Expert Systems With Applications 213 (2023) 119108

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Real-time vehicle detection algorithm based on a lightweight You-Only-Look-Once (YOLOv5n-L) approach

Minglin Bie [a], Yanyan Liu [a,*], Guoning Li [b], Jintao Hong [c,*], Jin Li [d,*]

[a] Changchun University Science and Technology, Department of Electronics and Information Engineering, Changchun University Science and Technology, Jilin 130022, China
[b] Chinese Academy of Sciences, Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China
[c] University of Cambridge, Photonics and Sensors Group, Department of Engineering, University of Cambridge, CB3 0FA, UK
[d] Beihang University, School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing 100191, China

ARTICLE INFO

Keywords:
Vehicle detection
YOLO
Lightweight
Real-time

ABSTRACT

A vehicle detection algorithm is of great significance for automatic driving technology. Current vehicle detection algorithms suffer from the complex structure, high configuration of hardware requirements, and the difficulty to apply to mobile terminal equipment. In order to solve these issues, this paper proposes an improved YOLOv5 algorithm, named YOLOv5n-L, for lightweight. First, a depthwise separable convolution and a C3Ghost module are used to replace several C3 modules to reduce the model parameters and improve the detection speed. Then a Squeeze-and-Excitation attention mechanism is integrated into backbone network to improve the accuracy of the algorithm and suppress the environmental interference. Finally, a bidirectional feature pyramid network is used for multi-scale feature fusion to enrich feature information and improve the feature extraction ability of the proposed algorithm. The experimental results demonstrate that compared with the original algorithm, the model weight is reduced by 40 % to only 2.3 M. The mean average precision (mAP@0.5) is increased by 1.7 %. The detection speed reaches 80 FPS, which could accurately detect vehicle targets in real-time.

1. Introduction

Vehicles have become an indispensable part of our life. While vehicles bring convenience to our life, they also cause traffic congestion, traffic accidents and other problems. Autonomous driving technology can solve the above problems well, so it has been paid more and more attention by researchers. The most important part of autonomous driving technology (Sonata et al., 2021; Tao et al., 2021; Woźniak et al., 2022) is the vehicle detection algorithm. Vehicle detection algorithm combined with millimeter wave radar technology or visual ranging algorithm can accurately identify and range vehicle targets and prevent the occurrence of traffic accidents. Therefore, vehicle detection algorithm has great market potential.

Machine learning is an important foundation of automatic driving technology and is vital to the field of computer vision (Zinchenko et al., 2020), which has been widely used in different tasks (Kondratenko et al., 2022; Sova et al., 2020). Object detection algorithm based on deep learning is an important branch of machine learning. In recent years, the development of object detection algorithm based on deep learning is particularly rapid, and it has been applied to various fields of our life (Mathew and Mahesh, 2022; Ke et al., 2022). At present, there are mainly-two kinds of object detection algorithms based on deep learning (LeCun et al., 2015). One is a two-stage detection algorithm based on region proposals, such as Faster R-CNN (Ren et al., 2015; Sun et al., 2018). First, the bounding box is generated, and then the bounding box needs to be classified and regressed. Another is one-stage detection algorithms, such as YOLO (Redmon et al., 2016) and SSD, which can directly predict the classes and positions of different objects by treating detection task as a regression problem (Benjdira et al., 2019; Maity et al., 2021). The two-stage detection algorithm has high accuracy, but its detection speed is slow, which cannot meet the requirements of real-time detection. The detection accuracy of the one-stage detection algorithm is not as high as that of the two-stage detection algorithm, but its detection speed is fast. This paper improves the one-stage detection algorithm YOLO, so that it can be applied to mobile terminals and achieve appropriate accuracy. At present, different versions of the improved YOLO algorithm have been widely used in vehicle detection. Some authors (Miao et al. 2020; Taheri Tajar et al., 2021) directly use the

* Corresponding authors.
E-mail addresses: liuyy306@163.com (Y. Liu), jh2101@cam.ac.uk (J. Hong), jl11269@buaa.edu.cn (J. Li).

https://doi.org/10.1016/j.eswa.2022.119108
Received 6 September 2022; Received in revised form 9 October 2022; Accepted 18 October 2022
Available online 25 October 2022

original YOLO algorithm for vehicle target detection. Although the vehicle target can be detected accurately, when it is applied to the mobile terminal (Huang et al., 2018; Rani, 2021), it will impose a huge burden on the mobile terminal hardware, and the detection frame rate during operation cannot achieve real-time detection. However, YOLO tiny's weight file is relatively small, and it is suitable for mobile devices with low computing power, but its detection accuracy is low, which is not enough to detect vehicles well, so it is not suitable for vehicle target detection. In addition, some algorithms (Huang et al., 2021; Li et al., 2020) use lightweight networks such as MobileNet or EfficientNet to replace the whole backbone of YOLO. This method can greatly reduce the weight of the algorithm, and the real-time frame rate also meets the requirements, but compared with the original algorithm, the accuracy is very low. In terms of improving algorithm accuracy and real-time target detection, (Kondratenko et al., 2020) analyzed the advantages and disadvantages of neural network architectures (ResNet, U-Net, SegNet, YOLOv3). Different artificial neural network algorithms and architectures are compared to obtain the highest recognition accuracy. (Yang et al., 2022) proposed a novel deep convolutional network structure TS-YOLO with three spatial pyramid pooling (SPP) modules in YOLOv4, demonstrating the excellent performance of their model in multi-scale object detection. (Ahmadi et al., 2021) proposed an improved algorithm based on yolov3. This algorithm can successfully count several objects in a single image with reduced calculation time and a very light process. (Wieczorek et al., 2021) propose a model of face detection in risk situations, the designed model works with maximum simplicity to support mobile devices.

In order to meet the requirements of real-time detection and detection accuracy in complex environments, we proposed a YOLOv5n-L algorithm. The proposed algorithm uses efficient and simplified network structure to replace inefficient and complex network structure. Our algorithm can highlight the feature information of vehicles by improving the feature extraction network and add attention mechanism to obtain the details of vehicles that need to be concerned. The main goal of the proposed algorithm is to ensure the real-time detection frame rate of the algorithm, simplify the algorithm structure while improving the detection accuracy of the algorithm, achieve accurate and real-time detection of vehicle targets.

