Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.048998

ARTICLE

Unmanned Aerial Vehicles General Aerial Person-Vehicle Recognition Based on Improved YOLOv8s Algorithm

Zhijian Liu*

School of Electrical Engineering and Electronic Information, Xihua University, Chengdu, 610036, China

*Corresponding Author: Zhijian Liu. Email: liuzj984296576@163.com

Received: 24 December 2023 Accepted: 22 January 2024 Published: 26 March 2024

ABSTRACT

Considering the variations in imaging sizes of the unmanned aerial vehicles (UAV) at different aerial photography heights, as well as the influence of factors such as light and weather, which can result in missed detection and false detection of the model, this paper presents a comprehensive detection model based on the improved lightweight You Only Look Once version 8s (YOLOv8s) algorithm used in natural light and infrared scenes (L_YOLO). The algorithm proposes a special feature pyramid network (SFPN) structure and substitutes most of the neck feature extraction module with the Special deformable convolution feature extraction module (SDCN). Moreover, the model undergoes pruning to eliminate redundant channels. Finally, the non-maximum suppression algorithm of intersection-union ratio based on minimum point distance (MPDIOU_NMS) algorithm has been integrated to eliminate redundant detection boxes, and a comprehensive validation has been conducted using the infrared aerial dataset and the Visdrone2019 dataset. The comprehensive experimental results demonstrate that when the number of parameters and floating-point operations is reduced by 30% and 20%, respectively, there is a 1.2% increase in mean average precision at a threshold of 0.5 (mAP(0.5)) and a 4.8% increase in mAP(0.5:0.95) on the infrared dataset. Finally, the mAP on the Visdrone2019 dataset has experienced an average increase of 12.4%. The accuracy and recall rates have seen respective increases of 9.2% and 3.6%.

KEYWORDS

YOLOv8s; SPFN; SDCN; pruning; MPDIOU_NMS

1 Introduction

With the rapid advancement of drone technology and computer vision, drones have found extensive applications in law enforcement, traffic control, surveillance, and reconnaissance. In particular, unmanned aerial vehicles (UAVs) equipped with infrared imaging cameras have the potential to mitigate the effects of weather, lighting, and other environmental factors on UAV imaging [1]. Nevertheless, variations in UAV altitude result in differences in imaging dimensions. The absence of contextual semantic information in the infrared image presents challenges in differentiating the foreground and background of the target. Additionally, the absence of semantic information in certain detection targets hinders accurate target identification following the deployment of the UAV

[2]. Hence, the primary objective of this paper is to streamline the model's parameter quantity and complexity, with the aim of enhancing the model's detection accuracy.

The conventional detection algorithm primarily depends on manual screening of image features for training, which makes the process cumbersome and results in poor model robustness [3]. Nevertheless, deep learning algorithms have the potential to compensate for this drawback. Currently, deep learning-based detection algorithms are widely utilized, for instance, in the precise identification of cattle in animal husbandry [4] and the detection of soybean pests in intricate environments [5]. Numerous classical target detection algorithms rooted in deep learning exist, including multi-stage algorithms (e.g., Detection Transformer with YOLOv2 [6], Mask convolutional neural networks [7]) and single-stage algorithms (e.g., You Only Look Once [8], Single shot multibox detector (SSD) [9]). In this paper, the YOLOv8 algorithm has been chosen as the foundational algorithm. The algorithm considers both accuracy and detection speed, and its overall performance surpasses that of other algorithms [10]. Extracting semantic features of small targets in dense target scenes while maintaining a streamlined model has consistently posed a research challenge. Various solutions have previously been proposed for different application scenarios [11]. For instance, Zhao et al. [12] introduced an enhanced YOLOv7 model designed to tackle the challenges related to ship detection and recognition tasks, including irregular ship shapes and size variations. Wang et al. [13] proposed a traffic sign detection algorithm utilizing residual network, to minimize missed and false detections of traffic signs in complex environment conditions. Chen et al. [14] introduced a YOLO algorithm-based UAV for the purpose of detecting the poles and quantifying the distribution network, to improve the efficiency of post-disaster distribution network repairs.

The paper is structured as follows: Section 2 introduces the current research status of UAV aerial photography, outlines the existing problems, and presents the proposed methods. Section 3 primarily presents a detailed introduction to our proposed Synthetic Fusion Pyramid Network (SFPN) structure, Structural Deep Clustering Network (SDCN) module, pruning, and MPDIOU_NMS algorithm. It also provides a brief overview of the evaluation index and experimental parameter setting. In Section 4, a brief analysis of the two datasets is presented, followed by a detailed examination of the ablation experiments, a comparison of different algorithms, and a comparison of different datasets. The fifth section provides a summary of the paper's findings.

2 Related Work

The conventional detection algorithm relies on the manual extraction of target semantic feature information, followed by transmitting the extracted semantic information to the algorithm network to produce the detection results [15]. However, these detection algorithms are unsuited for complex scenes such as small targets and dense target types. On the one hand, the dataset contains numerous detection targets, while on the other hand, manual feature screening may result in insufficient feature extraction [16]. However, the target detection algorithm based on deep learning can automatically filter and extract image features, requiring only the processed images to be directly input into the network. Therefore, this feature enables it to process large-scale data sets and accommodate more intricate detection tasks.

To address the inadequate network feature extraction capability issue, Cao et al. [17] introduced an algorithm for detecting small targets using an improved YOLOv5s model for UAVs, resulting in a 9.2% increase in mAP(0.5). It is important to note that the analysis needs to fully account for the impact of factors such as inadequate lighting. Hui et al. [18] introduced a small target detection algorithm for UAV remote sensing images, which relies on an enhanced Shifted Window Transformer and a

class-weighted classification decoupling head. However, the algorithm's parameter size is excessive for deployment on edge devices. Ali et al. [19] incorporated the Kalman filter into the YOLO algorithm to enable the detection and tracking of vehicles by drones. Lu [20] introduced a hybrid CNN-Transformer model for detecting targets in UAV images, utilizing the Cross-Shaped Window Transformer. However, the model's parameter count approaches 70 M. To address the issue of model lightweight, Qian et al. [21] introduced a lightweight YOLO feature fusion network aiming at multi-scale defect detection. The network demonstrated promising results across three different datasets. Zhu et al. [22] introduced a lightweight and efficient network target detection network for remote sensing, capable of achieving an inference speed of 487 frames per second. Zou et al. [23] proposed a method for lightweight target detection in a coal seam tracking system, utilizing knowledge distillation and model pruning. The proposed approach achieves a Central Processing Unit (CPU) processing speed of 45 frames per second.

In summary, this paper primarily addresses the lightweight nature of the model and its limited feature extraction capability. Consequently, this paper has implemented four enhancements to the YOLOv8s algorithm. The paper first proposed an SFPN structure to facilitate the comprehensive exchange of semantic information across each layer. Secondly, the Structural Deep Clustering Network (SDCN) module is proposed to improve the model's feature extraction capability. Subsequently, the model undergoes further pruning to remove redundant channels. Finally, the MPDIOU_NMS algorithm has been incorporated to mitigate the presence of redundant detection boxes.

3 Materials and Methods

3.1 YOLOv8 Network Structure

The YOLOv8 algorithm framework is depicted in Fig. 1. The network comprises four components: data preprocessing, backbone network, neck network, and detection head. First, the detection image undergoes preprocessing through mosaic data, followed by transmission of the processed image to the YOLOv8 backbone network. Subsequently, the fused features are sent to the neck network for further feature extraction. Ultimately, the detection head identifies and distinguishes the target based on the features extracted from the neck network [24].

