A Block Object Detection Method Based on Feature Fusion Networks for Autonomous Vehicles

Qiao Meng,1,2 Huansheng Song,1 Gang Li,1 Yu’an Zhang,2 and Xiangqing Zhang1

1School of Information Engineering, Chang'an University, Xi'an, Shaanxi 710064, China
2Computer Technology and Application Department, Qinghai University, Xining 810016, China

Correspondence should be addressed to Qiao Meng; 250345481@qq.com

Received 24 November 2018; Accepted 21 January 2019; Published 6 February 2019

Guest Editor: Ke Deng

Copyright © 2019 Qiao Meng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Nowadays, automatic multi-objective detection remains a challenging problem for autonomous vehicle technologies. In the past decades, deep learning has been demonstrated successful for multi-objective detection, such as the Single Shot Multibox Detector (SSD) model. The current trend is to train the deep Convolutional Neural Networks (CNNs) with online autonomous vehicle datasets. However, network performance usually degrades when small objects are detected. Moreover, the existing autonomous vehicle datasets could not meet the need for domestic traffic environment. To improve the detection performance of small objects and ensure the validity of the dataset, we propose a new method. Specifically, the original images are divided into blocks as input to a VGG-16 network which add the feature map fusion after CNNs. Moreover, the image pyramid is built to project all the blocks detection results at the original object size as much as possible. In addition to improving the detection method, a new autonomous driving vehicle dataset is created, in which the object categories and labelling criteria are defined, and a data augmentation method is proposed. The experimental results on the new datasets show that the performance of the proposed method is greatly improved, especially for small objects detection in large image. Moreover, the proposed method is adaptive to complex climatic conditions and contributes a lot for autonomous vehicle perception and planning.

1. Introduction

Environment perception is an important part of the autonomous driving system, and the sensors used for sensing include ultrasonic radar, millimetre wave radar, LiDAR (Light Detection and Ranging), and cameras. Through the fusion of LiDAR, millimetre wave radar, and cameras, objects can be detected, and object space ranging and recognition can be realized. Specially, the fusion of cameras and LiDAR not only can realize high precision positioning of objects but also can realize the detection of multiple types of objects. However, due to the high cost of LiDAR, this sensor fusion cannot become a popular method in the future. In contrast, low-cost cameras will be applied in the autonomous vehicle perception system, such as object detection and classification.

Currently, there are two object detection algorithms, namely traditional image processing and deep learning classification. Through the analysis and processing of images, both methods can return the location and classification information of objects and provide effective information for the planning and decision-making system. However, due to the extremely rich information of images and the difficulty in manual modelling, the accuracy of the traditional image processing method is worse than the deep learning method. Therefore, more and more camera-based deep learning algorithms can make the perception of autonomous vehicles much more accurate, fast and comprehensive. At present, the existing deep learning systems can be divided into two categories. One is the region proposal method, such as R-CNN [1], Fast R-CNN [2] and Faster R-CNN [3]. The other is the proposal-free method, such as You Only Look Once (YOLO) [4] and Single Shot Multi-box Detector (SSD) [5]. In recent years, SSD model has obvious advantages for video object detection in terms of detection speed and accuracy. However, some problems still exist in the SSD model. The first problem is the dataset. A rich dataset is crucial for object detection. At present, the current datasets for autonomous driving are based on foreign traffic scenarios, such as KITTI, Cityscapes, etc. The second problem lies in the classification accuracy. The detection accuracy of SSD model is lower.
than Faster R-CNN. Specially, as the network deepens, small objects are gradually lost during the convolution process of the SSD model.

In this study, we propose a novel approach to detect the object for autonomous driving. Our contribution of this paper includes four points. Firstly, we divide the original image into blocks (the block size is 400×400), which can detect small objects from the image, and then we resize each block to a fixed size (512×512) for training. Secondly, the original image is down-sampled in multiples of 1/2 times until the image size is close to the block size, which ensures that large objects can be completely covered in a single image block. Thirdly, as the feature map of the SSD model gradually shrinks, the characteristic information of the small objects disappears or becomes inconspicuous. Therefore, a feature fusion method is added in the SSD model to ensure the detection precision of small objects in the large image. Fourthly, the samples are collected and labelled by ourselves, where we have designed object categories and the annotation method.

