



A fast SSD model based on parameter reduction and dilated convolution

Xinliang Zhang¹ · Heng Xie¹ · Yunji Zhao¹ · Wei Qian¹ · Xiaozhuo Xu¹

Received: 22 August 2020 / Accepted: 13 April 2021 / Published online: 25 April 2021
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

Deep learning networks always compromise between speed and accuracy for their in-depth feature extraction. In this paper, we present a modified single shot multibox detector (SSD) model to achieve high speed while maintaining satisfactory accuracy for target detection. Firstly, the operational parameters are reduced by deleting the convolution layers and reducing the channels within. Thus, the parameters are reduced by 50% with a permissible precision loss, and the detection speed of the model is significantly improved. Secondly, a light multiple dilated convolution (LMDC) operator is introduced to compensate for the precision loss. The LMDC functions as a filter to extract global and semantic information from the feature map, thereby making feature information complete and more accurate. Moreover, to reduce the computation quantity and increase the computation efficiency of the network, the feature extraction and fusion of the convolution layer are separated. It transforms the complex multiplication into addition among the parameters. Finally, the LMDC-SSD is evaluated on 3 datasets for 300×300 -sized inputs. It yields 98.99% mean average precision (mAP) and 85 frames per second for the apple datasets. The speed and accuracy are improved by 44% and 8.1%, respectively, compared to the original model. The speed and accuracy are improved by 0.99% and 65.71%, respectively, for the bicycle and person datasets. The speed and accuracy are improved by 0.26% and 112.9%, respectively, for the vehicle datasets. The experimental results have shown that the proposed LMDC-SSD is rather promising for detection with high detection speed and accuracy performance.

Keywords Channel reduction · LMDC operator · SSD · Feature extraction · Convolution separation

1 Introduction

In recent years, deep learning-based methods have become the mainstream in target detection. The existing detection algorithms can be divided into two broad categories: the two-stage detection algorithm, represented by the Rich

Feature Hierarchies for Accurate Object Detection and Semantic Segmentation (R-CNN) series [1–3], and the so-called one-stage detection algorithm, such as single-shot multibox detector (SSD) [4] and You Only Look Once: Unified, Real-Time Object Detection (YOLO) [5]. The two-stage model presents a relatively high precision but slow speed because of its preparation and the subsequent detection of the region proposal [6]. Furthermore, due to the feature-extraction ability of the convolution networks, the accuracy of the detection model is improved with an increase in the convolution layers [7]. Typically, fast R-CNN enhances R-CNN using ROI Pooling and softmax for classification, which further improves the detection performance [8]. The mask R-CNN adds a branch for segmentation tasks based on the faster R-CNN, which can enhance precise segmentation while detecting the target. On the contrary, the one-stage detection does not require a region proposal; thus, it achieves faster detection by direct regression. Typically, YOLO divides the input image into $N \times N$ regions responsible for object detection. The detection results are derived

✉ Xinliang Zhang
zxldq@hpu.edu.cn
Heng Xie
708998966@qq.com
Yunji Zhao
auijz@hpu.edu.cn
Wei Qian
qwei@hpu.edu.cn
Xiaozhuo Xu
xxzhpu@163.com

¹ School of Electrical Engineering and Automation, Henan Polytechnic University, 2001 Century Avenue, Jiaozuo, China

directly after the IOU mapping. The SSD achieves end-to-end detection [9, 10] via multi-scale detection. It detects the image at a higher resolution and introduces an anchor mechanism similar to the faster R-CNN to predict the offset value and the confidence of the anchor box. It provides higher accuracy compared to YOLO. However, for practical applications, the SSD consumes much computation time to achieve high precision.

Many researchers resorted to truncating channels to improve the detection speed [11, 12]. The deep compression method compressed deep neural networks through channel pruning and Huffman coding [13]. In Ref. [14], the less weighted channels within convolutional networks were pruned to exploit a linear structure for efficient evaluation. For the lightweight of the SSD model, a Pelee-SSD model was proposed, where a Stem Block was adopted to improve the feature extraction capability of the network [15]. However, the Pelee-SSD model experienced a precision decline compared to the original SSD [4].

In this paper, unlike the previous speed improvement algorithms, we minimize the computation cost by removing the convolution layers and channels within the SSD model, where the single convolution computation is reduced by the convolution separation. At the same time, we introduce a light multiple dilated convolution (LMDC) operator as a filter at the feature extraction layer to improve the detection accuracy. Then, we propose a combined LMDC-SSD model to improve the detection speed without compromising the accuracy.

The rest of this paper is organized as follows. Section 1 summarizes the relevant literature and highlights relevant issues. Section 2 details the construction of the LMDC operator. Section 3 describes the procedure for the parameter reduction and convolution separation. Then, the LMDC-SSD model is presented in Sects. 4, and 5 gives the detection evaluation results on a new apple dataset. Finally, Sect. 6 presents the conclusions and discusses possible future work.

2 Related work and problem statement

The vast majority of target-detection neural networks continuously deepen the network level in an attempt to achieve higher accuracy. A large number of convolution layers have been added to the network, resulting in a huge increase in computation. Typically, the detection performance of a CNN model, such as SSD, is determined by the complex convolution operation. The increase of convolution layers gives a rise of the computation amount, leading to a low detection speed and redundant parameters. A quantity of parameters contributes less to the detection precision, and there exist many channels within convolution layers providing negligibly extra features for the

object description [12]. From a point of improving the computation speed of the network, we delete part of the convolutional layer and channels to reduce directly the computation amount firstly. Secondly, after calculating the number of operational parameters at each layer, the convolutional layers with a large amount of computation are split to further improve the speed of the network. At last, a LMDC operator is adopted to improve the feature extraction ability, so that the network derives a high detection speed and precision.

The computation cost is determined by the number of convolutional layers and the convolution operation of a single convolutional layer, i.e.,

$$N = \sum_{k=1}^Q g(x, y) \quad (1)$$

where Q is the number of convolution layers and $g(x, y)$ the number of operating parameters for a convolution, which is defined as;

$$g(x_i, y_j) = \sum_{i=1}^P \sum_{j=1}^C f(H, W, m, n, s) \quad (2)$$

where C and P are the number of convolution channels and layers, respectively; H and W denote the size of the feature map under the current channel; m and n are the size of the convolution kernel; s is the step length of the convolution. Then, the amount of a single computation under the current channel is governed by

$$f(H, W, m, n, s) = (H - s) \times (W - s) \times m \times n \quad (3)$$

Considering that the convolution operation can be regarded as a process of feature extraction and fusion, there is an amount of repeated calculation that contributes less to performance improvement. Thus, it is expected to have a considerable decrease in the computation complexity if the feature extraction and fusion are separated. Then, the operation of Eq. (2) is alternatively expressed as;

$$g_1(x_i, y_i) = r(x_i, y_i) \quad (4)$$

$$g_2(x_i, y_i) = z(x_i, y_i) \quad (5)$$

where g_1 and g_2 represent the feature extraction and fusion, respectively. They extract features firstly, and then, conduct feature fusion.

For the reinforcement of the feature extraction-convolution operation, a novel LMDC operator is adopted as a filter at the feature extraction layer. Thus, the removal of the redundant convolution channels and layers can directly reduce the convolution parameters with no performance loss at a decreased network depth.

