from the backbone and neck networks. The fundamental concept of the C3 module is to extract and integrate features through multiple convolutional blocks and CSP structures to enhance the accuracy and efficacy of object detection.

As shown in Figure 1, in this study, to attain a lightweight model structure, we replaced the bottleneck module in C3 with the faster block module, naming it the FaB-C3 module. Faster block serves as the backbone network for FasterNet [19]. This effectively reduces the computational requirements of the model but also results in a reduction in model accuracy. To improve the model recognition accuracy and small object detection capabilities, we substituted the MPDIoU loss function for the CIoU loss function and replaced the upsampling module with the DySample module. Ultimately, we obtained a new model, which we called FBD-YOLO (a YOLO model incorporating the faster block and DySample modules).

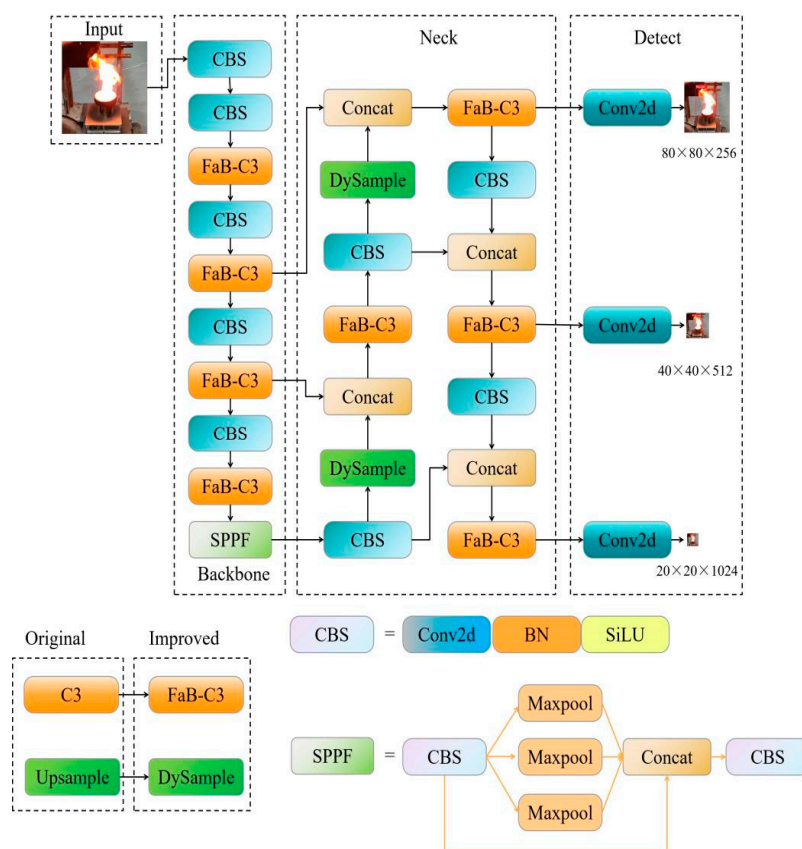


Figure 1. FDY-YOLOv5 structure diagram.

3.2. FaB-C3 Network

The structural differences between the C3 module and the FaB-C3 module are illustrated in Figure 2. The bottleneck module is composed of two ordinary convolutional modules, whereas the faster block, which serves as the backbone network for FasterNet, comprises a partial convolution (PConv) and a pointwise convolution (PWConv).

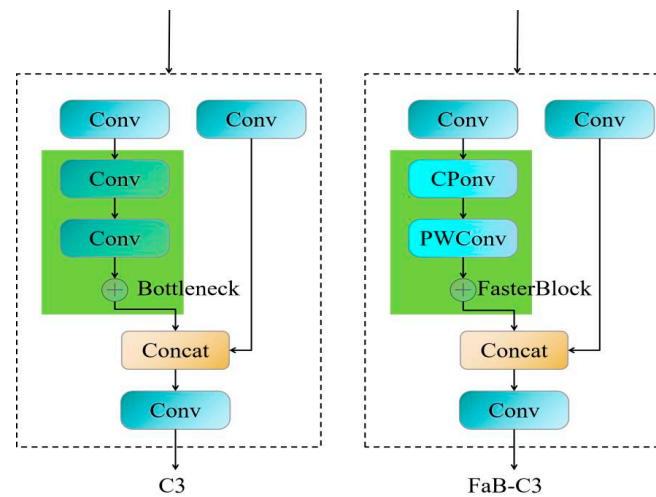


Figure 2. Comparison diagram of C3 module and FaB-C3 module structures.

The computational complexity of the bottleneck module in C3 is shown in Equation (1) as follows:

$$G_{FB} = 2 \times h \times w \times c^2 \times k^2 = 2 \times h \times w \times c^2 \times 1 \times 1 \quad (1)$$

In the equation, w represents the width of the input image, and h represents its height. c signifies the number of input channels, and k denotes the dimension of the convolutional kernel. The PConv (partial convolution) only applies ordinary convolution to 1/4 of the channels focused on extracting spatial features, while the other channels remain unaltered for the next step of feature fusion. The complexity of computation for the PWConv (pointwise convolution) module, the complexity involved in the faster block module is shown in Equation (2) as follows:

$$G_{C3} = h \times w \times \left(c_p^2 \times k^2 + c_p \times c \right) = h \times w \times c^2 \times \left(\frac{1}{4} + \frac{9}{16} \right) \quad (2)$$

$$G_{FB} < G_{C3} \quad (3)$$

Therefore, introducing the faster block module into the C3 module can markedly decrease the model's computational complexity, achieving a lightweight design for the YOLOv5 model.

