Flame Image Detection Algorithm Based on Computer Vision

Xiaoqing Sun, Wenhua Cui, Ye Tao, and Zhaoyang Wang

Abstract—**Fire is one of the most common disasters for human beings. It is also one of the disasters that cameras can easily catch. In order to detect a series of building fires efficiently, a fire image detection algorithm based on YOLOv4 is proposed in this paper. This algorithm can realize the real-time fire warning by identifying all images in the video. The comparison of different evaluation results found that the YOLOv4 fire image detection algorithm using both optimization algorithms achieved higher AP and recall rates.**

Index Terms—**Computer vision, Object detection, Fire alarm, YOLOv4**

I. Introduction

Fire is a coexistence of pros and cons for human beings. It illuminates human civilisation's direction and brings life to countless disasters. Fire not only causes property damage to human beings but also threatens safety, which makes humans alert to fire. Therefore, it is necessary to develop a real-time fire detection system to reduce fire threats to human life and property safety.

There are many ways to detect building fires. One of the most common and accurate detection methods is detecting flame images through surveillance cameras.[1] Early computer vision-based flame image detection algorithms mainly extract the color information of flames through feature engineering. However, because of many objects in buildings with similar colors to flames, this method often misidentifies other objects as flames. Subsequently, researchers combined the temporal difference method with it and proposed a video fire recognition method based on color space and moving object detection.[2] The problem of a high false detection rate in the complex indoor scene based on the algorithm still results in a waste of human resources.

With the continuous development of deep learning, especially after AlexNet proposed to apply convolutional neural networks to image recognition tasks, deep learning algorithms based on convolutional neural networks can extract deep-level features through many convolution operations.[3] Some researchers have focused on using convolutional neural networks for flame image detection. In 2018, Muhammad et al. used a convolutional neural network to realize the detection task of flame images, and the video recognition accuracy rate reached 94.45%.[4] Due to the continuous iteration of the algorithm, the shortcomings of the algorithm used by Muhammad's team are that it cannot locate the flame, has low accuracy in complex scenes, and is gradually replaced by algorithms with high accuracy and efficient positioning. The primary method to improve the performance of convolutional neural network is to improve the recall rate, precision rate, detection speed and the coverage of the predicted frame and the actual object. In terms of improving the algorithm's accuracy, Avula et al. adopted a method based on fuzzy entropy optimization threshold and introduced spatial transformer networks to optimize the traditional neural network, which improved the model accuracy and recall rate.[5] In terms of improving the running speed and reducing the parameters, Danni Tang et al. proposed a forest fire detection algorithm based on channel pruning YOLOv3 and a forest fire detection algorithm based on MobilenetV3-YOLOv4.[6] It can reduce the parameters to 1/6 of the original YOLOv3 algorithm while improving accuracy. Yang et al. proposed a method for generating flame images based on a generative adversarial network to improve the quality of datasets.[7] The flame images are transferred to specific scenes, increasing the number of fire video samples in restricted scenes and ensuring the diversity of flames.

In order to improve the accuracy of fire detection and avoid unnecessary tragedies, this paper proposes two accuracy optimization methods based on YOLOv4. Firstly, this paper proposes an optimized YOLOv4 flame image detection algorithm based on the attention mechanism. The improvement method combines a multi-scale channel attention mechanism with a spatial attention mechanism and then applies it to the YOLO head part of the YOLOv4 algorithm. Secondly, a YOLOv4 flame image detection algorithm based on optimized vector concat is proposed. The proposed YOLOv4 flame image detection algorithm optimized based on attention mechanism and YOLOv4 flame image detection algorithm based on optimized vector concat are combined with the original YOLOv4 algorithm to judge the advantages and disadvantages of the two algorithms in different scenarios. Finally, many simulation results show the algorithm has high accuracy and practicability.

Manuscript received January 8, 2023; revised July 11, 2023. This work was supported by Joint Fund Project of the National Natural Science Foundation of China (U1908218), the Natural Science Foundation project of Liaoning Province (2021-KF-12-06), and the Department of Education of Liaoning Province (LJKFZ20220197).

Xiaoqing Sun is a lecturer of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: 1299095882@qq.com).

Wenhua Cui is a Professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (Corresponding author to provide phone: +86-133-0422-4928; e-mail: taibeijack@126.com).

Ye Tao is a lecturer of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: taibeijack@163.com).

Zhaoyang Wang is a graduate student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: 591824136@qq.com).

Ⅱ. Evaluation Metrics for Object Detection Tasks

A.Loss Function in Object Detection

In the target detection task, the role of the loss function is to measure the output error of the model for a single sample. The primary sources of errors are prediction box error, confidence error and category error in the one-stage object detection task. The prediction box error consists of two types of errors.[9] One is the error between the coordinates of the predicted object's centre point and the actual object's centre point. The other is the error between the length and width of the predicted object and the actual object. The small deviation of the large box is smaller than that of the small box. The square root of the width and height needs to be calculated when predicting the width and height of the object.[10] Confidence error refers to the error in the confidence of an object. Class error is the error caused by misclassifying objects. Through the above errors, the loss function formula of the one-stage target detection task can be calculated as follows:

$$loss=lbox+lobj+lcls \tag{1}$$

In the above formula, *lbox* is the prediction frame error, *lboj* is the confidence error, and *lcls* is the category error. In the target detection task, these three kinds of errors often require different loss functions for different problems.[11] Therefore, fully understanding the characteristics of different loss functions can help get better detection task results.

B.Multiple Intersection Over Union

Intersection over union(IoU) is one of the calculation methods of the prediction box error (lbox) in Equation 1. When predicting the position of the target, prediction box A needs to be used to match the ground truth box B.[12] However, errors often occur in the matching process. Therefore, an evaluation metric must be defined to reward those prediction boxes that match ground truth boxes better. Intersection over union is an index to evaluate the quality of predicted position matching.[13] If the coordinates and size of the prediction box A match the ground truth box B, there is a higher correlation between A and B. The value of IoU is also large. The formula for calculating IoU is:

$$IoU=A\cap B\div A\cup B \tag{2}$$

Among them, A ∩ B is the overlapping area between the prediction box A and the ground truth box B.[14] A ∪ B is a total area occupied by the prediction box A and the ground truth box B. Usually, the threshold of IoU is set to 0.5. If the IoU exceeds 0.5, the algorithm successfully detects the object. Otherwise, the detection fails.[15] IoU also cannot accurately reflect the degree of coincidence between the prediction box A and the ground truth box B when they overlap. Even if the numbers are equal, the two boxes may not have the same degree of coincidence. Only considering the overlapping area but not the overlapping shape will affect the loss calculation.[16] In order to solve the problem that the prediction box A is too far away from the ground truth box B, which is not considered by IoU, GIoU (Generalized Intersection over Union) is mentioned. It solves the above problem by adding a minimum box C that can be completely covered in the calculation. The calculation formula of GIoU is as follows:

$$GIoU=IoU-\frac{C-(A\cup B)}{C} \tag{3}$$

When GIoU is the loss function and boxes A and B do not intersect, loss equals one minus GIoU. The value of A ∪ B remains unchanged. The smaller the value of C, the larger the value of GIoU. At this time, the distance between the prediction box A and the ground truth box B will continue to approach. The proposal of GIoU solves the above problems to a certain extent. However, GIoU needs more iterations to converge. The performance is the same as IoU when one box contains another box. Therefore, based on GIoU, Zheng et al. further proposed DIoU(Distance-IoU loss). The calculation formula is as follows:

$$DIoU=1-IoU+\frac{\rho^2(A,B)}{c^2} \tag{4}$$

In the above formula, ρ (A, B) is the Euclidean distance between the prediction box A and the center point of the ground truth box B. *c* is the diagonal distance of C in DIoU.[17] DIoU penalty term is based on the ratio of the distance between the center point and the diagonal. DIoU avoids the situation that the loss value is too large and difficult to optimize due to the generation of a large outer box when the prediction box A is far away from the ground truth box B.[18] When a box contains another box, ρ (A, B) can be used to make an evaluation of the coincidence of the prediction box A and the ground truth box B. Complete-IoU Loss(CloU) calculates the coincidence of the center point of the prediction box A and the ground truth box B on the basis of Diou and introduces the aspect ratio of the box. The calculation formula is as follow.

$$CIoU=1-IoU+\frac{\rho^2(A,B)}{c^2}+\alpha v \tag{5}$$

α is the weight function. *v* is used to measure the consistency of the aspect ratio. This paper uses CIoU as the calculation method for the prediction box error in the loss function.[19]

C.Non-maximum Suppression in Object Detection

Non-maximum suppression is to suppress elements that are not maximum values. It is usually used to calculate the intersection-over-union to select those prediction boxes with the best intersection ratio. Non-maximum suppression is also a standard local maximum search in target detection. In the target detection task, non-maximum suppression needs to traverse all areas in an image.[20] Then, calculate the prediction box with a score greater than the threshold in each area. Instead of calculating the prediction box with a score less than the threshold, It can reduce the calculated number of boxes. Finally, judge the class and score of the box obtained in the previous step.

D.Target Detection Accuracy Evaluation Index

mAP(mean Average Precision) is an accuracy evaluation index mainly used in object detection. The average value of the area under the curve of precision and recall of all objects to be detected in a target detection task is the average value of each AP curve. mAP is an evaluation metric for the entire dataset. All images are required to participate in the evaluation. AP (Average Precision) is an evaluation index for a certain category in the data set. In calculating the VOC data set, AP selects the maximum value of the corresponding precision rate for each different recall rate in the calculation result and connects these corresponding maximum values of precision rate into a curve. The connected curve is the P-R (Precision-Recall) curve. The area under the curve is the calculated AP.

$$Precision=\frac{TP}{TP+FN} \tag{6}$$

$$Recall=\frac{TP}{TP+FN} \tag{7}$$

TP (True Positive) refers to the samples assigned as positive, and the correct samples are assigned. Currently, the IoU of the prediction box A and the ground truth B is greater than the set threshold. FP (False Positive) refers to samples assigned as positive but wrongly assigned. When misclassified, IoU of the predicted box A and the actual box B is greater than the set threshold.[21] FN (False Negative) refers to samples that are assigned as negative and are assigned incorrectly. When misclassified, IoU of the prediction box A and the actual box B is less than the set threshold. TN (True Negative) refers to the samples that are assigned as negative and the assignment is correct. In the case of correct classification, the intersection ratio of the prediction box A and the ground truth box B is less than the set threshold.

E. Flame Image Detection Process Based on Convolutional Neural Network

The object detection task is to identify objects in images and determine the location information of objects. The advantage of using the target detection method for fire detection is that it further realizes the localization of the flame. It is helpful for better processing the flame information in the image. In addition to judging whether a fire has occurred in the image, it can also judge the process of the fire.[22] As shown in figure 1, fire detection is performed as a frame of pictures according to the identified flame image information.

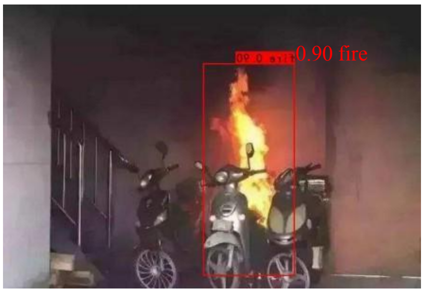


Fig.1 Example of flame image detection

First, to achieve the above results, preprocess the input image and convert a raw image to an "RGB" format image. Then the sliding window counts the confidence of each area's flame. Regions with a confidence more significant than a set threshold are compared with the confidence of the neighborhood by using non-maximum suppression. Finally, the optimal area is selected for the prediction box to be drawn.