3 LMDC operator and dimension splicing

The performance of the detection neural network model depends on the completeness of the feature extracted from the convolution layers therein. To retain the information of the original feature map, which attenuates with an increase in the convolution layers, an LMDC operator is introduced into the convolution layers. It assumes the residual network structure, as shown in Fig. 1. The input feature map channel is reduced to 1/8 of the original size after the convolution. Subsequently, diverse information is perceived through the specially shaped convolution and the dilated convolution. The information of the feature map is obtained by the dilated convolution with different receptive fields. Then the features are fused by dimension splicing module with those from the original feature image to form a new feature map providing more comprehensive information.

The dilated convolution and the ordinary convolution are compared in Fig. 2. The dilated convolution can obtain various ranges of information without changing the amount of computation [16, 17]. Hence, it improves the ability to acquire global information. The number of receptive fields is determined by the range of feature map on which the dilated convolution kernels operate. For the number limitations of the dilated convolution layers to be less than six, five receptive fields are adopted. Then, the LMDC is constructed by superpositioning parallelly multiple dilated convolution layers. Each receptive field of the dilated convolution operator uses a fraction of the original

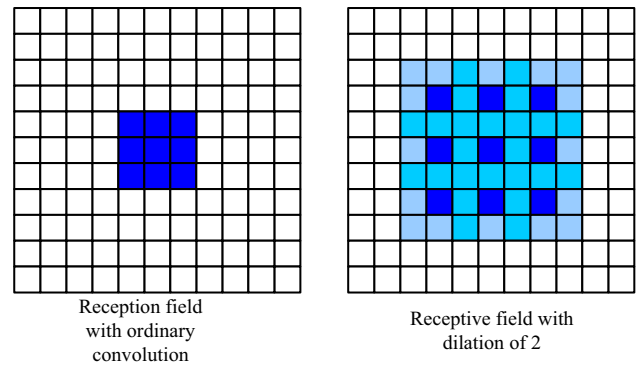


Fig. 2 Receptive field of dilated convolution

feature map channel. After the fusion of five groups of different receptive field features, the information becomes more comprehensive and richer.

LMDC operators reduce the calculation amount by transforming the serial computation into parallel computation during feature extractions. A five-layered parallel feature extraction structure uses 1/8 channel of the original feature map of size $H \times W \times C$. Since the number of channels in the network is a multiple of 8, the five groups of channel parameters are set to 1/8 of the original channel. Therefore, the number of arguments is reduced by 7/8 of the original channel.

Within the LMDC operator, following the feature fusion, the dilated convolution with smaller receptive fields operate directly on the feature map fraction. For the three groups with larger receptive fields, the 3×1 and 1×3 specialized

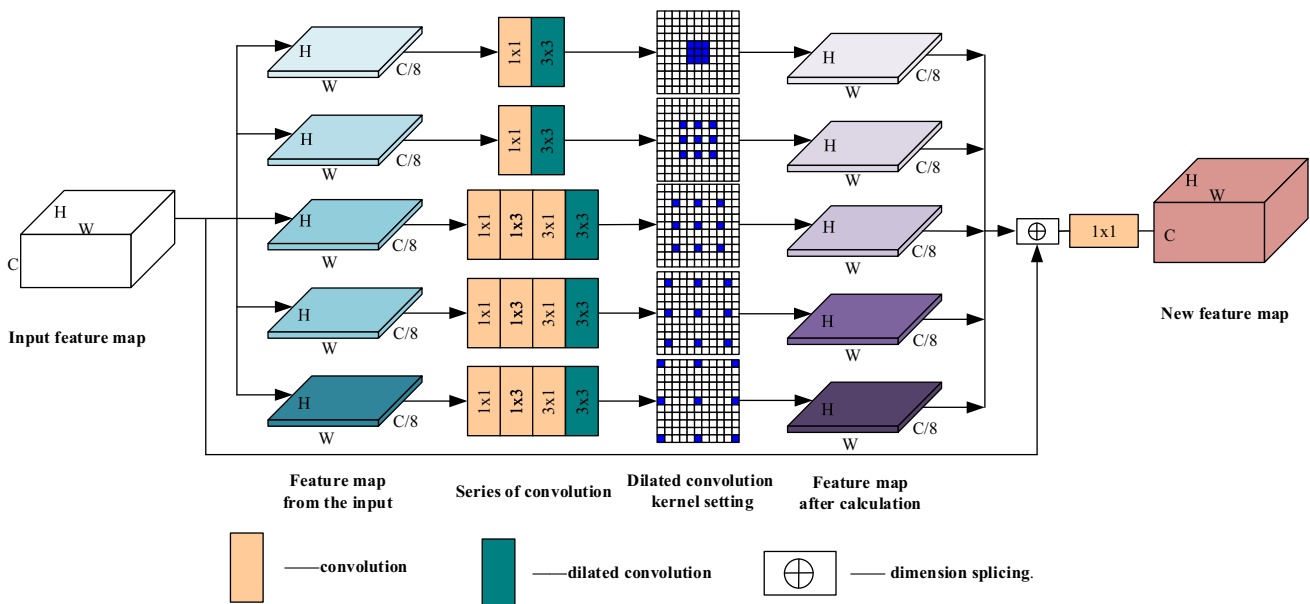


Fig. 1 LMDC operator

convolution kernels are respectively used for operations before the dilated convolution. Besides, an asymmetric convolution pair with horizontal and vertical convolutions is employed to improve the multidirectional features. Figure 3 shows the asymmetric convolution. The ordinary convolution is replaced by two specially shaped convolutions. The feature map A undergoes a 1×3 convolution to generate the feature map B and subsequently a 3×1 convolution to generate the feature map C'. The new feature map C' takes on an identical size to the ordinary convolution.

Finally, five groups of feature maps are spliced and fused to generate a new feature map identical to the size of the original.

4 Parameters reduction and convolution separation

The operation parameters are reduced by deleting the convolution layers and channels therein. Figure 4a and b show the deletion procedure. Based on the original SSD structure depicted in the left column of Fig. 4a, the LMDC-SSD is derived by the channel deletion and feature enhancement, as shown in the right column of the figure. For the i^{th} convolution layer conv- i , it is composed of the ordinary convolution, the transposition convolution and the LMDC

operator, as shown in Fig. 4b. Where $n \times n$ is the size of the convolution kernel and k is the number of the involved convolution channels. After the deletion of the convolutional layers and channels, the LMDC operator and transposition convolution are applied to obtain more detailed information and realize the feature fusion, respectively.

The size of the channels is set to 1024. The 'conv' stage of each layer indicates the convolution operation on the input; convW-V represents the W^{th} convolution layer and its V^{th} stage. The number of the parameters is reduced by the following procedures. The functional layers are slimmed by deleting five layers in the original SSD network, i.e., conv3-3, conv4-3, conv5-3, conv6-2, and conv7-2, and halving the outlet number of conv1 to conv4. Consequently, the channel size of conv6 decreases from 1024 to 256, whereas the number of channels for conv8-1, conv9-1, and conv10-1 increases from 256 to 128. Simultaneously, feature fusion layers are added at the end of conv4, conv6, conv7, and conv8, whereas conv4-3, conv6-2, and conv7-1 are replaced by LMDC operators, respectively. Finally, the transposition convolution is adopted at conv6-3, conv7-3, conv8-3, and conv9-3 to raise the feature map to the same dimension as the upper layer. The size of the ends of conv6, conv7, and conv8 is reduced to half of the dimension of the original feature map.

