fire

MDPI

Article

A Flame-Detection Algorithm Using the Improved YOLOv5

Xingang Xie [1], Ke Chen [2], Yiran Guo [2], Botao Tan [2], Lumeng Chen [2,*] and Min Huang [2,*]

[1] School of Mechatronics and Information Engineering, China University of Mining and Technology (Beijing), Beijing 100083, China; BQT2000401003@student.cumtb.edu.cn
[2] School of Mechanical Engineering, Beijing Information Science and Technology University, Beijing 100192, China; changku2021@163.com (K.C.); guoj.cn@163.com (Y.G.); t403905443@163.com (B.T.)
* Correspondence: Lu-100@163.com (L.C.); huangmin@bistu.edu.cn (M.H.)

Abstract: Flame recognition is an important technique in firefighting, but existing image flame-detection methods are slow, low in accuracy, and cannot accurately identify small flame areas. Current detection technology struggles to satisfy the real-time detection requirements of firefighting drones at fire scenes. To improve this situation, we developed a YOLOv5-based real-time flame-detection algorithm. This algorithm can detect flames quickly and accurately. The main improvements are: (1) The embedded coordinate attention mechanism helps the model more precisely find and detect the target of interest. (2) We advanced the detection layer for small targets to enhance the model's associated identification ability. (3) We introduced a novel loss function, α-IoU, and improved the accuracy of the regression results. (4) We combined the model with transfer learning to improve its accuracy. The experimental results indicate that the enhanced YOLOv5's mAP can reach 96.6%, 5.4% higher than the original. The model needed 0.0177 s to identify a single image, demonstrating its efficiency. In summary, the enhanced YOLOv5 network model's overall efficiency is superior to that of the original algorithm and existing mainstream identification approaches.

Keywords: flame recognition; artificial intelligence; real-time detection; YOLOv5; boundary loss function

Citation: Xie, X.; Chen, K.; Guo, Y.; Tan, B.; Chen, L.; Huang, M. A Flame-Detection Algorithm Using the Improved YOLOv5. *Fire* **2023**, *6*, 313. https://doi.org/10.3390/fire6080313

Academic Editor: Khan Muhammad

Received: 29 June 2023
Revised: 9 August 2023
Accepted: 11 August 2023
Published: 13 August 2023

1. Introduction

In this period of rapid economic development and urbanization, fire has become one of the main disasters that threatens people's property and safety, with the potential to cause serious economic losses and casualties. Despite the increasing sizes of urban buildings, the main fire-extinguishing method at the scene is still manual fire extinguishing by firefighters. However, firefighters are often injured or even lose their lives in the process. Therefore, introducing firefighting robots to replace manual firefighting will gradually become a trend. With firefighting robots, accurate and real-time detection of flames will be the key to smooth firefighting. At present, traditional fire-detection methods have the disadvantages of slow response speeds, depending on a single detection approach, and low accuracy. As such, traditional detection cannot meet firefighting robots' requirements for a real-time and accurate approach.

Traditional flame detection techniques are mainly used to extract features. Celik et al. proposed the use of an RGB color space to detect flames with different characteristics for the three RGB channels [1]. Liu et al. used the characteristics of red and blue light generated when a flame burns and used the YCbCr color space with brightness and red and blue color information to extract the flame features and perform flame detection [2]. Song et al. used the frame difference approach, with the area growth ratio as the basis for judging changes over frames [3]. To a certain extent, traditional flame-identification methods meet the requirements for flame detection. However, in a complex urban environment, there are abundant problems like identification precision, a high false detection rate, and an inability for real-time identification.

In recent years, deep learning (DL) networks based on image processing have developed rapidly, and flame recognition based on machine vision has gradually become a trend. Machine vision has the advantages of high recognition accuracy and fast recognition speed. Current object recognition methods are mainly divided into two categories. One class is to first generate a pre-selected box for the region that may contain the detected object, and then combine it with the two-stage method of CNNs for sample classification. DL methods, represented by convolutional neural networks (CNNs), can effectively improve recognition accuracy and speed [4,5]. Common algorithms include R-CNN [6], Faster R-CNN [7], SPP-Net [8], and so on. Zhong et al. implemented CNN-based video flame detection [9]. Zhang et al. proposed an improved Faster R-CNN flame recognition method, which effectively improves detection accuracy by using deep networks [10]. Yu et al. added a bottom-up feature pyramid to Mask R-CNN to improve flame detection accuracy [11]. Fires develop extremely quickly, especially in forest environments where there is a lot of flammable material. If the flame detection algorithm fails to detect the flame in the first instance, it may miss the best time to extinguish the fire, leading to rapid spread and greater damage. However, CNN networks usually contain a large number of neurons and parameters. For a large image and video data, CNNs need to perform a large number of computations, including operations such as convolution, pooling, and full connectivity. This can lead to processing delays that are not favorable for real-time flame detection. Another class is the use of one-step methods for sample classification; common algorithms include YOLO series [12], SSD [13], EfficientDet [14], etc. Abdusalomov et al. proposed a fire detection method based on YOLOv3 [15]. Zheng et al. proposed a fire detection method based on MobileNetV3 and YOLOv4 [16]. However, in the forest flame detection scenario, the size and shape of forest flames in an image may vary greatly; some flames may be very large and some may be very small, and one-step algorithms usually put more emphasis on the speed of operation, while in some cases accuracy may be sacrificed. Therefore, although one-step algorithms work well for many target detection tasks, further improvements and optimizations may be needed for forest flame detection to increase the accuracy of detection.

To effectively solve the mentioned problems, we adopt an enhanced YOLOv5 approach based on YOLOv4 [17]. The YOLOv5 approach has the characteristics of high precision and high speed in image detection, but it has poor identification ability for small targets. Nonetheless, the work in this paper focuses on making improvements to solve the poor small-target identification problem using the YOLOv5 algorithm. We greatly improve the detection accuracy for a small flame area while ensuring the image flame-detection speed. The improved model is compared to the original one, and the results indicate that its power is greatly increased, which proves its effectiveness.

2. Enhancement of the YOLOv5 Network Framework

2.1. Introduction to the YOLO Algorithm

Before YOLOv1 was proposed, the R-CNN series of algorithms came out on top in target identification. However, despite the high identification accuracy of the R-CNN series, the network structure uses a two-step method, which means that the detection speed cannot attain real-time efficiency. In 2016, Redmon et al. [18] presented a single-step object-identification network with fast recognition ability, processing 45 frames per second and easily executable in real-time. The main idea of YOLO is to convert target identification into a regression problem and then employ the whole image as the network input to obtain the position of the bounding box and its corresponding group via the neural network [19]. By developing YOLOv1, Redmon et al. made significant improvements and proposed YOLOv2, in which the k-means clustering approach was utilized to attain a better anchor template in the training set. This effectively improved the algorithm's recall rate. Combined with the image's fine-grained characteristics, the shallow characteristics were combined with the deep ones to enhance the identification of small-sized objects. YOLOv3 was based on YOLOv2, but its feature-extraction setup adopted the Darknet-53 network structure,

replacing the original Darknet-19. A feature pyramid network framework was utilized to effectively achieve multi-scale identification. The classification approach was logistic regression rather than softmax. While considering the real-time efficiency, it also effectively guaranteed the precision of target identification [20]. YOLOv4 retained the head of YOLOv3 and combined the original Darknet-53 with CSPNet [21]. The trunk component stacked the original residual blocks, and the branch component was equivalent to a residual edge, which was directly linked to the end after a little processing. The Mish activation function was utilized rather than the original ReLu. The idea of SPPF [22] was used to extend the receptive field and isolate the essential contextual characteristics. This study mainly used PANet [23], instead of the original FPN, as a parameter aggregation approach. For various detector levels, parameters were aggregated from various backbone layers.

YOLOv5, as the latest network structure of the YOLO series, comprises four components: input, backbone, neck, and detect. Figure 1 presents its network framework.