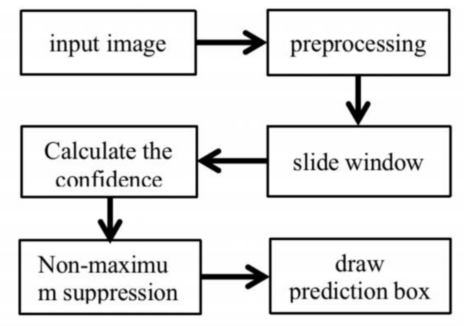


Fig.2 Step of flame image detection

Figure 2 is the overall flow chart of flame image detection. The first step is to build a high-quality flame image detection data set containing multiple scenes. Next, some initial hyperparameters are set for the model used to ensure that it can converge to the optimal result as quickly as possible under the condition that the computer hardware supports it. Then, input the data from the data set into the algorithm for training. Finally, the images captured by the camera are successively input into the saved model. A fire is determined by whether all of the images captured.

F. Production and Use of Flame Image Detection Dataset

The flame image detection data set used in this paper consists of 2491 pictures, including images containing flames in indoor, forest, vehicle and other scenes and images without flames in indoor, forest, vehicle, street lights, sunset and other scenes without flames. The following figure shows the process of the data set.

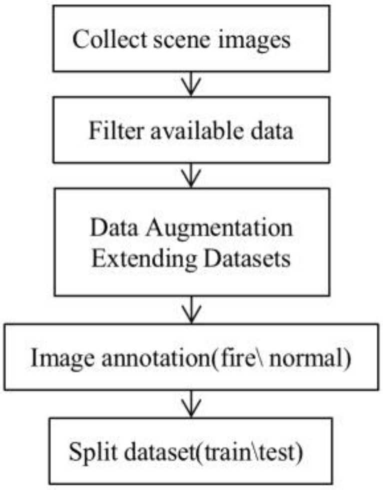
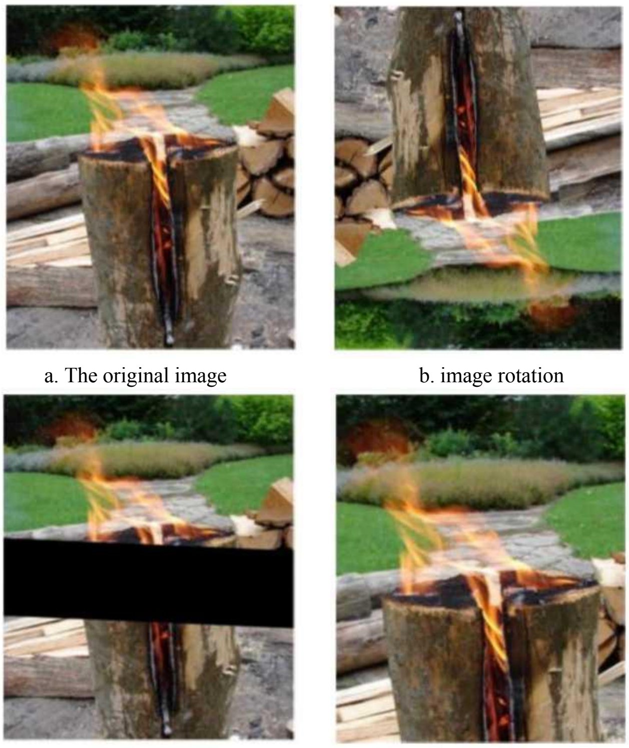

Collect scene images
Filter available data
Data Augmentation Extending Datasets
Image annotation(fire\ normal)
Split dataset(train\test)

Fig.3 Dealing process of flame image dataset

The data set in this article is collected from Google, CSDN and other websites, including about 3000 flame image scene pictures and flame-free images. Subsequently, these 3000 images were sorted and screened. Discard images that are difficult to label and unclear. Increase the number of partial samples by methods such as rotation, occlusion, and cropping in data enhancement.

Finally, classify and count the selected high-quality pictures. If there are too few negative samples in a scene, increase the number of negative samples. If there are too few positive samples in a particular scene, increase the number of positive samples. The data set labelling rule used in this paper is to label the flame individuals seen fully.



a. The original image

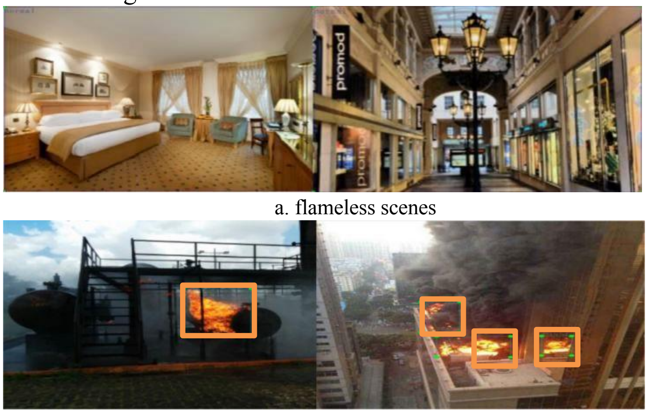
b. image rotation

b. random occlusion

d. image cropping

Fig.4 Data enhancement contrast of an image in flame image dataset

Mars that is too far away from the fire source are not marked. It is also considered as one flame when two flames overlap or an elongated object blocks one flame, which is shown in figure 4.



a. flameless scenes

b. flame scenes

Fig.5 Part of the flame image detection dataset

The flameless scene in figure 5 is the flameless state in a certain scene. The dataset labels this type of image with the "Normal" label, which means there is no flame in the scene. A fire scene is a scene that contains fire. In this data set, the smallest area that can contain a flame individual is marked as the "Fire" label, which means that there is a flame in this area. At the same time, mars, far away from the individual and with no combustible objects attached, are not marked.

TABLE I
THE COMPOSE OF FIRE IMAGE DETECTION DATASET

	train		trainval		test		total data set	
	Images	Objects	Images	Objects	Images	Objects	Images	Objects
Fire	-	-	1763	2013	190	213	1953	2226
Normal	-	-	478	478	60	60	538	538
Total	2017	-	2241	2491	250	273	2491	2764

As can be seen from the table, the flame image detection data set has a total of 1953 pictures of flame scenes, 2226 labelled flame individuals and 538 pictures without flame scenes. The number of images used for training accounts for 81% of the total dataset. The number of images used for validation accounts for 9% of the total. The number of images used for testing accounts for 10% of the total. Images represent the number of pictures. Objects represent the number of marked objects.

G. Prior Bounding Box

K-means clustering method is a simple and efficient clustering algorithm. It can be used to obtain prior boxes for object detection tasks. The first step of K-means clustering needs to determine the number k of centroids and cluster out k according to centroids. The second step is to assume k random data points in the dataset as centroids. Calculate the distance between each non-centroid data and each centroid. Divide each data into the set to which its nearest centroid belongs, a total of k sets. The third step is to reset the centroid. The position of the centroid is the average value of each set in step two. The fourth part is iteration. Set the centroid obtained in the previous step as the initial centroid and repeat steps two and three until the centroid does not change or reaches the preset number of iterations. Then use the final set as the result of K means clustering.[22]

This article uses the K-means clustering method and sets the value of k to nine in the flame image detection dataset. Nine prior boxes were clustered. Among them, there are three prior frames for detecting large flames, namely: (209,360), (310,187), (414,413). There are three prior boxes for detecting medium-sized flames, namely: (109,241), (111,115), (174,238). There are three prior boxes for detecting small flames, namely: (28,50), (57,157), (72,89). Among them, w in each (w, h) represents the width of the prior box, and h represents the height of the prior box.

Ⅲ. YOLOV4 FLAME IMAGE DETECTION ALGORITHM BASED ON ATTENTION MECHANISM

A. YOLO Algorithm

The YOLO algorithm is an efficient target detection algorithm proposed by Redmon. The YOLO algorithm solves target detection as a regression problem. After an inference on the input image, all objects' orientation, category and confidence can be detected.[23]

In the training phase, YOLO uses a network of 20 convolutional layers, an average pooling and fully connected layers. Moreover, Setting the input image resolution to 224*224 was used to train the 1000 classification data of the ImageNet dataset. The algorithm divides the input image into S^2 small grids in the prediction phase. Each grid is responsible for detecting objects in the grid. If the coordinates of the center of an object are in a certain grid, this grid is responsible for detecting the object. S is the quotient of the length (width) of the image and the length (width) of each small grid.

After finding which small grid each object should be predicted by, each small grid draws B rectangular areas containing objects and calculates the probability that the object belongs to a certain category C. In the YOLO algorithm, 5 data values contain the information that a rectangular area of an object has, namely x, y, w, h, and c. x and y are the horizontal and vertical coordinates of the center position of the rectangular area containing the object predicted by the current grid. w and h are the width and height of the rectangular area where the object is located. c is the confidential information, which reflects whether the object is contained in the rectangular area currently containing the object and the accuracy of the position when the object is contained. In the actual training process, the image's width w and height h can also be normalized to the [0,1] interval. Calculate the offset value of the center position of the rectangular area containing the object in x and y relative to the current small grid position, which can also be normalized to the [0,1] interval.

B. Application of Attention Mechanism in Flame Image Detection

(1) The role of attention mechanisms in neural networks

There are three reasons why computer vision-based flame image detection tasks use attention mechanisms.

a. These attention mechanism models are excellent models proposed in recent years. Moreover, researchers have tried to use these models to optimize flame image detection algorithms. This shows that using the attention mechanism to optimize the model can improve results by replacing some structures in YOLOv4.

b. The attention mechanism can improve the interpretability of neural networks. Using the attention mechanism to optimize the YOLOv4 algorithm model can help the model fairness, accountability and transparency of the neural network. It is more convenient to find more suitable parameters when modifying model parameters.

c. The attention mechanism helps to overcome problems such as performance degradation as the input length increases or low computational efficiency caused by unreasonable input order in the recurrent neural network. In particular, the smaller the model, the more images it can process at a time in image detection tasks that require batch normalization.

(2) Channel attention in optimizing flame image detection process

Channel attention proposes a method of adjusting the relationship between channels so that the algorithm can better learn the importance of features between different channels. For example, in the flame image detection task, the researchers used squeeze and excitation to optimize the backbone of the YOLO series of neural networks in the early time. They believed such optimization could increase the model's sensitivity to the color dimension. However, this paper finds that adding an attention mechanism to the backbone network can significantly increase the training time of the flame image detection model. Using the attention mechanism in the regression network of the model only adds a small amount of training time. Therefore, compared to using the attention mechanism in the backbone part, adding the attention mechanism in the regression network part can save the time of training the model and improve the accuracy of the network.

Channel attention proposes the Squeeze-and-Excitation (SE) module. Firstly, the feature map obtained by convolution is squeezed to obtain the global feature of the channel dimension. Then exciting the global feature, learning the difference between each channel, and obtaining each channel's weight. Finally, it is multiplied with the initial feature map to obtain the final features. Therefore, the SE module is an attention mechanism in the channel dimension. On the one hand, it can make the model pay more attention to the channels with the maximum information and suppress unimportant channels. On the other hand, SE modules are general, which means that SE modules can be embedded in various existing network models. The basic structure of the SE module is shown in Figure 6.