2. The principle of YOLO algorithm

2.1. Introduction to YOLO algorithm

Joseph Redmon et al. proposed a YOLO (You-Only-Look-Once) algorithm, which uses single network to directly predict bounding boxes and classes from images. The detection speed of the YOLOv1 is relatively fast, but the YOLOv1 is not effective for objects that are close to each other and small targets. The YOLOv2 algorithm adopts Darknet19 as the feature extraction network, which can adapt to different sizes of images and improve the low detection accuracy for small targets. The YOLOv3 (Redmon & Farhadi, 2018; Liu et al., 2021) constructs a new feature extraction network Darknet53, which introduces the idea of residual networks to enable the algorithm to extract deeper features. The YOLOv3 further improves the detection accuracy of small targets, while maintaining the advantage of the detection speed. The YOLOv4 (Bochkovskiy et al., 2004) takes CSPDarknet53 as the backbone network, it reduces the amount of network computing, the memory consumption and achieves the surpassing of the YOLOv3 algorithm in speed and accuracy (Sozzi et al., 2022; Nepal and Eslamiat, 2022).

There are five versions of the YOLOv5, namely YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. YOLOv5n network has the smallest depth and width in the YOLOv5 series, while others are deepened and widened on their basis. With the deepening and widening of the structure, the detection accuracy of the algorithm is constantly improved, but the training consumes more time and requires higher hardware configuration during the operation. When an object detection algorithm is applied to mobile terminal equipment, the detection time and the hardware performance need to be considered.

Fig. 1 shows the network structure of the YOLOv5 algorithm. The YOLOv5 network consists of four parts: Input, Backbone, Neck, and Output.

At the input, the YOLOv5 uses mosaic data enhancement, adaptive image scaling, and adaptive anchor calculation. The mosaic data enhancement method is to randomly scale, splice and stack four random images, enriching the background of the detection objects. The adaptive image scaling step is to adaptively add the smallest edge to images, which can improve the detection speed. The adaptive anchor calculation step is to output the bounding boxes according to the preset anchor boxes, then compare them with the actual anchor boxes, constantly iterate the parameters, and adaptively calculate the best anchor boxes value.

The backbone network consists of a Conv module, a CBS module, a C3 module and a SPPF module. In YOLOv5 version 6.0, a new Conv module is used to replace the Focus for slicing operations. The CBS module consists of a standard convolution layer, a batch normalization layer, and an activation function Silu. The C3 module consists of several ResUnit modules and three standard convolutional layers. The C3 module can strengthen the feature fusion ability of the convolutional neural network and improve the inference speed. The SPPF module is proposed on the basis of the SPP module. It is a spatial pyramid pooling layer (Huang et al., 2020), which can expand the receptive field, achieve local and global feature fusion, and enrich feature information.

At the neck, the YOLOv5 combines a Feature Pyramid Network (FPN) and a Path Aggregation Network (PAN) (Liu et al., 2018) to enhance the capacity of the feature fusion. The Feature Pyramid Network (FPN) conveys features from the top of the network to the bottom, while the Path Aggregation Network (PAN) conveys features from the bottom of the network to the top.

At the output, the YOLOv5 uses a Generalized Intersection over Union (GIOU) loss function (Rezatofighi et al., 2019) and a Non-Maximum Suppression (NMS). The loss function of the YOLOv5 consists of a bounding box loss (L_{box}), a classification loss (L_{cls}) and a confidence loss (L_{obj}). The bounding box Loss (L_{box}) adopts a GIoU Loss function. The classification loss (L_{cls}) and the confidence loss (L_{obj}) adopt a Binary Cross Entropy (BCE) loss function. The YOLOv5 uses a multi-scale detection way to predict classes and positions of objects with different sizes on three scales.

2.2. Issues of YOLO algorithm

When applying the YOLO algorithm to a vehicle detection task, it is necessary to take into account the limited hardware performance of mobile devices. The YOLOv5x algorithm cannot be simply chosen. It has the best detection effect, but a vehicle detection task requires high real-time performance. Consequently, this paper selects the smallest weight model of the YOLOv5n algorithm, but in the actual detection process, the YOLOv5n algorithm has several problems. Firstly, the YOLOv5n algorithm uses a large number of the standard convolutions and C3 modules, which improves the accuracy of the algorithm, but reduces the running speed and increases the parameters of the model. Secondly, the scene will be changed rapidly in the city, and enough detection accuracy is required. However, the YOLOv5n algorithm is not ideal for detection processing under complex conditions, resulting in issues with the wrong detection and missing detection. Finally, the feature extraction ability of the algorithm is insufficient, and the detection effect of the algorithm is not good in darkness, occlusion, and other conditions.

