electronics

MDPI

Article

DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor

Haitong Lou [1], Xuehu Duan [1], Junmei Guo [1], Haiying Liu [1,*], Jason Gu [2], Lingyun Bi [1] and Haonan Chen [1]

[1] The School of Information and Automation Engineering, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250300, China; 10431210431@stu.qlu.edu.cn (X.D.); gjm@qlu.edu.cn (J.G.)
[2] The School of Electrical and Computer Engineering, Dalhousie University, Halifax, NS B3J 1Z1, Canada
* Correspondence: haiyingliu2019@qlu.edu.cn

Abstract: Traditional camera sensors rely on human eyes for observation. However, human eyes are prone to fatigue when observing objects of different sizes for a long time in complex scenes, and human cognition is limited, which often leads to judgment errors and greatly reduces efficiency. Object recognition technology is an important technology used to judge the object's category on a camera sensor. In order to solve this problem, a small-size object detection algorithm for special scenarios was proposed in this paper. The advantage of this algorithm is that it not only has higher precision for small-size object detection but also can ensure that the detection accuracy for each size is not lower than that of the existing algorithm. There are three main innovations in this paper, as follows: (1) A new downsampling method which could better preserve the context feature information is proposed. (2) The feature fusion network is improved to effectively combine shallow information and deep information. (3) A new network structure is proposed to effectively improve the detection accuracy of the model. From the point of view of detection accuracy, it is better than YOLOX, YOLOR, YOLOv3, scaled YOLOv5, YOLOv7-Tiny, and YOLOv8. Three authoritative public datasets are used in these experiments: (a) In the Visdron dataset (small-size objects), the map, precision, and recall ratios of DC-YOLOv8 are 2.5%, 1.9%, and 2.1% higher than those of YOLOv8s, respectively. (b) On the Tinyperson dataset (minimal-size objects), the map, precision, and recall ratios of DC-YOLOv8 are 1%, 0.2%, and 1.2% higher than those of YOLOv8s, respectively. (c) On the PASCAL VOC2007 dataset (normal-size objects), the map, precision, and recall ratios of DC-YOLOv8 are 0.5%, 0.3%, and 0.4% higher than those of YOLOv8s, respectively.

Keywords: small-size objects; object detection; camera sensor; feature fusion

Citation: Lou, H.; Duan, X.; Guo, J.; Liu, H.; Gu, J.; Bi, L.; Chen, H. DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor. *Electronics* **2023**, *12*, 2323. https://doi.org/10.3390/electronics12102323

Academic Editor: Donghyeon Cho

Received: 6 April 2023
Revised: 15 May 2023
Accepted: 16 May 2023
Published: 21 May 2023

1. Introduction

As one of the most widely used devices, cameras have been an essential device in various industries and families, such as robotics, monitoring, transportation, medicine, autonomous driving, and so on [1–5]. A camera sensor is one of the core sensors of the above requirements; it is composed of a lens, a lens module, a filter, a CMOS (complementary metal oxide semiconductor)/CCD (charge-coupled device), ISP (image signal processing), and a data transmission part. It works by first collecting images using optical imaging principles and finally performing image signal processing. The application of cameras in traffic, medicine, automatic driving, etc., is crucial to accurately identify an object, so the object recognition algorithm is one of the most important parts in a camera sensor.

Traditional video cameras capture a scene and present it on a screen; then, the shape and type of the object are observed and judged by the human eye. However, human cognitive ability is limited, and it is difficult to judge the category of the object when the camera resolution is too low. A complex scene will also strain the human eye, resulting in the inability to detect some small details. A viable alternative to this problem is to use camera sensors to find areas and categories of interest [6].

At present, technology for object recognition through a camera is one of the most challenging topics, and accuracy and real-time performance are the most important indicators applied in a camera sensor. In recent years, with the ultimate goal of achieving accuracy or being used in real time, MobileNet [7–9], ShuffleNet [10,11], etc., which can be used on a CPU, and ResNet [12], DarkNet [13], etc., which can be used on a GPU, have been proposed by researchers.

At this stage, the most classical object detection algorithms are divided into two kinds: two-stage object detection algorithms and one-stage object detection algorithms. Two-stage object detection algorithms include R-CNN (Region-based Convolutional Neural Network) [14], Fast R-CNN [15], Faster R-CNN [16], Mask R-CNN [17], etc. One-stage object detection algorithms include YOLO series algorithms (you only look once) [13,18–22], SSD algorithms (Single Shot MultiBox Detector) [23], and so on. The YOLO series of algorithms is one of the fastest growing and best algorithms so far, especially the novel YOLOv8 algorithm released in 2023, which has reached the highest accuracy so far. However, YOLO only solves for object of full sizes. When the project becomes a special scene with a special size, its performance is not as good as some current small-size object detection algorithms [24,25]. In order to solve this problem, this paper proposed an improved algorithm for YOLOv8. The detection accuracy of this algorithm had a stable small improvement for normal-scale objects and greatly improved the detection accuracy of small objects in complex scenes. The pixels of small objects are small, which make the detector extract features accurately and comprehensively during feature extraction. Especially in complex scenes such as object overlap, it is more difficult to extract information, so the accuracy of various algorithms for small objects is generally low. Greatly improving the detection accuracy of small objects in complex scenes while the detection accuracy of normal-scale objects remains stable or shows slight improvement, the main contributions of the proposed algorithm are as follows:

(a) The MDC module is proposed to perform downsampling operations (the method of concatenating depth-wise separable convolutions, maxpool, and convolutions of dimension size 3×3 with stride = 2 is presented). It can supplement the information lost by each module in the downsampling process, making the contextual information saved in the feature extraction process more complete.
(b) The C2f module in front of the detector in YOLOv8 is replaced by the DC module proposed in this paper (the network structure formed by stacking depth-wise separable convolution and ordinary convolution). A new network structure is formed by stacking DC modules and fusing each small module continuously. It increases the depth of the whole structure, achieves higher resolution without significant computational cost, and is able to capture more contextual information.
(c) The feature fusion method of YOLOv8 is improved, which could perfectly combine shallow information and deep information, make the information retained during network feature extraction more comprehensive, and solves the problem of missed detection due to inaccurate positioning.

