Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **9**, Number **3**, August **2024**

A Lightweight YOLOv5s Based Method for Automotive Glass Defect Detection and the Deployments on Full-size Inference

Zhe Chen[1,2], Hui Lv[1,2], Shi-Hao Huang*[1,2] , Zhi-Xue Luo[1,2], Jin-Hao Liu[1,2]

[1]Fujian Key Laboratory of Automotive Electronics and Electric Drive
Fuzhou 350118, China
[2]School of Electronic, Electrical Engineering and Physics, Fujian University of Technology
Fuzhou 350118, China

*Corresponding author: Shi-Hao Huang

Received October 17, 2023, revised January 27, 2024, accepted March 30, 2024.

Abstract. *The current mainstream target detection models exhibit several issues including high complexity and large volume. These models display poor detection speed and accuracy when it comes to identifying automotive glass defects. To address these problems, an automotive glass defect detection method based on improved YOLOv5s is proposed for the process of detecting glass defects in the face of the low accuracy of defect recognition at multiple scales, as well as the real-time detection requirements. To begin, the lightweight model ShuffleNetv2 was used to replace the model's backbone network in order to increase model performance while lowering model complexity; Second, the model compression module Ghost is introduced to generate feature maps that match the number and size of the ordinary convolutional channels at a lower cost, achieving model compression while also ensuring model performance; and finally, the necking network PANet is replaced by Bifpn, which fully integrates feature information of different scales and improves the model's detection accuracy for various types of defects. The experimental findings demonstrate that the modified YOLOv5s model in this study improves the Map by 2.8% and decreases the parameter amount calculation and weight size by 79.11% and 71.53%, respectively, when compared to the original YOLOv5s;Furthermore, this paper incorporates SAHI (Slicing Aided Hyper Inference) technology into the lightweight automotive glass defect detection model to investigate the full-size automotive glass defect detection method, which solves the problem of heavy GPU burden caused by the traditional target detection model directly reasoning about the large resolution images, and serves as a reference for the large-size image reasoning Program.*
Keywords: *automotive glass, YOLOv5s, Ghost module, Bifpn, SAHI*

1. **Introduction.** Due to the influence of the production process or operation, automotive glass may form several different types of defects during the production process, and these defects will not only affect the aesthetics of the automotive glass, interfere with the driver's field of vision, but also pose serious safety hazards. In order to pursue better safety and aesthetics, efficient classification and identification of automotive glass defects has become a key part of many automobile companies' quality control of automotive glass, making automotive glass defect detection [1, 2, 3] one of the focuses of target detection research.

Traditional manual inspection methods not only have low inspection efficiency and high work intensity, but also need a significant investment in manpower. Many industries

employ classical machine vision to identify and recognize glass defects as machine vision inspection technology advances. Despite this, machine vision-based defects detection technology has largely replaced manual inspection and has a high detection rate in several areas. Traditional machine vision approaches, on the other hand, have drawbacks such as the necessity for specific preprocessing methods to extract features, susceptibility to ambient light, and high equipment operating and maintenance costs.

Deep learning [4] target detection techniques based on single-stage and two-stage algorithms began to be proposed with the rapid growth of deep learning in the field of target detection. Girshick et al. [5] introduced a region-based convolutional neural network (R-CNN) in 2014, breaking the scenario when target identification algorithms were caught in the bottleneck of development.Following that, deep learning-based target recognition algorithms were introduced one after the other, a summary of the related work is shown in Table 1. Moreover, the application field of deep learning object detection and Segmentation [6, 7, 8, 9] technology has been expanding rapidly, thanks to the development of deep learning technology and the integration of reinforcement learning [10], transfer learning [11], and other advanced technologies.

The single-stage algorithms have faster detection speeds than the two-stage algorithms, and in order to meet the real-time need of modern industrial production, the single-stage method YOLOv5s is chosen as the benchmark model in this study. Due to the broadening of the network depth through down-sampling, the realization of multi-scale detection at the same time caused a large amount of feature information to be lost, resulting in the detection of small and medium-sized target detection on the detection accuracy is not good for YOLOv5s. Furthermore, using the C3 structure on the backbone network results in poor detection speeds and limited applications. To solve these issues, this study provides a lightweight detection algorithm for YOLOv5s automotive glass defects and realizes automotive glass defect identification based on full-size photos using SAHI technology. This paper's primary contributions are as follows:

1. Replace the YOLOv5 backbone network with the lightweight backbone feature extraction network ShuffleNetv2, and insert the Ghost module into the neck network at the same time. The replacement backbone network realizes the lightweight model while reducing the number of times of down-sampling and the loss of feature maps due to down-sampling, and the neck network replaces the ordinary convolution and bottleneck layer with the Ghost module to achieve the effect of reducing model computation volume without losing feature information.

2. To realize the fusion of deep and shallow feature information and increase the model's capacity for tiny and medium-sized target identification, the Bifpn feature pyramid structure is employed to replace the FPN feature fusion network.

3. By incorporating SAHI slicing inference, slicing inference for large-size images is realized, and the hardware configuration requirements of the deployed detection platform are not increased.

The rest of the paper is organized as follows: Section 2 introduces the YOLOv5s algorithm; Section 3 presents an improved automotive glass defect detection model incorporating SAHI inference; Section 4 describes the preparation of the experiments and the argumentative metrics of the experiments; Section 5 analyzes the proposed model through the experimental results and applies the improved model to full-size image inference; and Section 6 concludes the current work.

2. **The YOLOv5 algorithm.**

YOLOv5s uses CSPDraknet as the backbone network for feature extraction, and the input image first goes through the Focus network structure to increase the number of channels, and then goes through down-sampling to deepen the

TABLE 1. Major milestones in object detection research based on the single-stage algorithms and two-stage algorithms since

Researcher	Model	year	Type	Observations
Girshick et al. [5]	R-CNN	2014	two-stage	ROI-driven candidate frame extraction,excessive training and testing time.
Girshick et al. [12]	Fast R-CNN	2014	two-stage	Using ROI pooling to extract features, the training steps are cumbersome and cannot be adapted to large resolution images.
Ren et al. [13]	Faster R-CNN	2015	two-stage	Integration of feature extraction, Proposal extraction, etc. into one network and generation of region proposals using RPN network, the comprehensive performance has been improved and insensitive to the detection of small targets.
Redmon et al. [14]	YOLO	2016	single-stage	Transforms the detection problem into a regression problem and utilizes a separate convolutional neural network for prediction, with fast detection but low accuracy.
Redmon and Farhadi [15]	YOLO9000	2017	single-stage	Improved and proposed a joint training algorithm based on YOLO, but the small target detection capability is insufficient.
Redmon and Farhadi [16]	YOLOv3	2018	single-stage	Using multi-scale feature maps for prediction and optimizing the network structure by residual networks, a good balance of speed and detection accuracy is achieved, but small target detection accuracy is still limited.
Bochkovskiy et al. [17]	YOLOv4	2020	single-stage	Improvements to the input,backbone network, necking network and LOSS function based on YOLOv3 to improve detection accuracy and speed.
Liu et al. [18]	SSD	2016	single-stage	Utilizing multi-scale feature maps to improve detection accuracy, capable of real-time inference,poor detection of small targets.
Li and Zhou [19]	FSSD	2017	single-stage	At the sacrifice of speed, the adoption of a lightweight feature fusion module improves SSD recognition of small targets.
Fu et al. [20]	DSSD	2017	single-stage	The addition of an anti-convolution module on top of SSD has resulted in a significant improvement in the detection of small targets, although the detection speed is significantly slower than that of SSD.