3.3. DySample

Upsampling operations can integrate feature maps from various scales, thereby combining information from various scales and boosting the model's capacity to identify targets of different sizes. The YOLOv5 architecture employs nearest-neighbor upsampling for the upsampling operation. The computational process of the nearest-neighbor upsampling algorithm, as depicted in Figure 3, begins with the first pixel in the top-left corner of the original feature map. It replicates this pixel with a stride of 2 and then reassembles the replicated pixels into a 2×2 shape. This process continues by sequentially traversing all remaining pixels in the original feature map, ultimately resulting in a new 4×4 feature map. This method is simple to implement, requires minimal computation, and operates quickly. However, it has the limitation of overly relying on the spatial position relationships of pixels and failing to fully utilize the abundant semantic content contained in the feature map. This limitation results in a typically smaller receptive field, posing a challenge for the model to effectively capture the required extensive and rich contextual semantic information. Therefore, this study introduced the lightweight upsampling operator, Dysample [20],

into the neck network to better preserve the detailed information of the feature map and capture subtle features within it.

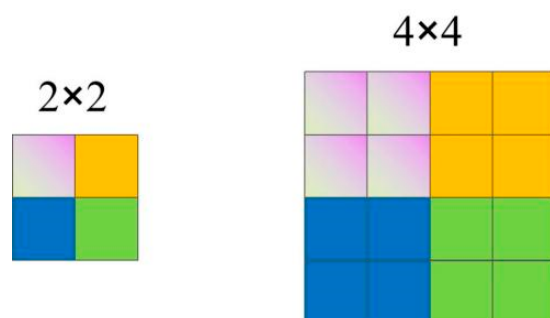


Figure 3. Diagram of the calculation process for nearest-neighbor upsampling.

The Dysample operator, based on the idea of point sampling, achieves the upsampling process by introducing offsets. It has the capability to adjust these offsets dynamically according to the information contained within the input feature map, which enables the upsampled result to have better positional awareness and better preserves and reconstructs the specific details contained within the feature map, thus improving the model's performance.

Figure 4 illustrates the primary upsampling process within the DySample module. It accepts a feature map χ of size $C \times H \times W$ and a point sampling set δ of size $2g \times sH \times sW$ as inputs, where the first dimension of $2g$ represents the x- and y-coordinates, C denotes the channel count, H denotes the height, and W denotes the width. The grid-sampling function utilizes the coordinates in δ to resample the hypothetically bilinearly interpolated χ into χ' with dimensions $C \times sH \times sW$. This process is referred to as

$$\chi' = \text{grid_sample}(\chi, \delta) \quad (4)$$

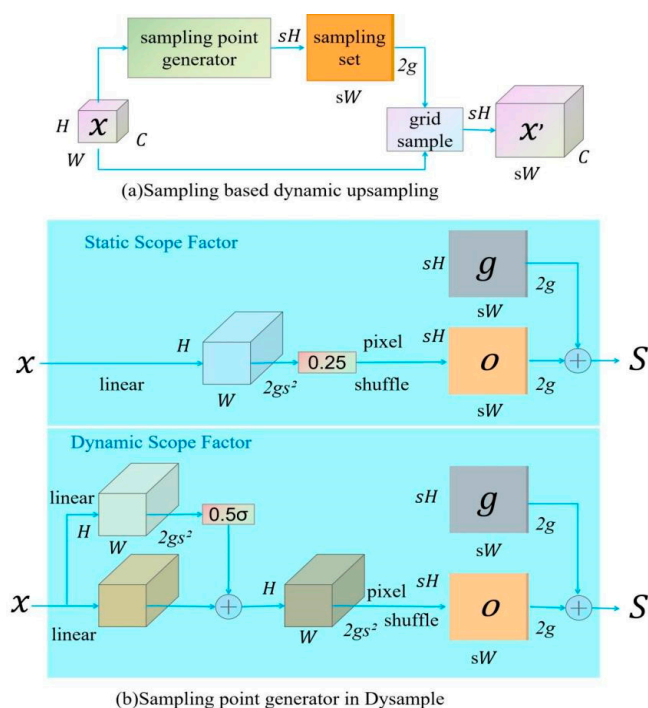


Figure 4. Diagram of the DySample dynamic upsampler structure.

Given a feature map X with dimensions $C \times H \times W$ and an upsampling scale factor s , a linear layer with an input channel dimension of C and an output channel dimension of $2s^2$ is used to generate offset values O with a size of $2s^2 \times H \times W$.

$$O = \text{linear}(X) \quad (5)$$

These offset values are then reshaped using pixel shuffle to form $2 \times sH \times sW$. The sampling set S is obtained by adding the offset values O with the original grid sampling G , as shown in Equation (6):

$$S = G + O \quad (6)$$

The Dysample module takes an input feature map X , maps it onto an output through a convolution operation, applies a sigmoid function for activation, and then scales the values by multiplying them by 0.5 to obtain a range parameter. This range parameter adjusts the scale of the targets by multiplying with an offset, followed by a normalization operation. Afterward, a pixel rearrangement function is utilized to reorder the coordinates, which are then combined with the initial positions to obtain the ultimate position coordinates. To enhance the flexibility of the offsets and accommodate various feature distributions and data scenarios, the Dysample module generates point-wise dynamic range factors by linearly projecting the input features. These dynamic range factors can adaptively modify the offsets based on the attributes of the input data. The Dysample dynamic upsampling method preserves fine details in images better than traditional upsampling methods, avoiding blurring or aliasing artifacts caused by simple pixel replication.

3.4. MPDIoU

The default loss function used in the YOLOv5 model is the CIOU loss function [21]. However, in cases where the predicted bounding boxes and the ground truth bounding boxes share the same aspect ratio but have different width and height values, the CIOU loss function may produce invalid predicted bounding boxes. MPDIoU [22] is a metric for evaluating the likeness of bounding boxes based on the smallest point-to-point distance. As shown in Figure 5, it streamlines the computation by directly reducing the distance between the top-left and bottom-right points of the predicted and ground-truth boxes. It takes into account all the pertinent factors that are considered in the existing loss functions. Additionally, it resolves the inefficiency issue that arises when the forecasted bounding box and the actual bounding box share the same aspect ratio yet differ in their width and height dimensions. This accelerates the convergence speed of model training.

