Lightweight Fire Detection Algorithm Based on Improved YOLOv5

Dawei Zhang[1], Yutang Chen[2]*
School of Information Engineering, Liaodong University, Dandong, China[1, 2]
School of Computer Science and Technology, Shenyang University of Chemical Technology, Shenyang, China[2]

***Abstract*—Among all kinds of disasters, fire is one of the most frequent and common major disasters that threaten public safety and social development. At present, the widely used smoke sensor method to detect fire is susceptible to factors such as distance, resulting in untimely detection. With the development of computer vision technology, image detection technology based on machine learning has been superior to traditional detection methods in terms of detection accuracy and speed, and has gradually become the emerging mainstream in the field of fire detection. At this stage, most of the methods proposed in related studies are based on high-performance hardware devices, which limits the practical application of relevant results. This paper proposes an improved fire detection algorithm based on the YOLOv5 model to address the common issues of high memory usage, slow detection speed, and high operating costs in current fire detection algorithms. The algorithm introduces FasterNet network into the backbone network to reduce model memory usage and improve detection speed. Using Ghost-Shuffle Convolution (GSConv) in the neck network reduces the number of model parameters and computational costs. Introducing a one-time aggregation cross-stage partial network module (VoV-GSCSP) to enhance feature extraction capability and improve the detection accuracy of the model. The experimental results show that compared with the original YOLOv5 model, the improved model achieves better recognition performance, with an average accuracy of 98.3%, a 31.4% reduction in memory usage, and a 13% increase in detection speed. The number of parameters decreased by 33%, and the computational workload decreased by 35%. The improved algorithm can achieve fast and accurate identification of fires, and the lightweight model is more suitable for the deployment and implementation of general embedded hardware.**

***Keywords*—YOLOv5; FasterNet; GSConv; VoV-GSCSP; Fire detection**

I. Introduction

Fire detection has important application value in safety monitoring, fire warning and other fields. Traditional fire detection methods typically rely on physical sensors and signal processing technologies such as smoke detectors, flame detectors, and temperature detectors [1-4]. Although fire detection and alarm can be achieved in specific scenarios, there are certain limitations in detection accuracy and real-time performance. With the development of computer vision technology, machine learning based image detection technology has surpassed traditional detection methods in terms of detection accuracy and speed, gradually becoming an emerging mainstream in the field of fire detection. Huang et al. proposed a new fire detection classifier model that effectively improves fire detection performance and reduces false alarm rates in both front-end and back-end systems by using rough set theory and support vector machine methods [5]. Sandip et al. utilized a hybrid ensemble technique of maximum average voting classifiers and combined four classifiers to design a fire detection algorithm [6]. They successfully applied it to an intelligent multi-sensor embedded fire detection node prototype, achieving real-time data transmission and analysis. However, the above methods require manual feature selection and extraction, which is time-consuming and labor-intensive, and the algorithm has low robustness.

In recent years, image detection techniques based on deep learning have gradually developed due to their higher detection accuracy and real-time performance. The current mainstream deep learning models include R-CNN [7] (Region based Convolutional Neural Networks), RNN [8] (Feedforward Neural Networks), SSD [9] (Single Shot Multibox Detector), and YOLO [10] (You Only Look Once), among others. Many researchers have optimized and improved these models. Shi Lei et al. improved SSD using DenseNet network to enhance the detection ability of small targets, and introduced Focal loss function to solve the problem of imbalanced positive and negative samples, thereby significantly improving the detection performance of the network [11]. However, the complexity of the DenseNet network structure increases too much additional computational cost and memory usage, while also slowing down inference speed. In addition, Wang Yinkai et al. improved YOLOv5 by introducing decoupling heads, CBAM attention mechanisms, and weighted bidirectional feature pyramid networks (BiFPN), achieving significant improvements in average accuracy and other indicators [12], but also increasing memory usage and inference time. In addition, Zhang Wei et al. added dilated convolution and DenseNet networks to the feature extraction network of YOLOv3, improving the detection accuracy of the algorithm [13], but also increasing memory usage and inference time.

In order to more efficiently and accurately identify and prevent fire incidents, and better adapt to and meet the common hardware infrastructure requirements in the algorithm deployment process. Based on the YOLOV5 object detection algorithm, this paper proposes another lightweight YOLOV5 improved fire detection algorithm. The proposed algorithm integrates the advantageous model structure of YOLOv5, and introduces FasterNet [14] to optimize the backbone network. GSConv [15] and C3GS are applied in the neck network to replace the original convolution and C3, thus achieving lightweight algorithm. By applying VoV-GSCSP, Gather-Execite [16] attention mechanism, and C3GE, the model can focus on important features and improve detection accuracy. Based on publicly available datasets, relevant experiments were organized to verify the effectiveness and superiority of the algorithm.