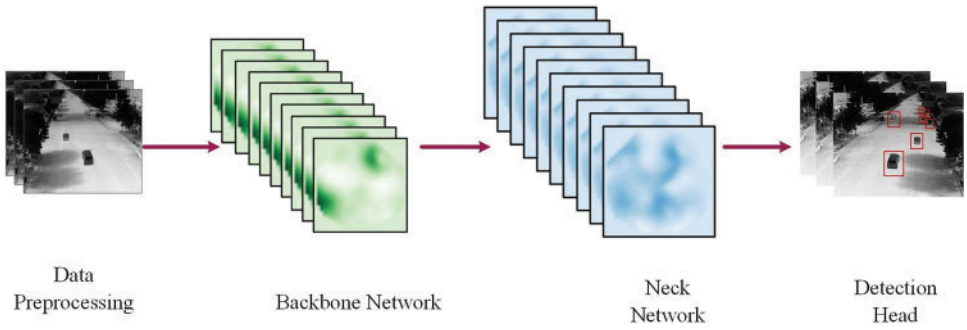


Figure 1: YOLOv8 network structure

3.2 L_YOLO Network Architecture

The workflow comparison between the YOLOv8 algorithm and the L_YOLO algorithm is depicted in Fig. 2. Figs. 2a and 2b depict the flow charts of the YOLOv8 algorithm and the L_YOLO algorithm, respectively. Compared with the YOLOv8 algorithm, the L_YOLO algorithm primarily

enhances three aspects, as depicted in Fig. 2b. This paper first improves the network structure. The light blue rectangle represents the bloated network after pruning, and the light yellow indicates that the MPDIOU_NMS algorithm is used to assist in verification or testing during the model inference stage. Compared to the YOLOv8s algorithm, the L_YOLO model demonstrates better detection performance with approximately 30% fewer parameters.

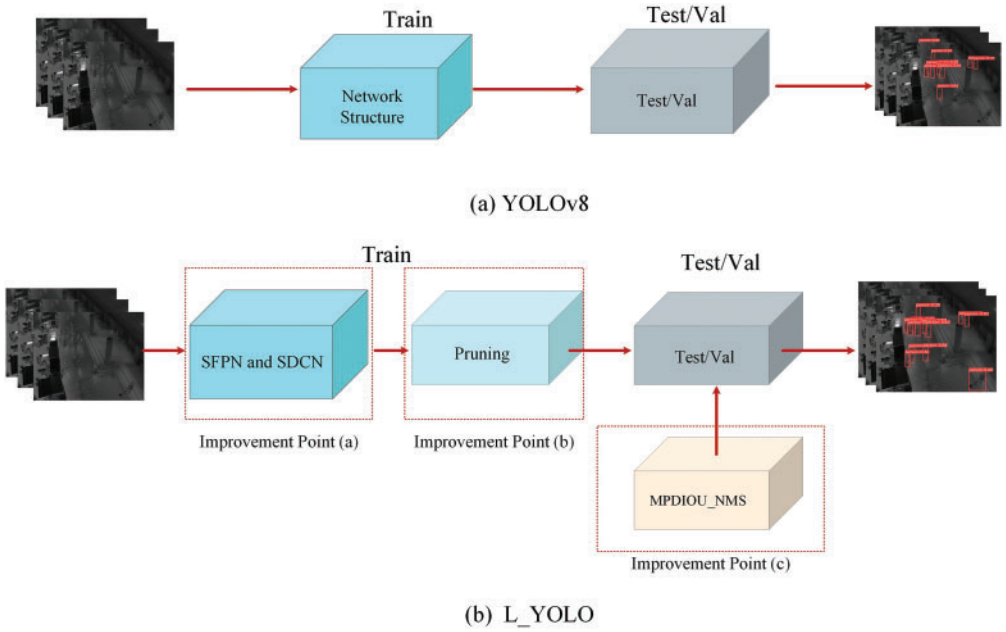


Figure 2: Comparison diagram of YOLOv8 algorithm and L_YOLO algorithm workflow

The specific improvement is depicted in Fig. 3, the YOLOv8s network structure has undergone improvements in four key aspects: (a) The original YOLOv8s network's feature pyramid network does not fully facilitate the exchange of semantic information between adjacent layers. As a solution, the SFPN structure is proposed to comprehensively exchange the semantic features of different layers. (b) An SDCN feature extraction module is proposed. (c) The trained model undergoes channel pruning to reduce redundant channels and further decrease its overall weight. (d) A MPDIOU_NMS algorithm is proposed.

3.2.1 SFPN Structure

Considering that the texture characteristics of small targets may diminish or vanish as the network layers increase, this study proposes a SFPN framework derived from the original FPN (Fig. 4a) framework [25]. The objective is to facilitate the complete exchange of semantic features across various detection layers and address the issue of semantic feature degradation resulting from the increased depth of layers. The primary improvement is depicted in Fig. 4b, where the characteristics of each layer of the trunk network are incorporated into the 12th feature extraction layer, aiming to comprehensively integrate semantic information. Furthermore, the characteristics of layer 4 and layer 6 have been consolidated into layer 21 and layer 24, respectively, thereby enhancing the semantic information within the network.