Then, the number of calculation parameters is expressed as:

Fig. 3 Asymmetric convolution

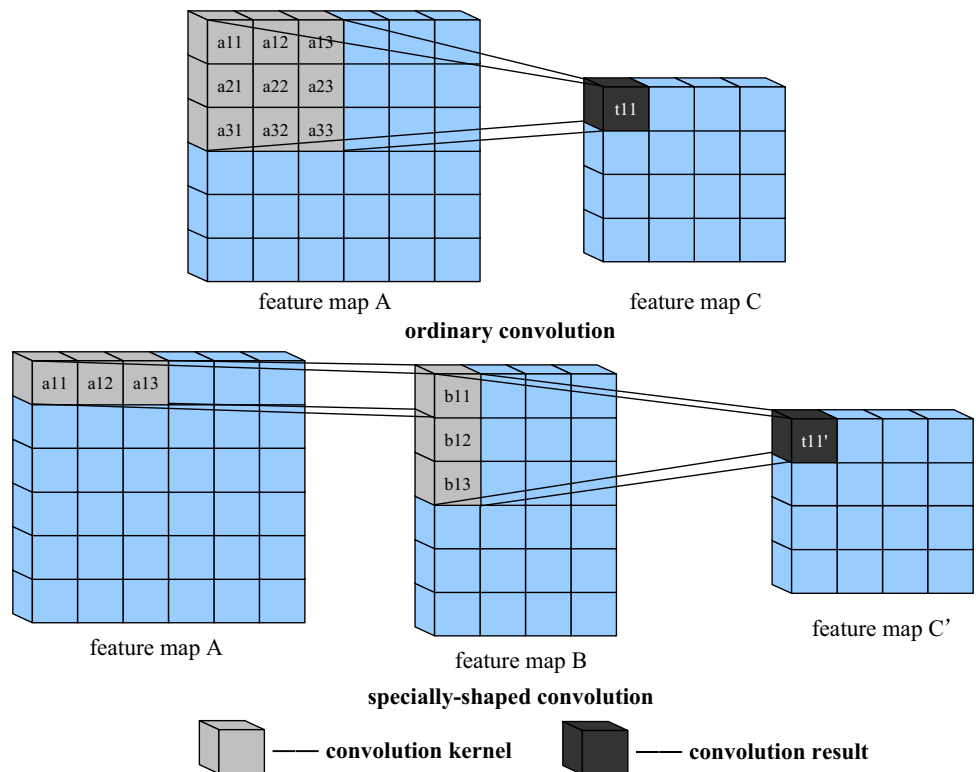
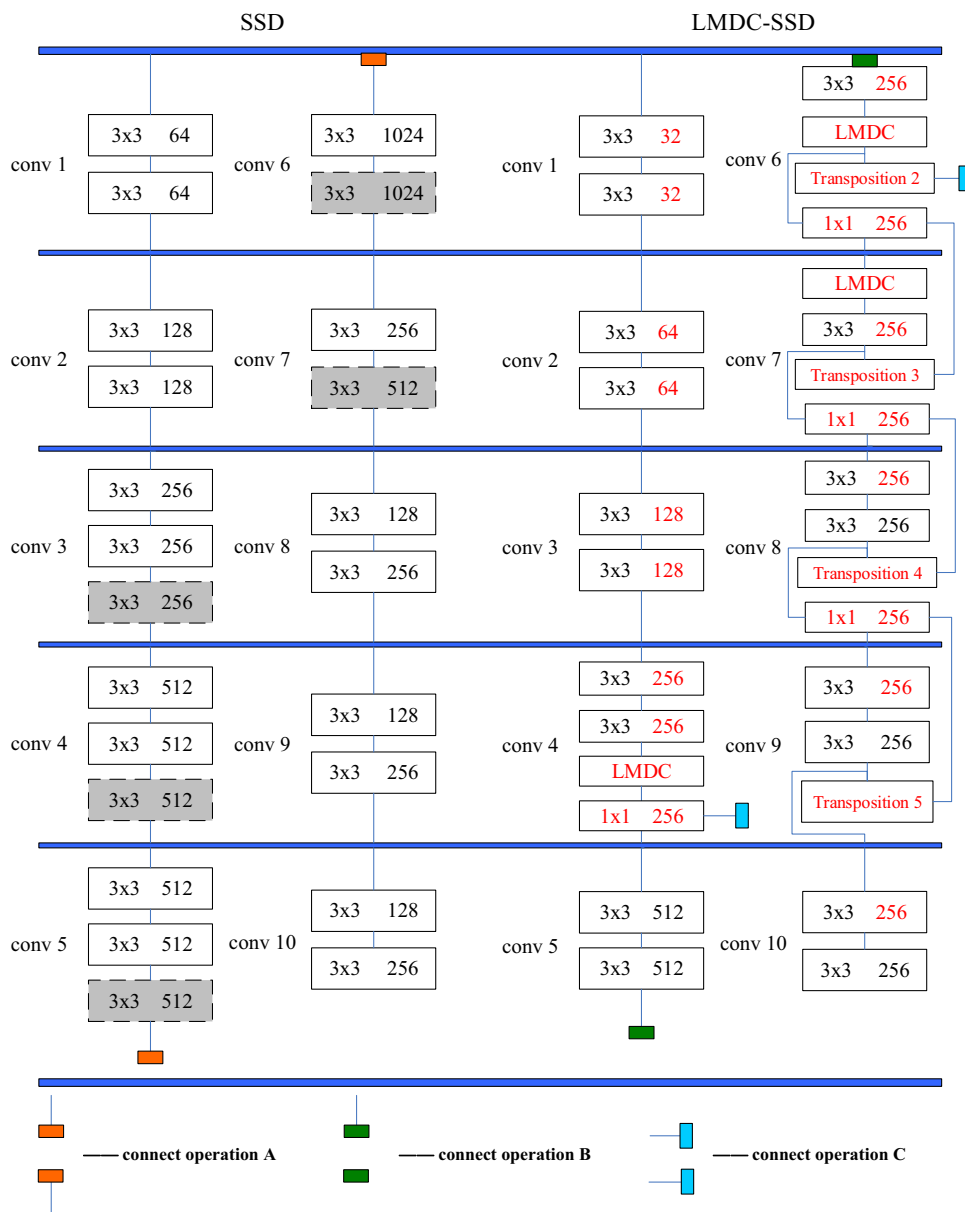
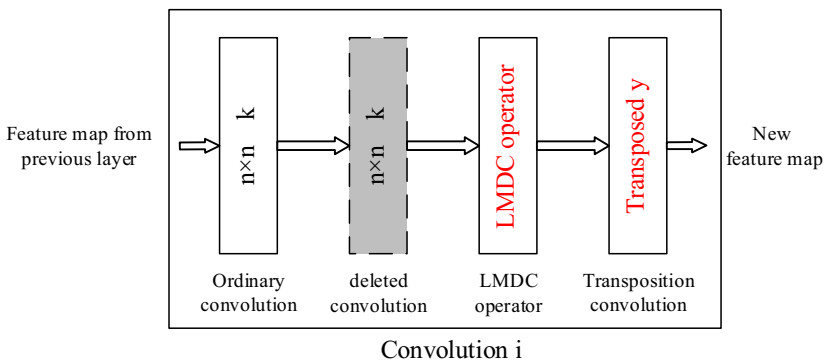


Fig. 4 Channel deletion diagram of LMDC-SSD



(a) Parameter setting



(b) Parameter interpretation

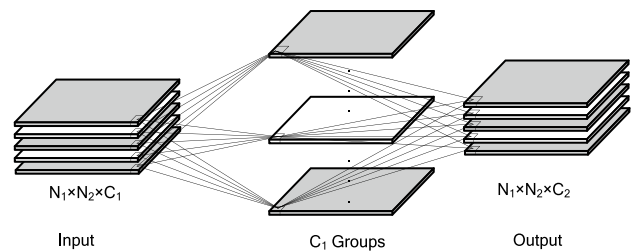
$$g_l = \sum_{i=1}^{P-m} \sum_{j=1}^{C/2} f(x, y) \tag{6}$$

The calculated parameters are reduced compared to the original model expressed by Eq. (1).

