

Article

A Thermal Imaging Flame-Detection Model for Firefighting Robot Based on YOLOv4-F Model

Sen Li ¹, Yeheng Wang ¹, Chunyong Feng ², Dan Zhang ¹, Huaizhou Li ¹ , Wei Huang ³  and Long Shi ^{4,*}¹ The School of Building Environment Engineering, Zhengzhou University of Light Industry, Zhengzhou 450001, China² The School of Mechanical and Electrical Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China³ The School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450001, China⁴ The State Key Laboratory of Fire Science, University of Science and Technology of China, Hefei 230026, China

* Correspondence: shilong@ustc.edu.cn

Abstract: Fire robots are an effective way to save lives from fire, but their limited detection accuracy has greatly hampered their practical applications in complicated fire conditions. This study therefore proposes an advanced thermal imaging flame detection model of YOLOv4-F based on YOLOv4-tiny. We replaced the Leaky ReLU activation function with the Mish activation function in the YOLOv4-tiny feature extraction network. A Spatial Pyramid Pooling (SPP) was also added to increase the receiving range of the feature extraction network. To improve the feature fusion efficiency between multi-scale feature layers, a Path Aggregation Network (PANet) was adopted to replace the YOLOv4-tiny Feature Pyramid Network (FPN) with full use of feature information; a high-quality dataset containing 14,757 thermal imaging flame images was built according to the PASCAL VOC 2007 dataset standard. The results show that, when compared to the YOLOv4-tiny, YOLOv5-s, and YOLOv7-tiny models, the average detection accuracy of the proposed YOLOv4-F model is 5.75% higher, the average mAP of the five IOU cases rises by 7.02%, and the average detection confidence of three scaled flames shows a 18.09% gain. The proposed YOLOv4-F meets the requirements of fire robots on real-time responses and accurate flame detection, offering an important tool to improve the performance of the current fire robots.

Keywords: flame detection; yolov4-tiny; deep learning; thermal imaging; firefighting robot



Citation: Li, S.; Wang, Y.; Feng, C.; Zhang, D.; Li, H.; Huang, W.; Shi, L. A Thermal Imaging Flame-Detection Model for Firefighting Robot Based on YOLOv4-F Model. *Fire* **2022**, *5*, 172. <https://doi.org/10.3390/fire5050172>

Academic Editor: James A. Lutz

Received: 18 September 2022

Accepted: 20 October 2022

Published: 21 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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/>).

1. Introduction

Firefighting robots are an effective tool to improve the efficiency of firefighting and rescue processes and therefore avoid casualties among firefighters. Under dangerous environments such as high temperature, oxygen lean combustion, and dense smoke, firefighting robots can replace those firefighters and enter the fire scene without putting them in a considerable danger. A firefighting robot is shown in Figure 1. Due to the complicated fire environment, flame detection offers more stable and reliable information for the autonomous firefighting and intelligent fire-field reconnaissance compared to the other detection approaches. Therefore, improving the flame detection performance for firefighting robots is of great significance to enhance their effectiveness and intelligence.

Many flame detection methods have been proposed previously for firefighting robots [1]. Traditional firefighting robots usually use flame sensors for flame detection [2–4]. However, their detection ranges are relatively small, which limits the practical implementation, especially under those complicated fire scenes. To overcome this limitation, the image-based detection method has been proposed based on vision sensors [5–10]. The image-based flame detection method can detect a wide range of flames, showing high reliability and accuracy [11].



Figure 1. Firefighting robots approach the fire source and extinguish the fire instead of firefighters (Photo by reporter Zhou Weihai from Science and Technology Daily).

Visual sensors can be divided into visible light and thermal imaging cameras. The visible light camera is greatly affected by smoke, making it difficult to effectively detect flames in the fire environment [12]. A thermal imaging camera performs imaging processes passively after receiving the thermal radiation of the objects. Those thermal infrared images mainly reflect the infrared radiation intensity of the objects, disturbed by the surrounding smoke. A comparison of the outputs obtained by the thermal imaging and visible light cameras from a typical fire environment is shown in Figure 2. Visible wavelengths are 0.4–0.75 μm , near-infrared wavelengths are 0.75–1.4 μm , far-infrared wavelengths are 50–1000 μm , and thermal infrared wavelength regions are 8–15 μm . The infrared camera in Figure 2 has a wavelength of 7–14 μm , which belongs to the region with thermal infrared wavelengths. It was observed that the infrared imaging method shows much obvious contrast and clear output in the fire environment. Considering its advantages, this study then focused on a flame-target-detection algorithm based on the thermal infrared image method.

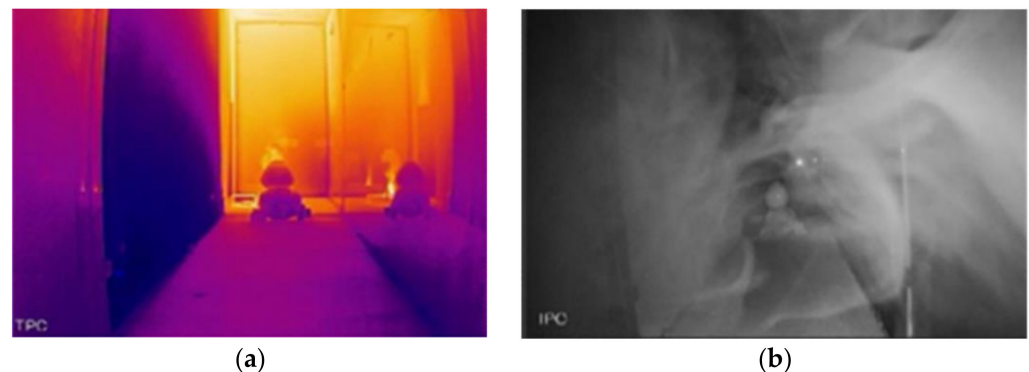


Figure 2. Different camera imaging methods for fire environment: (a) Infrared camera imaging; (b) Visible light camera.

2. Related Work

Currently, the primary methods for infrared image flame detection are divided into two categories: traditional approaches and machine learning [13]. The flame characteristics produced by the burning material serve as the foundation for traditional methods of fire detection. Different decision bases are developed for flame recognition based on different wavelengths [14,15]. Qin et al. [16] binarized the infrared images by spectral analysis to detect a flame using the flame's profile between different frames. However, the algorithm has difficulty locating accurate contour information in the face of multi-scale flames. Gong et al. [17] combined the dynamic detection of flames with a color approach to construct a flame center-of-mass stabilization algorithm using the similarity between video frames. Automatic flame detection is achieved by calculating the center-of-

mass of the flame region in each image frame and training with support vector machines. However, the fire environment is complex, and the calculation of a flame's center of mass is too complicated to meet real-time requirements. Jeong et al. [18] used infrared image features to set thresholds and introduce a preliminary decision mechanism. This mechanism determines the condition in which the texture correlation of the flame candidate area is consistent with the intensity test. Applying this algorithm to a fire environment is challenging as it is limited to simulation. Therefore, traditional methods are difficult to detect infrared flame images accurately and efficiently.

Machine learning is increasingly being applied to flame detection with the development of machine learning [19]. Wang et al. [20] formed a two-dimensional temperature field by hue saturation and color transformation of infrared images and targeted high-temperature regions with the help of image threshold segmentation. Combined with the time series prediction model, flame prediction can be achieved. Wang et al. [21] proposed a fire detection algorithm combining a convolutional neural network (CNN) and a support vector machine (SVM). It uses convolution for feature extraction and trains a linear support vector machine to achieve flame detection. Agrawal et al. [22] used histograms to preprocess the input image to obtain the color of the flame and used a support vector machine (SVM) to obtain the temperature information. The acquired color and temperature information detects the flame. The above algorithms do not simulate the real fire environment but only simulate the detection by the characteristics of the flame, which makes it challenging to realize fast and accurate flame detection in a real fire.