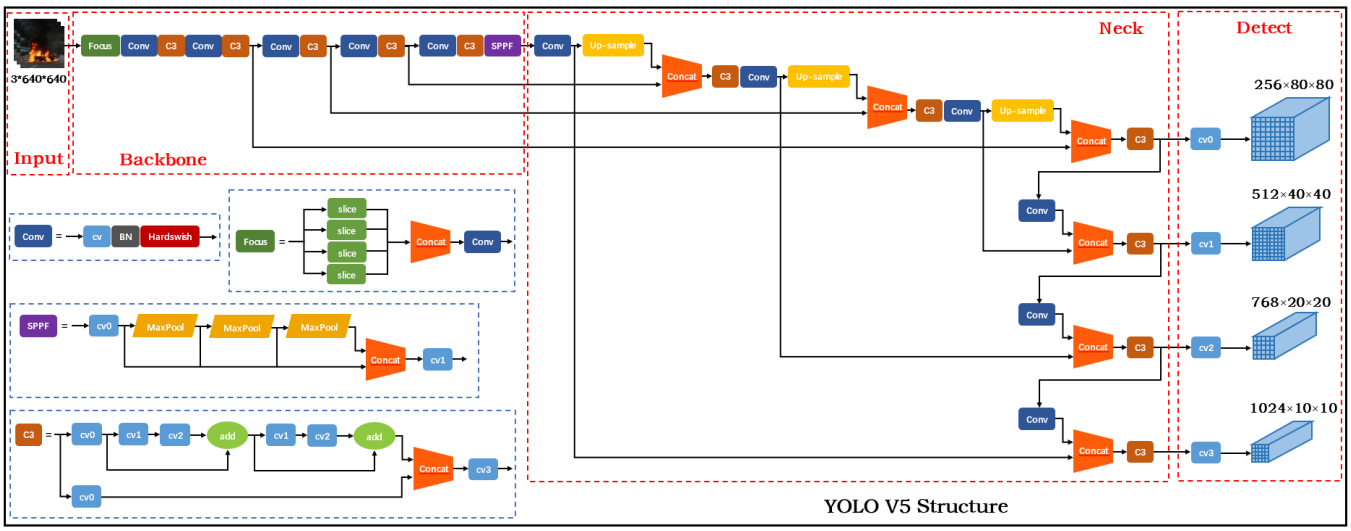


Figure 1. Network structure diagram for YOLOv5.

2.1.1. Input

The YOLOv5 input adopts a similar mosaic data-improvement approach as YOLOv4. Stitching by random scaling, cropping, and arrangement can effectively enhance the identification impact for small targets. In addition, YOLOv5 adds an adaptive anchor box calculation. In different datasets, the optimal anchor box value is calculated adaptively, and YOLOv5 adaptively adds the least black borders to the original image. At the height of the image, the black borders at both ends are reduced. The identification time is effectively reduced in the target identification process.

2.1.2. Backbone

YOLOv5 adds a focus framework to the network (shown in Figure 2). The most critical step in the focus framework is slicing. A $4 \times 4 \times 3$ image is transformed into a $2 \times 2 \times 12$ feature map after slicing. In the YOLOv5 network, the original $3 \times 640 \times 640$ image is fed into the focus framework. After the slicing operation, it first becomes a $12 \times 320 \times 320$ feature map and becomes a $64 \times 320 \times 320$ feature map (Figure 3) through the convolution operation of 64 convolution kernels.

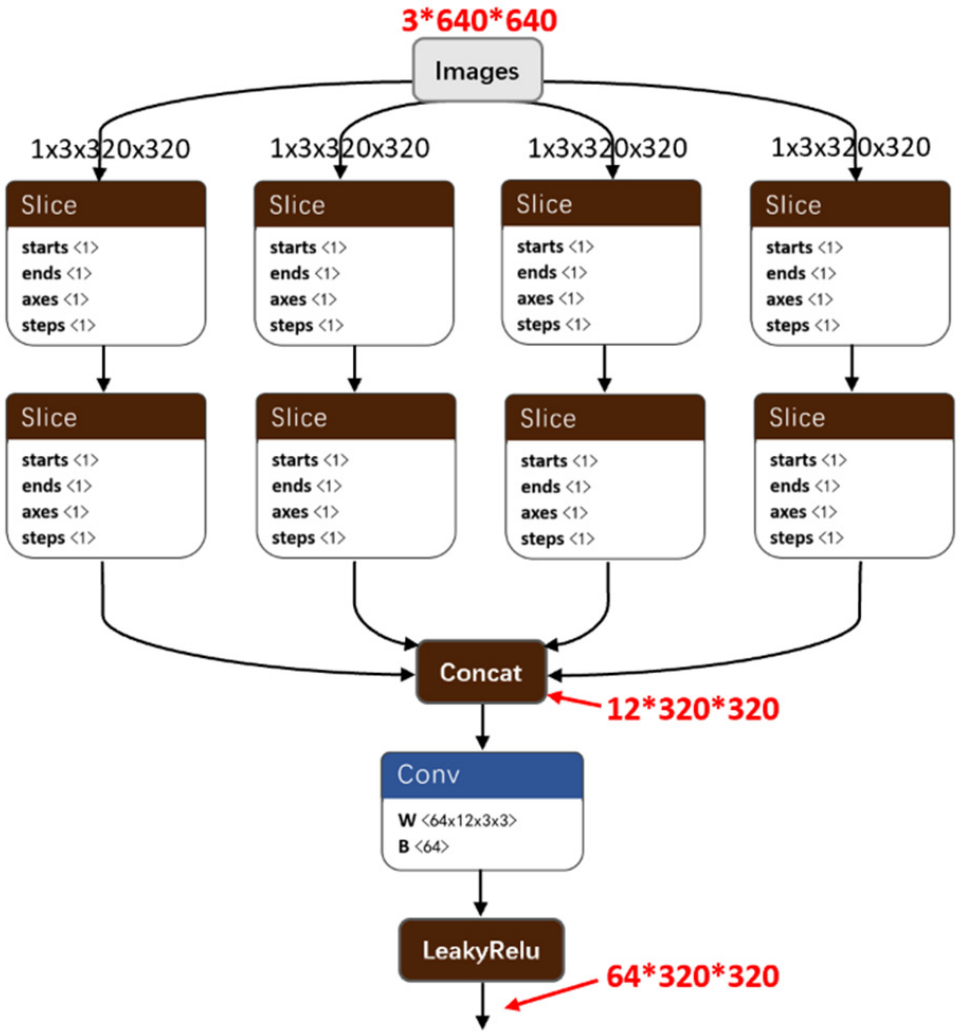


Figure 2. Focus schematic.

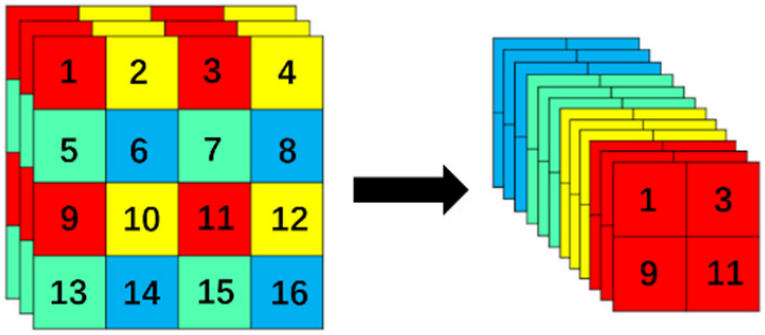


Figure 3. Schematic diagram of slice operation.

2.1.3. Neck

Both YOLOv5 and YOLOv4 adopt the FPN + PAN framework, as presented in Figure 4, which contains two PAN structures. The FPN layer transforms powerful semantic characteristics from top to bottom, and the characteristic pyramid transforms strong localization characteristics from the bottom to the top. The combined operation is used to effectively aggregate the features of the identification layer from various backbone layers to promote the feature-extraction capability.

