Biomimetic Intelligence and Robotics 4 (2024) 100190

Contents lists available at ScienceDirect

Biomimetic Intelligence and Robotics

journal homepage: www.elsevier.com/locate/birob

Research Article

DS-YOLO: A dense small object detection algorithm based on inverted bottleneck and multi-scale fusion network

Hongyu Zhang [a], Guoliang Li [b,*], Dapeng Wan [a], Ziyue Wang [c], Jinshun Dong [a], Shoujun Lin [a], Lixia Deng [a,*], Haiying Liu [a]

[a] School of Information and Automation Engineering, Qilu University of Technology, Jinan 250353, China
[b] Unimation Intelligent Technology Co., Jinan 250101, China
[c] School of Computer Science, University of St Andrews, St Andrews KY16 9AJ, United Kingdom

ARTICLE INFO

Article history:
Received 5 July 2024
Revised 6 October 2024
Accepted 15 October 2024
Available online 26 October 2024

Keywords:
Dense objects detection
LFS-PAFPN
DOConv
C2fUIB
YOLOv8

ABSTRACT

In the field of security, intelligent surveillance tasks often involve a large number of dense and small objects, with severe occlusion between them, making detection particularly challenging. To address this significant challenge, Dense and Small YOLO (DS-YOLO), a dense small object detection algorithm based on YOLOv8s, is proposed in this paper. Firstly, to enhance the dense small objects' feature extraction capability of backbone network, the paper proposes a lightweight backbone. The improved C2fUIB is employed to create a lightweight model and expand the receptive field, enabling the capture of richer contextual information and reducing the impact of occlusion on detection accuracy. Secondly, to enhance the feature fusion capability of model, a multi-scale feature fusion network, Light-weight Full Scale PAFPN (LFS-PAFPN), combined with the DO-C2f module, is introduced. The new module successfully reduces the miss rate of dense small objects while ensuring the accuracy of detecting large objects. Finally, to minimize feature loss of dense objects during network transmission, a dynamic upsampling module, DySample, is implemented. DS-YOLO was trained and tested on the CrowdHuman and VisDrone2019 datasets, which contain a large number of densely populated pedestrians, vehicles and other objects. Experimental evaluations demonstrated that DS-YOLO has advantages in dense small object detection tasks. Compared with YOLOv8s, the Recall and mAP@0.5 are increased by 4.9% and 4.2% on CrowdHuman dataset, 4.6% and 5% on VisDrone2019, respectively. Simultaneously, DS-YOLO does not introduce a substantial amount of computing overhead, maintaining low hardware requirements.

1. Introduction

Object detection plays a vital role in the field of computer vision, with applications spanning various domains such as robotics, autonomous driving systems, and industrial production line monitoring. In the realm of intelligent security, object detection can intelligently monitor and capture various information of interest objects, playing a crucial role in tasks such as traffic monitoring and hazard prediction. However, these scenarios often involve a large number of dense and small objects, with severe occlusion between them, leading to issues such as low accuracy for small objects, high false detecting rate, and significant miss rate for dense objects. Therefore, in-depth research and continuous innovation in dense small objects detection technology hold significant practical importance and long-term value.

In recent years, with the development of science and technology, deep learning-based object detection algorithms have become increasingly mature. YOLO series algorithms [1–3] are extremely outstanding one-stage object detection algorithms based on Convolutional Neural Network (CNN). They have attracted significant attention from researchers due to their high accuracy, fast speed and high practical application value. YOLO treats detection as a regression task, dividing the entire image into an S×S grid, with each grid responsible for detecting objects within it. It uses bounding box localization, conditional probability, confidence, etc., to complete predictions in one go [4]. To date, the mainstream version of the algorithm has been updated to YOLOv10 [5].

In addition to the YOLO series, there are other algorithms such as two-stage algorithms based on CNN [6–8] and algorithms based on Transformer [9,10]. These algorithms generally have higher accuracy, while have more complex network structure and higher hardware requirement. The YOLO series algorithms, on the other hand, have relatively simpler structures, lower model

* Corresponding authors.
E-mail addresses: glli@unimationtech.com (G. Li), lixiadeng@qlu.edu.cn (L. Deng).

https://doi.org/10.1016/j.birob.2024.100190

parameter counts, and lower computational complexity, making them less demanding on hardware and more broadly compatible.

As the "eyes" of the Internet of Everything era, video surveillance holds immense application value and has garnered extensive attention from researchers. Deepak Kumar Jain et al. [11] addressed the issue of low pedestrian detection accuracy in video surveillance systems by proposing the Robust Multi-modal Pedestrian Detection using a Deep Convolutional Neural Network with an Ensemble Learning (RMPD-DCNN-EL) model. This model combined ensemble learning with computer vision methods to produce more accurate results. It used the SimAM EfficientNet model for feature extraction and employed three deep learning models for ensemble classification: Nested Long Short-Term Memory (NLSTM), Deep Belief Networks (DBN), and Extreme Learning Machine (ELM). Experimental results demonstrated that this model outperforms other deep learning methods, achieving robust pedestrian detection even when pedestrian positions and postures change. However, the model performed poorly in scenarios with occlusion and dense crowds. Xu et al. [12], addressing the need for real-time pedestrian detection in intelligent surveillance systems, proposed an efficient real-time pedestrian detection model based on an improved ShuffleNet and YOLOv3. This model aimed to solve real-time pedestrian detection with high accuracy. By lightweighting the YOLOv3 backbone, it reduced 67.5% of floating-point operations per second (FLOP) and 65.1% of parameters. Experiments showed that the proposed method can achieve an FPS of 180, meeting the requirements for implementing end-to-end object detection algorithms. However, the performance of the lightweight model declined, and its detection effectiveness for small objects was insufficient. Hsu et al. [13] tackled the challenge of poor pedestrian detection in low-resolution (LR) images by proposing an end-to-end Multi-scale Structure-Enhanced Super-Resolution (MsSE-SR) method. This method used super-resolution technology to upscale LR images to high-resolution (HR) images and then detected pedestrians using YOLOv4. This algorithm effectively addressed the low performance of pedestrian detection in LR images but did not adequately solve the issue of high occlusion among pedestrians.

The development of dense small object detection algorithms currently faces many significant challenges, with occlusion being the primary issue. In practical applications, objects are often occluded by surrounding objects such as trees, railings, and vehicles. These occlusions make it difficult for traditional data-driven detectors to accurately detect the key features of objects. Secondly, scale variation is another important problem. The scale of objects in images changes with the distance from the camera, angle, etc., resulting in significant differences between large and small objects, which increases the difficulty of detection. Additionally, there are many small objects in images, and enhancing the model's ability to detect small objects while maintaining low computational overhead and reducing miss rate is an urgent problem to be solved.

Based on the above questions, this paper proposes a more powerful model, DS-YOLO, to address the issue of missed detection and low accuracy caused by occlusions in intelligent surveillance scenarios with a large number of dense small objects. The main contributions are as follows:

- To enhance the backbone's ability to extract features of small objects while maintaining computational efficiency, the backbone's structure and channel numbers of each layer were redesigned. To address the issue of occlusion among objects affecting detection performance, the improved C2fUIB module was introduced into the backbone. This allows the network to achieve a larger receptive field, further improving its ability to capture contextual information and reducing the impact of occlusion on detection.
- To address the suboptimal feature fusion capability of the PAFPN used in Neck of YOLOv8s, a new Neck, LFS-PAFPN, was designed to replace. It better captures and fuses spatial and semantic information, improving the accuracy of both large and small objects. To lightweight the neck structure, the DO-C2f module, which integrates DOConv, was used.
- To mitigate the loss of dense object features during the upsampling process, a more efficient upsampling module, DySample, was employed.

In summary, given that intelligent surveillance tasks in the security field often involve a large number of dense and small objects with significant occlusion, this paper proposed DS-YOLO to reduce the issue of missed detections.