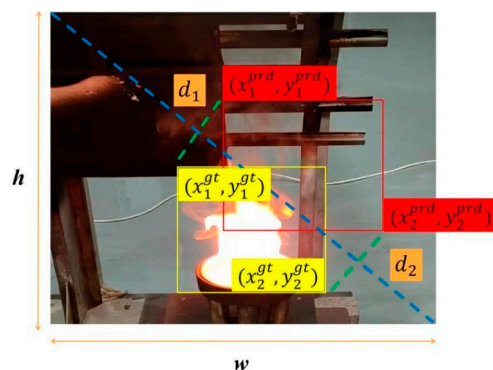


Figure 5. MPDIoU loss function considering the coordinates of the top-left and bottom-right points of the bounding box.

All components of the MPDIoU loss function can be calculated using the coordinates of the four vertices. The calculation process is shown below:

$$d_1^2 = (x_1^{\text{prd}} - x_1^{\text{gt}})^2 + (y_1^{\text{prd}} - y_1^{\text{gt}})^2 \quad (7)$$

$$\text{MPDIoU} = \text{IOU} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2} \quad (8)$$

$$|c| = (\max(x_2^{\text{gt}}, x_2^{\text{prd}}) - \min(x_1^{\text{gt}}, x_1^{\text{prd}})) \times (\max(y_2^{\text{gt}}, y_2^{\text{prd}}) - \min(y_1^{\text{gt}}, y_1^{\text{prd}})) \quad (9)$$

$$x_c^{\text{gt}} = \frac{x_1^{\text{gt}} + x_2^{\text{gt}}}{2}, y_c^{\text{gt}} = \frac{y_1^{\text{gt}} + y_2^{\text{gt}}}{2} \quad (10)$$

$$x_c^{\text{prd}} = \frac{x_1^{\text{prd}} + x_2^{\text{prd}}}{2}, y_c^{\text{prd}} = \frac{y_1^{\text{prd}} + y_2^{\text{prd}}}{2} \quad (11)$$

$$w_{\text{gt}} = x_2^{\text{gt}} - x_1^{\text{gt}}, h_{\text{gt}} = y_2^{\text{gt}} - y_1^{\text{gt}} \quad (12)$$

$$w_{\text{prd}} = x_2^{\text{prd}} - x_1^{\text{prd}}, h_{\text{prd}} = y_2^{\text{prd}} - y_1^{\text{prd}} \quad (13)$$

In the formula, the input image has a width of w and a height of h . $(x_1^{\text{gt}}, y_1^{\text{gt}})$ and $(x_2^{\text{gt}}, y_2^{\text{gt}})$ are the top-left and bottom-right coordinates of the actual bounding box, while $(x_1^{\text{prd}}, y_1^{\text{prd}})$ and $(x_2^{\text{prd}}, y_2^{\text{prd}})$ are the top-left and bottom-right coordinates of the predicted bounding box. These coordinates are used to calculate the distance between the top-left and bottom-right corners of the diagonals of the predicted bounding box and the ground truth bounding box. $|c|$ denotes the area of the smallest enclosing rectangle that covers both the forecasted and actual bounding boxes. $(x_c^{\text{gt}}, y_c^{\text{gt}})$ and $(x_c^{\text{prd}}, y_c^{\text{prd}})$ denote the central coordinates of the actual and forecasted bounding boxes, respectively. The dimensions of the ground-truth bounding box are given by its width, w_{gt} , and its height, h_{gt} , while w_{prd} and h_{prd} denote the dimensions (width and height) of the predicted bounding box. The formula for calculating the MPDIoU loss function is shown in Equation (14).

$$\text{MPDIoU_loss} = 1 - \text{MPDIoU} = 1 - \text{IOU} + \frac{d_1^2}{w^2 + h^2} + \frac{d_2^2}{w^2 + h^2} \quad (14)$$

This paper introduces the MPDIoU loss metric to simplify calculations and enhance the recognition accuracy of fire and smoke targets. It mainly leverages the advantages and characteristics of MPDIoU, such as its ability to more directly reflect the similarity between the predicted bounding box and the actual bounding box, its comprehensive evaluation of the differences between the forecasted bounding box and the actual bounding box, and its significant performance improvements observed in multiple models.

3.5. RFID

Recently, radio frequency identification (RFID) technology has undergone rapid development and has found widespread applications in various types of electronic products [23], bringing great convenience to device management [24]. The basic modules of a typical RFID device include electronic tags, readers/writers, antennas, and processors [25]. This paper incorporates RFID technology into the camera system of the aircraft cargo hold to facilitate the management of image data captured by the cameras. The image acquisition structure, as shown in Figure 6, includes cameras, electronic tags, readers/writers, and controllers.

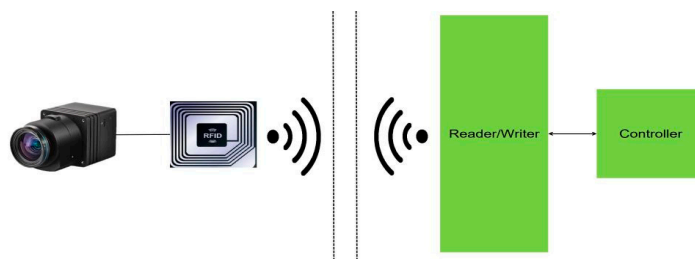


Figure 6. Image acquisition system based on RFID technology.

