Hindawi
Complexity
Volume 2019, Article ID 6409630, 15 pages
https://doi.org/10.1155/2019/6409630

Research Article

Vehicle Attribute Recognition for Normal Targets and Small Targets Based on Multitask Cascaded Network

Fang Liu,[1] **Yong Zhang**,[2] **Hua Gong**,[1] **Ke Xu**,[1] **and Ligang Cai**[3]

[1]*College of Science, Shenyang Ligong University, Shenyang 110159, China*
[2]*Technology on Electro-Optical Information Security Control Laboratory, Tianjing 300308, China*
[3]*College of Science, Shenyang University of Technology, Shenyang 110178, China*

Correspondence should be addressed to Hua Gong; gonghua@sylu.edu.cn

Fang Liu, Yong Zhang, and Hua Gong contributed equally to this work.

Received 20 June 2019; Revised 14 October 2019; Accepted 7 November 2019; Published 6 December 2019

Guest Editor: Teddy Craciunescu

The interference of the complex background and less information of the small targets are two major problems in vehicle attribute recognition. In this paper, two cascaded networks of vehicle attribute recognition are established to solve the two problems. For vehicle targets with normal size, the multitask cascaded convolution neural network MC-CNN-NT uses the improved Faster R-CNN as the location subnetwork. The vehicle targets in the complex background are extracted by the location subnetwork to the classification subnetwork CNN for the classification. The implementation of this task decomposition strategy effectively eliminates the interference of the complex background in target detection. For vehicle targets with small size, the multitask cascaded convolution neural network MC-CNN-ST applies the network compression strategy and the multilayer feature fusion strategy to extract the feature maps. These strategies enrich the location information and semantic information of the feature maps. In order to optimize the nonlinear mapping ability and the hard-to-detect samples mining ability of the networks, the activation function and the loss function in the two cascaded networks are improved. The experimental results show that MC-CNN-NT for the normal targets and MC-CNN-ST for the small targets achieve the state-of-the-art performance compared with other attribute recognition networks.

1. Introduction

Vehicle attribute recognition can provide the support for the statistics of the road traffic flow [1], the automatic driving of the vehicles [2, 3], and the detection and the tracking of the illegal vehicles [4]. Location and recognition for the vehicles with different sizes in complex natural scenes are important issues in the intelligent transportation researches.

The traditional methods of vehicle recognition are mainly to build 3D models of the vehicles and extract the features of the vehicles manually. In the aspect of vehicle attribute recognition based on 3D models, the Bayesian algorithm is applied to generate a 3D vehicle model for matching the features and realizing the vehicle classification [5]. A 3D curve alignment method [6] is established to identify the types of the vehicles from a single image. The image gradient is used to calculate the attitude scores of the targets, and the real-time vehicle location is realized [7]. In the aspect of vehicle attribute recognition based on the feature extraction manually, the directional gradient histogram (HOG) [8] is fused with the rectangular filter. The features are extracted manually to recognize the vehicle targets. Scale-Invariant Feature Transform (SIFT) is proposed to describe the edge features and classify the vehicles [9]. The edge-oriented histogram is obtained to extract the vehicle features [10]. These features are input into the support vector machine (SVM) classifier to classify the vehicles. The traditional vehicle recognition methods essentially use the human-made features to represent the images. Since the hand-made features are sensitive to illumination, shooting angle, and target background, the generalization ability of the algorithms is weak. Moreover, the extraction speed of the manual features is slow, which does

not meet the needs for the recognition of massive data of intelligent transportation.

