EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Multiclass objects detection algorithm using DarkNet-53 and DenseNet for intelligent vehicles

Lina Yang[1*], Gang Chen[1] and Wenyan Ci[2]

*Correspondence: 122299755@qq.com

[1] College of Information Science and Engineering, Jiaxing University, Jiaxing 314001, China
[2] College of Engineering, Huzhou University, Huzhou 313000, China

Abstract

Intelligent vehicles should not only be able to detect various obstacles, but also identify their categories so as to take an appropriate protection and intervention. However, the scenarios of object detection are usually complex and changeable, so how to balance the relationship between accuracy and speed is a difficult task of object detection. This paper proposes a multi-object detection algorithm using DarkNet-53 and dense convolution network (DenseNet) to further ensure maximum information flow between layers. Three 8-layer dense blocks are used to replace the last three downsampling layers in DarkNet-53 structure, so that the network can make full use of multi-layer convolution features before prediction. The loss function of coordinate prediction error in YOLOv3 is further improved to improve the detection accuracy. Extensive experiments are conducted on the public KITTI and Pascal VOC datasets, and the results demonstrate that the proposed algorithm has better robustness, and the network model is more suitable for the traffic scene in the real driving environment and has better adaptability to the objects with long distance, small size and partial occlusion.

Keywords: Multiclass objects detection, DarkNet-53, DenseNet, Downsampling layers, Loss function

1 Introduction

The main purpose of object detection is to determine whether there are any objects from given categories in a still image or video data, if present, to return the spatial location and extent of each object. It is the basis of many other computer vision detection tasks, such as image description [1], instance segmentation [2, 3], scene understanding [4], and target tracking [5]. It plays an important role in intelligent transportation, video surveillance, automatic driving, etc. The collision warning system is an important part of the advanced driving assistant system (ADAS). It can make environmental awareness and safety warning around the vehicle, and can make corresponding driving decisions. However, the application scenarios of multi-object detection in the real world are usually complex and changeable, so how to balance the relationship between accuracy and computing costs is a difficult task of object detection.

The object detection methods have evolved from traditional stage based on feature extraction plus classifier to end-to-end learning stage based on deep learning. Many features extracted manually made some breakthroughs at that time and were used in today's object detection system. The more famous ones are LBP (local binary pattern), Haar (haar-like features), HOG (histogram of oriented gradient), and SIFT (scale-invariant feature transform). Following feature extraction, the kNN (k-nearest neighbor), decision tree, SVM (support vector machine), Adaboost or Bayesian classifier is designed to obtain object information.

The traditional feature extraction models can only determine low-level feature insformation, such as color information, texture information and contour information. The common points of manual feature extraction are intuitive and easy. However, due to the limitations of detecting multi-object in complex scenes, they have very poor generalization performance. The quality of feature extraction directly affects the performance of the classifier. How to extract more stable features to reflect the essential attributes of the object has always been a research hotspot in the field of object detection.

As the performance of handcrafted features tends to be saturated, CNN (convolution neural network) are reborn worldwide. In the deep learning method, many tasks adopt the end-to-end scheme, that is, input an image and output the final desired results. The details of the algorithm and learning process are all handed over to CNN. Two-stage detection algorithm generates candidate regions and then classifies objects. Representative algorithms include R-CNN (region proposal convolution neural network) series. One-stage detection algorithm integrates candidate region proposal, feature extraction and classification. The representative algorithms include SSD [6], YOLO [7], etc. The CNN-based deep learning method can not only extract detailed texture features from the previous convolution network, but also obtain high-level information from the subsequent convolution layers. Although the features extracted by CNN are not as intuitive and abstract as those manually extracted, they are deep-level features, more stable and more able to reflect the essential attributes of objects. The more convolution layers are used, the more abstract features are obtained, but it also means that the smaller the convolution feature map is, which is not conducive to object detection and location.

Based on the above analysis, we proposed a novel method for multiclass object detection. The method combines DarkNet-53 [7] with DenseNet [8] to further ensure maximum information flow between layers. On the one hand, three DenseNets are used to replace the last three downsampling layers in DarkNet-53 structure to increase the network depth, so that the network can make full use of the information from multi-layer convolution before prediction. On the other hand, the loss function is designed to improve the detection performance. Finally, the experimental comparison and analysis are carried out on Pascal VOC and KITTI datasets.

Compared to the literature, the main contributions of our work are summarized as follows:

1. A new feature extraction network is proposed in this paper. By combining DarkNet-53 with DenseNet, the network can make full use of the information from multi-

layer convolution before prediction, so as to reduce the difficulty of detecting small objects or partially occluded objects, and improve the performance of the object detection system in the driving environment of intelligent vehicle.
2. The loss function of coordinate prediction error in YOLOv3 [9] is further improved to improve the accuracy of the detection system.
3. We conduct extensive experiments on Pascal VOC and KITTI datasets, and the results indicate that the proposed algorithm in this paper is robust to the driving environment and achieves promising detection performance.

The remainder of this paper is structured as follows: Section 2 briefly introduces the related work. Section 3 describes the proposed algorithm in detail. In Sect. 4, we report the dataset, implementation detail and evaluation criteria of the experiments. In Sect. 5, we analyze and discus the experiments results on Pascal VOC and KITTI datasets. The conclusion and future works are given in Sect. 6.

2 Related work

In the past 2 decades, with the rapid development of machine learning theory, object detection algorithms have made great progress in theory and practice. Various algorithms with higher precision, faster speed and stronger robustness emerge in endlessly. Object detection technology based on vision can be roughly divided into three methods: traditional object detection, machine learning-based object detection and deep learning-based object detection. The traditional object detection algorithm is based on template matching techniques and simple part-based models. It is generally divided into two stages: hypothesis generation (HG) and hypothesis verification (HV). HG is responsible for generating all possible region of interests (RoIs) in the image. HV is responsible for verifying the RoIs determined in HG stage, removing no object areas, and identifying and locating existing objects. The object detection method based on machine learning usually uses the multi-scale sliding windows to slide the detected area. Compared with the traditional object detection method, its HV stage is to extract the low-level features such as gray, symmetry, texture and gradient of the object in RoIs or the middle-level features obtained through machine learning, train them into an object classifier through statistical method, and realize the recognition and verification of the object. At that time, most detection methods were built based on handcrafted local invariant features. People can only design more complex feature representations from lack of effective image representation, and use various acceleration technologies to fully utilize computing resources, for example, Viola-Jones detector (VJ) [10] and Histogram of HOG [11]. Later, in order to detect objects with more complex appearance, Girshick et al. [12] proposed deformable part-based model (DPM) , which made full use of the advantages of HOG and SVM and made an important breakthrough in face recognition and other tasks.