3.6. RFID System Application Process

The benefit of applying RFID in this paper is that it can improve the efficiency of managing cameras within the cabin, allowing airline personnel to quickly check the status of the cameras via handheld devices. The RFID system in this paper is mainly used to manage the operational status of camera sensors. The RFID system has a simple structure and is easy to operate [26,27]. With reference to typical RFID systems [28], the usage method of the RFID system in this paper was designed.

1. The crew install the corresponding equipment tags for the fire detectors.
2. The crew use handheld mobile devices to scan and record the tags and utilize software to bind the tags with the corresponding detectors to ensure consistency between them. They also synchronize the detector data in a timely manner, aggregate the data information of the detectors, and upload it to the onboard management platform for review by the crew.
3. The various data that have passed the review are synchronized to the onboard management system, and the detectors are counted and recorded regularly. If any detectors are found to be malfunctioning, they are repaired or replaced promptly to ensure the normal operation of all detectors. This ensures that the detectors can sound an alarm in the event of a fire accident, avoiding unnecessary casualties caused by detector failures.

In the cargo hold, we reduce metal interference by adjusting the relative positions and orientations of the devices, avoiding direct contact between the readers/writers and metal objects, and utilizing multiple antennas. The environmental conditions, such as temperature, humidity, and vibrations in aircraft cargo compartments, vary with flight altitude, requiring more comprehensive experimental conditions for validation. This is also one of our future research areas and directions. The limitation of using RFID technology is its susceptibility to interference from metal objects within the cargo hold. How to eliminate such interference is a direction for future research.

4. Results and Discussion

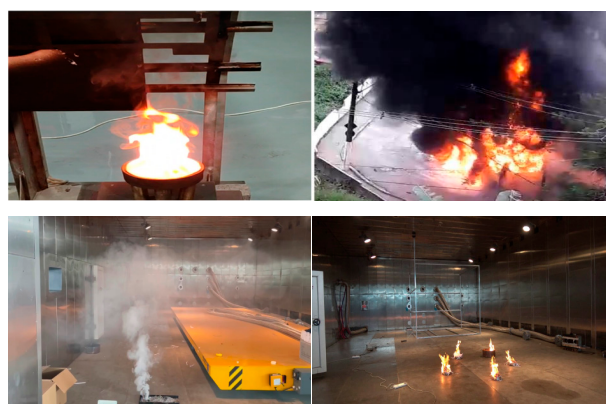
4.1. Experimental Setup and Data Collection

The ablation experiments, comparative analysis experiments, and comparison experiments with other algorithms in this paper all adopted the same experimental configuration, with the specific configuration parameters shown in Table 1. The model introduced in this paper utilizes the Python programming language and the Pytorch framework for deep learning.

Table 1. Configuration of the experimental environment.

Parameter Name	Configuration
CPU	Intel(R) Core(TM) i5-4590 CPU @ 3.30 GHz
RAM	48 G
GPU	NVIDIA Quadro RTX 8000
Operation system	Windows 10 64 bit
Language	Python3.8
Deep learning architecture	Torch 1.13 deep learning framework
CUDA	11.7

Based on fire experiments conducted in a simulated aircraft cargo hold, combined with fire images collected from the internet, this paper established a self-made fire dataset that includes flame and smoke scenes. Figure 7 displays sample images taken from the dataset. The detailed information of the fire database is shown in Table 2.

**Figure 7.** Example of the dataset image.**Table 2.** Information on the fire dataset.

Data Type	Quantity
Only flame	4396
Only smoke	3784
Both flame and smoke	2787

In this study, the fire dataset was enhanced to include a grand total of 10,967 images, which were subsequently divided into a training dataset and a test dataset in a 9:1 ratio. In the training phase, the stochastic gradient descent (SGD) algorithm was utilized for optimization, utilizing a batch size of 16 images. The following learning parameters were established: a starting learning rate set at 0.01, a momentum factor of 0.937, and a weight decay rate of 0.0005. The decision was made to conduct the training over 60 epochs, with each input image rescaled to 640×640 pixels. Once the training was finalized, the weight file of the recognition model was preserved.

As depicted in Figure 8, following 60 epochs of iterations, the model parameters reach a stable state. Notably, the box regression loss concludes at 0.02, the objectness loss descends to 0.0075, and the classification loss diminishes to 0.001. Upon examining the convergence scenario, it becomes clear that the training results of the refined algorithm exhibit notably superior satisfaction, underscoring its improved efficacy. In the course of training the model presented in this paper, we consistently scrutinized its performance on the validation dataset. An increase in validation loss served as an indication of the onset of

overfitting, prompting us to terminate the training process preemptively to mitigate the risk of overfitting.

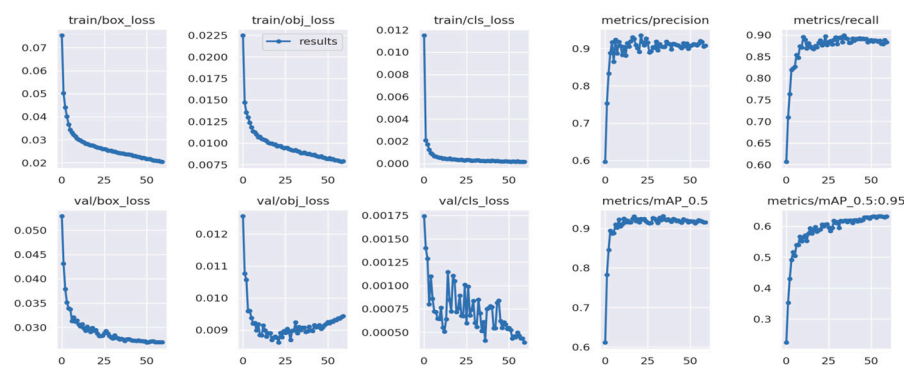


Figure 8. Loss function convergence plot.