This paper is divided into the following parts: Section 2 introduces the reasons for choosing YOLOv8 as the baseline and the main idea of YOLOv8; Section 3 mainly introduces the improved method of this paper; Section 4 focuses on the experimental results and comparative experiments; Section 5 provides the conclusions and directions of subsequent work and improvement.

2. Related Work

Currently, camera sensors are crucial and have been widely used in real life. Existing researchers also applied a large number of camera sensors to a variety of different scenarios. For example, Zou et al. proposed a new camera-sensor-based obstacle detection method for day and night on a traditional excavator based on a camera sensor [1]. Additionally, robust multi-target tracking with camera sensor fusion based on both a camera sensor and object detection has been proposed by Sengupta et al [26]. There is also the

camera-sensor approach proposed by Bharati applied to assisted navigation for people with visual impairments [27]. However, in order to be applicable to real life. It can ensure real-time detection was the most important indicator, so we used the most popular one-stage algorithm. The YOLO family of algorithms is the state of the art for real-time performance.

2.1. The Reason for Choosing YOLOv8 as the Baseline

This section introduces the most popular algorithms in recent years and describes in detail some main contents of this paper for YOLOv8 improvement.

YOLO is currently the most popular real-time object detector and can be widely accepted for the following reasons: (a) lightweight network architecture, (b) effective feature fusion methods, (c) and more accurate detection results.

In terms of current usage, YOLOv5 and YOLOv7 are the two most widely accepted algorithms. Deep learning technology to achieve real-time and efficient object detection tasks is used in YOLOv5. Compared with its predecessor YOLOv4, YOLOv5 had been improved in terms of model structure, training strategy, and performance. The CSP (Cross-Stage Partial) network structure was adopted by YOLOv5, which could effectively reduce repeated calculations and improve computational efficiency. However, YOLOv5 also has some drawbacks. For example, it still has some shortcomings in small object detection, and the detection effect of dense objects also needs to be improved. Additionally, the performance of YOLOv5 in complex situations such as occlusion and pose change still needs to be strengthened.

YOLOv7 proposed a novel training strategy, called Trainable Bag of Freebies (TBoF), for improving the performance of real-time object detectors. The TBoF method included a series of trainable tricks, such as data augmentation, MixUp, etc., which could significantly improve the accuracy and generalization ability of the object detector by applying TBoF to three different types of object detectors (SSD, RetinaNet, and YOLOv3). However, YOLOv7 is also limited by the training data, model structure, and hyperparameters, which leads to performance degradation in some cases. In addition, the proposed method requires more computational resources and training time to achieve the best performance.

YOLOv8, published in 2023, aimed to combine the best of many real-time object detectors. It still adopted the idea of CSP in YOLOv5 [28], feature fusion method (PAN-FPN) [29,30], and SPPF module. Its main improvements were the following: (a) It provided a brand new SOTA model, including P5 640 and P6 1280 resolution object detection networks and YOLACT's instance segmentation model [31]. In order to meet the needs of different projects, it also designed models of different scales based on the scaling coefficient similar to YOLOv5. (b) On the premise of retaining the original idea of YOLOv5, the C2f module was designed by referring to the ELAN structure in YOLOv7 [22]. (c) The detection head part also used the current popular method (separating the classification and detection heads) [32]. Most of the other parts were still based on the original idea of YOLOv5. (d) YOLOv8 classification loss used BCE loss. The regression Loss was of the form CIOU loss + DFL, and VFL proposed an asymmetric weighting operation [33]. DFL: The position of the box was modeled as a general distribution. The network quickly focused on the distribution of the location close to the object location, and the probability density was as near the location as possible, as shown in Equation (1). s_i is the output of sigmod for the network, y_i and y_{i+1} are interval orders, and y is a label. Compared with the previous YOLO algorithm, YOLOv8 is very extensible. It is a framework that can support previous versions of YOLO and can switch between different versions, so it is easy to compare the performance of different versions.

$$DFL_{(s_i,s_{i+1})} = -((y_{i+1} - y)\log(s_i) + (y - y_i)\log(s_{i+1})) \quad (1)$$

YOLOv8 uses Anchor-Free instead of Anchor-Base. V8 used dynamic TaskAlignedAssigner for matching strategy. It calculates the alignment degree of Anchor-level for each instance using Equation (2), s is the classification score, u is the IOU value, and α and β are the weight hyperparameters. It selects m anchors with the maximum value (t) in each

instance as positive samples and selects the other anchors as negative samples, and then trains through the loss function. After the above improvements, YOLOv8 is 1% more accurate than YOLOv5, making it the most accurate detector so far.

$$t = s^{\alpha} \times u^{\beta} \tag{2}$$

The key feature of YOLOv8 is that it is extensible. YOLOv8 is designed to work with all versions of YOLO and to switch between them, making it easy to compare their performances, which provides a great benefit to researchers working on YOLO projects. Therefore, the version YOLOv8 was selected as the baseline.

2.2. The Network Structure of YOLOv8

The backbone part of YOLOv8 is basically the same as that of YOLOv5, and the C3 module is replaced by the C2f module based on the CSP idea. The C2f module learned from the ELAN idea in YOLOv7 and combined C3 and ELAN to form the C2f module [22], so that YOLOv8 could obtain more abundant gradient flow information while ensuring its light weight. At the end of backbone, the most popular SPPF module was still used, and three Maxpools of size 5×5 were passed serially, and then, each layer was concatenation, so as to guarantee the accuracy of objects in various scales while ensuring a light weight simultaneously.

In the neck part, the feature fusion method used by YOLOv8 is still PAN-FPN, which strengthens the fusion and utilization of feature layer information at different scales. The authors of YOLOv8 used two upsampling and multiple C2f modules together with the final decoupled head structure to compose the neck module. The idea of decoupling the head in YOLOx, was used by YOLOv8 for the last part of the neck. It combined confidence and regression boxes to achieve a new level of accuracy.

YOLOv8 can support all versions of YOLO and can switch between different versions at will. It can also run on various hardware platforms (CPU-GPU), giving it strong flexibility. The YOLOv8 network architecture diagramias shown in Figure 1. The CBS in Figure 1 is composed of convolution, batch normalization, and SiLu activation functions.