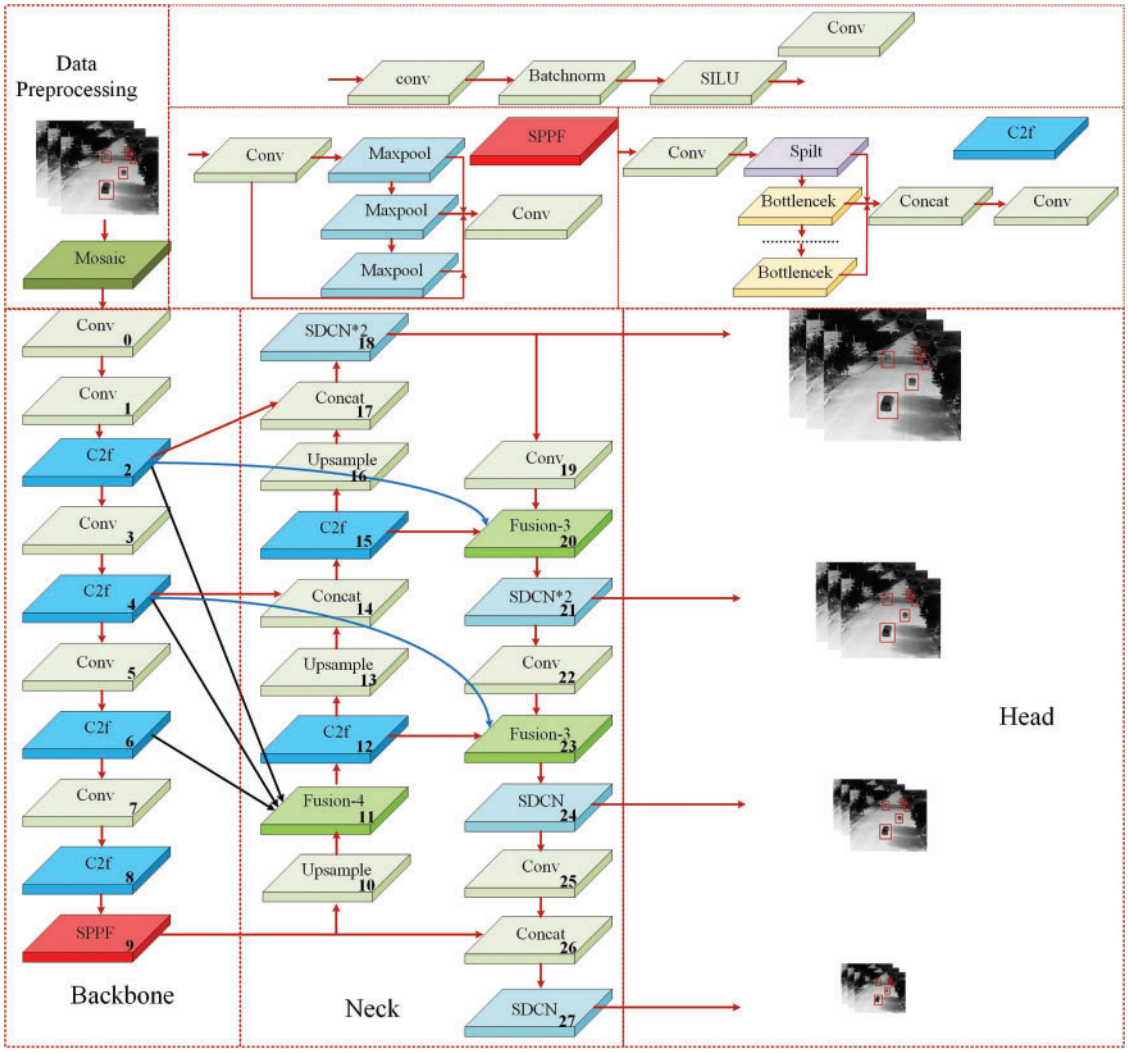


Figure 3: Framework of L_YOLO algorithm

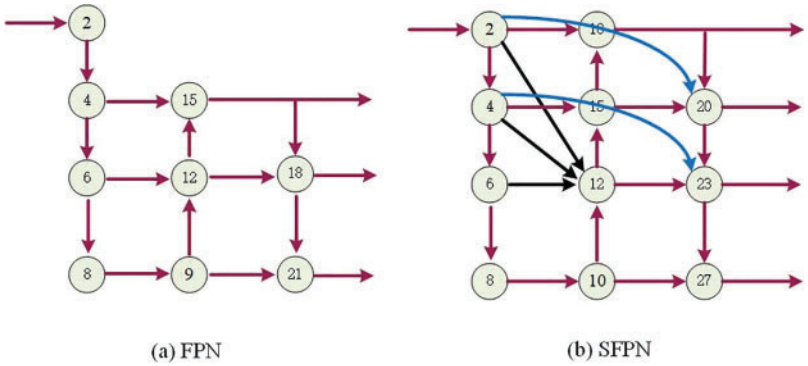


Figure 4: SFPN

3.2.2 SDCN Module

The original YOLOv8s has limited feature extraction capability for dense and small targets. This paper introduces the SDCN module to replace the majority of feature extraction modules in the neck network. The specific replacement is illustrated in Fig. 5. The module exhibits further improved between the convolution feature extraction module version2 (DCNV2) module [26] and the squeeze Aggregated Excitation layer (SaElayer) [27]. The DCNV2 module is designed to effectively detect dense targets. The module primarily enhances the network's receptive field by stacking multiple deformable convolution modules and incorporating additional skip connections to achieve a more comprehensive structure of gradient flow.

Excessive incorporation of the DCNV2 module may result in prolonged training time, while insufficient incorporation may lead to inadequate feature extraction capability of the model. Consequently, this study introduces a SaElayer module to the output of DCNV2, and the specific structure of the SDCN module is illustrated in Fig. 5. The SaElayer module combines the squeeze excitation network module and the dense layer. It also introduces multi-branch fully connected layers with different branch sizes to enhance the network's ability to capture global knowledge. Consequently, incorporating the SaElayer module can enhance the network's focus on valuable semantic information and optimize network bandwidth to reduce model training time.

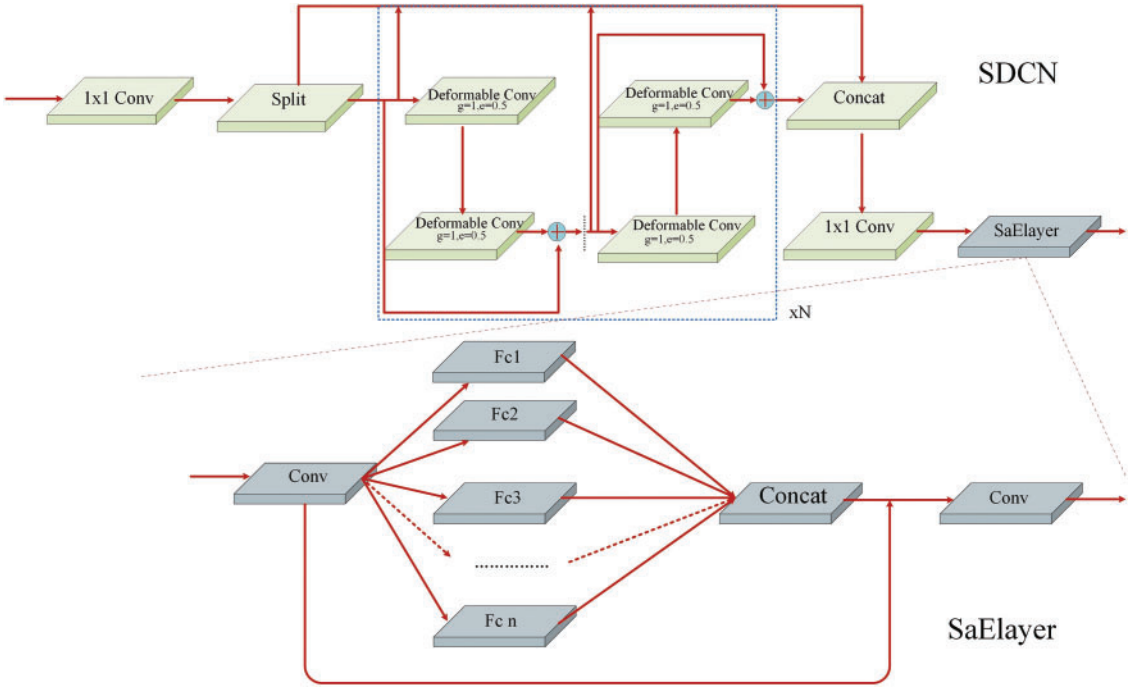


Figure 5: SDCN module

3.2.3 Channel Pruning

Owing to the constraints of mobile device capabilities, models with excessive parameters cannot be accommodated, necessitating the pruning of the trained YOLOv8s model [28]. Given the low mAP of the basic models in both datasets, the approach taken in this study is to reduce model parameters and complexity while preserving accuracy. In this paper, a Slim pruning method is employed [29], and its operational principle is depicted in Fig. 6. This method involves three steps to simplify the initial network. Firstly, the network must undergo sparse training, followed by model pruning, and ultimately, the pruned model is restored. If the model is multi-channel, the algorithm also generates supplementary branches.