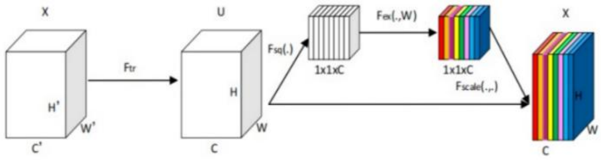


Fig. 6 The basic structure of SE model

In the SE module, the input X is first converted into an output U through a standard convolution operation (3*3 convolution). The width, height, and number of channels are kept unchanged. Assuming that the current output U is not optimal, the importance of each current channel is different. Some channels have more effect, and others have little or no effect. Next, perform global average pooling to obtain the 1*1*C feature for each output channel. New channel weights are obtained after full connection-activation function-full connection-Sigmoid. Finally, each channel of the original output U is multiplied by the corresponding weight with each element of the corresponding channel of the latest feature and the weight to obtain the result of SE attention. The global average pooling operation for each channel H*W is called Squeeze. The process of multiplying each element of each original H*W feature by the weight of the corresponding channel to obtain the final feature of a single SE module is called excitation.

(3) Application of pyramid segmentation attention module (PSA) to SE attention optimization

The PSA module proposed by the pyramid split attention is replaced by the 3*3 convolution in the ResNet network Bottleneck to obtain a new module used in the backbone network, which can provide stronger multi-scale feature expression and serve downstream tasks. As a new efficient pyramidal attention segmentation module, the PSA module is similar to the SE module. However, the PSA module has an extra segmentation operation. After the segmentation, multi-scale feature extraction is carried out. This module can extract finer-grained multi-scale spatial information while establishing longer-distance channel dependencies. Due to the strong flexibility and scalability of the EPSA module, it can be directly applied to various computer vision network architectures. The PSA module can learn richer multi-scale feature representations and adaptively perform feature recalibration on multi-dimensional channel attention weights.

The PSA module is mainly realized through four steps. The first step is to use the SPC module to split the channel. Multi-scale feature extraction is performed by targeting the spatial information on each channel feature map. The second step is to use the SEWeight module to extract SE attention for feature maps of different scales. Get channel attention vectors at each different scale. The third step is to use Softmax to

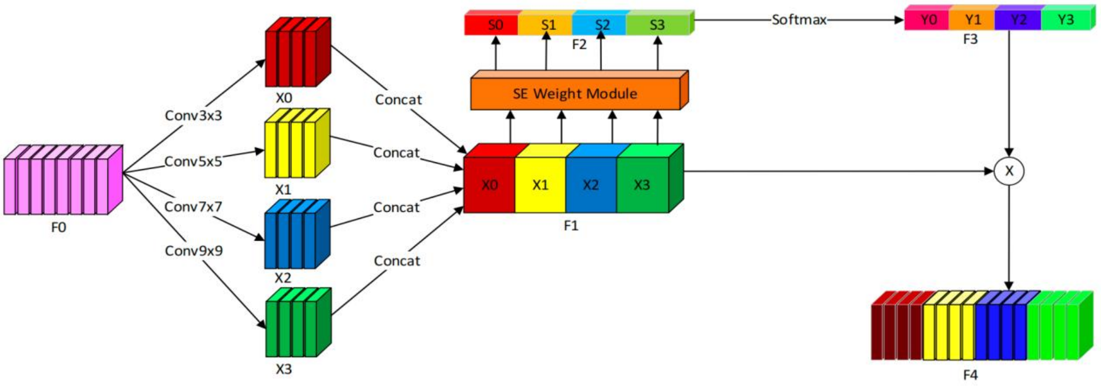


Fig.7 The main structure of PSA model

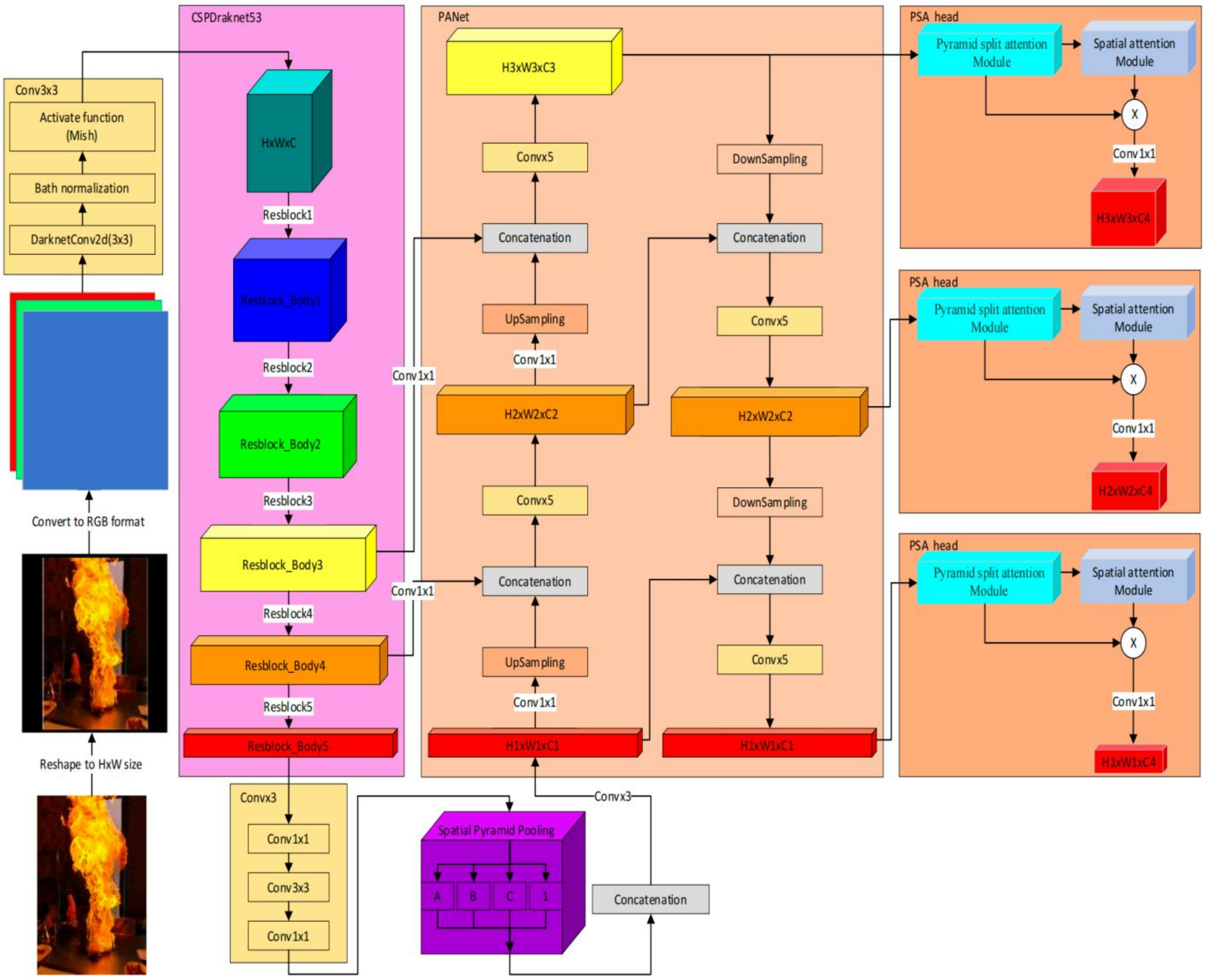


Fig.8 The structure of optimize YOLOv4 algorithm

perform feature recalibration on multi-scale channel attention vectors and obtain the attention weights after the new multi-scale channel interaction. The fourth step is to perform an element-wise dot product operation on the recalibrated weights and the corresponding feature maps. The output is a feature map after attention weighting of multi-scale feature information. The main structure of PSANet is shown in Figure 7.

The PSA module first uses the convolution of the four cores Conv3*3, Conv5*5, Conv7*7, and Conv9*9 for *F0* and then concatenates to obtain *F1*. At the same time, grouped convolution is used to reduce the parameters of this process. The relationship between the size of the convolution kernel and the number of grouped convolution groups can be written as $G=2^{(K-1)/2}$. The number *K* is the size of the convolution kernel, and *G* is the number of groups. Then *F1* gets *F3* through the SEWeiget module and Softmax. Finally, *F4* is obtained by multiplying *F1* and *F3*. Although this optimization increases the model's parameter amount and calculation time to a certain extent, its effect on improving accuracy is also obvious.

(4) An optimized flame image detection method using spatial attention mechanism (SAM)

The spatial attention mechanism is a convolutional block attention module (CBAM) module. CBAM is a module combining the spatial and SE attention mechanisms simultaneously. Compared with SE attention, CBAM can achieve better results in many scenarios. The model with the added CBAM module has better performance and better interpretability than the original model and can also pay more attention to the flame itself in the image. Therefore, the detection of flame images by the YOLO series of algorithms in recent years often needs to add the idea of CBAM to the backbone. The structure of the SAM module is shown in figure 9.

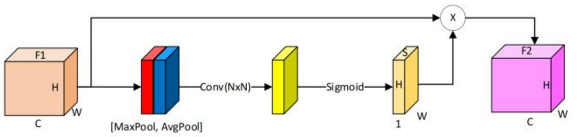


Fig.9 The basic structure of SAM

C. Algorithm Optimization

In order to improve the accuracy of high flame detection, this paper optimizes the channel attention mechanism and the spatial attention mechanism and applies it to the regression network of YOLOv4. The main process is as follows:

The attention mechanism is usually a plug-and-play optimization algorithm. It is an efficient module that can replace the residual network to a certain extent. However, adding the attention mechanism to the model backbone that has already used the residual network will significantly increase the algorithm's training time and detection time, regardless of whether it will improve the accuracy. So the attention mechanism should be added outside the residual structure.

YOLOv4 is based on YOLOv3 by optimizing the body and neck. However, YOLOv4 still uses the YOLO head of YOLOv3 in the head part and has not been optimized. In the YOLO series of algorithms, the YOLO head acts as a classifier. The objects in the picture can be located and classified through the YOLO head. This is an essential step in the YOLOv4 algorithm. Therefore, optimizing the YOLO head will improve the precision and recall of YOLOv4. The structure of the YOLOv4 algorithm optimized by the attention mechanism is shown in Figure 8. The method proposed in this paper is the PSA head structure in the last structure of the network.

As shown in figure 8, the YOLOv4 algorithm requires image preprocessing. The size of the input image is unified to 416*416 pixels by scaling down the large image and adding a black area around the small image. Then, the information on the three RGB channels is extracted from each converted picture. The picture's three component vectors of 'red, green, and blue' are obtained. If the number of images input in a time is greater than 1, each input feature needs to be concatenated in the channel dimension. Finally, the concatenated results of the channel dimension are subjected to convolution, normalization and Mish activation function operations. For each input image of the same bath size, a 416*416*32 feature will be obtained. After obtaining the initial features, the backbone of YOLOv4 uses CSPDarknet53 to perform feature extraction. This process is carried out by multiple sets of residual structures (Resblock). Each set of residual blocks contains two paths. One is a path with only one 1*1 convolution. The length and width are halved and the number of channels is doubled. The other is that the length and width are halved after five convolutions. The channel is finally doubled after three 3*3 convolution expansions and two 1*1 convolution reductions. The features obtained from the two paths are spliced in the channel dimension. The length and width are halved, and the channel becomes four times the number of initial channels. Then use a 1*1 convolution to extract this feature. Finally, the length and width of a feature are halved and the number of channels is doubled. After five residual structures, the backbone network finally obtained five features, which are 208*208*64, 104*104*127, 52*52*256, 26*26*512 and 13*13*1024. The three features of 52*52*256, 26*26*512 and 13*13*1024 are reserved for the neck operation of YOLOv4. The rest of the features are discarded.