The main contributions of this research are as follows:

- Identifying the limitations and research gaps in existing computer vision-based fire detection systems.
- Proposing a deep learning-based approach using the YOLOv5 algorithm to address these challenges as well as improve the Lightweight degree and detection speed.
- GSConv replaces the original convolution of Bottleneck in C3 to form a new optimized structure module, C3GS structure module, to save computing costs.
- Gather-Exact is introduced into C3 to form a new optimized structure module C3GE to improve the detection accuracy.
- Evaluating the proposed method on a public dataset using extensive performance evaluation metrics.

The remaining part of the paper is organized as follows. Section II discusses the improvement and optimization strategy of the algorithm and describes the design process of the algorithm. Section III discusses the steps and details of algorithm-related validation experiments, and analyzes and explains the experimental results. Section IV draws the paper to a conclusion and suggests areas for further research.

II. Establishment of Improved Algorithms

YOLO algorithm is an object detection algorithm based on deep learning, which has high detection efficiency, simple structure and strong generalization ability, and is widely used in the field of object detection. YOLOV5 is a more mature version of the YOLO algorithm. Compared to other versions, it has faster detection speed and higher detection accuracy, and its lightweight model structure and scalability are more convenient for later deployment [17]. This article uses YOLOv5s-6.0 version as the basic algorithm model to improve and optimize the algorithm.

A. FasterNet

The lightweighting level of the model is an important optimization objective of object detection algorithms. In the past, MobileNet [18], GhostNet [19] and other strategies were widely used to improve the model, achieving a certain degree of effectiveness [20,21]. Although this strategy can reduce the number of model parameters, computation, and memory usage, it also increases the additional memory access overhead, leading to a decrease in the inference speed of the model.

FasterNet can effectively reduce redundant calculations and memory accesses, thereby speeding up model inference [22]. Fig. 1 is the diagram of FasterNet and PConv modules. As the main structure Block of FasterNet, its structure is shown in Fig. 1(a). Each FasterNet Block contains a PConv layer (Partial convolution), followed by two 1*1Conv layers (Pointwise Convolution). Each Conv layer is connected to batch normalization (BN) and ReLU activation function. In addition, a residual structure is also introduced to better ensure the good generalization ability of the network. The PConv layer structure is shown in Fig. 1(b). Some of the input channels in the structure are used as representatives of the entire feature map to perform conventional convolution (Conv) to extract spatial features, and the remaining channels are directly mapped to the output, thus reducing redundant calculations. PConv can better process the local information of the image, while 1*1Conv can extract global features through point-by-point convolution. The combination of the two extracts rich and diverse feature information at different scales.

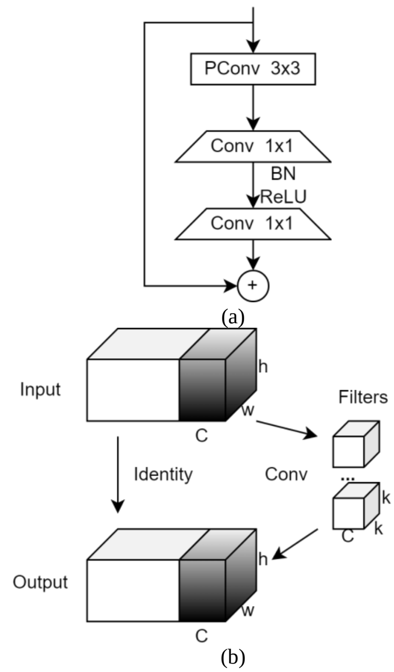

Fig. 1. Diagram of fasternet and PConv modules.

The computational complexity of Pconv and conventional convolution is shown in Eq. (1) and Eq. (2), where h and w are the height and width of the channels, c is the number of channels for partial convolution, C is the number of channels for the input feature map, and k is the filter size. In typical cases, the ratio of the number of channels in partial convolution to the number of channels in the input feature map is c/C=1/4, and the computational cost of PConv is only 1/16 of that of conventional convolution.

$$h \times w \times k^2 \times c^2 \tag{1}$$

$$h \times w \times k^2 \times C^2 \tag{2}$$

In addition, the memory access of PConv is relatively small, and the computational cost is calculated as shown in Eq. (3):

$$h \times w \times 2c + k^2 \times c^2 \approx h \times w \times 2c \tag{3}$$

The Proposed algorithm using FasterNet, which has a small number of parameters and fast inference speed, to build a backbone feature extraction network to accelerate the detection speed of the model.

B. GSConv and VoV-GSCSP

Convolutional layers are an important structural component of the YOLOv5 model. Convolutional layers are the most computationally and memory intensive part of algorithms, especially when the number of parameters is large, their computation and memory usage will be more significant.Ghost-shuffle Convolution (GSConv) is a convolution operation that is spliced by splicing ordinary convolutional convolution and Depthwise Separable Convolution (DWConv). It can achieve the same learning effect with less than 70% of the computational cost of ordinary convolutions [16].