2. YOLOv8

YOLOv8 [3] is the latest generation algorithm released by Ultralytics. The YOLO series of algorithms are renowned for their balance of accuracy and speed, and YOLOv8 is the culmination of these advancements. YOLOv8s is one of the smaller models(n, s, m, l), with fewer parameters and lower computational complexity, making it easier to deploy on various devices. Therefore, this paper selects YOLOv8s as the benchmark model. Its structure is shown in Fig. 1.

YOLOv8 is designed based on YOLOv5, incorporating advanced concepts from other excellent YOLO series algorithms such as YOLOv7 and YOLOX. It adopts an Anchor-Free approach, significantly enhancing the algorithm's generalization performance across various scenarios and making it more adaptable to multi-scale object detection tasks. The network mainly consists of three parts: the Backbone, the Neck, and the Head. The C2f structure, based on C3 and ELAN designs, has a stronger feature extraction capability. Mainly due to the recursive design of the Bottleneck allows for the acquisition of more gradient flow information. The Neck still employs a multi-scale feature fusion structure, FPN + PAN, which integrates the rich spatial information from shallow layers with the rich semantic information from deep layers, mitigating the information loss problem during network propagation. The detection head uses a decoupled design, separating the classification and regression operations.

The series of designs in YOLOv8 have enabled it to achieve state-of-the-art (SOTA) performance. However, YOLOv8 primarily targets conventional multi-scale objects and performs poorly in detecting small objects. The YOLOv8-P2 places more emphasis on detecting small objects, while deepening the network brings better performance, it also increases a large number of computational burden.

3. The improved model DS-YOLO

To further enhance the detection effectiveness for dense small objects, this paper proposes an improved model based on YOLOv8s, named the DS-YOLO. The structure is shown in Fig. 2.

Firstly, in order to enhance the feature extraction capability of the model's backbone for the objects, DS-YOLO adopted a redesigned backbone network, which strengthened the model's utilization of the shallow layer rich in object features, making it more adaptable to the task scenarios of dense small objects. The new backbone reconstructed the structure and number of channels of each layer to avoid creating a large computational burden. By introducing DO-C2f and C2fUIB as the main modules for feature extraction of the model, performance was improved while ensuring low parameter volume and computational burden. Secondly, to better utilize the obtained multi-scale features, DS-YOLO introduced a new LFS-PAFPN to enhance the model's

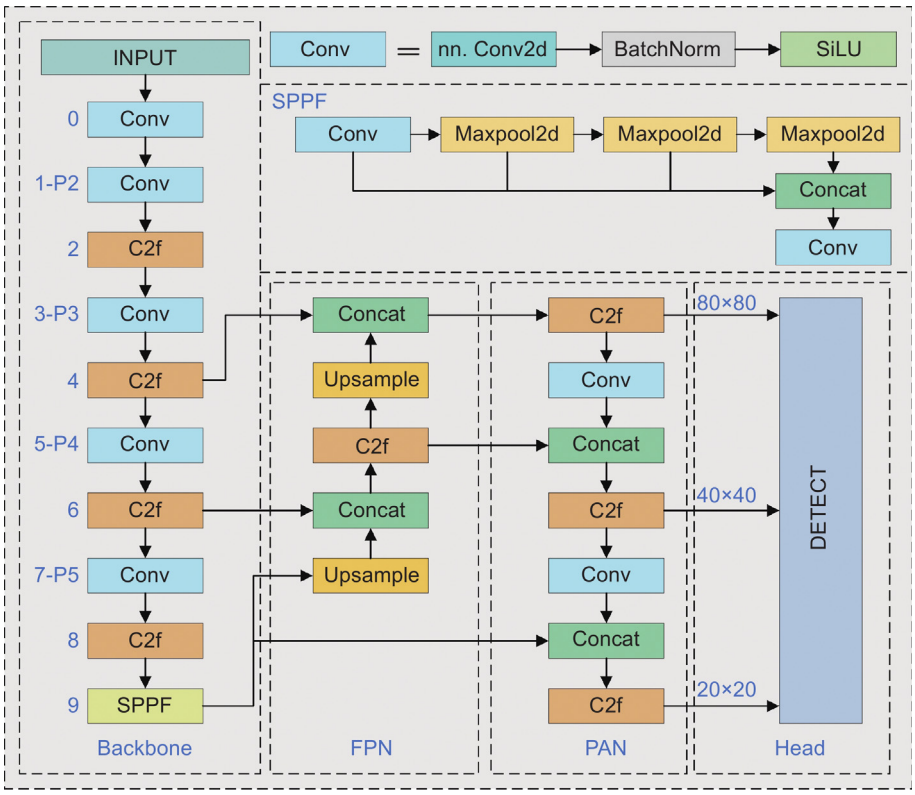

Fig. 1. The structure of YOLOv8.

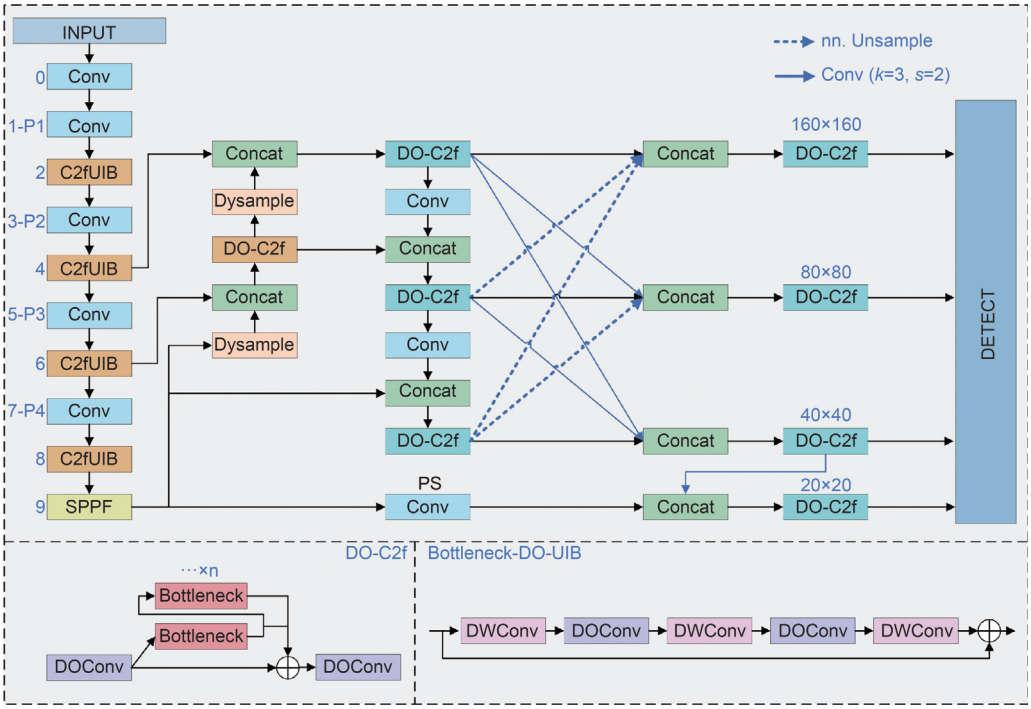

Fig. 2. The structure of DS-YOLO.

multi-scale feature fusion capability. Dense small objects contain more information including various details of the object in the shallow high-resolution feature map, and more positional, category information in the deep low-resolution feature map. A more efficient multi-scale feature fusion network could more completely aggregate the captured information of various kinds, thereby improving detection accuracy and reducing the rate of missed detections. Finally, to reduce the problem of feature loss of the object during the up-sampling process, DySample was introduced. In dense scenes, the feature information of the object was highly overlapping, and DySample's adaptive sampling position and weight generation mechanism could retain more complete