3. Algorithm improvement

3.1. Depthwise separate convolution

A standard convolution calculation is that all feature channels

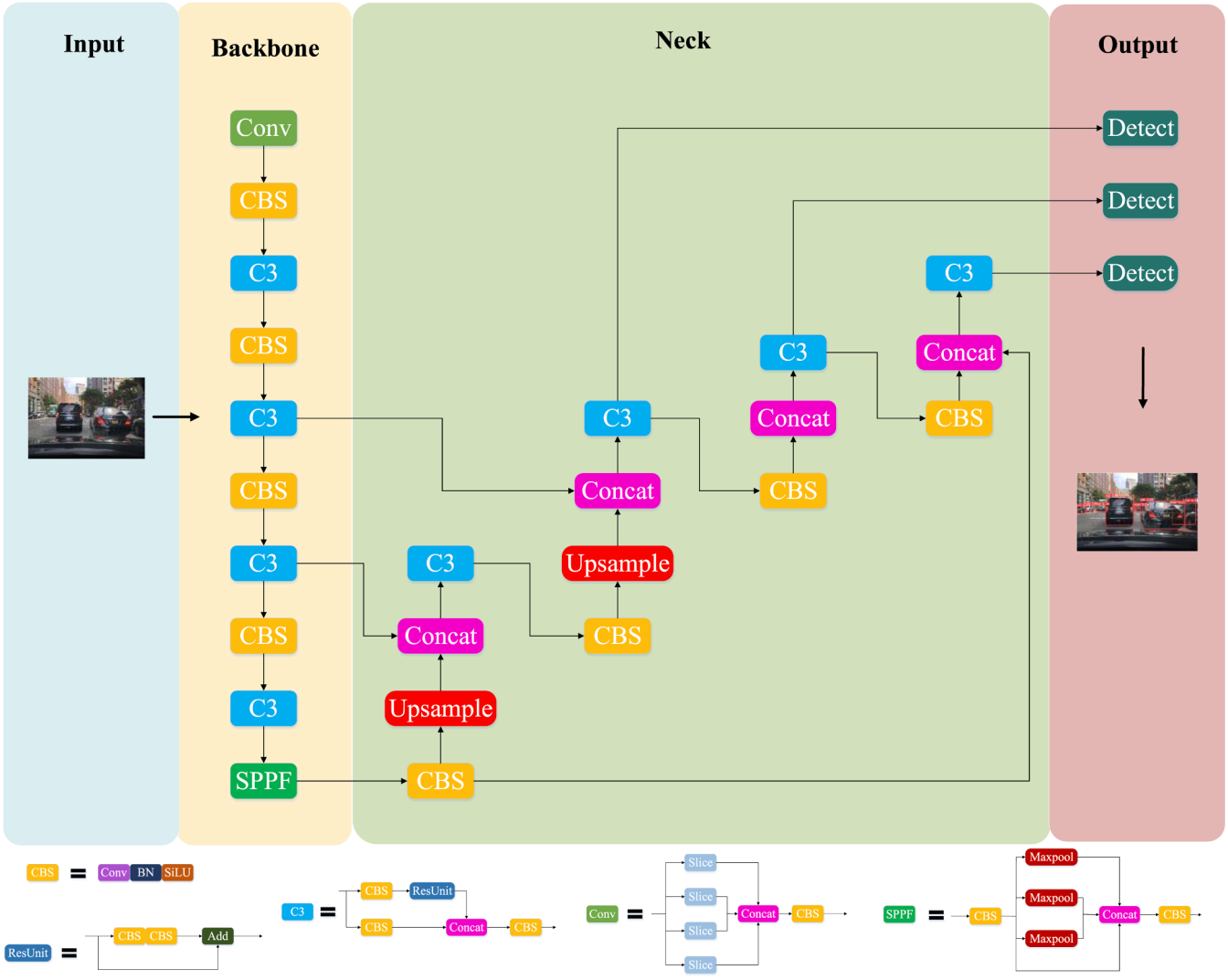

Fig. 1. YOLOv5 network structure.

convolve the corresponding convolution kernel, and then add all the results and output features. So, the convolution process requires plenty of parameters and calculations. The essential idea of depthwise separable convolution (Howard et al., 2017) is to decompose a complete convolution operation into two parts: a depthwise convolution and a pointwise convolution. The calculation principal of the depthwise convolution is that one feature channel convolves with one corresponding convolution kernel, and then fuses the output of the depthwise

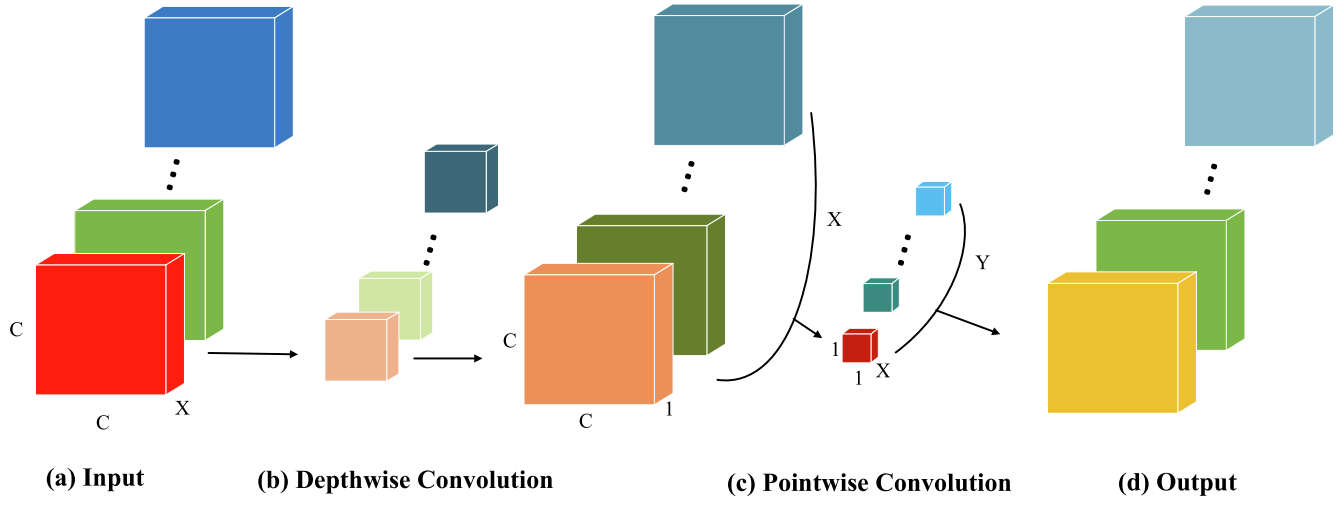

Fig. 2. Depthwise separate convolution structure.

convolution with the pointwise convolution. This decomposition method can greatly reduce the calculated amount while maintaining an accuracy comparable to the standard convolution.

As can be seen from Fig. 2, the calculated amount of a standard convolution is $C{\cdot}C{\cdot}X{\cdot}Y{\cdot}D{\cdot}D$, the calculated amount of the depthwise separable convolution is $C{\cdot}C{\cdot}X{\cdot}D{\cdot}D + X{\cdot}Y{\cdot}D{\cdot}D$. Therefore, it is concluded that the calculation amount of the depthwise separable convolution is compared with the calculation amount of the standard convolution:

$$\frac{C{\cdot}C{\cdot}X{\cdot}D{\cdot}D + X{\cdot}Y{\cdot}D{\cdot}D}{C{\cdot}C{\cdot}X{\cdot}Y{\cdot}D{\cdot}D} = \frac{1}{Y} + \frac{1}{C^2} \tag{1}$$

where X represent the number of input channels, Y represent the number of output channels, the kernel size is $C{\cdot}C$ and the feature map is $D{\cdot}D$.