network depth while generating feature layers at different scales for feature extraction. By introducing the residual network, both CSPDarknet and Darknet53 effectively deal with the problem of gradient disappearance that may be caused by network deepening. CSPDarknet, on the other hand, introduces the CSPnet structure, which divides residual block stacking into two parts, one of which is the normal process of stacking residual blocks, and the other part is simply processed and then connected in the form of residual edges to be merged at the end. The strategy of enhancing gradient information difference and lowering gradient reuse through partial transition employing split gradient flow improves model learning ability while reducing model computation [21].

YOLOv5s harvests multi-scale features by building a bidirectional PANet [22] feature pyramid that feeds into the YOLO head portion to detect targets. In the neck feature fusion network, YOLOv5s is divided into two sections, as opposed to the previous YOLOv3 method. The first component up-samples the semantic information from the deep network to the shallow network and fuses the position information within. The second component improves the feature fusion effect by down-sampling to fully utilize the contextual information. Finally, YOLOv5s picks the deep three feature layers to undertake a series of feature fusion operations across the neck network before recognizing the target in the YOLO head component.

3. **Improved YOLOv5s with SAHI inference.**

3.1. **Improved YOLOv5s structure.**

Figure 1 shows the enhanced YOLOv5s. In comparison to the original YOLOv5s, the model's backbone is lightened by a lightweight backbone feature extraction network to reduce model complexity. In addition, to ensure high detection accuracy, the YOLOv5s SPPF module is preserved following the feature extraction network based on ShuffleNetV2. This arrangement allows for multi-scale feature fusion and adaptive size output.The Ghost module is used to modify and replace the convolution module and bottleneck layer in the neck network. Ghost module generates feature maps at a lesser cost in order to ensure detection accuracy while compressing the model further. The Bifpn bi-directional feature pyramid network is inserted into the neck network, and the Bifpn core idea is borrowed to add a cross-scale feature fusion connecting line to boost the effect of feature fusion and improve the model's detection accuracy. The ShuffleNetV2 network, SPPF module, Ghost module, and Bifpn structure are represented in Figure 1 by orange, blue, green, and gold modules, respectively.

3.2. **Lightweight automotive glass defect detection model.**

3.2.1. *Lightweight backbone feature extraction network.* The YOLOv5 model's high complexity increases deployment difficulty, and the backbone network is prone to losing minor target feature information during the down-sampling stage [23]. As a result, this study introduces the lightweight ShuffleNetv2 backbone feature extraction network to replace the original backbone network in order to achieve lightweight backbone network and reduce deployment complexity.

ShuffleNetv2 [24] is a lightweight feature extraction network. Unlike most lightweight networks that use FLOPs to determine network lightness, ShuffleNet fully considers the impact of other factors on detection speed, such as memory access cost and parallelism. ShuffleNetv2 is improved on the basis of ShuffleNetv1 based on the principles of minimizing the memory access cost of channels such as inputs and outputs, careful use of group convolution, avoiding network fragmentation, and reducing element-level operations, and proposes the operation of channel shuffling as shown in Figure 2. By allowing information sharing between two branching feature map channels, the channel shuffling operation

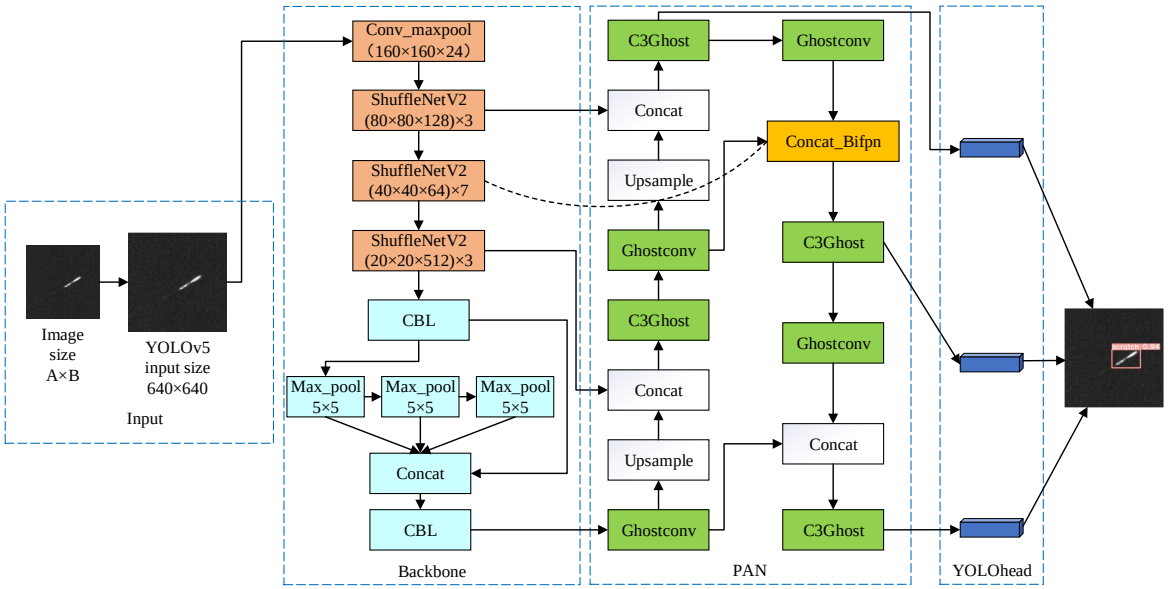


FIGURE 1. Improved YOLOv5 structure.