In recent years, since the deep convolution networks have excellent performance in dealing with the big data samples and nonlinear mapping, they are widely used in the field of target recognition. At present, there are mainly two kinds of target recognition methods based on the deep convolution networks. The first method is the region proposal method, such as R-CNN [11], Fast-CNN [12], Faster R-CNN [13], and SPP Net [14]. The other method is the region-free method, such as YOLO [15] and SSD [16]. These two deep network recognition methods are also applied in vehicle recognition. The multitask R-CNN method is established to recognize four types of vehicles (car, truck, bus, and van) [17]. The improved Faster R-CNN [18] is constructed to detect the vehicles in aerial images. The use of the hyper region proposal network (HRPN) and the multiple boosted classifiers reduces false detection. SSD with the feature fusion method [19] is proposed to recognize the six categories of vehicles (cyclist, motocycle, bus, minibus, car, and truck) and persons. In [19], an image segmentation strategy is employed to improve the recognition effect for the small targets. Vehicle attribute recognition methods based on the deep convolution networks are driven by data to extract the features; this strategy eliminates the sensibility of the hand-made features, and the methods are universal and accurate. However, the abovementioned vehicle recognition methods based on the deep convolution networks all adopt one-stage networks to complete vehicle location and recognition at the same time. The strategy of doing multiple tasks in the same network has two main shortcomings. First, it is easy to produce false acceptance and false rejection in the complex background (illumination change, local occlusion, target scale change) [20]. Second, the target images undergo several convolutions and pooling, which results in the disappearance or the transformation of the location of the feature points for the small targets. The recognition accuracy for the small targets is reduced.

In view of two shortcomings of one-stage deep convolution neural networks, the multitask cascaded neural networks and the multiscale feature fusion networks are established. In the aspect of the multitask cascaded neural networks, the two-stage multitask cascaded CNN [3] is applied to recognize the vehicles. The multitask cascaded network based on IFR-CNN and CNN [21] is obtained to recognize different types of vehicles. The two-stage cascaded YOLO [22] is established to carry out the vehicle location and license plate recognition. The multitask convolution neural networks are devised to segment the targets from the complex background firstly and then recognize the targets. These methods divide the background and the targets and eliminate the interference of the complex background for the target recognition. The accuracy of target attribute recognition is higher than that of one stage deep convolution networks. In the aspect of the multiscale feature fusion, the feature pyramid network (FPN) [23] as a feature extractor achieves the most advanced single-model recognition results on COCO datasets. The FPN network uses multiscale fusion features to describe the target information, which solves the problem of the feature disappearance for the small targets. The feature fusion networks are widely used in the fields of human body detection [24], situation assessment [25], and face recognition [26]. However, the multiscale feature fusion models for the small targets are few in the vehicle attributes recognition.

We devote this paper to the study of vehicle attribute recognition models. Our contributions are mainly as follows. (1) Two cascaded models MC-CNN-NT and MC-CNN-ST are proposed. MC-CNN-NT is applied to recognition vehicle targets with normal size. MC-CNN-ST is used to recognition vehicle targets with small size. (2) The activation function and the loss function in the two cascaded networks are improved. The performance of feature extraction and classification of two networks is enhanced. (3) The strategies of network compression and feature fusion in MC-CNN-ST are employed. The object edge information extracted by the bottom filter and semantics information extracted by the high-level filter are fused to realize the precise location of the vehicles. (4) The SYIT-Vehicle dataset and the COCO-Vehicle dataset are constructed and annotated. The target quantity and quality in the two dataset provide the guarantee for verifying network performance.

The rest of this paper is outlined as follows. Section 2 and Section 3 describe the architectures of MC-CNN-NT and MC-CNN-ST in detail, respectively. Section 4 reports the experimental results of the two multitask cascaded networks. Section 5 denotes the conclusions.

2. Architecture: The Multitask Cascaded Network MC-CNN-NT

This section demonstrates the architecture of the multitask cascaded network MC-CNN-NT for recognizing vehicle targets with normal size. Section 2.1 demonstrates the framework of MC-CNN-NT. Section 2.2 introduces the improvement of the activation function and the loss function in MC-CNN-NT. Section 2.3 shows the basic processes of attribute recognition using MC-CNN-NT in detail.