In the case of the same input and output, using a 3×3 depthwise separate convolution, the calculated amount is much less computation than the standard convolution. Therefore, the depthwise separable convolution can effectively reduce the calculation cost of the algorithm. Replacing the C3 module with the depthwise separable convolution can greatly reduce the model parameters and improve the detection speed of the algorithm.

3.2. Build C3Ghost module

The feature maps of many feature channels in the standard convolution are very similar. That is to say, the features extracted from the standard convolution are repeated to a certain extent. So, there is no need to carry out complete convolution operations to obtain the required feature maps. The Ghost module (See Fig. 3) uses the standard convolution to obtain part of the feature maps and then generates more feature maps through a linear operation (Han et al., 2020). Finally, two sets of feature maps are spliced in the specified dimension to obtain more feature maps with fewer parameters and calculations.

The operation of producing n feature maps in any convolution layer can be formulated as:

$$Y = X{\cdot}f + b \tag{2}$$

where $X \in R^{c\times h\times w}$ represents the input of convolution, c represents the number of input channels, h and w respectively represent the height and width of the input feature map, $Y \in R^{h'\times w'\times n}$ is the output feature map with n channels, h' and w' are the height and width of the output feature map, $f \in R^{c\times k\times k\times n}$ is the convolution filter of this layer, and its kernel size is $g \times g$, b is the bias term. In this convolution process, the calculation is:

$$n{\cdot}h'{\cdot}w'{\cdot}c{\cdot}g{\cdot}g \tag{3}$$

Let the size of the linear operation kernel be $r \times r$, and each basic feature corresponds to s redundant features, $s \ll c$. Suppose that m characteristic graphs are obtained by the original method ($n = m{\cdot}s$), and there is an identity in the transformation process of Ghost Module, so the actual effective transformation quantity is:

$$m{\cdot}(s-1) = \frac{n}{s}{\cdot}(s-1) \tag{4}$$

The calculation amount using the Ghost module is:

$$\frac{n}{S}{\cdot}h'{\cdot}w'{\cdot}c{\cdot}g{\cdot}g + (s-1){\cdot}\frac{n}{s}{\cdot}h'{\cdot}w'{\cdot}r{\cdot}r \tag{5}$$

The calculation amount of the standard convolution improved by Ghost module is:

$$\frac{n{\cdot}h'{\cdot}w'{\cdot}c{\cdot}g{\cdot}g}{\frac{n}{s}{\cdot}h'{\cdot}w'{\cdot}c{\cdot}g{\cdot}g + (s-1){\cdot}\frac{n}{s}{\cdot}h'{\cdot}w'{\cdot}r{\cdot}r} = \frac{c{\cdot}g{\cdot}g}{\frac{1}{s}{\cdot}c{\cdot}g{\cdot}g + \frac{s-1}{s}{\cdot}r{\cdot}r} \approx \frac{s{\cdot}c}{s+c-1} \approx s \tag{6}$$

Therefore, the calculation amount of the Ghost module is about $1/s$ of the standard convolution. Combining the GhostBottleneck with the C3 module, a C3Ghost module is formed, which is shown in Fig. 4. Using the C3Ghost module to replace the C3 module in the network can reduce the model parameters and ensure the detection accuracy of the network while the realized network is lightweight.

3.3. Squeeze-and-Excitation (SE) attention mechanism

In a vehicle detection task, due to the complex scenes of urban roads, sometimes the number of vehicles will increase sharply, so it is necessary to improve the detection accuracy of the algorithm. The interference of environmental factors can be solved by adding an attention mechanism to improve the algorithm's detection accuracy.

Attention mechanism can make up for the problem of strong local and insufficient global of CNN (Convolutional neural network), so as to obtain the global context information and improve the accuracy of the algorithm. In order to obtain more important features in the channel dimension, we introduce the Squeeze-and-Excitation attention mechanism. The Squeeze-and-Excitation attention mechanism represents the importance of each feature channel by learning a set of weight values, rearranging the feature channels according to the size of the weight value, paying more attention to the feature channels with more information, and suppressing unimportant features channels (Hu et al., 2018). The Squeeze-and-Excitation (SE) Block is divided into two parts: Squeeze and Excitation, which is shown in Fig. 5. The squeeze operation compresses the corresponding feature maps in one dimension through global pooling, converting the feature map of W × H × C into 1 × 1 × C. After getting global features, the relationship between each channel is extracted through an excitation operation and the weights of each channel are generated. The excitation operation adopts the gating mechanism in the sigmoid function. By introducing full connection (FC1) layer, the number of channels is changed by using parameter W1.

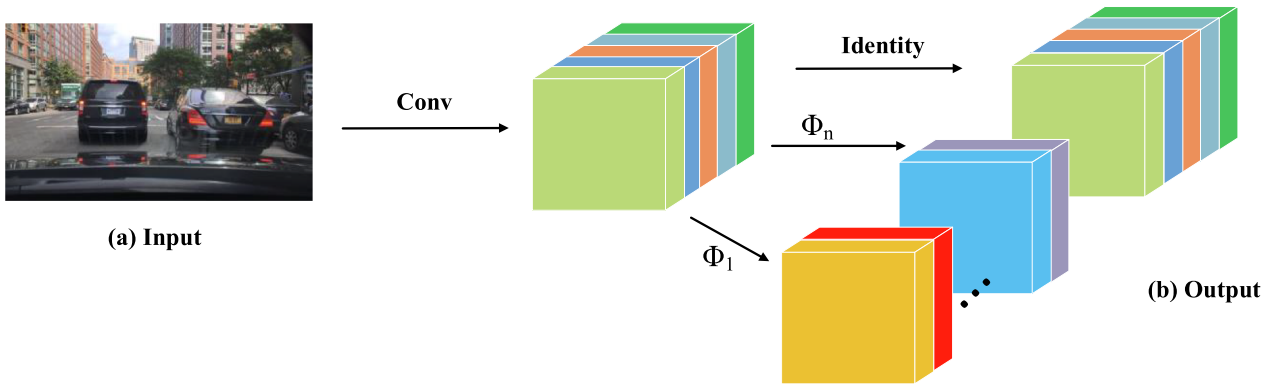

Fig. 3. Ghost module structure.