The C3 module is another important structure of the YOLOv5 model. The introduction of C3 module can effectively improve the performance and efficiency of YOLOv5 model. Although the C3 module has a smaller computational cost compared to traditional residual connections, its complex structural design and parameter count, as well as the computational and memory consumption, are still significant expenses for the system. Propose an algorithm to replace the original convolution of Bottleneck with GSConv convolution, forming a new C3 structure, denoted as C3GS. Thanks to the advantageous structure of GSConv, the YOLOv5 model has been improved and optimized in the convolutional layer and C3 module, resulting in a significant increase in the lightweight level of the model. Fig. 2 shows the module structure diagram of GSConv and C3GS. Fig. 2(a) shows the structure of GSConv. Fig. 2(b) shows the structure of C3GS.

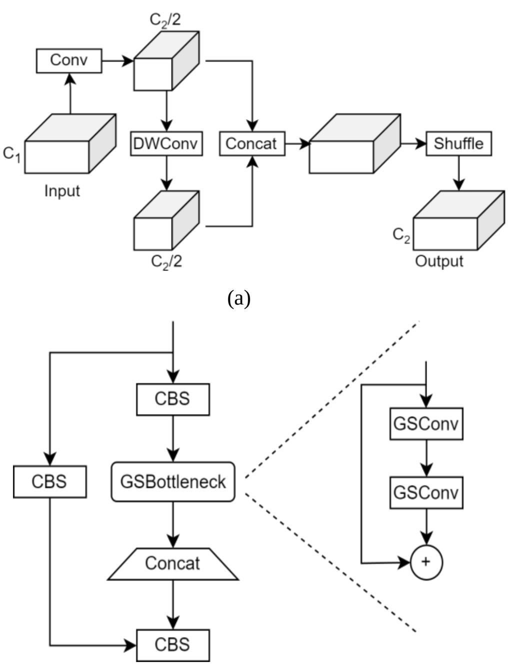

Fig. 2. Diagram of GSConv and C3GS modules.

The one-time aggregation cross stage partial network module (VoV-GSCSP) is a paradigm design of Slim Neck, which constructs cross level partial networks through a one-time aggregation method. The module extracts richer semantic information by adding low dimensional feature maps to input feature maps. Furthermore, it can reduce the complexity of network structure and computation while maintaining sufficient accuracy. The specific structure of VoV-GSCSP is shown in Fig. 3.

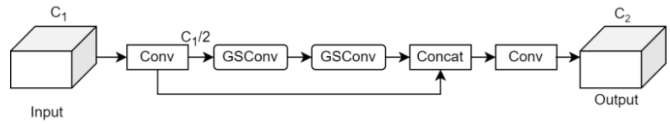

Fig. 3. Diagram of VoV-GSCSP module.

GSConv and VoV-GSCSP have performed well in the field of fire detection [23-24]. Propose an algorithm to replace the original convolution and C3 with GSConv convolution and C3GS in the original structure of the neck network, reducing the number of model parameters and computational complexity, and introducing VoV-GSCSP to improve the detection accuracy of the model.

C. Gather-Excite and C3GE

Gather Excite is an attention mechanism that motivates, mainly composed of two parts: Gather and Excite. Among them, the core function of Gather is to aggregate feature responses over a large spatial range, thereby providing rich contextual information for the model. The function of Excite is to redistribute the information obtained through Gather aggregation to the original features, making them more suitable for the current task needs. The combination of the two enhances the expression ability of module features, which is more conducive to improving the generalization ability of the model.

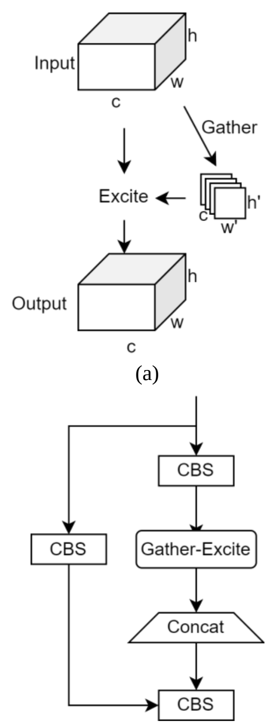

Fig. 4. Diagram of gather-excite and C3GE modules.

The C3 module improves the expression ability of features through residual connections, but it can easily lead to the model not being able to fully obtain global contextual information during the feature extraction process, and increase sensitivity to noise in input data, thereby affecting the quality and stability of feature extraction. Introducing attention mechanism into the C3 structure can effectively compensate for the shortcomings of C3 and leverage the advantages of both [25]. Propose an algorithm based on the C3 structure, introducing the Gather Excite to form a new composite C3 structure, denoted as C3GE. By utilizing the gather-Excite attention mechanism, the shortcomings of the C3 module in the feature extraction process can be compensated for, thereby improving the performance and efficiency of the model. Fig. 4 shows the structural diagram of the gatherer excite and C3GE modules. Among them, Fig. 4 (a) shows the structure diagram of the gatherer excite module, and Fig. 4(b) shows the structure diagram of the C3GE module.