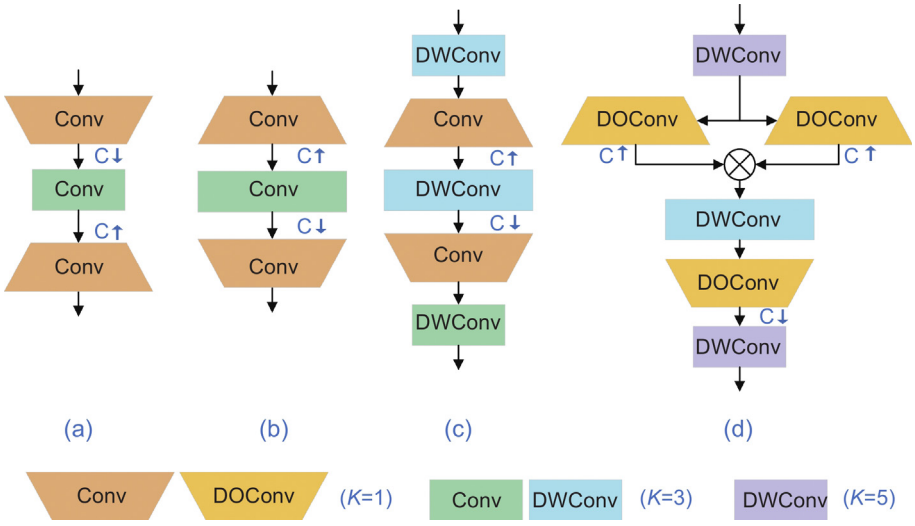

Fig. 3. The structure of NR, IR, UIR, DO-UIB.

detail information in the feature maps at all levels. The following sections would detail the specific design ideas of each module of DS-YOLO.

3.1. DO-C2f based on DOConv

CNNs have achieved tremendous success in image processing and computer vision tasks. However, standard convolution operations face bottlenecks in terms of parameters and computational load, especially in scenarios requiring high model expressiveness and limited computational resources. Depthwise Over-parameterized Convolution (DOConv) [14] was introduced to address this issue by incorporating parameter redundancy in the depth direction, thereby enhancing the representational capacity of convolutional kernels while maintaining computational efficiency.

DOConv can be implemented in two mathematically equivalent ways: feature composition and kernel composition. The feature composition method involves first applying a depthwise convolution kernel D to the input feature map P, resulting in the transformed feature P'.

$$P' = D \odot P \tag{1}$$

Then, a conventional convolution kernel is applied to, resulting in the output feature O.

$$O = W * P' \tag{2}$$

Kernel composition method: First, the conventional convolution kernel W is transformed by the depthwise convolution kernel D^T, resulting in the composite convolution kernel W'.

$$W' = D^T \odot W \tag{3}$$

Then, W' is applied to the input feature map P, resulting in the output feature O.

$$O = W' * P \tag{4}$$

This paper proposes DO-C2f to enhance the network's feature extraction capability and reduce computational burden. DOConv significantly improves the representation and feature extraction capabilities of convolutional kernels by introducing parameter redundancy in the depth direction, solving the bottleneck problem of traditional convolutional operations under high performance requirements and limited computing resources. Its adaptive design not only enhances the model's expressive power but also maintains high computational efficiency, making it an effective solution for improving model performance in various image processing and computer vision tasks, especially on mobile and embedded devices.

3.2. Lightweight backbone network with C2fUIB

The Inverted Residual module was first proposed in MobileNetV2 [15], as shown in Fig. 3(b). Its key feature is to first expand the channels, then perform depthwise separable convolution, and finally compress the channels. It is a flexible modeling module suitable for efficient network design, which can adapt to various optimization objectives without causing a drastic increase in computational complexity. This is exactly the opposite of the channel design of the traditional residual module [16] (Normal Residual), as shown in Fig. 3(a).

MobileNetV4 [17] proposed the Universal Inverted Residual structure. Based on this efficient module, this paper designs a new bottleneck structure Depthwise Over-parameterized Universal Inverted Bottleneck (DO-UIB), which expands the receptive field of this layer structure while lightening the network, enabling the network to capture more multi-scale contextual information. The specific structure is shown in Fig. 3(c).

The DO-UIB module introduces DOConv into the Universal Inverted Bottleneck block, enhancing the feature representation ability while lightening the network.

C2fUIB, based on DOC2f, embeds DO-UIB as a Bottleneck, greatly reducing computational complexity and the number of parameters. To avoid a significant drop in performance and to enhance the feature extraction ability of small objects, the backbone network structure and the number of channels have been redesigned. The new backbone better utilizes the rich small object information from the shallow layers and does not increase the computational load compared to the original structure. The specific structure is shown in Fig. 4.

3.3. Multi-scale feature fusion network LFS-PAFPN

The main role of the neck network is feature extraction and feature fusion. As the network deepens, the resolution of the feature map gradually decreases, and the semantic features of the object such as category and contour gradually enrich. At the same time, the rich spatial information of the shallow network is gradually lost. The feature fusion ability of the Neck allows the network

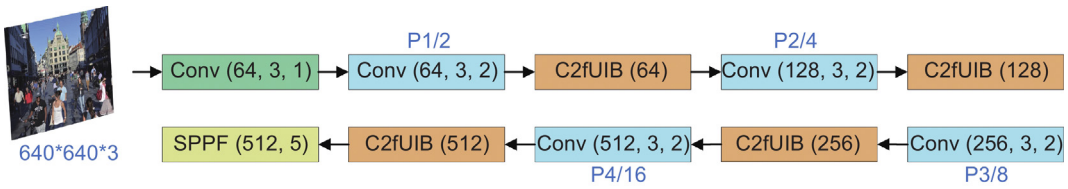

Fig. 4. The structure of backbone.

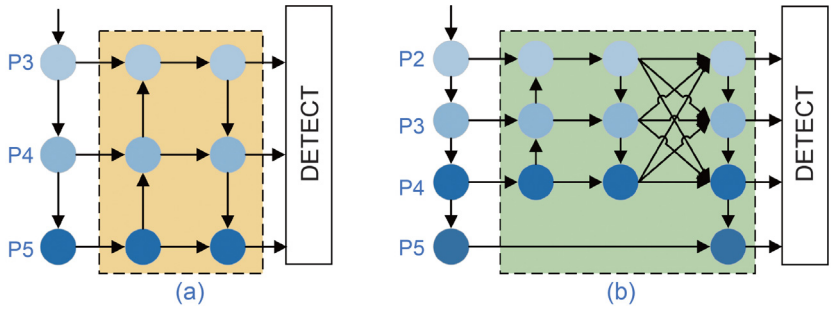

Fig. 5. The structure of PAFPN and LFS-PAFPN.

to combine the rich spatial information of the shallow layers with the rich semantic information of the deep layers, to some extent solving the problem of information loss and degradation.

The FPN + PAN structure used in YOLOv8, in order to balance multi-scale objects under ordinary scenes, chose 20×20, 40×40, and 80×80 feature maps for detection. But in tasks like pedestrian detection, due to issues such as shooting distance and angle, there are many objects with too few pixels. At this time, the detection effect of the original network cannot meet the requirements, and the phenomenon of missed detection is serious.

In response to the problems of feature fusion and poor small object detection ability in the Neck, this paper proposes a new neck feature fusion network, LFS-PAFPN. The specific structure is shown in Fig. 5.

As can be seen, a full connection operation between each feature map is designed before the feature map inputs to the detection head. This operation can fully fuse the spatial and semantic information contained in each layer, improving the detection accuracy of the algorithm. Specifically, in order to increase the ability to capture dense small object features, the main neck network is constructed using the high-resolution feature maps P2, P3, and P4 of the backbone. The P5 feature map of the backbone is also used to further fuse with the P4 feature map that has fused multi-scale information, ensuring the detection performance of large objects. LFS-PAFPN not only better detects small objects but also ensures the detection effect of large objects, while reducing the redundant computational burden for large objects.