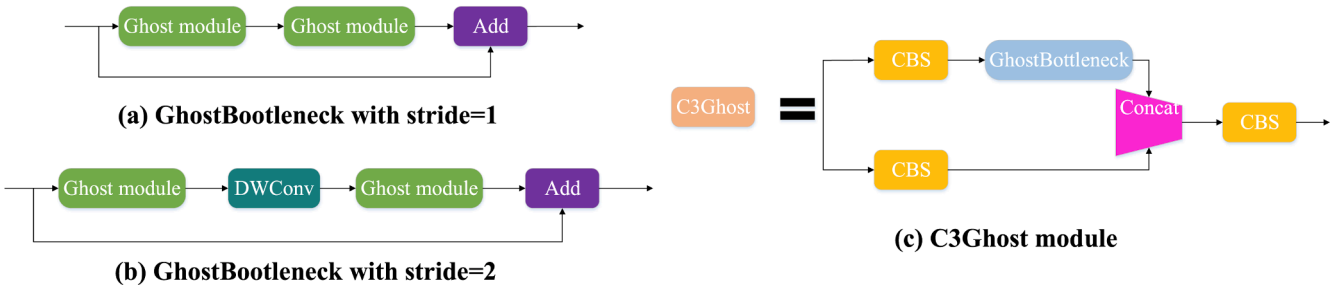

Fig. 4. C3Ghost structure.

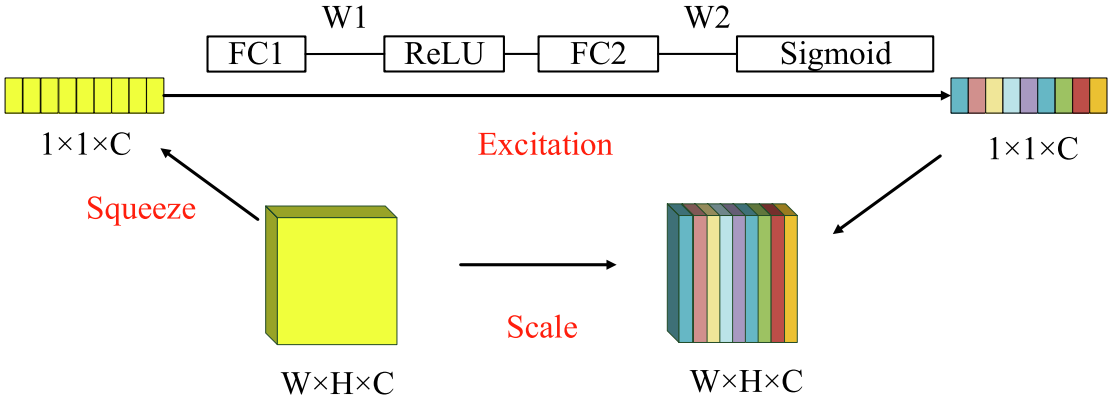

Fig. 5. Squeeze-and-Excitation Block structure.

After being activated by the ReLU function, the channel is restored to the original number of channels with the parameter W2 through the full connection (FC2) layer, and finally the weight of each channel is generated by the sigmoid function. Finally, the generated weight is applied to the corresponding feature channel through a scale operation to obtain the final output.

A mobile inverted bottleneck convolution shown in Fig. 6 includes a pointwise convolution, a depthwise convolution and a Squeeze-and-Excitation Block. In the first place, the input is convoluted by the pointwise convolution, and changing the output channel dimension according to the expansion ratio, then using the depthwise convolution. After that, the Squeeze-and-Excitation Block for squeeze and excitation operation is introduced, and then the original channel dimension with the pointwise convolution is restored. Finally, it is added with the input to form a residual jump connection structure and get the output feature maps. The mobile inverted bottleneck convolution uses the pointwise convolution and the depthwise convolution to extract features, which reduce the number of parameters. Meanwhile, the introduction of the Squeeze-and-Excitation attention mechanism can improve the algorithm's detection accuracy.

3.4. Bidirectional feature pyramid network (BiFPN)

The YOLOv5 adopts the combination of a Feature Pyramid Network (FPN) and a Path Aggregation Network (PAN) for the feature extraction. However, this method is transformed from all the feature maps into the same resolution, unable to take full advantage of the features of different resolutions because the contribution of input features to output features is unequal at different resolutions.

The main idea of a bidirectional feature pyramid network is efficient bidirectional cross-scale connections and weighted feature fusion (Tan et al., 2020). The bidirectional feature pyramid network has three major improvements: deleting nodes with only one input edge; adding an additional edge between the original input nodes and output nodes at the same level; each bidirectional path is regarded as one feature network layer and repeated several times.

In Fig. 7, the orange nodes are located in the middle of the first and last layer, which have only one input edge with no feature fusion, and deleting them has little impact on fusing different feature information, but simplifies the bidirectional network. Then, a jump connection between the yellow node and the purple node in the second layer is added to fuse more features without increasing too much cost. Finally, connections between the yellow node in the first layer, the orange node in the second layer and the purple node in the last layer are added, which can obtain higher-level feature fusion (Fig. 8).

The bidirectional feature (Fig. 9) pyramid network structure can aggregate features of different resolutions. Since deeper networks can extract more complex features, they can be used to optimize feature fusion of different scales, which can enrich feature information and improve the feature extraction ability of the algorithm.

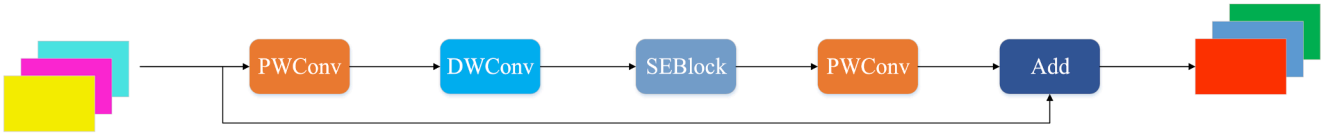

Fig. 6. Mobile inverted bottleneck convolution.