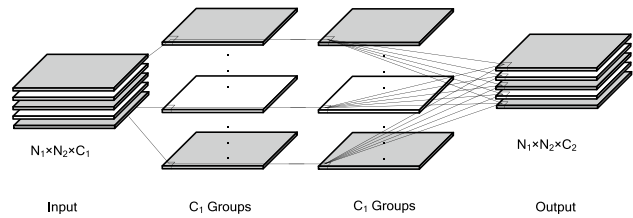
To accelerate the speed, the separation of the feature extraction and fusion in the convolution layer were investigated. The numbers of computational parameters at each layer are listed in Table 1. According to the complexity in the distribution of computations in Table 1, the convolution separation is performed on the convolution layer with a large amount of computation. Figure 5 shows the separation diagram. In the original network model, as shown in Fig. 5a, the feature fusion and extraction are carried out at the same time. However, via the convolution separation, the feature extraction is conducted first for each channel and subsequently the feature fusion is completed, as depicted in Fig. 5b.

The original total number of parameters q is calculated as:

$$g = N_1 \times N_2 \times \Gamma \left[\sum_{i=1}^{C_1} Y(n_1, n_2) \right] \times K \tag{7}$$



(a) Original convolution



(b) Convolution separation

Fig. 5 Schematic diagram of convolution separation

Table 1 Number of calculation parameters in each convolution layer

| Component | Kernel size | Stride | Input channels | Output channels | Feature map size | Calculated amount |
|-----------|-------------|--------|----------------|-----------------|------------------|----------------------|
| conv1-1 | 3×3 | 1 | 3 | 32 | 300 | 77,760,000 |
| conv1-2 | 3×3 | 1 | 32 | 32 | 300 | 829,440,000 |
| conv2-1 | 3×3 | 1 | 32 | 64 | 150 | 414,720,000 |
| conv2-2 | 3×3 | 1 | 64 | 64 | 150 | 829,440,000 |
| conv3-1 | 3×3 | 1 | 64 | 128 | 75 | 414,720,000 |
| conv3-2 | 3×3 | 1 | 128 | 128 | 75 | 829,440,000 |
| conv4-1 | 3×3 | 1 | 128 | 256 | 38 | 425,852,928 |
| conv4-2 | 3×3 | 1 | 256 | 256 | 38 | 851,705,856 |
| conv5-1 | 3×3 | 1 | 256 | 512 | 38 | 1,703,411,712 |
| conv5-2 | 3×3 | 1 | 512 | 512 | 38 | 3,406,823,424 |
| conv6-1 | 3×3 | 1 | 512 | 256 | 19 | 425,852,928 |
| conv6-2 | 3×3 | 1 | 256 | 256 | 19 | 212,926,464 |
| conv7-1 | 3×3 | 1 | 256 | 256 | 10 | 58,982,400 |
| conv7-2 | 3×3 | 2 | 256 | 256 | 10 | 58,982,400 |
| conv8-1 | 3×3 | 1 | 256 | 256 | 5 | 14,745,600 |
| conv8-2 | 3×3 | 2 | 256 | 256 | 5 | 14,745,600 |
| conv9-1 | 3×3 | 1 | 256 | 256 | 3 | 5,308,416 |
| conv9-2 | 3×3 | 1 | 256 | 256 | 3 | 5,308,416 |
| conv10-1 | 3×3 | 1 | 256 | 256 | 1 | 589,824 |
| conv10-2 | 3×3 | 1 | 256 | 256 | 1 | 589,824 |

The calculated amount is the product of the kernel size, the reciprocal of the stride, the output channels number, the input channels number and the square of the feature map size. At the layer of conv1-2, conv2-2, conv3-2, conv4-2, conv5-1 and conv5-2, the calculated amounts in bold are larger than the other's. Thus the convolution separation is used to reduce the number of operational parameters

$$Y(n_1, n_2) = \Gamma \left[\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} (z(i, j) \times n(i, j)) \right] \tag{8}$$

$$q = g, \tag{9}$$

where $n_1 \times n_2$ is the kernel size of the original convolution layer, $N_1 \times N_2 \times C_1$ the number of the input channel, and C_2 the output channel.

When the convolution layer is split, the number of parameters is calculated as:

$$g_1 = N_1 \times N_2 \times \Gamma [Y(n_1, n_2)] \tag{10}$$

$$g_2 = N_1 \times N_2 \times \Gamma \left[\sum_{i=1}^{C_1} Y(1, 1) \right] \times K \tag{11}$$

$$q = g_1 + g_2, \tag{12}$$

where Γ is the convolution operation for the matrix, and Y the single-point convolution of the feature map. This yields a computation relief by transforming from the compound computation to an explicit addition operation.

5 LMDC-SSD model

5.1 Network architecture of LMDC-SSD

Figure 6 shows the LMDC-SSD network. The first five layers are subsampled using pooling layers and other layers using convolution layers. The output of the LMDC operator and the adjacent feature layer are spliced to obtain information

features. A total of 11,620 prior boxes are generated. Finally, target detection is completed by non-maximum suppression. However, only the accurate and single boxes are left, and the redundant invalid boxes are removed.

The LMDC operator is adopted to extract features within the original feature detection layers. Therefore, Conv3 and Conv5 irresponsible for the feature extraction, Conv8 and Conv9 with a smaller size of the feature map than the kernels in the LMDC are excluded from the use of the LMDC. While Conv4, Conv6 and Conv7 are suspended with a