The features of two scales, 52*52*256 and 26*26*512, need to be converted into 152*52*128 and 26*26*256 features by reducing the number of channels by half through 1*1 convolution, respectively. The features of 13*13*1024 need to be converted into 13*13*512 features by reducing the number of channels by half through the spatial pyramid pooling module. Finally, the obtained three features of 52*52*128, 26*26*256 and 13*13*512 are used as the three input features of PANet for feature fusion. In the neck network, the 13*13*512 features are converted to 26*26*256 features through upsampling. This feature is merged with the 26*26*256 feature obtained by the 1*1 convolution of the backbone network and the 1*1 convolution to obtain a new feature. The new feature upsampling and fused with 52*52*128 features and 1*1 convolution to get the first input of the PSA head. Next, we continue to downsample and feature fusion of the obtained results to obtain the remaining two PSA head inputs.

The input features obtained will get a feature with 18 channels and 18 dimensions of information through the PSA head, including location, type, confidence, etc. So far, the final result of the flame image detection algorithm has been obtained. The PSA head module consists of three parts:the PSA module, the SAM module and a 1*1 convolution. The structure of the PSA head is shown in figure 10.

In the PSA head module, the feature image F0 first obtains new feature images X0, X1, X2 and X3 through four convolutions with kernel sizes of 3*3, 5*5, 7*7 and 9*9, respectively. In this process, each convolution will halve the number of channels of the feature image. After that, X0, X1, X2 and X3 are spliced in the channel dimension to obtain F1 with twice the number of channels as F0. The formula for calculating F1 is:

$$F1=Cat([X0,X1,X2,X3]) \tag{8}$$

The calculation formulas of X0, X1, X2 and X3 are:

$$X_n=(f*g_n)(u,v)=\sum_i \sum_j f(i,j)g_n(u-i,v-j) \sum_i \sum_j a_{i,j}b_{u-i,v-j} \tag{9}$$

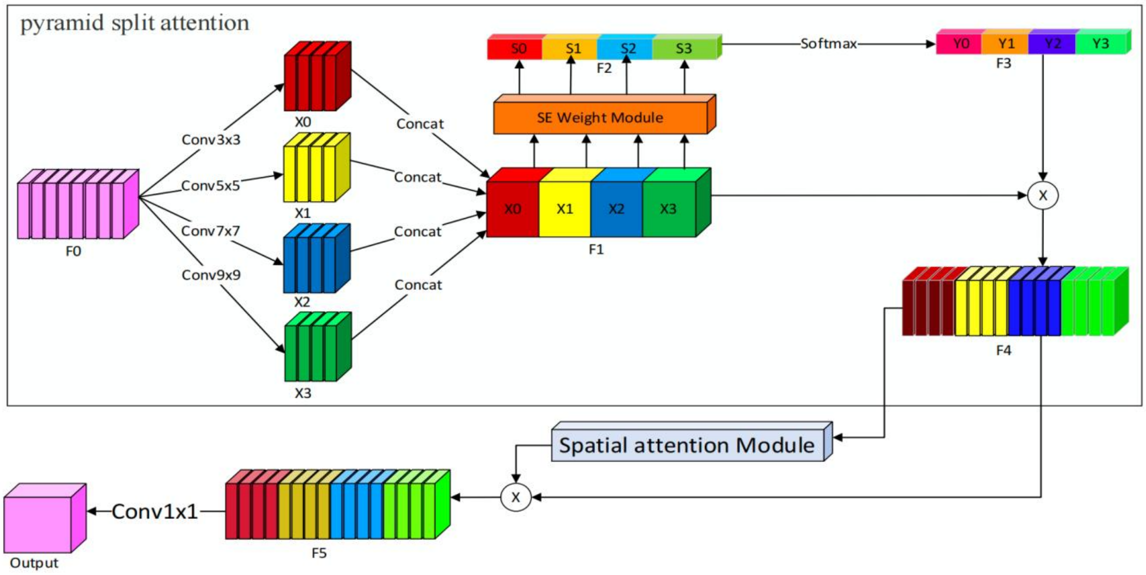


Fig.10. The structure of PSA head

In the formula, *f(u,v)*is the value of each pixel of the feature image to be detected. $f(u,v)=a_{u,v}$. a is an N*N matrix. $g(u,v)$ is value for each point of the convolution. $g(u,v)=b_{u,v}$. g_n is 3*3 matrix, 5*5 matrix, 7*7 matrix and 9*9 matrix respectively. Subsequently, the channel attention module (SE) is used to extract channel features for F1 to obtain F2. F2 is passed through a softmax to obtain F3. The calculation formula of Y0, Y1, Y2 and Y3 in F3 is

$$Y_i=\sigma(g(z,W))=\sigma(W_2\delta(W_1z)) \tag{10}$$

σ is the sigmoid activation function. W_1, W_2 is fully connected layer. δ is the ReLU activation function. z is the squeeze operation, calculated as:

$$z_c=F_{sq}(u_c)=\frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{w}u_c(i,j) \tag{11}$$

Then multiply F3 and F1 to get F4. The formula for F4 is:

$$F4=Cat([X_0\otimes Y_0,X_1\otimes Y_1,X_2\otimes Y_2,X_3\otimes Y_3]) \tag{12}$$

Then use the spatial attention mechanism to extract F4 features to get F5. The formula for calculating F5 is as followed:

$$F5=\sigma(f^{N\times N}[AvgPool(F4);MaxPool(F4)]) \tag{13}$$

Finally, F5 obtains the final result through a 1*1 convolution. This process is a regular convolution.

D. Experiment and Result Analysis

This paper compares the YOLOv4 image detection algorithm, the optimized YOLOv4 flame image detection algorithm (YOLOv4*), the YOLOv3 flame image detection algorithm, the optimized YOLOv3 flame image detection algorithm (YOLOv3*), and the SSD flame image detection algorithm. The results are shown in Table 2.

TABLE Ⅱ
RESULTS OF THE PROPOSED METHOD AND OTHER FIRE DETECTION METHODS

Authors	Algorithm	AP	F1-score	Precision	Recall	Miss rate
Zheng et al.	Efficientdet	91.92%	0.88	88.21%	87.79%	0. 14%
Wu et al.	SSD	94.07%	0.90	93.03%	87.79%	0. 12%
LI and Zhao	YOLOv3	94.54%	0.95	97.54%	92.96%	0.07%
Proposed*	YOLOv3*	95.42%	0.95	97.09%	93.90%	0.06%
Kumar et al.	YOLOv4	95.12%	0.94	94.76%	93.43%	0.07%
Proposed*	YOLOv4*	95.88%	0.95	96.15%	93.90%	0.06%

It can be seen from Table 2 that the YOLOv4 flame image detection algorithm proposed in this paper improves the data in all aspects compared with the algorithm used by Kumar et al.[8] After applying the PSA head proposed in this paper to the YOLOv3 algorithm, some evaluation indicators are improved to a certain extent compared with Li et al. AP increased by 0.88%. The recall rate increased by 0.94%, and the miss rate decreased by 0.01.

(1) Comparison with the accuracy of the YOLOv4 image detection algorithm used by Kumar et al.

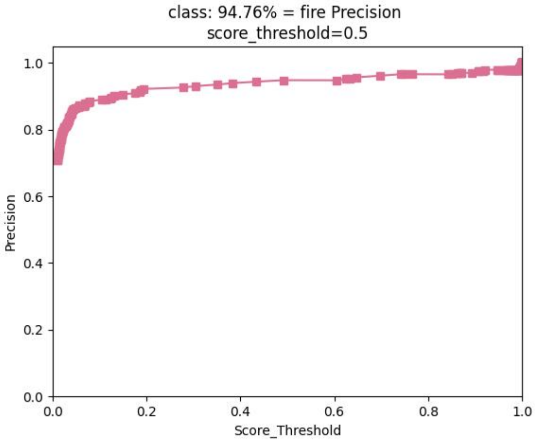


Fig.11 The precision curve of two YOLOv4 algorithms (1)

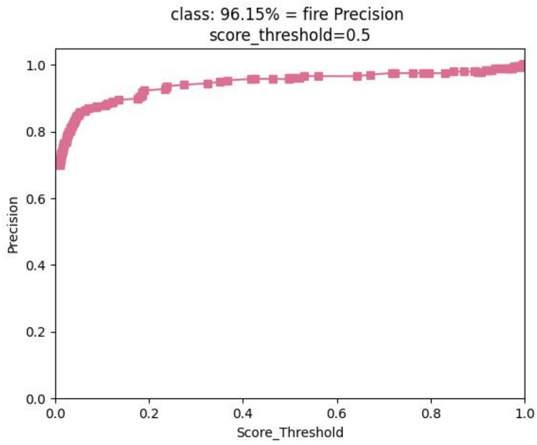


Fig.11 The precision curve of two YOLOv4 algorithms (2)

The figure above shows the image comparison of the accuracy rate of the YOLOv4 image detection algorithm used by Kumar et al. and the detection algorithm proposed in this paper as the score threshold increases. The score threshold is a decimal between 0 and 1, meaning all detection results less than the threshold are counted as precision. At the same time, connecting the precision rates under all score thresholds into a curve can see the change in the precision rate as the score threshold changes. Where "class: 94.76% = fire Precision; score_threshold = 0.5" means that when the confidence threshold is 0.5, the accuracy rate of individuals classified as fire is 94.76%. "class: 96.15% = fire Precision; score_threshold = 0.5" means that the accuracy of classifying individuals as fire is 96.15% when the confidence threshold is 0.5.

It can be seen from figure 11 that the accuracy of the YOLOv4 image detection algorithm continues to improve as a whole as the confidence threshold increases. The more stringent the flame discrimination requirements, the more likely it is to find areas where flames exist above that requirement. When the confidence threshold is set to 0.5, the accuracy rates of the two are 94.76% and 96.15%, respectively. The algorithm proposed in this paper will misjudge a normal scene as a flame scene 1.39 times less for every 100 frames of pictures.

(2) Comparison with the recall rate of the YOLOv4 image detection algorithm used by Kumar et al.

The figure below shows the image comparison of the recall rate of the YOLOv4 image detection algorithm used by Kumar et al. and the detection algorithm proposed in this paper as the score threshold increases. The figure shows that as the confidence threshold increases, the recall rate of the YOLOv4 algorithm decreases as a whole. The stricter the requirements for flame discrimination, the more it may not be possible to detect the presence of a flame in the area when it is known that there is a flame in a specific area of an image. When the confidence threshold is set to 0.5, the recall rates of the two are 93.43% and 93.90%, respectively. In the case of judging the individual as a flame with a confidence level higher than 50%, the algorithm proposed in this paper will correctly identify 0.47 more times per 100 frames of pictures on average.