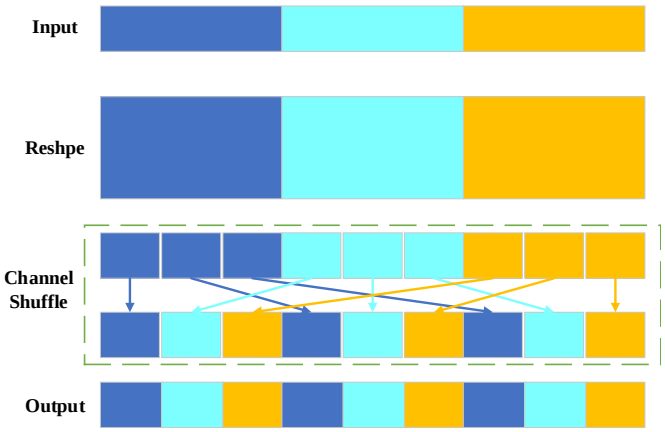
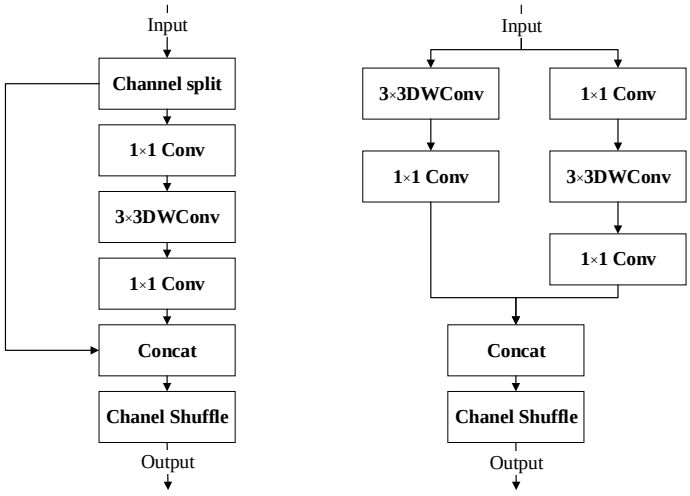


FIGURE 2. Channel shuffle.

improves the learning ability of intergroup feature information while also reducing the network's computational quantity.

Figure 3 depicts the basic structure of the ShuffleNetV2 network, which consists of two unit structures. The basic module divides the input channels evenly by channel split, where one branch is stacked with the other branch that is not operated through three convolutional operations to maintain the same number of channels before and after. Finally, channel shuffling facilitates information sharing between the two branches. The down-sampling module removes the base module's channel split operation, doubling the number of generated channels and increasing feature information. The extracted feature information is made more comprehensive by performing depth-separable convolution and convolution operations in the blank branch of the base module, further improving the model's detection performance.

3.2.2. *Ghost module.* The designed traditional feature extraction method in the neck network part goes through a large number of convolutions to obtain a large amount of featured information, resulting in a large number of redundant features and a large number



(a) ShuffleNetv2 basic unit (b) ShuffleNetv2 sub-sampling unit

FIGURE 3. ShuffleNetv2 module.

of parameters and calculations required to process these features. Many researchers have efficiently decreased the number of parameters through parameter clipping and quantization of the model to solve this situation. However, there are limitations of complex model design and training difficulties [25]. To improve real-time detection, this research includes the Ghostconv and C3Ghost modules in GhostNet [26] in the neck network to effectively maintain feature map information and reduce model computation.

In comparison to the classic convolution module, Ghostconv, as illustrated in Figure 4, can produce redundant features at a lower computational cost by executing constant mapping and linear operations on feature maps generated with only a few convolutions.Ghostconv splits the convolution in two. The first phase performs direct constant mapping on the eigenfeature map to obtain the feature map Y', which is generated from standard convolution and may be written as:

$$Y' = X \times f \tag{1}$$

where X and Y' are input and output, respectively, and f is the corresponding convolution kernel

The second component uses a linear procedure to produce the Ghost feature map y_{ij} from the feature map y_i of each channel corresponding to Y'.

$$y_{ij} = \phi_{ij}(y_i) \tag{2}$$

where $\phi_{i,j}$ is the convolution kernel is 3×3 or 5×5 deep convolution. Finally, the intrinsic feature map Y' and the Ghost feature map y_{ij} obtained in the first and second parts are spliced to obtain the final feature map.

C3Ghost is obtained by replacing the C3 Bottleneck structure with GhostBottleneck, which consists primarily of two Ghostconvs, the first of which increases the number of channels to obtain additional feature information, and the second of which decreases the number of channels to match the network structure in order to realize the shortcut operation of the two Ghost modules. Figure 5 depicts the structure of GhostBottleneck and C3Ghost.

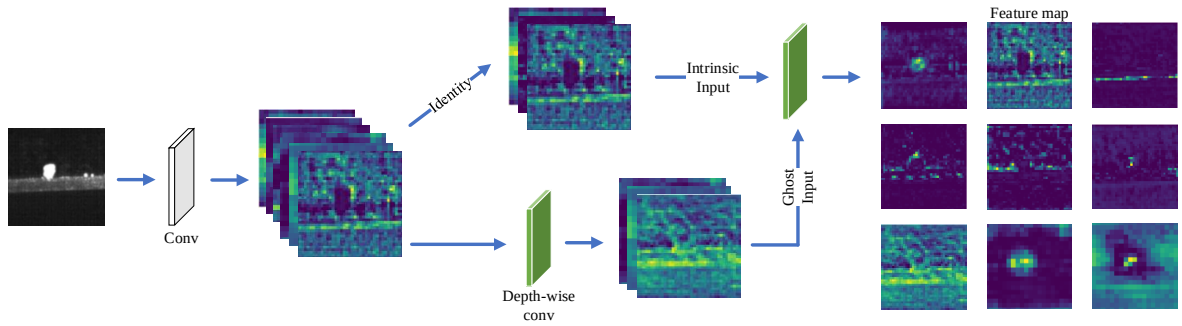


FIGURE 4. Ghostconv structure.

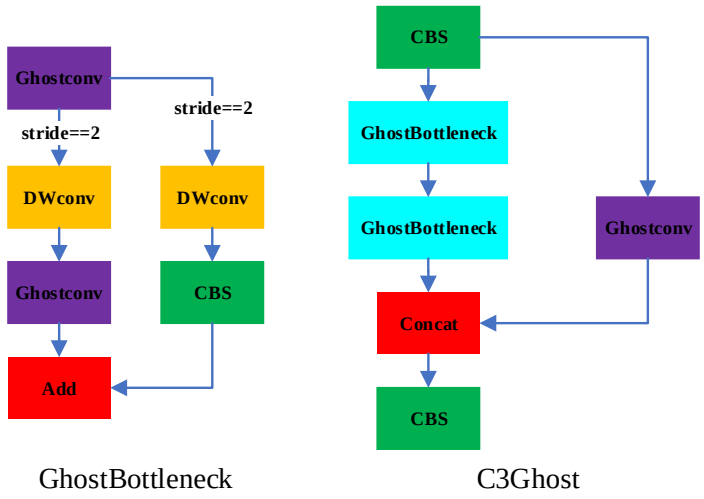


FIGURE 5. Ghost module structure diagram.