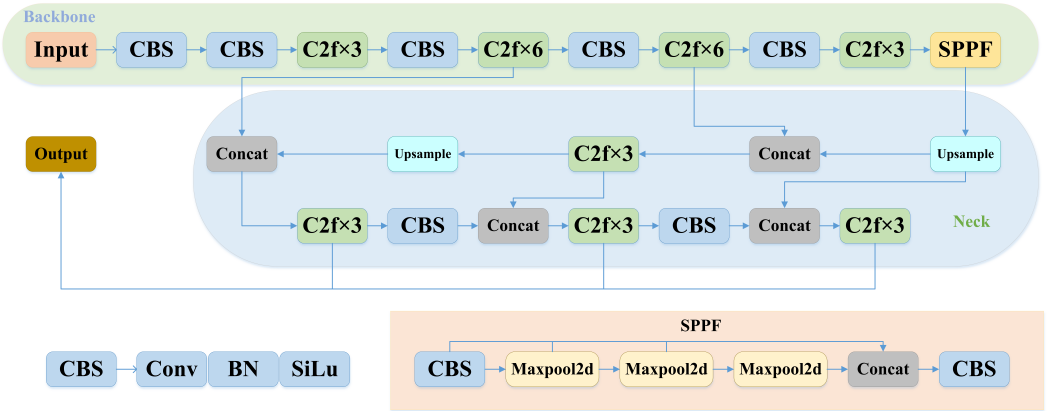


Figure 1. The network structure of yolov8.

3. The Proposed DC-YOLOv8 Algorithm

YOLOv8 has been very perfect in all aspects, but there are still some problems in the identification of small objects in complex scenes. The reasons for the inaccurate detection of small objects are analyzed as follows: (a) When the neural network performs feature extraction, small-size objects are misled by large-size objects, and the features extracted at a deep level lack a lot of small-size object information, which leads to the neglect of small objects in the whole learning process, so the detection effect is poor. (b) Compared with normal size, small-size objects are more easily overlapped by other objects and are easily

partially blocked by other size objects, making it difficult to distinguish and locate in an image.

In order to solve the above problems, we proposed a detection algorithm that could greatly improve the detection effect of small-size objects on the basis of ensuring the detection effect of normal-size objects. First, we proposed the MDC module for the downsampling operation, which adopted depth separable convolution, Maxpool and a convolution of size 3×3 with a stride = 2 for concatenation. This can fully supplement the information lost in the downsampling process of each item and can more completely preserve the contextual information of the feature extraction. Secondly, the feature fusion method was improved to better combine shallow information and deep information, so that the information retained in the process of network feature extraction was more comprehensive, and the problem of not detecting the object due to inaccurate positioning and being misled by large-size targets was solved. Finally, a DC module (depth-wise separable convolution + convolution of size 3×3) that was constantly stacked and fused was proposed to form a new network structure, and it was replaced by the C2f module in front of the detection head. This method increases the depth of the whole structure and obtains higher resolution without increasing significant computational cost. It can capture more contextual information and effectively improve the problem of a low detection accuracy caused by object overlap.

3.1. A Modified Efficient Downsampling Method

The downsampling method used in this paper mainly contains three parts, which are Maxpool, depth-wise separable convolution, and convolution module of size 3×3.

Common downsampling methods generally include a separate 3×3 convolutional module or Maxpool. Maxpool can alleviate the over-sensitivity of the convolutional layer to the location, which can better improve the robustness of the target. However, Maxpool will filter out the information that it thinks is not important when performing downsampling, and information about small-size targets can be easily misled or even masked by the information about large-size targets. Therefore, when Maxpool is carried out, part of the target information will be automatically filtered out, and only the important information that is considered by itself will be left, which reduces the resolution and has a great impact on the final accuracy. To keep it lightweight. Google proposed MobileNet and depth-wise separable convolution (DW) for the first time. DW convolution has a smaller volume and less calculations, requiring only one third of the calculation of regular convolution in the training process. However, it also loses a lot of useful information when reducing the amount of calculation needed. The convolution module of the size 3×3 has a high degree of nonlinearity and can represent more complex functions. Small convolution kernels can extract small features, so every researcher is willing to use ordinary convolution for downsampling operation. However, during the whole process of network extraction, there are many downsampling operations used, so the amount of calculation is not ignored.

Therefore, this paper first used the convolution of size 1×1 for dimensionality reduction and then used the convolution of size 3×3 for downsampling, which reduced a lot of calculation. During this operation, the Maxpool layer and depth-wise separable convolution were concatenated. This can fully supplement the information lost in the downsampling process of each item, and can more completely preserve the context information during feature extraction. After many experiments, it was proved that the MDC module was more efficient than the downsampling method of YOLOv8 original. The specific structure is shown in the Figure 2.

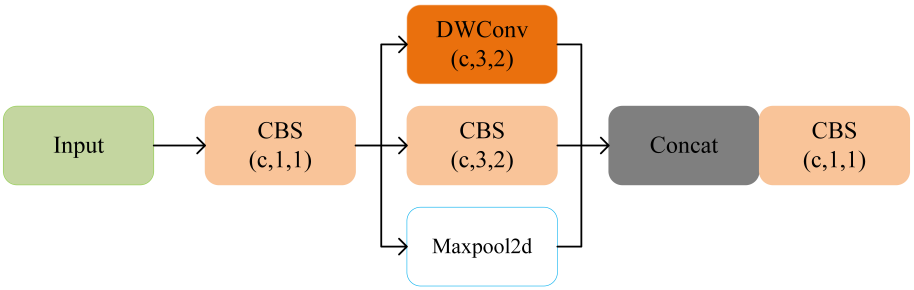
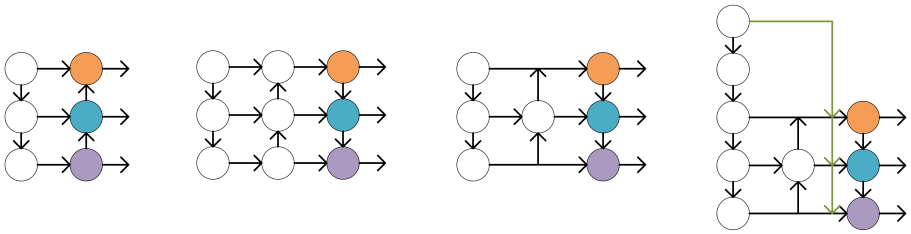


Figure 2. Downsampling method.