4.2. Comparative Analysis of Heatmaps

Often seen as “black boxes”, deep learning models conceal their internal operations, making it hard for people to understand which features the model considers significant for its decisions. This, in turn, complicates the interpretation of the model’s outputs and erodes trust in its findings. Therefore, to obtain a deeper understanding of the model and confirm the efficacy of the enhanced method presented in this study, we employed Grad-CAM [29] to generate heatmaps that aided in analyzing the outcomes of the model’s detection.

Grad-CAM, a neural network visualization tool, creates heatmaps by relying on the weights of various parts, offering a thorough and precise portrayal of the model’s efficacy. As illustrated in Figure 9, a deeper shade of red in the image signifies higher feature importance. As is shown, the pre-modified YOLOv5 model assigns considerable weights to the brighter areas at the heart of the flames, resulting in a confused heatmap distribution prone to incorrect recognition and misjudgments. In contrast, the enhanced FDY-Yolo model prioritizes feature extraction from the flame region, enhancing recognition accuracy. This underscores the effectiveness of the method of model improvement presented in this study for detecting fires in aircraft cargo compartments.

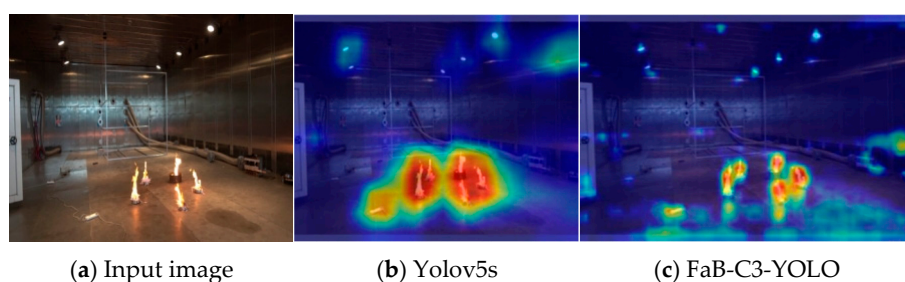


Figure 9. Comparison of heatmaps before and after model improvement.

4.3. Evaluation Index

In this paper, the GFLOPs (giga floating-point operations per second) parameter is adopted to analyze the optimization of computational complexity of the model. Meanwhile, to demonstrate the model’s detection accuracy, the parameters of P (precision), R (recall), and mAP (mean average precision for multiple categories) are utilized.

The calculation of precision and recall is as follows:

$$P = \frac{TP}{TP + FP} \quad (15)$$

$$R = \frac{TP}{TP + FN} \quad (16)$$

In the equation, TP stands for the count of accurately detected targets, FP for the number of incorrectly identified targets, and FN for the count of undetected targets. Additionally, P signifies the ratio of correctly forecasted positive samples in all forecasted positive instances, whereas R represents the ratio of accurately predicted positive samples in all actual positive instances.

The calculation of the multi-category average precision (mAP) is outlined as follows:

$$AP = \int_0^1 P(R) dR \quad (17)$$

$$mAP = \frac{\sum_{i=1}^N AP_i}{N} \quad (18)$$

In the aforementioned equation, N indicates the overall count of categories to be classified. AP (average precision) signifies the mean precision of a particular target class, computed by averaging the precision scores obtained at various recall levels. Conversely, mAP (mean average precision) signifies the average of all AP values across all classes, offering a holistic assessment of the algorithm's performance across multiple categories.

GFLOPs, an acronym for giga floating-point operations per second, signifies the rate of floating-point operations executed per second, equating to one billion operations per second. It functions as an indicator to quantify the overall computational demand of a model. As a measure, GFLOPs can indicate the computational complexity of a neural model, encapsulating both its execution speed and computational capability.

4.4. Ablation Experiments

To fully evaluate the effectiveness of each component in the FDY-YOLO algorithm, we carried out ablation studies focusing on the integration of the FaB-C3 module, the DySample upsampling module, and the MPDIoU loss function. The outcomes of these studies are detailed in Table 3. When only the FaB-C3 module was substituted, while the model's computational requirement decreased from 16.0 G to 9.6 G, there was a corresponding decline in performance: a 1.4% drop in mAP, a 2.5% decrease in precision (P), and a 4.7% reduction in recall (R), in comparison with the baseline model. Meanwhile, introducing the DySample upsampling module independently led to a 0.3% enhancement in the model's mAP. Similarly, the inclusion of the MPDIoU loss function alone resulted in a 0.4% increase in mAP. The combination of both the DySample and MPDIoU modules in the baseline model resulted in the most substantial enhancement, with a 1.1% increase in mAP to 92.9%. However, this configuration also incurred the highest computational load, totaling 16.5 G, which may render it unsuitable for applications requiring high real-time detection capabilities.

Table 3. Ablation experiment results.

YOLOv5s	FaB-C3	DySample	MPDIoU	P/%	R/%	mAP@0.5/%	GFLOPs/G
✓	×	×	×	89.9	90.1	91.8	16.0
✓	✓	×	×	87.4	85.4	90.4	9.6↓
✓	×	✓	×	88.9	88.1	92.1	16.5
✓	×	×	✓	89.9	90.2↑	92.2	16.0
✓	✓	✓	×	88.4	89.5	91.6	10.2
✓	✓	×	✓	88.7	89.8	91.7	9.6↓
✓	×	✓	✓	92.1	86.7	92.9↑	16.5
✓	✓	✓	✓	90.1	88.9	92.6	10.6

The experimental findings indicate that while the FaB-C3 module effectively lowers the computational burden, it comes at the cost of a minor decline in recognition accuracy. Conversely, the integration of the DySample upsampling module and the MPDIoU loss function boosts recognition accuracy. Given these insights, we propose a novel model, FDY-YOLO, which combines the three components: FaB-C3, DySample, and MPDIoU. Notably, the FDY-YOLO model achieves a substantial reduction in computation, totaling only 9.6 G, which marks a 40% decrease compared with the original model. Impressively, despite this reduction, the model's mAP is improved by 0.8%. Thus, the model presented in this paper demonstrates superior overall performance.