The proposed algorithm incorporates the Gather Extreme attention mechanism into the backbone network and neck network, and purposefully concentrates the model on the detection object by applying weight sparsity, further improving the model's feature extraction ability and computational efficiency.

Based on the above research and analysis, the improvement plans for model lightweighting related to the YOLOv5 algorithm model and the optimization strategies for deep learning object detection algorithms can be summarized as shown in Table I. Overview of Model Lightweighting Approaches.

TABLE I. OVERVIEW OF MODEL LIGHTWEIGHT APPROACHES

Reference	Approach	Primary techniques	Main objectives
[14]	Maintaining accuracy while increasing operational speed.	Proposing the FasterNet model based on PConv.	Improving the floating-point operation efficiency of neural networks
[15]	Maintaining accuracy while reducing the parameter redundancy and the calculation overhead	Introducing the GSConv and the FPN into the feature fusion network, and Introducing DOConv and Shufflenet into the backbone network.	Implementing lightweight real-time detection models
[16]	Achieveding high-precision detection of small targets	Embedding Gather Excite attention into the model and replacing IoU with Normalized Wasserstein distance.	Improving the detection accuracy of small targets and weak signals
[17]	Realizeding high-precision and high recall detection of smoke.	Applying the YOLOv5 model and introducing the CBAM module and Mish activation function.	Realizing smoke detection in remote sensing images
[18]	Achieving highly accurate classification	Implementing MobileNet architecture in conjunction with hyperparameters and optimization	Designing models to achieve precise classification
[19]	Implementing significantly better FLOP parameters than SOTA and CNN models.	Introducing GhostNet to Reduce Feature Map Redundancy	Designing a lightweight model to deploy devices with limited memory and computing resources
[20]	Maintaining highly accuracy While reducing the amount of computation, model size, and hardware cost.	Introducing MobileNetV3 and the CBAM attention mechanism	Improving the defects of the algorithm, such as complex network, many parameters, and large amount of calculation.
[21]	Realizing high-precision detection with low computational cost consumption	Propose the DFM-CPFN method and the VoVNet,using the ShuffleNetV2 to lightweighting the network.	Lightweighting the models to meet the deployment of mobile and embedded devices
[22]	Implementing high-precision and high-speed detection	Introducing the FasterNet model and proposing a dual attention feature fusion module	Meeting the real-time requirements of unstructured road scene segmentation
[23]	Improving target detection accuracy and position detection accuracy	Adopting optimized Slim Neck structure, introducing the GSConv and the VoV GSCSP	Improving the detection accuracy of the fire detection system and the accuracy of the detection position
[24]	Achieving a higher computational cost-effectiveness of the real-time detectors	Introducing the GSConv	Lightweighting the real-time inspection system
[25]	Improving detection accuracy and recall while maintaining comparable speeds	combining C3 and the attention mechanism	Implementing detection of small targets in complex backgrounds

D. The proposed Improved YOLOV5 Algorithm

In the current application environment of video surveillance-based security systems, fire detection models often need to be deployed on mobile or embedded devices to exert their effectiveness. From the previous research, it can be seen that the lightweighting of the model has important practical significance for the detection model of the algorithm, and the relevant optimization strategies of the previous sequence have practical effects on achieving model lightweighting. The lightweight of the algorithm lies in meeting the requirements of fire detection accuracy and real-time response, while reducing model size and hardware loss, and improving operational efficiency. Based on this goal, this article proposes a lightweight YOLOv5 improved algorithm for fire detection model.

Fig. 5 shows the overall structure of the improved YOLOV5 lightweight fire detection algorithm. As shown in Fig. 5, the improved algorithm network structure mainly consists of four parts: input end, backbone network, neck network, and head network. The image to be detected is introduced into the algorithm detection network by the input end, and after recognition and processing by each module, the final detection result is generated and output by the head network.