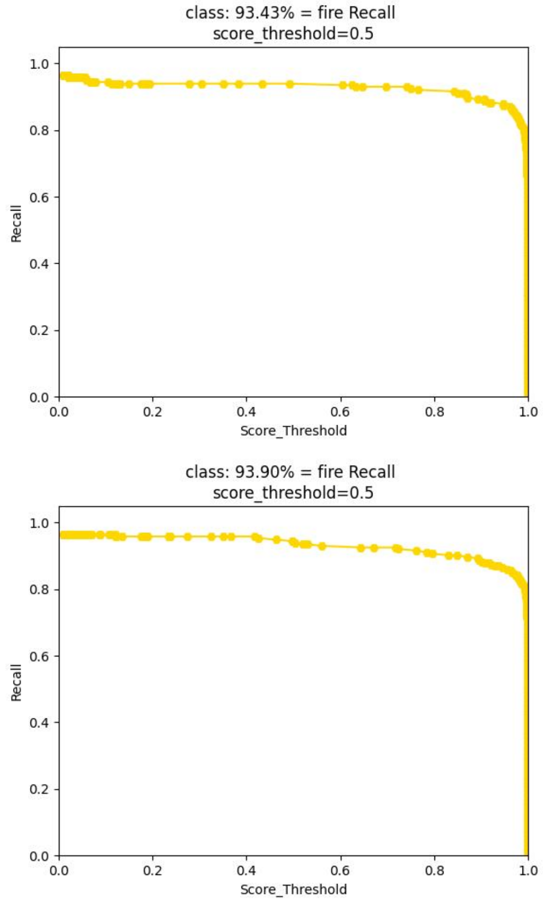


Fig.12 The recall curve of two YOLOv4 algorithms

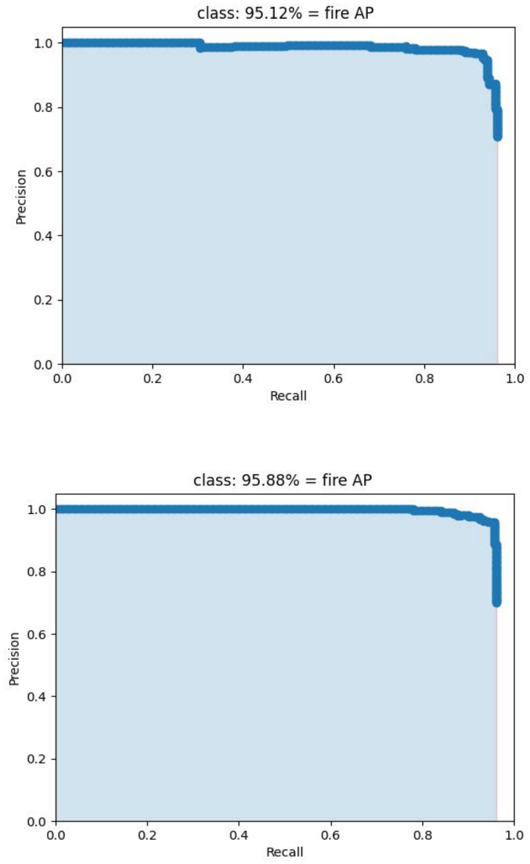


Fig.13 The AP curve of two YOLOv4 algorithms

(3) Comparison with the YOLOv4 image detection algorithm used by Kumar et al. on the AP evaluation index.

As the recall rate continues to increase, the confidence threshold at this time continues to decrease. The precision corresponding to the confidence threshold should be continuously reduced. Figure 13 shows the correspondence between precision and recall. Then, connect the precision rate values corresponding to each recall rate with a curve. The area of the curve enclosed by the curve and the positive part of the coordinate axis is divided by one as the comprehensive evaluation index (AP). It can be seen from the figure that the AP of the two algorithms are 95.12% and 95.88%, respectively. ‘class: 95.12% = fire AP’ means that the result of dividing the area of the part by 1 is 0.9512. ‘class: 95.88% = fire AP’ means that the area of the part divided by one is 0.9588.

(4)Visual comparison with the YOLOv4 image detection algorithm used by Kumar et al.

By using the optimized YOLOv4 algorithm to detect the flame image, it can be found that the optimized YOLOv4 algorithm not only has higher confidence in the target area but also can locate the flame better.

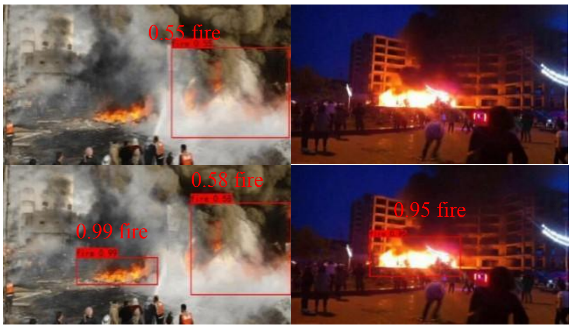
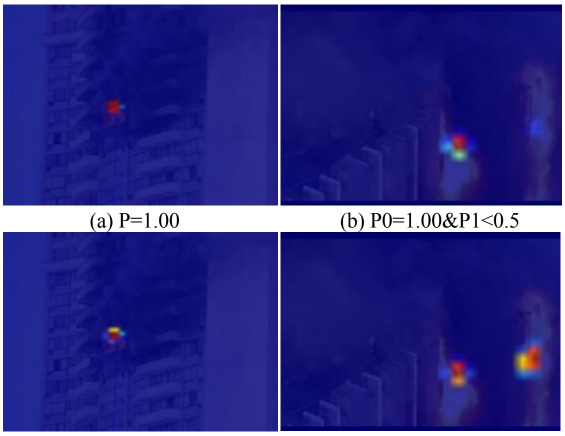


Fig.14 Recognition results of two kinds of YOLOv4 algorithms in some pictures

The upper part of the figure is the YOLOv4 image detection algorithm used by Kumar et al. The lower part is the optimized YOLOv4 flame image detection algorithm. The YOLOv4 image detection algorithm used by Kumar et al. identified only one flame in the image, and the confidence level was only 0.55. The optimized YOLOv4 algorithm can accurately judge two flames in the image with the confidence level of 0.58 and 0.99, respectively. The YOLOv4 image detection algorithm used by Kumar et al. could not successfully identify the flames in this scene. The optimized YOLOv4 flame image detection algorithm can fully identify the flames in the scene with a confidence level of up to 95%. Therefore, it can be seen that the optimized YOLOV4 flame image detection algorithm has a better detection effect on the scene of flames in practical applications. This algorithm has a higher resistance to the interference capacity of the environment and background. It can also adapt to the use of multiple environments.

Four heatmaps are drawn respectively through feature visualization with the YOLOv4 image algorithm used by Kumar et al. and the optimized YOLOv4 flame image detection algorithm. The detection ability of the network can be judged more intuitively through the heat map. Areas with redder colors represent greater weights and greater attention paid to that area by the network.

As seen from the heat map in figure 15, while both algorithms judge the area as having a flame with a confidence of 1, the original YOLOv4 algorithm emphasizes identifying a part of the flame. The optimized YOLOv4 algorithm will judge whether the flame exists based on multiple parts of the flame. Although, in some cases, paying more attention to a part of the flame will get better results, it is also likely to cause missed judgments. The YOLOv4 image detection algorithm used by Kumar et al. can determine the presence of a flame somewhere with a confidence level of 1 but fails to detect the less-characteristic flame on the right. The optimized YOLOv4 flame image detection algorithm can identify all the flames with confidence levels of 0.93 and 0.92, respectively.



(a) P=1.00 (b) P0=1.00&P1<0.5

(c)P=1.00 optimized (d)P0=0.93&P1=0.92optimized

Fig.15 Visualization results of two kinds of YOLOv4 algorithms

(5)Comparison of running speed (FPS) with the YOLOv4 image detection algorithm used by Kumar et al.

TABLE Ⅲ
RUNNING SPEEDS BETWEEN SOME YOLO ALGORITHMS

Algorithm	FPS of big images	FPS of middle images	FPS of small images
YOLOv3	69	67	66
YOLOv3 with PSA head	50	49	49
YOLOv4	48	47	47
YOLOv4 with PSA head	38	37	37

In addition to the evaluation indicators, such as precision rate and recall rate, the running speed of an algorithm is also the criterion for judging the algorithm's quality. The evaluation standard for running speed generally chooses frames per second (FPS). FPS refers to the number of image frames the algorithm can process per minute. The FPS of each algorithm in the YOLO series is shown in Table 3. It can be seen from the above table that the FPS of the optimized YOLOv4 flame image algorithm is about 10% lower than that of the algorithm used by Kumar et al. Large pictures' processing speed is better than small and medium pictures to a certain extent.

(6)Result verification

The image recognition task is mainly through cross-validation to demonstrate the robustness of the model. This paper divides the dataset into ten parts by a 10-fold cross-validation method. Taking one copy at a time as the validation set and the remaining dataset as the training set performed multiple cross-validations. The results show that the dataset does not cause overfitting to the training and validation data. In addition, this paper applies the network model to the VOC2007 dataset for retraining and testing while keeping the parameters changed as little as possible. The mAP of the new YOLOv4 flame image detection algorithm is improved by 0.09% compared to the original YOLOv4 algorithm.

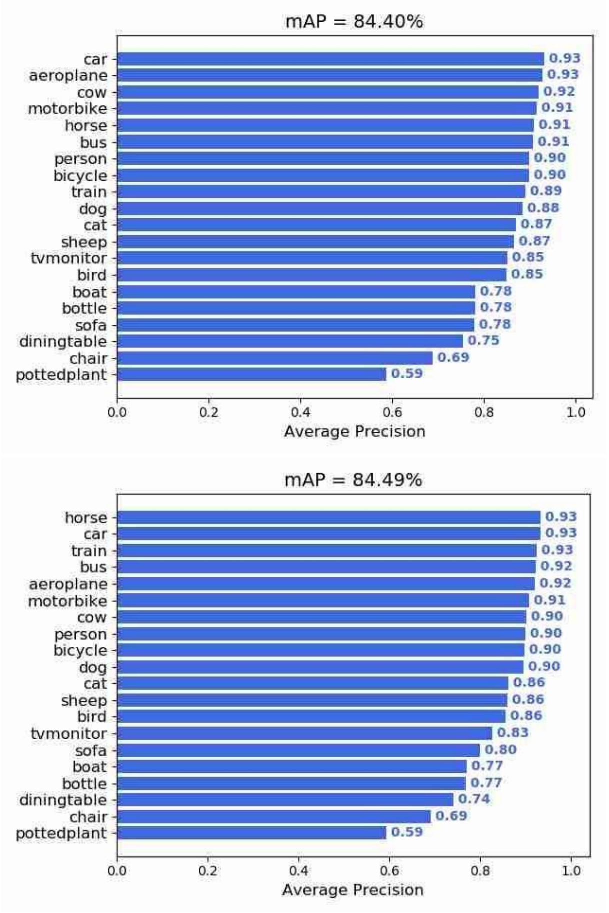


Fig.16 The mAP of two YOLOv4 algorithms

Ⅳ. YOLOv4 FLAME IMAGE DETECTION ALGORITHM BASED ON OPTIMIZED VECTOR CONCAT

In convolutional neural networks, vector concatenation (Concat) is an operation that concatenates feature vectors. After the vector concat operation is completed, a 1x1 convolution for feature fusion is generally added to form a new feature. And then, the feature is continuously updated under the constraints of the loss function to generate more features that meet the requirements. This will take many iterations of the network to produce an appropriate weight. In order to allow the vector concat operation to play a more robust feature fusion role, this chapter proposes an optimization for vector concat.