Deep learning has revolutionized a wide range of machine learning tasks. State-of-the-art methods for detecting objects of general classes are mainly based on deep CNNs. The object detection based on deep CNNs can be divided into two main categories: (1) Two-stage detector based on region proposal, such as Fast R-CNN [13], Faster R-CNN [14], and Mask R-CNN [3]. This kind of method decomposes the detection problem

into two stages, including region extraction and object classification. As a representative algorithm, Faster R-CNN [14] provides a high degree of precision. However, it still imposes a burden on the speed of detection. Later, many improvement methods including R-FCN [15] were proposed, but they still cannot satisfy the real-time requirements of the system. (2) One-stage detector, such as SSD [6] and YOLO [7]. Different from the two-stage detector, one-stage detector integrates two steps of two-stage detector. Given an input image, the detector directly regresses the boundary box and the category of the object at multiple positions of the image.

YOLOv3 [9] complements all the shortcomings of the YOLO series algorithms and achieves high detection speed and accuracy. However, it is still a very challenging task for the multiclass object detection system applied to intelligent vehicles in complex traffic environment. Because there are many kinds of objects on the road, the process of image acquisition will be affected by many factors, such as illumination, viewpoint, local occlusion, scale transformation, and complex background, which greatly changes the appearance characteristics of the object, thus increases the difficulty of detection. In addition, YOLOv3 needs to abstract features by downsampling mode in the process of feature extraction, which will inevitably ignore the information of many small objects and dense objects, reducing the overall detection accuracy. All these make it a long way to apply multi-object detection algorithm to the actual traffic scenes.

At present, the optimization direction of object detection mainly includes backbone network, positive and negative sample sampling, intersection of union (IoU), loss function, non-maximum suppression (NMS), anchor and learning rate. Reference [16–21] improved YOLO structure, so as to improve the detection precision or speed of the original model on different dataset. Reference [22] improved SSD model and proposed Mask- SSD including a detection branch and a segmentation branch, which could effectively detect small objects in the forms of both traffic signs and pedestrian. According to the different information contained in low-level and high-level features, a multi-scale detection method with feature pyramid networks (FPN) was proposed by Girshick et al. [23]. This method integrated the high-resolution of low-level features and richer semantics of high-level features of the image, which has an obvious detection effect on small objects. Girshick et al. [24] proposed the RetinaNet model again, in which focal loss was used to successfully solve the problem that the object detection loss was easily controlled by a great quantity of negative samples due to the extremely unbalanced area of positive and negative samples in object detection.

In 2017, Huang et al. [8] proposed DenseNet, which got rid of the traditional thinking formula of deepening the network layers and broadening the network structure to improve the network performance. From the view point of features, it connected each layer to the other layers in a feed-forward pattern. The network parameters were greatly reduced; the problems of gradient disappearance and network degradation were alleviated. The network was easier to train and extract the effective features, so as to improve the detection precision of the model. In order to extract the more representative local and detail information, Zhu et al. [25] designed a mixed attention dense network to improve the classification accuracy. In recent years, the researchers have designed many models by combining DenseNet with other CNN structures and employed for various types of object detection applications. For example, Shen

et al. [26] proposed a Dsod object detector without pretraining in combination with SSD and DenseNet. Zhai et al. [27] proposed an improved SSD object detection algorithm based on DenseNet and feature fusion (DF-SSD), which made it have advanced detection effect on small objects and objects with specific relationships. Li et al. [28] attempted to incorporate densely connected networks and spatial pyramid pooling and proposed an improved YOLO lightweight network. Chen et al. [29] proposed a multi-scale feature reuse detection model, which includes DenseNet, feature fusion network, multi-scale anchor region proposal network, and classification and regression network. Pan et al. [30] proposed an adaptively dense feature pyramid network (ADFPNet), to detect objects cross various scales. In addition, Nizarudeen et al. [31] also used a multi-layer DenseNet-ResNet architecture with improved random forest classifier to detect the subtypes of intracranial hemorrhage. Albahli et al. [32] utilized DenseNet-65 for computing the deep features from the given sample on which Faster R-CNN is trained for diabetic retinopathy recognition. Wang et al. [33] proposed a novel framework based on YOLO-Dense to solve the problem of tomato anomaly detection in the complex natural environment. Roy et al. [34] proposed real-time object detection framework based on YOLOv4-DenseNet to detect different growth stages of mango with high degree of occultation in a complex orchard scenario. Xu et al. [35] enhanced feature extraction capability by introducing 2 DenseNet in YOLOv3, achieved higher accuracy of multi-scale remote sensing target detection. Zhao et al. [36] proposed a positioning bolt detection method based on DenseNet-4 and YOLOv3 to improve the detection accuracy and speed of palletizing robot positioning bolts in complex scenes. In addition to improving the network structure, Rezatofighi et al. [37] proposed a new metric for bounding box regression, named Generalized Intersection over Union (GIoU), which was applied to the most popular object detection methods and showed better performance. Lyu et al. [38] proposed a new loss function named tan squared error (TSE), which effectively reduced the influence of the gradient disappearance for sigmoid function, accelerated the convergence of the model and improved the detection accuracy. Wang et al. [39] proposed a new method of generating anchors, named Guided Anchoring, which improve detection performance by using high quality proposals.

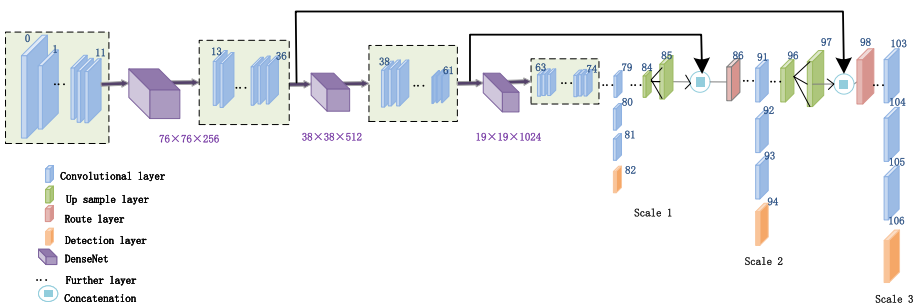

Fig. 1 Overall framework combined DarkNet-53 with DenseNet

3 Proposed method

In order to make full use of local features, improve the accuracy of object detection, we proposed a detection model by combining DarkNet-53 with DenseNet. The overall framework is shown as Fig. 1. The backbone network for performing feature extraction is DarkNet-53 network. It is larger and has 107 layers. The blue structure represents the convolution layer, the green structure represents the upsample layer, the red structure represents the route layer, and the orange structure represents the detection layer. In addition, we add 3 purple DenseNets with a size of 76×76, 38×38 and 19×19 in the No. 12, 37 and 62 layer in DarkNet-53 to modify the feature extractor. It absorbs the advantages of the DenseNet to alleviate gradient disappearance, enhance feature reuse, and reduce the number of parameters.