Due to its success in image classification tasks, many studies have used deep learning for target detection [23–25]. Deep learning models use relatively large datasets with better feature representations with powerful learning capabilities of convolutional neural networks [26–28]. However, the learning ability of those traditional manual features is limited, and more training data still cannot improve the learning effect. The target detection algorithm based on deep learning is proven to be one of the most powerful target-detection algorithms [29].

The target detector based on deep learning can be divided into a single- and a two-stage target detector [30]. Representative two-stage algorithms are R-CNN (convolutional neural network) series [31], fast R-CNN [32], faster R-CNN [33,34], R-FCN [35], Mask R-CNN [36], Libra R-CNN [37], etc. Two-stage series algorithms require two steps for target recognition. Although they show high accuracy, they limit the processing speed and cannot achieve real-time detection [38]. Guo et al. [39] proposed a firefighting robot flame detection algorithm based on faster R-CNN. The algorithm shows a high detection accuracy, but the real-time performance still needs improvement, with a limited detection speed of 1.4 FPS.

One-stage algorithms mainly include YOLO [40–42], SSD (Single Shot MultiBox Detector) [43], RetinaNet [44], etc. Although the detection accuracy of the one-stage series of algorithms is slightly lower, it shows a better real-time performance [45]. The flame detection model trained by Pu Li et al. [46] based on YOLOv3 showed an average detection accuracy of 83.7%, which is higher than those of the faster R-CNN, R-FCN, and SSD algorithms. Its detection speed was 27 FPS, which meets the real-time requirements. Therefore, the YOLO algorithm was selected in this study for further analysis.

At present, the most representative YOLOv4 shows good detection accuracy and processing speed [47]. Wang et al. [48] proposed a smoke and flame detection algorithm based on the YOLOv4 model, which balances detection accuracy and speed. Still, the study is limited to visible images and has many parameters. The complex network structure of the YOLOv4 has very high requirements in terms of the computers' performance, which limits its applications. To overcome these limitations, YOLOv4-tiny has been popularly adopted for target detection in standard computers. YOLOv4-tiny is a simplified version of YOLOv4, which significantly improves the speed and meets real-time requirements, but the model detection accuracy is much lower than YOLOv4's. This study investigated whether the detection accuracy can be improved based on the YOLOv4-tiny algorithm. The YOLOv4-F

target detection model studied in this paper improves the accuracy of flame detection for different scales based on the speed of the YOLOv4-tiny detection model. Algorithm experiments were conducted comparing lightweight detection algorithms YOLOv4-tiny, YOLOv5-s, and YOLOv7-tiny.

3. Methodology

3.1. YOLOv4-Tiny Flame Detection Model

The target detection network model based on deep learning usually consists of three parts, namely Backbone, Neck, and Head, which are adopted to extract target features in images, collect features in images and enable target prediction network, respectively.

The Backbone part consists of 2 CBL modules, 3 CSP (Cross-stage partial connections) modules, and 3 Maxpool modules, which perform 5 downsampling operations in total. The CBL module comprises Conv (convolutional layer), BN (batch normalization), and Leaky ReLU activation functions. The CBL module has the function of downsampling, its convolution kernel size is 3×3 , and its sliding step size is 2. The CSP module refers to the CSPNet network structure and consists of four CBL modules of Concat, where Concat represents tensor splicing. The size of the Maxpool module is 2×2 , and the sliding step length is 2, which has the effect of down-sampling.

YOLOv4-tiny adopts the feature pyramid network FPN module to perform feature fusion, which can significantly improve the running speed of the model. At the same time, it also shows insufficient feature fusion and low utilization of extracted features, which can easily lead to missed detection or detection errors in the model. FPN is a predictive fusion network that transmits strong semantic features from top to bottom. After FPN feature fusion, it outputs feature maps with two scales of 38×38 and 19×19 , as shown in Figure 3.

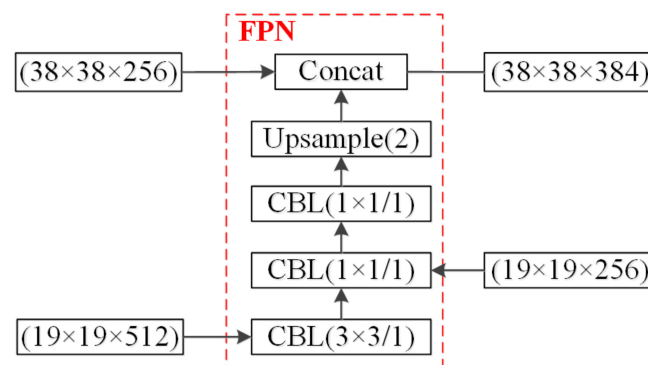


Figure 3. FPN feature fusion of YOLOv4-tiny.

The feature map $19 \times 19 \times 512$ output by the Backbone network is processed by two CBL modules, where one has a convolution kernel size of 3×3 and a step size of 1 and the other has a convolution kernel size of 1×1 and a step size of 1. Feature maps of $19 \times 19 \times 256$ are processed by the CBL module with a convolution kernel size of 1×1 and a step size of 1. Additionally, the feature maps after upsampling are $38 \times 38 \times 128$. The feature map is a Concat tensor splicing with the feature map $38 \times 38 \times 256$ output by the third CSP module in the Backbone network to output the feature map $38 \times 38 \times 384$.

The head target detection part of the YOLOv4-tiny model performs target detection through the feature maps of two scales of 19×19 and 38×38 . It shows a high recognition accuracy for both small- and medium-sized targets, but it is not appropriate for such large-scale target detection due to poor performance. The equation to calculate the depth of the feature map is $3 \times (5 + C)$, where 3 is the number of anchor boxes (anchor boxes) responsible for the prediction in each scale grid; 5 represents the width, height, center point abscissa, ordinate, and confidence of the prediction frame; and C represents the type of target detection. In this paper, the detection target only includes flame, so C is equal to 1, and the output depth of the feature map is 18.

3.2. YOLOv4-F Flame Detection Model

The choice of the activation function is very important for the deep learning model. An excellent activation function could improve the effectiveness of the gradient propagation without increasing the amount of extra calculation. In many deep learning models, the performance of the Mish activation function is found to be better than the Leaky ReLU activation function and the other standard activation functions [49,50].

The activation functions of Mish and Leaky ReLU are shown in Equations (1) and (2). The value of a in the Leaky ReLU function is 0.2. The function curve is shown in Figure 4. The Mish function curve is smoother than that of the Leaky ReLU function curve. However, the Mish function contains an exponential function, which requires a longer calculation time. The Mish function curve is smoother, and it shows a relatively stronger feature extraction capability with richer feature information. So, the target detection model is easier to optimize with an improved generalization ability. Therefore, this study used the Mish activation function in the Backbone of the YOLOv4-F model to achieve a better feature extraction effect. The CBM module in the backbone network is composed of a convolutional layer, a normalization layer, and a Mish activation function. The rest of the model still uses the Leaky ReLU activation function.

$$\text{Mish} = x \times \tanh(\ln(1 + e^x)), \quad (1)$$

$$\text{Leaky_relu} = \begin{cases} x, & x \geq 0 \\ ax, & x < 0 \end{cases} \quad (2)$$

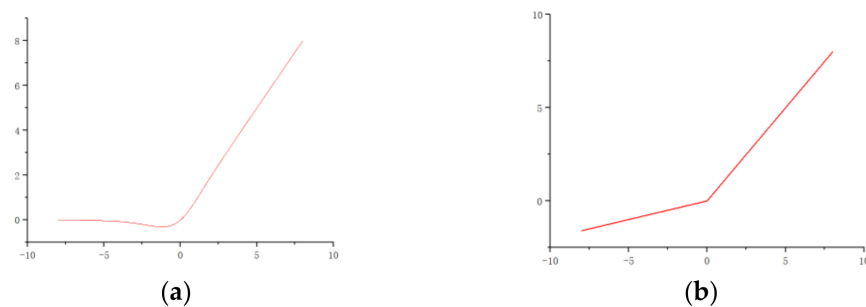


Figure 4. Activation function: (a) Mish activation function; (b) Leaky_relu activation function.