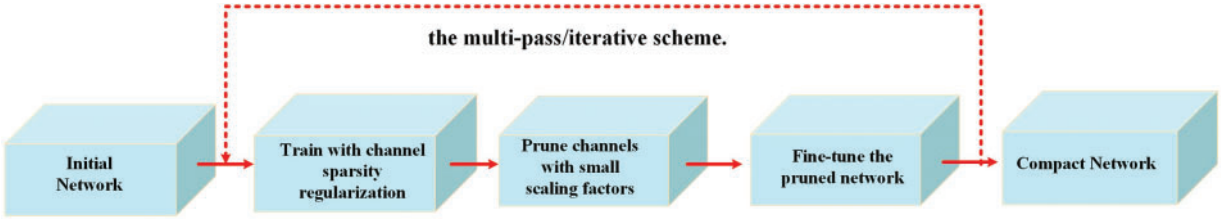


Figure 6: Pruning flow chart

In this paper, the batch normalization layer (BN) scaling factor in the convolution is initially utilized as the channel scaling factor γ of the pruning. It is subsequently multiplied by the output of the channel. Secondly, the network weights and scaling factors are jointly trained, and sparse regularization is applied to the scaling factors. Following the application of channel-level sparse-induced regularization during training, a model is derived wherein numerous scaling factors approach zero. Subsequently, it is possible to eliminate channels with scaling factors close to zero by removing all inbound and outbound connections, as well as the associated weights. Ultimately, the pruned network undergoes fine-tuning [29]. The details are shown in Fig. 7.

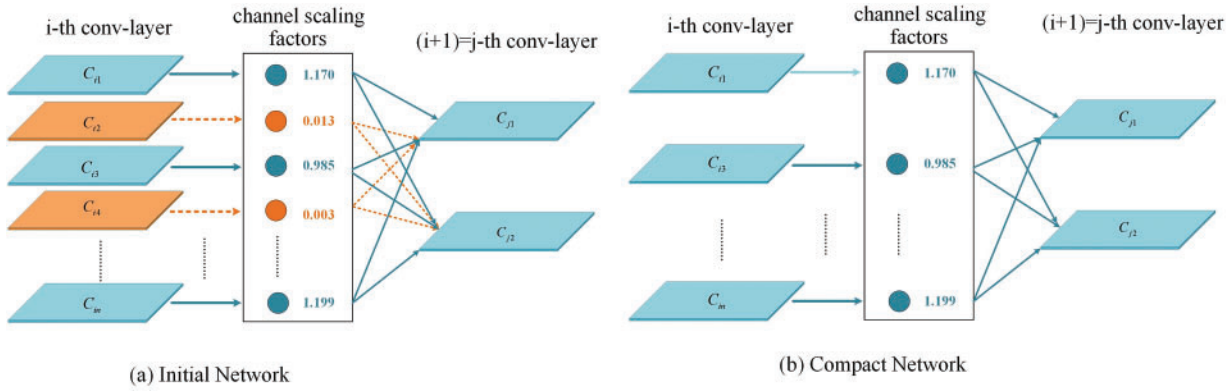


Figure 7: Pruning comparison diagram

As depicted in Fig. 8, the diagram compares the pruning channels discussed in this paper. The x-axis denotes the name of each layer in the network, while the y-axis represents the parameter quantity. The yellow bars indicate the parameter quantity of the primary model channel, whereas the red bars represent the parameter quantity after pruning. To facilitate the successful completion of pruning, it is necessary to bypass network layers that are not amenable to pruning. The DCNv2 layer in the

SDCN feature extraction module is skipped in this paper, as well as specific convolution layers and Distribution Focal Loss layers in the detection head layer [30]. Given that the primary objective of this paper is not to achieve absolute lightweight, it is imperative also to consider the trade-off between precision and model size. The pruned model and the basic model depicted in the figure are not expected to exhibit significant differences in appearance.

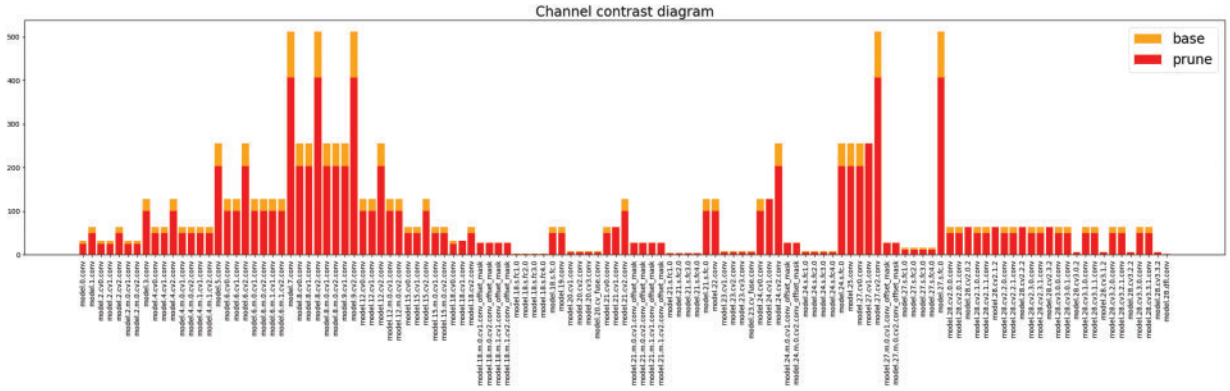


Figure 8: Pruning channel comparison diagram

3.2.4 MPDIOU_NMS Algorithm

Since both datasets fall within the dense target detection category, redundant detection boxes can impact the model's final target assessment. Given that the non maximum suppression algorithm (NMS) of the YOLOv8s algorithm may result in information loss, and the traditional Soft-NMS algorithm [31] could lead to the suppression of detection boxes due to discriminant errors. Consequently, this paper incorporates the MPDIOU_NMS algorithm in the model verification and testing phase to eliminate redundant detection boxes.

The MPDIOU loss function [32] is depicted in Eq. (1). Boxes A and B represent two detection areas, respectively. The coordinates $\left(x_1^A, y_1^A\right)$ and $\left(x_2^A, y_2^A\right)$ denote the position of the upper left corner and the lower right corner of the detection box for A, respectively. The coordinates $\left(x_1^B, y_1^B\right)$ and $\left(x_2^B, y_2^B\right)$ denote the position of the upper left corner and the lower right corner of the B detection box, respectively.

$$MPDIOU = \frac{A \cap B}{A \cup B} - \frac{d_1^2 + d_2^2}{w^2 + h^2} \tag{1}$$

$$d_1^2 = \left(x_1^B - x_1^A\right)^2 + \left(y_1^B - y_1^A\right)^2 \tag{2}$$

$$d_2^2 = \left(x_2^B - x_2^A\right)^2 + \left(y_2^B - y_2^A\right)^2 \tag{3}$$

This paper introduces a novel MPDIOU-NMS algorithm, which is founded on the MPDIOU loss function. The formula is presented in Eq. (4). In contrast to the conventional soft-nms algorithm, this algorithm demonstrates enhanced capability in effectively suppressing redundant detection boxes, thereby improving the model's classification and recognition performance.

$$S_i = \begin{cases} S_i, & MPDIOU\,(M, b_i) < N_t \\ S_i\,(1 - MPDIOU\,(M, b_i)) & MPDIOU\,(M, b_i) \geq N_t \end{cases} \tag{4}$$

$$S_i = S_i e^{-\frac{MPDIOU\left(M, b_i\right)}{\sigma}}, \forall b_i \notin D \tag{5}$$

3.3 Experimental Parameters Setting

In the training of the YOLOv8 algorithm, this study employs the stochastic gradient descent (SGD) algorithm to optimize the loss function. In this study, the batch size was configured as 32, and the number of threads was set to 16. To achieve the optimal model, 220 training iterations are required.

