Journal of Computer and Communications, 2022, 10, 1-14
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Forest Fire Smoke Detection Method Based on MoAm-YOLOv4 Algorithm

Yihong Zhang, Qin Lin, Changshuai Qin, Hang Ge

School of Information Science and Technology, Donghua University, Shanghai, China
Email: 2211904@mail.dhu.edu.cn

How to cite this paper: Zhang, Y.H., Lin, Q., Qin, C.S. and Ge, H. (2022) Forest Fire Smoke Detection Method Based on MoAm-YOLOv4 Algorithm. *Journal of Computer and Communications*, **10**, 1-14.
https://doi.org/10.4236/jcc.2022.1011001

Received: October 10, 2022
Accepted: October 29, 2022
Published: November 1, 2022

Abstract

To improve the performance of the forest fire smoke detection model and achieve a better balance between detection accuracy and speed, an improved YOLOv4 detection model (MoAm-YOLOv4) that combines a lightweight network and attention mechanism was proposed. Based on the YOLOv4 algorithm, the backbone network CSPDarknet53 was replaced with a lightweight network MobilenetV1 to reduce the model's size. An attention mechanism was added to the three channels before the output to increase its ability to extract forest fire smoke effectively. The algorithm used the K-means clustering algorithm to cluster the smoke dataset, and obtained candidate frames that were close to the smoke images; the dataset was expanded to 2000 images by the random flip expansion method to avoid overfitting in training. The experimental results show that the improved YOLOv4 algorithm has excellent detection effect. Its mAP can reach 93.45%, precision can get 93.28%, and the model size is only 45.58 MB. Compared with YOLOv4 algorithm, MoAm-YOLOv4 improves the accuracy by 1.3% and reduces the model size by 80% while sacrificing only 0.27% mAP, showing reasonable practicability.

Keywords

Forest Fire Smoke Detection, Pattern Recognition and Intelligent Systems, YOLOv4, Channel Attention Mechanism

1. Introduction

Fire is one of the disasters that cause the most significant harm to forest resources. Due to its sudden solid and destructive characteristics, human beings have always wanted to avoid forest fires. Smoke is generally generated before flames, and due to the substantial expansion of smoke, it is easier to be captured by cameras. So, detecting smoke is vital in preventing forest fires [1].

Forest fire detection can be divided into two main types: traditional detection method and deep learning-based detection method. Early humans used traditional methods, including manual patrol, watchtower detection, satellite remote sensing and aviation patrol, etc. These methods have many shortcomings, such as manual patrol and watchtower detection being inefficient and having high labour costs, satellite remote sensing being too dependent on the performance of hardware equipment, and the detection cost being too high. This has led to increasing efforts into deep learning-based forest fire detection research [2] [3] [4] [5].

In recent years, deep learning has developed rapidly, and using object detection algorithms for smoke detection has become mainstream [6] [7] [8] [9]. The main detection alogorithm includes R-CNN [10], Faster R-CNN [11], Mask R-CNN [12], and YOLO [13], Fire-Net [14] etc. Given the above problems, Li Hongchang *et al.* detected the forest fire smoke through the method of always having boundary variations, which combines the idea of block smooth analysis and feature fusion and clustering treatment to obtain the suspected smoke area, only focusing on the movement characteristics of the smoke, so avoids the error caused by the tedious calculation. Considering the time spent training the traditional convolutional neural network algorithm, Zhang *et al.* [15] expanded the dataset by synthesising forest fire smoke and detected it with the Faster R-CNN algorithm, eliminating the complex process of manually extracting features in traditional video detection methods. Sun *et al.* [16] proposed an improved convolutional neural network (CNN) to achieve rapid fire smoke detection. In order to improve the robustness of CNN and avoid overfitting, they applied an optimization strategy to improve the loss function in the multiple convolutional kernel and batch normalization. Mohnish *et al.* [17] proposed a forest fire detection model based deep learning, which improved the model's accuracy to 93% and 92% on training and testing datasets, respectively. Jin Y *et al.* [18] used an anchor-free network structure to realise the fire detection model chose Mobilenetv2 as the backbone network for feature extraction and added an FSAF feature selection module to select the appropriate feature incoming prediction layer. The loss and the smallest feature layer return gradient were selected during training, and the feature layer with the highest target confidence was selected during prediction. Zhang *et al.* [19] realised that fire detection based on the Faster R-CNN improved the feature extraction process, introducing the FPN feature fusion network to integrate shallow and high-level features. Qian *et al.* [20] used the channel pruning method to lightweight YOLOv3. They used fire detection to judge the importance of the channel according to the coefficient of the BN layer weights, remove the unimportant channels, and reduce convolution parameters. Xu *et al.* [21] pointed out that it is difficult for a single network model to achieve feature extraction in multi-complex scenarios, and every single network can extract different features. Therefore, the model-improved feature extraction process integrating three deep networks is designed for forest fire detection.