The geometric ratio of the images through the firefighting robot usually changes during the movement, and the flame is also dynamically changing, which makes the detection more difficult. Therefore, besides Backbone, a feature fusion network was added to the SPP module to fuse local and global features in this study. This can improve the expression ability of network features, increase the Receptive Field of the network, and enhance the detection accuracy of flame. The SPP (Spatial Pyramid Pooling) module comprises four parallel branches, which are the largest pooling layer with cores of 5×5 , 9×9 , and 13×13 (all sliding steps are 1) and a jump connection layer. Its structure is shown in Figure 5. The SPP module can effectively increase the reception range of the feature extraction network and effectively isolate the most significant features in the data.

PANet (Path Aggregation Network) is since a top-down and bottom-up bidirectional converged network structure, as shown in Figure 6. This structure can accelerate the fusion of multi-scale feature information by shortening the information flow path between the low- and high-level feature layers, and then improve the multi-scale feature layer's ability to determine the position. As a feature enhancement network, the algorithm in this paper used the PANet structure instead of the FPN (Feature Pyramid Network) structure in YOLOv4-tiny, which is because it can improve the efficiency of feature fusion between different feature layers of the network and improve the disadvantages of the FPN structure such as too weak perceptual field, inadequate feature fusion, and too low feature information utilization.

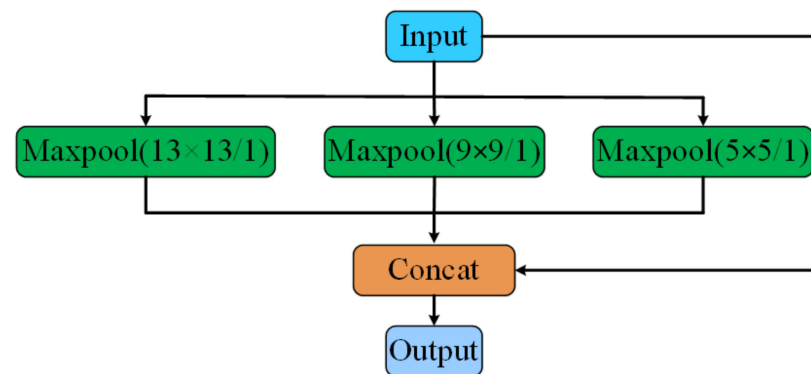


Figure 5. Schematic of the SPP module structure.

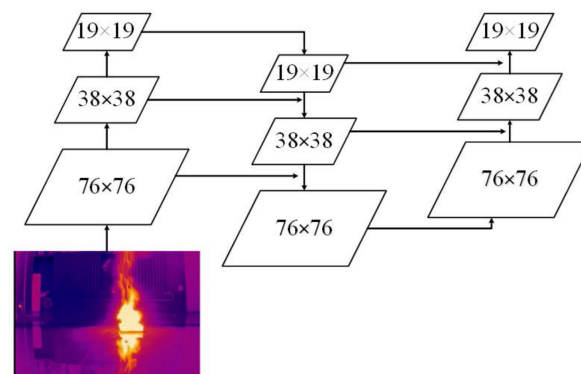


Figure 6. Network structure of PANet.

YOLOv4-tiny only performs the fusion of two feature scales, and the detection performance for large-scale targets is low. To improve the accuracy of the flame detection model, the feature fusion part adopted the feature map fusion strategy for the three scales, namely 76×76 , 38×38 , and 19×19 . The target detection part performs flame detection based on the feature maps of these three scales. The network structure of the YOLOv4-F flame detection model proposed in this study is shown in Figure 7. Its input image size of $608 \times 608 \times 3$ and output feature maps of $19 \times 19 \times 18$, $38 \times 38 \times 18$, and $76 \times 76 \times 18$.

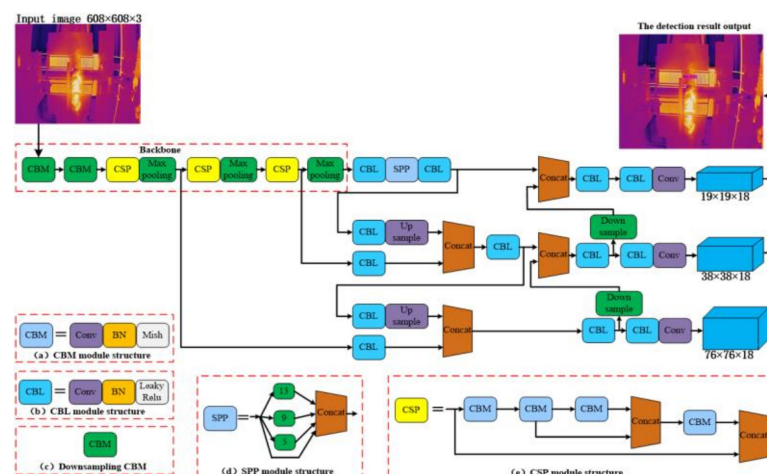


Figure 7. Network structure of the YOLOv4-F flame detection model.

4. Development of Experimental Platform

4.1. Detection Dataset Platform

Datasets of high-quality images are essential to the training of the deep learning models described above. However, the current open-source datasets mainly contain normal

images with visible light, such as ImageNet, Cityscapes, and Pascal VOC [51–53]. Thermal imaging datasets are limited, which is reflected by not only small numbers but also slow updates, such as FREE FLIR Thermal Dataset [54], OTCBVS [55], and SCUT FIR Pedestrian Dataset [56]. There is no standard dataset for fire scenes, and some organizations or individuals have open-sourced some visible light datasets for researchers to use, such as PNG still image set, MIVIA, and the FIRESENSE database [57–60].

Therefore, this study carried out a series of experimental tests to establish a dataset for thermal images under fire conditions. First, we selected the high-quality flame images obtained from fire tests. Secondly, we marked the flames inside those thermal images. Finally, we divided the training set and test set to build a standard dataset.

Different combustible materials under various conditions show different flame shapes along with the combustion processes and time. To enrich the flame types in the dataset and achieve better training, this study considered different flame types, sizes, and combustion scenes and other influencing factors during the collection of thermal images, as shown in Figure 8. The firefighting robot used in this study adopted a crawler-type chassis, which can rotate in situ. It is equipped with a DH-TPC-BF5400 thermal imaging gun-type camera supplied by Zhejiang Dahua Technology Co., Ltd. (Hangzhou, China), which has a temperature range of $-40\text{ }^{\circ}\text{C}$ to $550\text{ }^{\circ}\text{C}$, a resolution of 1280×1024 , a wavelength range of $7\sim 14\mu\text{m}$, and a thermal sensitivity of $<40\text{mK@f}/1.0$. The thermal imaging camera was installed on a two-degree-of-freedom pan tilt, which can rotate 360° horizontally and rotate 180° vertically.



Figure 8. The firefighting robot collects thermal imaging flame images.

Considering the different motion states of the robot when performing flame recognition, the robot is set in motion and stationary when capturing images, and rotates the camera by controlling the head to capture flame images from multiple angles with different backgrounds. Factors such as the size and resolution of the images affect the learning quality of the deep learning model. The higher the data quality is, the better the learning effect will be, and this applies to the features as well. Therefore, most of the selected images are high-definition images with 1280×1024 pixels. Additionally, the image acquisition from the videos was carried out in different conditions, such as indoor, outdoor, daytime, night, and light on and off. The main types of flames are firewood, straw, cardboard boxes, polystyrene foam, alcohol, n-heptane, and others.

The flame recognition model uses a picture dataset, so it is important to divide the collected video into pictures using a Python program. Since the flame is dynamically changing, the flame in each frame of the video is not the same. To ensure the diversity of the dataset, every two frames from the video were chosen. The segmented flame images were manually screened. Finally, a total of 14,757 thermal infrared flame images were selected for the dataset and some of which are shown in Figure 9.