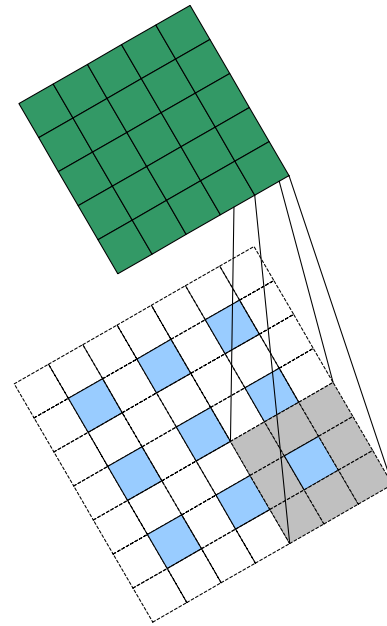


Fig. 7 Transposition convolution

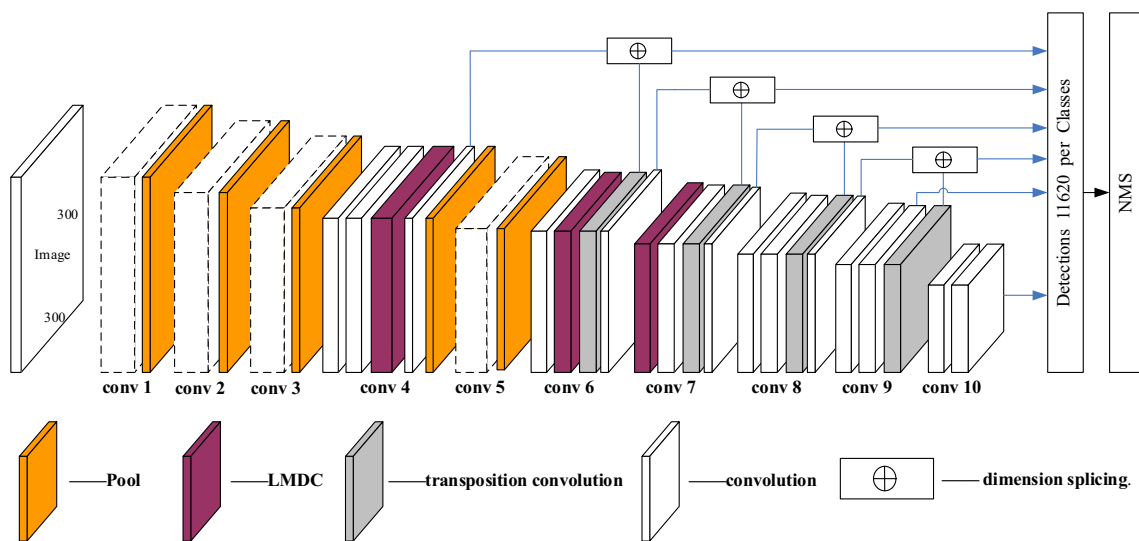


Fig. 6 Network-structure diagram of LMDC-SSD

LMDC operator. The feature map obtained by the LMDC operator is then concatenated with the next layer through the transposition convolution. Figure 7 shows the operation of the transposition convolution. The conversion of a feature map from a low dimension to a high dimension is achieved, as shown in Fig. 7. Thus, the cascaded feature map captures the global information obtained by the LMDC operator and the semantic information of the two-layer feature map.

5.2 Configuration of prior boxes

The prior boxes are set based on the single-point multi-box detection method. According to the size of the feature map, six prior boxes of size [4, 4, 6, 6, 6, 6] were constructed for six layers in the feature map. The scale of the prior boxes is computed as;

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m - 1}(k - 1), k \in [1, m] \tag{13}$$

where s_{\min} and s_{\max} (the scales of the lower and highest layers) are 0.2 and is 0.9, respectively, and all layers therein are regularly spaced. Different aspect ratios are assigned to the default boxes as follows:

$$a_r \in \left\{ 1, 2, 3, \frac{1}{2}, \frac{1}{3} \right\} \tag{14}$$

The width is computed as

$$w_k^a = s_k \sqrt{a_r} \tag{15}$$

and the height as

$$h_k^a = \frac{s_a}{\sqrt{a_r}} \tag{16}$$

for each default box indexed by k .

For an aspect ratio of 1, the default box is scaled by a factor,

$$s'_k = \sqrt{s_k s_{k+1}} \tag{17}$$

For the prior box specified by

$$d = (d^{cx}, d^{cy}, d^w, d^h) \tag{18}$$

its corresponding bounding box is denoted by

$$b = (b^{cx}, b^{cy}, b^w, b^h) \tag{19}$$

With

$$b^{cx} = d^w(\text{variance}[0] * l^{cx}) + d^{cx} \tag{20}$$

$$b^{cy} = d^w(\text{variance}[0] * l^{cy}) + d^{cy} \tag{21}$$

$$b^w = d^w \exp(\text{variance}[2] * l^w) \tag{22}$$

$$b^h = d^h \exp(\text{variance}[3] * l^h) \tag{23}$$

where $l = (l^{cx}, l^{cy}, l^w, l^h)$ is the predicted value of the bounding box. It is a representation of b relative to d and scaled by a hyperparameter *variance*.

During the object detection, the matching degree between the prior box and the real target is determined by the intersection over union [18] as follows:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \in [0, 1] \tag{24}$$

A large value of intersection over union indicates a high correlation. A threshold of 0.5 for the intersection over union is used to classify objects from the background. The prior box with an intersection over union larger than the threshold is assigned as the object labels.

5.3 Loss function formulation

The loss function is governed by the label and coordinate deviation of each prior box to tune the parameters in the network through back propagation algorithm. Specifically, the deviation is calculated by comparing the prediction box coordinates, category confidence and prior boxes in the network operation results with the target results. The target loss of the improved SSD model is defined as the weighted sum of the confidence loss L_{conf} and position loss L_{loc} , i.e.,

$$L(x, c, l, g) = \frac{1}{N} L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g) \tag{25}$$

where α is the weight parameter and N the number of matches.

L_{conf} and L_{loc} are expressed as:

$$L_{\text{conf}}(x, c) = - \sum_{i \in \text{Pos}} x_{ij}^p \log(\hat{c}_i^p) - \sum_{i \in \text{Neg}} \log(\hat{c}_i^o) \tag{26}$$

$$L_{\text{loc}}(x, l, g) = \sum_{i \in \text{Pos}} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k \text{smooth}_{L1}(l_i^m - \hat{g}_i^m), \tag{27}$$

where

$$\hat{g}_j^{cx} = \frac{(g_j^{cx} - d_i^{cx})}{d_i^w} \quad \hat{g}_j^{cy} = \frac{(g_j^{cy} - d_i^{cy})}{d_i^h} \tag{28}$$

$$\hat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right) \quad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right) \tag{29}$$

$$\text{and smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & \text{others} \end{cases} \quad (30)$$

l is the prior box position, c the confidence of the prior frame, $g = (g^{cx}, g^{cy}, g^w, g^h)$ the position parameter of the real box, and $x_{ij}^k \in \{0, 1\}$ the matching degree between the predicted box i and the real box j . The superscript p and o are the probability of x_{ij}^k in the positive and negative samples, respectively.

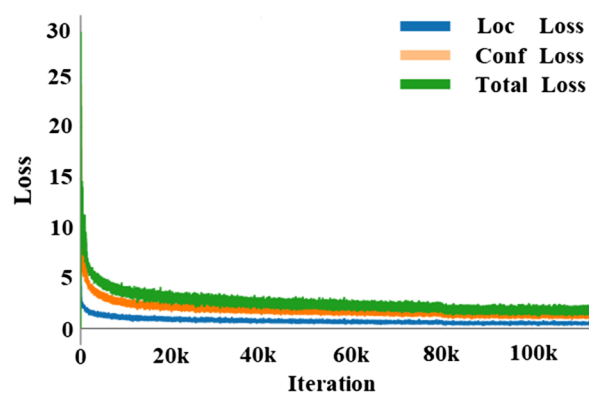
6 Experimental results

The SSD detection model in Ref. [4] has shown a competent superiority in precision and speed performance. The proposed LMDC-SSD model derives from the SSD model by the parameters reduction, LMDC operator and feature cascade. Therefore, the effectiveness of the LMDC-SSD model was validated by the comparison to the SSD model with the same experimental conditions. Experimental verification of the LMDC-SSD was performed on three datasets according to the requirement of PASCAL VOC type. The validation datasets contain the sample information from different environments including apple datasets, bicycle and person datasets and vehicle datasets. Diverse background conditions like overlap, occlusion, strong light, etc., are involved in the samples for the model evaluation. Table 2 present the experimental configurations.

