Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 5155970, 13 pages
https://doi.org/10.1155/2022/5155970

Research Article

Little-YOLOv4: A Lightweight Pedestrian Detection Network Based on YOLOv4 and GhostNet

Hongtao Zheng, Hong Liu, Wei Qi, and Hao Xie

School of Information and Electrical Engineering, Zhejiang University City College, Hangzhou 310015, China

Correspondence should be addressed to Hong Liu; liuhong@zucc.edu.cn

Received 22 July 2022; Accepted 22 August 2022; Published 5 September 2022

Academic Editor: Chia-Huei Wu

Nowadays, the areas such as intelligent-assisted driving, intelligent monitoring, pedestrian analysis, and attitude detection are developing rapidly. Inseparable from these fields is pedestrian detection technology. In order to deploy pedestrian detection algorithms on devices with limited hardware resources, a recognition algorithm that takes into account both performance and speed is required. Thus, a Little-YOLOv4 network structure is proposed, where the GhostNet is used to extract image features and the PANet is ameliorated by adding BiFPN path fusion which can integrate richer semantic features and preserve spatial information. The DO-DConv and DSC take place of standard convolution and the ReLU6 replaces the Leaky ReLU, which reduce the computational cost. The squeeze-and-excitation network is added to YOLOv4 head network, which could greatly reduce the interference information. The pedestrian detection results show that the mAP is 90.11% and the FPS is 79 by using Little-YOLOv4. The Little-YOLOv4 network structure could achieve a good compromise between algorithmic accuracy and speed.

1. Introduction

Nowadays, great progress has been made in the field of artificial intelligence, and many related applications have appeared. Pedestrian detection is one of the most important technologies. It has many application scenarios, such as video surveillance in literature [1] and automatic driving. Due to the continuous development of hardware devices and algorithms, the accuracy and speed of pedestrian detection using deep learning have been significantly improved.

Existing pedestrian detection methods normally adopt a deep neural network structure to extract deep features with strong expressive ability. The deep neural network requires a lot of storage space and computing resources. Most mobile devices could not meet this demand, which limits the development and application of this pedestrian detection algorithm. Nowadays, the development of lightweight algorithms makes it possible to solve these problems, such as R-CNN series [2–5], SSD series [6, 7], and YOLO series [8–11]. Literature [12] proposed a novel deep small-scale sense network (termed SSN) for small-scale pedestrian detection and design a novel loss function based on cross-entropy loss to increase the loss contribution from hard-to-detect small-scale pedestrians. Literature [13] proposed a real-time pedestrian detection algorithm based on tiny-yolov3. The proposed method uses K-means clustering on our training set to find the best priors. The lightweight model mainly focuses on reducing the computational cost and parameters, reducing the actual running time, and simplifying the underlying implementation method. But at the same time, simple lightweight will inevitably reduce the recognition accuracy. While reducing weight, we also need to consider adding some structures that can deeply extract effective information, such as attention mechanisms.

We need to introduce some structures to extract effective information at a deep level. Through the lightweight structure and the introduction of some special structures, our algorithm can improve the running frame rate. At the same time, it is ensured that the recognition accuracy rate of a specified category is maintained at a high level. This idea

can currently be used to solve the lightweight problem of pedestrian detection.

(1) We use GhostNet [14] to replace the traditional CSPDarknet53 backbone. In this way, the number of parameters and computation can be effectively reduced and the network can be more lightweight while maintaining certain detection accuracy

(2) We ameliorate PANet by adding BiFPN [15] path fusion. The BiFPN path fusion could integrate richer semantic features and preserve spatial information

(3) We introduce depthwise overparameterized depthwise (DO-DConv) to replace the 3×3 and 5×5 standard convolutions of YOLOV4neck, which could speed up the reasoning without increasing the complexity of calculation

(4) We replace the 1×1 standard convolution in the CBL1 module of the YOLOV4 head with a deep separable convolution (DSC [16]) to further reduce the amount of calculation

(5) We further replaces the Leaky ReLU in the CBLn module with ReLU6 as the activation function to make our network run better on mobile devices

(6) We add squeeze-and-excitation (SE) [17] network to YOLOv4 head network. The SE network could greatly reduce the interference information and improve the network prediction head's ability to extract the feature information

(7) We conducted experiments on VOC2012 and VOC2007 and the WiderPerson dataset and then contradistinguish them with existing methods. Our method has achieved satisfactory results in terms of accuracy and speed

2. Related Work

The innovation of this study focuses on the transformation of the YOLO algorithm. We describe the related work from the development of pedestrian detection, the development of YOLO algorithm, and the research and development of network structures similar to GhostNet.

2.1. Algorithm Development for Pedestrian Detection. The traditional pedestrian detection method was mainly to extract features of the pedestrians. Dalal and Triggs [18] proposed a histogram of oriented gradient (HOG) which used the directionality of the edge to describe the overall appearance of pedestrians. Yan et al. [19] solve the speed bottleneck of deformable part model (DPM) while maintaining the accuracy in detection on challenging datasets. Three prohibitive steps in cascade version of DPM are accelerated, including 2D correlation between root filter and feature map, cascade part pruning, and HOG feature extraction. However, the extraction steps of the manual extraction method were cumbersome. The recognition algorithm was computationally expensive and the real-time performance was unsatisfactory.

2.2. Development of Deep Learning Algorithms Similar to YOLO Algorithm. Due to the latest development of deep learning research, pedestrian detection had achieved rapid development. At present, target detection algorithms based on deep learning could be divided into two categories: (1) a two-stage detection algorithm represented by region-based fully convolutional neural network (R-FCN) [20] and (2) a single-stage detection method represented by You Only Look Once (YOLO). The two-stage detection method realized the cascade structure and the amount of network calculation increased. This method had high accuracy and poor real-time performance. Regarding the single-stage detection method, Redmon proposed YOLO which was the first single-stage detection method based on deep learning in 2016. It creatively combined the candidate area and target recognition and had very good real-time performance. Then, Redmon future proposed YOLOv2 and YOLOv3, which significantly improved the detection performance and made the YOLO series of methods widely used in various tasks. In 2020, Bochkovskiy improved the network structure of YOLOv3 and proposed YOLOv4. The YOLOv4 greatly improved the detection accuracy. Recently, Jocher proposed YOLOv5 which greatly integrated other most advanced techniques. Although YOLOv5 is more flexible and faster than YOLOv4, the accuracy of YOLOv5 is lower than YOLOv4.