3.2. Improved Feature Fusion Method

When feature extraction is carried out in special scenarios, small-size targets are easily misled by normal-size targets, resulting in less and less information until it disappears completely. Moreover, the problem of poor target positioning in the whole network has always existed. Therefore, we improved the feature fusion method.

Observing the entire network structure diagram of YOLOv8, it can be seen that although the feature fusion method has retained both shallow information and deep information, the target positioning information mainly exists in the shallower position. During feature extraction, information that is not obvious is automatically deleted by the network, so a lot of important information for small-size objects is deleted in the shallowest layer, which is also the main reason for the poor detection of small objects. The improved feature fusion method in this paper focused on solving this problem. In the first layer of feature extraction, the size of the picture was changed to 80 × 80, 40 × 40, and 20 × 20 by Maxpool. It was then concatenated with the outputs of different scales separately. The reason we used Maxpool for downsampling is that Maxpool can extract the main location information and filter out other useless information during downsampling and has a very low amount of calculation. Through the feature extraction of each layer, what is missing is the most original feature information, so only using Maxpool for the downsampling operation can meet the needs of this paper. The specific structure is shown in Figure 3.



(**a**) FPN (**b**) PAN (**c**) Yolov8 (**d**) DCS-Yolov8

Figure 3. In the figure, orange represents the detector for small objects, blue represents the medium size object detector, and purple represents the large size object detector. (**a**) FPN fuses feature maps from different levels to form a feature pyramid with multi-scale information by constructing a top-down feature pyramid and adds lateral connections between feature maps at different levels in order to better utilize high-frequency detail information contained in low-level features. (**b**) PAN aggregates features from different scales to produce feature maps containing multiple resolutions by adding a path aggregation module between the encoder and decoder. This hierarchical feature pyramid structure can make full use of feature information at different scales, thereby improving the accuracy of semantic segmentation. (**c**) YOLOv8 adopts the combination of SPP and PAN, which improves the accuracy and efficiency of object detection. (**d**) The feature fusion method proposed in this paper.

3.3. The Proposed Network Structure

In order to solve the problem of losing a lot of important information due to being misled by large-size objects during the feature extraction process, a deeper network architecture (DC) was proposed in this paper. The DC module adopts the ideas of DenseNet and VOVNet [34,35]. It gathers each important module in the last layer at the same time and gathers important information from the previous layers in each layer, which can thus avoid the problem of information loss and ensure that normal-size information and small-size information can be well preserved. However, during the experiment, we found that the results are not as good as we imagined. After many experiments, the previous feature extraction was found to use a single convolution module, and the convolution module cannot extract complete information well. The parallel concatenation of convolution and depth-wise separable convolution can learn from each other and improve the learning ability and stability of the network.

Via the experiment, the most stable state could be achieved by replacing the C2f module in front of the head detector with the DC module. The specific structure is shown in the Figure 4.

After the improvement in the above three improved methods, the new network learning ability was greatly improved and enhanced. The DC-YOLOv8 network structure is shown in Figure 5.

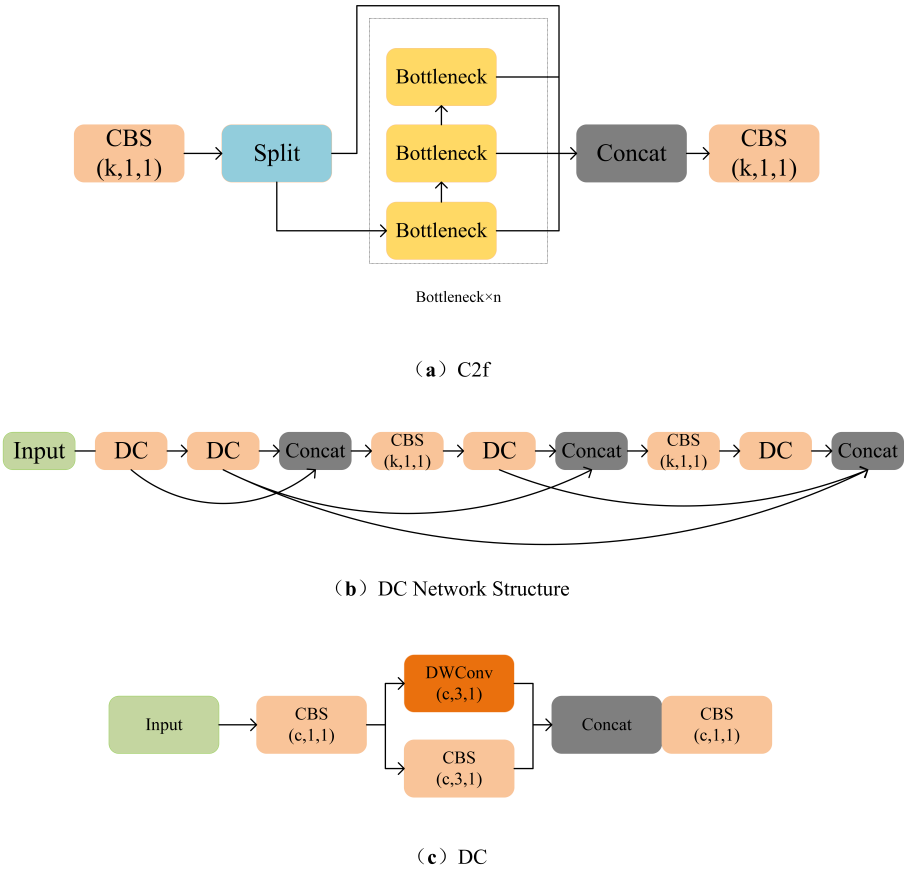


Figure 4. (**a**) The C2f module, which is designed by referring to the idea of the C3 module and ELAN, so that YOLOv8 can obtain more abundant gradient flow information while ensuring light weight. (**b**) The network structure proposed in this paper. It not only adopts the ideas of DenseNet and VOVNet but also replaces the original convolution with a parallel cascade of convolutions and depth-wise separable convolutions. (**c**) The basic block in the network architecture, which is composed of convolutions and depth-wise separable convolutions.