This paper will be described as follows: Section 2 introduces our method in detail. Section 3 describes the experiment of our method. Section 4 shows the experimental results and analyzes the experimental results. Section 5 provides a discussion and the future work for this paper.

## 2. Methodology

In this section, we described the details of the improved SSD model using feature fusion and image block segmentation methods, and introduced the method for creating an autonomous driving dataset.

### 2.1. Feature Fusion Network

The SSD model directly extracts different scales from different feature map layers of the CNNs, as shown in Figure 1(a). This approach cannot fuse feature maps of different scales, so the feature maps of different scales are independent of each other. Therefore, based on the SSD model, we propose a new image feature fusion algorithm, which requires multiple feature merging processes and usually consumes so much time.

As shown in Figure 1(b), after VGG16 network, seven convolution layers were added to extract features, including conv6_1, conv6_2, conv7_2, conv8_2, conv9_2, and conv11_2. The feature map sizes of these seven convolution layers are 32×32, 32×32, 16×16, 8×8, 4×4, 2×2, 1×1. Through analysis, when the feature map size is less than 16×16, objects continue to shrink, and its characteristics gradually disappear, so feature fusion cannot be implemented. In this paper, the bilinear function is used to fuse feature maps of different sizes. The feature map is upsampled starting from conv7_2 and its size is not less than 16×16. Since the bilinear function converts one pixel to four pixels, that is, the current image is doubled, therefore, conv7_2, conv6_2, and conv4_3 satisfying the 2-fold relationship are selected for upsampling. Before the bilinear operation, a conv 1×1 operation is performed on the image, which can reduce the dimension of the feature and accelerate the computation [6]. The specific calculation method is as follows:

$$Y_i = \text{Bilinear}(y_i)$$  \hspace{1cm} (1)

where $y_i$ means a feature map that needs to be merged, \text{Bilinear} is linear interpolation function, and $Y_i$ is a feature map that is magnified double. The calculation of this equation, the fused feature maps are adjusted to the same size. Moreover, the feature maps of the same size are fused using the element-wise-max method, which preserves the maximum value of the corresponding position pixel values in the two feature maps. The specific calculation method is as follows:

$$\text{element-wise-max} (Y_i, Y_j) = T\left(\text{max} (X_{i,j}, Y_{i,j})\right)$$  \hspace{1cm} (2)

where $Y_i, Y_j$ represent feature maps with the same size, $T$ represents a new feature map matrix generated after fusion, and $X_{i,j}, Y_{i,j}$ respectively, represent the pixel values of the merged feature map matrix. The feature fusion of the element-wise-max function, the pixel layer is generated, and the pixel layer is continuously sampled to generate a pyramid feature map, this is the feature fusion method of this paper. Table 1 compares the mean average precision (mAP) and the frame rate (fps) for different SSD models.

### 2.2. Dataset

For deep learning, the importance of datasets is unquestionable. A rich dataset is crucial for object detection. Currently, datasets in the field of computer vision include ImageNet [7], COCO [8] and PASCAL VOC [9], etc. Moreover, the dataset for autonomous driving primarily uses the KITTI [10] dataset or the Cityscapes [11] dataset. Table 2 compares the two datasets.

Since the KITTI or Cityscapes dataset for autonomous driving satisfies the requirements of computer visual, the perspective and categories of samples do not match well with domestic demand. Therefore, it is necessary to establish a new dataset which not only satisfy the perspective requirements but also match the current domestic traffic environment well.
Table 2: Comparison of different datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Categories</th>
<th>Function</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>KITTI</td>
<td>Car, Van, Truck, Pedestrian, cyclist, Tram</td>
<td>Evaluation of computer vision algorithm in automatic driving scene</td>
<td>2012</td>
</tr>
</tbody>
</table>

![Diagram (a)](image_a)

**Figure 1:** Our SSD512 framework. The part of the red box labelled in figure (a) is the SSD framework proposed in [5], and the part of the red box labelled in figure (b) is our SSD512 framework. Figure (b) uses a pyramid feature fusion method that fuses feature maps with a 2-fold relationship in a recursive manner.