3.4. DySample

In image processing tasks, fixed sampling methods (such as nearest neighbor interpolation and bilinear interpolation) are computationally simple, but their static characteristics cannot be adaptively adjusted according to the complexity of input features. This static sampling often cannot retain enough detail information when facing complex scenes, leading to feature loss and model performance degradation. To solve this problem, Liu et al. [18] proposed DySample, a dynamic sampling strategy. DySample adjusts the sampling position and weight dynamically, allowing the sampling process to adaptively handle different input features, thereby retaining more useful information in the feature transformation and downsampling process. The main calculation process is shown in algorithm 1:

Algorithm 1 Algorithm of DySample

Input: A feature map X (b, c, h, w), upsampling scale s, offset function:f_{offset} (usually a neural network), static or dynamic range constraint factor: α.

Output: Upsampled feature map: X $(b, c, s*h, s*w)$.

1: S = Initialize Sampling Positions (X, s); // Initialization of sampling positions (depends on nearest neighbor or bilinear)

2: Offset $\Delta P = f_{offset}(X)$; // Offset calculation

3: α = dynamic_factor = sigmoid (linear1 (X)) $\times$ 0.5; // Using a dynamic range constraint factor, first calculate the dynamic factor

4: $\Delta P' = \alpha \times \Delta P$; // Constrained offset

5: $S' = S + \Delta P'$; // Adjusted sampling positions

6: Feature resampling; // Resampling using bilinear interpolation or other methods according to S' to obtain the upsampled feature map X'

7: Return: X' ; // Output upsampling results

By introducing a dynamic sampling strategy, DySample has solved the limitations of traditional fixed sampling methods in dealing with complex features. Its adaptive sampling position and weight generation mechanism allow the model to retain more useful information in the feature transformation and downsampling process, significantly improving the overall performance of the model. Compared with traditional methods, DySample has shown significant advantages in detail preservation, model performance, and computational efficiency.

4. Experimental verification and result analysis

4.1. Dataset analysis

The DS-YOLO validation experiments used the public datasets CrowdHuman and VisDrone2019. CrowdHuman dataset contains three classes: head, full body, and visible body. The dataset includes 150,00 images and over 1.2 million annotated human instances for train, 4370 images for valid and 5018 images for test. The images in the dataset come from a variety of scenes and environments, including streets, shopping malls, and parks, and

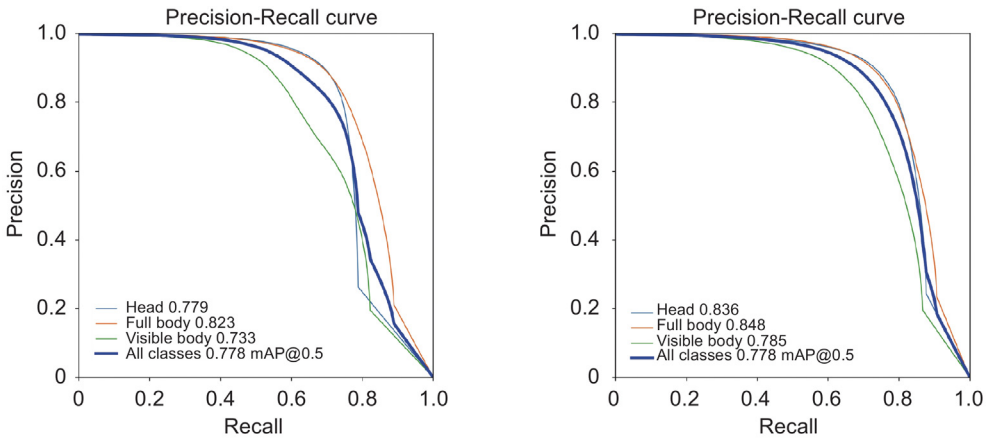

Fig. 6. P-R curves of YOLOv8s(left) and DS-YOLO(right) on CrowdHuman dataset.

Table 1
Model training hyperparameters.

Items	Version
Epoch	100
Batch	8
Image_size	640
Optimizer	SGD
Learning rate	0.01–0.0001
Learning rate scheduling strategy	Linear

exhibit high diversity. A notable characteristic of CrowdHuman is that the images contain a large number of crowded people and include people of different sizes, from small, distant figures to large, close-up figures. The target in Visdrone2019 contains 10 categories, it has a total of 7016 images in the training and validation sets. Most of the objects in the two datasets are small and dense, which brings great challenges to the detection task.

4.2. Experimental environment

The model experiments were based on the PyTorch (Version: 2.1.2) framework. The experiments were conducted using an Nvidia RTX 3090 (24.0 GB) GPU, CUDA version 12.1, Python version 3.10.8, and the Ubuntu operating system. No pre-trained weights were used during model training, and the hyperparameter settings are shown in Table 1.

4.3. Evaluation metrics

The main evaluation metrics used for experimental validation were precision (P), recall (R), mean average precision at threshold 0.5 (mAP@0.5), parameter count (Params), and floating-point operations (GFLOPs). The calculation formulas for P, R, and mAP@0.5 are shown in Eqs. (5), (6), and (7), respectively.

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

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

$$mAP = \frac{1}{n}\sum_{i=0}^{n} AP_i, AP = \int_0^1 PRdr \tag{7}$$

4.4. Experimental results of the DS-YOLO on CrowdHuman dataset

Fig. 6 shows the P-R curves of YOLOv8s and DS-YOLO on CrowdHuman. It can be seen that DS-YOLO's accuracy improved across all categories. This is mainly due to the structural improvement of DS-YOLO, which greatly enhances the feature extraction and feature fusion capabilities of the model. In the monitoring scenario, the dense small target feature information is captured more completely. As shown in Fig. 6, DS-YOLO improves accuracy on "Head" by 5.7%, "Full body" by 2.5%, and "Visible body" by 5.2% than those of YOLOv8s. Overall, the detection accuracy of DS-YOLO reaches 82.3% in the mAP@0.5 indicator, which is 4.5% higher than that of YOLOv8s.

To evaluate the superiority of DS-YOLO, a number of comparative experiments were conducted for verification. Table 2 shows the comparison between DS-YOLO and other models on CrowdHuman dataset. The selection criteria for the comparison algorithms are that the parameter count and computation complexity should be within a reasonable range.

As shown in Table 2, compared with the baseline models YOLOv8s and YOLOv8m, DS-YOLO has higher performance, with Recall improved by 4.9% and 2.1%, and mAP@0.5 improved by 4.2% and 1.6%, respectively. Compared with YOLOv8s, DS-YOLO's parameter count decreases by about 2M, while the computation complexity increases by 2.3 GFLOPs. But Recall and average precision are significantly improved, even higher than those of the larger model, YOLOv8m. Compared with YOLOv8-P2, both Recall and mAP@0.5 are improved by 0.6%, and DS-YOLO's parameter count and computation complexity are reduced by nearly 13% and 16%, respectively. Compared with other models, like YOLOv5m, YOLOv9s, YOLOv10s, the Recall is improved by 4.8%, 5.6%, 3.9%, and the mAP@0.5 is improved by 6%, 5.2%, 3.7%, respectively. Compared with RT-DETR, DS-YOLO's parameter count and computation complexity are both reduced by nearly 50%, and the Recall is only 1.7% lower. Compared with HF-YOLO [19], the Recall is improved by 1.4% and the mAP@0.5 is improved by 0.7%.

The improvement of DS-YOLO backbone structure and the introduction of LFS-PAFPN, DOConv, C2fUIB and other modules together improve the detection accuracy and reduce the missed detection rate. At the same time, DS-YOLO does not create an excessive increase in computing burden, ensuring its wide scalability. The application of the model in the intelligent security scenario is more advantageous.

Table 3 shows the reasoning time of DS-YOLO and the other three models with the highest detection accuracy. On the test set of CrowdHuman dataset, the average detection time of DS-YOLO is 20.1 ms. Compared with YOLOv8s-P2 and YOLOv8m, the time is longer by 6 ms and 5.5 ms, respectively. But the detection accuracy of DS-YOLO is higher and the miss rate is lower. Moreover, the DS-YOLO is 2.9 ms faster than the RT-DETR. In the case of similar detection accuracy, DS-YOLO has faster detection speed and less computational overhead.