2.1. The Framework of MC-CNN-NT. Aiming at the low accuracy of the vehicle target recognition in the complex background with one stage network, a multitask cascaded network MC-CNN-NT is constructed. The problem of vehicle attribute recognition is decomposed into two subproblems: target location and target classification. The improved Faster R-CNN is employed as the location subnetwork of MC-CNN-NT. This subnetwork consists of three parts: the network of the feature extraction, the network of the region proposal (RPN), and the network of the object location. CNN is used as the classification subnetwork in the cascaded networks. The framework of MC-CNN-NT is shown in Figure 1.

The feature extraction network in the MC-CNN-NT location subnetwork applies VGG-D as the backbone network to extract the image features. 13 convolution layers and the first four subsampling layers in the VGG-D model are selected. The fifth subsampling layers and three full-connection layers

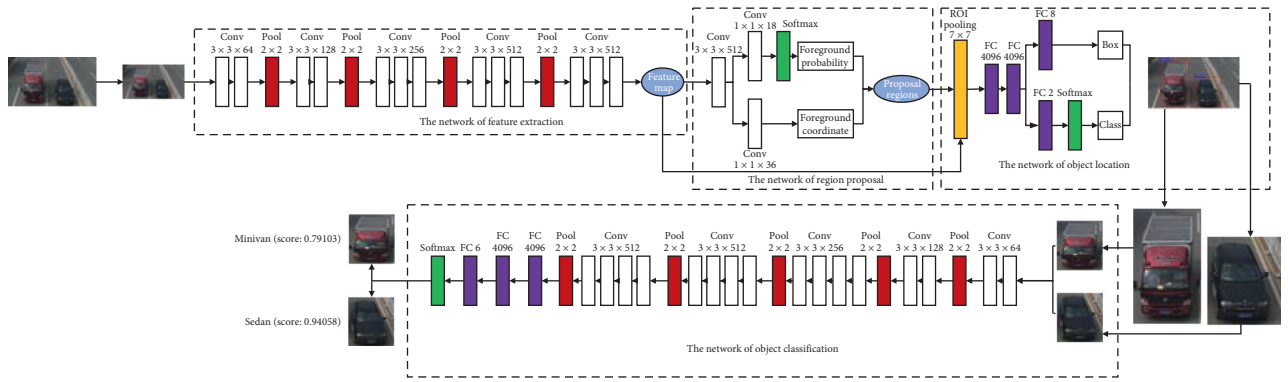


Figure 1: The framework of MC-CNN-NT.

in the VGG-D model are discarded. The parameters of the feature extraction network are detailed in Table 1. In RPN, the sizes of anchors are set to $\{128 \times 128, 256 \times 256, 512 \times 512\}$. The length-width ratios of anchors are set to $\{1:2, 1:1, 2:1\}$. The number of anchors is $3 \times 3 = 9$. A 3×3 sliding box is used to traverse the top-level feature map of the feature extraction network. Each pixel on the feature map corresponds to 9 anchors of different sizes in the original maps. In RPN, the classification layer and the regression layer output the scores of 9 anchors corresponding to each pixel and their respective location coordinates. Let the top feature map size of the shared convolution be $w \times h$. The classification layer outputs the scores of $w \times h \times 9 \times 2$ candidate regions. The regression layer outputs $w \times h \times 9 \times 4$ coordinate parameters. As Fast R-CNN, ROI pooling is carried out for the proposal regions of RPN network. The coordinate parameters of the targets are output through the multiple full-connection layers.

The classification subnetwork of MC-CNN-NT is based on CNN. VGG-E is selected as the backbone network of the classification subnetwork. According to the coordinate values output by the location subnetwork, the target regions are cut out from the original images. The extracted image size is normalized to $227 \times 227 \times 3$. The normalized single target images are input into the classification subnetwork for vehicle type recognition. The parameters of the classification subnetwork are detailed in Table 2.

2.2. Activation Function and Loss Function. In order to enhance the recognition performance of the cascaded network MC-CNN-NT, the activation function and the loss function are improved in Faster R-CNN and CNN.