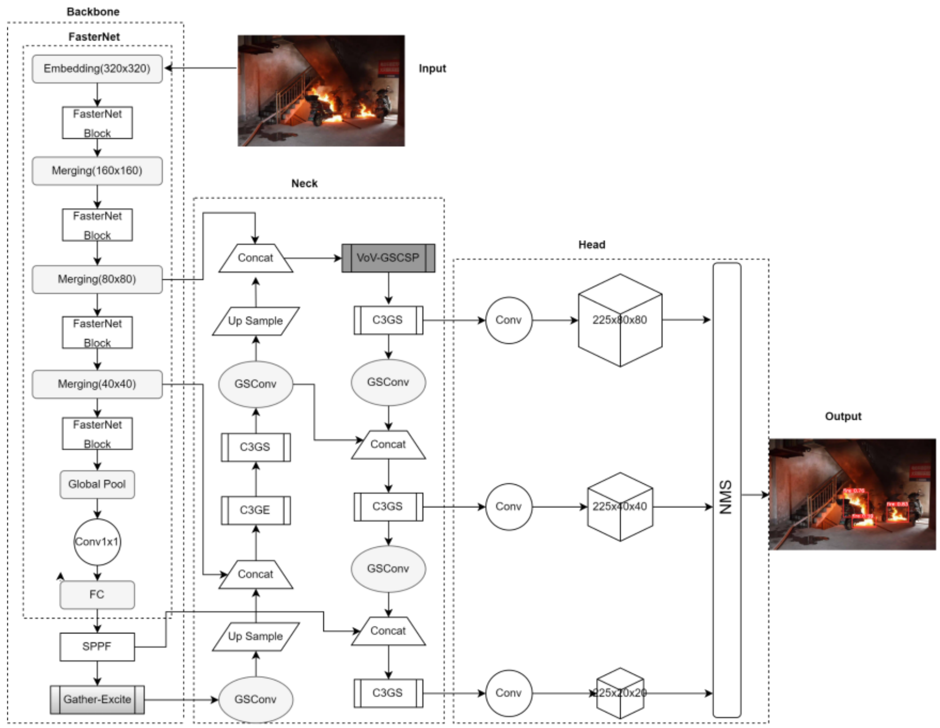

Fig. 5. Improved algorithm structure diagram.

The working principles of each module of the improved algorithm are as follows:

- At the input end, the input image is adjusted to the uniform size required by the model and normalized.
- The image is processed through a backbone network to extract semantic and spatial features from the image.
- The neck network fuses low resolution feature maps with upper-level feature maps through a bottom-up path, capturing the detailed information of lower-level features and fusing them with higher level features. At the same time, through a top-down path, lower resolution feature maps are gradually generated starting from high-level feature maps. A series of fusion operations with different scale feature maps are included in the two paths. Through bottom-up and top-down paths and feature fusion, multi-scale feature fusion and alignment are achieved, providing rich contextual and detailed information for subsequent processing.
- The head network converts the features extracted by the backbone network and the neck network into the category probability and bounding box coordinate information of the target. Finally, NMS (non-maximum suppression) and threshold filtering are used to obtain the final detection results, which are used to locate and identify the target in the input image.

The improved algorithm introduces FasterNet into the backbone network to lighten the model as much as possible while maintaining accuracy; In the neck network section, GSConv and C3GS are used to replace the original convolution and C3, while further lightweight the model; Finally, VoV-GSCSP, Gather-Excite attention module and C3GE are introduced to make the model focus more on important information and improve model accuracy when learning feature representations.

III. Experiments and Result Analysis

A. Experimental Environment

The relevant experiments in this article use Python 1.12.0 as the software framework, Python as the programming language, and CUDA version 11.3 to accelerate model training. The hardware environment for model training includes a CPU model of Intel (R) Xeon (R) CPUE5-2686 v4, 32GB of memory, and a GPU model of Nvidia GeForce RTX 3090 24GB.

B. Experimental Dataset

To ensure the detection accuracy of the model, the quality of the dataset is crucial. The dataset related to the research experiment in this article is a public dataset, which includes 2637 images of various environments, angles, and brightness for fire detection tasks, with fire as the detection object[1]. During the experiment, the dataset was randomly shuffled to eliminate possible sequential correlations and ensure the reliability of training and evaluation. Subsequently, 2056 images were randomly selected from the entire dataset as the training set, and the remaining 617 images were used as the validation set. Conduct subsequent experiments on the training and validation sets established around the aforementioned principles.

C. Ablation Experiment

In order to verify the contribution of each algorithm module to the overall algorithm network, an ablation experiment of the algorithm was designed. The overall experimental results are shown in Table II. Functional modules participating in the evaluation include FasterNet, GSConv, C3GSC, VoV-GSCSP, Gather-Excite and C3GE. Among them, the GSConv and C3GS modules are both perfect optimizations after the introduction of the basic GSConv structure, and are merged into one project, recorded as GSConv. The introduction of strategies and modules such as VoV-GSCSP, Gather-Excite and C3GE aims to maintain the advantageous parameters while improving the recognition accuracy of the algorithm from a multi-faceted perspective, so it is merged into one project, recorded as Gather-Excite. The evaluation indicators to evaluate the comprehensive performance include model detection accuracy, memory usage, inference speed (FPS), parameter amount, floating point operation amount/algorithm complexity (GFLOPs). The detection accuracy indicators use mAP and mAP95 in the COCO evaluation standard, where mAP refers to the average detection accuracy when the confidence threshold is 0.5. mAP95 represents the average detection accuracy within the range of confidence thresholds from 0.5 to 0.95 with a step size of 0.05. The following is an analysis of the ablation effects of each module.