Furthermore, the pruning parameter reg is established at 0.0005, and the sparse training count is specified as 500. In light of the model's accuracy post-pruning, this study specifies a 'speed _ up' value of 1.5, does not activate the global pruning branch, and conducts 300 rounds of pruning recovery training. The computer configuration utilized in the experiment is presented in Table 1.

Table 1: Computer configuration

Platform	Configuration information
System	Ubuntu 20.04
GPU	NVIDIA GeForce RTX A5000(24G)
CPU	15 vCPU AMD EPYC 7543 32-Core Processor
Language	Python 3.8.0
GPU calculate platform	CUDA 11.8
Deep learning framework	Pytorch 2.0.0

3.4 Evaluation Indicators

Precision, Recall, and mAP serve as critical metrics for evaluating the accuracy of a network.

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

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

$$AP = \int_0^1 p(r)dr \tag{8}$$

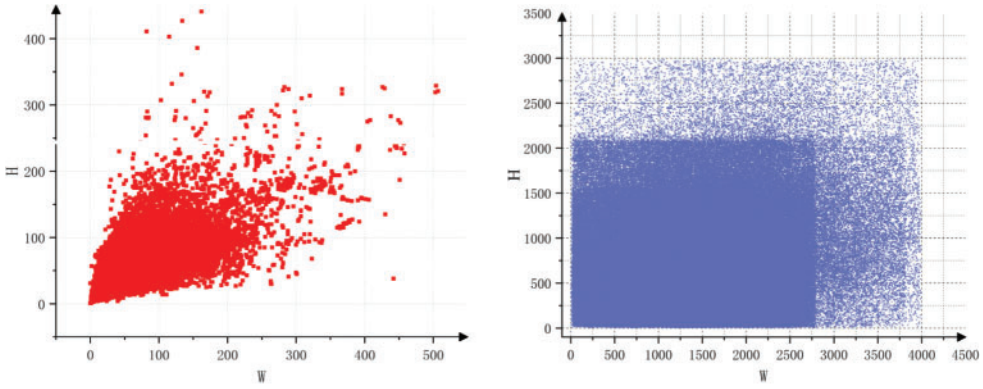
$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{9}$$

where TP is True Positive, FP is False Positive, FN is False Negative, p(r) is the function of the P-R curve, and K is the number of categories. This paper also uses the number of parameters, and floating point operations (FLOPS), where FLOPS denotes the amount of computation required by the model.

4 Results Analysis

4.1 Dataset Analysis

This paper uses a UAV infrared dataset and a Visdrone2019 auxiliary dataset. The analysis of the detection box size in the UAV infrared dataset is depicted in Fig. 9a. Evidently, the dataset pertains to the domain of multi-scale target detection and small target detection. The dataset comprising 6996 pictures was gathered by Shandong Yantai Arrow Photoelectric Technology Co., Ltd. (China). There are six categories: pedestrians, cars, buses, bicycles, trucks, and other targets. This paper's dataset is partitioned in a 8:1:1 ratio, specifically 5724:636:636.



(a) UAV infrared dataset (b) Visdrone2019 dataset

Figure 9: Dataset analysis

The analysis of the detection box size for the Visdrone2019 dataset is depicted in Fig. 9b. The AISKYEYE team at Tianjin University collected the data set. The dataset comprises ten categories: pedestrians, cars, bicycles, and tricycles. Evidently, the Visdrone2019 dataset exhibits greater complexity compared to the infrared dataset.

4.2 Ablation Experiment

Owing to the unique characteristics of certain modules, this study employs the superposition method to conduct the ablation experiment. This paper focuses on the improvement of four modules, namely m1 (SPAN module), m2 (SDCN feature extraction module), m3 (pruning the trained model), and m4 (adding MPDIOU_NMS algorithm). S_i represents the combination of various modules m_i, with S0 indicating the absence of any additional improvement module.

As depicted in Table 2, S1 demonstrates an increase in the SFPN structure and a 0.7% increase in mAP compared to S0. However, its GFLOPS has increased by 7.9. S2 denotes that the SDCN module is incorporated on the foundation of S1 to augment the model's feature extraction capability, albeit at the expense of increased model complexity that can be disregarded. Compared to the four indicators of S1, there is an average increase of 1.15%. S3 denotes the pruning of redundant channels in the model. Approximately 30% of the parameters are pruned to uphold optimal model performance. The objective is to reduce the excessive size of the model and to vary the degree of increase for different indicators, making them incomparable to other modules. Upon adding the MPDIOU_NMS algorithm to verify and test the model detection performance, S4 demonstrates a decrease in mAP(0.5) compared to S3, while other aspects show varying degrees of improvement.

Table 2: Ablation experiment

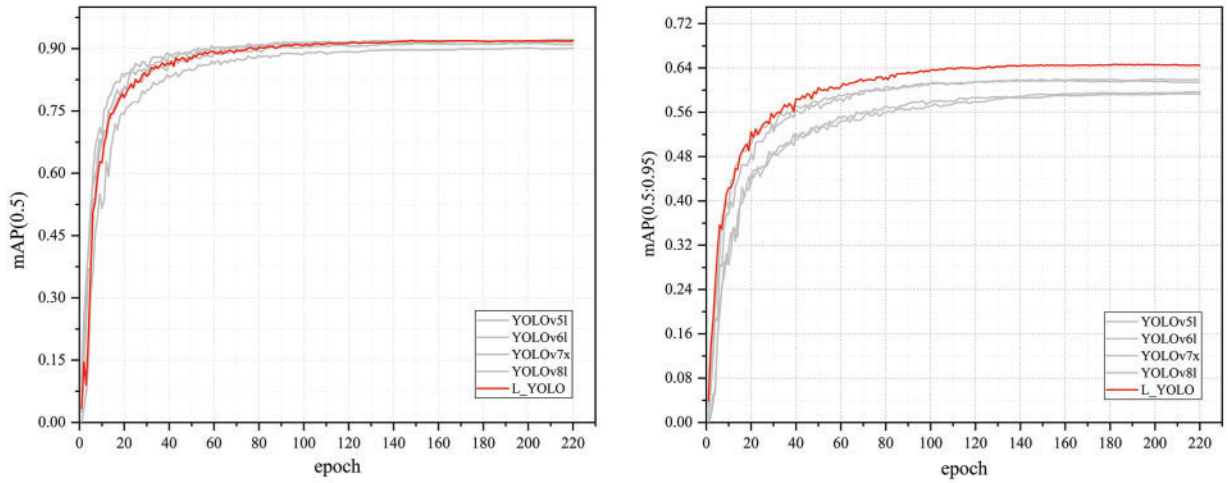
Models	m1	m2	m3	m4	mAP(0.5)	mAP(0.5:0.95)	P	R	Parameters	GFLOPS
S0					0.902	0.595	0.854	0.870	11137096	28.7
S1	1				0.910	0.602	0.858	0.874	10576056	36.6
S2	1	1			0.916	0.616	0.870	0.884	11037676	35.5
S3	1	1	1		0.917	0.624	0.874	0.863	7726636	23.3
S4	1	1	1	1	0.914	0.643	0.872	0.871	7726636	23.3

4.3 Comparison between Different Algorithms

This paper compares four size models (n, s, m, l) in YOLOv5 [33], YOLOv6 [34], and YOLOv8 algorithms, as well as YOLOv7 [35] and YOLOv7x. As depicted in Table 3, YOLOv5l, YOLOv6l, YOLOv7x, and YOLOv8l exhibit superior performance, and their visualization comparison with the L_YOLO algorithm is presented in Fig. 10. To emphasize the superiority of this algorithm, the L_YOLO curve is depicted in red, while the other curves are represented in grey. Evidently, given that the average number of parameters is nine times lower, the algorithm exhibits a slightly higher performance than other algorithms on mAP(0.5). It significantly outperforms any other algorithm on mAP(0.5:0.95).