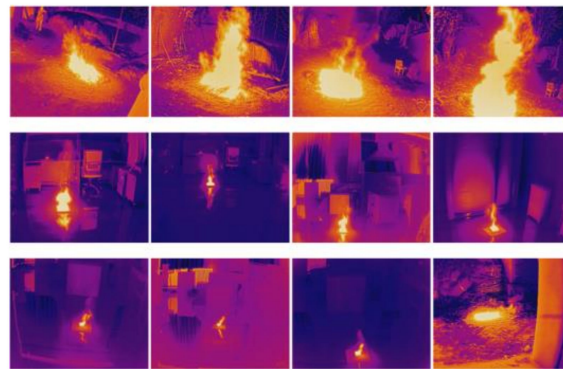


Figure 9. The firefighting robot collects thermal imaging flame images.

The experimental data used for the Faster R-CNN and YOLOv4 algorithms come from the public dataset PASCAL VOC 2007. This study also imitates the format of PASCAL VOC 2007 to construct the dataset. The dataset contains a total of four folders, namely Annotations, ImageSets, JPEGImages, and Labels. The structure of the flame dataset data storage is shown in Figure 10.

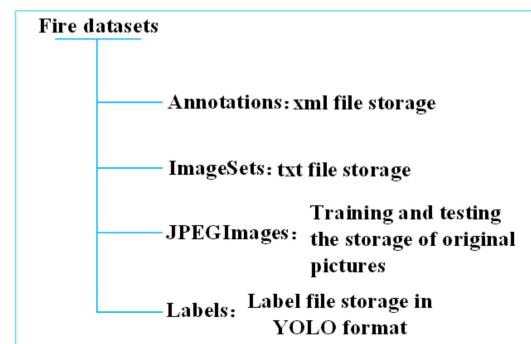


Figure 10. Data storage structure of flame dataset.

The thermal imaging flame dataset is processed in the PASCAL VOC format, and the label information is normalized. The normalization is achieved through Equation (3):

$$\begin{cases} x = \frac{x_{\max} + x_{\min}}{2width}, y = \frac{y_{\max} + y_{\min}}{2height} \\ w = \frac{x_{\max} - x_{\min}}{width}, h = \frac{y_{\max} - y_{\min}}{height} \end{cases} \quad (3)$$

where (*width* and *height*) are the width and height of the original image, respectively; (x_{\min} , y_{\min}) and (x_{\max} , y_{\max}) are the position of the upper left corner and the lower right corner of the original sample true bounding box, respectively; and (x , y) and (w , h) are the center point coordinates and width and height of the target after normalization, respectively.

After normalizing these images, bounding box information contains a total of five parameters (x , y , w , h) and the label number corresponding to the category.

The Labelling1.8.2 image annotation tool was used to select the flame in the image as the annotation to generate an xml file, which contains the image size, the type of the labeled target, and the coordinate information. Then, a python script was developed to parse the xml file and convert it into a txt file. The training set directly participates in all samples of the model training and fitting. The model determines the learning parameters, such as the weight and bias of the model, through the training set fitting. The test set does not participate in the selection process of learning parameters and tuning parameters, and its main function is to evaluate the performance of the training model.

The YOLOv4 network model is a target detection network based on Anchors boxes. The reasonable setting of the Anchors boxes shows a significant impact on the detection accuracy. YOLOv4 selects nine general Anchors boxes based on the sample size clustering

in the MS COCO dataset. The sizes of the Anchors boxes are (12, 16), (19, 36), (40, 28), (36, 75), (76, 55), (72, 146), (142, 110), (192, 243), and (459, 401). Since the data quality and target types of the dataset constructed in this paper and the MS COCO dataset are quite different, a large difference between the size of the Anchors and the size of the detection flame will lead to the problem of missing and false detection of the flame target. Therefore, this study used a K-means++ clustering algorithm to determine the appropriate size of the Anchors boxes.

This study used a K-means++ clustering algorithm and generated nine Anchors boxes with various sizes to train the flame detection model based on flame images with a resolution of 608×608 . The nine Anchor boxes are (43, 59), (49, 122), (67, 181), (94, 282), (225, 194), (176, 345), (297, 336), (214, 484), and (313, 536). Among them, (297, 336), (214, 484), and (313, 536) correspond to the feature map of the largest receptive field of 19×19 . Additionally, (94, 282), (225, 194), and (176, 345) correspond to the 38×38 feature map of the medium receptive field. Finally, (43, 59), (49, 122), and (67, 181) correspond to the feature map of 76×76 minimum receptive field. The cluster center distribution of those Anchors boxes is shown in Figure 11. The black dots in the figure are the flame labels in the dataset, and the five-pointed stars are the clustered Anchors boxes.

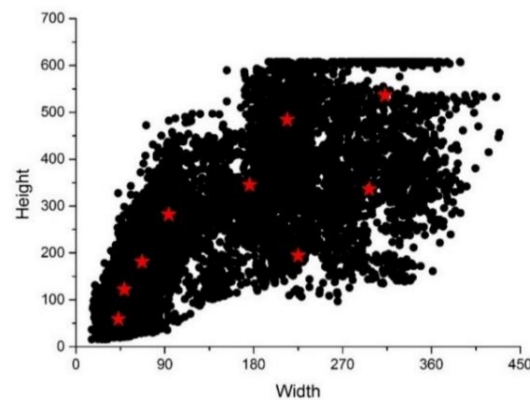


Figure 11. Cluster center distribution of Anchors boxes.

4.2. Experimental Test Platform

The configuration of the model training environment is shown in Table 1. During the training process, 70% of the images in the dataset were selected as the training set, while the remaining 30% were used as the test set.

Table 1. Experimental environment configuration.

Hardware Specifications	Configuration Information
Processor	Intel(R) Xeon(R) CPU E5-2630 v2 @ 2.60GHz
Graphics card	NVIDIA Quadro K2000 4G
RAM	48G
Operating system	Ubuntu 16.04

The target detection models involved in the experiments include YOLOv4-F, YOLOv4-tiny, YOLOv5-s, and YOLOv7-tiny. The configuration of the main training parameters of the model is shown in Table 2. To speed up convergence and improve learning speed, the training uses the pre-training weight files of the official YOLOv4 dataset for migration learning.

Table 2. Configuration of training parameters.

Width	Height	Channels	Momentum	Batch
608	608	3	0.9	64
subdivision	learning_rate	max_batches	steps	scales
32	0.001	4000	3200, 3600	0.1, 0.1

5. Results and Discussion

5.1. Model Training Result Analysis

The indicators of the model performance typically include Precision, Recall, F1-score, IOU (Intersect Over Union), AP (Average Precision), and mAP (mean Average Precision). The model training results are shown in Table 3. The table shows that the performance metrics of the YOLOv4-F flame detection model proposed in this paper are better than those of the lightweight models such as YOLOv4-tiny, YOLOv5-s and YOLOv7-tiny.

Table 3. Comparison of performance indicators of target detection models.

Detection Model	Precision	Recall	F1-Score	Average IOU
YOLOv4-tiny	0.82	0.75	0.79	58.43%
YOLOv5-s	0.85	0.79	0.82	60.54%
YOLOv7-tiny	0.89	0.85	0.85	64.64%
YOLOv4-F	0.95	0.96	0.96	72.63%

Both Precision and Recall determine AP. AP is calculated by drawing a Precision–Recall (PR) curve equal to the area under the curve. The PR curves of the YOLOv4-F, YOLOv4-tiny YOLOv5-s, and YOLOv7-tiny in the test set are shown in Figure 12. It can be seen that the AP of the YOLOv4-F, YOLOv4-tiny, YOLOv5-s, and YOLOv7-tiny are 97.46%, 91.35%, 91.49%, and 92.28%, respectively. Furthermore, YOLOv4-F is higher than the other models by 6.11%, 5.97%, and 5.18%, respectively.