A. Vector Concatenation Method

(1) The adjustment method of the convolutional layer

Generally speaking, the most direct way to improve network performance is to increase the depth of the network, that is, the number of layers in the network. But this method will bring two deficiencies. One is when the depth and width continue to increase. The parameters that need to be learned also continue to increase. The second is to increase the network size evenly, which will lead to an increase in the amount of calculation. The deep serial convolution structure is generally divided into a ‘convolution-pooling structure’ and a ‘3*3 convolution-1*1 convolution alternating structure’. The two structures are shown in Figure 17.

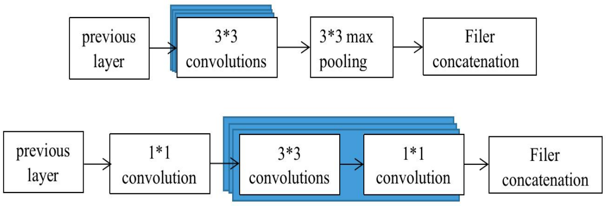


Fig.17 The structure of deep convolutions

In addition to deepening the number of layers of the convolutional neural network, changing the series connection to a parallel connection can also improve the performance of the neural network and the calculation amount is lower. The Inception series of models are mainly to solve the problem of how to increase the depth of the network and reduce the calculation and memory overhead of the model when the classification accuracy of the classification network is improved or maintained. Among them, the module structure of InceptionV1 in the Inception series model is shown in figure.18.

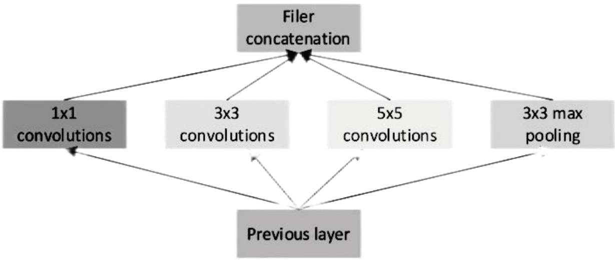


Fig.18 Module structure diagram of InceptionV1

Among them, the previous layer represents the output of the previous layer. Filter concatenation represents the operation of concat the information obtained by convolving the four scale convolution kernels in the previous step along the channel dimension. ‘n*n convolutions’ represents a convolution operation with a kernel size of n. "3*3 max polling" represents the maximum pooling operation with a pooling core 3x3. The network can be widened after using the method of changing series to parallel. On the premise of ensuring the quality of the model, reducing the number of parameters and extracting high-dimensional features solves the problem of network enlargement. The purpose of parallel operation after InceptionV1 is to reduce the number of parameters. Firstly, use 1x1 convolution to reduce the number of channels to gather information. Then, performing feature extraction and pooling at different scales to obtain information at multiple scales. Finally, the features are superimposed and output.

The idea of decomposing the size of the convolution kernel and replacing the large-size convolution kernel with multiple small-size convolution kernels is also a method that can optimize the running time without reducing the accuracy. After Inceptionv2 proposed to convert large convolution kernels into small convolution kernels, convolution kernels other than ‘1*1 convolution kernel’ and ‘3*3 convolution kernel’ gradually withdrew from the stage of history. The commonly used ‘5*5 convolution kernel’ can be replaced by two serially connected ‘3*3 convolution kernels’. The ‘7*7 convolution kernel’ can be replaced by three ‘3*3 convolution kernels’ connected in series. On the one hand, it will not reduce the receptive convolution field. On the other hand, it can speed up the screening of invalid features. By converting a large number of stacked convolutional layers into convolutions with certain parallel connections and changing a single large convolution kernel into multiple serial 3*3 convolutions, the network can extract more valid information.

(2) Applications of residual networks

Intuitively, deeper neural networks perform better with the addition of non-linear activation functions. However, it is found that training a deep feedforward neural network is prone to overfitting and usually encounters the following two problems: gradient dispersion and network degradation. In the face of the above two difficulties, the residual component can add a layer directly connected from head to tail, adding the input of the residual component directly to the sum of the output and then applying the activation function. Experiments show that the residual network solves the degradation problem of deep neural network well. It has achieved good results on image tasks such as ImageNet and CIFAR-10. When the number of layers is equal, the convergence is faster. Feedforward neural networks can be designed to be deeper. Figure 19 is the structural diagram of the residual module.

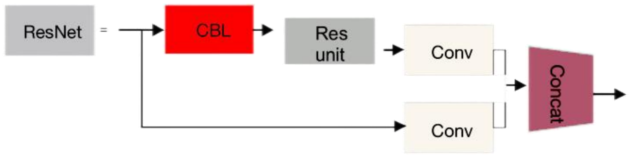


Fig.19 The structure diagram of CSP module

The CBL in the above figure represents a convolution-normalization-Leaky ReLU activation function process. Among them, the Res unit is the residual component. It is necessary to decide how many residual components to use according to different tasks. Among them, standardization can alleviate the negative impact of weight updates on subsequent layers during stochastic gradient descent by smoothing the distribution of hidden layer inputs. Then, keep the input of each neural network layer to maintain the same distribution. The Leaky ReLU activation function sets a slope for the negative interval based on the ReLU activation function, and the slope is not zero. It solves the case where the output and the first derivative of the ReLU activation function are both 0 when the input value of the ReLU activation function is negative. It also avoids the problem that neurons cannot update parameters.

(3) Convolution kernel optimization method

In convolutional neural networks, 1*1 convolution kernels appear in many algorithm models. The 1*1 convolution kernel slides on the input to multiply the input data by a coefficient, which can realize cross-channel information interaction and integration. For a single convolution kernel function, add the convolution results on multiple channels, and then take the activation function value. The output value of the current position can be obtained. Therefore, the information of multiple channels can be integrated without changing the input size when the size of the convolution kernel function is 1*1.

B. Optimizing PANet with an Optimized Vector Concatenation Method

The role of FPN is to fuse high-level and low-level features through upsampling from top to bottom to obtain a predicted feature map, which is used to convey solid semantic features. PAN, conversely, conveys strong positioning features by fusing features from bottom to top through downsampling. Adding bottom-up PAN after FPN can convey strong semantic and positioning features, promoting the exchange of information within the network. According to Inception, widening the network structure can effectively reduce the number of parameters and enhance the ability to extract high-dimensional features while ensuring the quality of the model. So this article uses the parallel method to obtain a structure that works on network optimization. The optimized structure is shown in figure 20.

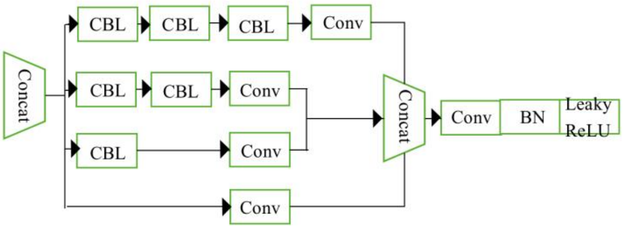


Fig.20 The optimized Concat module A’s structure diagram

In figure 20, the concat on the left represents the method of concat information of two different scales along the channel dimension by downsampling at a large scale or upsampling at a small scale. CBL represents the structure of a 3*3 volume Warp-Normalized-Leaky ReLU activation function. Concat on the right represents concat the four convolution results of the previous step along the channel dimension. Conv represents a 1*1 convolution with a stride of 1 and padding of 0. The role of the two CBL structures is to replace a 5*5 convolution. The role of the three CBL structures is to replace a 7*7 convolution.

However, replacing the Concat structure in PANet with this structure will lead to a sharp increase in the number of parameters and drag down the detection time of the network, making it challenging to implement the application. By adjusting and choosing different channels, this paper obtains a set of vector concat structures that consume less computing time and can improve accuracy, as shown in figure 21.

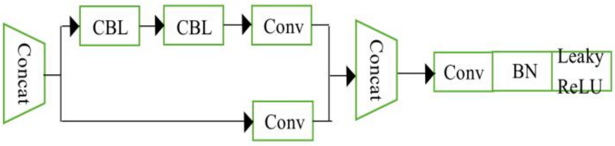


Fig.21 The optimized Concat module B’s structure diagram

Combining this structure with PANet, the optimized structure of PANet is shown in figure 22. It can be seen from the figure that adding the simplified vector concat structure to PANet does not change the input and output of the steps.

As can be seen from Figure 22, the structure first converts the '13*13*512' feature image acquired by the backbone network into a '26*26*256' feature image through upsampling. When the length and width are doubled, the number of information channels is reduced to half. It saves itself for subsequent operations. Next, the '26*26*256' feature image obtained by upsampling and the '26*26*256' feature image obtained by the backbone network through 1*1 convolution are optimized for vector concat to obtain the

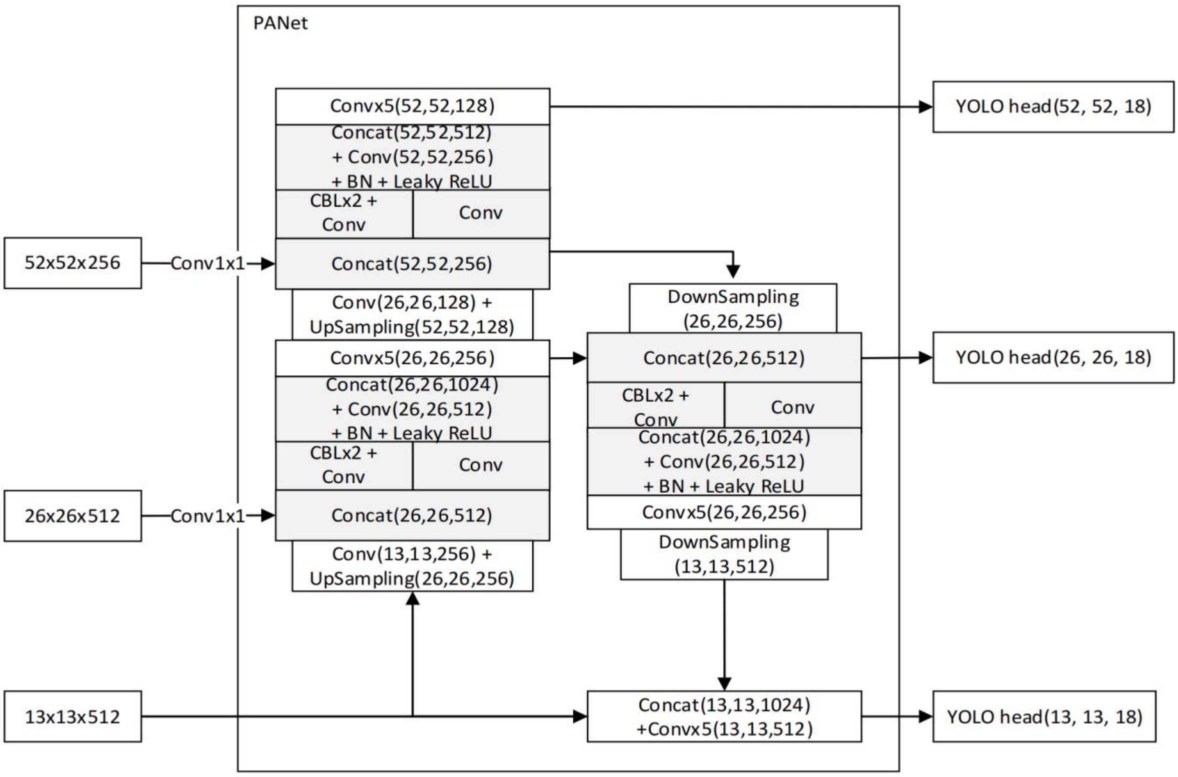


Fig.22 The optimized PANet structure

'26*26*512' feature image. Subsequently, the '26*26*512' feature image obtained in the previous step is converted into a '52*52*128' feature image through 5 convolutions and one upsampling, and the '52*52*128' feature image for optimized vector mosaic. It is saved after five convolutions for subsequent operations. Finally, the result of the vector concat in the previous step is convolved five times to obtain the feature image required for detecting large-scale images. The eatured image required to detect intermediate-scale images is obtained by downsampling the vector concat result and optimizing the concat with the five convolution results saved in the previous step. Downsample the feature image required to detect intermediate-scale images and splice with the feature image saved "13*13*512" and perform five convolutions to obtain the feature image required to detect small-scale images.