The stochastic gradient descent algorithm was adopted for training the LMDC-SSD model. The input size was set to 300×300 and batch-size to 32. A step-wisely attenuated learning rate was adopted to improve the learning ability of the model. The initial learning rate was 0.05. A damping coefficient was used to attenuate the learning rate of the training process to a smaller value. The damping coefficient was 0.1 for the first 30 and 100 epochs, and the subsequent damping coefficient was chosen as 0.1 per 50 epochs. Besides, a batch normalization layer (BN) [19] was used to improve the training speed and the generalization ability of the network. The BN layer can train the network at a large

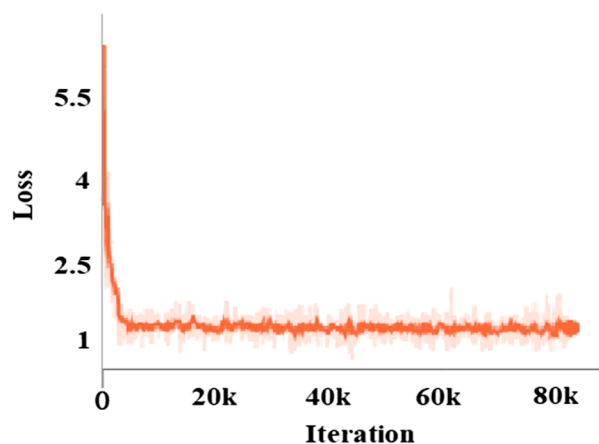
Table 2 Experimental configuration

| Component | Configuration |
|--------------------------|-------------------------------|
| Operating system | Ubuntu 16.04 |
| CPU | I7-8700 K @3.7 GHz |
| Memory | 32 GB |
| GPU | NVIDIA GeForce 1080Ti*2 |
| GPU acceleration library | CUDA 10.1 cuDNN v7.5.0 |
| Deep learning framework | torch 1.2.0 torchvision 0.4.0 |
| Programming language | Python3.5 |



(a) SSD training loss curve

Train/total Loss



(b) LMDC-SSD training loss curve

Fig. 8 Training loss curve

learning rate, accelerate the convergence of the network, and control the overfitting problem.

During the training, the pretraining model was removed, and the training process started from 0. After 80,000 iterations, a stable result was obtained. Figure 8, which was generated by the visualization tool TensorboardX, shows that the LOSS curve of LMDC-SSD took on a similar changing trend as SSD. LMDC-SSD maintained good stability with the introduction of the LMDC module and the reduction of the BN layer. The training loss gradually declined during the first 2500 iterations. After about 10,000 iterations, the loss attained a stable value.

Figure 9a shows the detection of the yellow apples with the object concealed by similar-colored background leaves. Figure 9b and c show the detection results of the red apples, which were partially occluded by the leaves or overlapped each other. Figure 9d shows the detection result of the apples with a simple and uniform background. The validation results showed that the LMDC-SSD provided

Fig. 9 Object detection samples of apple datasets

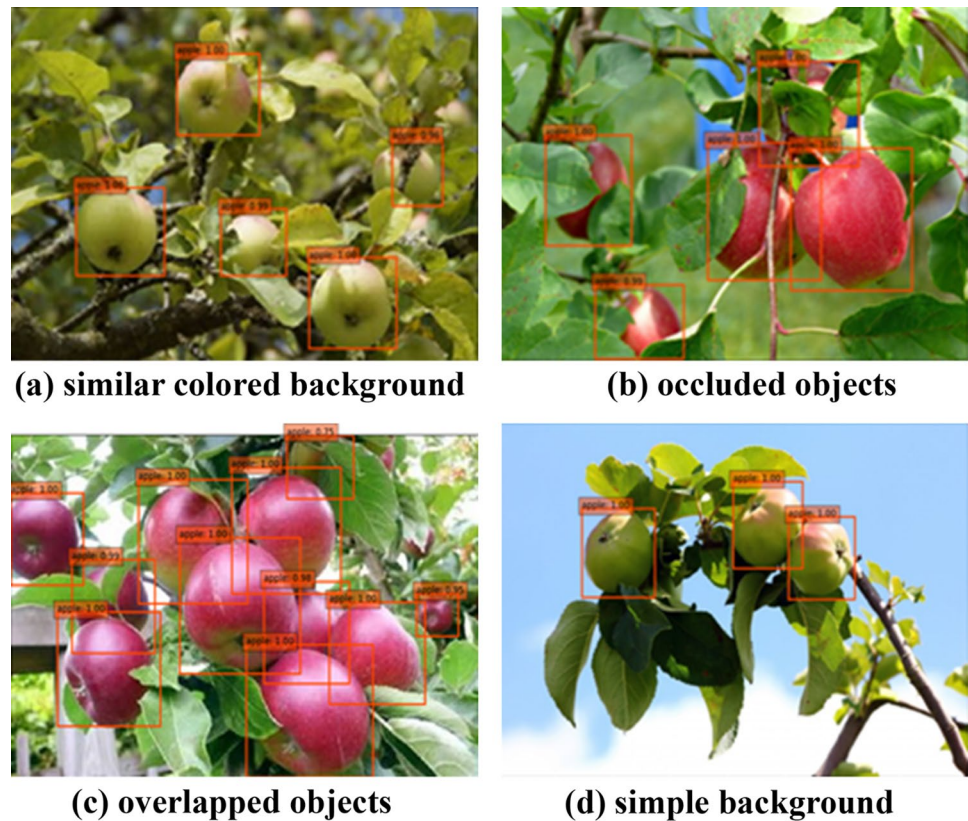


Table 3 Performance comparison between LMDC-SSD and SSD

| Component | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 | Test 6 |
|------------------------|--------|--------|--------|--------|--------|--------|
| Pretraining | – | – | – | – | ✓ | – |
| BN | ✓ | ✓ | ✓ | ✓ | – | – |
| Parameters reduction | ✓ | ✓ | ✓ | – | – | – |
| LMDC | ✓ | ✓ | – | – | – | – |
| Convolution separation | ✓ | – | – | – | – | – |
| mAP | 98.99 | 99.38 | 93.49 | 89.67 | 90.89 | NaN |
| FPS | 85 | 76 | 100 | 57 | 59 | NaN |

excellent robustness in detecting objects against background interference.