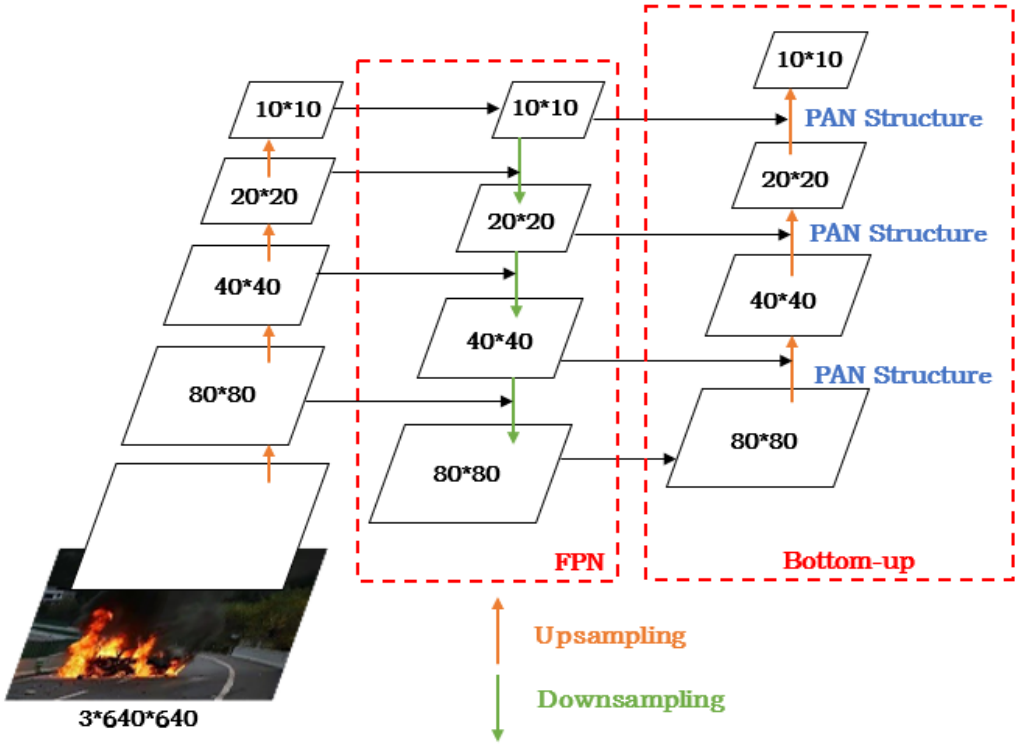


Figure 4. The FPN and PAN frameworks of YOLOv5.

2.1.4. Output

YOLOv5 uses CIoU_Loss as the bounding box's loss function. In post-processing target identification, it is usually crucial to employ nms to screen the target frame. Based on DIOU_LOSS, YOLOv4 used DIoU nms, which is not sufficient for YOLOv5, so instead, it uses weighted nms.

2.2. YOLOv5 Algorithm Optimization

2.2.1. Adding a Small Target-Detection Layer

In the original model of YOLOv5, there are only four detection layers, which are 80×80, 40×40, 20×20, and 10×10. The 80×80 detection layer is utilized to recognize targets with a size of 8×8 or more, the 40×40 detection layer is utilized to identify targets with a size of 16×16 or more, the 20×20 detection layer is used to identify targets above 32×32, and the 10×10 detection layer is used to recognize targets above 64×64. These detection layers are subjected to six down-sampling operations of the YOLOv5 network, and then four feature maps are obtained: 10×10, 20×20, 40×40, and 80×80. The 80×80 feature map is mainly employed to identify small targets, corresponding to 640×640, and each feature map's receptive field is $640/80 = 8 \times 8$. If the width or height of the target in the original image is smaller than 8 pixels, some information will be lost after layer-by-layer convolution. As a result, the shallow special information cannot be fully utilized. Furthermore, the neural network cannot learn the target's feature information, leading to low detection accuracy for a small flame area. To fully enhance the network's capability to fuse multi-scale characteristics, we added a 160×160 small target identification layer, which was mainly used to detect targets above 4×4. To increase the small target detection ability, several feature extraction layers were specially set up. After the 24th layer, we performed upsampling and other methods of processing on the feature map so that it continued to expand; at the 26th layer, the acquired 160×160 feature map was concatenated and fused with the second layer feature map in the backbone network. Larger feature maps can be attained for effective small object identification. Figure 5 presents the enhanced network framework.

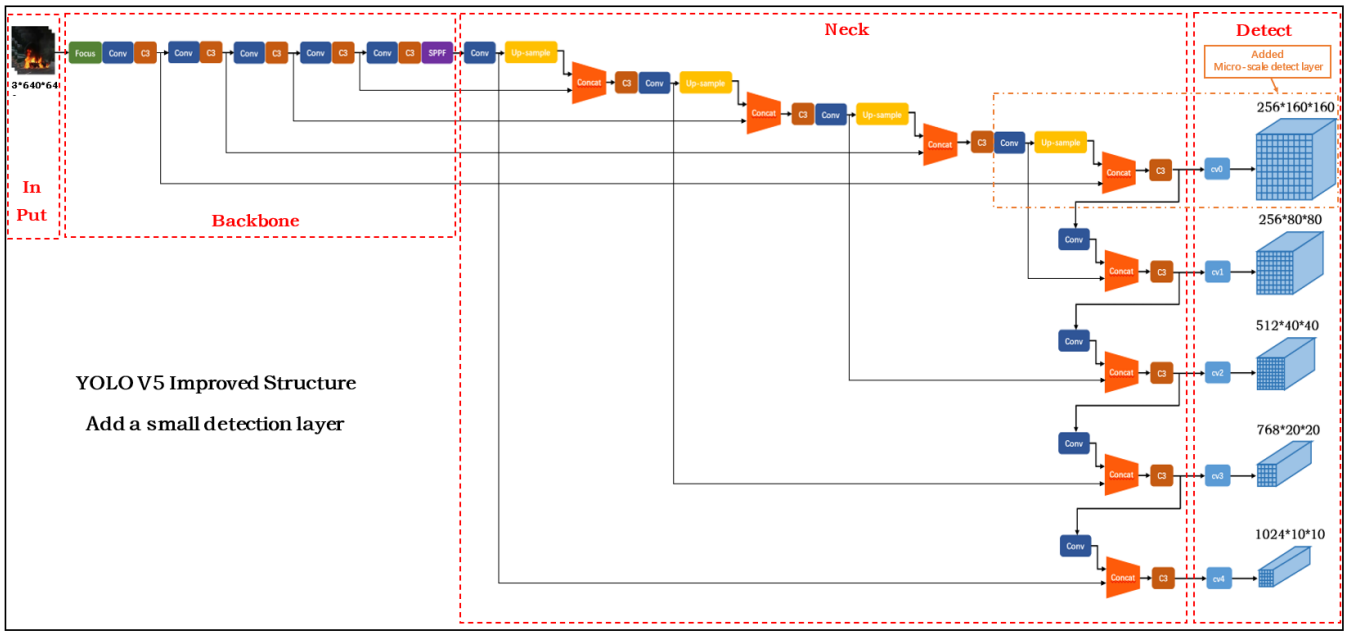


Figure 5. Network framework after adding the micro-scale detection layer.

2.2.2. Increase Attention Mechanism

Recently, attention mechanism modules have been utilized on a large scale in computer vision. The attention mechanism aims to find the information of interest and eliminate ineffective information. Most attention mechanisms are used in deep neural networks, which can lead to performance improvements. Currently, the commonly used attention mechanisms are SE [24], BAM [25], and CBAM [26]. Nevertheless, SE only assumes the internal channel information while neglecting the necessity of position information, although the target's spatial framework in vision is very important. BAM and CBAM try to present location information by global pooling across channels. Nevertheless, this method can only capture local information and cannot attain long-range dependency information. Therefore, this study introduces a flexible and lightweight attention mechanism (coordinate attention) into the method [27]. Coordinate Attention (CA) is a novel attention mechanism presented by Hou et al., where embedding location information into the channel attention allows the neural network to attain information from a broader area and reduces the computing power requirement. The CA module mainly sustains a channel relationship and long-range encoding via accurate location information. It is categorized into two stages: coordinate information embedding and coordinate attention production, as presented in Figure 6.