Observing Figure 10, it becomes evident that on the identical dataset, replacing the FaB-C3 module alone results in inferior recognition, with some flame targets missing from the images in comparison with those from the initial model. This decrement in effectiveness can be ascribed to the FaB-C3's lightweight structure, which decreases computational load but also sacrifices recognition accuracy. However, upon incorporating the DySample upsampling module, the recognition accuracy improves, reaching levels comparable to the original model. Furthermore, the addition of the MPDIoU loss function on top of this enhancement significantly outperforms the original model, enabling it to identify all flames existing within the images. This demonstrates that the model proposed in this paper not only minimizes computational requirements but also elevates recognition accuracy, making it better adapted to instantaneous fire detection in various circumstances.

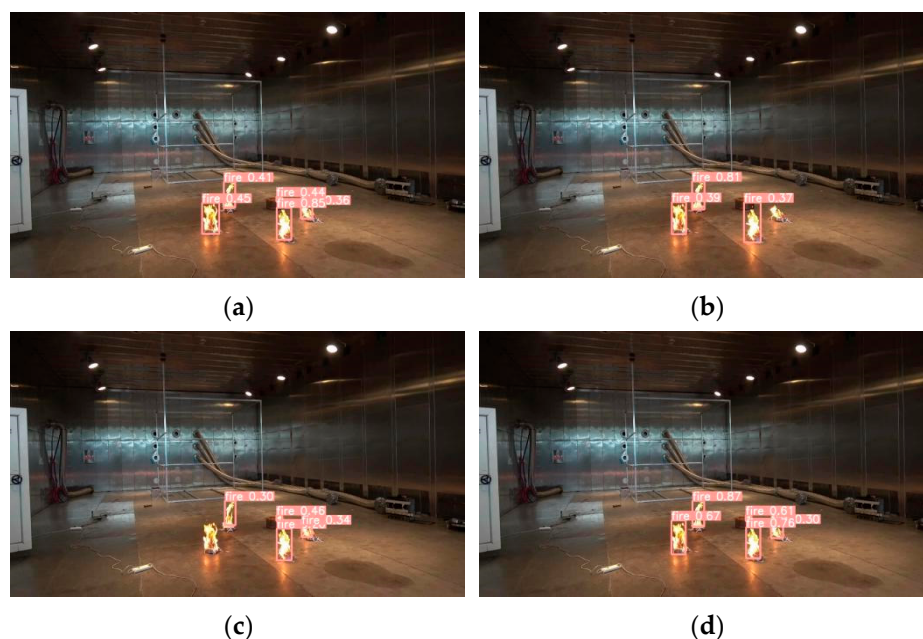


Figure 10. Comparison of detection effects. (a) Only the original YOLOv5, (b) YOLOv5 + FaB-C3, (c) YOLOv5 + FaB-C3 + DySample, and (d) FDY-YOLO.

4.5. Comparative Analysis of Different Models

For a more in-depth comparison of performance disparities, we contrasted the algorithm presented in this paper with several prevalent network algorithms. We utilized D-Fire [30], which is a publicly available fire dataset that contains over 17,000 images of fire and smoke, encompassing a range of fire situations, including indoor fires, forest fires, and fires on highways, and Figure 11 displays some sample images of fire scenarios from the D-FIRE dataset.



Figure 11. Example of a D-Fire dataset image.

This paper benchmarked the FDY-YOLO model against popular object detection methods, such as YOLOv4, YOLOv6s, YOLOv7-tiny, YOLOv8s, and Faster R-CNN. The assessment metrics for these models encompassed recall (R), precision (P), and mean average precision (mAP) at a 0.5 IoU threshold (mAP@0.5), and computational complexity measured in gigaflops (GFLOPs). The experimental outcomes are summarized in Table 4. While YOLOv7-tiny offers comparable detection performance to our model, its higher computational demand of 13.9 G, in contrast to our model's 10.6 G, poses challenges for practical lightweight applications. Faster R-CNN achieves the highest recognition accuracy but at the cost of the highest computational load. Conversely, YOLOv4, YOLOv6s, and YOLOv8s fall behind our model across all performance indicators. In essence, the FDY-YOLO model introduced in this study exhibits better performance than the other models in terms of accuracy, average precision, and detection speed.

Table 4. Performance comparison of various models on D-Fire database.

Method	P/%	R/%	mAP@0.5/%	GFLOPs
FDY-YOLO	90.1	88.9	92.6	10.6
YOLOv4	91.4	87.4	91.2	154.9
YOLOv5s	91.5	87.5	91.6	16.0
YOLOv6s	90.6	89.1	92.2	13.9
YOLOv7-tiny	90.4	89.2	92.3	13.1
YOLOv8s	88.8	88.1	92.5	28.8
Faster R-CNN	71.4	90.4	93.4	370.2

To visually demonstrate the disparate recognition efficacies exhibited by various algorithms, we selected an indoor fire image from a publicly available dataset, and we present a comparative analysis of their recognition outcomes in Figure 12. As illustrated in Figure 12, the enhanced model presented in this paper demonstrates a recognition performance that is on par with that of conventional detection models. Nevertheless, our model demands the lowest computational overhead. Hence, the improved YOLO approach presented in this paper is well-suited for aircraft cargo hold fire detection scenarios that necessitate high real-time performance while simultaneously imposing limitations on the computational capabilities of the involved hardware processors.