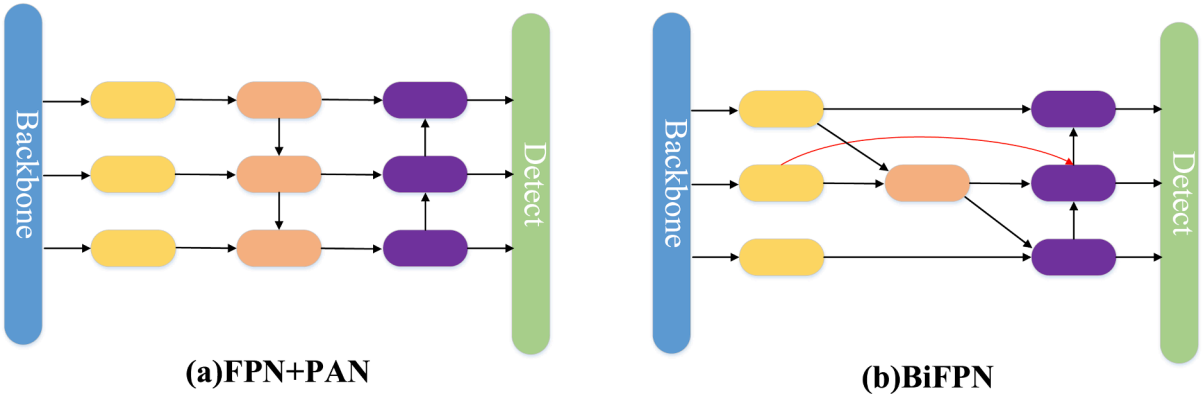

Fig. 7. Bidirectional feature pyramid network structure.

Fig. 8. Partial images in BDD100k dataset.

4. Results

4.1. Data set

At present, the largest and most diverse driving data set is the BDD100K data set, which contains images from different times like day and night; different scenes such as expressways, urban roads, and parking lots; different weather such as sunny, snowy and rainy. The BDD100K data set is widely used in automatic driving research. There are 70,000 photos in the BDD100K data set, in which 3000 photos are randomly selected as data set (Yu et al., 2018). We selected 2400 photos for training, 300 photos for testing and 300 photos for verification. The ratio between training set, testing set, verification set are 8:1:1.

4.2. Experimental equipment and evaluating indicator

This experiment uses the Ubuntu 18.04 operating system; CPU is Intel Core i7-8700, 16G memory; GPU is Nvidia GeForce GTX 1070 Ti, 8G display memories; Deep learning framework is Pytorch 1.9.0, CUDA 10.2, and CUDNN 7.5.0.

During testing the accuracy of the algorithm, this paper uses the mean average precision (mAP@0.5) and FPS (frames per second) as the main evaluation indicators.

The formula for calculating the mean accuracy of n categories is as follows:

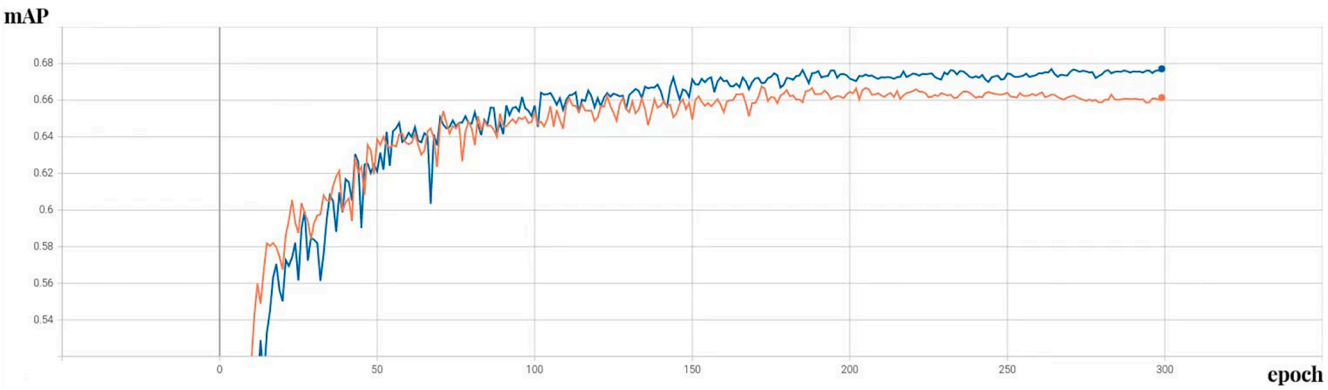

Fig. 9. Comparison of the detection accuracy between the YOLOv5n and YOLOv5n-L, where the YOLOv5n is the orange curve and the YOLOv5n-L is the blue curve.

$$mAP = \frac{1}{n}\sum_{i=1}^{n}\int_{0}^{1} P(R)dR \tag{7}$$

In the above formula, P and R represent accuracy rate and recall rate respectively:

$$P = \frac{TP}{TP + FP}$$
$$R = \frac{TP}{TP + FN} \tag{8}$$

where the TP represents the number of correct targets in the detection results, the FP represents the number of wrong targets in the detection results, the FN represents the number of missing targets in the correct targets.

The FPS refers to the number of detected frames per second, and its size is not only related to the weight of the algorithm, but also to the hardware configuration of the experimental equipment.

4.3. Experimental comparison

In order to verify the accuracy of the improved algorithm in this paper, experiments need to be carried out by building several models named YOLOv5n-MB, YOLOv5n-DW, YOLOv5n-Gh, YOLOv5n-Bi, and YOLOv5n-L. Among them, the YOLOv5n-MB is to replace part of standard convolutions with the mobile inverted bottleneck convolutions (MBConv) in the backbone and head network of YOLOv5n. The YOLOv5n-DW is achieved by changing some C3 modules into the depthwise separable convolutions (DWConv) in the backbone network of YOLOv5n. The YOLOv5n-Ghost replaces part of the C3 modules with the C3Ghost modules in the head of YOLOv5n. The YOLOv5n-Bi is to improve the combination of the Feature Pyramid Network (FPN) and the Path Aggregation Network (PAN) structure of YOLOv5n to the bidirectional feature pyramid network (BiFPN) structure. The YOLOv5n-L is the algorithm proposed in this paper.