3.1 DenseNet

It is well known that the deeper the network is, the more likely it is to extract more discriminative features. However, the first problem to be considered is that the gradient disappearance caused by the increase of network layers. Many scholars have proposed solutions including Highway Network [40], FractalNet [41] dealing with this question. Although the structures of these networks are different, the core is to establish a short path between layers. DenseNet [8] continues this idea by directly concatenating all layers on the premise of ensuring the maximum information transmission between layers in the network.

DenseNet is mainly composed of dense blocks and transition blocks. The layout of one dense block is shown in Fig. 2. It is a 4-layer dense block with a growth rate of $k = 4$. $H(\cdot)$ is defined as a composite function of three consecutive operations such as batch normalization (BN), rectified linear unit (ReLU) and 3×3 convolution (Conv). Each layer obtains additional inputs from the previous layers and passes on its own feature maps to the subsequent layers in the feed-forward manner. It is one of the best convolutional neural networks at present.

In DenseNet, the features of l-th layer are denoted as x_l:

$$x_l = H\left([x_0, x_1, \cdots x_{l-1}]\right) \tag{1}$$

where $[x_0, x_1, \cdots x_{l-1}]$ refers to the concatenation of the feature maps produced in layers 0, 1,..., l − 1. For each concatenating, the number of output channels may increase dramatically. In order to control the complexity of the model, the transition block is introduced, which not only halves the length and width of the input, but also changes the number of channels by using 1×1 convolution. DenseNet backbone network establishes the connection relationship of different layers and enhances the feature flow; thus, it has a strong feature learning ability.

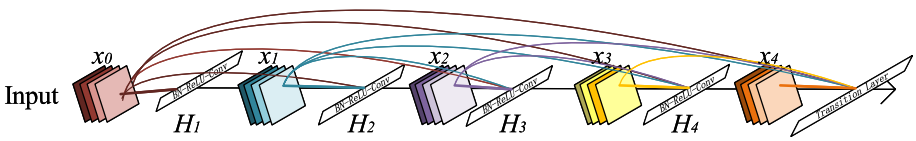

Fig. 2 Dense block

3.2 Feature extraction

In order to make the extracted object information more complete, we introduced the idea of dense connection in DenseNet, improved the DarkNet-53 [9] feature extraction network, and proposed a novel detection model, as shown in Fig. 3. Input images are resized to 608×608, the network architecture consists of 1×1 and 3×3 convolution layer, residual layer, upsampling layer, routing layer, detection layer and DenseNet layer. The new network structure has three 8-layer dense blocks that each has an equal number of layers. The feature-map sizes in the three dense blocks are 76×76, 38×38 and 19×19, respectively, x_l is composed of 16, 32 and 64 sub-feature layers, and the feature layers are concatenated into $76 \times 76 \times 256$, $38 \times 38 \times 512$ and $19 \times 19 \times 1024$ by $[x_0, x_1, x_2, x_3]$ and continue to propagated forward, respectively, so that the network can fully receive the multi-layer convolution features before prediction.

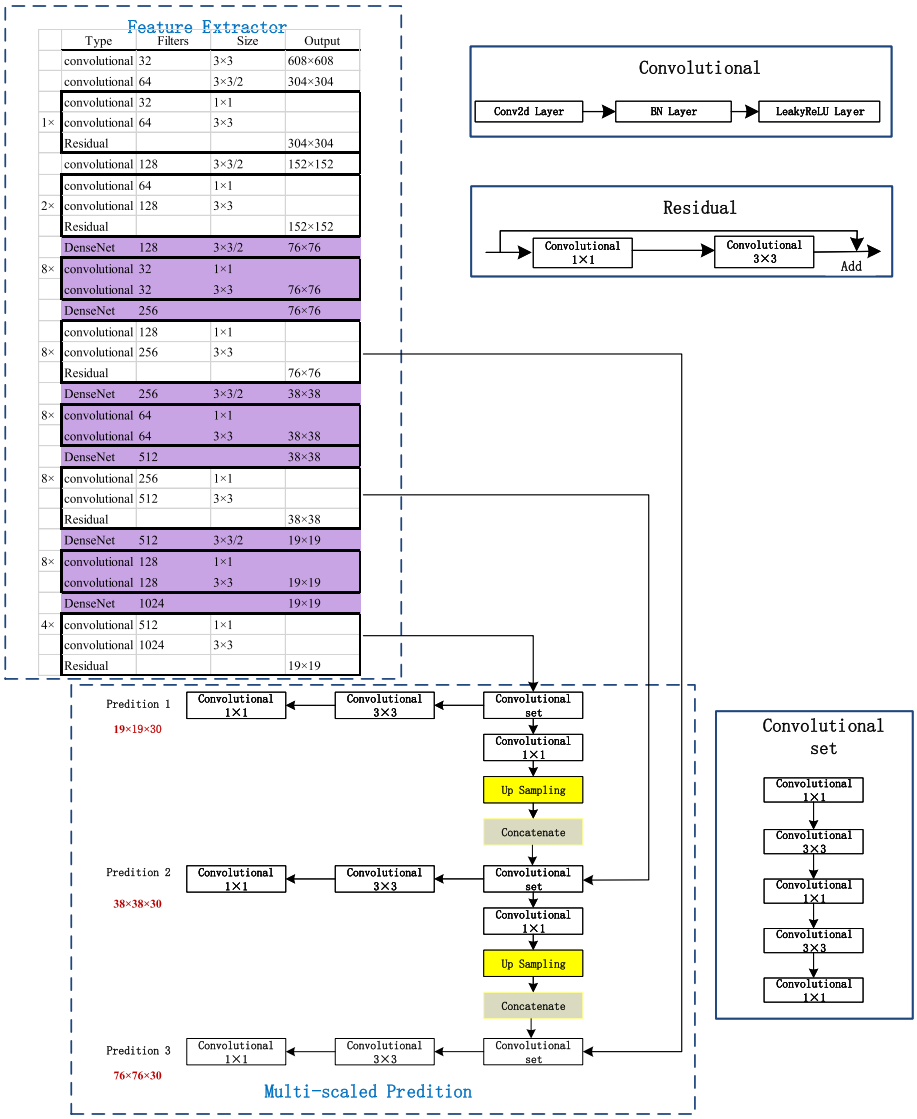

Fig. 3 Network structure combined DarkNet-53 with DenseNet

For object detection in real scenes, the accurate detection of small objects will determine whether detection information is lost. The proposed network predicts at 3 different scales at 3 different positions. In our experiments, we predict 3 bounding boxes at each scale. The size of the convolution kernel used for detection is $1 \times 1 \times \mathrm{B} \times (4+1+5)$, where $B = 3$ represents the number of the prior boxes, '4', '1' and '5' represent 4 bounding boxes offsets, 1 objectness prediction and 5 class predictions respectively. 608×608 images are input into the network and obtain 3 feature maps of different scales. As shown in Fig. 3, the resolutions of each feature map from top to bottom are 76×76, 38×38 and 19×19, corresponding to the detection results of $76 \times 76 \times 30$, $38 \times 38 \times 30$, and $19 \times 19 \times 30$. Among the three feature maps used for detection, the small feature map is responsible for providing deep semantic information, while the large feature map is responsible for providing object location information. The small feature map is fused with the large feature map through upsampling. Because the multi-layer features from DenseNet are received before prediction, the new network can greatly improve the detection and recognition of objects, especially small objects.