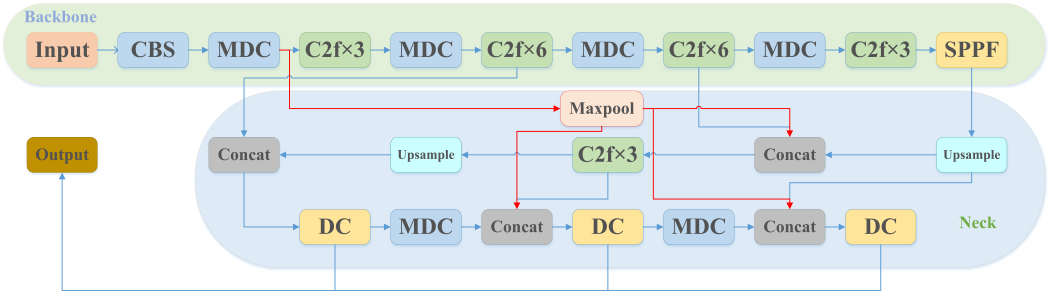


Figure 5. Network structure diagram of DC-YOLOv8.

4. Experiments

The new algorithm was trained and tested on the Visdrone dataset to improve each stage and compared with YOLOv8. In order to verify that this algorithm could improve the detection accuracy of small-size targets without reducing the accuracy of other scale targets, comparative experiments was carried out on the PASCAL VOC2007 dataset and the Tinperson dataset. Finally, we selected complex scene pictures in different scenarios to compare the detection effects of the proposed algorithm and YOLOv8 algorithm in actual scenes.

After many experiments, it can be known that the algorithm basically iterates 120 times and then begins to converge. According to the hardware facilities and multiple experimental attempts, we set the following parameters: batch size = 8 and epoch = 200.

4.1. Experimental Platform

The system used for the experiments in this paper is Windows 11, and the system hardware facilities were 16G RAM, NVIDIA GTX3070 GPU, and Intel i512400f CPU. The software platform was torch 1.12.1 + cu113, Anaconda.

4.2. Valuation Index

The evaluation metrics included mean average precision (mAP), average precision (AP), precision (P), and recall (R). The formulas for P and R are as shown in Equations (3) and (4).

$$P = \frac{TP}{(TP + FP)} \tag{3}$$

$$R = \frac{TP}{(TP + FN)} \tag{4}$$

TP is the number of correctly predicted bounding boxes, FP is the number of incorrectly judged positive samples, and FN is the number of undetected targets.

Average precision (AP) is the average accuracy of the model. Mean average precision (mAP) is the average value of the AP. k is the number of categories. The formulas for AP and mAP are as shown in Equations (5) and (6).

$$AP = \int_0^1 p(r)dr \tag{5}$$

$$mAP = \frac{1}{k}\sum_{i=1}^{k} AP_i \tag{6}$$

4.3. Experimental Result Analysis

In order to verify the detection effect of the improved method in this paper on small-size targets at each stage, we conducted ablation experiments on each stage in the Visdrone dataset and compared it with YOLO v8s. The Visdrone dataset was collected by the

AISKYEYE team at Tianjin University, China. This dataset is acquired by a UAV and has a wide coverage; it is collected in different scenes, weather, and light, so there are numerous small-size targets in complex environments. This dataset also provides some attributes such as scene visibility, object class, and occlusion. The Visdron dataset is extensive and authoritative, and this dataset conforms to the content studied in this experiment in all aspects. So, we used this dataset for control experiments.

In order to clearly show the authenticity of the experiment, this experiment used mAP0.5 and mAP0.5:0.9 as the evaluation index. The test results are shown in Table 1.

Table 1. Algorithm comparison at each stage.

Detection Algorithm	Module			Result			
	MDC	Feature Fusion	DC	mAP0.5	mAP0.5:0.95	P	R
YOLOv8				39	23.2	50.8	38
DC-YOLOv8	✓			39.5	23.5	51.2	38.8
DC-YOLOv8	✓	✓		40.3	24.1	51.8	39.4
DC-YOLOv8	✓	✓	✓	41.5	24.7	52.7	40.1

Table 1 showed that for the detection of small-size targets in complex scenes, the improved algorithm has a certain improvement at each stage. Additionally, the recall rate is improved by 2%, which means that there is a lot of room for improvement. It can be proved that the three methods improved in this experiment are obviously effective: (a) The improvement in the downsampling method can fully supplement the information lost in the downsampling process and can save on context information during feature extraction more completely. (b) The improvement in the feature fusion method effectively prevented the problem of small targets being ignored in the whole learning process due to location information. (c) The improvement in the network structure effectively solved the problem of losing a lot of important information due to being misled by large-size objects in the feature extraction process. The experimental results showed that improvement in the algorithm at each stage can improve the learning ability of the model.

In order to compare the detection effect of different types of objects in the DC-YOLOv8 algorithm, we recorded the mAP of 10 kinds of objects in the Visdrone dataset, and the specific results are shown in Figure 6. From the results, we can see that there are four categories in which the recognition accuracy is higher than the average level of the whole dataset. The modified algorithm shows steady improvement with larger objects such as cars and great improvement with smaller objects such as tricycles, bicycles, awning-tricycles, etc.

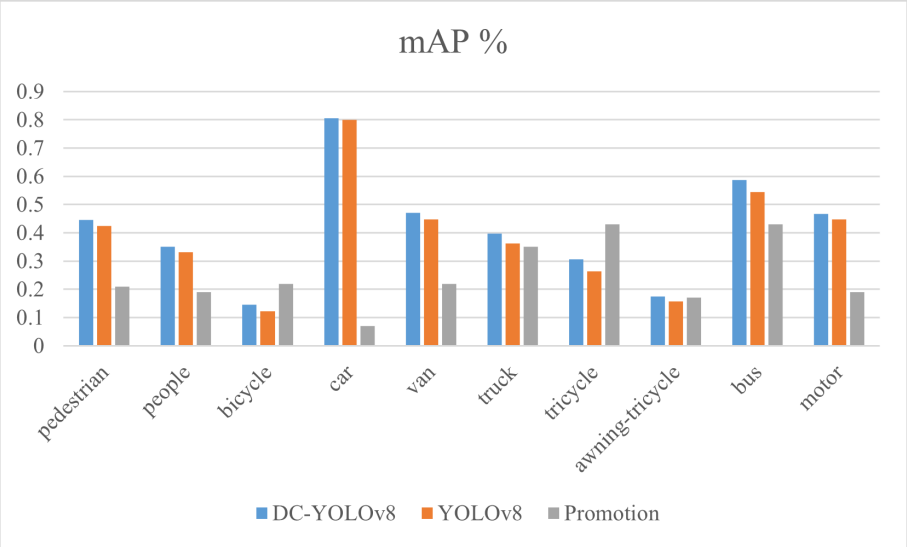


Figure 6. Comparing the 10 categories of YOLOv8 and DC-YOLOv8: blue is the result of DC-YOLOv8 proposed in this paper, orange is the result of YOLOv8, and gray is the accuracy of the difference between the two algorithms.