2.3. Improved Method for YOLO. Literature [21] proposed a new feature learning method based on sparse autoencoder (SAE) for human body detection of in-depth pictures, and in order to further reduce the computational cost of SAE, it also introduced convolutional neural networks and pooling to reduce training complexity. Literature [22] proposed a method to replace the CSPDark53 backbone network with MobilNet v3 [23]. At the same time, it introduced a deep separable attention module based on deep separable convolution and coordinated attention network to replace standard convolution and achieved good accuracy and running speed. Literature [24] proposed a small target deep convolution recognition algorithm based on the improved YOLOV4 network, introduced different pool core size spatial pyramids, and adopted an improved adaptive anchor structure, while adopting two cross-level partial parallel structures, and achieved good small target detection results. Wu et al. [25] trimmed the trained model by using the channel pruning algorithm of YOLOv4, which simplified the structure and parameters of the model to a certain extent. Yu et al. [26] used focal loss to optimize the loss function to improve accuracy and used the pruning algorithm to simplify the network. Yang et al. [27] replaced the activation function of YOLOv4 with the activation function of ELU and added the SE attention module on the backbone network to realize the deployment of the model on the embedded platform. Ke et al. [28] use a nonoverlapping image block data enhancement method for processing and then input it into the YOLOv3 detector to obtain target location information and

use a pedestrian recognition model based on lcnn to extract target features.

3. Methodology

We proposed a neural network called Little-YOLOv4. The backbone network of Little-YOLOv4 was GhostNet, which is used for preliminary feature extraction. We propose a neural network named Little-YOLOv4. The backbone network of Little-YOLOv4 is GhostNet for preliminary feature extraction. The neck detection network consists of spatial pyramid pooling (SPP) [29] and improved PANet for further data feature extraction; meanwhile, DO-DConv is introduced in this study to replace the 3×3 and 5×5 criteria for neck convolution, which can reduce the computational complexity of the algorithm without affecting the extraction strength. The SENet is also embedded in the YOLOv4 head detection network to greatly improve the ability to extract effective information, and the 1×1 convolution in the original CBL1 module is replaced by DSC to further reduce the amount of computation. Finally, we change the activation function inside the CBLn module in the neck and head network to ReLU6. The specific overall algorithm framework is shown in Figure 1.

3.1. Pedestrian Feature Extraction Based on GhostNet. In order to efficiently extract image features, YOLOv4 adopted CSPDarknet53 which required a huge amount of computation during operations. In order to ensure the efficiency and accuracy of feature extraction, we chose GhostNet to replace CSPDarknet53. GhostNet is a lightweight network structure composed of a series of Ghost bottleneck layers (G-bneck). G-bneck is similar to the basic residual block in ResNet [30]. It mainly composes of 2 stacked Ghost modules; the specific structure is shown in Figure 2. The first Ghost module is used to increase the number of channels. The second Ghost module is used to decrease the number of channels. By using these two stages, the shortcut is consistent. The input and output of these two Ghost modules are connected. G-bneck with stride 2 inserts depthwise separable convolutional layers to reduce the impact of feature set changes and parameter scale. The Ghost module is a module that combines ordinary convolution operations and linear operations. It performs linear transformation on the generated convolution feature maps to obtain similar feature maps and the high-dimensional convolution is produced. The Ghost module can reduce the model parameters and calculations. In Figure 3, Y in the figure represents the convolution feature maps and Y' represents the similar feature maps generated by linear operation.

The GhostNet used in this study is shown in Table 1. The first process was an ordinary convolution with 16 convolution kernels. The second process was a series of G-bnecks, and the stride of G-bnecks used in each stage is shown in Table 1. The third process was an ordinary convolution and the output channel was increased to 960. The final process was the average pooling layer and ordinary convolution which converted the feature map into a 1280-dimensional feature vector for classification. Extrusion and excitation modules were applied to the residual layers in some G-bnecks. The network structure of YOLOv4 had been to lightweight by using GhostNet, but it also reduced the ability of feature fusion.

The Ghost module is used to replace the ordinary convolution module, and the theoretical speedup is analyzed as follows. The operation of generating n feature images for any convolutional layer can be expressed as

$$Y_0 = X \odot f + b, \tag{1}$$

where $X \in R^{h \times c \times w}$, $f \in R^{c \times k \times k \times m}$ is the convolution kernel of this layer, $\odot$ is the convolution operation, b is the bias term, c is the number of channels, h and w are the images, respectively. The size of the convolution kernel is $k \times k$; after the above operations, the feature map $Y_0 \in R^{h' \times w' \times m'}$; the required floating-point number $h' \times w' \times m' \times c \times k \times k$. The original output features are some intrinsic features and usually a small number can be generated by an ordinary convolution operation, namely,

$$Y = X * f. \tag{2}$$

In the formula, $Y_0 \in R^{h' \times w' \times m}$ is the ordinary convolution output; $f' \in R^{c \times k \times k \times m}$ is the convolution kernel used; since $m \le n$, the bias term is simplified.

Now, it is necessary to obtain an n-dimensional feature map and perform a series of simple linear transformations on the inherent feature map with only m dimensions:

$$y_{ij} = \Phi_{ij}\left(y_i'\right) \quad \forall_i = 1, \cdots, m; j = 1, \cdots, s. \tag{3}$$

y_i' in the formula is the ith feature map in the inherent feature map; $\Phi_{i,j}$ is the linear transformation function of the jth linear transformation of the ith feature map. Finally, an identity map $\Phi i, s$ is added, and the inherent feature map is added to the feature map obtained by linear transformation to retain the inherent feature map.

Assuming that the Ghost module contains an inherent feature map and $m \times (s-1) = n/s \times (s-1)$ linear transformation operations, the size of each operation kernel is $d \times d$, and the Ghost module improves the theory of ordinary convolution. The speedup is

$$\begin{aligned} r_c &= \frac{n \times h' \times w' \times c \times k \times k}{(n/s) \times h' \times w' \times c \times k \times k + (s-1) \times (n/s) \times h' \times w' \times d \times d} \\ &= \frac{c \times k \times k}{(1/s) \times c \times k \times k + ((s-1)/s) \times d \times d} \approx \frac{s \times c}{s + c - 1} \approx s. \end{aligned} \tag{4}$$

If $d \times d = k \times k$, $s << c$, then the theoretical parameter compression ratio is

$$r_c = \frac{n \times c \times k \times k}{(n/s) \times c \times k \times k + ((s-1)/s) \times n \times d \times d} \approx \frac{s \times c}{s + c - 1} \approx s. \tag{5}$$