As shown in Table 1 that the improved algorithm uses more efficient network structures to improve the network structures of YOLOv5n, and the accuracy has been promoted, and the weight value of the model is reduced. It is also proved that the depthwise separable convolution and the C3Ghost module do not reduce the accuracy of the algorithm, but reduce the parameters of the model. The bidirectional feature pyramid network structure does not increase the weight value of the algorithm. The mobile inverted bottleneck convolution not only effectively improves the detection accuracy but also reduces the model parameters. The combination of all the above improvements with the YOLOv5n algorithm can minimize the model parameters and greatly improve the accuracy of the algorithm. Compare YOLOv5n algorithm with YOLOv5n-L algorithm in TensorBoard, the result is in Fig. 10. It is concluded that, compared with YOLOv5n, the mean average precision of the YOLOv5n-L is significantly improved.

In order to verify the detection accuracy of our algorithm. We compared our algorithm with SSD and Faster-RCNN under the same data set. SSD use VGG16 as the backbone network, Faster-RCNN use ResNet50 as the backbone network. It is known from Table 2that the weight of the YOLOv5n-L model is 1.2 % of SSD, and its accuracy is 4 % higher than SDD. The accuracy of Faster-RCNN is 1.2 % higher than YOLOv5n-L, but its weight is more than 140 times larger than YOLOv5n-L. Therefore, by comparing with the one-stage detection algorithm SSD and two-stage detection algorithm Faster-RCNN, it can be concluded that our algorithm can accurately detect vehicle targets.

We select the same data set and compare our algorithm with the homogeneous algorithms, YOLOv3-tiny, YOLOv4-tiny, and YOLOv5n. The YOLOv3-tiny and YOLOv4-tiny are based on a Darknet framework. The YOLOv5n and our algorithm are based on a Pytorch framework. The experimental results are shown in Table 3. It is known from the table that the size of the YOLOv5n-L model is 6.6 % of YOLOv3-tiny, 9.8 % of YOLOv4-tiny, and 60 % of YOLOv5n. The mean average precision of the YOLOv5n-L is better than YOLOv3-tiny and YOLOv4-tiny is increased by 29.7 % and 22.5 %, respectively. It also gains 1.7 % higher than YOLOv5n. The detection speed of the YOLOv5n-L reaches 80 FPS, which can meet the requirements of real-time detection.

4.4. Comparison of detection results

We randomly select several images and use the YOLOv3-tiny, YOLOv4-tiny, YOLOv5n and YOLOv5n-L algorithms to perform the vehicle detection experiments. The results are presented in Fig. 10.

As shown in Fig. 10, the YOLOv3-tiny and YOLOv4-tiny have poor detection results, with problems of missing detection and low detection

Table 1
Comparison of detection effects of six models.

Algorithm	MBConv	DWConv	Ghost	BiFPN	mAP@0.5	Weight(M)	FLOPs(G)
YOLOv5n	×	×	×	×	0.661	3.8	4.5
YOLOv5n-MB	√	×	×	×	0.670	3.2	4.4
YOLOv5n-DW	×	√	×	×	0.664	3.4	4.3
YOLOv5n-Gh	×	×	√	×	0.663	3.5	4.3
YOLOv5n-Bi	×	×	×	√	0.669	3.8	4.5
YOLOv5n-L	√	√	√	√	0.678	2.3	4.0

(a) (b) (c) (d) (e)

Fig. 10. Comparison of detection results: (a) the original images (b) the YOLOv3-tiny detection results (c) the YOLOv4-tiny detection results (d) the YOLOv5 detection results (e) the YOLOv5n-L detection results.

Table 2
Comparison of the homogeneous algorithms.

Algorithms	mAP@0.5	Weight(M)
SSD	0.638	181
Faster-RCNN	0.690	331.1
YOLOv5n-L	0.678	2.3

Table 3
Comparison of the homogeneous algorithms.

Algorithms	mAP@0.5	Weight(M)	FPS
YOLOv3-tiny	0.381	34.7	25
YOLOv4-tiny	0.453	23.5	33.3
YOLOv5n	0.661	3.8	76.9
YOLOv5n-L	0.678	2.3	80

accuracy, and due to the lack of detection ability and feature extraction ability in complex backgrounds, the YOLOv5n algorithm suffers from the wrong detection and missing detection. From the first image, there is only one car on the left, but the YOLOv5n algorithm detects two cars, which is the wrong detection. There are three cars on the left side of the second image and part of a van is blocked by a black car in the fourth image, which is not detected by the YOLOv5n algorithm. These are missing detections. However, the algorithm in this paper can improve the detection accuracy of the algorithm by incorporating the Squeeze-and-Excitation attention mechanism. The bidirectional feature pyramid network is used for multi-scale feature fusion to improve the feature extraction ability of the algorithm. Therefore, the proposed algorithm can detect vehicles more accurately, and the detection effect is better than the original algorithm under the conditions of distance, occlusion, or darkness.

5. Conclusion

Aiming at the complex structure of the current vehicle detection algorithm, the high configuration required by the hardware, and the difficulty of applying to mobile devices, an improved YOLOv5n-L algorithm is proposed.

The experimental results demonstrate that the improved algorithm increases the mean average precision by 1.7 % on the BDD100K data set compared with the YOLOv5n algorithm. Compared with SSD , the mAP@0.5 is increased by 4 %. Compared with Faster-RCNN, the mAP@0.5 is decreased 1.2 %. Compared with homogeneous algorithms YOLOv3-tiny and YOLOv4-tiny, the mAP@0.5 is increased by 29.7 % and 22.5 %, respectively. Under different scenes, weather, and other conditions, the accuracy of the improved algorithm has been promoted. The main contribution of our algorithm is that, by improving the YOLOv5 algorithm with efficient and simplified network structures, our method enhances the detection ability and reduces the model parameters. The weight value of our model is only 2.3 M, and the detection speed reaches 80 FPS, which is conducive to real-time vehicle detection at mobile terminal equipment. However, although our algorithm can achieve real-time detection, the accuracy has room for improvement. The potential limitation of our algorithm is that it needs to be applied to mobile terminal equipment with limited computing power. If mobile devices with higher computing power are used, higher precision algorithms are needed to match. In the future work, we will continue to study how to improve the accuracy of the algorithm in vehicle detection.