4.4. Comparison of Experiments with Different Sized Objects

The second set of experiments was a comparison experiment of different sizes, and the datasets used were the Pascal VOC2007 and Tinyperson datasets. Pascal VOC2007 is one of the most authoritative datasets, and its object types are divided into four categories—vehicle, household, animal, person and twenty small types. The resolution of the Tinyperson dataset is very low, basically less than 20 pixels, and it is used for small target detection at long distances and for complex backgrounds. There are two categories in the Tinyperson dataset: sea person and earth person. Each algorithm (YOLOv3, YOLOv5, YOLOv7, YOLOv8, and DC-YOLOv8) was tested on three datasets at the same time and recorded for comparison, as shown in the Table 2. The total number of iterations was set to 200 rounds and their mAP0.5 and mAP0.5:0.95 were recorded. From Table 2, we can conclude that the experimental results of DC-YOLOv8 are significantly higher than those of other classical algorithms in the experiments with small-size targets and even for extremely small-size targets, and DC-YOLOv8 also shows slightly higher performance than other algorithms in the experiments with normal-size targets. In order to facilitate subsequent verification, the weight file was saved with the highest mAP value during the experiment.

Table 2. Comparison of algorithms on different dataset.s

Datasets	Result	YOLOv3	YOLOv5	YOLOv7	YOLOv8	DC-YOLOv8
Visdrone	mAP0.5	38.8	38.1	30.7	39	**41.5**
	mAP0.5:0.95	21.6	21.7	20.4	23.2	**24.7**
VOC	mAP0.5	79.5	78	69.1	83.1	**83.5**
	mAP0.5:0.95	53.1	51.6	42.4	63	**64.3**
Tinyperson	mAP0.5	18.5	18.3	16.9	18.1	**19.1**
	mAP0.5:0.95	5.79	5.81	5.00	6.59	**7.02**

The reasons why the DC-YOLOv8 algorithm performs better than other algorithms were analyzed: (a) Most of the feature fusion methods used by classical algorithms are FPN + PAN, and small-size targets are easy to be misled by normal-size targets when extracting features layer by layer, resulting in loss of most information. The feature fusion method of DC-YOLOv8 fuses shallow information in the final information result well and effectively

avoids the problem of information loss in shallow layers. (b) Unimportant information will be automatically ignored in feature extraction, and small-size target pixels will be ignored when extracting features, resulting in reduced accuracy. However, DC-YOLOv8 adopts the ideas of DenseNet and VOVNet, which use deeper networks and are able to learn more detailed information. The class loss of DC-YOLOv8 and YOLOv8 is shown in Figure 7. We could see from the Figure 7 that DC-YOLOv8 had a certain degree of reduction in three category losses: box loss, class loss, and dfl loss.

In order to intuitively see the detection effect of DC-YOLOv8, two sets of complex scenes' graphs were selected for testing. The weight files with the highest accuracies using DC-YOLOv8 and YOLOv8 were retained and used for test comparison. The image selection criteria included complex scenes with targets of various sizes and overlap. The differences between DC-YOLOv8 and YOLOv8 can be clearly seen based on the above requirements.

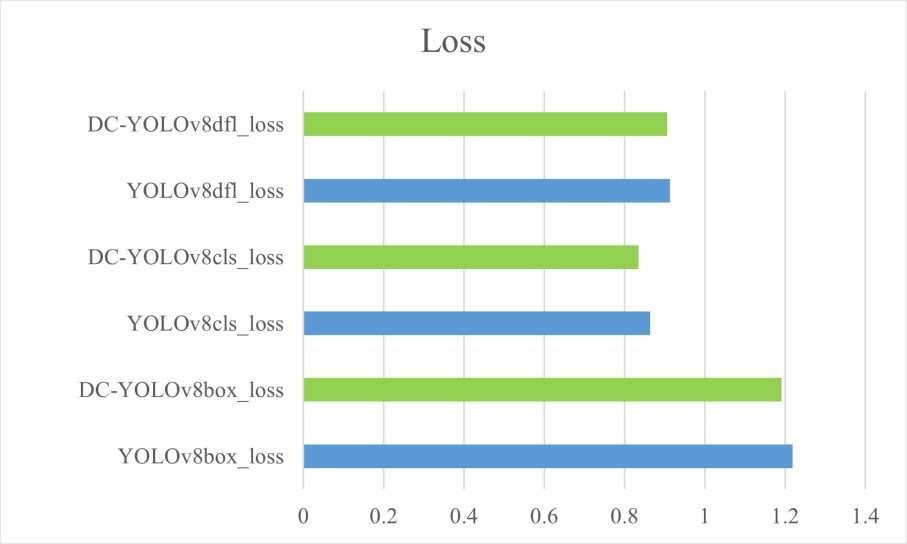
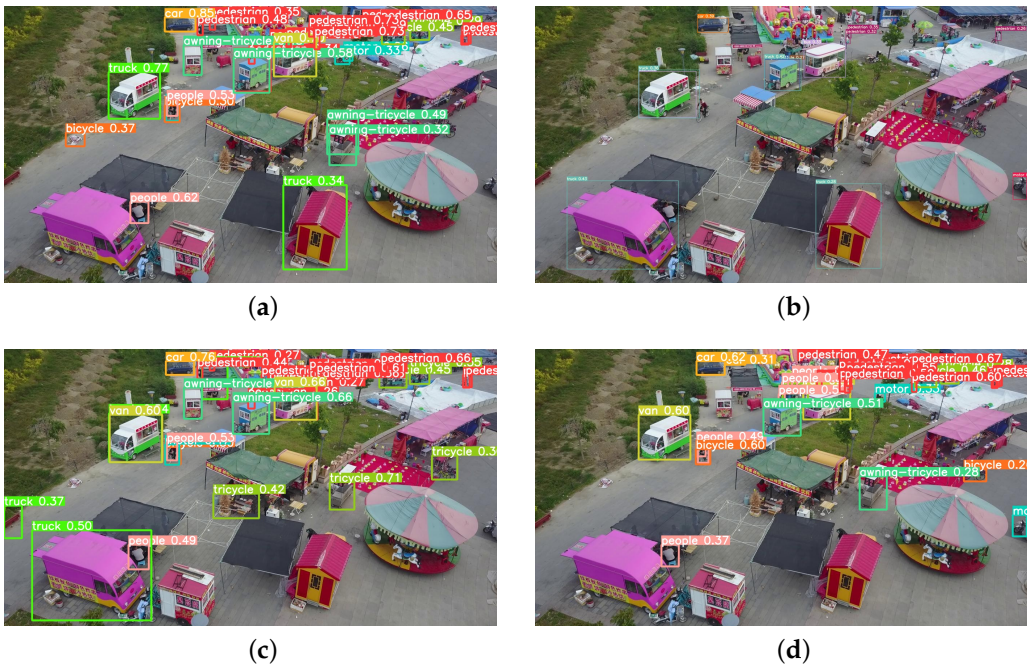