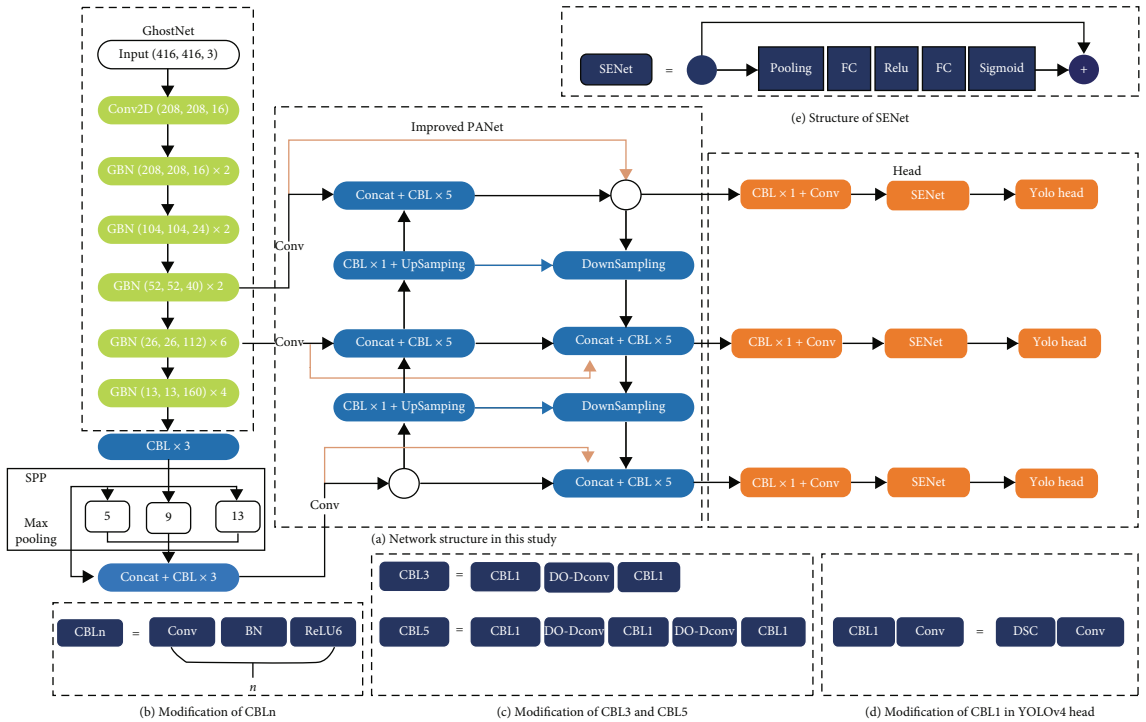


Figure 1: The network structure of Little-YOLOv4.

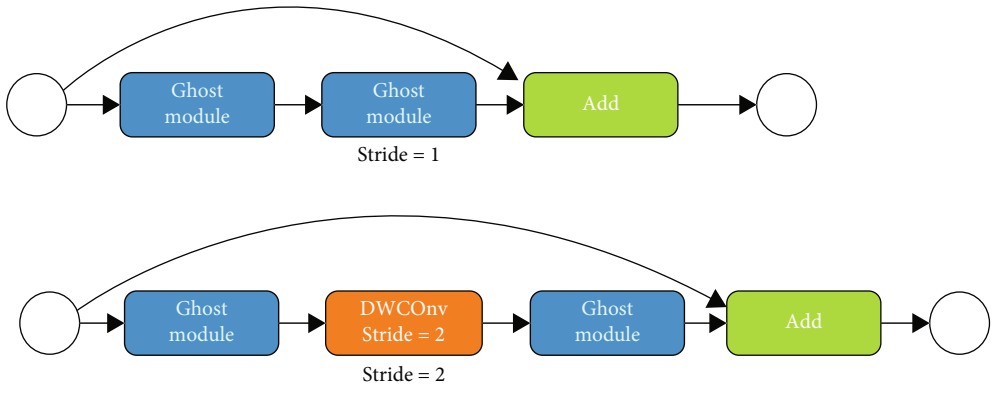


Figure 2: Ghost bottleneck layer.

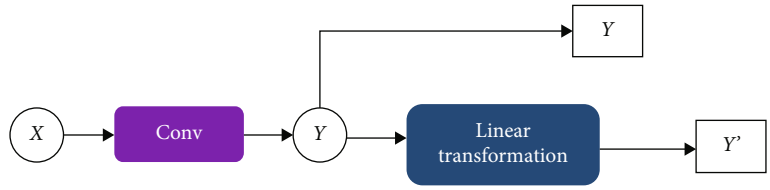


Figure 3: Schematic diagram of Ghost module.

The conclusion is as follows: when $d \times d = k \times k$, the theoretical parameter compression ratio of using Ghost module instead of ordinary convolution is approximately equal to the theoretical speedup ratio, and the improved speed is theoretically s times higher than the original.

3.2. Improved PANet. We applied the path fusion ideas used in BiFPN to improve PANet in YOLOv4. In BiFPN, the input nodes and output nodes of the same layer could be connected across layers to ensure that more features were incorporated without increasing the loss. The Little-

Table 1: GhostNet construction method diagram (reproduced from Hongtao Zheng and Yan Liu in 2022 from document [31]).

Input	Operator	#exp	Out	SE	Stride
4162 × 3	Conv2d	—	16	—	2
2082 × 16	G-bneck	16	16	—	1
2082 × 16	G-bneck	48	24	—	2
1042 × 24	G-bneck	72	24	—	1
1042 × 24	G-bneck	72	40	1	2
522 × 40	G-bneck	120	40	1	1
522 × 40	G-bneck	240	80	—	2
262 × 80	G-bneck	200	80	—	1
262 × 80	G-bneck	184	80	—	1
262 × 80	G-bneck	184	80	—	1
262 × 80	G-bneck	480	112	1	1
262 × 112	G-bneck	672	112	1	1
262 × 112	G-bneck	672	160	1	2
132 × 160	G-bneck	960	160	—	1
132 × 160	G-bneck	960	160	1	1
132 × 160	G-bneck	960	160	—	1
132 × 160	G-bneck	960	160	1	1
132 × 160	Conv2d	—	960	—	1
132 × 960	AvgPool	—	—	—	—
12 × 960	Conv2d	—	1280	—	1
12 × 1280	FC	—	1000	—	—

YOLOv4 performed cross-layer connections on the same level of PANet (the orange lines in Figure 1). The path from low-level information to high-level information could be shortened, and the semantic features could be combined together. In BiFPN, the adjacent layers could be merged in series. The adjacent layers of PANet in Little-YOLOv4 were merged in series (the blue lines in Figure 1). The improved PANet had the characteristics of bidirectional cross-scale connection and weighted feature fusion, which improved the feature fusion ability.

3.3. DO-DConv. CNN is applied in the field of computer vision. Increasing the number of linear layers-nonlinear layers in CNN can increase the expressive ability of the network and improve the performance of the network. However, few people consider only adding a linear layer, which will lead to overfitting, because multiple continuous linear layers can be replaced by a linear layer, which makes a linear layer with fewer parameters. In response to this phenomenon, the author in [32] formed an overparameterized convolution layer by adding an additional depthwise convolution operation to a common convolution layer and named it DO-Conv. Using DO-Conv instead of ordinary convolution can not only speed up the training process of the network but also achieve better results than using ordinary convolutional layers in a variety of computer vision tasks. At inference time, DO-Conv can be converted to traditional convolution operations, so replacing traditional convolution in one network with DO-Conv does not increase computational requirements. Similarly, DO-DConv is obtained by adding an additional depthwise convolution operation to the depthwise convolution.