The algorithm introduces FasterNet to optimize the backbone network. The experimental results show that the memory usage of the algorithm has decreased from 13.7M to 10.9M, the parameter count has decreased from 7.02M to 5.54M, the computational complexity has decreased from 15.9 to 11.2, and the inference speed frame rate FPS has increased from 43.6HZ to 48.4HZ. On the contrary, the average accuracy of the algorithm has decreased significantly. From this, it can be seen that the introduction of FasterNet has made a significant contribution to the lightweighting of the model. With the strengthening of lightweighting, the inference speed of the algorithm is also improved. However, the cost of lightweight is the decrease in model recognition accuracy, which requires further optimization strategies to compensate for the corresponding accuracy loss.

The algorithm uses GSConv convolution and optimization structure module C3GS to replace the original convolution and C3 structures of the neck network. The experimental results show that the detection accuracy and inference speed of the algorithm remain basically unchanged. The memory usage of the algorithm has decreased from 13.7M to 11.9M, and the computational complexity of the algorithm has also decreased from 7.2 to 6. From this, it can be seen that the optimization strategy of the relevant modules can improve the lightweight degree of the model while maintaining a relatively stable processing speed and accuracy. But the more stable cost is that the optimization level of the model's spatial and temporal complexity is not significant.

The algorithm introduces VoV-GSCSP and introduces attention mechanism to optimize the C3 structure of the original model. According to the experimental results, this optimization strategy can effectively improve the detection accuracy, average accuracy, and mAP@0.5 0.95 increased from 99.2% and 73.2% to 99.4% and 75.3%, respectively. However, it inevitably leads to a sharp decrease in the lightweight level of the model, with significantly lower memory usage, parameter count, floating-point operations, and detection speed compared to the original parameters. From this, it can be seen that optimizing the structure increases the model's attention to important features, improves the model's ability to integrate and extract features, and the optimization strategy of related modules can significantly improve the average accuracy of the algorithm. At the same time, it can be seen that a single application model detection accuracy optimization strategy will inevitably lead to an increase in model space occupation and complexity, as well as an increase in operating costs. Therefore, the high performance and high degree of lightweighting of algorithms require a balance between multiple optimization strategies to achieve the ultimate lightweighting goal of the algorithm.

After the algorithm is comprehensively introduced into the above-mentioned optimization strategy scheme, the performance of the algorithm has been significantly improved. From the experimental results, it can be seen that the average accuracy of the algorithm is basically stable, reaching 98.3%, the memory occupation is reduced by 31.4%, and the detection speed is increased by 13%. The number of parameters is reduced by 33%, and the amount of computation is reduced by 35%. It can be seen that the above-mentioned structural adjustment and optimization strategies for the YOLOV5 algorithm model are effective. Compared with the original algorithm, the improved algorithm achieves higher lightweight and recognition speed while ensuring high accuracy. It has high applicability to deployment and implementation environments with limited hardware resources that are common in the field of fire detection.

TABLE II. ABLATION EXPERIMENTS RESULTS TABLE

Models	**FasterNet**	**GSConv**	**Gather-Excite**	**mAP/%**	**mAP95/%**	Memory usage /M	Parameter quantity /M	GFLOPs	FPS/HZ
YOLOV5				99.2	73.2	13.7	7.02	15.9	43.6
Proposed	✓			95.5	66.9	10.9	5.54	11.2	48.4
		✓		99.1	71.1	11.9	6	14.5	43.5
	✓	✓		97.9	68.2	9.1	4.53	9.9	47.6
			✓	99.4	75.3	14.1	7.18	16.2	43.2
	✓	✓	✓	98.3	71.2	9.4	4.69	10.4	49.1

Meanwhile, experimental results have shown that while achieving higher lightweighting and detection speed, the new model has a certain decrease in detection accuracy compared to the original YOLOV5 model. This loss of accuracy stems from the adjustment and compression of the model structure during the process of model lightweighting. This will result in proposing algorithms that consume more detection time while maintaining the same detection accuracy. The lightweighting level and detection accuracy of the model are a dynamic balance. Although the current loss of detection accuracy is still within the allowable range, further in-depth research is needed in the future to determine the relationship between the two and obtain optimal results.

D. Comparative Experiments

In order to further verify the superiority and progressiveness of the proposed improved algorithm, under the same conditions of initialization weight, parameter setting and hardware environment, based on the same data set mentioned above, a comparative experiment was conducted between the improved algorithm and Faster R-CNN, SSD, YOLOv3 and YOLOv7 target detection algorithms.