2.2.1. Activation Function: PReLU. The activation function is an important part of the deep network. The form of the activation function plays a key role in the quality of the feature extraction. The activation function can be divided into two categories: the saturated activation function and the unsaturated activation function. Since the unsaturated activation function has the advantages of solving "gradient disappearance" and fast convergence speed, it attracts more attentions from researchers of deep learning [27, 28].

Table 1: Parameters of the feature extraction network of MC-CNN-NT.

Layer	Type	Number	Size	Stride	Padding
1	Conv	64	3×3	1	2
2	Conv	64	3×3	1	2
3	Pool	64	2×2	2	0
4	Conv	128	3×3	1	2
5	Conv	128	3×3	1	2
6	Pool	128	2×2	1	2
7	Conv	256	3×3	2	0
8	Conv	256	3×3	1	2
8	Conv	256	3×3	1	2
10	Pool	256	2×2	2	0
11	Conv	512	3×3	1	2
12	Conv	512	3×3	1	2
13	Conv	512	3×3	1	2
14	Pool	512	2×2	2	0
15	Conv	512	3×3	1	2
16	Conv	512	3×3	1	2
17	Conv	512	3×3	1	2

In this paper, ReLU [29] is updated to PReLU [27]. PReLU function formula is described as follows:

$$g(x_i) = \max(0, x_i) + k_i \min(0, x_i). \quad (1)$$

ReLU function formula is shown as follows:

$$g(x) = \max(0, x), \quad (2)$$

where x is the input signal of the activation function, k is a coefficient controlling the slope of the negative part in the PReLU function, and the subscript i denotes the channel i. The improved activation function PReLU adds a linear term to the negative signals. This strategy alleviates the problem of gradient disappearance when the network propagates back to the negative signals. When the activation function has better nonlinear mapping ability for different negative signals, the learning strategy is adopted for the slope k_i in model training.

In this paper, k_i is trained by using backpropagation and updated using the momentum method. According to the chain rule, the gradient derivative formula of k_i is formed as follows:

Table 2: Parameters of the classification subnetwork of MC-CNN-NT.

Layer	Type	Number	Size	Stride	Padding
1	Conv	64	3×3	1	2
2	Conv	64	3×3	1	2
3	Pool	64	2×2	2	0
4	Conv	128	3×3	1	2
5	Conv	128	3×3	1	2
6	Pool	128	2×2	2	0
7	Conv	256	3×3	1	2
8	Conv	256	3×3	1	2
8	Conv	256	3×3	1	2
10	Conv	256	3×3	1	2
11	Pool	256	2×2	2	0
12	Conv	512	3×3	1	2
13	Conv	512	3×3	1	2
14	Conv	512	3×3	1	2
15	Conv	512	3×3	1	2
16	Pool	512	2×2	2	0
17	Conv	512	3×3	1	2
18	Conv	512	3×3	1	2
19	Conv	512	3×3	1	2
20	Conv	512	3×3	1	2
21	Pool	512	2×2	2	0
22	FC	4096	1×1	—	—
23	FC	4096	1×1	—	—
24	FC	6	1×1	—	—
25	Softmax	—	—	—	—

$$\frac{\partial\sigma}{\partial k_i}=\sum_{x_i}\frac{\partial\sigma}{\partial g(x_i)}\frac{\partial g(x_i)}{\partial k_i}, \tag{3}$$

where σ is the objective function of the model and $\partial\sigma/\partial g(x_i)$ is the gradient function transferred by the deeper convolution neural network. The gradient derivative of the activation function $g(x_i)$ is demonstrated as follows:

$$\frac{\partial g(x_i)}{\partial k_i}=\begin{cases}0, & \text{if } x_i\geq 0,\\ x_i, & \text{if } x_i<0.\end{cases} \tag{4}$$

The reverse update formula of k_i is adopted as follows:

$$\Delta k_i:=\delta\Delta k_i+\eta\frac{\partial\sigma}{\partial k_i}, \tag{5}$$

where δ represents the momentum and η represents the learning rate of the network.