For the feature map X produced through the previous layer of convolution, all channels are separately encoded along with the horizontal and vertical coordinates by an average pooling kernel of size $(H, 1)$ and $(1, W)$. The following relation represents the output of the cth channel with height h and the cth channel with width w:

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i < W} x_c(h, i) \tag{1}$$

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le j < H}^{0} x_c(j, w)_0^0 \tag{2}$$

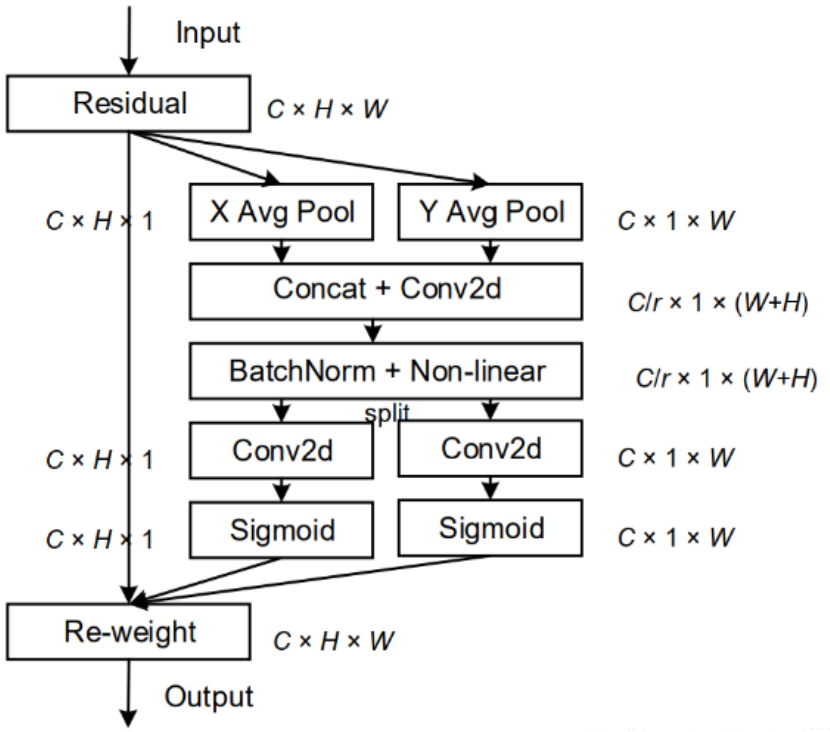


Figure 6. Mechanistic framework of coordinate attention.

The above transformation performs feature aggregation in two spatial orientations, returning a pair of orientation-aware attention maps. The two feature maps z^h and z^w generated by concatenation employ a shared 1×1 convolution operation F_1. Then, the intermediate feature f of the spatial information in the horizontal and vertical orientations is obtained through the following relation:

$$\mathbf{f} = \delta\left(F_1\left(\left[\mathbf{z}^h, \mathbf{z}^w\right]\right)\right) \tag{3}$$

The intermediate feature **f** is categorized into two separate tensors, f^h and f^w, along with the spatial dimension. Feature maps f^h and f^h are converted into a similar number of channels as channel x by 1×1 convolution. The formulas are:

$$\mathbf{g}^h = \sigma\left(F_h\left(\mathbf{f}^h\right)\right) \tag{4}$$

$$\mathbf{g}^w = \sigma(F_w(\mathbf{f}^w)) \tag{5}$$

f^h and f^w are extended by the sigmoid activation function. Taking g^h and g^w as the attention weights, the final formula for the CA module can be obtained as follows:

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{6}$$

CA decomposes channel attention into two one-dimensional feature-encoding processes aggregating features along 2 spatial orientations. Accordingly, long-range dependencies can be captured along a spatial orientation, while accurate location information can be maintained along the other. The produced feature maps are encoded as direction-aware and position-sensitive attention maps, respectively, which can be applied complementary to the input feature maps to improve the description of objects of interest. We added coordinate attention to the backbone network. Figure 7 presents the enhanced network framework.

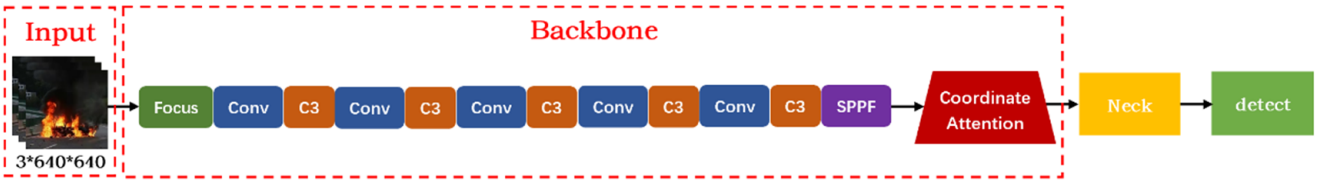


Figure 7. Network framework after adding an attention mechanism.

2.2.3. Boundary Loss Function

The full name of IoU [28] is the intersection over union employed to calculate the ratio of the intersection and union of the "predicted bounding box" and the "true bounding box". IoU is an important function of mAP calculation of the object-detection algorithmic performance. It is a precision measure when identifying the corresponding objects in a given dataset. When the predicted bounding box is closer to the ground truth bounding box, the IoU is closer to 1. By continuously reducing the loss, the model obtains better prediction results. However, IoU does not assume the distance between boxes, and it has corresponding drawbacks when employed as a loss function. For example: if the two borders do not overlap, the IoU is 0, and no gradient will be returned at this time; multiple iterations are required, and learning cannot be performed. Loss is only related to the intersection ratio and intersection area of the two boxes. Therefore, a phenomenon of the same intersection area and different coincidence degrees will occur, as shown in Figure 8.

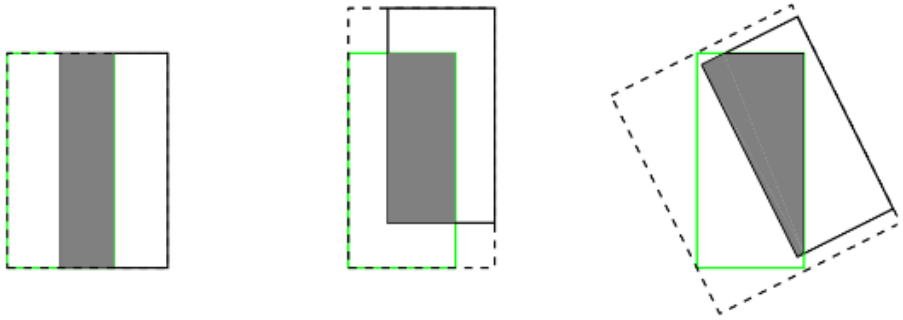


Figure 8. Three various kinds of overlap between two rectangles with similar IoU values.