3.3. **Bidirectional Feature Pyramid Network.** This research introduces the Bifpn [27] structure to improve the neck feature network's ability to extract enough feature information. The glass defect picture is sent into the ShuffleNetv2 backbone network, and the feature maps of each network layer are shown in Figure 6. The figure shows that from layer 0 to the last layer of the backbone's SPPF module, the image contours of layers 0-3 are still relatively clear, and the semantic information of the image from layer 4 onwards becomes gradually enriched, and this type of feature information can improve the effect of classification for the target [28]. As a result, we do not consider adding the feature layers 0-3 to the Bifpn feature fusion in this paper.

The YOLOv5s model's neck feature fusion network adds a top-down pathway to the FPN of the YOLOv3 model [29]. It ensures that the feature map has rich semantic and spatial information, hence improving the model's classification and localization performance. This paper introduces a weighted bidirectional feature fusion network to replace the PANet in YOLOv5s, which is particularly effective for shallow features of small targets through this higher-level multi-scale feature fusion [30]. As illustrated in Figure 7, Bifpn simplifies the topology of PANet by removing single-input edge nodes that have little influence on the network; and adding an extra edge at the input and output nodes

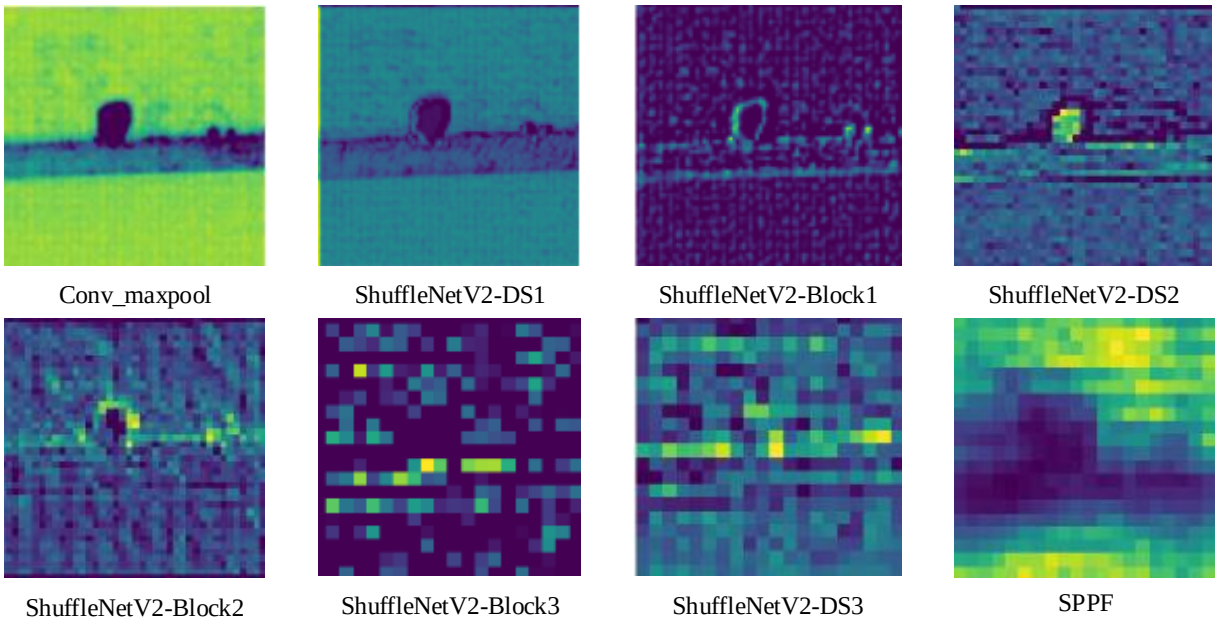


FIGURE 6. Backbone network feature map visualization.

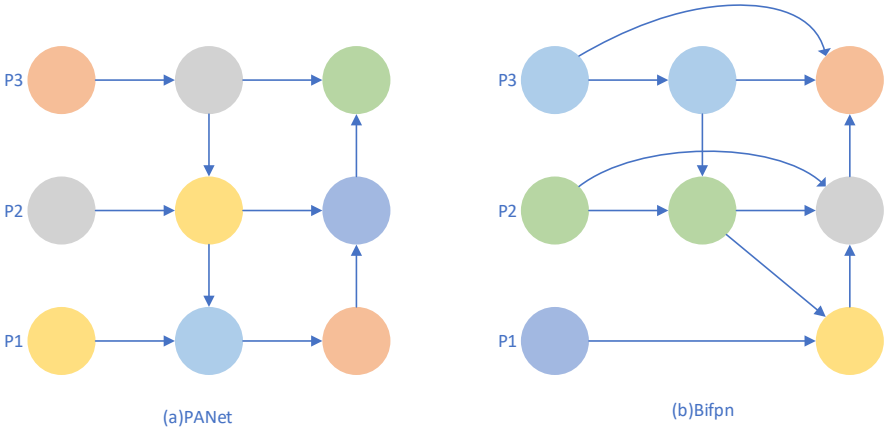


FIGURE 7. Comparison of two neck network structures.

in the same layer for feature fusion, allowing for more feature information without adding too many parameters.

Furthermore, because traditional feature fusion only performs operations like concat on the input feature maps without distinguishing the importance of different feature maps to the final fused feature maps, simply superimposing all of the involved feature maps does not produce the best results. As a result, Bifpn employs a fast normalized fusion approach to discriminate multiple input feature maps to varying degrees of importance, and the feature map computation formula is indicated in Equation (3).

$$\text{Out} = \sum_i \frac{W_i}{\varepsilon + \sum_i W_j} \times I_i \tag{3}$$

where out is the size of the fused feature map, W_i learned parameter used to distinguish the importance of different feature maps, ε=0.0001, and I_i is the size of each input feature map.

3.4. **Automotive Glass Defect Detection Combined with SAHI Inference.** Because the resolution of automobile glass images significantly exceeds the standard input size of the target detection model, training or reasoning directly with the original image size necessitates a very high level of setup for the detection platform's deployment. Scaling the input image will result in significant defect feature loss, affecting the final detection findings. As a result, in order to realize the inference on the original image of large size, this paper introduces the SAHI [31] technology, which is a sliding window based on the slice inference of large-size images. Figure 8 shows how this paper achieved the defect detection principle with the help of SAHI.

The procedure of detecting defects in automotive glass is described in the first half. Because the input image resolution is too high and contains too many small defects, the image is first sliced, and the resulting image containing defects is fed into the enhanced YOLOv5s model of this study for training. Finally, by incorporating the SAHI technique, it is possible to detect defects in full-size auto glass photos without losing feature information about the defects, provided that only the input is enhanced. Furthermore, this detection approach requires no additional GPU memory allocation and achieves full-size detection without increasing deployment costs.