Table 2
Experimental comparison between DS-YOLO and other models on CrowdHuman dataset.

Model	*P* (%)	*R* (%)	mAP@0.5 (%)	mAP@0.5-0.95 (%)	Params (M)	FLOPS (G)
YOLOv5m	84.0	68.3	76.3	45.7	20.897	48.2
YOLOv8s	85.1	68.2	78.1	50.4	11.137	28.7
YOLOv8s-P2	85.4	72.5	81.7	53.3	10.638	37.0
YOLOv8m	86.1	71.0	80.7	53.9	25.858	79.1
YOLOv9s	84.8	67.5	77.1	50.5	9.600	38.7
YOLOv10s	83.9	69.2	78.6	50.6	8.037	24.5
HF-YOLO [19]	86.3	71.7	81.6	–	–	–
RT-DETR	85.8	74.8	82.3	54.1	20.086	58.3
DS-YOLO	85.5	73.1	82.3	54.3	8.288	31.0

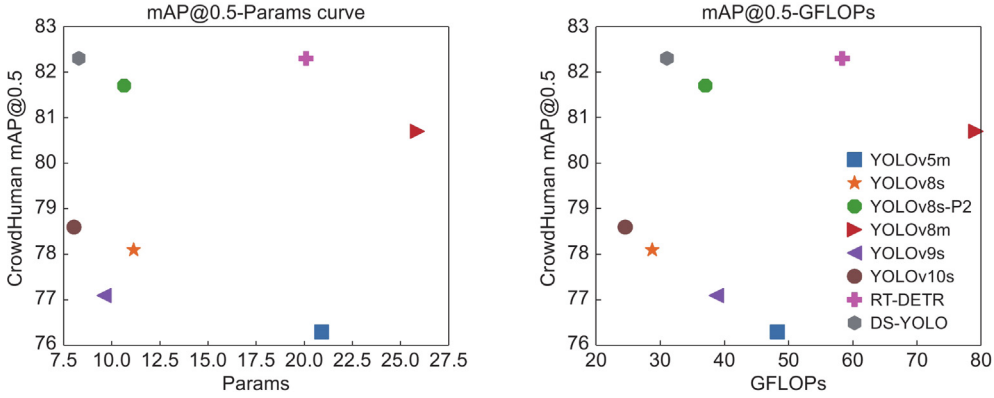

Fig. 7. MAP@0.5-Params (left) and mAP@0.5-GFLOPs (right) scatter diagram of models on CrowdHuman dataset.

Table 3
Comparison of inference speed of different models on test set.

Datasets	Models	Speed (ms)
CrownHuman	YOLOv8s-P2	14.1
	YOLOv8m	14.6
	RT-DETR	23.0
	DS-YOLO	20.1

Fig. 7 shows the models' mAP@0.5 in relation to the number of parameters and the computational complexity. The better the model is in the upper left corner of the scatter plot. DS-YOLO struck a better balance between precision and the number of parameters and computational complexity. DS-YOLO achieved a higher mAP@0.5 and lower parameter count and computational complexity compared with other models.

4.5. Experimental results of the DS-YOLO on VisDrone2019 dataset

In order to verify the detection effect of DS-YOLO on other objects, VisDrone2019 dataset, with richer object categories, was selected for further experimental comparison. As shown in Fig. 8, compared with YOLOv8s, the DS-YOLO's average detection precision of all categories have improved by around 4%. In addition, the detection precision of the DS-YOLO has improved by 4.9%.

The VisDrone2019 dataset contains many dense and small vehicle objects in addition to pedestrians. As shown in Table 4, DS-YOLO also achieves higher average precision accuracy and recall rates. Compared with YOLOv8s, DS-YOLO increases mAP@0.5 and Recall by 5%, 4.6%, respectively. Compared with the larger model YOLOv8m, mAP@0.5 and Recall are increased by 1.6%, 1.2%, respectively. Compared with other models, like YOLOv5m, YOLOv9s, YOLOv10s, DS-YOLO has a great improvement in detection accuracy and recall rate. Moreover, DS-YOLO keeps the number of model parameters and computational complexity within a certain range, 9.330M and 30.7 G, without introducing a lot of computational overhead.

Table 4
Experimental comparison between DS-YOLO and other models on VisDrone2019 datdaset.

Model	*P* (%)	*R* (%)	mAP@0.5 (%)	mAP@0.5-0.95 (%)	Params (M)	FLOPS (G)
YOLOv5m	44.9	36.5	35.5	19.9	20.907	48.0
YOLOv8s	50.3	37.0	38.1	22.7	11.129	28.5
YOLOv8s-P2	52.6	41.0	41.7	24.3	10.629	37.0
YOLOv8m	52.4	40.4	41.5	25.2	25.846	79.1
YOLOv9s	51.0	36.5	38.1	23.1	9.605	38.8
YOLOv10s	48.6	37.0	37.3	22.3	8.024	24.5
DS-YOLO	52.4	41.6	43.1	26.0	9.330	30.7

Table 5
Comparison of inference speed of different models on test set.

Datasets	Models	Speed (ms)
VisDrone2019	YOLOv8s-P2	35.8
	YOLOv8m	33.1
	YOLOv9s	49.6
	DS-YOLO	37.9

Table 5 shows the average inference time of the models. To be specific, on the VisDrone2019 dataset test set, the average detection time of DS-YOLO is 37.9 ms. Compared with YOLOv8s-P2 and YOLOv8m, the time is 2.1 ms and 4.8 ms longer, respectively. Compared to YOLOv9s, DS-YOLO's detection speed is 11.7 ms faster. Compared with other models, DS-YOLO has higher detection precision, and the detection speed can also meet the requirements of real-time.

Fig. 9 shows the models' mAP@0.5 in relation to the number of parameters and the computational complexity. As shown in the figure, DS-YOLO is located in the upper left corner of the image. This shows that the model can better balance the calculation cost and detection accuracy. DS-YOLO is more suitable for intensive small object detection tasks. And the required computing

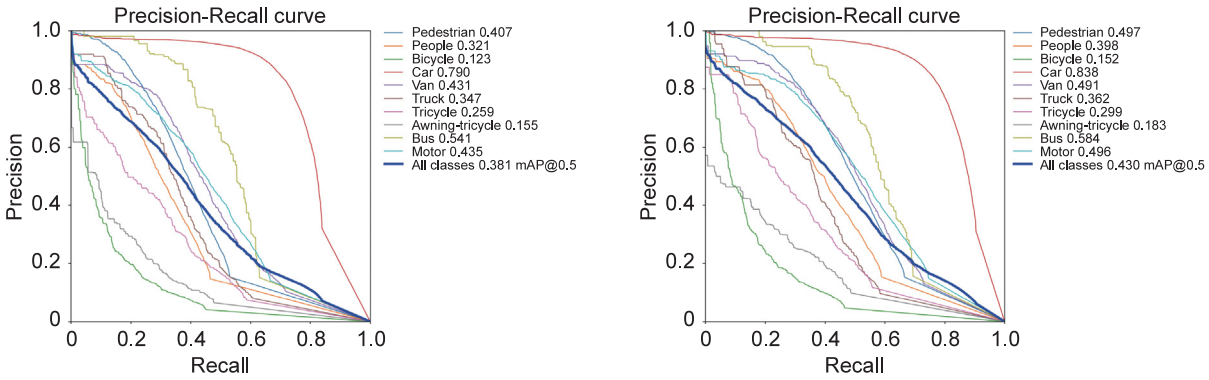

Fig. 8. P-R curves of YOLOv8s (left) and DS-YOLO (right) on VisDrone2019 dataset.