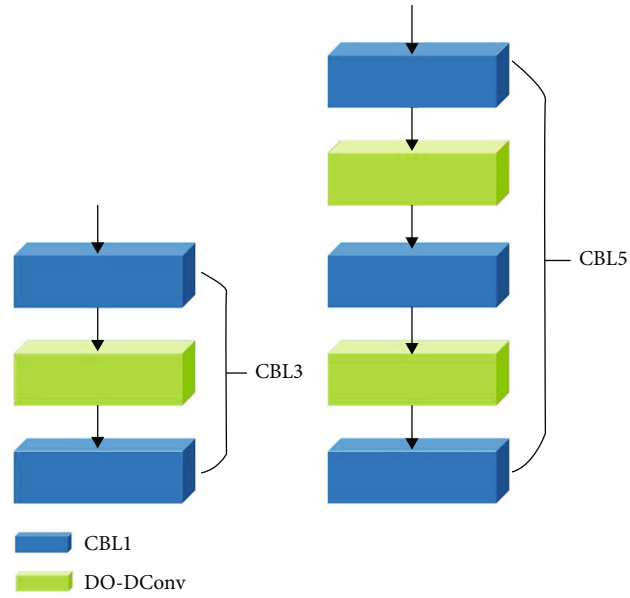


Figure 4: Structure of modified CBL3 and CBL5.

We replaced the standard convolution of 3 × 3 and 5 × 5 of YOLOv4 neck network with DO-DConv. Figure 4 shows the structure of the modified CBL3 and CBL5.

And DO-Conv can be converted to traditional convolution operations, so replacing traditional convolution with DO-Conv will not increase computing requirements. Not only can we use DO-Conv instead of traditional convolution to speed up the convergence speed and improve network performance; we can also use the same operation in depthwise convolution to form the DO-DConv of our article.

As shown in Figure 5, DO-DConv is similar to DO-Conv in the training phase to get two kinds of weights. In the inference phase, these two weights will be merged into one weight.

3.4. DSC Further Reduces Algorithm Parameters. The 1 × 1 standard convolutional network in the YOLOv4 head CBL1 module is replaced by a deeply separable convolutional network, which further reduces the network calculation cost in practical applications. The modified part of CBL1 is shown in Figure 6. The standard convolutional network calculation uses a weight matrix to realize the joint mapping of spatial dimension features and channel dimension features. The cost is high computational complexity, large memory overhead, and many weight coefficients. DSC has the advantages of high computational efficiency.

DSC specifically divides the traditional convolution operation into two steps. Assuming that it was originally a 3 × 3 convolution, then DSC first generates M results without summing the M feature maps input by $M3 \times 3$ convolution kernels one-to-one convolution and then uses $N1 \times 1$

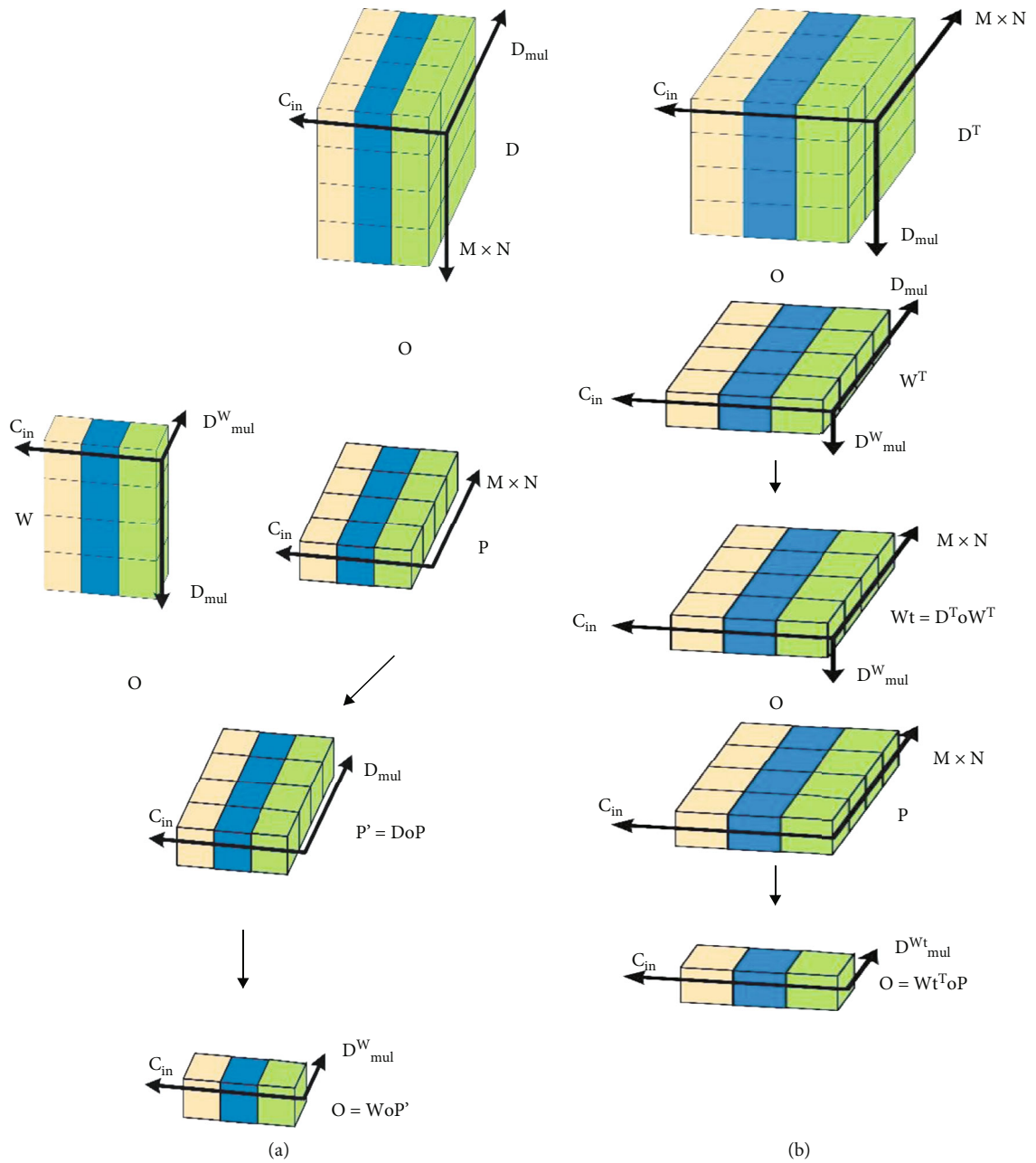


Figure 5: DO-DConv formation method.

convolution kernels to normally convolve the previously generated M results, sum, and finally generate N results. Therefore, the literature divides DSC into two steps, as shown in Figure 7; one step is called depthwise convolution, which is b in the figure below, and the other step is pointwise convolution, which is c in the figure below.