3.3 Loss function

Designing an effective loss function is very important for the network training. It can guide the learning of network parameters by back propagating the error between the predicted value and the true value. The total loss function consists of three parts: the coordinate error of bounding box, the IoU error and the class error.

This paper takes the road object detection applied to intelligent vehicles as the research background. In the image to be detected, different road objects have significant differences in size, while small objects have a higher risk of missed detection due to low downsampling or quantity. Therefore, in order to reduce the rate of missed detection or false detection for the occluded objects or small objects and improve the locating precision of objects, this paper mainly modifies the loss from coordinate predictions in YOLOv3.

$$
\begin{aligned}
L_{\text{coord}} = {} & \lambda_{\text{coord}} \sum_{i=0}^{S\times S} \sum_{j=0}^{B} I_{ij}^{\text{coord}} \left[(x_i - \hat{x}_i)^2 + (y + \hat{y}_i)^2 \right] \\
& + \lambda_{\text{coord}} \sum_{i=0}^{S\times S} \sum_{j=0}^{B} I_{ij}^{\text{obj}} \left[\left(\frac{w_i - \hat{w}_i}{w_i} \right)^2 + \left(\frac{h_i - \hat{h}_i}{h_i} \right)^2 \right] \\
& + \lambda_{\text{noobj}} \left(\sum_{i=0}^{S\times S} \sum_{j=0}^{B} I_{ij}^{\text{obj}} \left[\left(C_i - \widehat{C}_i \right)^2 \right] - \sum_{i=0}^{S\times S} \sum_{j=0}^{B} I_{ij}^{\text{noobj}} \left[\left(C_i - \widehat{C}_i \right)^2 \right] \right) \\
& + \sum_{i=0}^{S\times S} I_{ij}^{\text{obj}} \sum_{c \in \text{class}} \left[p_i(c) - \hat{p}_i(c) \right]^2
\end{aligned}
\tag{2}
$$

As shown in the first line of Eq. (2), it consists of two parts. The first is the central coordinate error and the second is the coordinate error of width and height. λ_{coord} is the weight coefficient of coordinate prediction error, λ_{noobj} is the weight coefficient that does not contain the object. S is the number of grids divided, and B is the number of the prior

boxes by each grid. In this paper, we use $\lambda_{\text{coord}} = 5$, $\lambda_{\text{noobj}} = 0.5S = 7$, $B = 9$. $(\mathrm{x}_i, y_i, w_i, h_i)$ and $\left(\hat{\mathrm{x}}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i\right)$ represent the ground truth and predict value of each bounding box. (x, y) represent the center of the box relative to the boundary of the grid cell. (w, h) represent the width and the height relative to the whole image. When the object falls into cell i and the jth bounding box, $I_{ij}^{\text{obj}} = 1$, otherwise, $I_{ij}^{\text{obj}} = 0$.

4 Experiments

4.1 Experimental setup and Dataset

The hardware configuration is Intel (R) core (TM) i7-10510u processor, the main frequency is 4.10 GHz, and the memory is 16 GB. The GPU is NVIDA 2080TI GeForce. Python version is 3.6.10, and TensorFlow version is 1.14.0.

We have conducted experiments on the PASCAL VOC [42] and KITTI [43] datasets. PASCAL VOC has two version: VOC2007 and VOC2012. The dataset includes 11,540 images and 20 categories that are common in everyday life. We focus exclusively on objects on the road, so we choose 5 categories (person, bicycle, bus, car and motorbike) for the training and testing (training set: 5000 images; and test set: 720 images). KITTI dataset covers scenes such as urban, rural and highway in Germany, including pictures and videos. There are up to 15 vehicles and 30 pedestrians in each image, as well as various degrees of occlusion and truncation. Among them, the 3D object detection training set contains 7481 stereo images, including 8 different categories: car, truck, van, pedestrian, sitting people, cyclist, tram and mixture. We exclude those categories with a very low number of objects (sitting people, tram and mixture) (training set: 5385 images; and test set: 749 images).

4.2 Implementation details

We adopt DarkNet-53 as the basic backbone and use the TensorFlow framework to implement our model, which is trained on an NVIDA GeForce 2080TI GPU. We use three eight-layer dense blocks on 608×608 images. During training, we use the initial learning rate of 10^{-4} in the training process, reducing it to 10^{-5} after the first 40 K iterations, and to 10^{-6} after the next 60 K iterations. According to the actual performance of the GPU, we use a batch size of 64, a decay of 0.0005, a momentum of 0.9, a loss threshold of IoU of 0.5, and non-maximum suppression.

4.3 Evaluation criteria

In order to evaluate the effectiveness of the proposed algorithm, Average Precision (AP), mAP and Frames Per Second (FPS) are selected as the evaluation criteria of the performance of algorithms.

$$\mathrm{AP} = \frac{1}{N} \sum_{r(0,0.1,\ldots,1)} p_{r=i} \tag{3}$$

where r represents the recall of an object detection, $p_{r=i}$ represents the precision of $r = i$. In general, the better the classifier, the higher the AP. Whether an object detected is determined according to the intersection over union (IoU).

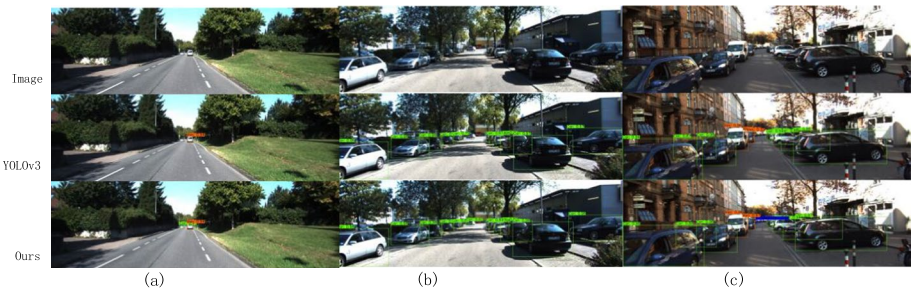

Fig. 4 Visual results on KITTI. The first row to the last row are the original images and the results of the YOLOv3 and the proposed algorithm, respectively