As shown equation (4), the gradient derivative of PReLU activation function only adds a very small number of parameters. The computational complexity of the network and the risk of overfitting can be neglected. The adaptability of rectifier parameter k_i improves the training accuracy of the cascaded network. The inherent unsaturation of PReLU function makes it perform better in controlling gradient and convergence rate.

2.2.2. Loss Function. The location subnetwork of MC-CNN-NT is the improved Faster R-CNN; it needs to train RPN and Fast R-CNN. The classification subnetwork of MC-CNN-NT needs to train CNN. For three different networks, three loss functions $L_{\mathrm{RPN}}(\{p_i\},\{t_i\})$, $L_{\mathrm{Fast}\,R-\mathrm{CNN}}(\{p_i\},\{t_i\})$, and $L_{\mathrm{CNN}}(\{p_i\})$ are described as follows:

$$L_{\mathrm{RPN}}(\{p_i\},\{t_i\})=\frac{1}{N_{\mathrm{cls}}}\sum_i \mathrm{FL}_{\mathrm{cls}}(p_i)+\omega\frac{1}{N_{\mathrm{reg}}}\sum_i \delta_i L_{\mathrm{reg}}(t_i,t_i^*), \tag{6}$$

$$L_{\mathrm{Fast}\,R-\mathrm{CNN}}(\{p_i\},\{t_i\})=\sum_i \mathrm{FL}_{\mathrm{cls}}(p_i)+\sum_i L_{\mathrm{reg}}(t_i,t_i^*), \tag{7}$$

$$L_{\mathrm{CNN}}(\{p_i\})=\sum_i \mathrm{FL}_{\mathrm{cls}}(p_i). \tag{8}$$

RPN and Fast R-CNN are two multitask networks. Their loss functions are composed of classification loss and regression loss. CNN only completes the classification task, and the loss function is only related to the classification loss. Here, $\mathrm{FL}_{\mathrm{cls}}(p_i)$ represents classification loss, and $L_{\mathrm{reg}}(t_i,t_i^*)$ represents regression loss.

In the R-CNN series network framework, the commonly used loss function of classification is the cross-entropy loss function. The cross-entropy loss function uses the logarithmic loss $\log(p_i)$ to characterize the difference between the real sample and the prediction box. Although the cross-entropy loss function has a low loss value for a single easy-to-detect sample, it still contributes a lot to the overall loss of the network due to the large number of the easy-to-detect samples. However, due to the small number of the hard-to-detect samples, the contribution of the hard-to-detect samples to the loss function is small. As a result, the training of the network is towards the easy-to-detect samples, which affects the quality of the network recognition. In this paper, we update the cross-entropy classification loss function to the Focal loss function. The strategy of assigning different weights to different samples is adopted to increase the contribution of the hard-to-detect samples in the loss function.

The Focal Loss function is defined as follows:

$$\mathrm{FL}_{\mathrm{cls}}(p_i)=-(1-p_i)^{\gamma}\log(p_i), \tag{9}$$

$$p_i=\frac{e^{z_i}}{\sum_{q=1}^{k}e^{z_q}}, \tag{10}$$

where z_i is the network output of the category i, and p_i $(0\leq p_i\leq 1)$ is the output probability of the category i. The Focal Loss function adds a modulating factor $(1-p_i)^{\gamma}$ to the standard cross-entropy loss function. As shown in Figure 2, $-\log^{p_i}$ and $(1-p_i)^{\gamma}$ are small for the easy-to-detect samples (p_i is large). They lead that $\mathrm{FL}_{\mathrm{cls}}(p_i)$ is small. The weights are slightly adjusted when the deep network backpropagation occurs. $-\log^{p_i}$ and $(1-p_i)^{\gamma}$ are large for the hard-to-detect samples (p_i is small). They lead that $\mathrm{FL}_{\mathrm{cls}}(p_i)$ is large. The weights are dramatically adjusted when the deep network backpropagation occurs. The learning of the hard-to-detect samples is strengthened. In equation (9), γ is a parameter for adjusting the weight rate, which is called focusing parameter. When $\gamma=0$, the Focal Loss function is equal to the cross-entropy loss function. The influence of the modulation factor is increased with the increase of the value γ.