Assuming that the size of the feature map we input is $D_F \times D_F$, the dimension is M, the size of the filter is $D_k \times D_k$, and the dimension is N and assuming that padding is 1 and the strike is 1, then for the original convolution operation, the number of matrix operations required is $D_k \times D_k \times M \times N \times D_F \times D_F$, and the parameter of the convolution kernel is $D_k \times D_k \times M \times N$; and the number of matrix operations required for DSC is $D_k \times D_k \times M \times D_F \times D_F + M \times N \times D_F \times D_F$, and the parameter of the convolution kernel is $D_k \times D_k \times M + M \times N$. Since the convolution process is mainly the process of reducing spatial dimensions and increasing channel dimensions, that is, $N > M$, the amount of convolution kernel parameters of standard convolution is greater than that of DSC. At the same time, the ratio of the parameter amount of DSC to the standard convolution parameter amount is as follows:

$$\frac{D_k \times D_k \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_k \times D_k \times M \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_k^2}. \quad (6)$$

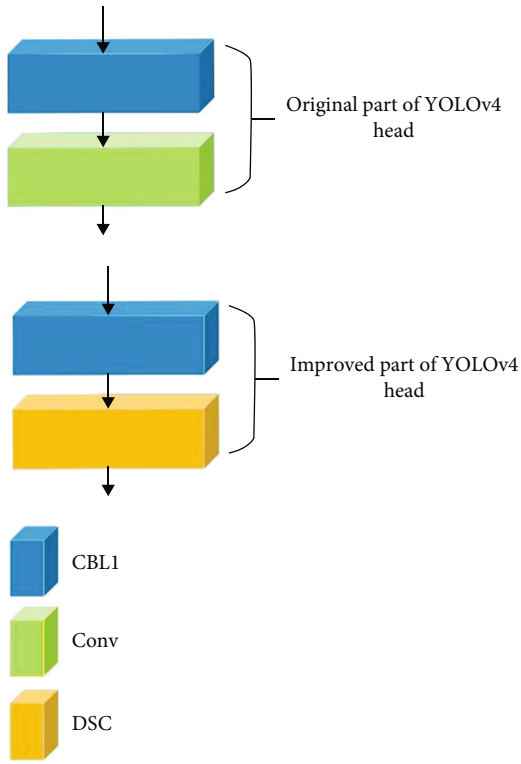


Figure 6: The modified part of CBL1 on the head of Little-YOLOv4.

From formula (11) we can get a convolution kernel with a size of 3×3, and his calculation amount is reduced to 11.1% of the standard convolution.

3.5. ReLU6 Reduces Learning Cycle. The linear rectified function (ReLU) is also known as the modified linear unit. Literature [33] conducted a comparative test on ReLU and ordinary Sigmoid series functions, and it can be concluded that the use of ReLU can greatly reduce the learning cycle, and the overall efficiency and speed are good. It is often used to achieve parameter sparsity training. The relationship between ReLU6 and ReLU is mainly to have good numerical resolution under the low precision of float16 on the mobile terminal. ReLU6 is an ordinary ReLU but limits the maximum output value to 6 (clip the output value). Because if there is no restriction on the activation range of ReLU, the output range is from 0 to positive infinity. If the activation value is very large and distributed in a large range, the low-precision float16 cannot describe such a large range well and accurately, which will bring accuracy loss. Its calculation formula is as follows:

$$\text{Re LU}(x) = (x)^{+} = \max(0, x), \tag{7}$$

$$\text{LeakyReLU}(x) = \begin{cases} x, & x \geq 0, \\ ax, & \text{otherwise}, \end{cases} \tag{8}$$

$$\text{Re LU6}(x) = \min(\max(0, x), 6). \tag{9}$$

3.6. Introduction of SE (Squeeze-and-Excitation Networks). Since pedestrian detection is often accompanied by complex environments, this paper embeds SENet in the YOLOv4 head network, which has the following advantages: first, it reduces the input interference of interference information, which can greatly improve the head network's ability to understand feature information. Extraction ability thereby improves the recognition performance of the entire network. Second, only a small amount of parameters is introduced, which greatly improves the recognition ability and reduces the running frame rate by a small amount.

The attention mechanism has been widely used in neural networks. It adjusts the weight of each channel information, assigns different weights to each channel information, and then filters the channel information according to the weight, which can effectively reduce the influence of interference information. SENet is a typical representative of channel attention mechanism.

According to the analysis in Figure 8, the input feature map X has C channels, the space size of each channel is $H \times W$, and the global average pooling is performed on each channel; the calculation formula of the channel weight Z is shown as follows:

$$Z = F_{sq}(X_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_C(i, j). \tag{10}$$

The output Z is a one-dimensional array of length C, which represents the weight obtained by the compression channel; (i, j) represents the point whose abscissa and ordinate are i and j, respectively, on the feature map of size $H \times W$.

Next, the activation function needs to be used to model the correlation degree of each channel weight. The formula is shown in

$$S_c = F_{ex}(Z, W) = \text{Sigmoid}(W_2 \times \text{Re } LU(W_1, Z)). \tag{11}$$

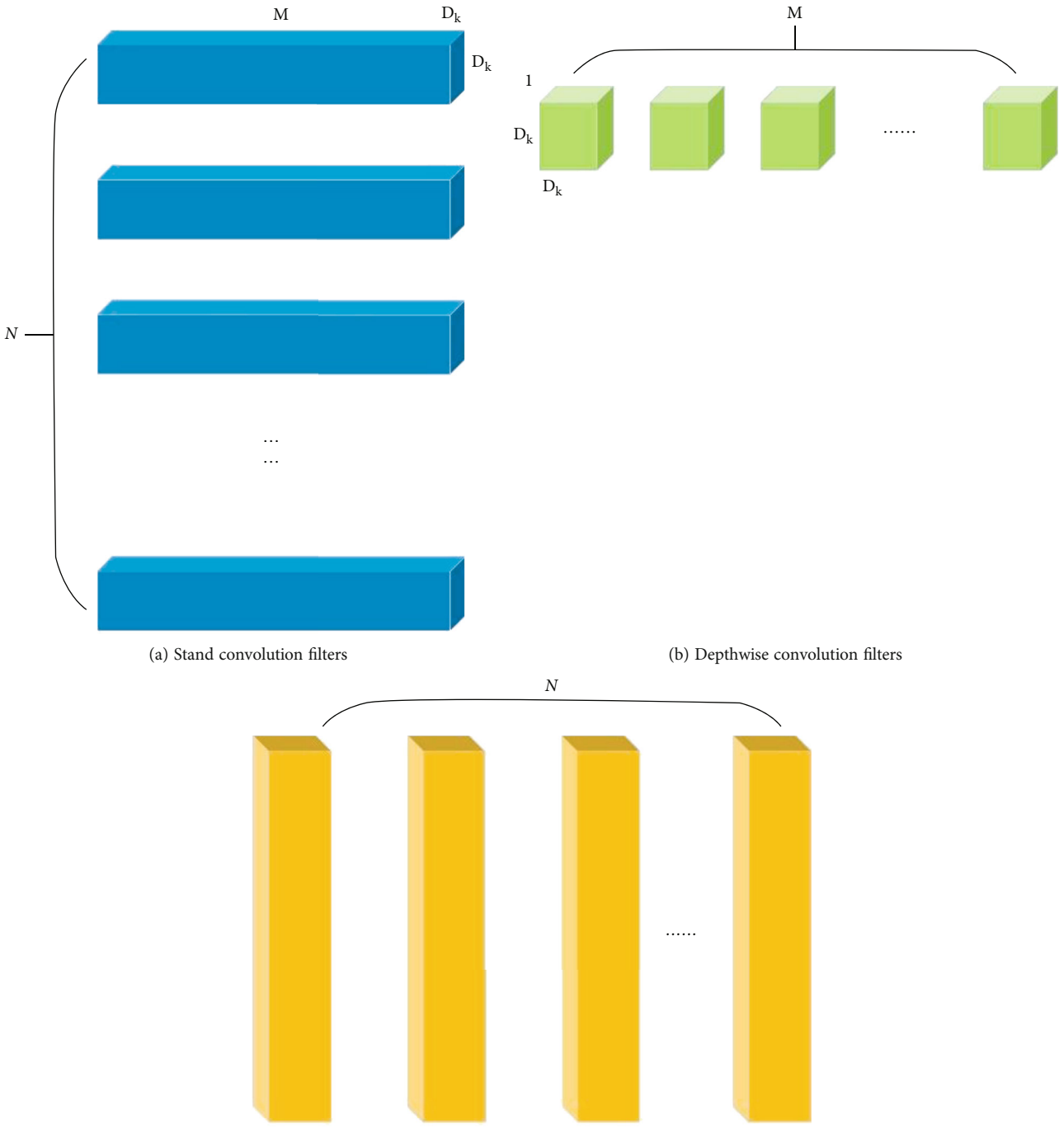
Among them, the dimension of S_c is $1 \times 1 \times C$. The channel attention weights need to be obtained through operations such as fully connected layers and nonlinear learning. The dimension of W_1 is $C/r \times C$, the dimension of W_2 is $C \times C/r$, and r is the scaling factor.