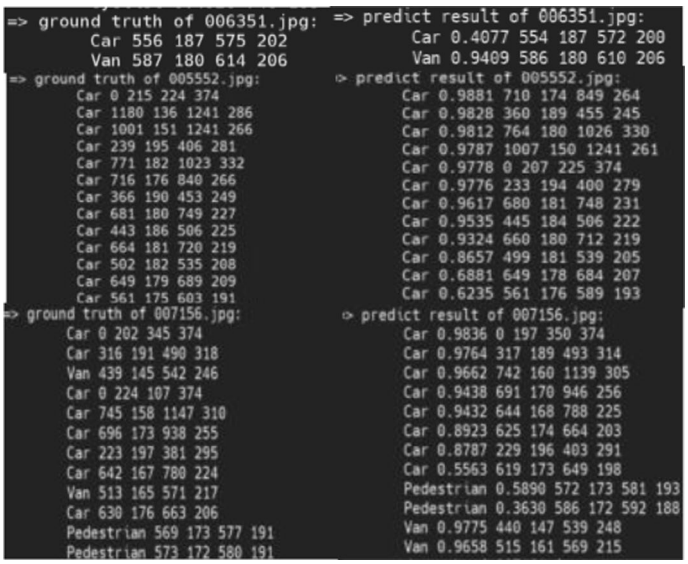

Fig. 5 Ground truth boxes and predict boxes

$$\mathrm{IoU} = \frac{\mathrm{area}\left(\mathrm{RIO}_{\mathrm{det}} \cap \mathrm{RIO}_{\mathrm{gt}}\right)}{\mathrm{area}\left(\mathrm{RIO}_{\mathrm{det}} \cup \mathrm{RIO}_{\mathrm{gt}}\right)} \tag{4}$$

where $\mathrm{RIO}_{\mathrm{det}}$ and $\mathrm{RIO}_{\mathrm{gt}}$ represent the detection box and the ground truth box, respectively. AP is generally evaluated in a category specific manner, i.e., calculated for each object category separately. In order to compare the performance of all objects categories, the mean AP of all objects categories, that is, mAP is always used as the final measure of performance. mAP is the most important index in object detection algorithm.

5 Results and analysis

5.1 Qualitative analysis

DarkNet-53 and dense block are used as feature extractor to extract features from RGB images. The visual results are compared with the results of feature extraction only on DarkNet-53, as shown in Fig. 4. The first row is the RGB images, the second row is YOLOv3 detection results, and the last row is our results. Figure 4a~c are three different scenes on KITTI. The ground truth boxes and the predict boxes of the proposed method

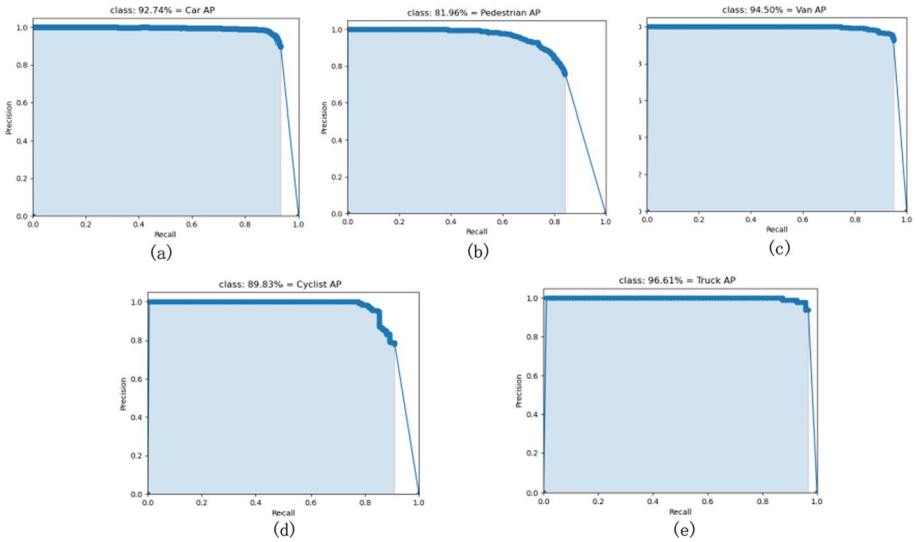

Fig. 6 P-R curve of 5 classes of objects. **a** Car. **b** Pedestrian. **c** Van. **d** Cyclist. **e** Truck

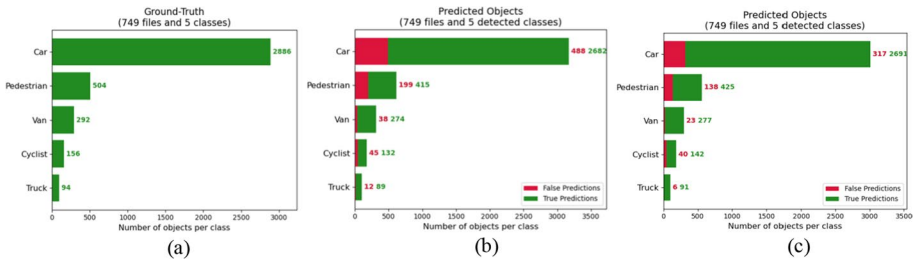

Fig. 7 Results of predicted objects. **a** Ground truth. **b** Predicted objects by YOLOv3. **c** Predicted objects by DarkNet-53 and DenseNet

are showed in Fig. 5. Although there are few objects in the scenario (a), YOLOv3 misses a car due to the long distance and small objects, and all objects are successfully detected by using the proposed algorithm. For the following two scenes, the background is more complex, there are many categories and more than 10 objects with different sizes. The size of distant objects is smaller, and there are also varying degrees of occlusion and truncation. From the comparison in Fig. 4, it can be observed that both algorithms can detect some objects when there are dense targets on the road. However, YOLOv3 algorithm is insensitive to overlapping targets and pedestrian background information, while our algorithm can accurately detect an overlapping and distant car more than YOLOv3 algorithm in scene (b). In scene (c), YOLOv3 not only misjudges one more object, but also misses two pedestrians in the distance. According to Fig. 5, all objects using the proposed algorithm are successfully detected and correctly classified, which shows that our algorithm reduces the difficulty of detecting small objects or partially occluded objects, and improves the performance of the object detection.

The P-R curve of 5 classes of objects generated by the proposed algorithm is shown in Fig. 6. The lateral axis represents the recall and the vertical axis represents the precision. It intuitively shows the comprehensive performance of the recall and precision of the 5 objects of cars, pedestrians, trucks, cyclists and trucks. Figure 7 shows the results

of predicted objects using YOLOv3 and the proposed method. Figure 7a–c shows the ground truth, predicted objects by YOLOv3 and the proposed algorithm, respectively. For example, the actual number of the cars in 749 testing images is 2886, and YOLOv3 predicts 3170 cars, of which 2682 objects are correctly classified and 488 objects are misjudged. The proposed algorithm predicts 3008 cars, of which 2691 are correctly classified and 317 are misjudged. The network structure used in this paper has improved the missed or false detection rate of objects and improved detection accuracy.