To effectively resolve the mentioned drawbacks, Rezatofighi et al. presented the Generalized Intersection over Union (GIoU) [29]. For any two boxes, A and B, the smallest box C is found that can enclose them, and the ratio of the area of C\(A ∪ B) to the area of C is obtained. Note: The area of C\(A ∪ B) can be obtained by subtracting the area of A ∪ B from the area of C. The ratio is subtracted from the IOU values of A and B to obtain GIoU. The following formula calculates the GIoU:

$$GIoU = IoU - \frac{C - (A \cup B)}{C} \tag{7}$$

GIoU still has some problems. At first, with GIoU, it is necessary to make the detection result intersect with the target frame, then start to reduce the detection result to coincide with the GT. This results in the need for more iterations to converge, especially in respect of horizontal and vertical boxes. Therefore, Zheng et al. proposed DIoU [30] and CIoU [31], the DIoU formula can be described as follows:

$$DIoU = 1 - IoU + \frac{\rho^2(A, B)}{c^2} \tag{8}$$

$d = \rho(A, B)$ describes the Euclidean distance between the coordinates of the center point of the A and B frames. c indicates the diagonal distance of the smallest box that encloses them. The penalty term for DIoU is calculated according to the ratio of the center point distance to the diagonal distance. This avoids the generation of a larger outer frame when the two frames are far apart, such as in GIoU. The loss value is large, and it is difficult to optimize the problem. Therefore, the convergence rate of DIoU is higher than GIoU loss. In the calculation of DIoU, only the center point distance and overlapping area are considered; the aspect ratio is not considered. Therefore, Zheng et al. proposed CIoU based on DIoU. Compared to GIoU, CIoU adds penalty items for the aspect ratio, including a and v (a represents the weight function, such as in Formula (9); v is utilized to determine the aspect ratio's similarity, such as in Formula (10)). CIoU can converge quickly by assuming the overlapping area, center point distance, and aspect ratio. Even if the predicted box is included in the real box, it still has an accurate convergence rate. The CIoU loss function is presented in Formula (11):

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{9}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{10}$$

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{11}$$

He et al. introduced power transformation based on IoU loss and proposed a novel IoU loss function, α-IoU [32]. Setting α gives the detector more flexibility in attaining various levels of box regression precision. α-IoU is more robust against small datasets and noise. Equation (12) describes the α-IoU loss function:

$$L_{\alpha - IoU} = \frac{1 - IoU^{\alpha}}{\alpha}, \alpha > 0 \tag{12}$$

The equation applies to lightweight models. Therefore, this paper adopts the α-IoU loss function, $\alpha = 3$.

2.3. Network Training

2.3.1. Training Device

The experimental platform used personal desktops (Intel® Core™ i9 11900 k CPU, 128 GB running memory; NVIDIA® GeForce RTX 3090 GPU, 24 GB video memory). In order to perform this research, a PyTorch DL structure was constructed on a Windows 10 operating system. We used Python to write program code and call up libraries like CUDA, CUDNN, and OpenCV. The software environment was CUDA 10.1, CUDNN 7.6, and Python3.8. Accordingly, the firefighting drone flame-detection model was learned and evaluated efficiently.

The original and enhanced YOLOv5 were learned separately. The parameter settings were as follows: the maximum number of iterations was 600, the batch size was 64, the momentum factor was 0.937, and the weight decay rate was 0.0005. The enhancement coefficients of hue (H), saturation (S), and brightness (V) were 0.015, 0.7, and 0.4, respectively. After training, we saved the established recognition model's weight file. We combined it with the test set to verify the model's efficiency.

2.3.2. Data Acquisition and Preprocessing

In 2021, the Nansha Fire Brigade in Guangzhou collected 20,000 different flame images, which we used in this research. In order to enhance the training efficiency and increase the sample diversity, the acquired image data were monitored before training. This was combined with labeling to process the images, then we saved the processed images in JPG format with a resolution of 640 × 640. In addition, to effectively improve the network learning ability, we used a data-augmentation method to promote the network model's generalizability, and we selected three methods: image rotation, image flipping, and brightness balance. Rotating and flipping images can effectively enhance the network's identification efficiency and robustness. Brightness balancing removes the effects of ambient lighting variations and brightness deviations caused by sensor differences on the network performance. After data augmentation, 13,733 images were acquired and used as the training set. We randomly selected 600 flame pictures and 300 non-flame pictures as the validation set. Then, 300 unlabeled flame pictures were chosen as the test set. Figure 9 presents the data-augmentation results.

Figure 9. Data augmentation.

2.3.3. Transfer Learning

Transfer learning is often used in machine learning; it refers to the further application of knowledge or patterns trained in a specific area or task to various but relevant areas or problems. The key in this study was to train the model and transfer the results to the YOLOv5 network to finally complete the flame target-identification. Since the flame target data in this paper were very limited, transfer learning was also used to initialize the YOLOv5 network. In doing so, we guaranteed the successful transfer of the trained knowledge and enhanced the ability of the novel network to learn rapidly. This can enhance the over-fitting problem caused by insufficient flame datasets to a certain extent. The generalizability of flame target-identification is also effectively improved, which promotes the establishment of a recognition model. This gives it a good transfer learning recognition ability, even in complex fire situations. In addition, in image DL, there are various datasets that are applicable to different fields, and it is necessary to analyze the datasets in-depth. We selected the most commonly and widely used dataset, ImageNet, which has an excellent efficiency in image classification, identification, localization, and other areas. The improved YOLOv5 neural network was pre-trained using the ImageNet dataset. During the pre-

training process, the network learned to extract generic image features by performing backpropagation and parameter updates on the ImageNet dataset. After the pre-training was completed, the weights of the model were saved and the pre-trained weights were loaded into the YOLOv5 model as initial weights to flame recognition model training.

2.3.4. Model Verification Metrics

The current work utilized objective verification indicators such as identification precision and speed to verify the efficiency of the trained target-recognition model. Frames Per Second (FPS) is an identification speed measure; True Positives (TP) indicates the number of truly identified flame targets; False Positives (FP) indicates the number of lights or shadows detected as flame targets; and False Negatives (FN) indicates the number of unidentified flame targets. If the IoU obtained by the predicted and ground-truth flame boxes exceeded 0.5, the identification box was indicated by TP; otherwise, it was marked as FP. If the detected real flame target did not match the corresponding prediction frame, it was marked as FN. The following relations can be employed to compute precision and recall:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{13}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{14}$$

There was no interplay between precision and recall. Therefore, to better evaluate the detection accuracy, we introduced mAP to represent the detection accuracy, where m represented the average and AP referred to integrating the P index to the *R* index in the range 0–1, which was the area under the *P*-*R* curve. The greater the AP, the higher the network accuracy. The AP and mAP calculation formulas are as follows:

$$\text{AP} = \int_0^1 \text{Pr}(\text{Re})\text{dRe} \tag{15}$$

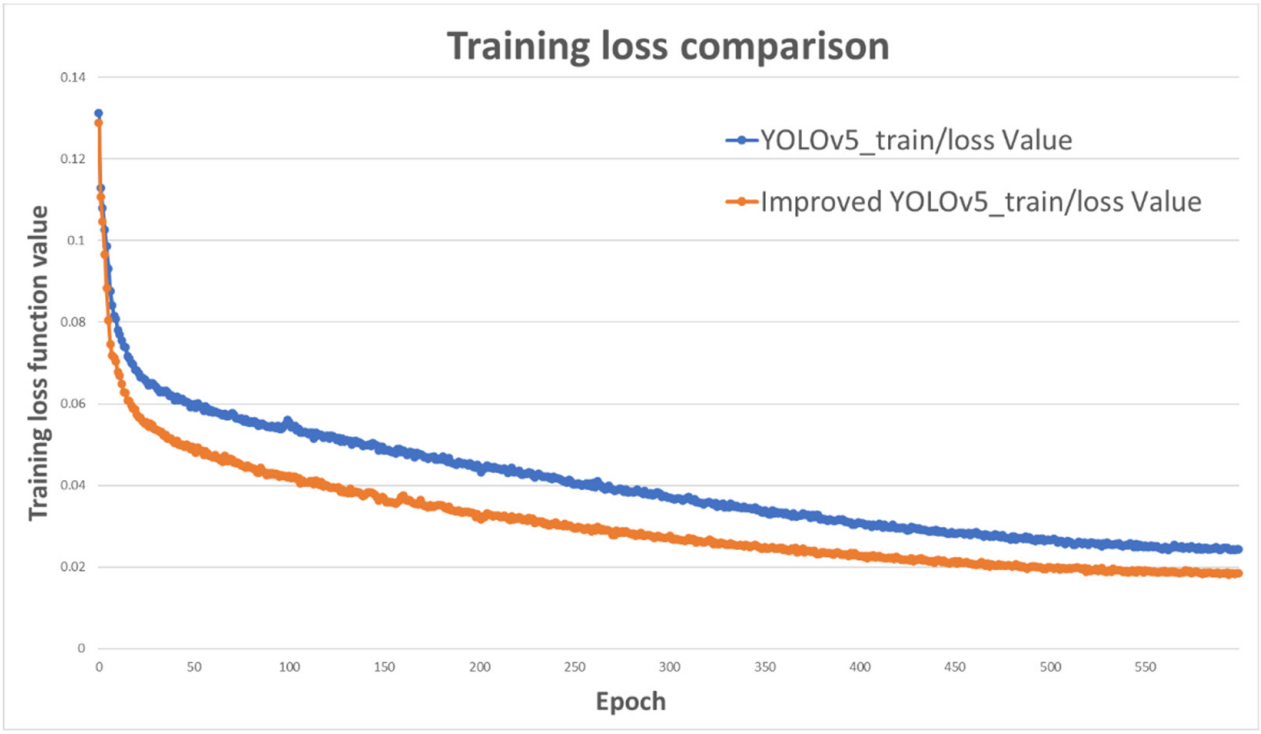
$$\text{mAP} = \frac{1}{C}\sum_{K=i}^{N} P(k)\Delta R(k) \tag{16}$$