Finally, the input channel is weighted and adjusted, and the channel attention weighting formula is the following:

$$\widehat{X} = F_{\text{scale}}(X_C, S_C) \otimes S_C. \tag{12}$$

where $\otimes$ represents the multiplication of elements and $\widehat{X}$ represents the result after attention network processing.

Although SENet can be embedded in various convolutional layers, it will lead to an excessive amount of overall network parameters. Therefore, this paper chooses to introduce SENet into some convolutional blocks. The formula for the number of parameters of SENet is shown in



(a) Stand convolution filters

(b) Depthwise convolution filters

(c) Depthwise separable convolution

Figure 7: Structure diagram of DSC.

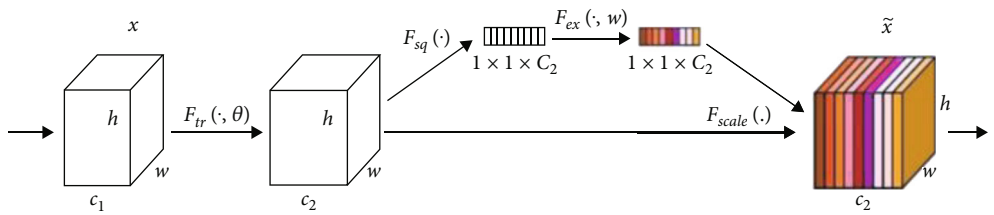


Figure 8: A squeeze-and-excitation block (reproduced from Jie Hu and Li Shen and Gang Sun in 2018 from document [17]).

$$N = \sum_{i=1}^{n} \left(\frac{2 \times C_i \times C_i}{r} \right). \tag{13}$$

Among them, N is the number of parameters, n is the number of embedded SENets, and the number of parameters mainly comes from the two fully connected layers placed, so C and r and n determine the size of the parameters brought to the network.

The introduction of SENet into the YOLOv4 head network can greatly improve the extraction ability of effective information (smoke and flame), thereby improving the recognition ability of the entire network.

4. Experiments and Analysis

4.1. Dataset Analysis. The datasets used for training in this experiment are mainly the tag information about people in VOC2012 and VOC2007 and the WiderPerson dataset, but in order to increase the generalization ability of the dataset, I used a car camera to randomly self-made 10,000 pedestrian pictures and distributed them into training set, validation set, and test set according to the ratio of 7 : 1 : 2.

4.2. Ancor Box. In order to fit the person category, the a priori box in the Little-YOLOv4 algorithm in this research is obtained by using the K-means clustering dataset method. The input picture size is 416 × 416. When the K-means clustering is used for 78 iterations, the ratio of the a priori box to the real box reaches about 76.54%, and nine a priori boxes are obtained, as shown in Table 2.

4.3. Model Building and Training. Our laboratory operating environment configuration is shown in Table 3.The experimental training and testing platform is RTX 3070, with 8G video memory. The network model training is based on the TensorFlow 2.5 deep learning framework of GhostNet and PSPDarknet53. The size of the input image is 416 by 416.

4.4. Evaluation Criteria. We use FPS, precision, recall, mAP, and other indicators to evaluate our proposed method. The test set is divided into two categories; one is a positive sample and the other is a negative sample. TP is the number of samples predicted to be positive as a positive sample; FP is the number of samples predicted to be positive as a negative sample. FN is the number of positive samples predicted to be negative samples; TN is the number of negative samples predicted to be negative.

4.4.1. Detection Speed FPS (Frames Per Second). The evaluation standard of detection speed used in this article is FPS, which refers to the number of frame per second. The larger the FPS, the more the frame rate transmitted by Meibiao, and the smoother the displayed image will be. In order to meet the real-time requirements of human detection, the larger the value of FPS, the smoother the picture will be and the better the use effect will be.

4.4.2. Accuracy mAP (Mean Average Precision). The definition of mAP is shown in formula (14), which represents the mean value of the average accuracy APi of n types of targets. In this experiment, $n = 1$.

Table 2: A priori frame size.

Size	Ancor box
13 × 13	(224,270),(251,370),(367,364)
26 × 26	(103,275),(140,361),(157,180)
52 × 52	(23,45),(63,115),(78,218)

Table 3: Software and hardware configuration.

Component	Configuration
Operating system	Ubuntu 18.04
GPU	Nvidia GeForce RTX 3070 16G
GPU acceleration library	CUDA 11.2 cuDNN v8.2.1
Deep learning framework	TensorFlow 2.5
Programming language	Python 3.9

$$\text{mAP} = \frac{\sum AP}{N(\text{class})} = \sum AP. \tag{14}$$

4.4.3. Recall. The recall rate is defined as the following formula (15), which represents the proportion of correctly detected targets among all targets in the test dataset.

$$\text{Re call} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{15}$$

4.4.4. Precision. Precision can measure the accuracy of object detection, and the specific definition is shown in

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{16}$$

4.5. Experimental Results and Analysis. First, we compare the overall parameters of Little-YOLOv4 and other classical detectors. As can be seen from Figure 9, the number of parameters of YOLOV4 is 64040000, while the total parameters of Little-YOLOv4 are about 9126761, which is 14.3% of YOLOv4, which is significantly smaller than most object detectors. This can prove that our proposed lightweight method can significantly reduce the total number of parameters of the original model.