The principle of the SAHI approach is demonstrated in the second part. The SAHI technique detection findings combine the inference results on the full-size image and the results after slicing inference using the SAHI approach. In which the slice inference separates the original image into k sub-images of the same size: P_1,P_2,P_3,...,P_k, and then these sub-images are scaled while retaining the aspect ratio and input into the model for prediction. Meanwhile, the addition of inference to the original image allows SAHI inference to be used for big target detection. Finally, NMS uniformly filters the outcomes of subgraph prediction and original image prediction, and the results are translated to the original image to realize big scale image inference.

4. Experimental Preparation and Method Validation.

4.1. **Experimental platforms.** The experimental platform used in this research is pycharm2021.1, Anaconda, Core(TM) i7-11700K cpu, 16G RAM, RTX3060Ti graphics card and python3.7 software platform. The optimizer used in the experiment is Adam, the Batch size of the model is set to 16, and the number of training rounds is 100.

4.2. **Auto Glass Defects Dataset.** The research object in this work is the four types of defects usually discovered in the manufacturing process of automotive glass, which are bubbles, stains, scratches, and chips. Figure 9 depicts a sample of the four types of automotive glass faults dataset. This paper uses labelimg annotation software to label the xml file after conversion to generate txt files in the annotation of the dataset, as shown in Figure 10, with the software interface and txt label file format. The first number range is 0-3 corresponds to the four types of defects chip, scratch, stain, bubble, respectively, after each two numbers for a group, the first group represents the center coordinates of the target, the second group represents the width and height of the labeling box, the coordinates are normalized to the ratio of the width and height of the image [32].

4.3. **Evaluation index.** Deep learning target detection mainly applies metrics such as Precision, Recall, and mean Average Precision (MAP) for model evaluation.

TP is the samples that are correctly predicted as positive samples, FP is the samples that are incorrectly predicted as positive samples, Precision is a parameter for calculating the proportion of TP samples to the overall proportion of samples that are predicted as

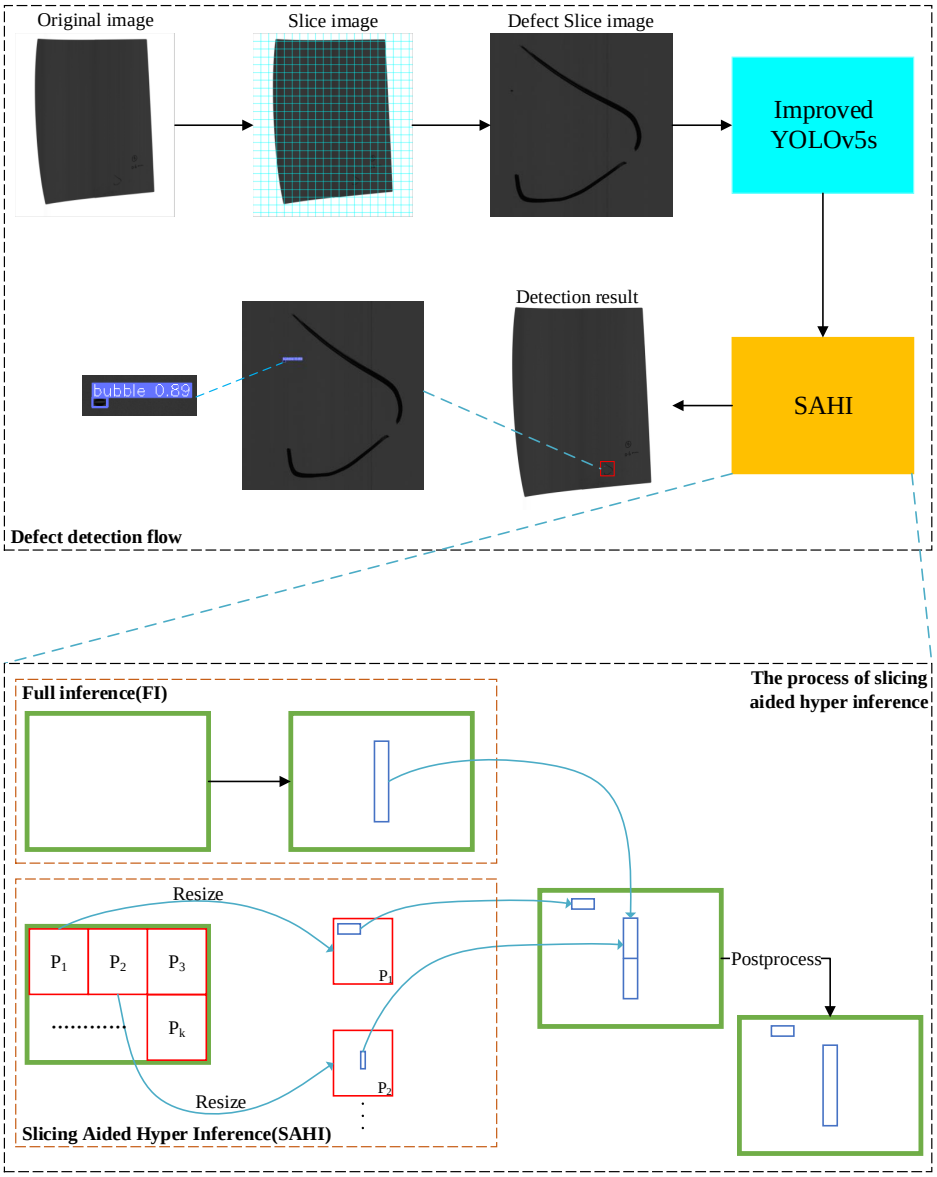


FIGURE 8. Automotive glass defect detection process combined with SAHI.

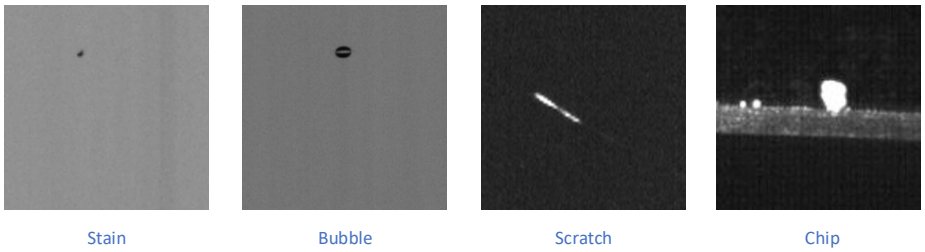


FIGURE 9. Automotive glass Defect datasets.

positive samples, and Recall is the proportion of TP samples to the overall proportion of positive samples, as shown in Equations (4)(5):