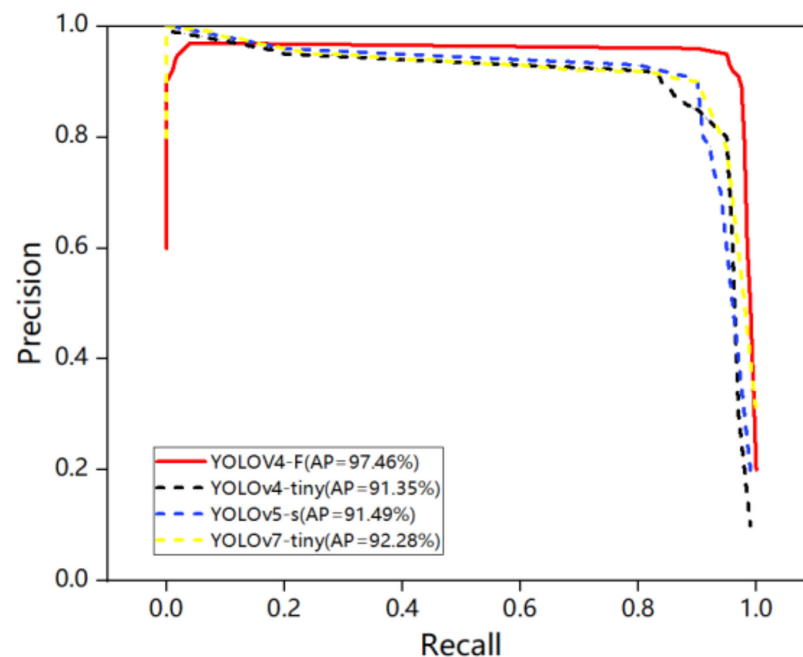
**Figure 12.** PR curve of the target detection model.

Table 4 shows the mAP values of the four models under different IOUs. The table shows that the proposed YOLOv4-F model has a higher mAP than the YOLOv4-tiny, YOLOv5-s and YOLOv7-tiny models when the IOU values are 0.5, 0.6, 0.7, 0.8, and 0.9. By setting five different values of IOUs, the detection accuracy (mAP) of the YOLOv4-F algorithm is compared with other algorithms in detecting the average of flames at different scales. Under the five IOUs, the average mAPs of YOLOv4-F, YOLOv4-tiny, YOLOv5-s, and YOLOv7-tiny models were found to be 51.95%, 43.14%, 44.47%, and 47.77%. It further shows that the YOLOv4-F model has a relatively reliable flame detection accuracy.

Table 4. mAP of the detection model under different IOUs.

Detection Model	mAP@0.5 (%)	mAP@0.6 (%)	mAP@0.7 (%)	mAP@0.8 (%)	mAP@0.9 (%)
YOLOv4-tiny	90.53	75.1	41.07	8.9	0.08
YOLOv5-s	91.23	74.64	47.89	8.5	0.07
YOLOv7-tiny	92.56	84.38	50.56	10.62	0.75
YOLOv4-F	96.54	85.83	56.75	19.81	0.82

5.2. Thermal Imaging Flame Image Detection

A total of 16 flame images with different scales, scenes, and combustion objects were selected from the dataset to test the performance of the flame detection model after the training process. The flame scale was determined according to the flame-mark frame area and the percentage of the flame in the entire image. For example, if the flame frame occupies less than 3% of the entire image, the flame was considered small-scale; flames with a 3–12% occupancy rate represent medium-scale; and flames with more than 12% were considered large-scale. The calculation diagram is shown in Figure 13, and the equation is shown in Equation (4), where $Fire_{scale}$ represents the flame scale. $Width$ and $Height$ represent the width and height of the image, respectively, and (x_{min}, y_{min}) and (x_{max}, y_{max}) represent the coordinates of the flame marking frame, respectively.

$$Fire_{scale} = \frac{(x_{max} - x_{min})(y_{max} - y_{min})}{Width \times Height} \times 100\%, \quad (4)$$

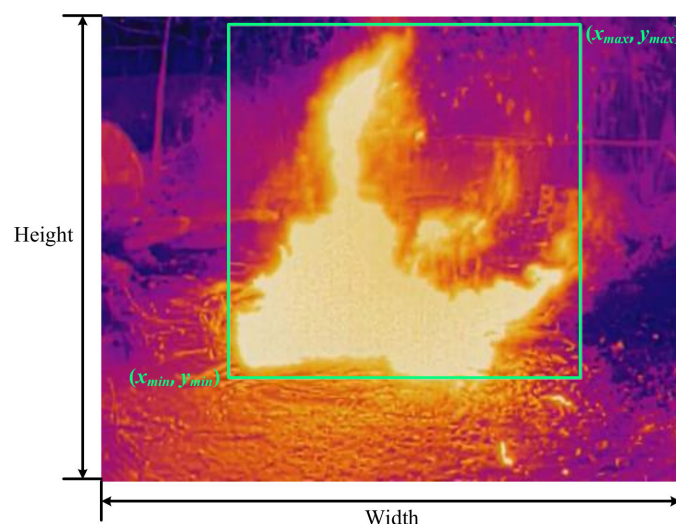


Figure 13. Schematic diagram of flame scale calculation.

When the resolution of the input image is 608×608 for the flame detection model, the confidence distribution of the flame detection for those 16 flame images is shown in Figure 14. Confidence refers to the prediction frame's accuracy in predicting an object's class and location. Figure 14a shows the confidence distribution of large-scale flame detection. It can be seen that the YOLOv4-tiny, YOLOv5-s, and YOLOv7-tiny models show low confidence in large-scale flame detection, and there exists missed detection as well, which

is represented by 0 confidence. In contrast, the proposed YOLOv4-F model can accurately identify the flame images missed by YOLOv4-tiny, YOLOv5-s, and YOLOv7-tiny.

The advantages of the proposed YOLOv4-F are also reflected in Figure 15. Compared to the YOLOv4-tiny, YOLOv5-s, and YOLOv7-tiny, the YOLOv4-F model can detect flames of three scales with much higher accuracy, and the confidence of flame detection is also higher. For all three scales, the detection accuracy of YOLOv4-F is, on average, 27.18%, 13.59%, and 13.49% higher than that of YOLOv4-tiny, YOLOv5-s, and YOLOv7-tiny.

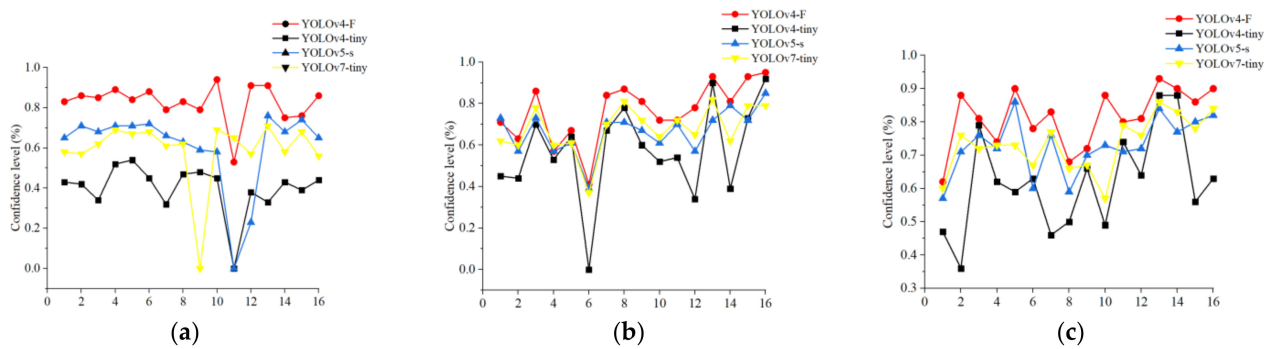


Figure 14. Three-scale flame detection confidence distribution of the target detection model: (a) large scale; (b) medium scale; (c) small scale.