C. Experiment and Result Analysis

This paper compares the original YOLOv4 algorithm, the YOLOv4 algorithm based on attention mechanism optimization, the YOLOv4 algorithm based on optimized vector concat, and the YOLOv4 algorithm optimized by both attention mechanism and optimized vector concat. The results show that the AP of the original YOLOv4 algorithm is 95.12%. The AP of the YOLOv4 algorithm optimized based on the attention mechanism is 95.88%. The AP of the YOLOv4 algorithm based on optimized vector concat is 95.62%. At the same time, the AP of the YOLOv4 algorithm optimized by the attention mechanism and optimized vector concat is 95.74%. The specific experimental data are shown in Table 4.

Compared with the original YOLOv4 flame image detection algorithm, the AP of the YOLOv4 flame image detection algorithm based on optimized vector concat has increased by 0.5%, and the recall rate has also increased by 0.44%, but the precision rate has decreased by 0.3%.

(1) Comparison of accuracy between the original YOLOv4 algorithm and the YOLOv4 algorithm based on optimized vector concat.

Figure 23 shows the image comparison between the original YOLOv4 flame image detection algorithm and the YOLOv4 flame image detection algorithm based on an optimized vector mosaic. The precision changes with the score threshold.

TABLE Ⅳ
THE COMPARISON OF VARIOUS OPTIMIZATION ALGORITHMS OF YOLOV4

Algorithm	AP	F1-score	Precision	Recall	Miss rate
YOLOv4	95.12%	0.94	94.76%	93.43%	0.07%
YOLOv4 with PSA head	95.88%	0.95	96.15%	93.90%	0.06%
YOLOv4 optimized Concat	95.62%	0.94	94.34%	93.90%	0.06%
YOLOv4 optimized all	95.74%	0.95	95.71%	94.37%	0.06%

The score threshold is a decimal between 0 and 1, representing the accuracy with which all detections below that threshold are counted as flames. At the same time, the accuracy rate under all score thresholds is connected into a curve, which can see the accuracy rate change with the score threshold change. Where "class: 94.76% = fire Precision; score_threshold = 0.5" means that the accuracy rate of individuals classified as fire is 94.76% when the confidence threshold is 0.5. "class: 94.34% = fire Precision; score_threshold = 0.5" means that the accuracy of classifying individuals as fire is 94.34%, when the confidence threshold is 0.5. As the confidence threshold increases, the accuracy of the YOLOv4 algorithm continues to improve overall. The stricter the flame discrimination requirements, the more likely to find areas where flames exist.

(2) Comparison of the recall rate between the original YOLOv4 algorithm and the YOLOv4 algorithm based on optimized vector concat

Figure 24 shows the image comparison of the original YOLOv4 algorithm and the YOLOv4 algorithm optimized based on the vector concat algorithm. "class: 93.43% = fire Recall; score_threshold = 0.5" represents a recall rate of 93.43% for individuals classified as fire when the confidence threshold is 0.5. "class: 93.90% = fire Recall; score_threshold = 0.5" represents a recall of 93.90% of individuals classified as fire when the confidence threshold is 0.5.The recall rate changes with the increase of the score threshold. It can be seen that as the confidence threshold increases, the recall rate of the YOLOv4 algorithm continues to decrease as a whole. If there is a flame in a certain area of an image, the stricter the flame discrimination requirements are, the more likely it is impossible to detect the presence of a flame in this area. When the confidence threshold is set to 0.5, the recall rates are 93.43% and 93.90%, respectively. When there is a flame individual in a certain area of an image, if the confidence level is higher than 50%, the individual is judged as a flame and the optimized YOLOv4 algorithm can detect more flames.

(3)Comparison of original YOLOv4 algorithm and YOLOv4 algorithm AP based on optimized vector concat

Figure 25 shows the relationship between precision and recall. Connect the precision rate values corresponding to each recall rate with a curve. The area of the curve enclosed by the curve and the positive part of the coordinate axis is divided by 1 as the comprehensive evaluation index (AP). It can be seen from the figure that the APs of the two algorithms are 95.12% and 95.62%, respectively. "class: 95.12% = fire AP" means that the area of the part divided by 1 is 0.9512. "class: 95.62% = fire AP" means that the area of the part divided by 1 is 0.9562.

Combining the above pictures, the recall rate and AP of optimizing YOLOv4 have improved, and the precision rate has decreased. Optimizing YOLOv4 has a lower miss rate when the number of predicted flames is equal. The algorithm can misjudge fewer flame-containing scenes as non-flame scenes, but there may also be missed detections.

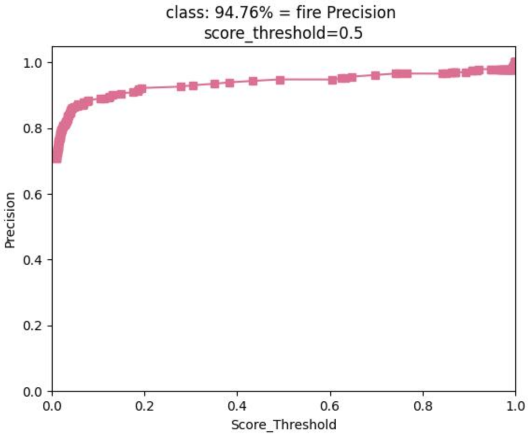


Fig.23 The precision curve of two YOLOv4 algorithms (1)

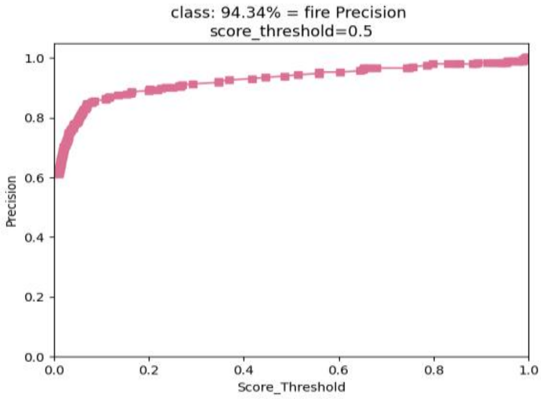


Fig.23 The precision curve of two YOLOv4 algorithms (2)

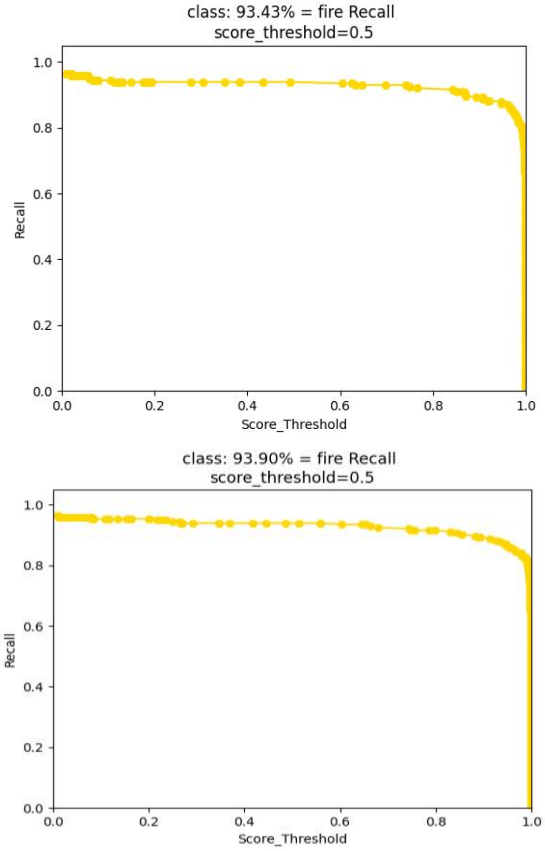


Fig.24 The recall curve of two YOLOv4 algorithms

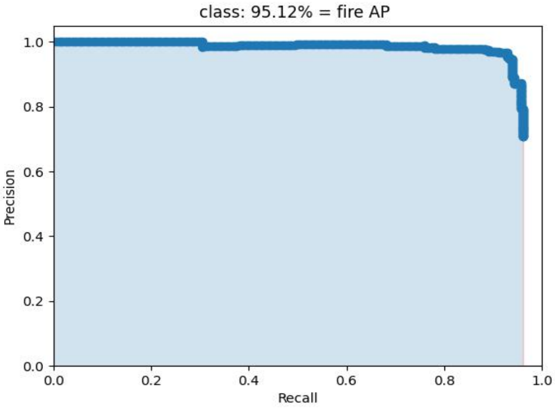


Fig.25 The AP curve of two YOLOv4 algorithms (1)

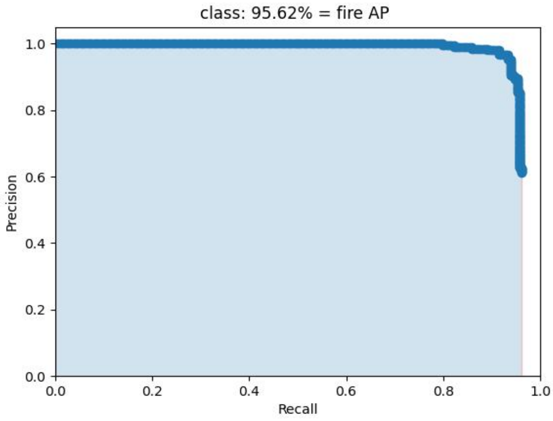


Fig.25 The AP curve of two YOLOv4 algorithms (2)

D. Analysis of Ablation Experiments

Figure26 show the comparison between the original YOLOv4 and the YOLOv4 using two optimization methods at the same time in terms of precision, recall and AP.

(1) Comparison of accuracy between original YOLOv4 and YOLOv4 using two optimization methods

Figure 26 shows a set of images comparing the accuracy of the original YOLOv4 algorithm with the two optimized YOLOv4 algorithms as the score threshold increases. The score threshold is a decimal between 0 and 1, which represents the accuracy rate when all detection results less than the threshold are counted as flames. It can be clearly seen that the change of precision rate as the score threshold changes when connecting the precision rate under all score thresholds into a curve. "class: 94.76% = fire Precision; score_threshold = 0.5" means that when the confidence threshold is 0.5, the accuracy rate of individuals classified as fire is 94.76%. "class: 95.71% = fire Precision; score_threshold = 0.5" means that when the confidence The accuracy of classifying individuals as flames at a threshold of 0.5 was 95.71%.

As the confidence threshold increases, the accuracy of the YOLOv4 algorithm continues to improve overall. When the confidence threshold is set to 0.5, the accuracy rates of the two are 94.76% and 95.71%, respectively. When the algorithm judges that the probability of flame individuals in a certain area of an image is greater than 50%, the area is judged to have flames. In this case, the algorithm proposed in this paper will misjudge normal scenes as flames less than 0.95 times per 100 frames.