To further demonstrate the superiority of the improved algorithm, performance comparisons were made between the improved algorithm and Faster R-CNN, SSD, YOLOv3, and YOLOv7 object detection models. The experimental results are shown in Table III.

TABLE III. COMPARATIVE EXPERIMENT RESULTS TABLE

Models	mAP /%	mAP95 /%	Memory usage /MB	FPS /Hz
Faster R-CNN	88.9	68.6	320	11.8
SSD	92.5	70.1	84.6	20
YOLOv3	94.1	70.8	45.1	20.1
YOLOv7	95.3	75.6	71.3	33.4
Proposed	98.3	71.2	9.4	49.1

The experimental results show that the improved YOLOv5 algorithm exhibits excellent performance in fire detection tasks. The two average accuracy averages of the improved YOLOv5 algorithm reached 98.3% and 71.2%, respectively, showing a significant improvement compared to other algorithms. In addition, the inference speed of the improved algorithm has significantly improved, reaching 49.1 FPS, which is better than other algorithms. At the same time, the model only occupies 9.4MB, which makes the algorithm easy to run on resource limited devices, providing the possibility for the widespread application of real-time fire detection.

The visual detection comparison results between the improved lightweight YOLOV5 algorithm model and the conventional YOLOV5 algorithm model are shown in Fig. 6. Fig. 6(a) shows the fire detection results of the unimproved conventional YOLOV5 detection algorithm, and Fig. 6(b) shows the detection results of the improved lightweight YOLOV5 detection algorithm proposed in this paper.

From Fig. 6, it can be observed that the proposed algorithm exhibits good detection performance for fire detection. Compared to conventional detection algorithms, the proposed algorithm has higher prediction confidence, better subjective effect, no missed detections, better overall performance, and meets the design expectations of the algorithm.

(a)

(b)

Fig. 6. Visual detection comparison results.

IV. CONCLUSION

Conventional fire detection algorithms generally have problems such as memory occupation, high operating cost, and low detection efficiency, which are difficult to be widely deployed and implemented in practice. In order to meet the requirements of the computational cost of the detection algorithm in the actual working environment and improve the detection speed, a lightweight fire detection algorithm based on the improved YOLOv5 model is proposed. The improved algorithm introduces FasterNet into the backbone network of the YOLOV5 algorithm, and the model is as lightweight as possible under the premise of maintaining accuracy. In the neck network, GSConv and C3GS are used to replace the original convolution and C3 to further lighten the model while maintaining stable running speed and accuracy, and VoV-GSCSP, Gather-Exact attention mechanism and C3GE are introduced to make the model focus more on important information and improve the accuracy of the model when learning feature representation. Experimental results show that the improved algorithm can achieve efficient and accurate fire detection while maintaining the advantages of lightweight model. However, challenges still exist, and it is proposed that the original model is compressed in the process of lightweighting, resulting in a partial loss of model detection accuracy. The follow-up research plans to deploy and test the algorithm under actual complex working conditions and on a variety of hardware devices, and at the same time explore more optimization strategies to reduce accuracy loss and improve the performance of the algorithm, so as to meet the needs of efficient and stable fire detection in the actual working environment.

ACKNOWLEDGMENT

This work was supported by the Liaoning Provincial Department of Education's Basic Research Project for Universities [Grant No. JYTMS20230711] and Liaoning Province Science and Technology Plan Joint Program (Fund) Project [Grant No. 2023JH2/101700009].

REFERENCES

[1] A. Shadab, M. T. L. Ansair, S. K. Raghuwanshi, and S. Kumar, "Smoke Detection Using rGO-Coated eFBG Sensor for Early Warning of Coal Fire in Mines," IEEE Sensor Journal, vol. 23, pp. 2153-2160, December 2022.

[2] S. Fouzar, T. A. Eftimov, I. P. Kostova, T. L. Dimitrova, A. Benmounah, and A. Lakhssassi, "A Simple Fiber Optic Temperature Sensor for Fire Detection in Hazardous Environment Based on Differential Time Rise/Decay Phosphorescence Response," IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1-8, August 2022.

[3] (3)S. Geetha, C. S. Abhishek, and C. S. Akshayanat, "Machine vision based fire detection techniques," A survey. Fire technology, vol. 57(2), pp. 591-623, 2021.

[4] (4)M. Prakash, S. Neelakandan, M. Tamilselvi, S. Velmurugan, S. Baghavathi Priya, and E. Ofori Martinson, "Deep Learning-Based Wildfire Image Detection and Classification Systems for Controlling Biomass," International Journal of Intelligent Systems, vol. 2023(1), pp. 7939516, 2023.

[5] X. C. Huang, L. Du, "A Fire Detection and Recognition Optimization Based on Virtual Reality Video Image," IEEE Access, vol. 8, pp. 77951-77961, 2020.