Figure 7. Comparison of class loss between DC-YOLOv8 and YOLOv8.

Among them, Figure 8 shows the comparison of a complex life scene (the highest weight file of the Visdrone dataset is used). Figure 9 shows the detection comparison of normal-sized objects (the highest weight file of the Pascal VOC2007 dataset is used). It can be seen from Figures 8 to 9 that DC-YOLOv8 has both a higher number of detected targets and higher accuracy of detected targets than YOLOv8.

In the first group of comparison experiments, images in the Visdrone dataset with complex scenes, more interference, and overlap were selected. Figure 8 shows that YOLOv8 falsely detected the leftmost part of the picture due to its dark color. False detections occurred at the middle tent, mainly due to overlapping objects. The right bike was misdetected due to the complex and overlapping environment nearby. On the far right, it was not detected because of incomplete information. In the middle, false detection occurred because of overlap. It can be seen that although YOLOv8 has many advantages, there are still some problems with small-size targets. In contrast, DC-YOLOv8 can accurately detect the right target when only partial information is available and can accurately detect the target in complex and overlapping scenes without false detections or missed detections. It can be seen from Figure 8 that the detection effect of DC-YOLOv8 is better than that of YOLOv8 when the target size is small.

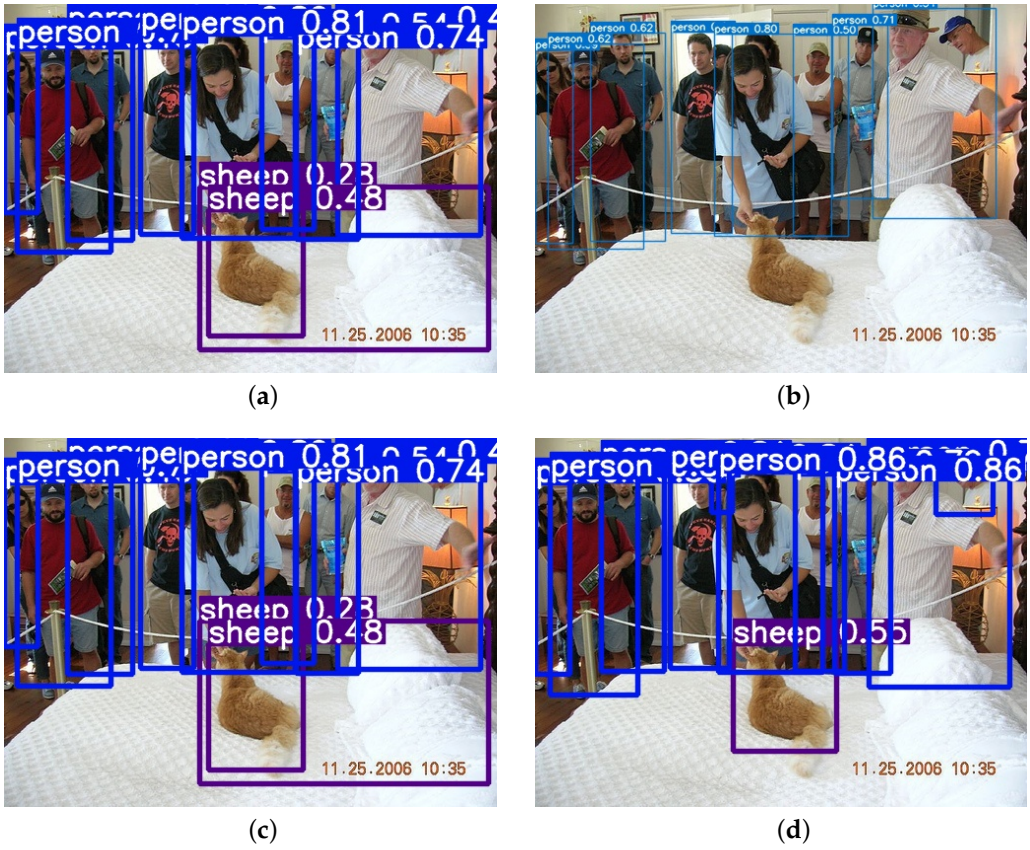


(**a**) (**b**)

(**c**) (**d**)

Figure 8. Experimental comparison of complex scenes in life. From the figure, it can be concluded that the inference times of YOLOv5 and YOLOv7-Tny are less than that of DC-YOLOv8 and that the inference time of YOLOv8 is the same as that of DC-YOLOv8, but their detection results are not as accurate as that of DC-YOLOv8. (**a**) Test results of YOLOv5 with inference time of 10 ms. (**b**) Test results of YOLOv7-Tiny with inference time of 8.2 ms. (**c**) Test results of YOLOv8 with inference time of 12 ms. (**d**) Test results of DC-YOLOv8 with inference time of 12 ms.

For the second set of comparison experiments, images in the PASCALVOC2007 dataset with overlap between multiple people were selected. Figure 9 shows that two people overlap in front of the middle door, and we can only see the head of the person behind due to occlusion from the person in front. In this case, the YOLOv8 detector failed to detect the person behind, with only their head visible. At the position of the cat, there was a false detection (detecting the human arm as a cat) because the color of the human arm is similar to that of a cat. In the case of severe overlap on the rightmost side, YOLOv8 did not detect the person behind. In contrast, in the case of overlap, DC-YOLOv8 accurately detected the person near the door and the person to the far right, and there was no false detection due to similar colors. It can be seen from Figure 9 that DC-YOLOv8 also outperforms YOLOv8 in the detection of normal-sized objects.