From Fig. 7a, it can be seen that the sample size of car is the highest, while the sample size of truck is the lowest. In Fig. 6, the AP of truck is 96.61%, indicating that although its sample number is small, it achieves the best AP in 5 categories due to its large 3D geometric size and strong distinguishability, In contrast, there is no significant difference in shape between car and van, but the sample size of car is too high, and due to various degrees of occlusion and truncation between objects on KITTI dataset, more cars are easily misclassified. However, the sample number and 3D size of pedestrian and cyclist are relatively small, and if there is partial occlusion, their detection difficulty will be greater.

5.2 Quantitative evaluation

5.2.1 Comparison with baseline methods

According to the evaluation criteria described in Sect. 4.3, we compared the performance of our method with baseline YOLOv3. The results on PASCAL VOC dataset show that the mAP of the proposed algorithm outperforms the baseline YOLOv3 by 2.92% with same backbone and same input image size (Table 1). For the KITTI dataset, the results demonstrate that the proposed algorithm increases the mAP from 88.34% to 91.13 (Table 2). From Tables 2 and 3, we can also see that the APs of 5 classes have

Table 1 The mAP (%) comparison of the proposed method with YOLOv3 on PASCAL VOC

Method	Backbone	Input size	AP (%)					mAP (%)
			Person	Bicycle	Bus	Car	Motorbike	
YOLOv3	DarkNet53	608 × 608	90.26	89.94	89.35	84.08	76.01	85.92
Ours	DarkNet53	608 × 608	91.34	91.56	89.92	88.54	82.88	88.84

Table 2 The mAP (%) comparison of the proposed method with YOLOv3 on KITTI

Method	Backbone	Input size	AP (%)					mAP (%)
			Car	Pedestrian	Cyclist	Van	Truck	
YOLOv3	DarkNet53	608 × 608	92.22	78.97	83.24	92.99	94.30	88.34
Ours	DarkNet53	608 × 608	92.74	81.96	89.83	94.50	96.61	91.13

Table 3 Impact of different dense block layers

Method	mAP (%)
Four-layer dense block	89.76
Eight-layer dense block	91.13
Sixteen-layer dense block	90.58

Table 4 Impact of different loss function

Method	mAP (%)
Original loss function	90.05
Improved loss function	91.13

Table 5 Comparison with different methods on KITTI dataset

Method	Backbone	AP (%)					mAP
		Car	Pedestrian	Cyclist	Van	Truck	(%)
Faster R-CNN	VGG-16	81.42	62.36	73.66	81.05	88.78	77.45
Faster R-CNN	ResNet-101	80.68	61.40	71.58	80.91	81.78	75.27
YOLOv3	DarkNet53	92.22	78.97	83.24	92.99	94.30	88.34
Ours	DarkNet53	92.74	81.96	89.83	94.50	96.61	91.13

significantly improved on both dataset. Thanks to the combination of DarkNet-53 and DenseNet, the information from multi-layer convolutions before prediction is fully utilized, thereby reducing the difficulty of detecting small objects. Therefore, the algorithm has a more significant improvement in the detection precision for small objects on the road, including pedestrian (person), cyclist (bicycle) and motorbike.

5.2.2 Impact of the dense block layers and the loss function

The KITTI dataset is currently one of the largest evaluation datasets in autonomous driving scenarios. Therefore, to verify the effectiveness of different dense block layers, we compare the results of using different layers dense block on KITTI. The experimental results are summarized in Table 3. We can see that the mAP of the eight-layer dense block is 91.13%, which is higher than that of the other two dense blocks. So, we choose the eight-layer dense block as our building block.

In addition, in order to verify the effectiveness of the improved loss function, we compare the effects of different loss function in Table 4. By comparing the effects of different loss function on the model, we can see that the mAP of the improved loss function is 91.13%, which is 1.08% higher than that of the loss function of YOLOv3.

5.2.3 Comparison with other methods

To verify the effectiveness of the proposed network, firstly, we compare with representative algorithms of two-stage detector and one-stage detector with the same input data on KITTI. All comparison algorithms are trained on the same hardware platform and software framework using the initialization and training mechanisms described in Sect. 4.2 for the four algorithms. The comparison results are shown in Table 5. In the comparison of several methods, whether VGG-16 or ResNet-101 is used as the backbone network, Faster R-CNN has relatively low detection accuracy. Faster R-CNN only uses the deep features of the last stage for prediction. The features extracted by both of the algorithms have incomplete information. It does not take into account that the actual

Table 6 Comparison of other methods combined with DenseNet on PASCAL VOC

Method	Data	Backbone	AP (%)					mAP (%)
			Person	Bicycle	Bus	Car	Motorbike	
Dsod [26]	07 + + 12	DS/64-192-48-1	84.6	85.3	83.6	80.6	86.8	84.18
DF-SSD [27]	07 + + 12	DenseNet-S-32-1	85.7	85.6	82.9	79.9	86.4	84.10
DS–YOLO [28]	07 + + 12	DarkNet-53	80.9	78.96	76.3	85.58	79.9	80.33
D-MIF [29]	07 +	DenseNet-Evo	85.12	85.65	86.10	89.50	82.86	85.85
ADFPNet [30]	07 + + 12	VGG-16	88.70	88.60	86.60	87.40	89.70	88.20
Ours	07 + + 12	DarkNet53	91.34	91.56	89.92	88.54	82.88	88.84

07 +:07 trainval + 07test in the Pascal VOC. P; 07 + + 12: 07 trainval + 07 test + 12trainval in the Pascal VOC

Table 7 Comparison with several popular methods on KITTI dataset (%)

Method	Data	Car			Pedestrian			Cyclist		
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
Faster R-CNN [14]	Image	88.97	83.16	72.62	79.97	66.24	61.09	72.40	62.86	54.97
YOLOv3 [9]	Image	88.71	74.40	65.58	67.23	49.47	44.99	50.88	36.89	32.64
YOLOv5X6	Image	96.64	93.82	81.54	81.88	64.53	57.44	75.21	52.99	45.67
Stereo R-CNN [44]	Stereo Image	93.98	85.98	71.25	–	–	–	–	–	–
3DOP [45]	Stereo Image	92.96	89.55	79.38	83.17	69.57	63.48	80.52	68.71	61.07
MV3D [46]	Image Lidar	96.47	90.83	78.63	–	–	–	–	–	–
F-PointNet [47]	Image Lidar	95.85	95.17	85.42	89.83	80.13	75.05	86.86	73.16	65.21
Ours	Image	92.05	86.89	80.52	82.91	67.37	58.45	75.36	58.97	46.31

More details can be found on KITTI benchmark homepage (https://www.cvlibs.net)

semantic information of small objects has been lost after the continuous convolution and the pooling. Therefore, it affects the overall detection performance of the algorithm. The proposed method independently predicts at different feature layers, overcoming the defect of Faster R-CNN only using deep features, and significantly improving the detection accuracy of the model. The AP and mAP of our method for 5 classes of objects are significantly better than other network structures.