In order to be able to detect the performance of the algorithm in this study, we conduct experiments on the verification set of the dataset and at the same time generate more samples by rotating the angle, adjusting the saturation, and adjusting the exposure and color. Figure 10 shows some detection situations of the algorithm on the dataset of this paper. The detection effect is significant, which proves that the network in this study is effective in human detection. At the same time, in order to verify the impact of various improvements in the network structure of this article on the detection performance of the detector, we conducted a series of additional tests on the training platform, as shown

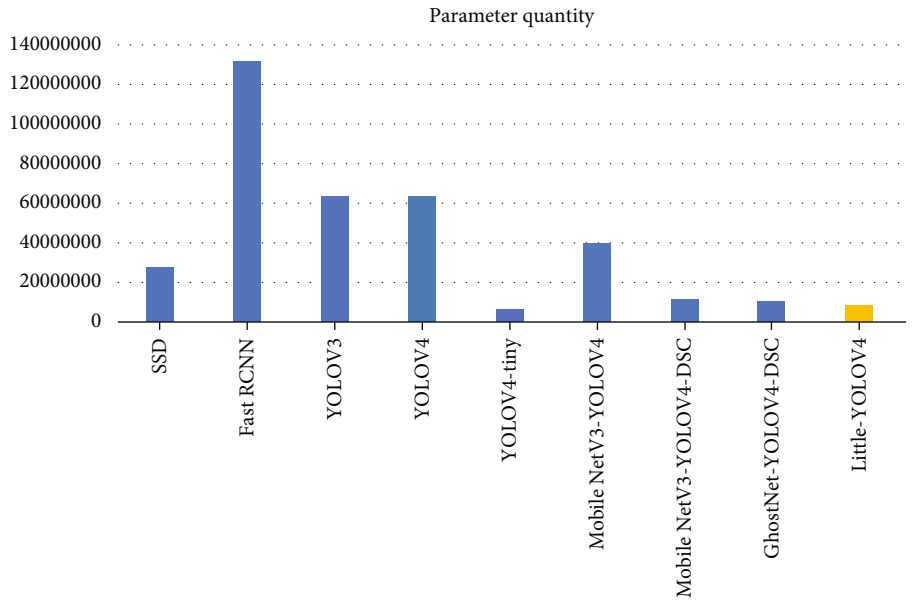


Figure 9: Statistics of parameters of various network structures.

Figure 10: The detection result of Little-YOLOv4.

in Table 4. Using GhostNet instead of CSPDarknet, mAP will decrease slightly, but FPS will increase greatly. For example, in experiments 1 and 2, mAP decreases from 87.27% to 86.28% but FPS increases from 48 to 76. It can be seen that the lightweight network GhostNet can greatly improve the real-time performance of the algorithm. When BiFPN's PANet is introduced, the FPS will decrease slightly, but the mAP will be improved. For example, in experiments 1 and 3, the FPS is reduced from 48 to 46, but the mAP is increased from 87.27% to 88.40%. It can be seen that the improved PANet can improve the performance of the algorithm. Introducing DO-DConv, the FPS is stable and

Table 4: Ablation study on the people dataset.

	GhostNet	Improved PANet	DO-DConv	DSC	SENet	mAP (%)	FPS
1 (YOLOv4)	—	—	—	—	—	87.27	48
2	+	—	—	—	—	86.28	76
3	—	+	—	—	—	88.40	46
4	—	—	+	—	—	87.58	48
5	—	—	—	+	—	86.51	63
6	—	—	—	—	+	91.29	45
7	+	+	—	—	—	87.68	74
8	+	+	+	—	—	87.64	74
9	+	+	+	+	—	87.41	85
10 (Little-YOLOv4)	+	+	+	+	+	90.11	79

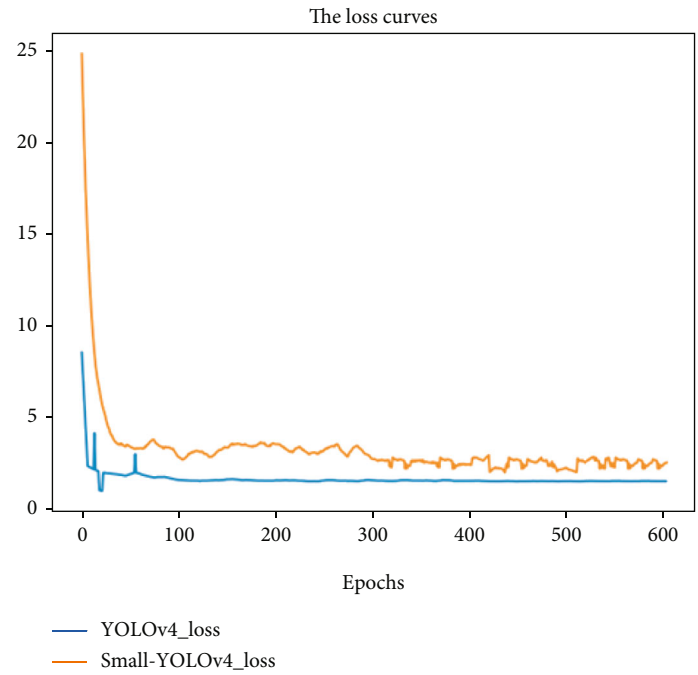


Figure 11: The network training process of YOLOv4 and Little-YOLOv4.

unchanged, but the mAP will be improved. For example, in experiments 1 and 4, the FPS is reduced from 48 to 46, but the mAP is increased from 87.27% to 87.58%. It can be seen that DO-DConv can improve the performance of the algorithm. When DSC is introduced, mAP is slightly reduced, but FPS is greatly improved. For example, in experiments 1 and 5, mAP is slightly reduced from 87.27% to 86.51%, but FPS is increased from 48 to 63. It can be seen that DSC can greatly realize the lightweight of the algorithm. After the introduction of SENet, the FPS will drop slightly, but the mAP will be greatly improved. For example, in experiments 1 and 6, the FPS drops from 48 to 45, but the mAP increases from 87.27% to 91.29%. It can be seen that SENet has a great contribution to the improvement of algorithm accuracy and is a necessary link. GhostNet, BiFPN-based PANet, DO-Dconv, DSC, and SENet have their own emphasis on the improvement of the algorithm and complement each other. Therefore, the GhostNet-YOLOv4 algorithm proposed in this paper has achieved good overall performance. For example, in experiment 10, the mAP of the algorithm in this paper is 90.11%, FPS is 79, and it can accurately identify pedestrians in real time.

We have compared the network training process of YOLOv4 and Little-YOLOv4 as shown in Figure 11. The training process is mainly concentrated after 600 hours. Comparing their loss curves, due to the structural characteristics of DO-DConv, the network has a good convergence speed and a low error rate. It shows good convergence and strong learning ability. It has good convergence and strong learning ability.

4.6. Comparison with Other Methods. To further verify the specific performance of Little-YOLOv4 in human detection, we compare it with methods commonly used in the

Table 5: Comparison of different methods.