3. Results

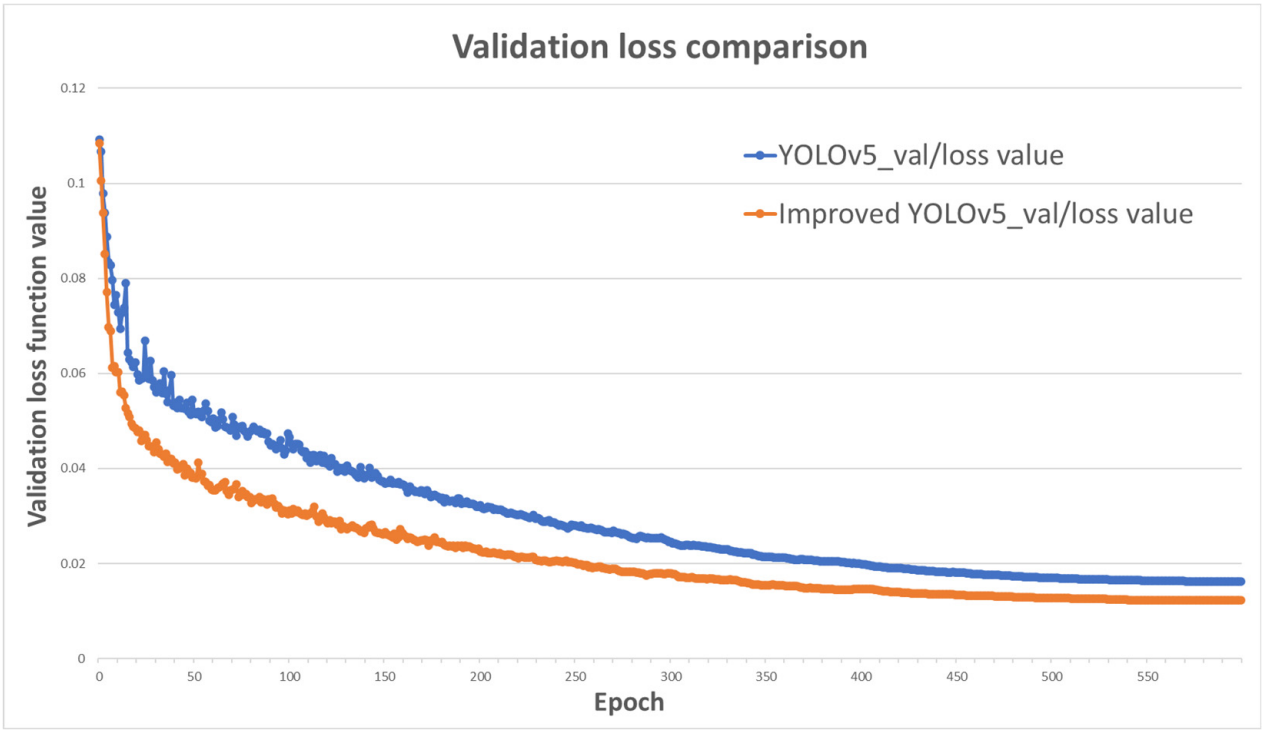
3.1. Experimental Results

We combined the loss function curve and the mean precision to judge the detection model's quality. In the network training process, the loss function can intuitively indicate whether the network model converges stably with increased iterations. Figure 10 presents the model loss function. Experiments showed that the convergence rate of the loss function of the improved YOLOv5 approach is higher than the original YOLOv5 algorithm. Moreover, when the improved YOLOv5 algorithm iterated 300 times in the model, the loss value was close to 0, meaning the network basically converged.

The highest accuracy was obtained using the enhanced YOLOv5 algorithm. The mAP value was utilized to judge the flame-identification model's quality. The higher the mAP value, the higher the identification precision and the more superior the network efficiency. When the threshold was 0.5, the predicted value, recall rate, mAP value, and fps of the improved YOLOv5 algorithm were 85.7%, 94.8%, 96.6%, and 68, respectively. In addition, the predicted value, recall rate, mAP value, and fps of the original YOLOv5 were 84.2%, 89.7%, 91.2%, and 71, respectively. As shown in Figure 11, when the improved YOLOv5 was iterated 300 times, the AP value reached 94% and tended to be stable, and the final maximum value reached 96.6%.



(**a**) Training loss comparison



(**b**) Validation loss comparison

Figure 10. Comparison of (**a**) training and (**b**) validation loss functions.

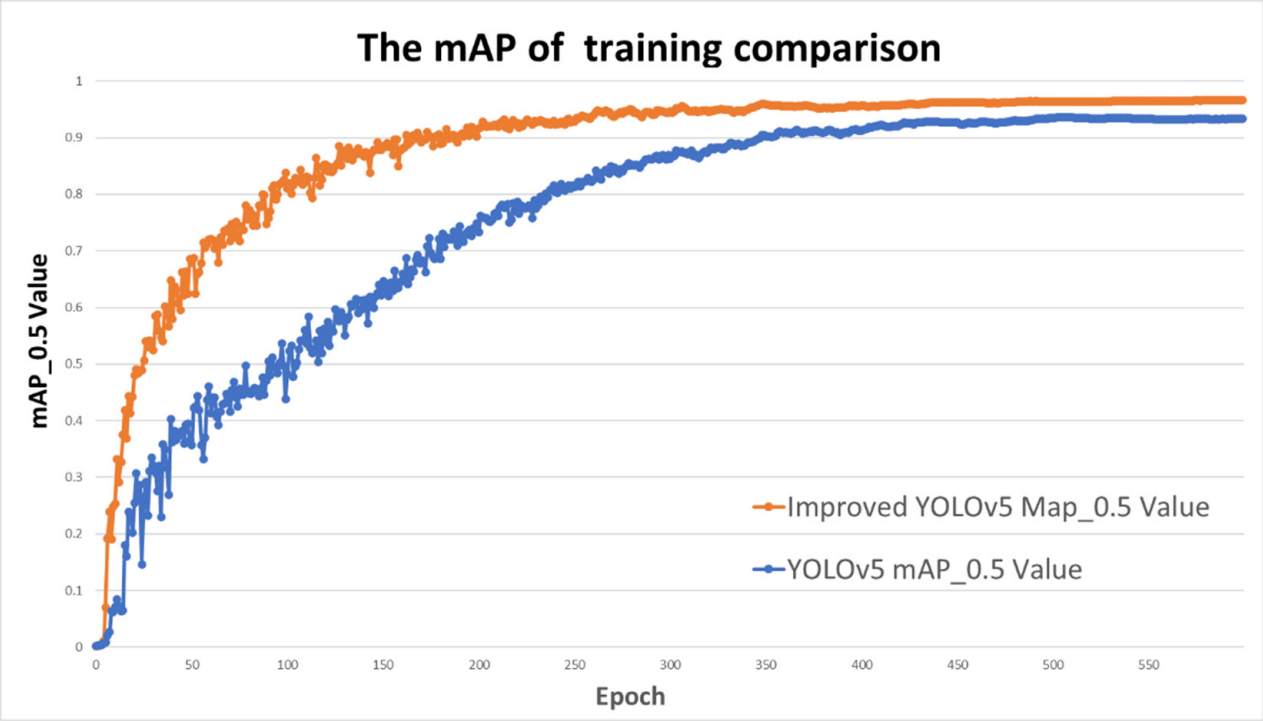


Figure 11. The mAP value comparison.