2.2.1. Sample Collection

(1) Sample Collection Platform. In order to create a new dataset that meets our requirements, we apply a data acquisition platform for autonomous vehicles to collect samples from real traffic environments. The data acquisition platform is equipped with a high-dynamic camera (acA1920), a velodyne 64-E LiDAR, and a differential GNSS receiver (Simpeak 982). The dataset we have collected contains real-world image data from urban, rural, freeway scenarios, etc. Moreover, each image contains at least one vehicle or one pedestrian. The entire system sampled and synchronized at 10Hz frequency. In the image acquisition system, the cameras shooting distance of our autonomous vehicle is 13 meters, and the resolution of the captured image is 1920×1200. Moreover, referring to the current status of China’s traffic roads, we classify the labels into seven categories, including car, truck, bus, minibus, cyclist, person, and motorcycle.

(2) Sample Augmentation. Traffic scenes captured by image acquisition platforms include urban roads, highways, tunnels,
and curved roads. Moreover, some samples were collected under complex climatic conditions, such as rainy days and foggy days. Since deep learning requires training a large number of samples to learn object characteristics well, we use sample augmentation method to expand the dataset. As shown in Figure 2, the specific process is as follows.

**Step 1.** Randomly select an image from all the pre-trained images.

**Step 2.** For each image, one of the small blocks is randomly sampled, the aspect ratio of the small block is set to $[1/2, 2]$, and the overlap ratio with the object is 0.1, 0.3, 0.5, 0.7, and 0.9.

**Step 3.** If the central point of the bounding box is in the sampled block, the overlapped portion is retained.

**Step 4.** This article uses a fixed size of 512×512. To resize each sample to the fixed size, and then shifts or rotates the fixed block at a random level with a probability of 0.5.

2.2.2. Annotation. The diversity of samples is crucial to ensure the accuracy of detection [12]. Therefore, the selection of samples should consider multiple angles, complex climatic conditions, occlusion ratio and truncation ratio. The principles we propose for image annotation are as follows.

1) **Select Samples from Multiple Angles.** There are slight differences in the characteristics of the samples from different angles [13]. Therefore, angles are crucial for the image. We label samples from the positive angle, the reverse angle, and the side angle to ensure the comprehensiveness of samples. Table 3 shows the statistics for the number of seven types of samples from different angles.

2) **Choose Complex Climate Conditions.** In severe weather conditions, the model is greatly affected by visibility. After analyzing the acquisition height and clarity of the cameras, we divided the bad weather into fog, rain, sunny, and cloudy. Moreover, we label samples in different climate conditions to ensure the adaptability of sample characteristics to the environment.

3) **Set the Occlusion Ratio.** The human eye can easily follow a specific object for a period time [14]. However, for the machine, this task is not simple. Generally, there are various complicated situations in the object tracking process, such as the occlusion ratio is an important problem. After investigation, we define a score for an occluded bounding box. The specific definition method is divided into three cases: heavy occlusion, partial occlusion and no occlusion. In this article, if the occlusion ratio of a vehicle is larger than 40%, we define it as the heavy occlusion. Similarly, when the occlusion ratio of a vehicle is between 1% and 40%, we think it is partial occlusion. In order to ensure the accuracy of the detection, we stipulate that only partial occlusion and no occlusion are labelled, and heavy occlusion is not labelled.

4) **Set Truncation Ratio.** Not all objects in the image are labelled. According to the requirements of network training, the truncation ratio is set to 1/3. In other words, if the area of the object beyond the image boundary is greater than 1/3 of the object area, then we do not label this object.