Table 3: Comparison between different algorithms

Models	Precision	Recall	mAP(0.5)	mAP(0.5:0.95)	Parameters	GFLOPS
YOLOv8n	0.853	0.837	0.884	0.563	3006818	8.1
YOLOv8s	0.854	0.87	0.902	**0.595**	11137906	28.7
YOLOv8m	0.859	0.878	0.912	0.614	25843234	78.7
YOLOv8l	0.868	0.889	0.920	0.619	43634450	165.4
YOLOv7	0.88	0.869	0.914	0.590	37223526	105.2
YOLOv7x	0.861	0.889	0.922	0.595	70848782	189
YOLOv6n	0.832	0.813	0.863	0.550	4238722	11.9
YOLOv6s	0.86	0.851	0.895	0.591	16298594	44
YOLOv6m	0.861	0.859	0.902	0.593	51998946	161.2
YOLOv6l	0.855	0.873	0.901	0.600	110897810	391.9
YOLOv5n	0.857	0.828	0.884	0.563	2509618	7.1
YOLOv5s	0.872	0.856	0.907	0.596	9113858	23.8
YOLOv5m	0.873	0.865	0.914	0.610	25068590	64.4
YOLOv5l	0.882	0.869	0.916	0.617	53136034	134.7
L_YOLO	0.872	0.871	0.914	**0.643**	7726636	23.3



(a) mAP(0.5) (b) mAP(0.5:0.95)

Figure 10: Comparison curves of different algorithms

To provide a more comprehensive representation of the model's performance, this study improves the model to three dimensions for comparative analysis, as illustrated in Fig. 11. The abscissa and ordinate denote the mAP(0.5:0.95) and the mAP(0.5), respectively. The area of the circle corresponds to the floating point operation, with a larger area indicating a more complex model. The study concludes that the L_YOLO model is positioned closer to the upper right corner than others, indicating its superior comprehensive performance.

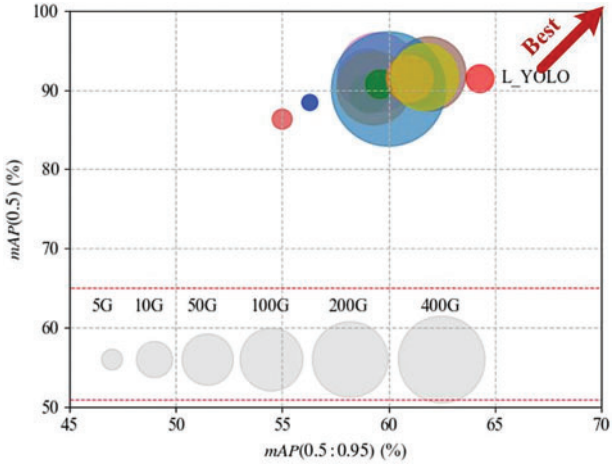


Figure 11: Performance comparison

4.4 Comparison between Different Datasets

In this paper, two data sets are selected, with Fig. 12a representing the infrared dataset and Fig. 12b representing the Visdrone2019 dataset. The red curve in the figure illustrates the comparison of two datasets based on mAP(0.5), while the blue curve depicts the comparison based on mAP(0.5:0.95). The solid line represents the L_YOLO algorithm, while the dotted line corresponds

to the YOLOv8s algorithm. Evidently, the L_YOLO algorithm outperforms the YOLOv8s algorithm in both datasets, particularly in the Visdrone2019 dataset. The specific data is presented in Table 4. The L_YOLO algorithm enhances the mAP(0.5:0.95) by 4.8% on the infrared dataset and the mAP by an average of 12.9% on the Visdrone2019 dataset. The frames per second (FPS) of the L_YOLO algorithm on two datasets is much lower than that of the YOLOv8 algorithm, primarily because the underlying code of the MPDIOU_NMS algorithm is written in Python. However, the FPS of the L_YOLO algorithm exceeds 25 on both datasets, meeting basic industrial requirements and ensuring suitability for daily use.

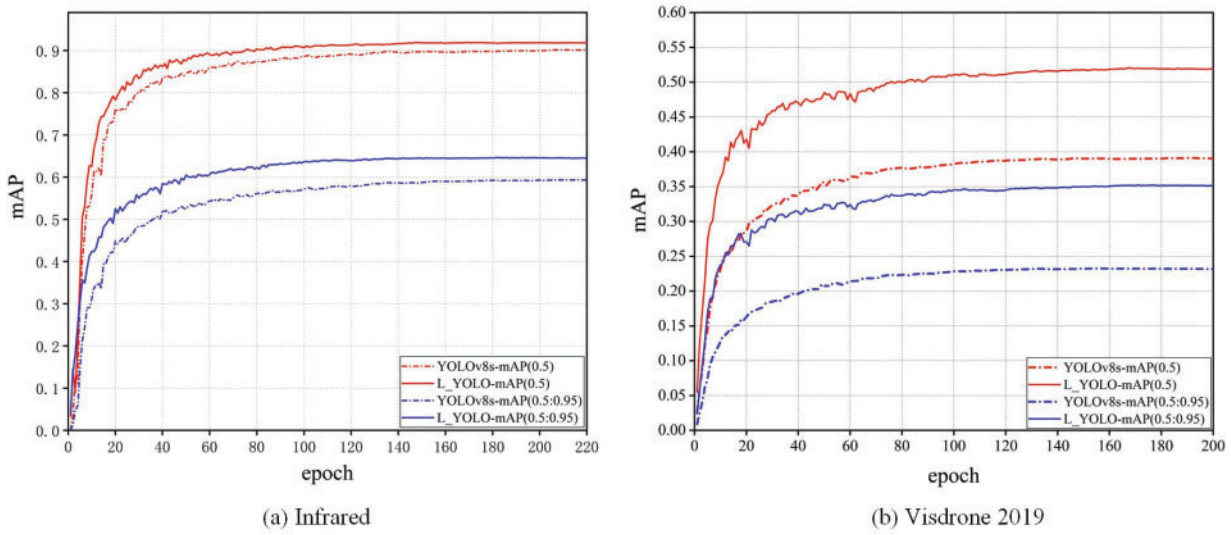


Figure 12: Comparison between different datasets

Table 4: Different datasets

Dataset	Modules	mAP(0.5)	mAP(0.5:0.95)	P	R	GFLOPS	FPS	Parameters
Visdrone2019	YOLOv8s	0.39	0.232	0.507	0.384	28.50	426	11129454
	L_YOLO	**0.519**	**0.351**	**0.600**	**0.420**	23.03	29.3	7720770
Infrared	YOLOv8s	0.902	0.595	0.854	0.870	28.70	521	11137906
	L_YOLO	**0.914**	**0.643**	**0.872**	0.871	23.30	31.6	7726636

This paper also presents the heatmap visualisation for the two datasets using the YOLOv8s algorithm and the L_YOLO algorithm, respectively. The first two images are from the infrared dataset, and the last two are from the Visdrone2019 dataset. Obviously, the L_YOLO algorithm pays more attention to useful semantic information than the YOLOv8s algorithm. The details are shown in Fig. 13.