Considering the requirement of real-time for forest fire smoke detection, propose an improved YOLOv4 [22] forest fire smoke detection algorithm, replace the backbone network with the lightweight network MobilenetV1 [23] to reduce the size of the model and improve the detection speed; use K-means clustering algorithm [24] to obtain more suitable candidate boxes; embed the channel attention mechanism before the output to improve the network to extract feature from forest fire smoke while increasing the model size as little as possible. The experimental results show that the improved algorithm can accurately detect forest fire smoke in real-time.

2. The Algorithm in This Paper

2.1. The YOLOv4 Network Structure

As the fourth version of the YOLO series, YOLOv4 is mainly improved by YOLOv3 [25]. Its network structure can be divided into four modules: Input, Backbone, Neck, and Head. The input side is the input image, this stage contains an image preprocessing stage; the Backbone is the backbone network for feature extraction, here the backbone network is the CSPDarknet53; the Neck is located between the Backbone and the Head, which contains two parts: the SPP additional module and the PANet. After the SPP reaches the last output layer of the backbone network, the maximum pooling operation of the four different convolutional kernel sizes is 1×1 (*i.e.*, no processing), 5×5, 9×9, and 13×13. The introduction of this additional module can help separate the most important context features without reducing the running speed of the model. PANet is used for feature fusion, combining the elements of shallow and deep networks through up and down sampling; Head output completes the final detection results, which is the unique module of YOLO algorithm compared with other target detection algorithms. Its network structure is shown as described in **Figure 1**.

2.2. Candidate Box Improvement

The candidate box size of the appropriate size is beneficial for the regression of the fire smoke prediction box. The experiment uses the K-mean clustering method to generate the candidate boxes, through which the size of the candidate box can be made closer to the size of the dataset image.

The image size of the network input is 416×416, and the nine candidate boxes generated by K-means are: 33×141, 45×164, 58×104, 66×165, 97×168, 129×37, 165×196, 177×76, 250×310.

2.3. Improvement of Lightweight Network Structure

The Backbone of the detection algorithm no longer uses CPSDarknet53 and instead replaces it with a lightweight network MobilenetV1. Google proposed mobilenetV1 in 2017, and its model has a small size and a small number of parameters, which is very suitable for application to mobile devices. The core idea of the

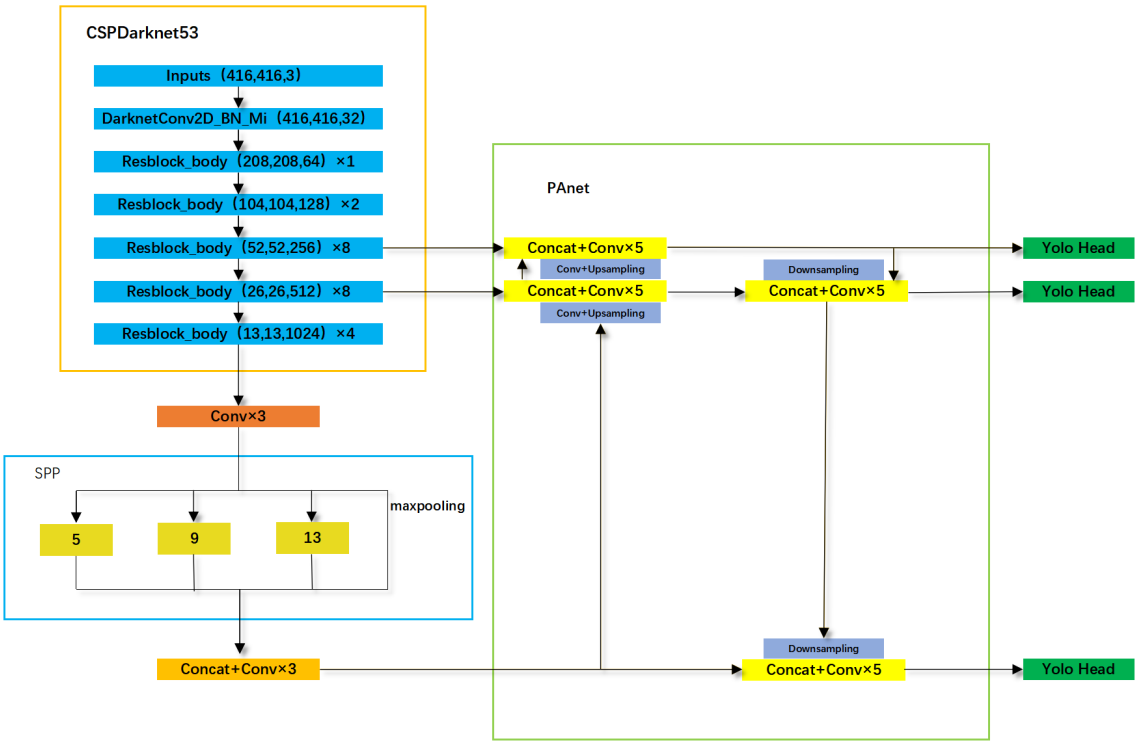

Figure 1. YOLOv4 network structure diagram.