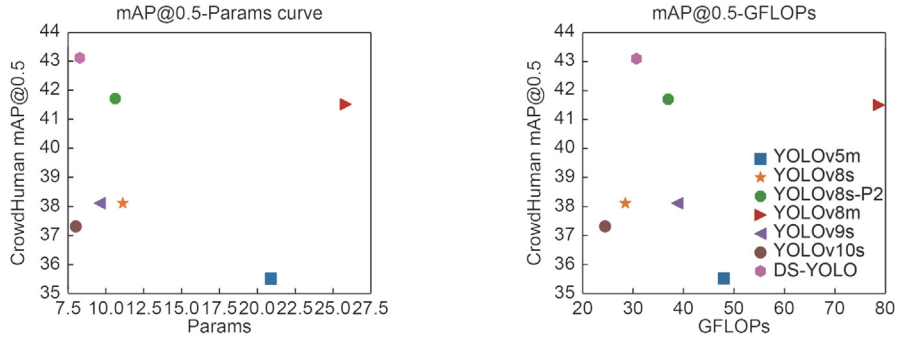

Fig. 9. mAP@0.5-Params and mAP@0.5-GFLOPs scatter diagram on VisDrone2019 dataset.

overhead is smaller, and thus the hardware requirements are lower.

4.6. Ablation experimental verification

To evaluate the extent to which each improvement contributed to the overall performance of the algorithm, ablation experiments were designed. The experimental results are shown in Tables 6 and 7. YOLOv8s[1] represents the main body lightweight design based on YOLOv8s and introduces C2fUIB, YOLOv8s[2] represents the introduction of the upsampling algorithm DySample and DO-C2f on the basis of YOLOv8s[1] for dense objects, and DS-YOLO further adopts a new neck design LFS-PAFPN and redesigns the backbone structure.

It can be found that the computational complexity of YOLOv8s[1] and YOLOv8s[2] decreases by around 11 GFLOPs, and the algorithm performance also decreases. On the CrowdHuman dataset, mAP@0.5 declined by 1.2% and Recall dropped by 1.8%. On the VisDrone2019 dataset, mAP@0.5 decreased by 0.5% and Recall fell by 0.3%. The performance deterioration of DS-YOLO is significantly less than FLOPS. This is mainly due to the introduction of lightweight modules DO-C2f and C2fUIB in the original structure, while reducing the model computation overhead and maintain detection accuracy. But, after the introduction of an improved backbone structure and LFS-PAFPN, the detection performance of DS-YOLO has been greatly enhanced. Higher detection accuracy is obtained while avoiding the introduction of a large amount of computation overhead. After introducing a enhanced backbone structure and LFS-PAFPN that are more friendly to small objects, the algorithm performance has been significantly improved. On CrowdHuman and VisDrone2019, mAP@0.5 has increased by 4.2% and 5%, and Recall has increased by 4.9% and 4.6%, respectively. The number of parameters has been reduced by more than 1M compared with the original algorithm. Moreover, the computational complexity has only increased by 2.3 GFLOPs. Higher detection performance is obtained at the cost of a smaller computational burden.

Table 6
Ablation experiment comparison of DS-YOLO on CrowdHuman dataset.

Model	*P* (%)	*R* (%)	mAP@0.5 (%)	mAP@0.5-0.95 (%)	Params (M)	GFLOPs
YOLOv8s	85.1	68.2	78.1	50.4	11.137	28.7
YOLOv8s[1]	84.5	66.4	76.9	49.1	10.036	17.7
YOLOv8s[2]	84.6	66.6	76.9	49.1	10.061	17.7
DS-YOLO	85.5	73.1	82.3	54.3	8.288	31.0

Table 7
Ablation experiment comparison of DS-YOLO on VisDrone2019 dataset.

Model	*P* (%)	*R* (%)	mAP@0.5 (%)	mAP@0.5-0.95 (%)	Params (M)	GFLOPs
YOLOv8s	50.3	37.0	38.1	22.7	11.129	28.5
YOLOv8s[1]	49.5	36.2	37.6	21.7	10.028	17.8
YOLOv8s[2]	49.8	36.3	37.8	22.0	10.018	17.7
DS-YOLO	52.4	41.6	43.1	26.0	9.330	30.7

4.7. Visual experiment on datasets

Tables 8 and 9 provide a statistical count of the number of categories on the test sets of CrowdHuman and VisDrone2019 datasets using the DS-YOLO and YOLOv8m, respectively. For instance, when examining the CrowdHuman dataset, we can observe a difference in the detection capabilities of two object detection models. Specifically, the statistical count for image 1 reveals that DS-YOLO outperforms YOLOv8m in terms of the number of objects detected. DS-YOLO identifies a total of 14 objects, which includes 58 instances of “head”, 69 instances of “full body”, and 67 instances of “visible body”. In contrast, YOLOv8m

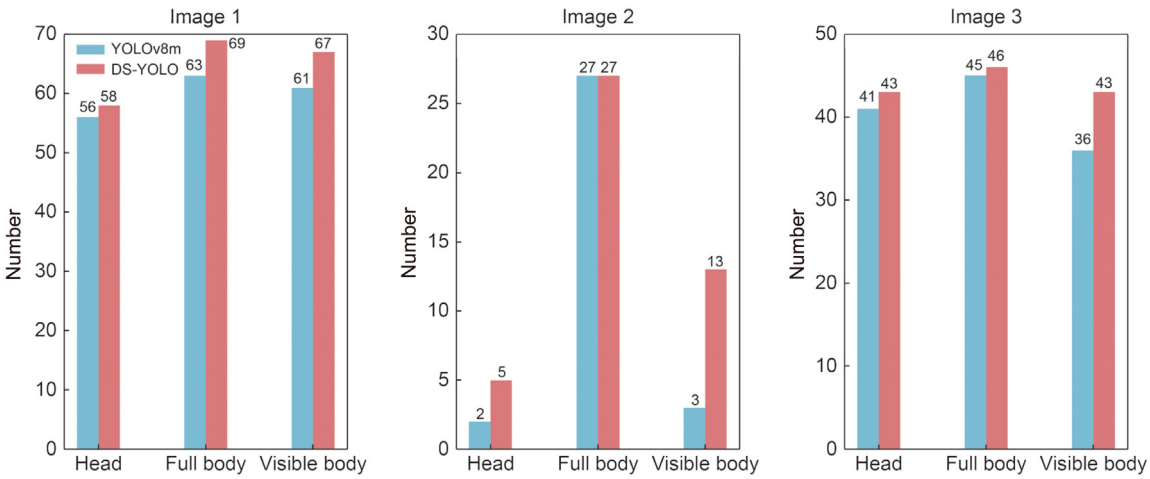

Fig. 10. Detection results comparison between DS-YOLO and YOLOv8m on CrowdHuman dataset.

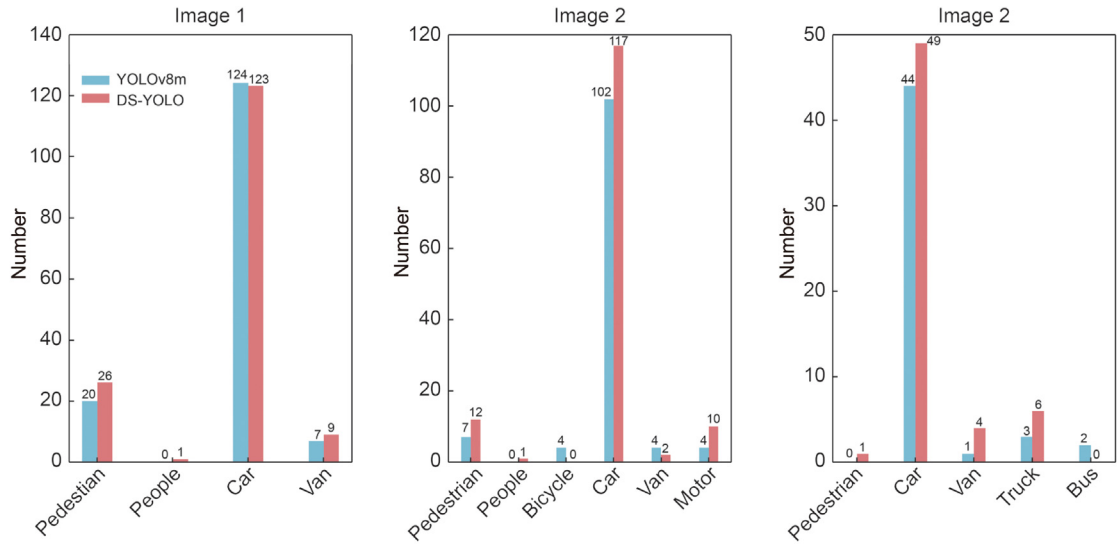

Fig. 11. Detection results comparison between DS-YOLO and YOLOv8m on VisDrone2019 dataset.