The above experiments fully demonstrated the efficiency of the improved network. The accurate detection of flames (especially small target flames) was achieved. Figure 12 presents the recognition results.

Figure 12. Recognition results.

3.2. Comparison of Recognition Results for Various Target-Identification Approaches

In order to comprehensively test the accuracy of the improved YOLOv5 algorithm for flame detection, a test was conducted to compare the proposed method with YOLOv3-5 based on 300 images used in this experiment using the same initialized weights. mAP values and FPS were the main validation metrics. The test results for the four approaches are presented in Table 1. The data in Table 1 indicate that the improved YOLOv5 had the maximum detection precision compared to the other three approaches. Compared to the YOLO series of algorithms, the algorithm in this paper was improved by 10.9%, 3.9%, and 5.4%, respectively, compared to YOLOv3, YOLOv4, and YOLOv5s.

Table 1. Comparison of recognition results for various algorithms.

Algorithm	mAP (%)	FPS	Model Size (MB)
YOLOv3	85.7	52	235
YOLOv4	92.7	54	244
YOLOv5s	91.2	71	14
The improved YOLOv5	96.6	68	25.1

To better compare the performance of the improved YOLOv5, we decided to compare some algorithms from the past few years that use the YOLO family and other methods for flame detection, as shown in Table 2.

Table 2. Comparison of recognition results for various algorithms.

Algorithm	mAP (%)
Lightweight YOLOv4 [33]	58
Reference [34] algorithm	94.5
Reference [35] algorithm	94.43
Reference [36] algorithm	94.39
Reference [37] algorithm	83.4
Reference [38] algorithm	86.0
Reference [39] algorithm	91.4
YOLOv8 [40]	95.7
The improved YOLOv5	96.6

Table 2 shows that YOLOv4, which uses lightweight YOLOv4, has a large disadvantage in the mAP metric. Because lightweight YOLOv4 uses fewer network layers and parameters, it results in lower detection performance on small targets. Compared to algorithms that also use the YOLOv5 network and the latest YOLOv8, our improved YOLOv5 network still achieved the best mAP values compared to the other networks, which proves the effectiveness of the attention mechanism and the loss function improvement, and our improved network was more suitable for small target flame detection. The flame detection algorithm using convolutional neural networks achieved better results in terms of accuracy, second only to our improved YOLOv5 algorithm. However, since convolutional neural networks require high computational resources, they may result in limited real-time or non-deployment for resource-constrained devices such as UAVs. The improved YOLOv5 network ensures both recognition accuracy and effective network lightweight, as shown by the above experimental results.

3.3. Ablation Experiment

To further illustrate the effectiveness of the improvements and to verify the impact of each improvement module on the model performance, we designed ablation experiments using YOLOv5s as the baseline network, mAP as the main evaluation index, and FPS as the auxiliary evaluation index. The specific data of the ablation experiment are shown in Table 3.

Table 3. Results of ablation experiments.

Algorithm	mAP (%)	FPS
YOLOv5s	91.2	71
YOLOv5s + STDL	92.8	64
YOLOv5s + CA	93.9	73
YOLOv5s + α-IoU	93.7	70
YOLOv5s + STDL + CA	95.3	67
YOLOv5s + STDL + α-IoU	94.8	65
YOLOv5s + STDL + CA + α-IoU	96.6	68

The comparison of the ablation experiments shows that increasing the small target detection layer increases the computational complexity and leads to a decrease in the detection speed, but it can increase the detection accuracy. The introduction of the CA module and the α-IoU can effectively improve the mAP value. The results of experiments 5–7 show that the use of the fusion of the three modules reduces the FPS value of the model, but the improvement of the mAP value is very obvious, which proves the effectiveness of our improvement of the YOLOv5 model.

4. Discussion

The current paper fully researched and tested a real-time flame-identification approach. To effectively meet the needs of fire rescue and firefighting, the latest YOLOv5 model was selected for research. Aiming to overcome the problem of insufficient recognition of a small target flame with the YOLOv5 network, we added a small-target identification layer to the YOLOv5 network. Thus, the detection capability concerning small target flames was efficiently enhanced. Adding an attention mechanism to the network improved the extraction of useful information and suppressed useless information. In addition, we also used α-IoU as the loss function, showing that the convergence rate and the regression's stability were improved. The above experimental results indicate that improving the YOLOv5 network can efficiently enhance the identification precision for flame targets, along with the detection speed. Our research proves that the enhanced approach has strong superiority and applicability. The specific advantages are as follows:

- Detection accuracy: the dataset in this paper consists of various sources such as artificially captured images, online images, and public datasets. Therefore, the dataset can successfully simulate a complex fire scene. The data reflect that the enhanced YOLOv5 network can meet the requirements of accurately identifying small target flames against complex backgrounds, thereby alleviating the probability of false identification of flames.
- Detection speed: the enhanced YOLOv5 network meets the real-time flame recognition requirements. In the original network-selection process, a comprehensive comparison of the one- and two-step methods was conducted, and the most representative YOLOv5 network in the one-step method was selected. The improved YOLOv5 network increases the network complexity, and is somewhat slower than the initial YOLOv5 network regarding detection speed, but still surpasses other neural networks. It also fully meets the needs for real-time detection.
- Lightweight network: this research mainly applied to equipment such as firefighting drones, and networking was considered at the beginning of the design. The enhanced YOLOv5 network model is very small, at only 25.1 MB. Accordingly, this network has very broad prospects in hardware applications. The model size directly depends on the hardware cost. It plays a crucial role in determining whether the detection algorithm can be employed on a large scale. In the comparison of different algorithms in Section 3.2, the YOLOv4 network achieved good results regarding identification precision and speed. However, the size of the YOLOv4 model is as high as 244 MB, which greatly limits the YOLOv4 network's practical usage. This further increases the

recognition algorithm's deployment cost when being put into use in the embedded devices of firefighting drone vision systems.

The YOLOv5 network is designed for industrial scenarios, including four network frameworks: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The complexity of the network structure increases sequentially. Users can choose the appropriate network architecture according to their actual needs. In this study, the selection and design of the recognition algorithm mainly considered the application and deployment of the detection algorithm on a firefighting drone for real-time identification of flames. Therefore, the identification precision, identification speed, and model size became the main considerations. The same improvement approach was utilized to enhance the four kinds of networks, and the same experimental dataset was trained. Table 4 presents the results.

Table 4. Comparing the recognition results of various network frameworks.

Algorithm	mAP (%)	Rate (Frame s^{-1})	Model Size (MB)
YOLOv5s	96.6	68	14
YOLOv5m	96.7	62	25.1
YOLOv5l	96.9	59	89.2
YOLOv5x	97.3	57	167

The experimental results indicate that the YOLOv5s network can attain superior detection accuracy and an optimal detection speed when using the same improved strategy and dataset, and it also has the characteristic of small model size. In summary, the network model based on YOLOv5s will have strong deployment potential in the embedded devices of firefighting drone vision systems.

However, firefighting drones may work at night, and the dataset in this paper was mainly based on daytime scenes, with only a small amount of night-time flame data. The experiments showed that the improved YOLOv5 has a certain error of detection and false detection capacity at night, which is a limitation of the current detection algorithm.