MobilenetV1 network is deeply separable for convolution, It also uses the width and resolution multiplier to reduce the number of parameters, allowing for better data throughput at minimal precision.

2.3.1. Depth-Separable Convolution

Deep separable convolution is a variant of ordinary convolution operations that can replace different convolutions to form convolutional neural networks [26]. It includes depthwise and pointwise convolution, which combine these two convolution methods to complete the image feature extraction. Depth-separable convolution dramatically reduces the number of parameters, so that the network can also meet the requirements of real-time detection on the CPU.

Assuming that the input image size is $D_F \times D_F \times M$, the size of the convolution kernel is $D_K \times D_K$, The size of the output feature map is marked as $D_F \times D_F \times N$, where the M is the number of input layer channels and the N is the number of output layer channels are assumed, here the width and height of the input and output feature maps are consistent.

For traditional convolution operations, the parameter calculation amount is:

$$D_K \times D_K \times M \times N \times D_F \times D_F \tag{1}$$

For depthwise convolution operations, the parameter calculation amount is:

$$D_K \times D_K \times M \times D_F \times D_F \tag{2}$$

For the pointwise convolution operation, the parameter calculation amount is:

$$M \times N \times D_F \times D_F \tag{3}$$

For depth-separable convolution operations, the parameter calculation amount is:

$$D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F \tag{4}$$

Dividing between Equation (4) and Equation (1) gives us that:

$$\frac{D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_K \times D_K \times M \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_K^2} \tag{5}$$

As can be seen from the above Equation (5), when the number of output channels N is large, if the convolution kernel size used is 3 × 3, then the deep separable convolution operation can reduce the parameter calculation by nearly 8 - 9 times compared with the traditional convolution operation, signifficantly reducing the calculation amount.

2.3.2. Width Multiply

To construct models that can satisfy lower computational costs, a parameter α in MobilenetV1 is introduced, called Width Multiplier.

The function of α is to refine the network on each layer, and for a given network layer, the number of input channels M becomes αM, and the number of output channels N becomes αN. After adding hyperparameters of α, the computation of depth separable convolution becomes:

$$D_K \times D_K \times \alpha M \times D_F \times D_F + \alpha M \times \alpha N \times D_F \times D_F \tag{6}$$

The values of α in Equation (6) range from (0, 1], and generally from 1, 0.75, 0.5, or 0.25. If all the layers in the network multiply by α, the size of the network model drops close to α^2 times the original network, and the computational amount drops to α^2 times the original size.

2.3.3. Resolution Multiplier

Another parameter introduced by MobilenetV1 is β which called Resolution Multiplier, which is used to control the resolution of the input image. If the input image size is 416 × 416, and the hyperparameter $\beta = 0.5$, the input image size will be doubled to 208 × 208. Both hyperparameters α and β are added to the MobilenetV1 network, and the parameter quantity of the deeply separable convolution operation becomes:

$$D_K \times D_K \times \alpha M \times \alpha D_F \times \beta D_F + \alpha M \times \alpha N \times \beta D_F \times \beta D_F \tag{7}$$

The range of β in the formula is (0, 1]. Suppose the image resolution of the input layer multiply the hyperparameter β. In this case, the size of the network model will not change, which is still the same size as the original network model, but the calculation amount of the model will decrease to β^2 times that of the original model. It is not difficult to see that the one of the two hyperparameters is to compress the network model, and the other is to reduce the computation of the model. After adding two parameters simultaneously, the purpose of shortening the model and a meagre count can be achieved.

2.4. Head Output Improvement

2.4.1. Channel Attention

During convolution, the interference information in the channel prevents the network from focusing on the information most important to the task, leading to reduced performance. Currently, the attention mechanism is widely used in convolutional neural networks. Different types of channel attention mechanism can reduce the influence of interference information to varying degrees after adjusting for the appropriate weights. The SENet (Squeeze and Excitation Networks) is the commonly used channel attention mechanism. The schematic diagram is shown in **Figure 2**.

F_{tr} is a traditional convolutional structure in **Figure 2**, X and U are the input and output respectively of F_{tr}, the size is $C'\times W'\times H'$ and $C\times W\times H$, that is, U has C' channel, each channel size is $H'\times W'$, U has C channel, each channel size is $H\times W$. The whole process can be divided into three steps. The first step is called the Squeeze process, the global average pooling of U, which is the process of $F_{sq}(.)$ in **Figure 2**. The output channel weights are as follows:

$$Z = F_{sq}(u_c) = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} u_c(i,j) \tag{8}$$

where Z is the number of the size of $1\times 1\times C$, (i,j) is the point of the corresponding coordinates on the feature graph of the size; the second step is called the Excitation process, which is two fully connected operations, that is $F_{ex}(.,w)$ in **Figure 2**, the output channel weight formula is:

$$S_c = F_{ex}(Z,W) = Sigmoid\left(W_2 \times ReLU\left(W_1, Z\right)\right) \tag{9}$$

S_c is the generated channel weight, the dimension is $1\times 1\times C$, the dimension of W_1 is $C/r\times C$, the dimension of W_2 is $C\times C/r$, r is a zoom parameter; the last step is called Reweight operation, corresponding to $F_{scale}(.,.)$ in **Figure 2**, the corresponding input features of the channel output are weighted by multiplication to reset the input features, thus making the extracted features more directional. The formula is:

$$\hat{X} = F_{scale}(X_c, S_c)\otimes S_c \tag{10}$$

$\hat{X}$ is the output result after SENet processing, the symbol "$\otimes$" indicates element-by-element multiplication.