Table 3 compares the LMDC-SSD and the original SSD. The average precision (mAP) [20] and the frames per second (FPS) [21] were used as evaluation indexes to demonstrate the detection accuracy and speed of the algorithm, respectively. Except that the SSD ran with its original parameters, other experiments were conducted with a learning rate of 0.05. It means that the models were trained based on a randomly initialized values for the parameters. Test 5 was the result of the original SSD operation, and Test 1 was the operation result of LMDC-SSD. The first four columns show that in the case of the pretraining removal while using BN, the derived SSD model effectively solved the non-convergence problem of the training, which occurred when BN

was not added, as in Test 6. Furthermore, the modified SSD model with merely reduced parameters, as in Test 3, obviously improved the detection speed by 75.4%, and the mAP increased by 4.2% compared to the SSD model in Test 4. Introducing LMDC into the SSD, as in Test 2, improved the mAP by 10.8% and greatly increased the FPS by 33.3%, compared to the SSD model in Test 4. The convolution separation further improved the performance of the LMDC-SSD model. As listed in Test 1, it yielded a rather satisfactory result; mAP was improved by 10.4% and FPS by 49%. The mAP of LMDC-SSD was 98.99%, which is 8.1% higher than that of SSD (90.89%). The FPS of LMDC-SSD was 85, which was much higher than that of SSD (59). Compared with the original SSD model, the FPS was 85, the accuracy

Table 4 A stepwise experiments results in Conv3 to Conv7

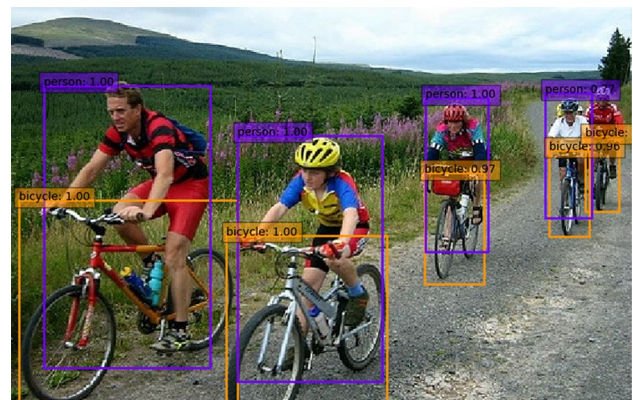
| Component | Test 0 | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 | Test 6 | Test 7 |
|-----------------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Channel reduction | – | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Convolution reduction | – | – | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| LMDC in Conv4 | – | – | – | ✓ | ✓ | ✓ | ✓ | ✓ |
| LMDC in Conv6 | – | – | – | – | ✓ | ✓ | ✓ | ✓ |
| Transposition 2 | – | – | – | – | ✓ | ✓ | ✓ | ✓ |
| LMDC in Conv7 | – | – | – | – | – | ✓ | ✓ | ✓ |
| Transposition 3 | – | – | – | – | – | ✓ | ✓ | ✓ |
| Transposition 4 | – | – | – | – | – | – | ✓ | ✓ |
| Transposition 5 | – | – | – | – | – | – | – | ✓ |
| mAP | 90.89 | 95.33 | 93.49 | 2.07 | 1.5 | 98.79 | 98.87 | 98.99 |
| FPS | 59 | 89 | 100 | 55.8 | 51.7 | 85.8 | 85.5 | 85 |

Table 5 Experimental detection results

| Method | Person and bicycle | | | | Vehicle | | | | All | |
|-------------|--------------------|-----|--------|---------|---------|-----|-------|-------|---------|-----|
| | mAP | FPS | Person | Bicycle | mAP | FPS | Car | BUS | mAP | FPS |
| SSD | 81.055 | 35 | 78.75 | 83.36 | 85.33 | 31 | 85.66 | 85 | 83.1925 | 33 |
| LMDC-SSD | 82.045 | 58 | 79.68 | 84.41 | 85.585 | 66 | 86.11 | 85.06 | 83.815 | 62 |
| Improvement | 0.99 | 23 | 0.93 | 1.05 | 0.255 | 35 | 0.45 | 0.06 | 0.6225 | 29 |

was 8.1% higher than that of SSD (90.89%), and the speed was 44% higher than that of SSD (59).

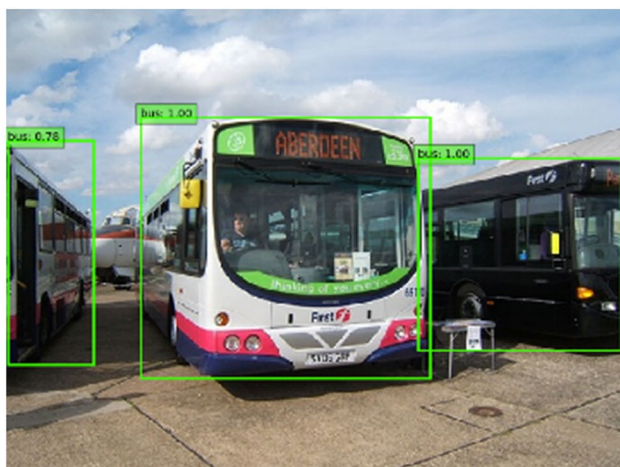
Further, to evaluate the influence of the modifications in conv3 to conv7, a series of stepwise experiments involving the parameters reduction, adoption of the LMDC and the feature cascade operation. The corresponding testing results are shown in Table 4. In Test 0, the original structure and components in SSD remained unchanged. It yielded a 90.89% mAP and 59 FPS on the apple dataset. In Test 1 and Test 2, the detection models took a parameter reduction in channel deletion and convolution reduction, respectively. They showed a better detection performance of 95.33% mAP with Test 1 and 93.49% mAP with Test 2, 89 FPS with Test 1 and 100 FPS with Test 2. A LMDC operator introduced in Conv4 for the Test 3. It leads to an abnormal degradation with the mAP and FPS. The similar results appeared to the Test 4, where Conv 4 adopted a LMDC operator while Conv6 were suspended by the LMDC operator and the successive transposition convolution. Test 4 gave a 1.5% mAP and 51.7 FPS. Based on the model in Test 4, Test 5 employed a LMDC operator in Conv7 and successive transposition convolution. It showed an improvement on the detection precision and speed with 98.79% mAP and 85.8 FPS. Test 6 and Test 7 adopted the transposition convolutions at conv 8, both conv 8 and conv 9, respectively. They provided a gradual improvement with the mAP compared to the Test 5, from 98.79% mAP, 98.87% mAP, to 98.99% mAP, while a negligible decline at the FPS, from 85.8 FPS, 85.5 FPS to 85 FPS. Thus, the LMDC-SSD model as depicted in Fig. 6

**Fig. 10** Detection results on bicycle and person datasets

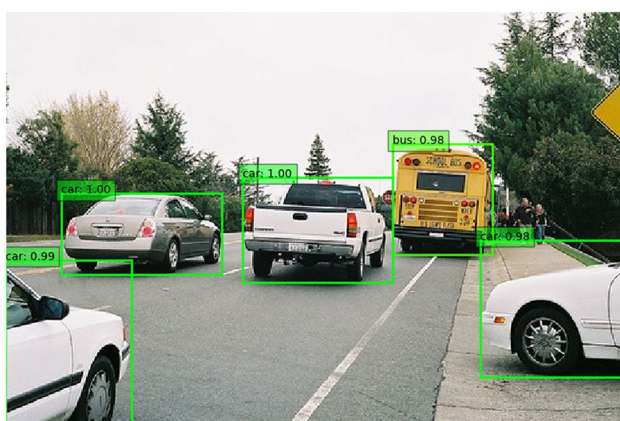
provided a satisfactory detection performance both in precision and speed.

Table 5 presents the detection results of LMDC-SSD on the vehicle, bicycle, and person datasets. The mAP of LMDC-SSD increased by 0.99% in the bicycle and person datasets, whereas the FPS increased by 23. For the vehicle datasets, the mAP increased by 0.26% and the detection speed by 35 FPS. Compared to SSD, LMDC-SSD improved by 0.62% in the average detection accuracy and improved the average detection speed by 29 FPS.