According to the defined labelling principle, the specific labelling process includes three steps. The first step is to determine the storage location of the labelled file and the storage location of the newly created dataset, and draw a bounding box for each object in the sample. The second step is to assign a label category to each object's bounding box. The third step is to determine whether the object in the sample is occluded or truncated, if it exists, a description field needs to be added for the object label.
<table>
<thead>
<tr>
<th></th>
<th>fog</th>
<th>rain</th>
<th>sunny</th>
<th>cloudy</th>
<th>fog</th>
<th>rain</th>
<th>sunny</th>
<th>cloudy</th>
<th>fog</th>
<th>rain</th>
<th>sunny</th>
<th>cloudy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>187</td>
<td>94</td>
<td>385</td>
<td>201</td>
<td>109</td>
<td>81</td>
<td>175</td>
<td>184</td>
<td>176</td>
<td>79</td>
<td>293</td>
<td>183</td>
</tr>
<tr>
<td>Cyclist</td>
<td>133</td>
<td>35</td>
<td>257</td>
<td>123</td>
<td>36</td>
<td>47</td>
<td>214</td>
<td>169</td>
<td>73</td>
<td>41</td>
<td>121</td>
<td>144</td>
</tr>
<tr>
<td>Motorcycle</td>
<td>89</td>
<td>47</td>
<td>101</td>
<td>82</td>
<td>59</td>
<td>42</td>
<td>97</td>
<td>59</td>
<td>33</td>
<td>29</td>
<td>61</td>
<td>64</td>
</tr>
<tr>
<td>Car</td>
<td>198</td>
<td>241</td>
<td>278</td>
<td>261</td>
<td>142</td>
<td>169</td>
<td>205</td>
<td>193</td>
<td>192</td>
<td>105</td>
<td>217</td>
<td>254</td>
</tr>
<tr>
<td>Bus(MiniBus)</td>
<td>106</td>
<td>166</td>
<td>183</td>
<td>201</td>
<td>137</td>
<td>121</td>
<td>156</td>
<td>75</td>
<td>143</td>
<td>152</td>
<td>224</td>
<td>104</td>
</tr>
<tr>
<td>Bus</td>
<td>99</td>
<td>102</td>
<td>129</td>
<td>79</td>
<td>101</td>
<td>66</td>
<td>98</td>
<td>106</td>
<td>136</td>
<td>97</td>
<td>165</td>
<td>144</td>
</tr>
<tr>
<td>Truck</td>
<td>129</td>
<td>89</td>
<td>153</td>
<td>128</td>
<td>96</td>
<td>108</td>
<td>124</td>
<td>152</td>
<td>185</td>
<td>123</td>
<td>148</td>
<td>267</td>
</tr>
</tbody>
</table>
In this work, a total of 11550 images are labelled, including 10394 training sets and 1156 testing sets. Moreover, the new dataset is named SSMCAR. Figure 3 is a snapshot of the SSMCAR dataset annotation.

2.3. Image Block Architecture. From the limited available memory of current GPUs, it is not feasible for deep convolutional networks to accept large images as input, especially for image sizes larger than 2000×2000 [14]. In the SSD detection model, it resized the entire image to a fixed size. As shown in Figure 1(a), it resizes the image to 512×512. The disadvantage of this approach is that it directly resizes the image to a fixed size, which not only reduces the resolution of the image itself but also affects the learning effectiveness of the object characteristics, especially for large images. Therefore, an object detection method based on image blocks is proposed. As shown in Figure 1(b), the input image is divided into blocks according to a certain strategy [15], and then each block is trained according to our SSD method.

In our SSD512 framework, in order not to change the quality of the image itself to the greatest extent, we propose a strategy to divide the original image into blocks with different sizes. Since the SSD model needs to resize the image to a fixed size in advance, such as resizing the image size to 300×300 or 512×512. At the same time, the characteristics of the convolutional neural network show that the minimal adjustment of the original image has a small influence on the final detection results. Therefore, we use the enumeration method to divide the original image into blocks around the size of 300×300 or 512×512, and then choose the best block scheme. After the image block is completed, each block will be resized to 300×300 or 512×512 before they are input into our SSD model. This approach has two advantages. On the one hand, it reduces the loss of small objects in the process of network learning, on the other hand, it reduces the problem of image quality degradation in the SSD method. Admittedly, the larger the size of the image, the better the detection result of this method. In this paper, the main purpose of our research is multi-object detection for automatic driving. Therefore, we use our own dataset as an example to illustrate the specific details of the block strategy.