[6] J. Sandip, K. S. Saikat, "Hybrid Ensemble Based Machine Learning for Smart Building Fire Detection Using Multi Modal Sensor Data," Fire Technology, vol. 59, pp. 473-496, 2023.

[7] K. Avazoy, M. Mukhiddinov, F. Makhmudov, and Y. I. Cho, "Fire Detection Method in Smart City Environments Using a Deep-Learning-Based Approach," Electronics, vol. 11, pp. 73, January 2022.

[8] R. Ghosh, A. Kumar, "A hybrid deep learning model by combining convolutional neural network and recurrent neural network to detect forest fire," Multimedia Tools and Applications, vol 81(27), pp. 38643-38660, 2022.

[9] S. H. Kang, J. S. Park, "Aligned Matching: Improving Small Object Detection in SSD," Sensors, vol. 23, pp. 2589, March 2023.

[10] Z. Zakria, J. H. Deng, R. Kumar, S. Muhannad, J. Y. Cai, and J. Kumar, "Multiscale and Direction Target Detecting in Remote Sensing Images via Modified YOLO-v4," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 15, pp. 1039-1048, 2022.

[11] L. Shi, H. G. Zhang, and J. F. Yang, "VIDEO-BASED FIRE AND SMOKE DETECTION BASED ON IMPROVED SSD," Computer Applications and Software, vol. 38, pp. 161-167, 2021.

[12] Y. K. Wang, L. Cao, J. C. Qian, and H. F. Lin, "Multi-scale Forest Fire Recognition Using Improved YOLOv5 Algorithm," Journal of Forestry Engineering, vol. 8, pp. 159-165, March 2023.

[13] W. Zhang, J. J. Wei, "Improved YOLO v3 Fire Detection Algorithm Embedded in DenseNet Structure and Dilated Convolution Module," Journal of Tianjin University (Science and Technology), vol. 53, pp. 976-983, June 2020.

[14] J. Chen, S. H. Kao, H. He, W Zhuo, S. Wen, and C. H. Lee, "Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks," IEEE, 2023.

[15] L. Yuan, Y. R. Chen, T. Hai, R. Gao, and W. H. Wu, "DGNet: An Adaptive Lightweight Defect Detection Model for New Energy Vehicle Battery Current Collector," IEEE Sensors Journal, Vol. 23, pp. 29815-29830, 2023.

[16] K. J. Sun, J. Huo, Q. Liy, and S. Y. Yang, "An Infrared Small Target Detection Model via Gather-Excite Attention and Normalized Wasserstein Distance," Mathematical biosciences and engineering, vol. 20, pp. 19040-19064, 2023.

[17] H. Liu, J. Li, J. Du, B. Zhao, Y. Hu, D. Li, and W. Yu, "Identification of Smoke from Straw Burning in Remote Sensing Images with the Improved YOLOv5s Algorithm," Atmosphere, vol. 13(6), pp. 925, 2022.

[18] E. Elfatimi, R. Eryigit, and L. Elfatimi, "Beans Leaf Diseases Classification Using MobileNet Models," IEEE Access, vol. 10, pp. 9471-9482, 2022.

[19] M. Alansari, O. A. Hay, S. Javed, A. Shoufan, Y. H. Zweiri, and N. Werghi, "GhostFaceNets: Lightweight Face Recognition Model From Cheap Operations," IEEE Access, vol. 11, pp. 35429-35446, 2023.

[20] Y. B. Yang, D. Li, "Lightweight Helmet Wearing Detection Algorithm of Improved YOLOv5," Computer Engineering and Applications, Vol. 58, pp. 201-207, 2022.

[21] J. H. Yang, H. Li, Y. Y. Du, Y. Mao, and Q. Liu, "A Lightweight Object Detection Algorithm Based on Improved YOLOv5s," Electronics Options & Control, vol. 30, pp. 24-30, 2023.

[22] C. Bai, L. Zhang, L. Gao, L. Peng, P. Li, and L. Yang, "Real-time segmentation algorithm of unstructured road scenes based on improved BiSeNet," Journal of Real-Time Image Processing, vol. 21(3), pp. 91, 2024.

[23] M. L. Luo, L. H. Xu, Y. L. Yang, M. Cao, and J. Yang, "Laboratory Flame Smoke Detection Based on an Improved YOLOX Algorithm," Applied Sciences, vol. 12, pp. 12876, 2022.

[24] H. Li, J. Li, H. Wei, Z. Liu, Z. Zhan, and Q. Ren, "Slim-neck by GSConv: a lightweight-design for real-time detector architectures," Journal of Real-Time Image Processing, vol. 21(3), pp. 62, 2024.

[25] Z. H. Hu, Y. Wang, "Improved Traffic Sign Detection Algorithm for YOLOv5," Computer Engineering and Applications, vol. 59, pp. 82-91, 2023.