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

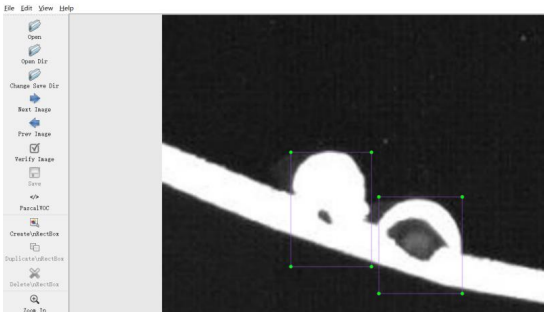


FIGURE 10. labelimg annotation interface and txt file format.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

The MAP is the average of the AP (Average Precision) of all the detected targets, AP is a measure of how good the model's performance is in recognizing the current category. The expressions for the computation of AP and MAP are shown below:

$$\text{AP} = \int_0^1 P(R)\, dR \tag{6}$$

$$\text{MAP} = \frac{1}{n} \times \sum_{i=1}^{n} AP_i \tag{7}$$

5. Model Evaluation and Full-Size Inference.

5.1. **Result and analysis.** This paper selects a number of backbone feature extraction networks for comparative experiments to verify the effectiveness of the backbone network introduced in this paper in the detection of automotive glass defects. Table 2 shows the detection performance of the model under different backbone networks. When compared to previous backbone networks, ShuffleNetV2 extracts better characteristics for auto glass flaws and ensures detection accuracy. It also takes less computation and produces lower weights. As a result, ShuffleNetV2 is chosen as the backbone network in this article.

Comparative tests using self-constructed datasets are done to demonstrate the dependability of the modified strategy provided in this study in order to verify its efficiency. The enhanced detection algorithm's performance is shown in Table 3. ShuffleNetV2, Ghost module, and Bifpn are chosen as independent variables for ablation tests to validate the suggested method's improvement.

As demonstrated in Table3, while using shuffleNetV2 to replace the backbone feature network can reduce model complexity, it will result in a significant decrease in detection accuracy due to the decrease in the receptive field of the backbone network after

TABLE 2. Comparison of backbone networks

Backbone	MAP/%	FLOPs/G	Weight size/MB
CSPDarknet	90.2	15.8	13.7
Resnet18	91.3	35.9	28.7
Resnet34	90.1	66.1	48.0
Resnet50	88.8	77.4	59.5
ShuffleNetV2	91.1	5.8	6.5

Table 3. Results of ablation experiments

Model	MAP/%	FLOPs/G	Weight size/MB
YOLOv5s	90.2	15.8	13.7
YOLOv5s+ ShuffleNetv2(Without SPPF)	84.4	5.6	5.9
YOLOv5s+ ShuffleNetv2(With SPPF)	91.1	5.8	6.5
YOLOv5s+ ShuffleNetv2(With SPPF) +Ghost(Neck)	91.7	3.3	3.8
YOLOv5s+ShuffleNetv2(With SPPF) +Ghost(Neck)+Bifpn	92.3	3.3	3.8

Table 4. Bifpn ablation experiment

Module	Position	Map/%	FLOPs/G	Weight size/MB
Bifpn	40×40+20×20	92.3	3.3	3.8
	40×40	93.0	3.3	3.9
	20×20	91.7	3.3	3.8

lightweighting, resulting in the model's poor recognition accuracy for large-size defects. To address this issue, the SPPF module is added to strengthen the effect of feature fusion, and the addition of the SPPF module to the lightweight backbone network increases the model's receptive field to some extent, making the model more capable of extracting multi-scale features. As a result, ShuffleNetv2 (With SPPF) is finally selected as the backbone network for the next experiments. In the following trials, the neck network is recreated using the Ghost module to accomplish further model compression without sacrificing detection accuracy. Finally, cross-scale feature fusion is carried out by incorporating the Bifpn structure to assist the model in better capturing multi-scale information and improving the model's detection accuracy. The experimental results reveal that, when compared to the original YOLOv5s, the improved model MAP is 2.1% higher, and the number of model parameters and volume are dramatically decreased to 20.9% and 27.7% of the original, respectively. This can help to increase the model's computational efficiency, reduce memory utilization, and speed up model inference.

This paper conducts ablation experiments on the position of Bifpn for weighted fusion of feature maps in order to further investigate the effect on model detection performance after the Bifpn structure is added to different positions of the neck network, and the experimental results are shown in Table 4. Where 40×40 and 20×20 represent the size of the feature map.

Tables 3 and 4 show the results of the experiments. The detection accuracy of the model is somewhat enhanced after the introduction of Shufflenetv2, a lightweight backbone neural network, due to the efficient structural design and feature reuse, and the Flops value and weight size are drastically lowered by 63.3% and 52.55%, respectively. Furthermore, because of the Ghost module's unique method of producing redundant feature maps, the model is further compressed without losing detection accuracy, and the FLOPs value and weight size are decreased by 43.1% and 41.54%, respectively. Finally, by including the Bifpn structure in the neck network, more feature maps of different sizes can be feature fused to obtain more feature information and improve the model's detection accuracy. And, after further investigating the efficiency of the Bifpn structure, the ablation experiment determines that the Bifpn structure is introduced at the feature map size of 40×40. As a result, the improvement strategy of this paper is Shufflenetv2+Ghost module(Neck)+Bifpn(40×40), and when compared to the original YOLOv5s model, the

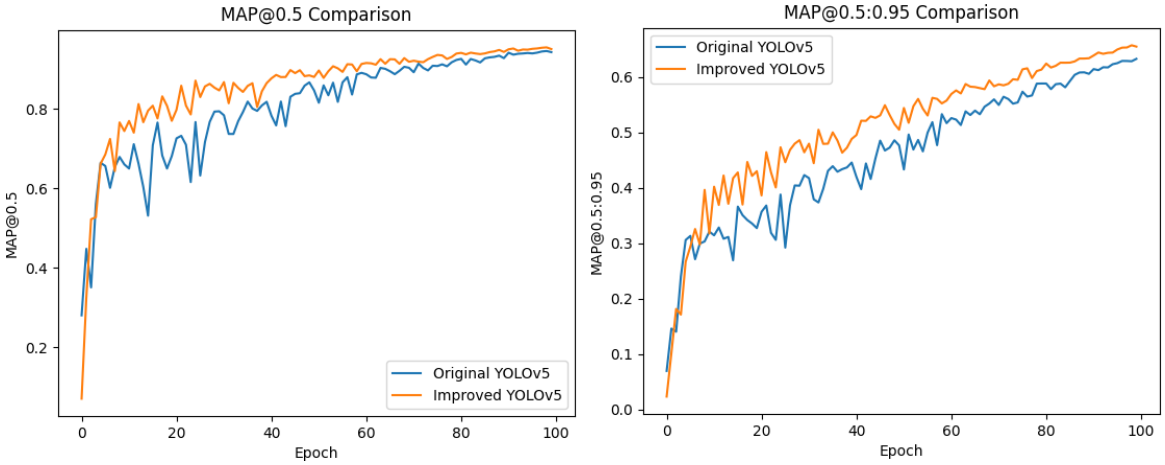


FIGURE 11. The comparation of MAP.