The SENet can be added to any convolutional layer, and in general, the more times the SENet is inserted, the more parameters it will bring to the network.

2.4.2. Add to the Attention Mechanism

Embedding the SENet attention mechanism in front of the YOLO Head output can educe the impact of interference information on the network's performance and will not bring too many parameters to the network. Therefore, the model only increases by 0.33MB compared to the original network.

The schematic diagram is shown in **Figure 3**, the SENet attention mechanism module is added to the three output channels of PANet, and the scaling parameter

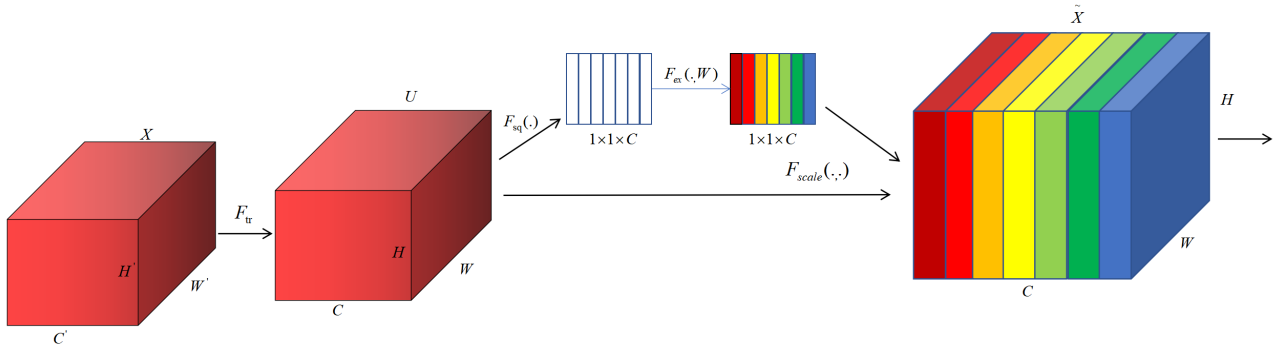

Figure 2. SENet schematic.

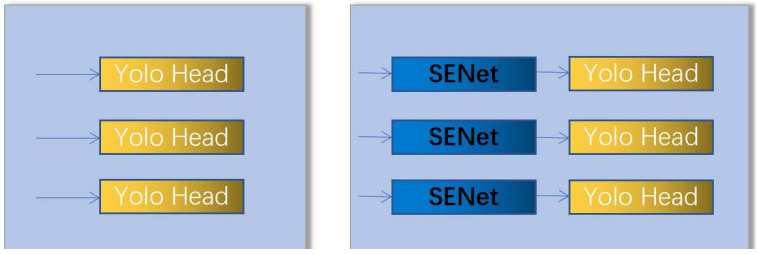

Figure 3. Comparison of algorithm before and after adding attention mechanism. Left: Before joining; Right: After joining.

r is generally taken 2, 4, 8, 16, in this paper, all four parameters are tested, and finally 8 takes the best effect, so we set $r = 8$.

Integrating the above innovation points, the final improved MoAm-YOLOv4 network structure diagram is shown in **Figure 4**.

3. Experimental Design

3.1. Experimental Environment

The experiment used a GPU of RTX2080ti, a 64-bit operating system of Win10, a deep learning framework of Pytorch, and Python programming language.

Training consists of two stages: the freezing stage and the thawing stage. The first stage is coarse tuning, which needs to freeze the backbone network and mainly used to adjust the parameters of the non-backbone network. Through multiple debugging, the learning rate is set to 0.001, the batch size is set to 16, and the 50 epoch are trained. The second stage is to release the backbone network parameters, which belong to the fine-tuning stage, setting the maximum learning rate to 0.00005, batch size to 4, and training 50 epoch.