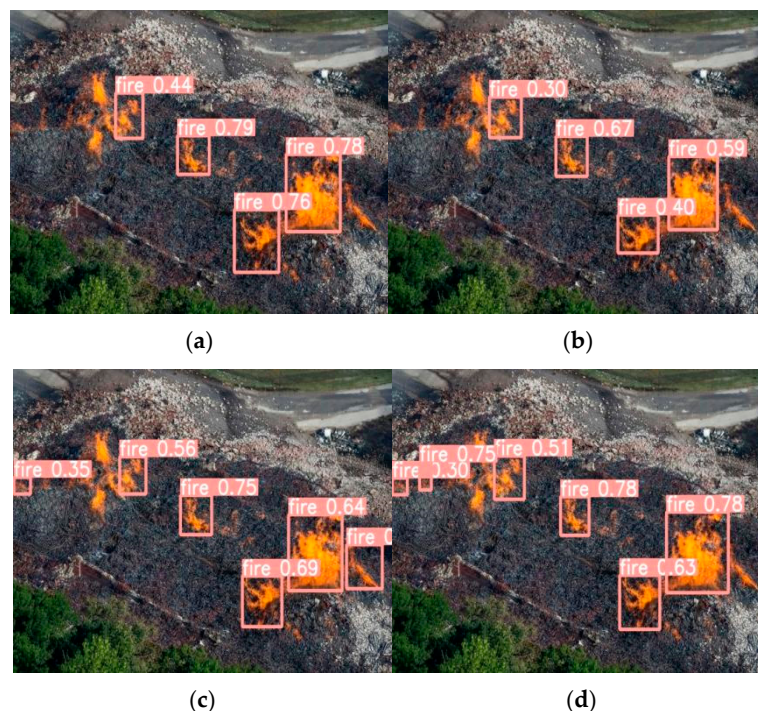


Figure 12. Comparison of detection effects. (a) YOLOv5s, (b) YOLOv8s, (c) FDY-YOLO, and (d) Faster R-CNN.

Due to the generation of a large amount of smoke during a fire, in addition to detecting flame targets, the method proposed in this study also has a good recognition capability for smoke targets. Figure 13 demonstrates the effectiveness of the proposed method in detecting smoke.



Figure 13. Diagram of the calculation process for nearest-neighbor upsampling.

5. Conclusions

Here, we introduced an enhanced fire detection technology named FDY-YOLO, which employs an optimized and lightweight model structure to achieve high-speed and accurate fire detection. Experimental analysis showed that by incorporating point convolution technology, the model can efficiently capture a richer set of features. Additionally, the integration of the FaB-C3 module simplifies the network architecture, substantially decreasing the computational burden of the model. By substituting the DySample module for the upsampling module, we augmented the network's capacity to capture fine-grained target features and improve the detection of irregularly shaped objects, leading to a modest enhancement of 0.3% in the model's mean average precision (mAP). By substituting the CIOU with the MPDIoU loss function, we improved the model's convergence speed and

recognition accuracy. Compared with the original YOLOv5, FDY-YOLO significantly reduces computational effort by 40% while simultaneously increasing the mAP@0.5/% by 0.8%. This demonstrates the efficacy of our approach in balancing accuracy and computational efficiency. Finally, we compared the performance of recognition by our method with other models by using a public fire dataset, and the fire detection method introduced in this study shows distinct advantages. The lightweight nature of the FDY-YOLO model enables it to meet real-time detection requirements, rendering it well-suited for fire detection scenarios in aircraft cargo compartments. Following the detection of fire or smoke, the methodology introduced in this paper relays images, in which flames or smoke are annotated with anchor boxes, to the aircraft's cockpit for visualization, thereby alerting the pilot to undertake firefighting procedures. Hence, this methodology effectively addresses the challenges pertinent to fire detection within such environments. This paper employed RFID technology to enhance the management efficiency of cameras within the cargo hold, reduce equipment maintenance costs, and improve the utilization efficiency of the fire detection system. The proposed method in this paper has limitations in distinguishing fire-like scenarios, such as sunrise or sunset, and there is room for improvement. This is also a direction for our future research.

Author Contributions: Conceptualization, K.W., W.Z. and X.S.; methodology, K.W.; software, K.W.; validation, X.S. and W.Z.; formal analysis, W.Z.; investigation, X.S. and K.W.; resources, W.Z. and K.W.; data curation, X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Some data from this study can be obtained by contacting the corresponding author.

Conflicts of Interest: The authors report no competing interests.