5. Conclusions and Future Research

The current study applied DL technology to the task of flame detection. In this way, a real-time identification approach for firefighting drone flame targets was proposed based on an improved YOLOv5. The YOLOv5 was employed for flame recognition for the first time. The capability to extract small target flame features was effectively improved based on the improved YOLOv5 network and by adding a small target-identification layer to the backbone network. In addition, a CA unit was added to the improved YOLOv5 network to fully enhance the flame target recognition precision. In addition, the DIoU in the original model was changed to α-IoU to enhance the capability of the model prediction framework to precisely find flames. This effectively improved the network's convergence rate and effect. The above experiments showed that the enhanced network model can effectively detect flame targets (especially small target flames). The improved YOLOv5 prediction value, recall rate, and mAP value were 85.7%, 94.8%, and 96.6%, respectively. Using the same dataset, the enhanced YOLOv5 algorithm was compared with another six algorithms; the mAP values increased by 10.9%, 3.9%, 5.4%, 15%, 7.9%, and 6.3%, respectively. Furthermore, the average recognition speed of the improved model was 0.014 s per image, which can fully satisfy the real-time flame identification needs.

In future research, we will gradually describe the established network framework and explain the network's semantics. We will explain how the individual hidden modules of a deep CNN guide the network to solve the flame-identification task. In addition, we will gradually optimize the flame-detection network framework, collect night-time flame data, and gradually improve the dataset. Ultimately, the network's ability to recognize flames at night will be fully enhanced to obtain a better flame-detection performance.

Author Contributions: X.X. contributed to the development of the algorithm, obtained flame images, and completed programming and writing. L.C. also performed the experiments and analyzed the results. K.C., Y.G. and B.T. helped in obtaining the images of the flame. M.H. reviewed and edited the draft. M.H. provided significant contributions to its development as the lead. All authors have read and agreed to the published version of the manuscript.

Funding: This project is supported by the High-Quality Project of the Ministry of Industry and Information Technology (Approval number: ZTZB-22-009-001) and the Scientific Research Project of the Beijing Municipal Education Commission (Approval number: KM202211232021).

Institutional Review Board Statement: The study did not involve humans or animals.

Informed Consent Statement: The study did not involve humans or animals.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: We sincerely thank the anonymous reviewers for their critical comments and suggestions for improving the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Celik, T.; Demirel, H. Fire detection in video sequences using a generic color model. *Fire Saf. J.* **2009**, *44*, 147–158. [CrossRef]
2. Liu, Z.; Yang, Y.; Ji, X. Flame detection algorithm based on a saliency detection technique and the uniform local binary pattern in the YCbCr color space. *Signal Image Video Process.* **2016**, *10*, 277–284. [CrossRef]
3. Kong, S.G.; Jin, D.; Li, S.; Kim, H. Fast fire flame detection in surveillance video using logistic regression and temporal smoothing. *Fire Saf. J.* **2016**, *79*, 37–43. [CrossRef]
4. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 84–90. [CrossRef]
5. Jiao, L.C.; Zhang, F.; Liu, F.; Yang, S.Y.; Li, L.L.; Feng, Z.X.; Qu, R. A Survey of Deep Learning-Based Object Detection. *IEEE Access* **2019**, *7*, 128837–128868. [CrossRef]
6. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. In Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
7. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [CrossRef]
8. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
9. Zhong, Z.; Wang, M.; Shi, Y.; Gao, W. A convolutional neural network-based flame detection method in video sequence. *Signal Image Video Process.* **2018**, *12*, 1619–1627. [CrossRef]
10. Zhang, L.; Wang, M.; Ding, Y.; Bu, X. MS-FRCNN: A Multi-Scale Faster RCNN Model for Small Target Forest Fire Detection. *Forests* **2023**, *14*, 616. [CrossRef]
11. Yu, L.; Liu, J. Flame image recognition algorithm based on improved Mask R-CNN. *Comput. Eng. Appl.* **2020**, *56*, 194–198.
12. Terven, J.; Cordova-Esparza, D. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *arXiv* **2023**, arXiv:2304.00501.
13. Zhai, S.; Shang, D.; Wang, S.; Dong, S. DF-SSD: An improved SSD object detection algorithm based on DenseNet and feature fusion. *IEEE Access* **2020**, *8*, 24344–24357. [CrossRef]
14. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
15. Abdusalomov, A.; Baratov, N.; Kutlimuratov, A.; Whangbo, T.K. An Improvement of the Fire Detection and Classification Method Using YOLOv3 for Surveillance Systems. *Sensors* **2021**, *21*, 6519. [CrossRef]
16. 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]
17. Bochkovskiy, A.; Wang, C.; Liao, H.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
18. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
19. Ruan, J. *Design and Implementation of Target Detection Algorithm Based on YOLO*; Beijing University of Posts and Telecommunications: Beijing, China, 2019.
20. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.

21. Wang, C.Y.; Liao HY, M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
22. Luvizon, D.C.; Tabia, H.; Picard, D. SSP-Net: Scalable sequential pyramid networks for real-Time 3D human pose regression. *Pattern Recogn.* **2023**, *142*, 109714. [CrossRef]
23. Wang, K.; Liew, J.H.; Zou, Y.; Zhou, D.; Feng, J. PANet: Few-Shot Image Semantic Segmentation with Prototype Alignment. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9197–9206.
24. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
25. Park, J.; Woo, S.; Lee, J.; Kweon, I.S. Bam: Bottleneck attention module. *arXiv* **2018**, arXiv:1807.06514.
26. Woo, S.; Park, J.; Lee, J.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
27. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
28. Jiang, B.; Luo, R.; Mao, J.; Xiao, T.; Jiang, Y. Acquisition of Localization Confidence for Accurate Object Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 784–799.
29. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over union: A metric and a Loss for Bounding Box Regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666. [CrossRef]
30. 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 AAAI Conference on Artificial Intelligence, New York, NY, USA, 3 April 2020; pp. 12993–13000.
31. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans. Cybern.* **2021**, *52*, 8574–8586. [CrossRef]
32. He, J.; Erfani, S.; Ma, X.; Bailey, J.; Chi, Y.; Hua, X.S. Alpha-IoU: A family of power intersection over union losses for bounding box regression. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 20230–20242.
33. Wang, S.; Chen, T.; Lv, X.; Zhao, J.; Zou, X.; Zhao, X.; Xiao, M.; Wei, H. Forest Fire Detection Based on Lightweight Yolo. In Proceedings of the 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 22–24 May 2021; pp. 1560–1565.
34. Muhammad, K.; Ahmad, J.; Lv, Z.; Bellavista, P.; Yang, P.; Baik, S.W. Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *49*, 1419–1434. [CrossRef]
35. Muhammad, K.; Ahmad, J.; Mehmood, I.; Rho, S.; Baik, S.W. Convolutional Neural Networks Based Fire Detection in Surveillance Videos. *IEEE Access* **2018**, *6*, 18174–18183. [CrossRef]
36. Muhammad, K.; Ahmad, J.; Baik, S.W. Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing* **2018**, *288*, 30–42. [CrossRef]
37. Li, G.; Chen, P.; Xu, C.; Sun, C.; Ma, Y. Anchor-Free Smoke and Flame Recognition Algorithm with Multi-Loss. *Fire* **2023**, *6*, 225. [CrossRef]
38. Yar, H.; Khan, Z.A.; Ullah, F.U.M.; Ullah, W.; Baik, S.W. A modified YOLOv5 architecture for efficient fire detection in smart cities. *Expert Syst. Appl.* **2023**, *231*, 120465. [CrossRef]
39. Liu, H.; Hu, H.; Zhou, F.; Yuan, H. Forest Flame Detection in Unmanned Aerial Vehicle Imagery Based on YOLOv5. *Fire* **2023**, *6*, 279. [CrossRef]
40. Talaat, F.M.; ZainEldin, H. An improved fire detection approach based on YOLO-v8 for smart cities. *Neural Comput. Appl.* **2023**, 1–16. [CrossRef]