Figure 10 shows the detection results of LMDC-SSD on the bicycle and people's datasets. As we can see, since most bicycles are in the cycling, there is overlap between bicycles and people, and people also have the problem of covering



(a) Vehicles with an incomplete exposure



(b) Vehicles involved with a partly capture

Fig. 11 Detection results on vehicle datasets

with each other. However, LMDC-SSD can successfully detect the target in the case of overlapping and shielding.

Figure 11 shows the detection results of LMDC-SSD for the vehicle datasets. The vehicles have an incomplete exposure because of the overlap in Fig. 11a, while the vehicles in Fig. 11b encounter a partly capture due to the viewpoint of the image acquisition equipment. Thus their appearance is in poor integrity, and some vehicles are even less than 50%. However, LMDC-SSD can overcome this difficulty and successfully identify the vehicles. In addition, some vehicles are in remote locations and overlap in large areas, and LMDC-SSD also can detect vehicles in a certain extent.

The LMDC-SSD proposed herein is different from the network compression or network acceleration schemes. The feature extraction ability was improved from the network itself by compensating for the deficiency caused by network compression. The detection speed was improved without reducing the detection accuracy.

7 Conclusions

This paper presents a fast target detection method by parameter reduction. Convolution layers deletion and channel pruning were realized based on the SSD network model. The LMDC operator was introduced for the feature extraction. The separation between the feature extraction and feature fusion was adopted to improve the detection speed. The experimental results confirmed the effectiveness of the proposed LMDC-SSD model. It greatly improves the speed of network detection, and the LMDC operator effectively compensates for the precision loss caused by parameter reduction.

The application of shallow level features in this paper is still insufficient, and the detection performance in the overlapping part of small targets is weak. Shallow level features contain much detailed information, but the network itself is less applied. Subsequent research will focus on the fusion of the front and rear level features of the network. The scheme of shallow and deep levels fusion could effectively improve the detection performance of the network.

Acknowledgements This research was partially supported by Scientific and Technological Research Projects in Henan Province(212102210244), Foundation of Henan Educational Committee (21A120004), Zhongyuan high level talents special support plan (ZYQR201912031), and the Fundamental Research Funds for the Universities of Henan Province (NSFRF170501).

References

1. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, pp. 580–587 (2014). <https://doi.org/10.1109/CVPR.2014.81>
2. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1137–1149 (2017). <https://doi.org/10.1109/TPAMI.2016.2577031>
3. Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X.: Mask scoring R-CNN. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, pp. 6402–6411 (2019). <https://doi.org/10.1109/CVPR.2019.00657>
4. Liu, W., Anguelov, D., Erhan, D. et al.: SSD: single shot multiBox detector[C]. In: Proceedings of the 14th European Conference on Computer Vision. Springer, Amsterdam, pp. 21–27 (2016). https://doi.org/10.1007/978-3-319-46448-0_2
5. Redmon, J., Divvala, S., Girshick, R., et al.: You only look once: unified, real-time object detection[C]. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer

- Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 770–778 (2016). <https://doi.org/10.1109/CVPR.2016.90>
7. He, K., Gkioxari, G., Dollár, P., et al.: Mask R-CNN[C]. In: International conference on computer vision, pp. 2980–2988 (2017)
 8. Girshick, R.: Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, pp. 1440–1448 (2015). <https://doi.org/10.1109/ICCV.2015.169>
 9. Fu, C., Liu, W., Ranga, A., et al.: DSSD: deconvolutional single shot detector. arXiv: Computer Vision and Pattern Recognition (2017)
 10. Li, Z., Zhou, F.: FSSD: feature fusion single shot multibox detector. arXiv: Computer Vision and Pattern Recognition (2017)
 11. Lane, N. D. et al.: DeepX: A software accelerator for low-power deep learning inference on mobile devices. In International Conference on Information Processing in Sensor Networks (IPSN), pp. 112 (2016)
 12. Liu, G., Wang, C.: A novel multi-scale feature fusion method for region proposal network in fast object detection. *Int J Data Warehousing Min (IJDWM)* **16**(3), 132 (2020)
 13. Han, S., Mao, H., Dally, W.J.: Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding. *Fiber* **56**(4), 37 (2016)
 14. Denton, E., Zaremba, W., Bruna, J., et al.: Exploiting linear structure within convolutional networks for efficient evaluation. arXiv preprint arXiv:1404.0736 (2014)
 15. Wang, R. J., Li, X., Ling, C. X.: Pelee: a real-time object detection system on mobile devices. arXiv preprint arXiv:1804.06882 (2018)
 16. Schuster, R., Wasenmüller, O., Unger, C., et al.: SDC—stacked dilated convolution: a unified descriptor network for dense matching tasks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2020)
 17. Wang, Z., Ji, S.: Smoothed dilated convolutions for improved dense prediction. In: arXiv: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2486–2495 (2018)
 18. Zhang, G. L., Ge, L. L., Yang, Y. N., et al.: Fused confidence for scene text detection via intersection-over-union. In: 2019 IEEE 19th International Conference on Communication Technology (ICCT). IEEE (2019)
 19. Santurkar, S., Tsipras, D., Ilyas, A., et al.: How does batch normalization help optimization? arXiv preprint arXiv:1805.11604 (2018)
 20. Revaud, J., Almazan, J., Rezende, R., Souza, C. D.: Learning with average precision: training image retrieval with a listwise loss. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
 21. Long, X., Hu, S., Hu, Y., et al.: An FPGA-based ultra-high-speed object detection algorithm with multi-frame information fusion. *Sensors* **19**(17), 3707 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Xinliang Zhang received his B.E degree in electrical engineering in 2000 at the Department of Automation, Shandong University, Shandong, China, M.E. degree in control theory and applications at Department of Automation in 2004, Nanjing University of Aeronautics and Astronautics, Nanjing, China, and PhD degree in control theory and application in 2009, Shanghai Jiaotong University, Shanghai, China. He is currently a deputy professor at school of Electric Engineering and Automation, Henan Polytechnic University. His research interests include measurement technology and automatic instrument, signal processing, intelligent modeling and control of non-smooth nonlinear systems.



Heng Xie received his B.E degree in electrical engineering in 2017 at the Department of Automation, Henan Institute of Science and Technology, Luoyang, China. He is currently pursuing his M.E. degree in control engineering at Department of Automation, Henan Polytechnic University, Jiaozuo, China. His research interests include measurement technology and automatic instrument, image processing, deep learning and pattern detection.



Yunji Zhao received the Ph.D degree from the School of Automation Science and Engineering, South China University of Technology, Guangzhou, China, in 2012. He is currently a Lecture with the School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo, China. His current research interests include pattern recognition and artificial intelligence.



Wei Qian received the Ph.D. degree from the State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, China, in 2009. He is currently a Professor with the School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo, China. His current research interests include time-delay systems, stochastic systems, networked control systems, and multiagent system.



Xiaozhuo Xu received the Ph.D. degree from Henan Polytechnic University in 2016. He joined the School of Electrical Engineering and Automation, at Henan Polytechnic University in 2006 as an assistant professor. His research interests include electrical machine design, intelligent control, and fault diagnosis.