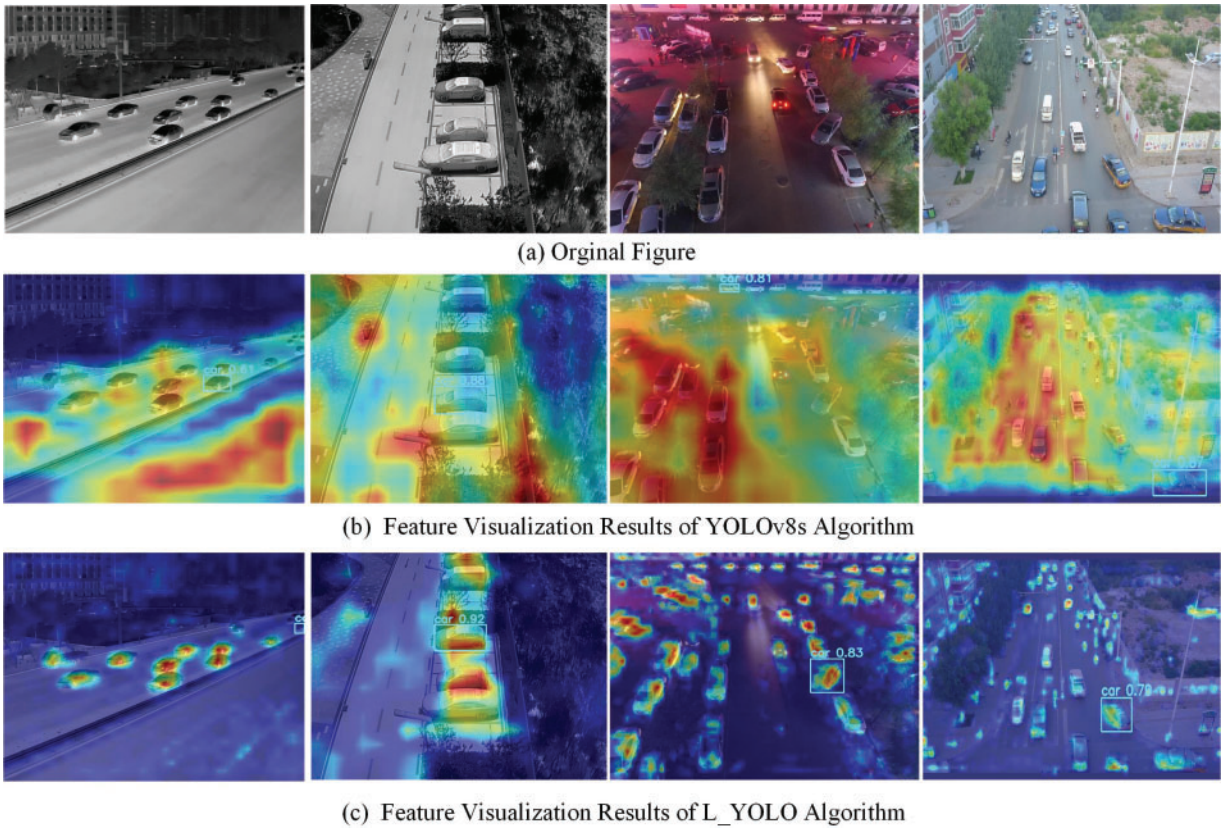


Figure 13: Comparison of UAV infrared image feature visualization results

5 Conclusion

Aiming at the different imaging of UAV aerial photography at different altitudes and the constraints of low light conditions at night make it challenging for operators to discern the target accurately. This paper presents an L-YOLO algorithm designed to address a series of problems. This paper primarily enhances YOLOv8 in the following ways. Initially, the PAN structure of YOLOv8 was substituted with the SPAN structure, followed by the replacement of the neck network feature extraction module with the SDCN module. Subsequently, the model was further pruned. Finally, the MPDIOU_NMS algorithm is added to assist the model in verification and testing. It has achieved good results on infrared datasets and Visdrone2019 datasets. Nevertheless, it is necessary to acknowledge that there are still numerous shortcomings in contemporary work. In the subsequent investigation, the following issues require resolution: (1) There is a need to enhance the speed of model inference further, for instance, by optimizing the NMS algorithm [36]. (2) Further pruning of the module, such as module pruning, can be implemented by utilizing group-level pruning [37], ensuring that no part of the network layer is skipped.

Acknowledgement: Thanks to the infrared dataset provided by Shandong Yantai Arrow Optoelectronics Technology Co., Ltd.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Study conception and design: Zhijian Liu; data collection, analysis, and interpretation of results: Zhijian Liu; draft manuscript preparation: Zhijian Liu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available at https://github.com/pastrami06/CMC.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] H. Z. Fang, L. Ding, L. M. Wang, Y. Chang, and L. X. Yan, "Infrared small UAV target detection based on depthwise separable residual dense network and multiscale feature fusion," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–20, Aug. 2022. doi: 10.1109/TIM.2022.3198490.

[2] T. Liu, R. Li, X. C. Zhong, M. Jiang, and X. L. Jin, "Estimates of rice lodging using indices derived from UAV visible and thermal infrared image," *Agr. Forest. Meteorol.*, vol. 252, pp. 144–154, Jan. 2018. doi: 10.1016/j.agrformet.2018.01.021.

[3] C. M. Wu, Y. Q. Sun, T. J. Wang, and Y. L. Liu, "Underwater trash detection algorithm based on improved YOLOv5s," *J. Real-Time. Image Process.*, vol. 19, no. 5, pp. 911–920, Oct. 2022. doi: 10.1007/s11554-022-01232-0.

[4] W. Hao, C. Ren, M. Han, L. Zhang, and F. Li, "Cattle body detection based on YOLOv5-EMA for precision livestock farming," *Anim.*, vol. 13, no. 22, pp. 3535, Nov. 2023. doi: 10.3390/ani13223535.

[5] L. Q. Zhu, X. M. Li, H. M. Sun, and Y. P. Han, "Research on CBF-YOLO detection model for common soybean pests in complex environment," *Comput. Electron. Agr.*, vol. 216, pp. 108515, Jan. 2024. doi: 10.1016/j.compag.2023.108515.

[6] H. Ouyang, "DEYOv2: Rank feature with greedy matching for end-to-end object detection," Jun. 2023. doi: 10.48550/arXiv.2306.09165.

[7] J. Wang, X. Zhang, G. Gao, and Y. Lv, "OP mask R-CNN: An advanced mask R-CNN network for cattle individual recognition on large farms," in *2023 Int. Conf. Netw. Netw. Appl. (NaNA)*, Qingdao, China, Oct. 2023, pp. 601–606.