Declaration of Competing Interest

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

Data availability

Data will be made available on request.

References

Ahmadi, M., Xu, Z., Wang, X., Wang, L., Shao, M., & Yu, Y. (2021, October). Fast Multi Object Detection and Counting by YOLO V3. In 2021 China Automation Congress (CAC) (pp. 7401-7404). IEEE.

Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A., & Ouni, K. (2019, February). Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and

yolov3. In 2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS) (pp. 1-6). IEEE.
Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020.
Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., & Xu, C. (2020). Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 1580-1589).
Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7132-7141).
Huang, R., Pedoeem, J., & Chen, C. (2018, December). YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 2503-2510). IEEE.
Huang, S., He, Y., & Chen, X. A. (2021, April). M-YOLO: A Nighttime Vehicle Detection Method Combining Mobilenet v2 and YOLO v3. In Journal of Physics: Conference Series (Vol. 1883, No. 1, p. 012094). IOP Publishing.
Huang, Z., Wang, J., Fu, X., Yu, T., Guo, Y., & Wang, R. (2020). DC-SPP-YOLO: Dense connection and spatial pyramid pooling based YOLO for object detection. *Information Sciences, 522*, 241–258.
Ke, Q., Siłka, J., Wieczorek, M., Bai, Z., & Woźniak, M. (2022). Deep Neural Network Heuristic Hierarchization for Cooperative Intelligent Transportation Fleet Management. *IEEE Transactions on Intelligent Transportation Systems*.
Kondratenko, Y., Atamanyuk, I., Sidenko, I., Kondratenko, G., & Sichevskyi, S. (2022). Machine Learning Techniques for Increasing Efficiency of the Robot's Sensor and Control Information Processing. *Sensors, 22*(3), 1062.
Kondratenko, Y., Sidenko, I., Kondratenko, G., Petrovych, V., Taranov, M., & Sova, I. (2020, October). Artificial neural networks for recognition of brain tumors on MRI images. In International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications (pp. 119-140). Springer, Cham.
LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444.
Li, X., Qin, Y., Wang, F., Guo, F., & Yeow, J. T. (2020, July). Pitaya detection in orchards using the MobileNet-YOLO model. In 2020 39th Chinese Control Conference (CCC) (pp. 6274-6278). IEEE.
Liu, R. W., Yuan, W., Chen, X., & Lu, Y. (2021). An enhanced CNN-enabled learning method for promoting ship detection in maritime surveillance system. *Ocean Engineering, 235*, Article 109435.
Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8759-8768).
Maity, M., Banerjee, S., & Chaudhuri, S. S. (2021, April). Faster r-cnn and yolo based vehicle detection: A survey. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 1442-1447). IEEE.
Mathew, M. P., & Mahesh, T. Y. (2022). Leaf-based disease detection in bell pepper plant using YOLO v5. *Signal, Image and Video Processing, 16*(3), 841–847.
Miao, Y., Liu, F., Hou, T., Liu, L., & Liu, Y. (2020, November). A nighttime vehicle detection method based on YOLO v3. In 2020 Chinese Automation Congress (CAC) (pp. 6617-6621). IEEE.
Nepal, U., & Eslamiat, H. (2022). Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs. *Sensors, 22*(2), 464.
Rani, E. (2021). LittleYOLO-SPP: A delicate real-time vehicle detection algorithm. *Optik, 225*, Article 165818.
Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.
Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.
Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 658-666).
Sonata, I., Heryadi, Y., Lukas, L., & Wibowo, A. (2021, April). Autonomous car using CNN deep learning algorithm. In Journal of Physics: Conference Series (Vol. 1869, No. 1, p. 012071). IOP Publishing.
Sova, I., Sidenko, I., & Kondratenko, Y. (2020, October). Machine learning technology for neoplasm segmentation on brain MRI scans. In Proceedings of the 2020 PhD Symposium at ICT in Education, Research, and Industrial Applications (ICTERI-PhD 2020), Kharkiv, Ukraine (pp. 6-10).
Sozzi, M., Cantalamessa, S., Cogato, A., Kayad, A., & Marinello, F. (2022). Automatic bunch detection in white grape varieties using YOLOv3, YOLOv4, and YOLOv5 deep learning algorithms. *Agronomy, 12*(2), 319.
Sun, X., Wu, P., & Hoi, S. C. (2018). Face detection using deep learning: An improved faster RCNN approach. *Neurocomputing, 299*, 42–50.
Taheri Tajar, A., Ramazani, A., & Mansoorizadeh, M. (2021). A lightweight Tiny-YOLOv3 vehicle detection approach. *Journal of Real-Time Image Processing, 18*(6), 2389–2401.
Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10781-10790).
Tao, C., He, H., Xu, F., & Cao, J. (2021). Stereo priori RCNN based car detection on point level for autonomous driving. *Knowledge-Based Systems, 229*, Article 107346.
Wieczorek, M., Siłka, J., Woźniak, M., Garg, S., & Hassan, M. M. (2021). Lightweight Convolutional Neural Network Model for Human Face Detection in Risk Situations. *IEEE Transactions on Industrial Informatics, 18*(7), 4820–4829.
Woźniak, M., Zielonka, A., & Sikora, A. (2022). Driving support by type-2 fuzzy logic control model. *Expert Systems with Applications, 207*, Article 117798.
Yang, W., Ding, B. O., & Tong, L. S. (2022, March). TS-YOLO: An efficient YOLO Network for Multi-scale Object Detection. In 2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC) (Vol. 6, pp. 656-660). IEEE.
Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., & Darrell, T. (2018). Bdd100k: A diverse driving video database with scalable annotation tooling. arXiv preprint arXiv:1805.04687, 2(5), 6.
Zinchenko, V., Kondratenko, G., Sidenko, I., & Kondratenko, Y. (2020, August). Computer vision in control and optimization of road traffic. In 2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP) (pp. 249-254). IEEE.