3.2. The Forest Fire Smoke Dataset

Part of the dataset used in the experiment came from publicly available online data such as the Key Laboratory of Fire of the University of Chinese Academy of Sciences, and part came from online crawlers and were downloaded from kaggle. There are two types of pictures, part is real forest fire smoke, part is the synthetic data of "RF_dataset" and "SF_dataset" published by Zhang *et al.*, where "RF_dataset"

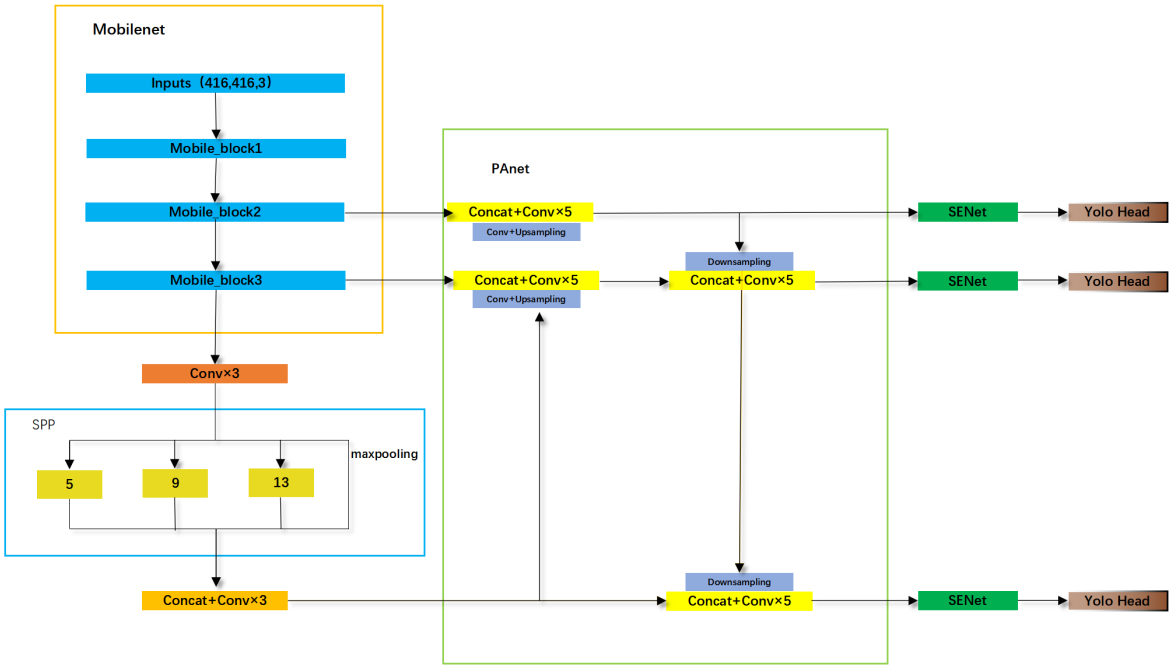

Figure 4. MoAm-YOLOv4 network structure diagram.

is to synthesise real smoke in a forest background, and "SF_dataset" is to synthesize virtual smoke in a forest background. To expand the data set, the data is doubled by: 1/3 of the picture is flipped left, 1/3 of the picture is flipped down, and 1/3 of the picture is flipped right. Images were annotated with the LabelImg tool, yielding a total of 2000 forest smoke images. Pictures of some forest smoke are shown in **Figure 5**.

4. Experimental Results and Analysis

4.1. Evaluation Indicators

The index parameters selected for the evaluation include the Precision, the recall rate (Recall), the mean average precision (mAP), the number of detected frames per second (FPS), and the model size. The calculation formulas of Precision and Recall are as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

In Equation (11) and Equation (12), TP is a positive sample predicted as the positive class by the model, TN is a negative sample predicted negative class, FP is a negative sample predicted positive class, and FN is a positive sample predicted negative class. But both Precision and Recall consider only one factor and cannot comprehensively measure the performance of a model. The mAP is the average of the accuracy of each category, which is more representative than the above two parameters.

Figure 5. Part of the forest fire smoke images.

4.2. Experimental Result

To verify the effectiveness of the algorithm, a total of five sets of comparative experiments were done on the collected forest fire smoke dataset, including YOLOv4, YOLOv4-tiny, replacing only the network with MobilenetV1, embedding the SENet attention mechanism only in front of the Head output, and the combination of the two. Table 1 shows the comparison diagram of the experimental results.

As can be seen from Table 1 above, after replacing the feature network with the lightweight network MobilenetV1, the mAP of the model decreased by nearly one percentage point, but the model size decreased from 245.52 MB to 48.42 MB, and the detection speed of FPS was also greatly improved; after embedding the SENet attention mechanism, the model size was increased by only 0.33MB, but the mAP increased by 0.82%.

YOLOv4-tiny is a lightweight architecture of YOLOv4. In addition to streamlining the backbone network, it also reduces the output prediction from 3 to 2, reduces the candidate boxes from 9 to 4, and uses the traditional FPN instead of PANet for feature fusion. The operation of these columns dramatically reduces the number of parameters of the network, and the model size is only 23.10 MB.

MoAm-YOLOv4 is the final improvement algorithm proposed in this paper. Although twice the model size of YOLOv4-tiny is obtained, Precision, Recall and mAP are all predominant by the MoAm-YOLOv4 algorithm.