(**a**) (**b**)

(**c**) (**d**)

Figure 9. Comparison diagram of the normal-size target experiment. From the figure, it can be concluded that the inference times of YOLOv5 and YOLOv7-Tny are less than that of DC-YOLOv8, and the inference time of YOLOv8 is the same as that of DC-YOLOv8, but their detection results are not as accurate as that of DC-YOLOv8. (**a**) Test results of YOLOv5 with inference time of 9 ms. (**b**) Test results of YOLOv7-Tiny with inference time of 8.4 ms. (**c**) Test results of YOLOv8 with inference time of 12 ms. (**d**) Test results of DC-YOLOv8 with inference time of 12 ms.

5. Conclusions

This paper proposed a small-size object detection algorithm based on a camera sensor; different from traditional camera sensors, we combined a camera sensor and artificial intelligence. Then, some problems in the newly released YOLOv8 and existing small-size object detection algorithms were analyzed and solved. New feature fusion methods and network architectures were proposed. It greatly improved the learning ability of the network. The test and comparison were carried out on the Visdrone dataset, the Tinyperson dataset, and the PASCAL VOC2007 dataset. Through an analysis and experiments, the feasibility of each part of the optimization was proved. DC-YOLOv8 outperformed other detectors in both accuracy and speed. Small targets in various complex scenes were easier to capture.

At present, the application of our proposed algorithm in traffic safety detection is more efficient. As shown in Figure 6, the accuracy of car detection is the highest, so application in traffic safety effect is the best, such as traffic signs, vehicles, pedestrians, etc.

In the future, we will continue to conduct in-depth research on camera sensors and will strive to outperform existing detectors in detection accuracy at various sizes as soon as possible.

Author Contributions: J.G. (Junmei Guo), H.L. (Haiying Liu) and J.G. (Jason Gu) gave technical and writing method guidance as instructors; H.L. (Haitong Lou), X.D., L.B. and H.C. performed the experiments and writing together as classmates. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is funded by Junmei Guo. This work was supported by QLUTGJHZ2018019.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zou, M.Y.; Yu, J.J.; Lv, Y.; Lu, B.; Chi, W.Z.; Sun, L.N. A Novel Day-to-Night Obstacle Detection Method for Excavators based on Image Enhancement and Multi-sensor Fusion. *IEEE Sens. J.* **2023**, *2023*, 1–11.
2. Liu, H.; Member, L.L. Anomaly detection of high-frequency sensing data in transportation infrastructure monitoring system based on fine-tuned model. *IEEE Sens. J.* **2023**, *2023*, 1–9.
3. Zhu, F.; Lv, Y.; Chen, Y.; Wang, X.; Xiong, G.; Wang, F.Y. Parallel Transportation Systems: Toward IoT-Enabled Smart Urban Traffic Control and Management. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 4063–4071.
4. Thevenot, J.; López, M.B.; Hadid, A. A Survey on Computer Vision for Assistive Medical Diagnosis from Faces. *IEEE J. Biomed. Health Inform.* **2018**, *22*, 1497–1511.
5. Abadi, A.D.; Gu, Y.; Goncharenko, I.; Kamijo, S. Detection of Cyclist's Crossing Intention based on Posture Estimation for Autonomous Driving. *IEEE Sens. J.* **2023**, *2023*, 1.
6. Singh, G.; Stefenon, S.F.; Yow, K.C. Yow, Interpretable Visual Transmission Lines Inspections Using Pseudo-Prototypical Part Network. *Mach. Vis. Appl.* **2023**, *34*, 41. https://doi.org/10.1007/s00138-023-01390-6.
7. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
8. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
9. Howard, A.; Wang, W.; Chu, G.; Chen, L.; Chen, B.; Tan, M. Searching for MobileNetV3 Accuracy vs MADDs vs model size. In Proceedings of the IEEE Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 1314–1324.
10. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
11. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet V2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; Volume 11218, pp. 122–138.
12. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778,
13. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
14. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
15. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
16. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Annual Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 28.
17. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Seattle, WA, USA, 13–19 June 2020; Volume 42, pp. 386–397.
18. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
19. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii, USA, 21–26 July 2017; pp. 6517–6525.
20. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
21. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
22. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
23. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Volume 9905, pp. 21–37.
24. Liu, H.; Duan, X.; Chen, H.; Lou, H.; Deng, L. DBF-YOLO:UAV Small Targets Detection Based on Shallow Feature Fusion. *IEEJ Trans. Electr. Electron. Eng.* **2023**, *18*, 605–612. https://doi.org/10.1002/tee.23758.

25. Liu, H.; Sun, F.; Gu, J.; Deng, L. SF-YOLOv5: A Lightweight Small Object Detection Algorithm Based on Improved Feature Fusion Mode. *Sensors* **2022**, *22*, 5817.
26. Sengupta, A.; Cheng, L.; Cao, S. Robust multiobject tracking using mmwave radar-camera sensor fusion. *IEEE Sens. Lett.* **2022**, *6*, 1–4.
27. Bharati, V. LiDAR+ Camera Sensor Data Fusion On Mobiles With AI-based Virtual Sensors to Provide Situational Awareness for the Visually Impaired. In Proceedings of the 2021 IEEE Sensors Applications Symposium (SAS), Sundsvall, Sweden, 23–25 August 2021; pp. 1–6.
28. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. 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, Seattle, WA, USA, 13–19 June 2020; pp. 1571–1580.
29. Lin, T.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii, USA, 21–26 July 2017; pp. 2117–2125.
30. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
31. Bolya, D.; Zhou, C.; Xiao, F.; Lee Jae, Y. Yolact: Real-time Instance Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9157–9166.
32. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
33. Cao, Y.; Chen, K.; Loy, C.C.; Lin, D. Prime Sample Attention in Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11583–11591.
34. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii, USA, 21–26 July 2017; pp. 2261–2269.
35. Lee, Y.; Hwang, J.W.; Lee, S.; Bae, Y.; Park, J. An energy and GPU-computation efficient backbone network for real-time object detection. In Proceedings of the IEEE Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 752–760.