Secondly, we compared with the combination of DenseNet and other CNN networks mentioned in Sect. 2 on PASCAL VOC dataset. As shown in Table 6, it can be seen that the AP of motorbike is in the middle level of several methods, while the AP of car is only slightly inferior to the method of D-MIF [29] method, and our method achieves the AP of 91.34%, 91.56% and 89.92% on person, bicycle and bus, which demonstrates the effectiveness and advantage of proposed method. Moreover, our method achieves the best mAP among 6 methods.

Lastly, we compare our method with the popular methods on KITTI detection benchmark [43]. The results are obtained from monocular image only, stereo image only and fusion data. Table 7 shows the AP of each methods for car, pedestrian and cyclist in

three scenario regimes: easy, moderate, and hard, which are defined according to the level of occlusion and truncation. We set the IoU threshold 70% for car and 50% for pedestrian and cyclist. Above these thresholds, the object is regarded as detected. Compared with Faster R-CNN and YOLOv3 based on monocular image, our method outperforms them. For YOLOv5X6, our method is slightly inferior to its detection on car, but slightly superior to its detection of pedestrians and cyclists. It is equivalent to 3DOP and Stereo R-CNN based on stereo image. Compared with MV3D and F-PointNet based on data fusion, the detection precision is slightly lower. However, according to Table 8, our detection speed is greatly improved.

6 Conclusions and future works

This paper proposed a multi-object detection algorithm combined DarkNet-53 with DenseNet. Three 8-layer dense blocks to replace the last three downsampling layers in DarkNet-53, making full use of the feature information extracted by the convolutional neural network before prediction. Three feature maps of different scales provide both the deep semantic information and the shallow location feature information, so that they not only strengthen the algorithm's sensitivity to objects of different sizes but also strengthen the transmission of feature information. The loss function of coordinate prediction error in YOLOv3 is further improved to improve the detection accuracy. We conducted extensive experiments on the KITTI and Pascal VOC datasets and made quantitative analysis and qualitative comparisons to demonstrate the effectiveness of our method. The results demonstrated that the AP and mAP of our method are significantly improved than other network structures, indicating that the proposed algorithm has better robustness, and the network model is more suitable for the traffic scene in the real driving environment and has better adaptability to the objects with long distance, small size and partial occlusion. In addition, this method also achieves a good balance between detection accuracy and inference speed. In the experimental environment, the detection time of each image is about 0.029 s, and our network run at 24 frames per second on NVIDA 2080TI GeForce GPU, which can better meet the real-time requirements of intelligent vehicle in complex traffic environment.

The future improvements can be conducted in the following aspects: (1) The algorithm will continue to be optimized by modifying the network structure to reduce the size of convolutional neural network and the amount of parameters, so as to shorten the

Table 8 Time inference comparison

Method	Environment	Runtime (1/FPS) (s)
Faster R-CNN [14]	GPU @ 3.5 Ghz (Python + C/C + +)	0.23
YOLOv3 [9]	GPU @ 2.5 Ghz (Python)	0.027
YOLOv5X6	GPU @ 3.5 Ghz (Python)	0.05
Stereo R-CNN [44]	GPU @ 2.5 Ghz (Python)	0.3
3DOP [45]	GPU @ 2.5 Ghz (Matlab + C/C + +)	3
MV3D [46]	GPU @ 2.5 Ghz (Python + C/C + +)	0.24
F-PointNet [47]	GPU @ 3.0 Ghz (Python)	0.17
Ours	GPU @ 2.5 Ghz (Python)	0.029

running time of the network. (2) Further design new loss function and use data augmentation technique to improve the positioning precision. (3) Further design the multiclass obstacles detection algorithm based on LiDAR and camera information fusion for intelligent vehicle.

Abbreviations

DenseNet	Dense convolution network
ADAS	Advanced driving assistant system
LBP	Local binary pattern
Haar	Haar-like features
HOG	Histogram of oriented gradient
SIFT	Scale-invariant feature transform
kNN	K-nearest neighbor
SVM	Support vector machine
CNN	Convolution Neural Network
R-CNN	Region Proposal Convolution Neural Network
HG	Hypothesis generation
HV	Hypothesis verification
RoIs	Region of interests
VJ	Viola–Jones detector
DPM	Deformable part-based model
IoU	Intersection of union
NMS	Non-maximum suppression
FPN	Feature pyramid networks
TSE	Tan squared error
BN	Batch normalization
ReLU	Rectified linear unit
AP	Average precision
FPS	Frames per second

Acknowledgements
Not applicable.

Author contributions
LY, CC and WC conceived and designed the study. LY initiated the project, proposed the method, implemented the algorithms, and wrote the paper. CC conducted experiments, analyzed the data and wrote the paper. LY, CC and WC reviewed and edited the manuscript. All authors read and approved the final manuscript.

Funding
This work was sponsored by Public Welfare Technology Application Research Projects of Zhejiang Province of China [Grant No. LGG22F030016] and Scientific Research Project of Jiaxing University [Grant No. CD70521022].

Availability of data and material
The datasets generated or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 16 January 2023 Accepted: 23 July 2023
Published online: 01 August 2023

References

1. Q. Wu, C. Shen, P. Wang et al., Image captioning and visual question answering based on attributes and external knowledge. IEEE Trans. Pattern Anal. Mach. Intell. **40**(6), 1367–1381 (2018)
2. J. Dai, K. He, J. Sun, Instance-aware semantic segmentation via multi-task network cascades, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (IEEE, New York, 2016), pp.3150–3158
3. K. He, G. Gkioxari, P. Dollar, R. Girshick. *Proceeding of the IEEE International Conference on Computer Vision (ICCV)*. Mask r-cnn. (IEEE, Venice, 2017), pp.2980–2988
4. Z. Guo, Y. Huang, X. Hu et al., A survey on deep learning based approaches for scene understanding in autonomous driving. Electronics **10**(4), 471–471 (2021)
5. K. Kang, H. Li, J. Yan et al., T-cnn: tubelets with convolutional neural networks for object detection from videos. IEEE Trans. Circuits Syst. Video Technol. **28**(10), 2896–2907 (2018)