As can be seen, compared to the original algorithm of YOLOv4. The final improvement algorithm of this paper, MoAm-YOLOv4 got no significant decline,

Table 1. Comparision of results of five models.

model	P/%	R/%	mAP/%	FPS	S/MB
YOLOv4	92.08	89.84	93.72	53.19	245.52
YOLOv4-tiny	89.54	86.99	92.96	100.74	23.10
Mobilenetv1-YOLOv4	91.02	90.65	92.94	85.30	48.42
YOLOv4+SENet	92.63	91.87	94.54	53.03	245.85
MoAm-YOLOv4	**93.28**	**90.24**	93.45	80.33	**45.85**

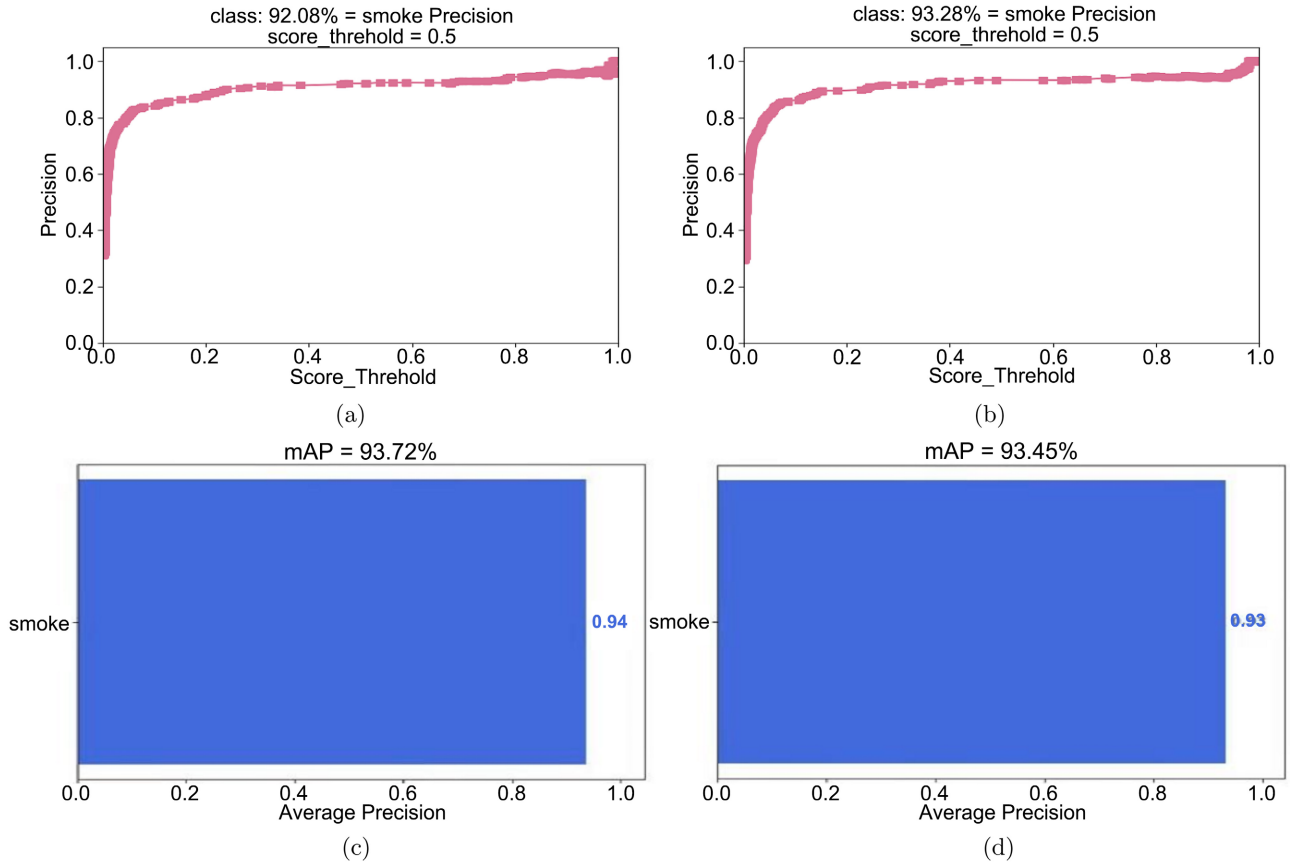

Figure 6. Comparing the mAP and Precision of the two algorithms on the test set.