TABLE 5. Comparison of the detection results of different models

Model	MAP@0.5/%	FLOPs/G	Weight size/MB
YOLOv3	84.9	66.1	235.1
YOLOv4	75.6	60.3	244.4
YOLOv5s	90.2	155.3	34.3
YOLOv7	90.5	104.8	142.3
Ours	93.0	3.3	3.9

MAP value of the lightweight YOLOv5s model proposed in this paper improves by 2.8%, while the FLOPs value and weight size decrease by 79.11% and 71.53%, respectively.

Figure 11 compares MAP before and after improvement, based on the average detection accuracy of all categories when the IOU threshold is 0.5 and 0.5:0.95. The blue line represents the YOLOv5s curve prior to improvement, and the orange line represents the curve of the lightweight model proposed in this paper. As seen in the figure, the improved model in this paper has increased overall detection performance.

Figure 12 depicts the detection of several defects. The figure shows that the improved YOLOv5s model proposed in this paper has more advantages for both large and small target detection than the original YOLOv5s model, and it can also detect neighboring defects better in the case of neighboring detection targets and certain overlapping regions of the detection frame.

Finally, Table 5 shows a comparison with the current mainstream target detection approach. The model in this article provides a greater detection accuracy for vehicle glass faults. Importantly, the model proposed in this paper has significant advantages in terms of weight size and parameter computation, which can help to reduce the model's deployment difficulty, memory and hardware demand, and allow the model to be deployed on platforms with lower computing power.

In conclusion, the lightweight YOLOv5s model proposed in this paper is better suited for automotive glass defect detection applications than the original YOLOv5.

5.2. **Analysis of full-size image inference results.** Since the resolution of the full-size auto glass image in this paper is around 16384*18000, it is directly fed into the model for detection, and if the input is 640*640 or scaled down according to the model, a large

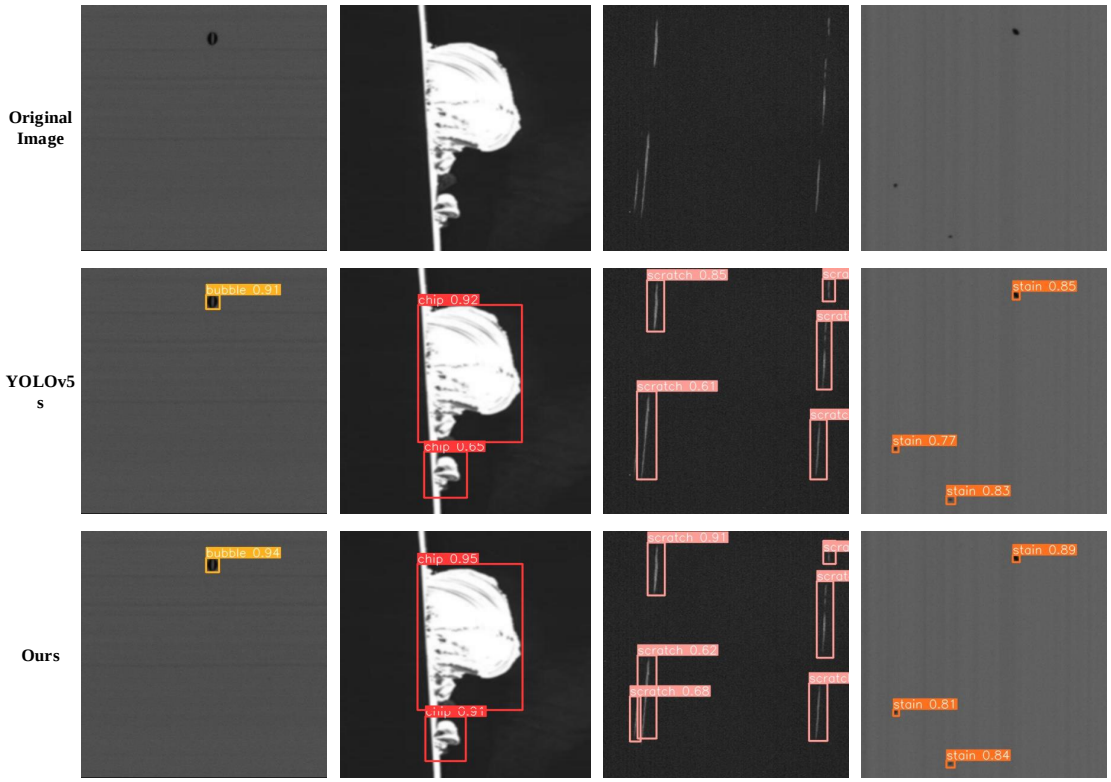


FIGURE 12. Comparison of detection results.

TABLE 6. Comparison of the detection results of different models

Model	Time-consuming (cpu)	Number of slice images under detect	Single sub-image inference time
YOLOv5s	89s	336	264.9ms
Ours	53s		157.7ms

amount of feature information is lost and detecting defects on the image becomes difficult. Direct detection without scaling, on the other hand, is extremely taxing on the GPU and prohibitively expensive to accomplish. As a result, this study solves this issue by combining SAHI techniques in order to achieve full-size inference. Figure 13 depicts the results of full-size image defect identification for automotive glass, with some typical defects detection results marked, where white denotes proper detection, red denotes misdetection, and green denotes missing detection. The identification of bubbles, stains, and chips is good, but there is still a leakage detection phenomena, which is due to the fact that stains belong to a small target, and extracting defect features is more difficult. The scratches are misdetected because the model's detection accuracy is insufficient, as well as the presence of a large amount of dust on the surface of the automotive glass in the industrial site, and the characteristics of the dust under coaxial light irradiation are similar to those of the shallow scratches, affecting the accuracy of the full-size inference. The enhanced model's detection speed in executing full-size inference is decreased by more than 40% when compared to the original YOLOv5s, as shown in Table 6.