In this paper, the size of the sample is 1920×1200. We assume that the fixed size of the input network is 512×512, and use the enumeration method to select different sizes close
to 512×512 to segment the original image. The calculation formula is shown as follows:

\[(x, y) = ((300, 300), (400, 400), (500, 500), (600, 600))\]  
\[x, y \in [300, 700)\]  

where \(x, y\) represent the horizontal and vertical block sizes. As shown in Figure 4, using the top-to-bottom and the left-to-right method divides the image along the horizontal and vertical directions.

Since the network model we used in Figure 1(b) is a 512×512 model, we need to resize the block to the size of 512×512 before sending it to the network. We define two criteria to choose the best block scheme. One is that the size of the block is the closest to 512×512; the other is that the difference between blocks and blocks is the smallest and the aspect ratio of each block is the largest. In Figure 4(b), the 400×400 block strategy yields two different sizes of 400×400 and 320×400 blocks, including 12 blocks of 400×400 and 3 blocks of 320×400, as depicted in (b). Moreover, the block strategies of 500×500 and 600×600 are the same as those of Figures (a) and (b), as shown in (c) and (d).

To 512×512 images. The same color represents blocks of the same size, and different colors represent blocks of different sizes. The 300×300 block strategy yields two different sizes of 300×300 and 120×300 blocks, including 24 blocks of 300×300 and 4 blocks of 120×300, as depicted in (a). Similarly, the 400×400 block strategy yields two different sizes of 400×400 and 320×400 blocks, including 12 blocks of 400×400 and 3 blocks of 320×400, as depicted in (b). Moreover, the block strategies of 500×500 and 600×600 are the same as those of Figures (a) and (b), as shown in (c) and (d).

### 3. Experiment

#### 3.1. Training of Our SSD Model

Our SSD model has a large training parameter. If we train all the characteristics of the network from scratch, it is not only time-consuming but also prone to data overfitting and gradient non-convergence [15]. In this paper, the transfer learning [16] method is applied. Based on the pre-trained model, the training accuracy and loss function of the model are compared using different datasets and different network.

During the training process, our SSD model takes all the anchors in each block in a graph as the window to determine whether there is an object in the window. If there is an object, it predicts the category and position information of the object, otherwise, it defines the anchor as the background. As shown in Figure 5, our SSD model cuts each block of the graph into 8732 anchors of different sizes, each of which has

![Figure 4: Different blocking strategies for 1920×1200 images. The same color represents blocks of the same size, and different colors represent blocks of different sizes. The 300×300 block strategy yields two different sizes of 300×300 and 120×300 blocks, including 24 blocks of 300×300 and 4 blocks of 120×300, as depicted in (a). Similarly, the 400×400 block strategy yields two different sizes of 400×400 and 320×400 blocks, including 12 blocks of 400×400 and 3 blocks of 320×400, as depicted in (b). Moreover, the block strategies of 500×500 and 600×600 are the same as those of Figures (a) and (b), as shown in (c) and (d).](image-url)
3.2. Loss Function of Our SSD Model. The loss function is applied to evaluate the network performance of our SSD model [18]. The loss function is divided into the localization loss (loc) and classification loss (conf) [19, 20], which is defined as follows:

\[ L(x, c, l, g) = \frac{1}{N} \left( L_{conf}(x, c) + \alpha L_{loc}(x, l, g) \right) \]  

(4)

where \( x \) is a matched default box; \( N \) is all matched default boxes, if \( N = 0 \), then \( Loss = 0 \). \( L_{conf} \) is the softmax loss over object classes which is actually the loss of confidence [5]; \( L_{loc} \) is the SmoothL1 loss [5] based on the predicted box; \( \alpha \) is set to 1 by cross validation.

### 3.3. Test Our SSD Model

#### 3.3.1. Image Pyramid Architecture. As our SSD model is designed to be sensitive to small objects, some large objects are divided into different blocks, which can cause the loss of features of large objects at the original resolution. An image pyramid is created to solve this problem. Specifically, an image pyramid rule for constructing image pyramid is proposed, in which the size of low-resolution image are 0.5 times than the size of high-resolution image. Moreover, since our network model is 512 × 512, see Figure 1(b), if the image size is less than 512 × 512 too much, it will be detrimental to the learning of object characteristics. As shown in Figure 6,
the resolution of our dataset image is 1920×1200, and the image size of the third layer of the image pyramid is 480×300. According to our pyramid construction method, the image size of the fourth layer is 240×150 which is no value in learning object features, so a three-layer image pyramid structure is constructed.