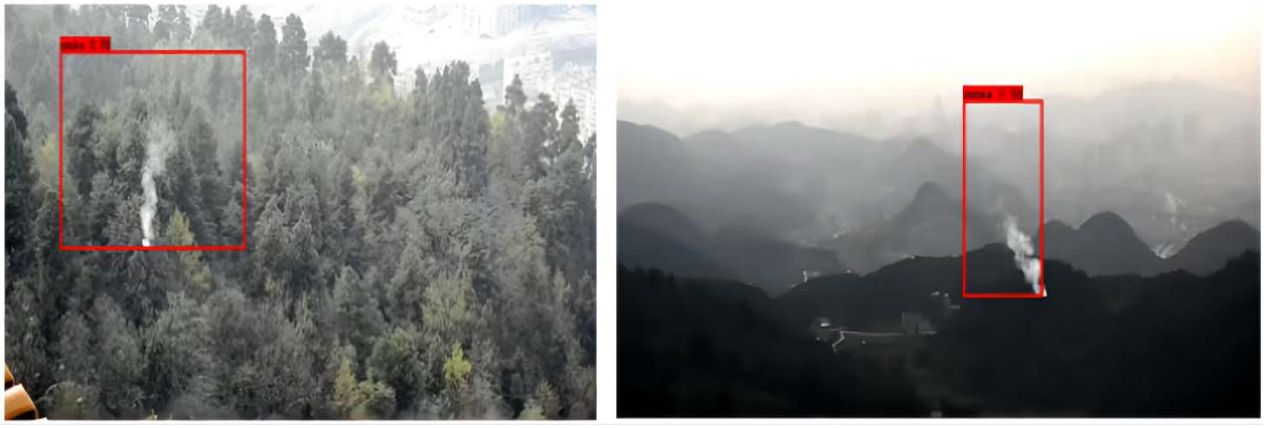

Figure 7. Experimental detection effect.

but Precision, Recall, and mAP increased by 1.8%, 1.6%, and 51%, respectively, and model size was down 80%, this is equivalent to the expense of only a tiny point of mAP, the other indicators are greatly improved, **Figure 6** shows the test results of the two algorithms.

The experimental results show that the proposed improved algorithm has certain advantages in maintaining high accuracy in forest fire smoke detection, and achieving the balance of speed and precision to a large extent, experimental detection effect as shown in **Figure 7**.

5. Conclusions

In this paper, the backbone network is replaced with lightweight network MobilenetV1 based on the YOLOv4 algorithm, which significantly reduces the model size of the network; using the K-means clustering method to find the candidate box size of the data set; adding SENet before the Head output improves the detection accuracy of the algorithm to some extent. The experimental results show that the proposed MoAm-YOLOv4 algorithm works well on the test set, with Precision reaching 93.28%, and mAP reaching 93.45%, while the model size was only 48.58 MB, and the FPS reaching 80.33. Compared with the YOLOv4 algorithm, it increased Precision by 1.2%, FPS by 51% and model size by 80% when sacrificing only 0.27% mAP.

Although the detection method proposed in this paper has achieved good results, there are still many shortcomings, which can be improved from the following aspects: 1) During the training and testing, only more than 2000 images were screened out. Considering the diversity of the forest fire environment, the data set collected did not fully consider the background environment of the forest. The data set lacks diversity. If conditions permit, data sets can be appropriately increased in the future to improve the algorithm's robustness. 2) The research of this paper mainly focuses on the forest environment in the daytime and does not consider the situation at night. In the future, further research can be carried out on forest fire smoke detection in the night-time environment. 3) After replacing the lightweight network, the detection speed of the algorithm has

been significantly improved, but there are still many possibilities for improving the accuracy. The accuracy can be improved by not changing the backbone network through different feature fusion methods. The algorithm follows the feature fusion method of PANet, and the algorithm can be improved in the future in the direction of feature fusion.

Acknowledgements

This research was funded by [the Fundamental Research Funding for the Central Universities of Ministry of Education of China] grant number [18D110408], and [the National Natural Science Foundation of China (NSFC)] grant number [18K10454].

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Jasa, M.K. and Kannan, S.A. (2021) Review on Fire Detection Techniques. *International Journal of Current Engineering and Scientific Research*, **8**, 100-106.

[2] Ma, S., Zhang, Y., Xin, J., Yi, Y., Liu, D. and Liu, H. (2018) An Early Forest Fire Detection Method Based on Unmanned Aerial Vehicle Vision. 2018 *Chinese Control and Decision Conference*, Shenyang, 9-11 June 2018, 6344-6349. https://doi.org/10.1109/CCDC.2018.8408244

[3] Cang, N.-M. and Yu, W.-J. (2020) Early Forest Fire Smoke Detection Based on Aerial Video. *Journal of Physics*: *Conference Series*, **1684**, Article ID: 012095. https://doi.org/10.1088/1742-6596/1684/1/012095

[4] Geetha, S., Abhishek, C.S. and Akshayanat, C.S. (2020) Machine Vision Based Fire Detection Techniques: A Survey. *Fire Technology*, **57**, 591-623. https://doi.org/10.1007/s10694-020-01064-z

[5] Gale, M.G., Cary, G., Van Dijk, A.I.J.M. and Yebra, M. (2021) Forest Fire Fuel through the Lens of Remote Sensing: Review of Approaches, Challenges and Future Directions in the Remote Sensing of Biotic Determinants of Fire Behaviour. *Remote Sensing of Environment*, **255**. Article ID: 112282. https://doi.org/10.1016/j.rse.2020.112282

[6] Zhao, C., Feng, Y., Liu, R. and Zheng, W. (2020) Application of Lightweight Convolution Neural Network in Cancer Diagnosis. *Proceedings of the* 2020 *Conference on Artificial Intelligence and Healthcare*, Taiyuan, 23-25 October 2020, 249-253. https://doi.org/10.1145/3433996.3434042