6. **Conclusion.** A lightweight automobile glass defect detection model based on YOLOv5s is proposed in this study, which is compressed by a lightweight backbone network and lightweight module, and feature fusion is improved in the neck network to improve the

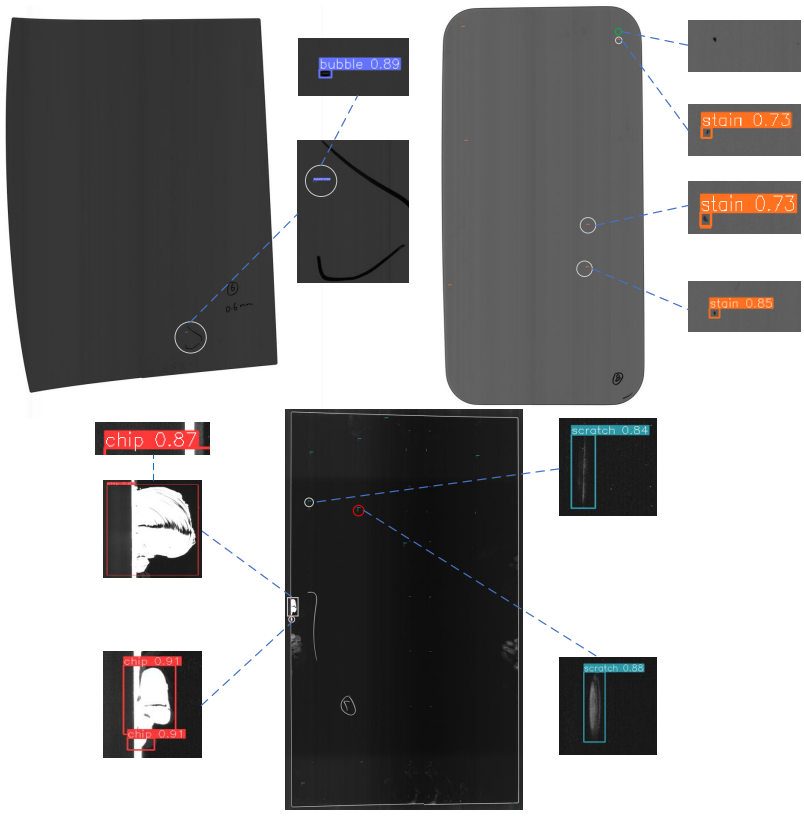


FIGURE 13. Inference on full size image.

model's detection performance. When compared to the original YOLOv5s, the experimental findings demonstrate that the model provided in this work improves 2.8% in Map, reduces 79.11% in parameter volume computation, and decrease 71.53% in weight size. Furthermore, by utilizing SAHI technology to achieve full-size automobile glass inspection, the solution incurs no additional deployment costs, no excessive GPU burden, and no increased memory requirement. Overall, the automobile glass inspection described in this work is a reference solution for use in industrial settings. Future work will concentrate on improving model accuracy and removing dust interference.

REFERENCES

[1] F. Sarı and A. B. Ulaş, "Deep learning application in detecting glass defects with color space conversion and adaptive histogram equalization," *Traitement du Signal*, vol. 39, no. 2, pp. 731–736, 2022.

[2] Z.-C. Yuan, Z.-T. Zhang, H. Su, L. Zhang, F. Shen, and F. Zhang, "Vision-based defect detection for mobile phone cover glass using deep neural networks," *International Journal of Precision Engineering and Manufacturing*, vol. 19, pp. 801–810, 2018.

[3] J. Jiang, P. Cao, Z. Lu, W. Lou, and Y. Yang, "Surface defect detection for mobile phone back glass based on symmetric convolutional neural network deep learning," *Applied Sciences*, vol. 10, no. 10, p. 3621, 2020.

[4] K. Wang, Z. Chen, X. Dang, X. Fan, X. Han, C.-M. Chen, W. Ding, S.-M. Yiu, and J. Weng, "Uncovering hidden vulnerabilities in convolutional neural networks through graph-based adversarial robustness evaluation," *Pattern Recognition*, vol. 143, p. 109745, 2023.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[6] T.-T. Nguyen, T.-D. Nguyen, and V.-T. Nguyen, "An optimizing pulse coupled neural network based on golden eagle optimizer for automatic image segmentation," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 13, no. 3, pp. 155–164, 2022.

[7] D.-T. Pham and D.-T.-T. Hoang, "An improved whale optimization algorithm for optimal multi-threshold image segmentation," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 14, no. 2, p. 41–53, 2023.

[8] T.-T. Nguyen, T.-D. Nguyen, T.-G. Ngo, and V.-T. Nguyen, "An optimal thresholds for segmenting medical images using improved swarm algorithm," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 13, no. 1, pp. 12–21, 2022.

[9] T.-T. Nguyen, H.-J. Wang, T.-K. Dao, J.-S. Pan, T.-G. Ngo, and J. Yu, "A scheme of color image multithreshold segmentation based on improved moth-flame algorithm," *IEEE Access*, vol. 8, pp. 174 142–174 159, 2020.

[10] E. K. Wang, C. Chen, M. S. Hossain, M. Ghulam, S. Kumar, and S. Kumari, "Transfer reinforcement learning-based road object detection in next generation iot domain," *Comput. Networks*, vol. 193, p. 108078, 2021.

[11] Y. Ma, Y. Peng, and T.-Y. Wu, "Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning," *Journal of Intelligent and Fuzzy Systems*, vol. 43, no. 2, pp. 2121–2133, 2022.

[12] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[15] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7263–7271.

[16] ——, "Yolov3: An incremental improvement," *ArXiv*, vol. abs/1804.02767, 2018.

[17] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[19] Z. Li and F. Zhou, "Fssd: feature fusion single shot multibox detector," *arXiv preprint arXiv:1712.00960*, 2017.

[20] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[21] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "Cspnet: A new backbone that can enhance learning capability of cnn," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 390–391.

[22] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.

[23] Y. He, J. Tian, Z. Zhang, Q. Wang, and P. Zhao, "Lightweight research of yolov5 target detection," *Computer Engineering and Applications*, vol. 59, no. 1, pp. 92–99, 2023.

[24] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 116–131.

[25] G. Tu, J. Qin, and N. N. Xiong, "Algorithm of computer mainboard quality detection for real-time based on qd-yolo," *Electronics*, vol. 11, no. 15, p. 2424, 2022.

[26] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghostnet: More features from cheap operations," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1577–1586.

[27] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 781–10 790.

[28] P. Yu, Y. Lin, Y. Lai, S. Cheng, and P. Lin, "Fusion of bifpn and yolov5s for dense log end face detection," *Journal of Forestry Engineering*, vol. 8, no. 1, p. 9, 2023.

[29] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.

[30] Z. Zhang, M. Luo, S. Guo, G. Liu, S. Li, and Y. Zhang, "Cherry fruit detection method in natural scene based on improved yolo v5," *Journal of Agricultural Machinery*, vol. 53, no. S1, pp. 232–240, 2022.

[31] F. C. Akyon, S. O. Altinuc, and A. Temizel, "Slicing aided hyper inference and fine-tuning for small object detection," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 966–970.

[32] W. Z. Lulu Zhao, Xueying Wang and M. Zhang, "Research on vehicle target detection technology based on yolov5s fusion senet," *Journal of Graphics*, vol. 43, no. 5, p. 776, 2022.