[8] P. Y. Jiang, D. J. Ergu, F. Y. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia Comput. Sci.*, vol. 199, pp. 1066–1073, Feb. 2022. doi: 10.1016/j.procs.2022.01.135.

[9] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Comput. Vis.-ECCV 2016: 14th Eur. Conf.*, Amsterdam, The Netherlands, Sep. 2016, pp. 21–37.

[10] F. M. Talaat and H. Z. Eldin, "An improved fire detection approach based on YOLO-v8 for smart cities," *Neural. Comput. Appl.*, vol. 35, no. 28, pp. 20939–20954, Jul. 2023. doi: 10.1007/s00521-023-08809-1.

[11] L. Wang, H. C. Zheng, C. H. Yin, Y. Wang, and Z. X. Bai, "Dense papaya target detection in natural environment based on improved YOLOv5s," *Agron.*, vol. 13, no. 8, pp. 2019, Jul. 2023. doi: 10.3390/agronomy13082019.

[12] W. Zhao, M. Syafrudin, and N. L. Fitriyani, "CRAS-YOLO: A novel multi-category vessel detection and classification model based on YOLOv5s algorithm," *IEEE Access*, vol. 11, pp. 11463–11478, Feb. 2023. doi: 10.1109/ACCESS.2023.3241630.

[13] X. Wang, Y. Tian, K. Zheng, and C. Liu, "C2Net-YOLOv5: A bidirectional Res2Net-based traffic sign detection algorithm," *Comput., Mater. Contin.*, vol. 77, no. 2, pp. 1949–1965, Sep. 2023. doi: 10.32604/cmc.2023.042224.

[14] B. Chen and X. Miao, "Distribution line pole detection and counting based on YOLO using UAV inspection line video," *J. Electr. Eng. Technol.*, vol. 15, pp. 441–448, Jun. 2020. doi: 10.1007/s42835-019-00230-w.

[15] A. Andoli, N. Mohammed, S. C. Tan, and W. P. Cheah, "A review on community detection in large complex networks from conventional to deep learning methods: A call for the use of parallel meta-heuristic algorithms," *IEEE Access*, vol. 9, pp. 96501–96527, Jul. 2021. doi: 10.1109/ACCESS.2021.3095335.

[16] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: Challenges, architectural successors, datasets and applications," *Multimed. Tools. Appl.*, vol. 82, no. 6, pp. 9243–9275, Jan. 2023. doi: 10.1007/s11042-022-13644-y.

[17] S. H. Cao, T. Wang, T. Li, and Z. H. Mao, "UAV small target detection algorithm based on an improved YOLOv5s model," *J. Vis. Commun. Image Rep.*, vol. 97, pp. 103936, Sep. 2023. doi: 10.1016/j.jvcir.2023.103936.

[18] Y. M. Hui, J. Wang, and B. Li, "STF-YOLO: A small target detection algorithm for UAV remote sensing images based on improved SwinTransformer and class weighted classification decoupling head," *Meas.*, vol. 224, pp. 113936, Jan. 2024. doi: 10.1016/j.measurement.2023.113936.

[19] S. Ali, A. Jalal, M. H. Alatiyyah, K. Alnowaiser, and J. Park, "Vehicle detection and tracking in uav imagery via yolov3 and kalman filter," *Comput., Mater. Contin.*, vol. 76, no. 1, pp. 1249–1265, Jun. 2023. doi: 10.32604/cmc.2023.038114.

[20] W. Lu, "A CNN-transformer hybrid model based on CSWin transformer for UAV image object detection," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 16, pp. 1211–1231, Jan. 2023. doi: 10.1109/JSTARS.2023.3234161.

[21] X. Qian, X. Wang, S. Yang, and J. Lei, "LFF-YOLO: A YOLO algorithm with lightweight feature fusion network for multi-scale defect detection," *IEEE Access*, vol. 10, pp. 130339–130349, Dec. 2022. doi: 10.1109/ACCESS.2022.3227205.

[22] S. Zhu and M. Miao, "SCNet: A lightweight and efficient object detection network for remote sensing," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, pp. 1, Dec. 2023. doi: 10.1109/LGRS.2023.3344937.

[23] Y. B. Zou and C. Y. Liu, "A light-weight object detection method based on knowledge distillation and model pruning for seam tracking system," *Meas.*, vol. 220, pp. 113438, Oct. 2023. doi: 10.1016/j.measurement.2023.113438.

[24] X. Q. Wang, H. B. Gao, Z. M. Jia, and Z. J. Li, "BL-YOLOv8: An improved road defect detection model based on YOLOv8," *Sens.*, vol. 23, no. 20, pp. 8361, Sep. 2023. doi: 10.3390/s23208361.

[25] G. Wang, Y. F. Chen, P. An, H. Y. Hong, and J. H. Hu, "UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios," *Sens.*, vol. 23, no. 32, pp. 7190, Jul. 2023. doi: 10.3390/s23167190.

[26] X. Z. Zhu, H. Hu, S. Lin, and J. F. Dai, "Deformable ConvNets V2: More deformable, better results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, Jun. 2019, pp. 9308–9316.

[27] N. Mahendran, "SENetV2: Aggregated dense layer for channelwise and global representations," Nov. 2023. doi: 10.48550/arXiv.2311.10807.

[28] D. D. Wang and D. J. He, "Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning," *Biosyst. Eng.*, vol. 201, pp. 271–281, Oct. 2021. doi: 10.1016/j.biosystemseng.2021.08.015.

[29] Z. Liu, J. G. Li, Z. Q. Shen, and G. Huang, "Learning efficient convolutional networks through network slimming," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Dec. 2017, pp. 2736–2744.

[30] T. T. Yang, S. Y. Zhou, A. J. Xu, and J. H. Ye, "An approach for plant leaf image segmentation based on YOLOV8 and the improved DEEPLABV3+," *Plants*, vol. 12, no. 19, pp. 3438, Sep. 2023. doi: 10.3390/plants12193438.

[31] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS–Improving object detection with one line of code," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Apr. 2017, pp. 5561–5569.

[32] S. L. Ma and Y. Xu, "MPDIoU: A loss for efficient and accurate bounding box regression," Jul. 2023. doi: 10.48550/arXiv.2307.07662.

[33] W. T. Wu, H. Liu, L. L. Li, Y. L. Long, and X. D. Wan, "Application of local fully convolutional neural network combined with YOLO v5 algorithm in small target detection of remote sensing image," *PLoS One*, vol. 16, no. 10, pp. 10259283, Sep. 2021. doi: 10.1371/journal.pone.0259283.

[34] C. Y. Li, L. L. Li, H. L. Jiang, K. H. Weng, and Y. F. Geng, "YOLOv6: A single-stage object detection framework for industrial applications," Sep. 2022. doi: 10.48550/arXiv.2209.02976.
[35] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," Jul. 2022. doi: 10.48550/arXiv.2207.02696.
[36] H. Zhao, J. K. Wang, D. Y. Dai, S. Q. Lin, and Z. H. Chen, "D-NMS: A dynamic NMS network for general object detection," *Neurocomput.*, vol. 512, pp. 225–234, Nov. 2022. doi: 10.1016/j.neucom.2022.09.080.
[37] G. F. Fang, X. Y. Ma, M. L. Song, M. B. Mi, and X. C. Wang, "DepGraph: Towards any structural pruning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Vancouver, BC, Canada, Jun. 2023, pp. 16091–16101.