3.3.2. Non-Maximum Suppression Method. We noticed that our SSD model predicts a result for each layer of images. As shown in Figure 6, we can see that there are three bounding boxes in the same location in our image pyramid architecture, which can generate non-uniqueness of detection. Therefore, the NMS [21, 22] algorithm can be used to eliminate redundant (cross-repeat) windows and find the best object detection location.

4. Results and Analysis

There are three parameters for evaluating object detection performance, including accuracy, loss, and detection rate [23–26]. Specifically, the parameter mAP is the average classification accuracy of the seven objects, and its value is between 0 and 1. The larger the value of mAP, the higher the classification accuracy. In addition, the frame rate (fps) is used to evaluate the detection speed.

4.1. Accuracy. In this paper, we, used SSD model and our SSD model to train and test VOC dataset, KITTI dataset, and our dataset (SSMCAR). In this process, the single NVIDIA 1080Ti GPU server is applied, the initial learning rate is set to 0.01, and the number of iterations is set to 100,000 and 120,000 times.

As shown in the left of Figure 7, the accuracy of the SSMCAR dataset is much higher than the VOC2007 dataset or KITTI dataset. Since the resolutions of the VOC2007 dataset and KITTI dataset are 500×375 and 1242×375, correspondingly, the resolution of the SSMCAR dataset is 1920×1200, the image quality of the SSMCAR dataset is higher than that of the VOC2007 dataset and the KITTI dataset. In addition, the collection perspective and the labelling principles of SSMCAR datasets are different from VOC2007 dataset and KITTI dataset. After further experiments, it shows that increasing the number of samples has no effects on improving the accuracy. Moreover, from Figures 7(a) and 7(b), it can be seen that the optimal solution occurs when the number of iterations of VOC dataset and KITTI dataset reaches 100,000 times and the highest accuracy is obtained. As shown in Figure 7(c), the optimal solution for the SSMCAR dataset we created appeared between 100,000 and 120,000 iterations.

As shown in the right of Figure 7, the accuracy of our SSD model is higher than the SSD model. Unlike the SSD model directly resize images to 512×512, our SSD model use the image block, image pyramid, and feature fusion method to protect the feature of the original images.

4.2. Loss. After each complete training process, a suitable learning rate can guarantee that the loss is reduced to a small value after a period of time [27, 28]. Too small learning rate often makes the loss reduction very slow. Conversely, if the learning rate is set too large, the initial loss can be reduced very fast, and then it repeats at a certain distance from the minimum loss value without falling [29]. Therefore, the initial learning rate is set to 0.01, and the learning rate decreases with each iteration. Moreover, loss is the total loss; it includes classification loss, localization loss and object detection loss. In this paper, as the training time increases, the total loss gradually decreases until it becomes stable and the training reaches convergence; otherwise, if the training continues when the training reached convergence, overfitting will occur.

As shown in Figure 8, our SSD model has longer training time than the SSD model, but its total loss value is the smallest. Moreover, the comprehensive performance of loss and time for our datasets is superior to the VOC2007 dataset and KITTI dataset.

4.3. Detection Rate. Randomly capture autonomous driving videos in different traffic scenarios to test the results of the objects detection. As shown in Figure 9, under the same detection model, the performance of our dataset is better than the KITTI dataset. Meanwhile, under the same dataset, the detection result of our SSD model is far superior to the SSD model. Especially, whether the object is small or large, our method can effectively detect and classify objects. Moreover, our approach is well adapted to environment and climate changes. At the same time, the statistics on Table 4 shows the classification detection accuracy, average detection accuracy, and detection rate of different methods.