References

1. Ai, H.-z.; Han, D.; Wang, X.-z.; Liu, Q.-y.; Wang, Y.; Li, M.-y.; Zhu, P. Early fire detection technology based on improved transformers in aircraft cargo compartments. *J. Saf. Sci. Resil.* **2024**, *5*, 194–203. [\[CrossRef\]](#)
2. Part, F.A.R. 25: *Airworthiness Standards: Transport Category Airplanes*; Federal Aviation Administration: Washington, DC, USA, 2002; Volume 7.
3. Zhou, Y.; Shi, L.; Li, C.; Zhang, H.; Zheng, R. Scattering Characteristics of Fire Smoke and Dust Aerosol in Aircraft Cargo Compartment. *Fire Technol.* **2023**, *59*, 2543–2565. [\[CrossRef\]](#)
4. Zhang, Q.; Wang, Y.C.; Soutis, C.; Gresil, M. Development of a fire detection and suppression system for a smart air cargo container. *Aeronaut. J.* **2021**, *125*, 205–222. [\[CrossRef\]](#)
5. Krüll, W.; Willms, I.; Zakrzewski, R.R.; Sadok, M.; Shirer, J.; Zelif, B. Design and test methods for a video-based cargo fire verification system for commercial aircraft. *Fire Saf. J.* **2006**, *41*, 290–300. [\[CrossRef\]](#)
6. Bai, Y.; Wang, D.; Li, Q.; Liu, T.; Ji, Y. Advanced Multi-Label Fire Scene Image Classification via BiFormer, Domain-Adversarial Network and GCN. *Fire* **2024**, *7*, 322. [\[CrossRef\]](#)
7. Buriboev, A.S.; Rakhmanov, K.; Soqiyev, T.; Choi, A.J. Improving Fire Detection Accuracy through Enhanced Convolutional Neural Networks and Contour Techniques. *Sensors* **2024**, *24*, 5184. [\[CrossRef\]](#)
8. Qu, N.; Li, Z.; Li, X.; Zhang, S.; Zheng, T. Multi-parameter fire detection method based on feature depth extraction and stacking ensemble learning model. *Fire Saf. J.* **2022**, *128*, 103541. [\[CrossRef\]](#)
9. Yongbo, H.; Wenjie, Z.; Wei, Y.; Yongqing, L. Research on multi-sensor smoke detection method for aircraft cargo compartment. *China Saf. Sci. J.* **2019**, *29*, 43.
10. Kaiyuan, L.; Hongyong, Y.; Tao, C.; Huang, L. Tunable diode laser absorption spectroscopy (TDLAS)-based optical probe initial fire detection system. *J. Tsinghua Univ. (Sci. Technol.)* **2023**, *63*, 910–916.
11. Wu, A.; Li, M.; Chen, Y. Research for image fire detection technology in large space. *Comput. Meas. Control* **2006**, *14*, 869–871.
12. Fang, S.; Qi, L.; Yu, L. Video smoke detection with multi-feature analysis. *Comput. Eng. Appl.* **2016**, *52*, 222–227.
13. Wang, Y.; Hua, C.; Ding, W.; Wu, R. Real-time detection of flame and smoke using an improved YOLOv4 network. *Signal Image Video Process.* **2022**, *16*, 1109–1116. [\[CrossRef\]](#)

14. Zheng, H.; Duan, J.; Dong, Y.; Liu, Y. Real-time fire detection algorithms running on small embedded devices based on MobileNetV3 and YOLOv4. *Fire Ecol.* **2023**, *19*, 31. [\[CrossRef\]](#)
15. Chen, D.; Xing, W.; Zengshou, D.; Yilei, W.; Zhonghao, J. Improved YOLOv5s Flame and Smoke Detection Method for Underground Garage. *J. Comput. Eng. Appl.* **2024**, *60*, 298.
16. Wang, A.; Liang, G.; Wang, X.; Song, Y. Application of the YOLOv6 combining CBAM and CloU in forest fire and smoke detection. *Forests* **2023**, *14*, 2261. [\[CrossRef\]](#)
17. Chen, X.; Xue, Y.; Hou, Q.; Fu, Y.; Zhu, Y. RepVGG-YOLOv7: A modified YOLOv7 for fire smoke detection. *Fire* **2023**, *6*, 383. [\[CrossRef\]](#)
18. Titu, M.F.S.; Pavel, M.A.; Michael, G.K.O.; Babar, H.; Aman, U.; Khan, R. Real-Time Fire Detection: Integrating Lightweight Deep Learning Models on Drones with Edge Computing. *Drones* **2024**, *8*, 483. [\[CrossRef\]](#)
19. Chen, J.; Kao, S.-h.; He, H.; Zhuo, W.; Wen, S.; Lee, C.-H.; Chan, S.-H.G. Run, don't walk: Chasing higher FLOPS for faster neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 12021–12031.
20. Liu, W.; Lu, H.; Fu, H.; Cao, Z. Learning to upsample by learning to sample. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 6027–6037.
21. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20), New York, NY, USA, 12 February 2020; Volume 4, pp. 12993–13000.
22. Ma, S.; Xu, Y. Mpdious: A loss for efficient and accurate bounding box regression. *arXiv* **2023**, arXiv:2307.07662.
23. Shen, E.; Duan, S.; Guo, S.; Yang, W. Object Localization and Sensing in Non-Line-of-Sight Using RFID Tag Matrices. *Electronics* **2024**, *13*, 341. [\[CrossRef\]](#)
24. Ali, M.; Hendriks, P.; Popping, N.; Levi, S.; Naveed, A. A Comparison of Machine Learning Algorithms for Wi-Fi Sensing Using CSI Data. *Electronics* **2023**, *12*, 3935. [\[CrossRef\]](#)
25. Wang, L.; Luo, Z.; Guo, R.; Li, Y. A Review of Tags Anti-Collision Identification Methods Used in RFID Technology. *Electronics* **2023**, *12*, 3644. [\[CrossRef\]](#)
26. Xie, S.; Ma, C.; Feng, R.; Xiang, X.; Jiang, P. Wireless glucose sensing system based on dual-tag RFID technology. *IEEE Sens. J.* **2022**, *22*, 13632–13639. [\[CrossRef\]](#)
27. Feng, R.H.; Li, J.H.; Xie, S.; Mao, X.R. Efficient Training Method for Memristor-Based Array Using 1T1M Synapse. *IEEE Trans. Circuits Syst. II-Express Briefs* **2023**, *70*, 2410–2414. [\[CrossRef\]](#)
28. Feng, R.; Xiang, X.; Xie, S.; Jiang, P. Sensing System for Mixed Inorganic Salt Solution Based on Improved Double Label Coupling RFID. *IEEE Sens. J.* **2023**, *23*, 13565–13574. [\[CrossRef\]](#)
29. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
30. de Venâncio, P.V.A.; Lisboa, A.C.; Barbosa, A.V. An automatic fire detection system based on deep convolutional neural networks for low-power, resource-constrained devices. *Neural Comput. Appl.* **2022**, *34*, 15349–15368. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.