Method	Precision (%)	mAP (%)	Recall (%)	FPS
1 (Faster-RCNN [2])	82.76	85.23	77.54	17
2 (SSD [6])	84.48	86.36	75.76	57
3 (EfficientDet-D2)	78.37	81.82	76.43	21
4 (YOLOv3)	82.71	85.33	78.31	51
5 (YOLOv4)	85.93	87.28	80.08	48
6 (YOLOv4-tiny)	72.49	75.61	68.63	168
7 (Little-YOLOv4)	86.69	90.11	81.55	79

literature in recent years on the test set of this dataset. The performance test results of these methods are shown in Table 5. For example, comparing experiments 1, 2, and 3 with 4 and 5, it can be seen that the comprehensive effect of experiment 5 is much better than that of the other 4 experiments. Compared with experiment 5, although the accuracy of experiment 6 is reduced with a certain probability and recognition, the speed is greatly improved, but comparing experiment 7 with experiments 5 and 6, the algorithm of experiment 7 not only has a considerable frame rate but also has a high accuracy. These results show that Little-YOLOv4 has achieved better accuracy and speed. They are only slower than YOLOv4-tiny and only less accurate than YOLOv4.

5. Conclusions

This paper proposes an improved YOLOv4 lightweight detector Little-YOLOv4. It uses a lightweight detection network GhostNet as the backbone network to replace CSPDarknet, which greatly reduces network parameters while reducing very little accuracy. We improve the PANet in YOLOv4 combined with the BiFPN path fusion method, which can shorten the path from low-level information to high-level information, and build the residual structure of the feature pyramid network, so as to integrate richer semantic features and save spatial information. We replace the 3×3 and 5×5 standard convolutions at the neck of YOLOV4 with DO-DConv to speed up inference without increasing the computational complexity of inference. We replace the 1×1 standard convolution in the CBL1 module of the YOLOV4 head with a depthwise separable convolution to further reduce the computational cost. Further replace the Leaky ReLU in the CBLn module with ReLU6 as the activation function to make our network run better on mobile devices. Finally, we introduce SE into the YOLOv4 head network, which can greatly improve the accuracy while introducing less computation. To construct the human dataset, we use the K-means algorithm for clustering and initialize the prior box. In the pedestrian detection task, the mAP for pedestrian detection for the Little-YOLOv4 detector is 90.11 and the FPS is 79. Compared with the YOLOv4 detector, the mAP is increased by 2.83%, the parameters are reduced to 14.3% of the YOLOv4 parameters, and the FPS is increased by 1.65 times. At the same time, the detection accuracy of the algorithm for complex environments such as poor lighting conditions, background confusion, and human occlusion is also very good.

The detection accuracy and speed of Little-YOLOv4 are more balanced, and it is suitable for laying on embedded hardware platforms such as Jetson Nano, by laying the algorithm on a corresponding small embedded device and installing the device on a street-like. In the monitoring place or vehicle monitoring, it will help greatly improve the accuracy and real-time performance of pedestrian detection, which will provide certain help for the development of unmanned driving, pedestrian monitoring, and other fields. Little-YOLOv4 on Jetson Nano reaches 26 FPS, which meets the real-time requirements. Compared with the existing human body detection methods, Little-YOLOv4 has made certain progress in practicability and innovation. But it is worth noting that it still has a certain possibility of false detection and false detection.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

There were no known competing financial interests or personal relationships that may have affected this work.

Acknowledgments

This work was supported by the Zhejiang University City College Scientific Research Fund (No. X-202106), the Natural Science Foundation of Zhejiang Province, China (Grant No. LQ22F010002), the 2021 National Innovation Training Project for College Students (No. 202113021008), and the Zhejiang University Student Science and Technology Innovation Activity Plan (Xinmiao Talent Plan) (2021R437010).

References

[1] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. H. Hoi, "Deep learning for person re-identification: a survey and outlook," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 2872–2893, 2022.

[2] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Venice, Italy, 2015.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, 2014.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[5] K. He, G. Gkioxari, P. Dollar, and R. Girshick, Eds., "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2961–2969, Venice, Italy, 2017.

[6] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: single shot multibox detector," in *European Conference on Computer Vision*, pp. 21–37, Cham, 2016.

[7] C. Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: deconvolutional single shot detector," 2017, https://arxiv.org/abs/1701.06659.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, Las Vegas, NV, USA, 2016.

[9] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, Honolulu, HI, USA, 2017.

[10] J. Redmon and A. Farhadi, "Yolov 3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[11] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: optimal speed and accuracy of object detection," 2020, https://arxiv.org/abs/2004.10934.

[12] B. Han, Y. Wang, Z. Yang, and X. Gao, "Small-scale pedestrian detection based on deep neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 3046–3055, 2020.

[13] Z. Yi, S. Yongliang, and Z. Jun, "An improved tiny-yolov3 pedestrian detection algorithm," *Optik*, vol. 183, pp. 17–23, 2019.

[14] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghostnet: more features from cheap operations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1580–1589, Seattle, WA, USA,2020, 2020.

[15] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10781–10790, Seattle, WA, USA, 2020.

[16] F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, Honolulu, HI, USA, 2017.

[17] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, Salt Lake City, UT, USA, 2018.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886–893, San Diego, CA, USA, 2005.

[19] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2497–2504, Columbus, OH, USA, 2014.

[20] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: object detection via region-based fully convolutional networks," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[21] S. Z. Su, Z. H. Liu, S. P. Xu, S. Z. Li, and R. Ji, "Sparse auto-encoder based feature learning for human body detection in depth image," *Signal Processing*, vol. 112, pp. 43–52, 2015.

[22] Y. Wang, C. Hua, W. Ding, and R. Wu, "Real-time detection of flame and smoke using an improved YOLOv4 network," *Signal, Image and Video Processing*, vol. 16, no. 4, pp. 1109–1116, 2022.

[23] A. Howard, M. Sandler, G. Chu et al., "Searching for mobilenetv 3," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, Seoul, Korea (South), 2019.

[24] F. Li, D. Gao, Y. Yang, and J. Zhu, "Small target deep convolution recognition algorithm based on improved YOLOv4," *International Journal of Machine Learning and Cybernetics*, pp. 1–8, 2022.

[25] D. Wu, S. Lv, M. Jiang, and H. Song, "Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments," *Computers and Electronics in Agriculture*, vol. 178, article 105742, 2020.

[26] Z. Yu, Y. Shen, and C. Shen, "A real-time detection approach for bridge cracks based on YOLOv4-FPM," *Automation in Construction*, vol. 122, article 103514, 2021.

[27] Y. Yang, G. Xie, and Y. Qu, "Real-time detection of aircraft objects in remote sensing images based on improved YOLOv4," in *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5, pp. 1156–1164, Chongqing, China, 2021.

[28] X. Ke, X. Lin, and L. Qin, "Lightweight convolutional neural network-based pedestrian detection and re-identification in multiple scenarios," *Machine Vision and Applications*, vol. 32, no. 2, pp. 1–23, 2021.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, Las Vegas, NV, USA, 2016.

[31] H. Zheng and Y. Liu, "Lightweight fall detection algorithm based on AlphaPose optimization model and ST-GCN," *Mathematical Problems in Engineering*, vol. 2022, Article ID 9962666, pp. 1–15, 2022.

[32] J. Cao, Y. Li, M. Sun et al., "Do-conv: depthwise over-parameterized convolutional layer," *IEEE Transactions on Image Processing*, vol. 31, pp. 3726–3736, 2022.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural n," *Advances in Neural Information Processing Systems*, vol. 25, 2012.