5. Conclusions and Future Work

In this study, we propose an objects detection algorithm for autonomous driving based on SSD model. Through the feature fusion of the convolution layers, the effective transmission of the object features is guaranteed. In the process of training, a strategy of image block method was added to improve the detection performance of small objects. Moreover, we propose an image pyramid for big objects, which effectively solves the problem of large objects feature loss caused by image segmentation. However, in this paper, we defined labelling criteria and object categories to create a new dataset for autonomous vehicle technologies. Our experimental results show that the proposed detection algorithm for autonomous driving has good detection performance in this paper.

In future work, we will track the objects based on the results which have been detected, and then analyze the motion trend of the object to provide effective support for decision-making and path planning of the autonomous vehicles [30].

Data Availability

Data availability in this article is true and reliable.
Figure 7: An illustration of the accuracy curve during training and testing stage. The abscissa expresses the value of the accuracy (%), and the ordinate expresses the number of iterations (times). Left: the accuracy curve of the SSD model on the VOC2007 dataset, KITTI dataset, and SSMCAR dataset. Right: the accuracy curve of our SSD model on the VOC2007 dataset, KITTI dataset, and SSMCAR dataset.
<table>
<thead>
<tr>
<th>Detection model</th>
<th>Data set</th>
<th>Number of categories</th>
<th>Total number of training samples</th>
<th>Number of verification samples</th>
<th>Number of test sets</th>
<th>Detection accuracy of each category</th>
<th>Total accuracy</th>
<th>Detection rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSD</td>
<td>VOC-2007</td>
<td>20</td>
<td>9963</td>
<td>5011</td>
<td>4952</td>
<td>0.5145 0.5379 0.5701 0.5883 0.5334 0.5499 0.5407 0.5507</td>
<td>0.5379 0.5407 0.5499 0.5507</td>
<td>0.5507 28Fps</td>
</tr>
<tr>
<td></td>
<td>KITTI</td>
<td>5</td>
<td>7982</td>
<td>7183</td>
<td>799</td>
<td>0.6145 0.6379 0.6701 0.6883 0.6534 0.6499 0.6407 0.6507</td>
<td>0.6379 0.6407 0.6499 0.6507</td>
<td>0.6507 24Fps</td>
</tr>
<tr>
<td></td>
<td>SSM-CAR</td>
<td>7</td>
<td>11550</td>
<td>10394</td>
<td>1156</td>
<td>0.7903 0.8057 0.8109 0.8044 0.8170 0.7968 0.8102 0.8050</td>
<td>0.8057 0.8102 0.8102 0.8050</td>
<td>0.8050 27Fps</td>
</tr>
<tr>
<td>Our SSD</td>
<td>VOC-2007</td>
<td>20</td>
<td>9963</td>
<td>5011</td>
<td>4952</td>
<td>0.7714 0.7856 0.7812 0.7821 0.7830 0.7904 0.7923 0.7841</td>
<td>0.7856 0.7821 0.7830 0.7904</td>
<td>0.7841 31Fps</td>
</tr>
<tr>
<td></td>
<td>KITTI</td>
<td>5</td>
<td>7982</td>
<td>7183</td>
<td>799</td>
<td>0.8145 0.8279 0.8001 0.8083 0.8234 0.8499 0.8407 0.8107</td>
<td>0.8279 0.8083 0.8234 0.8499</td>
<td>0.8407 29Fps</td>
</tr>
<tr>
<td></td>
<td>SSM-CAR</td>
<td>7</td>
<td>11550</td>
<td>10394</td>
<td>1156</td>
<td>0.9805 0.8972 0.9196 0.9585 0.9043 0.8807 0.8984 0.9085</td>
<td>0.8972 0.9196 0.9585 0.9043</td>
<td>0.9085 33Fps</td>
</tr>
</tbody>
</table>
**Conflicts of Interest**

There are no conflicts of interest regarding the publication of this paper.

**Acknowledgments**

This work is funded by the National Natural Science Foundation of China (Grant no. 61572083), the Project of Shaanxi
Figure 9: The result of the objects detection for autonomous driving. Left: it is the detection result of SSD+KITTI. Middle: it is the detection result of SSD+SSMCAR. Right: it is the detection result of our SSD+SSMCAR. (a) is the detection result of urban roads. (b) is the detection result of the intersection. (c) is the detection result of the highway.

References