Table 8
Comparison of the number of objects detected by YOLOv8m and DS-YOLO on CrowdHuman dataset.

Input	Model	Speed (ms)	Class Name		
			Head	Full body	Visible body
Image 1	YOLOv8m	23.1	56	63	61
	DS-YOLO	27.4	58	69	67
Image 2	YOLOv8m	23.8	2	27	3
	DS-YOLO	26.4	5	27	13
Image 3	YOLOv8m	25.5	41	45	36
	DS-YOLO	27.2	43	46	43

detects fewer objects. This comparison highlights the superior performance of DS-YOLO in reducing the rate of missed detection. Figs. 10 and 11 provide a more intuitive representation of the detection performance of DS-YOLO, which detected more objects than YOLOv8m and effectively reduced the phenomenon of missed detection.

It could be concluded from the statistical data in Tables 8 and 9 that DS-YOLO has superior detection performance and can detect more objects in a dense and small object scenario. Figs. 12 and 13 show the detection effects of DS-YOLO in two scenarios. It is relatively satisfactory. Although the detection time of DS-YOLO is longer, the detection effect is better, and the detection speed can still meet the needs of use.

4.8. Realistic environment visual experiment

In order to further reflect the detection effect of the algorithm, a low-altitude campus video used to simulate the monitoring scene was used for testing. The video is captured by the drone and algorithm is run on its onboard computer. The specific detection effect is shown in the Fig. 14. As shown in the figure, although the image resolution is slightly lower and the objects in the image are small and dense, a large number of objects are still detected.

5. Summary and prospect

In conclusion, this paper presented a more advanced dense small object detection model for video surveillance in the security field, DS-YOLO. It could effectively address the challenges related to dense small object detection, including density, occlusion, and scale variations. Compared to the baseline model YOLOv8s, DS-YOLO maintains a lower computational burden and parameter count, and is equally efficient in the deployment of related edge devices.

Table 9
Comparison of the number of objects detected by YOLOv8m and DS-YOLO on VisDrone2019 dataset.

Input	Model	Speed (ms)	Class Name							
			Pedestrian	People	Bicycle	Car	Van	Truck	Bus	Motor
Image 1	YOLOv8m	25.4	20	–	–	124	7	–	–	–
	DS-YOLO	30.1	26	1	–	123	9	–	–	–
Image 2	YOLOv8m	25.3	7	0	4	102	4	–	–	4
	DS-YOLO	30.1	12	1	0	117	2	–	–	10
Image 3	YOLOv8m	25.5	–	–	–	44	1	3	2	–
	DS-YOLO	30.0	1	0	0	49	4	6	–	–

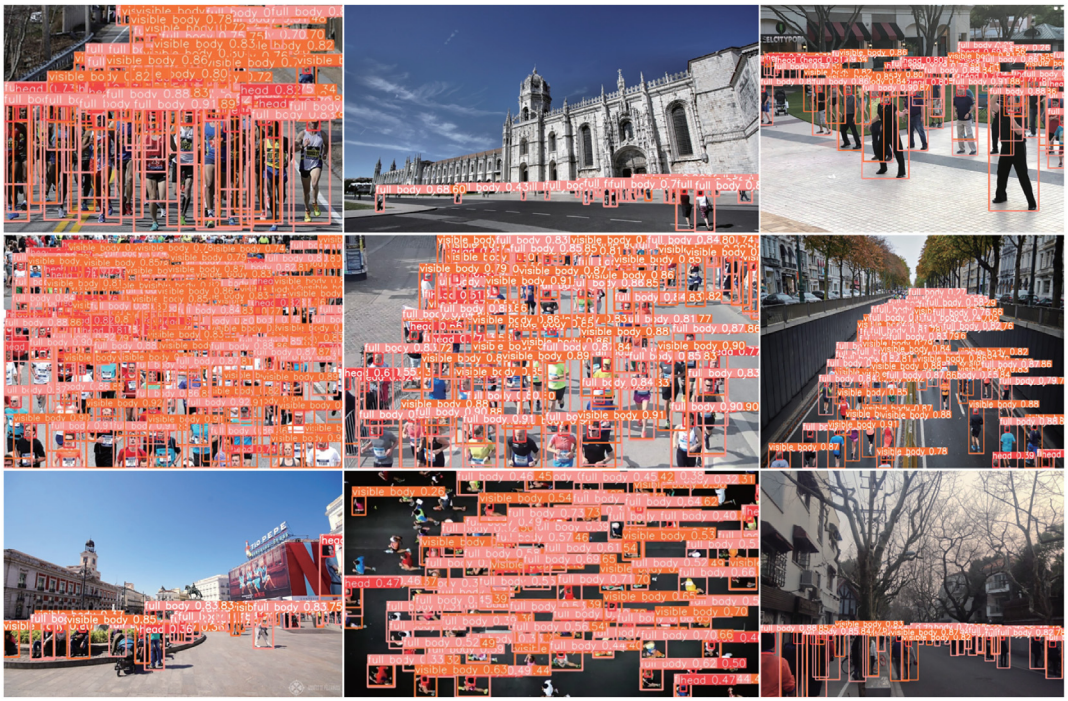

Fig. 12. DS-YOLO detection visualization results on CrowdHuman dataset.

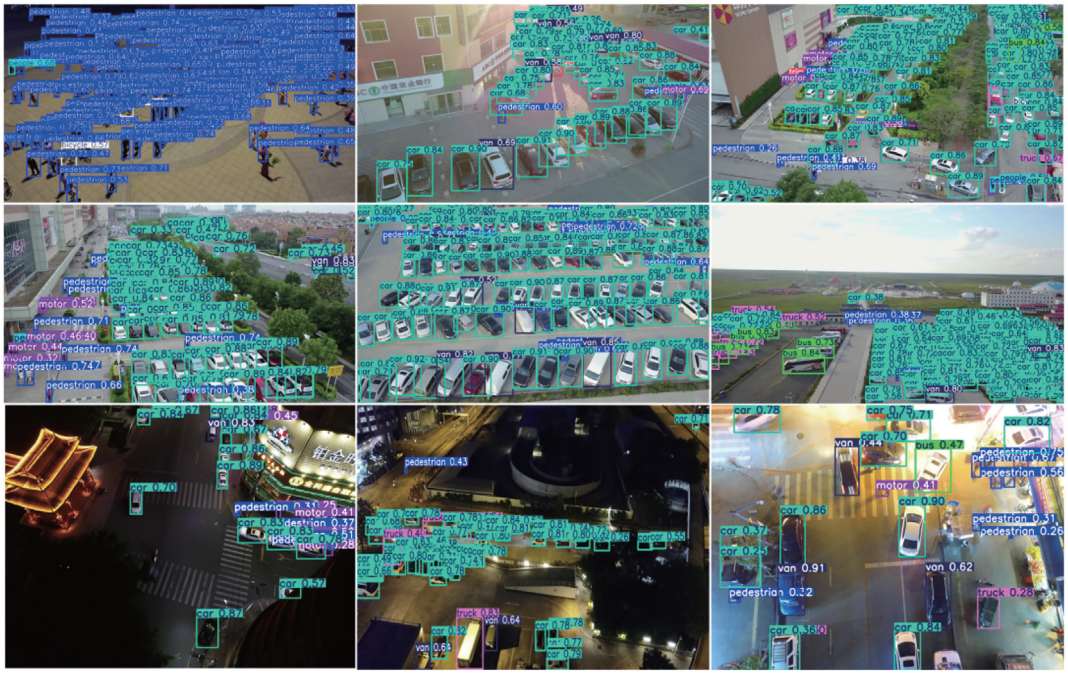

Fig. 13. DS-YOLO detection visualization results on VisDrone2019 dataset.