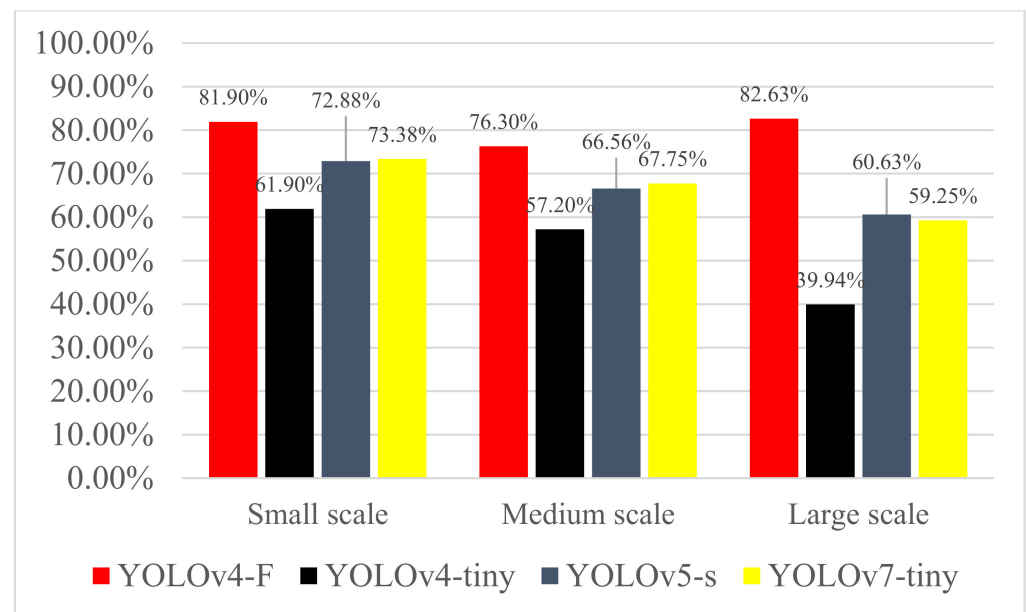


Figure 15. Three-scale flame detection confidence distribution of the target detection model.

Tables 5–7 show the flame images for the detection tests of the YOLOv4-F, YOLOv4-tiny, YOLOv5-s, and YOLOv7-tiny. The confidence level of the flame detection is added at the bottom of these images in order to see the detection results more clearly. It can be observed that the flame detection frame of YOLOv4-F is more advanced in detecting flame images under fire conditions.

Table 5. Images and results for large-scale flame detection.








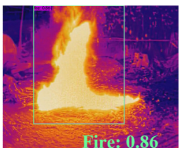













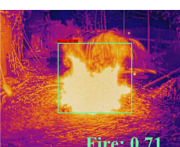


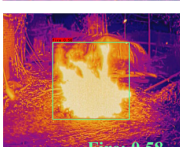
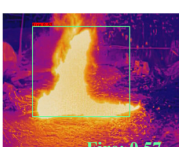




Large Scale Image	Large Scale Image 1	Large Scale Image 2	Large Scale Image 3	Large Scale Image 4	Large Scale Image 5	Large Scale Image 6
Original image of infrared flame						
YOLOv4-F flame detection	 Fire: 0.83	 Fire: 0.86	 Fire: 0.89	 Fire: 0.84	 Fire: 0.79	 Fire: 0.91
YOLOv4-tiny flame detection	 Fire: 0.43	 Fire: 0.42	 Fire: 0.52	 Fire: 0.54	 Fire: 0.32	 Fire: 0.38
YOLOv5-s flame detection	 Fire: 0.65	 Fire: 0.71	 Fire: 0.71	 Fire: 0.71	 Fire: 0.66	 Fire: 0.23
YOLOv7-tiny flame detection	 Fire: 0.58	 Fire: 0.57	 Fire: 0.69	 Fire: 0.67	 Fire: 0.61	 Fire: 0.57

Table 6. Images and results for medium-scale flame detection.








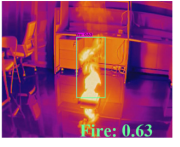

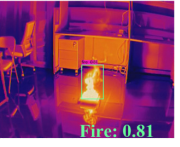


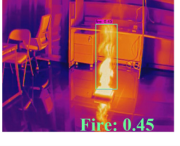
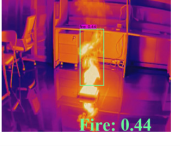

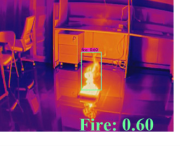


Medium Scale Image	Medium Scale Image 1	Medium Scale Image 2	Medium Scale Image 3	Medium Scale Image 4	Medium Scale Image 5	Medium Scale Image 6
Original image of infrared flame						
YOLOv4-F flame detection	 Fire: 0.71	 Fire: 0.63	 Fire: 0.84	 Fire: 0.81	 Fire: 0.78	 Fire: 0.93
YOLOv4-tiny flame detection	 Fire: 0.45	 Fire: 0.44	 Fire: 0.67	 Fire: 0.60	 Fire: 0.34	 Fire: 0.73

Table 6. Cont.

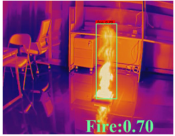
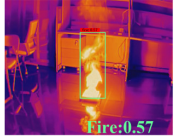

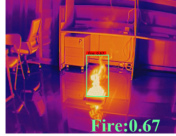



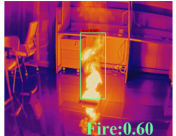

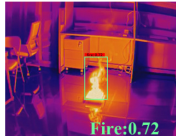
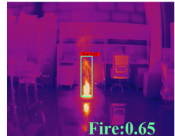




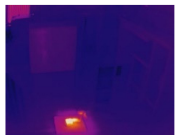

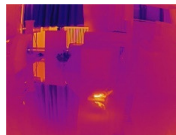

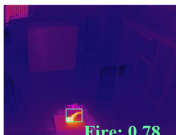



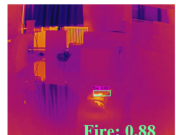







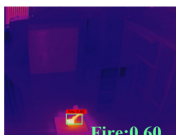
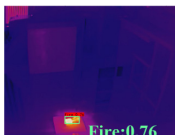





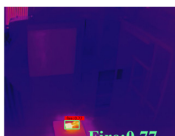
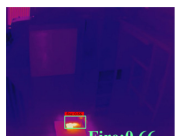


Medium Scale Image	Medium Scale Image 1	Medium Scale Image 2	Medium Scale Image 3	Medium Scale Image 4	Medium Scale Image 5	Medium Scale Image 6
YOLOv5-s flame detection	 Fire:0.70	 Fire:0.57	 Fire:0.71	 Fire:0.67	 Fire:0.57	 Fire:0.72
YOLOv7-tiny flame detection	 Fire:0.62	 Fire:0.60	 Fire:0.70	 Fire:0.72	 Fire:0.65	 Fire:0.78

Table 7. Images and results for small-scale flame detection.

Small Scale Image	Small Scale Image 1	Small Scale Image 2	Small Scale Image 3	Small Scale Image 4	Small Scale Image 5	Small Scale Image 6
Original image of infrared flame						
YOLOv4-F flame detection	 Fire: 0.62	 Fire: 0.78	 Fire: 0.83	 Fire: 0.68	 Fire: 0.72	 Fire: 0.88
YOLOv4-tiny flame detection	 Fire: 0.47	 Fire: 0.63	 Fire: 0.46	 Fire: 0.50	 Fire: 0.66	 Fire: 0.49
YOLOv5-s flame detection	 Fire:0.57	 Fire:0.60	 Fire:0.76	 Fire:0.59	 Fire:0.70	 Fire:0.73
YOLOv7-tiny flame detection	 Fire:0.60	 Fire:0.67	 Fire:0.77	 Fire:0.66	 Fire:0.67	 Fire:0.57

Although the detection time for the YOLOv4-F is longer than those of the YOLOv4-tiny, YOLOv5-s, and YOLOv7-tiny, this will not hamper the related applications as the related detection time can fulfill the requirements. The time consumed by YOLOv4-F, YOLOv4-tiny, YOLOv5-s, and YOLOv7-tiny to detect a flame image is 131.65 ms, 84.42 ms, 85.63 ms, and 96.42 ms, respectively. Reducing the resolution of the input image can also reduce the calculation time and increase the detection speed, but it may reduce the detection accuracy. In this study, therefore, we calculated the time and detection accuracy of the model when the input image resolutions were 608×608 , 512×512 , 416×416 , and 320×320 . The relationship between the target detection time and accuracy of the two models is shown in Figure 16. It can be seen that processing an image based on the YOLOv4-F mode is relatively longer when the input image resolution is the same. However, the YOLOv4-F model obtains a higher detection accuracy than those based on the YOLOv4-tiny, YOLOv5-s, and YOLOv7-tiny models under the same time frame.