(2) Comparison of recall rate between original YOLOv4 and YOLOv4 using two optimization methods

Figure 26 shows images comparing the recall of the original YOLOv4 algorithm and the optimized YOLOv4 algorithm using both optimization algorithms as the score threshold increases. It can be clearly seen that the change of recall rate as the score threshold changes when connecting the recall rate under all score thresholds into a curve. "class: 93.43% = fire Recall; score_threshold = 0.5" represents a recall rate of 93.43% for individuals classified as fire when the confidence threshold is 0.5.

As the confidence threshold increases, the recall of the YOLOv4 algorithm decreases overall. When the confidence threshold is set to 0.5, the recall rates of the two are 93.43% and 94.37%, respectively. If the confidence level is higher than 50%, the individual is judged as a flame when there is a flame individual in a certain area of an image. In this case, the algorithm proposed in this paper will correctly identify 0.94 more times per 100 frames of pictures on average.

(3) Comparison of AP between original YOLOv4 and YOLOv4 using two optimization methods

As the recall rate continues to increase, it can be inferred that the confidence threshold at this time continues to decrease. The precision rate corresponding to the confidence threshold should continue to decrease. The figure above shows the correspondence between precision and recall. Connect the precision rate value corresponding to each recall rate with a curve and divide the area of the curve enclosed by the curve and the positive part of the coordinate axis by one as the comprehensive evaluation index (AP). It can be seen from the figure that the APs of the two algorithms are 95.12% and 95.74%, respectively. Where "class: 95.12% = fire AP" means that the result of dividing the area of the part by 1 is 0.9512.

(4) Comparison of various evaluation indicators in ablation experiments

The PSA head in the table uses the PSA head proposed in this paper as the optimization method of YOLOv4. The Concat in the table uses the optimized vector concat method proposed in this paper as the optimization method of YOLOv4. The evaluation index of the upward arrow represents that the higher the value of the evaluation index, the better the algorithm's performance to achieve this effect. The number below the evaluation index is the value of the optimization algorithm compared to the original algorithm. The evaluation index of the downward arrow represents that the lower the value of the evaluation index, the better the algorithm's performance. The number below the evaluation index is the value of the optimization algorithm compared to the original algorithm.

TABLE V
ABLATION EXPERIMENT OF VARIOUS OPTIMIZATION ALGORITHMS OF YOLOV4

PSA head	Concat	AP ↑	Precision ↑	Recall ↑	Miss Rate ↓
√		0.76%	1.39%	0.47%	0.01%
	√	0.50%	-0.42%	0.47%	0.01%
√	√	0.62%	0.95%	0.94%	0.01%

It can be seen from Table V that both the PSA head and the optimized vector concat method can improve the AP. The PSA head can improve the precision rate even more. Combining the optimized vector concat method and the PSA head can improve the recall rate. Higher. Therefore, if you pay more attention to precision and detection time in flame image detection, you can use the flame image detection algorithm optimized by the PSA head. If you pay more attention to the recall rate, you can use the flame image detection algorithm that combines the two optimization methods.

To demonstrate the robustness of the model, this paper also takes into account the computational resource consumption of using the PSA head in conjunction with an optimized vector concatenation method. At this stage of this paper,

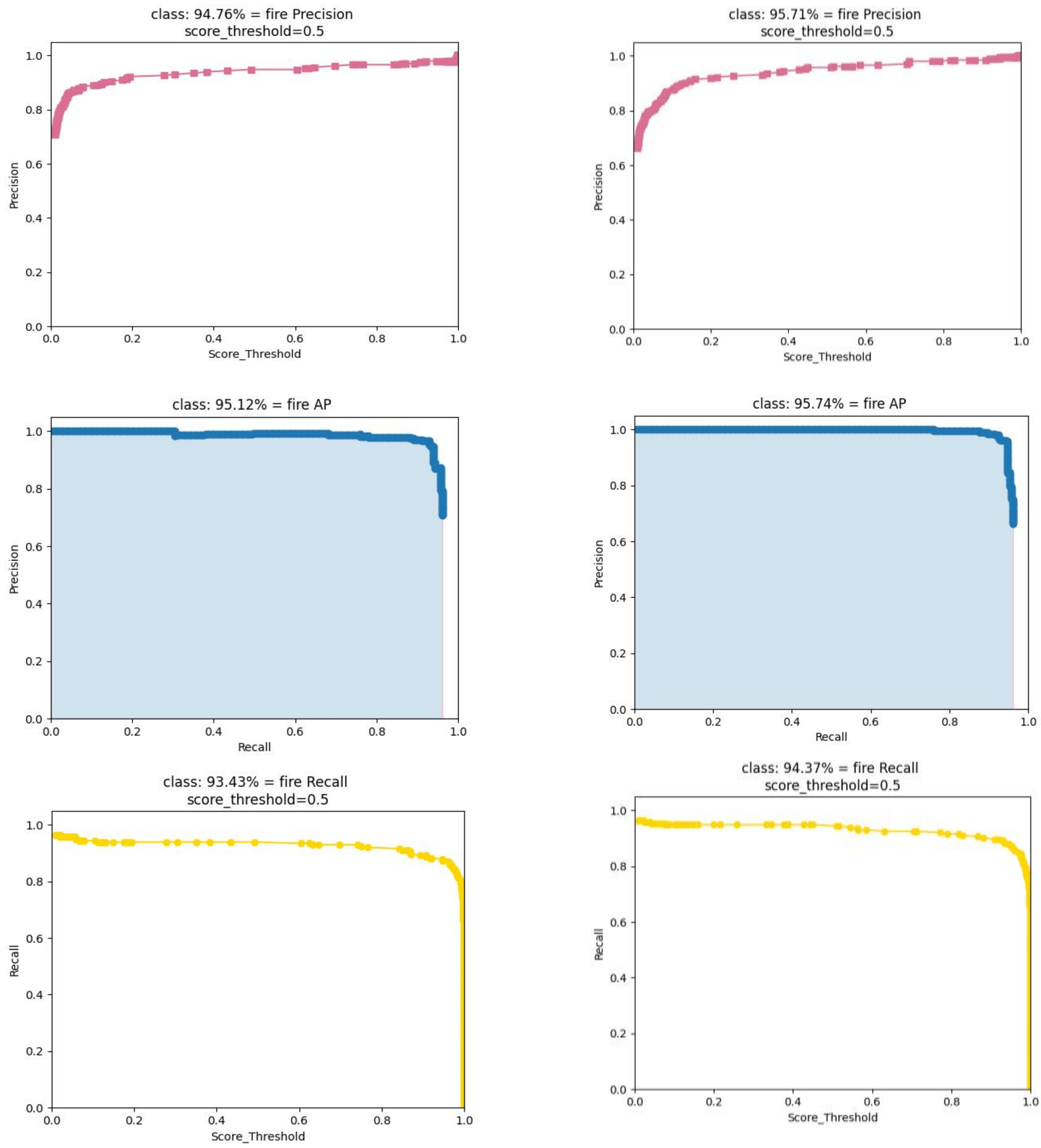


Fig.26 The precision, recall and AP curve of two YOLOv4 algorithms

cross-validation is used to verify the results. The data set is divided into ten parts on average through the method of 10-fold cross-validation, and one part is taken as the verification set at a time. The rest of the data set is used as the method of the training set for cross-validation. The results show that using different dataset partitions has little effect on the model.

Ⅴ. Conclusion

This paper proposes two optimization methods for YOLOv4. The accuracy evaluation indicators of the optimized YOLOv4 algorithm have been improved to a certain extent. However, further improvements are needed in the face of complex application environments.

References

[1] Hashemzadeh M, Zademehdi A. Fire detection for video surveillance applications using ICA Kmedoids-based color model and efficient spatio-temporal visual features[J]. Expert Systems with Applications, 2019, 130: 60-78.

[2] Qian Z, Xiaojun L, Lei H. Video image fire recognition based on color space and moving object detection[C]//2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), Beijing, China, 2020: 367-371.

[3] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in Neural Information Processing Systems, 2012, 25: 1097-1105.

[4] Muhammad K, Ahmad J, Mehmood I, et al. Convolutional neural networks based fire detection in surveillance videos[J]. IEEE Access, 2018, 6: 18174-18183.

[5] Gaur A, Singh A, Kumar A, et al. Video flame and smoke based fire detection algorithms: A literature review[J]. Fire Technology, 2020, 56(5): 1943-1980.

[6] Avula S B, Badri S J, Reddy G. A Novel forest fire detection system using fuzzy entropy optimized thresholding and STN-based CNN[C]//2020 International Conference on Communication Systems & Networks (COMSNETS), Paris, France, 2020: 750-755.

[7] Shahid M, Hua K. Fire detection using transformer network[C]//2021 Proceedings of the 2021 International Conference on Multimedia Retrieval. 2021: 627-630.

[8] Dutta S, Ghosh S. Forest Fire detection using combined architecture of separable convolution and image processing[C]//2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), Riyadh, Saudi Arabia, 2021: 36-41.

[9] Yang Z, Wang T, Bu L, et al. Training with augmented data: GAN-based flame-burning image synthesis for fire segmentation in warehouse[J]. Fire Technology, 2022, 58(1): 183-215.

[10] Rahmatov N, Paul A, Saeed F, et al. Realtime fire detection using CNN and search space navigation[J]. Journal of Real-Time Image Processing, 2021, 18(4): 1331-1340.

[11] Jiang B, Luo R, Mao J, et al. Acquisition of localization confidence for accurate object detection[C]//2018 Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 2018: 784-799.

[12] Rezatofighi H, Tsoi N, Gwak J Y, et al. Generalized intersection over union: A metric and a loss for bounding box regression[C]//2019 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, America, 2019: 658-666.

[13] Zheng Z, Wang P, Liu W, et al. Distance-IoU loss: Faster and better learning for bounding box regression[C]//2020 Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(07): 12993-13000.

[14] Zheng Z, Wang P, Ren D, et al. Enhancing geometric factors in model learning and inference for object detection and instance segmentation[J]. IEEE Transactions on Cybernetics, 2021:1-13.

[15] Neubeck A, Van Gool L. Efficient non-maximum suppression[C]//2006 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 2006, 3: 850-855.

[16] Padilla R, Netto S L, Da Silva E A B. A survey on performance metrics for object-detection algorithms[C]//2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niteroi, Brazil, 2020: 237-242.

[17] Buslaev A, Iglovikov V I, Khvedchenya E, et al. Albumentations: fast and flexible image augmentations[J]. Information, 2020, 11(2): 125-126.

[18] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//2017 Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 2017: 2980-2988.

[19] Sinaga K P, Yang M S. Unsupervised K-means clustering algorithm[J]. IEEE Access, 2020, 8: 80716-80727.

[20] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//2014 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, America, 2014: 580-587.

[21] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//2016 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, America, 2016: 779-788.

[22] Benyang, D., Xiaochun, L. and Miao, Y., 2020, November. Safety helmet detection method based on YOLO v4. In 2020 16th International conference on computational intelligence and security (CIS) . IEEE Access, 2020,pp. 155-158.

[23] Gai R, Chen N, Yuan H. A detection algorithm for cherry fruits based on the improved YOLO-v4 model. Neural Computing and Applications. 2021 May 26:1-2.