6. W. Liu, D Aaguelov, D. Erhan, et al., SSD: single shot multibox detector, in *European Conference on Computer Vision*. (Springer, Cham, 2016). pp. 21–37
7. J. Redmon, S. Divvala, R. Girshick, et al, You only look once: unified, real-time object detection, in *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (IEEE, Long Las Vegas, 2016) ,pp.779–788
8. G. Huang, Z. Liu, V. Laurens, et al., Densely connected convolutional networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. (IEEE, Honolulu, 2017), pp. 2261–2269
9. J. Redmon, A. Farhadi. *YOLOv3: An Incremental Improvement* (2018). arXiv:1804.02767v1
10. P. Viola, M.J. Jones, Robust real-time face detection. Int. J. Comput. Vision **57**(2), 137–154 (2004)
11. N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in *Proceeding of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR)*. (IEEE, San Diego, 2005), pp. 886–893
12. P. Felzenszwalb, D. McAllester, D. Ramanan. A discriminatively trained, multiscale, deformable part model, in *Proceeding of the IEEE Computer Vision and Pattern Recognition (CVPR)*. (IEEE, Anchorage, 2008), pp. 1–8
13. R. Girshick. Fast R-CNN, in *Proceeding of the IEEE International Conference on Computer Vision (ICCV)*. (IEEE, Santiagor, 2015), pp.1440–1448
14. S. Ren, K. He, R. Girshick et al., Faster R-CNN: towards real-time object detection with region proposal networks. Adv. Neural Inf. Process. Syst. **28**, 91–99 (2015)
15. J. Dai, Y. Li, K. He et al., R-fcn: object detection via region-based fully convolutional networks. Adv. Neural Inf. Process. Syst. **29**, 379–387 (2016)
16. S.P. Rajendran, L. Shine, R. Pradeep, et al., Real-Time Traffic Sign Recognition using YOLOv3 based Detector, in *Proceeding of International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. (IEEE, Kanpur, 2019)
17. C. Zou, R. Xue, Improved YOLOv3 object detection algorithm: combining GIoU and focal loss. Comput. Eng. Appl. **56**(24), 214–222 (2020)
18. J. Du Jinhang, M. He, Real-time road vehicles detection based on improved YOLOv3. Comput. Eng. Appl. **56**(11), 26–32 (2020)
19. S. Song, Y. Piaon, Z. Jiang, Vehicle classification and tracking for complex scenes based on improved YOLOv3. J. Shandong Univ. **50**(2), 27–33 (2020)
20. M. Ju, H. Luo, Z. Wang et al., Improved YOLOv3 algorithm and its application in small target detction. Acta Optica Sinica **39**(7), 0715004 (2019)
21. W. Zhang, C. Sun, M. Wang et al., An improved Yolov5 real-time detection method for small objects captured by UAV. Soft. Comput. **26**(1), 361–373 (2022)
22. C. Sun, Y. Ai, S. Wang et al., Mask-guided SSD for small-object detection. Appl. Intell. **6**(51), 3311–3322 (2021)
23. T.Y. Lin, D. Pollar, R. Girshick, et al., Feature Pyramid networks for object detection, in *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (IEEE, Hawaii, 2017), pp. 2117–2125
24. T.Y. Lin, P. Goyal, R. Girshick et al., Focal loss for dense object detection. IEEE Trans. Pattern Anal. Mach. Intell. **99**, 2999–3007 (2018)
25. M. Zhu, C. Chen, N. Wang et al., Mixed attention dense network for sketch classification. Appl. Intell. **51**(10), 7298–7305 (2021)
26. Z. Shen, L. Zhuang, J. Li, et al., DSOD: Learning Deeply Supervised Object Detectors from Scratch, in *Proceeding of the IEEE International Conference on Computer Vision (ICCV)*. (IEEE, Venice, 2017), pp. 1919–1927
27. S. Zhai, D. Shang, S. Wang et al., DF-SSD: an improved SSD object detection algorithm based on DenseNet and feature fusion. IEEE Access **8**, 24344–24357 (2020)
28. C. Li, J. Yao, Z. Lin et al., Object detection method based on improved YOLO light weight network. Laser Optoelectr Progress **57**(14), 141003 (2020)
29. B.Y. Chen, Y.K. Shen, K. Sun, Research on object detection algorithm based on multilayer information fusion. Math. Probl. Eng. **2020**, 1–13 (2020)
30. H. Pan, G. Chen, J. Jiang, Adaptively dense feature pyramid network for object detection. IEEE Access **2019**(7), 81132–81144 (2019)
31. S. Nizarudeen, G.R. Shunmugavel, Multi-layer ResNet-DenseNet architecture in consort with the XgBoost classifier for intracranial hemorrhage (ICH) subtype detection and classification. J. Intell. Fuzzy Syst. **44**(2), 2351–2366 (2023)
32. S. Albahli, T. Nazir, A. Irtaza et al., Recognition and detection of diabetic retinopathy using densenet-65 based faster-RCNN. Comput. Mater. Contin. **67**(5), 1333–1351 (2021)
33. X. Wang, J. Liu, Tomato anomalies detection in greenhouse scenarios based on YOLO-dense. Front. Plant Sci. **12**, 634103 (2021)
34. A.M. Roy, J. Bhaduri, Real-time growth stage detection model for high degree of occultation using densenet-fused yolov4. Comput. Electron. Agric. **193**, 106694 (2022)
35. D. Xu, Y. Wu, Improved YOLO-V3 with DenseNet for multi-scale remote sensing target detection. Sensors **20**(15), 42760 (2020)
36. K. Zhao, Y. Wang, Y. Zuo et al., Palletizing robot positioning bolt detection based on improved YOLO-V3. J. Intell. Rob. Syst. **104**, 41 (2022)
37. H. Rezatofighi, N. Tsoi, J.Y. Gwak, et al., Generalized intersection over Union: a metric and a loss for bounding box regression, in *Proceeding of the IEEE conference on computer vision and pattern recognition (CVPR)*. (IEEE, Long Beach, 2019), pp. 658–666
38. L. Shuo, X. Cai, R. Feng, YOLOv3 network based on improved loss function. Comput. Syst. Appl. **28**(2), 1–7 (2019)
39. J. Wang, K. Chen, S. Yang, et al., Region proposal by guided anchoring, in *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (IEEE, Long Beach, 2019), pp. 2965–2974
40. R.K. Srivastava, K. Greef, J. Schmidhuber, Training very deep networks, in *Advances in Neural Information Processing Systems*. (MIT Press, Montreal, 2015), pp. 2377–2385
41. G. Larsson, M. Maire, G. Shakhnarovich, *Fractalnet: ultra-deep neural networks without residuals* (2016). arXiv:1605.07648

42. M. Everingham, S.A. Eslami, L. VanGool et al., The pascal visual object classes challenge: a retrospective. Int. J. Comput. Vision **111**(1), 98–136 (2015)
43. A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? The KITTI vision benchmark suite, in *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* (IEEE, Providence, 2012), pp. 3354–3361
44. P. Li, X. Chen, S. Shen. *Proceedings of the IEEE conference on computer vision & pattern recognition (CVPR), Stereo R-CNN based 3D object detection for autonomous driving*. (IEEE, Long Beach, 2019), arXiv:1902.09738
45. X. Chen, K. Kundu, Y. Zhu et al., 3D object proposals using stereo imagery for accurate object class detection. IEEE Trans. Pattern Anal. Mach. Intell. **40**(5), 1259–1272 (2018)
46. X. Chen, H. Ma, J. Wan, et al. Multi-view 3D object detection network for autonomous driving, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Honolulu, 2017), pp. 691
47. C. Qi, W. Liu, C. Wu, et al., Frustum PointNets for 3D Object Detection from RGB-D data, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (IEEE, Salt Lake City, 2018), pp. 918–927

Publisher's Note