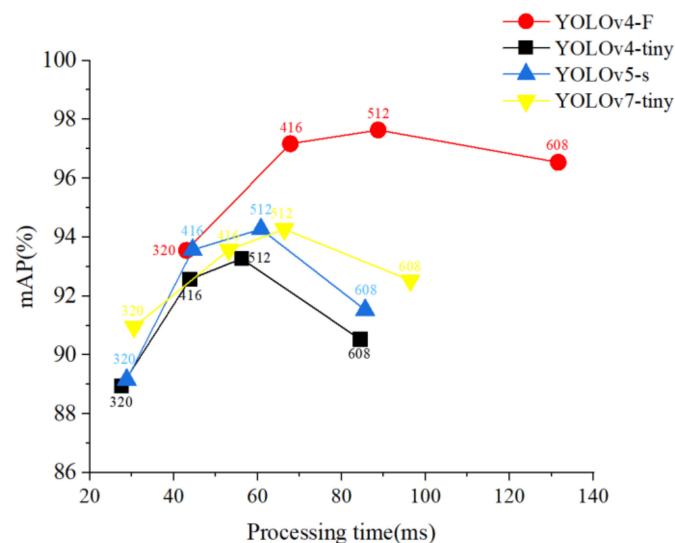


Figure 16. The relationship between target detection time consumption and accuracy.

6. Conclusions

An advanced thermal infrared flame detection model YOLOv4-F was proposed through this study to be applied to firefighting robots based on YOLOv4-tiny. To achieve more efficient feature fusion and better detection performance under complicated fire conditions, we replaced the Leaky ReLU activation function in the Backbone network of the YOLOv4-tiny model with the Mish activation function, added the SPP module after the Backbone network, and used PANet to replace the FPN in the YOLOv4-tiny model. It was known that the proposed model could significantly improve the detection performance of different scaled flame images. The training results showed that the average detection accuracy of the proposed YOLOv4-F model was 6.11%, 5.97%, and 5.18% higher than those of the YOLOv4-tiny, YOLOv5-s, and YOLOv7-tiny models, respectively; the average mAP of the five IOU cases increased by 8.81%, 7.48%, and 4.78%, respectively; and the average detection confidence of three-scaled flames rose by 27.18%, 13.59%, and 13.49%, respectively. It was also shown that the YOLOv4-F model obtains a higher detection accuracy under the same time frame. The proposed YOLOv4-F flame detection model provides the firefighting robot with an important tool for more accurate flame detection.

Author Contributions: Conceptualization, S.L., Y.W. and C.F.; methodology, S.L. and Y.W.; software, S.L., Y.W. and C.F.; validation, S.L., W.H. and L.S.; formal analysis, D.Z. and H.L.; investigation, Y.W. and C.F.; resources, S.L., D.Z. and W.H.; data curation, Y.W., C.F. and H.L.; writing—original draft preparation, S.L. and Y.W.; writing—review and editing, All Authors; visualization, S.L. and Y.W.; supervision, W.H. and L.S.; project administration, S.L. and L.S.; funding acquisition, S.L. and D.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation for Young Scientists of China, grant number No. 51804278. Training plan for young backbone teachers in Colleges and universities in Henan Province, grant number No. 2021GGJS094. Henan science and technology development plan, grant number No. 212102210535. The Key Scientific Research Projects of Universities of Henan, grant number Grant No. 20A620005.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Yu, N.; Chen, Y. Video flame detection method based on TwoStream convolutional neural network. In Proceedings of the 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 24–26 May 2019; pp. 482–486.
2. Gao, S.; Zhang, Z.; Zhao, Z.; Jamali, M.M. Vision and infra-red sensor based fire fighting robot. In Proceedings of the 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS), Windsor, ON, Canada, 5–8 August 2018; pp. 873–876.
3. Ryu, J.; Kwak, D. A Study on a Complex Flame and Smoke Detection Method Using Computer Vision Detection and Convolutional Neural Network. *Fire* **2022**, *5*, 108. [\[CrossRef\]](#)
4. Guede-Fernández, F.; Martins, L.; de Almeida, R.V.; Gamboa, H.; Vieira, P. A Deep Learning Based Object Identification System for Forest Fire Detection. *Fire* **2021**, *4*, 75. [\[CrossRef\]](#)
5. Barmpoutis, P.; Dimitropoulos, K.; Kaza, K.; Grammalidis, N. Fire detection from images using faster R-CNN and multidimensional texture analysis. In Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 8301–8305.
6. Sharma, J.; Granmo, O.-C.; Goodwin, M.; Fidje, J.T. Deep convolutional neural networks for fire detection in images. In *International Conference on Engineering Applications of Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 183–193. [\[CrossRef\]](#)
7. Shamsoshoara, A.; Afghah, F.; Razi, A.; Zheng, L.; Fulé, P.Z.; Blasch, E. Aerial imagery pile burn detection using deep learning: The FLAME dataset. *Comput. Netw.* **2021**, *193*, 108001. [\[CrossRef\]](#)
8. Umar, M.M.; Silva, L.C.D.; Bakar, M.S.A.; Petra, M.I. State of the art of smoke and fire detection using image processing. *Int. J. Signal Imaging Syst. Eng.* **2017**, *10*, 22–30. [\[CrossRef\]](#)
9. Dunning, A.J.; Breckon, T.P. Experimentally defined convolutional neural network architecture variants for non-temporal real-time fire detection. In Proceedings of the 2018 25th IEEE international conference on image processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 1558–1562.
10. Suresh, J. Fire-fighting robot. In Proceedings of the 2017 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India, 2–3 June 2017; pp. 1–4.
11. Ramasubramanian, S.; Muthukumaraswamy, S.A.; Sasikala, A. Fire detection using artificial intelligence for fire-fighting robots. In Proceedings of the 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 13–15 May 2020; pp. 180–185.
12. Li, S.; Feng, C.; Niu, Y.; Shi, L.; Wu, Z.; Song, H. A fire reconnaissance robot based on SLAM position, thermal imaging technologies, and AR display. *Sensors* **2019**, *19*, 5036. [\[CrossRef\]](#)
13. Jiang, Y.; Pang, D.; Li, C. A deep learning approach for fast detection and classification of concrete damage. *Autom. Constr.* **2021**, *128*, 103785. [\[CrossRef\]](#)
14. Matthes, J.; Waibel, P.; Vogelbacher, M.; Gehrmann, H.J.; Keller, H.B. A new camera-based method for measuring the flame stability of non-oscillating and oscillating combustions. *Exp. Therm. Fluid Sci.* **2019**, *105*, 27–34. [\[CrossRef\]](#)
15. Nadarajan Assari Syamala, S.R.; Santhosh, A.; Ramesh, R. Analysis of Forest Fire in Australia using Visible Infrared Imaging Radiometer Suite. In Proceedings of the 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Kunming, China, 12 March 2021; pp. 482–486.
16. Qin, C.; Zhang, M.; He, W.; Guan, C.; Sun, W.; Zhou, H. A New Real-Time Fire Detection Method Based On Infrared Image. In Proceedings of the 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)2019, Dalian, China, 19–20 October 2019; pp. 476–479. [\[CrossRef\]](#)
17. Gong, F.; Li, C.; Gong, W.; Li, X.; Yuan, X.; Ma, Y.; Song, T. A Real-Time Fire Detection Method from Video with Multifeature Fusion. *Comput. Intell. Neurosci.* **2019**, *2019*, 1939171. [\[CrossRef\]](#)
18. Jeong, S.Y.; Kim, W.H. Thermal Imaging Fire Detection Algorithm with Minimal False Detection. *KSII Trans. Internet Inf. Syst.* **2020**. [\[CrossRef\]](#)
19. Umoh, U.; Udo, E.; Emmanuel, N. Support Vector Machine-Based Fire Outbreak Detection System. *Int. J. Soft Comput. Artif. Intell. Appl.* **2019**, *8*, 1–18. [\[CrossRef\]](#)
20. Wang, Q.; Yang, C.; Duan, S.; Wei, S. Research on Fire Prediction Algorithm Based on Thermal Infrared Image. In Proceedings of the 2021 7th International Conference on Computing and Artificial Intelligence, Tianjin, China, 23–26 April 2021; Association for Computing Machinery: Tianjin, China, 2021; pp. 216–221.
21. Wang, K.; Zhang, Y.; Wang, J.; Zhang, Q.; Chen, B.; Liu, D. Fire detection in infrared video surveillance based on convolutional neural network and SVM. In Proceedings of the 2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP), Shenzhen, China, 13–15 July 2018; pp. 162–167.
22. Agrawal, G.; Mishra, R.; Ransingh, A.; Chakravarty, S. Flame Temperature Prediction Using Machine Learning Model. In Proceedings of the 2020 IEEE India Council International Subsections Conference (INDISCON), Visakhapatnam, India, 3–4 October 2020; pp. 157–162.
23. Xiao, Y.; Tian, Z.; Yu, J.; Zhang, Y.; Liu, S.; Du, S.; Lan, X. A review of object detection based on deep learning. *Multimed. Tools Appl.* **2020**, *79*, 23729–23791. [\[CrossRef\]](#)