[7] Vieira, P., Guede-Fernández, F., Martins, L., de Almeida, R.V. and Gambao, H. (2021) A Deep Learning Based Object Identification System for Forest Fire Detection. *Fire*, **4**, Article No. 75. https://doi.org/10.3390/fire4040075

[8] Preeti, T., Kanakaraddi, S., Beelagi, A., Malagi, S. and Sudi, A. (2021) Forest Fire Prediction Using Machine Learning Techniques. 2021 *International Conference on Intelligent Technologies* (*CONIT*), Hubli, 25-27 June 2021, 1-6. https://doi.org/10.1109/CONIT51480.2021.9498448

[9] Fan, R. and Pei, M. (2021) Lightweight Forest Fire Detection Based on Deep Learning. 2021 *IEEE* 31*st International Workshop on Machine Learning for Signal Processing*

(*MLSP*), Gold Coast, 25-28 October 2021, 1-6. https://doi.org/10.1109/MLSP52302.2021.9596409

[10] Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014) Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 580-587. https://doi.org/10.1109/CVPR.2014.81

[11] Ren, S., He, K., Girshick, R. and Sun, J. (2017) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**, 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

[12] He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2017) Mask R-CNN. 2017 *IEEE International Conference on Computer Vision* (*ICCV*), Venice, 22-29 October 2017, 2961-2969. https://doi.org/10.1109/ICCV.2017.322

[13] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You Only Look Once: Unified, Real-Time Object Detection. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*ICCV*), Vegas, 27-30 June 2016, 779-788. https://doi.org/10.1109/CVPR.2016.91

[14] Seydi, S.T., Saeidi, V., Kalantar, B., Ueda, N. and Halin, A.A. (2022) Fire-Net: A Deep Learning Framework for Active Forest Fire Detection. *Journal of Sensors*, **2022**, Article ID: 8044390. https://doi.org/10.1155/2022/8044390

[15] Zhang, Q.X., Lin, G.H., Zhang, Y.M., Xu, G. and Wang, J.J. (2018) Wildland Forest Fire Smoke Detection Based on Faster R-CNN Using Synthetic Smoke Images. *Procedia Engineering*, **211**, 441-446. https://doi.org/10.1016/j.proeng.2017.12.034

[16] Sun, X., Sun, L. and Huang, Y. (2021) Forest Fire Smoke Recognition Based on Convolutional Neural Network. *Journal of Forestry Research*, **32**, 1921-1927. https://doi.org/10.1007/s11676-020-01230-7

[17] Mohnish, S., Akshay, K.P., Pavithra, P., Sarath Vignesh, A., Gokul Ram, S. and Ezhilarasi, S. (2022) Deep Learning Based Forest Fire Detection and Alert System. 2022 *International Conference on Communication, Computing and Internet of Things*, Chennai, 10-11 March 2022, 1-5.

[18] Jin, Y. and Zhang, W. (2020) Real-Time Detection Algorithm with Anchor-Free Network Architecture. *Journal of Zhejiang University* (*Engineering Science*), **54**, 2430-2436. (In Chinese)

[19] Zhang, J.X., Guo, S.W., Zhang, G.L., *et al.* (2021) Fire Detection Model Based on Multi-Scale Feature Fusion. *Journal of Zhengzhou University* (*Engineering Science*), **42**, 13-18. (In Chinese)

[20] Qian, H., Shi, F., Chen, W., Ma, Y. and Huang, M. (2021) A Fire Monitoring and Alarm System Based on Channel-Wise Pruned YOLOv3. *Multimedia Tools and Applications*, **81**, 1833-1851. https://doi.org/10.1007/s11042-021-11224-0

[21] Xu, R., Lin, H., Lu, K., Cao, L. And Liu, Y. (2021) A Forest Fire Detection System Based on Ensemble Learning. *Forests*, **12**, Article No. 217. https://doi.org/10.3390/f12020217

[22] Bochkovskiy, A., Wang, C. and Liao, H.Y.M. (2020) Yolov4: Optimal Speed and Accuracy of Object Detection. ArXiv: 2004.10934.

[23] Howard, A., Zhu, M., Chen, B., Kalenichenko, D., *et al.* (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. ArXiv: 1704.04861

[24] Likas, A., Vlassis, N. and Verbeek, J. (2003) The Global K-Means Clustering Algorithm. *Pattern Recognition*, **36**, 451-461. https://doi.org/10.1016/S0031-3203(02)00060-2

[25] Redmon, J. and Farhadi, A. (2018) Yolov3: An Incremental Improvement. ArXiv: 1804.02767.

[26] Shi, J., Liu, Y., Liu, Q., Zhang, Q. and Fei, Y. (2020) Target Recognition Algorithm Based on Improved Depth Separable Convolution. *Journal of Physics*: *Conference Series*, **1693**, Article ID: 012056. https://doi.org/10.1088/1742-6596/1693/1/012056