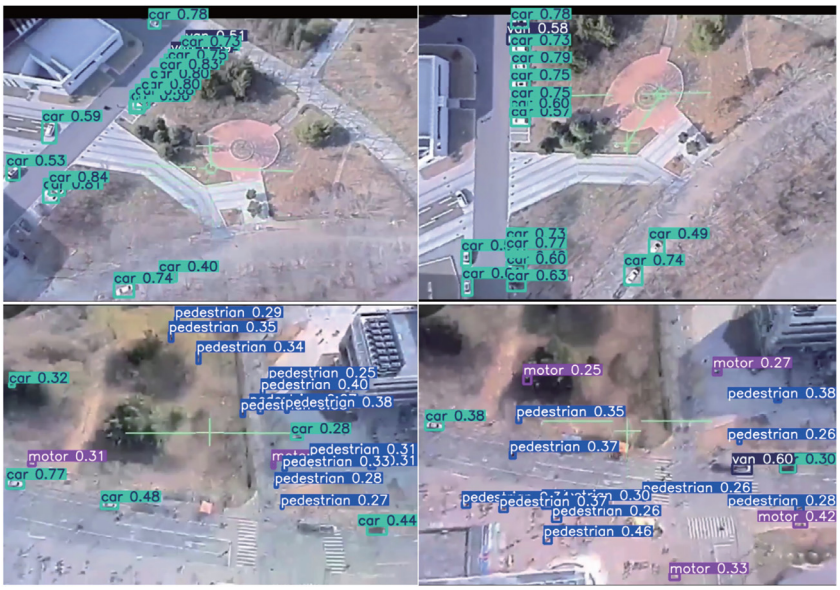

Fig. 14. DS-YOLO detection visualization results based on edge device of UAV.

Firstly, a backbone network with stronger dense small object feature extraction capability was designed, strengthening the utilization of shallow network information and avoiding a large increase in computational burden by rearranging the number of channels. At the same time, a lightweight C2fUIB module with a larger receptive field was introduced to fully capture the feature information of dense small objects. Secondly, LFS-PAFPN allowed the network to effectively detect multi-scale objects, and the performance of small object detection was greatly improved. Finally, for densely populated scenes, a DySample based on dynamic sampling strategy was introduced to reduce feature loss during the upsampling process. Considering the edge device versatility of the algorithm, each module of DS-YOLO had been lightweightly improved, so that the algorithm better balances the speed and accuracy of detection. DS-YOLO outperformed the baseline models YOLOv8s and the deeper YOLOv8m, with significant improvements in detection accuracy and overall performance. The research results highlighted the potential of DS-YOLO as an advanced dense and small object detection solution and had more practical application value in real-world scenarios.

Despite the algorithm's partial success in reducing missed detections caused by crowd density and occlusion, the model inference time remains lengthy, and the impact of various weather and lighting conditions on detection has not yet been investigated. In the future, efforts will continue to focus on designing lightweight models and detecting dense small objects in challenging environments. Training on more datasets will enhance model robustness. This aspect of work is also significant and will be further explored in the next steps.

CRediT authorship contribution statement

Hongyu Zhang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Data curation, Conceptualization. **Guoliang Li:** Writing – review & editing, Supervision, Resources, Investigation, Funding acquisition, Conceptualization. **Dapeng Wan:** Writing – review & editing, Validation, Supervision, Investigation. **Ziyue Wang:** Supervision, Methodology, Formal analysis, Data curation. **Jinshun Dong:** Writing – review & editing, Validation, Investigation. **Shoujun Lin:** Resources, Investigation. **Lixia Deng:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization. **Haiying Liu:** Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work has been supported by the Innovation Ability Enhancement Project of Shandong Province Science and Technology Small Medium Enterprises (2023TSCG0159 and 2022TSGC2175) and the Peiyou Fund of Qilu University of Technology (Shandong Academy of Sciences) (2023PY006).

References

[1] Redmon Joseph, Ali Farhadi, Yolov3: An incremental improvement, 2018, arXiv preprint arXiv:1804.02767.
[2] Glenn Jocher, Ultralytics YOLOv5, 2020, 7.0[EB/OL]. https://github.com/ultralytics/yolov5.
[3] Glenn Jocher, Ayush Chaurasia, Jing Qiu, Glenn jocher ayush chaurasia jing qiu ultralytics YOLOv8, 2023, 8.0.0[EB/OL]. https://github.com/ultralytics/ultralytics.
[4] Wei. Chen, Yuxuan. Zhu, Zijian. Tian, Fan. Zhang, Minda. Yao, Occlusion and multi-scale pedestrian detection a review, Array 19 (2023) 100318.
[5] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, Guiguang Ding, YOLOv10: Real-time end-to-end object detection, 2024, arXiv preprint arXiv:2405.14458.
[6] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 81, 2014, pp. 580–587.
[7] Ross Girshick, Fast R-CNN, in: IEEE International Conference on Computer Vision (ICCV), Vol. 169, 2015, pp. 1440–1448.
[8] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (06) (2017) 1137–1149.
[9] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, Baining Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 9992–10002.

[10] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, Jie Chen, Detrs beat yolos on real-time object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 16965–16974.
[11] Deepak Kumar Jain, Xudong Zhao, Salvador Garcia, Subramani Neelakandan, Robust multi-modal pedestrian detection using deep convolutional neural network with ensemble learning model, Expert Syst. Appl. 249 (2024) 123527.
[12] M. Xu, Z. Wang, X. Liu, L. Ma, A. Shehzad, An efficient pedestrian detection for realtime surveillance systems based on modified YOLOv3, IEEE J. Radio Freq. Identif. 6 (2022) 972–976.
[13] Weiyen Hsu, Peiyu Yang, Pedestrian detection using multi-scale structure-enhanced super-resolution, IEEE Trans. Intell. Transp. Syst. 24 (11) (2023) 12312–12322.
[14] Jinming Cao, Yangyan Li, Mingchao Sun, Ying Chen, Dani Lischinski, Daniel Cohen-Or, Baoquan Chen, Changhe Tu, Do-conv: Depthwise over-parameterized convolutional layer, IEEE Trans. Image Process. 31 (2022) 3726–3736.
[15] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, Mobilenetv2: Inverted residuals and linear bottlene-cks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.
[16] Kaiming. He, Xiangyu. Zhang, Shaoqing. Ren, Jian. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[17] Danfeng. Qin, Chas. Leichner, Manolis. Delakis, Marco. Fornoni, Shixin. Luo, Fan. Yang, Weijun. Wang, Colby. Banbury, Chengxi. Ye, Berkin. Akin, Vaibhav. Aggarwal, Tenghui. Zhu, Daniele. Moro, Andrew. Howard, MobileNetV4-universal models for the mobile ecosystem, 2024, arXiv preprint arXiv:2404.10518.
[18] Wenze. Liu, Hao. Lu, Hongtao. Fu, Zhiguo. Cao, Learning to upsample by learning to sample, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 6027–6037.
[19] Lihu Pan, Jianzhong Diao, Zhengkui Wang, Shouxin Peng, Cunhui Zhao, HF-YOLO: Advanced pedestrian detection model with feature fusion and imbalance resolution, Neural Process. Lett. 56 (2024) 90.