24. Gong, M.; Wang, D.; Zhao, X.; Guo, H.; Luo, D.; Song, M. A review of non-maximum suppression algorithms for deep learning target detection. In Proceedings of the Seventh Symposium on Novel Photoelectronic Detection Technology and Applications, Kunming, China, 12 March 2021; pp. 821–828.
25. Jiang, X.; Wang, C.; Fu, Q. Development and application of deep convolutional neural network in target detection. In *AIP Conference Proceedings*; AIP Publishing LLC: Xi'an, China, 2018; p. 040036.
26. Ding, Y.; Zhao, X.; Zhang, Z.; Cai, W.; Yang, N. Multiscale Graph Sample and Aggregate Network With Context-Aware Learning for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 4561–4572. [\[CrossRef\]](#)
27. Ding, Y.; Zhao, X.; Zhang, Z.; Cai, W.; Yang, N.; Zhan, Y. Semi-Supervised Locality Preserving Dense Graph Neural Network With ARMA Filters and Context-Aware Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–12. [\[CrossRef\]](#)
28. Ding, Y.; Zhang, Z.; Zhao, X.; Cai, Y.; Li, S.; Deng, B.; Cai, W. Self-Supervised Locality Preserving Low-Pass Graph Convolutional Embedding for Large-Scale Hyperspectral Image Clustering. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–16. [\[CrossRef\]](#)
29. Parico, A.I.B.; Ahamed, T. Real time pear fruit detection and counting using YOLOv4 models and deep SORT. *Sensors* **2021**, *21*, 4803. [\[CrossRef\]](#)
30. Yuan, N.; Kang, B.H.; Xu, S.; Yang, W.; Ji, R. Research on image target detection and recognition based on deep learning. In Proceedings of the 2018 International Conference on Information Systems and Computer Aided Education (ICISCAE), Changchun, China, 6–8 July 2018; pp. 158–163.
31. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. 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.
32. Girshick, R. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
33. Zhang, B.; Zhang, Y.; Pan, Q. Irregular Target Object Detection Based on Faster R-CNN. In Proceedings of the IOP Conference Series: Earth and Environmental Science, Xi'an, China, 15–16 December 2018; p. 042111.
34. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497v2. [\[CrossRef\]](#)
35. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*. [\[CrossRef\]](#)
36. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
37. Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; Lin, D. Libra r-cnn: Towards balanced learning for object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 821–830.
38. Fu, X.; Li, K.; Liu, J.; Li, K.; Zeng, Z.; Chen, C. A two-stage attention aware method for train bearing shed oil inspection based on convolutional neural networks. *Neurocomputing* **2020**, *380*, 212–224. [\[CrossRef\]](#)
39. Guo, J.; Hou, Z.; Xie, X.; Yao, S.; Wang, Q.; Jin, X. Faster R-CNN Based Indoor Flame Detection for Firefighting Robot. In Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 6–9 December 2019; pp. 1390–1395.
40. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
41. 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.
42. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767. [\[CrossRef\]](#)
43. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European conference on computer vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
44. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
45. Cao, Z.; Liao, T.; Song, W.; Chen, Z.; Li, C. Detecting the shuttlecock for a badminton robot: A YOLO based approach. *Expert Syst. Appl.* **2021**, *164*, 113833. [\[CrossRef\]](#)
46. Li, P.; Zhao, W. Image fire detection algorithms based on convolutional neural networks. *Case Stud. Therm. Eng.* **2020**, *19*, 100625. [\[CrossRef\]](#)
47. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934. [\[CrossRef\]](#)
48. 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. [\[CrossRef\]](#)
49. Zhao, Z.; Han, J.; Song, L. YOLO-highway: An improved highway center marking detection model for unmanned aerial vehicle autonomous flight. *Math. Probl. Eng.* **2021**, *2021*, 1205153. [\[CrossRef\]](#)
50. Misra, D. Mish: A self regularized non-monotonic neural activation function. *arXiv* **2019**, arXiv:1908.08681. [\[CrossRef\]](#)
51. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [\[CrossRef\]](#)

52. Yang, D.; Zhong, X.; Gu, D.; Peng, X.; Hu, H. Unsupervised framework for depth estimation and camera motion prediction from video. *Neurocomputing* **2020**, *385*, 169–185. [[CrossRef](#)]
53. Cheng, Q.; Li, H.; Wu, Q.; Ngan, K.N. Hybrid-loss supervision for deep neural network. *Neurocomputing* **2020**, *388*, 78–89. [[CrossRef](#)]
54. Li, S.; Li, Y.; Li, Y.; Li, M.; Xu, X. YOLO-FIRI: Improved YOLOv5 for Infrared Image Object Detection. *IEEE Access* **2021**, *9*, 141861–141875. [[CrossRef](#)]
55. Jia, X.; Zhu, C.; Li, M.; Tang, W.; Zhou, W. LLVIP: A visible-infrared paired dataset for low-light vision. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 3496–3504.
56. Xu, Z.; Zhuang, J.; Liu, Q.; Zhou, J.; Peng, S. Benchmarking a large-scale FIR dataset for on-road pedestrian detection. *Infrared Phys. Technol.* **2019**, *96*, 199–208. [[CrossRef](#)]
57. Dunning, A.; Breckon, T. Fire Image Data Set for Dunning's 2018 study-PNG still image set. *Durh. Univ.* **2018**. [[CrossRef](#)]
58. Foggia, P.; Saggese, A.; Vento, M. Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1545–1556. [[CrossRef](#)]
59. Dimitropoulos, K.; Barnpoutis, P.; Grammalidis, N. Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *25*, 339–351. [[CrossRef](#)]
60. Zeng, Y.; Zhang, L.; Zhao, J.; Lan, J.; Li, B. JRL-YOLO: A novel jump-join repetitious learning structure for real-time dangerous object detection. *Comput. Intell. Neurosci.* **2021**, *2021*, 5536152. [[CrossRef](#)]