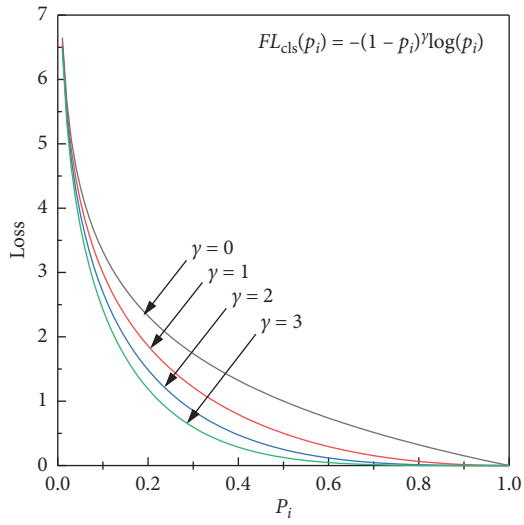


Figure 2: The curve of Focal Loss function.

The Smooth L_1 function is used as the regression loss in $L_{\mathrm{RPN}}(\{p_i\},\{t_i\})$ and $L_{\mathrm{Fast}\,R-\mathrm{CNN}}(\{p_i\},\{t_i\})$ functions [30]. The definition of the $L_{\mathrm{reg}}(t_i,t_i^*)$ function is denoted in the following equation:

$$L_{\mathrm{reg}}(t_i,t_i^*) = \mathrm{Smooth}_{L_1}(t_i,t_i^*) = \begin{cases} 0.5(t_i - t_i^*)^2, & |t_i - t_i^*| < 1, \\ |t_i - t_i^*| - 0.5, & \text{otherwise}, \end{cases} \tag{11}$$

where $t_i = (t_{ix}, t_{iy}, t_{iw}, t_{ih})$ represents the translation scaling values of the four prediction boxes. $t_i^* = (t_{ix}^*, t_{iy}^*, t_{iw}^*, t_{ih}^*)$ is the four coordinates of the ground-true box. δ_i is a label function in equations (6) and (7). If the label of the prediction box is a positive sample, $\delta_i = 1$. Otherwise, $\delta_i = 0$. According to the definitions of the regression loss of RPN and Fast R-CNN, it can be seen that the regression losses of two networks are calculated only for the positive samples. Here, the definitions of the positive samples and the negative samples in PRN and Fast R-CNN adopt the definitions of reference [13] and reference [12], respectively.

In equation (6), the classification loss is normalized by the mini-batch size N_{cls}. The regression loss is normalized by the number of anchors N_{reg}. Set $N_{\mathrm{cls}} = 256$, $\omega = 10$, $N_{\mathrm{reg}} = 2400$. By using the normalization strategy, the weight of the classification loss is approximately equal to the weight of the regression loss in RPN. In equation (7), the weight of the classification loss and the regression loss in Fast R-CNN is set to 1, and the contribution of two kinds of losses to the total loss is equal.

2.3. The Basic Procedures of the Multitask Cascaded Network MC-CNN-NT. The specific steps using MC-CNN-NT to recognize the vehicles are as follows:

Step 1 (partitioning the dataset): firstly, according to the ratio of 9 : 1, the dataset is divided into two parts: the training verification set and the testing set. Then, according to the ratio of 9 : 1, the training verification set is divided into two parts: the training set and the verification set. According to such partitioning rules, the original dataset is divided into three parts: the training set, the verification set, and the testing set.

Step 2 (annotating the dataset): the positions of the vehicle targets in the images are labeled. The coordinates of the upper left corner and the lower right corner of the vehicle targets in the images are recorded. The generated annotation information is saved in the corresponding XML file.

Step 3 (preprocessing the image data): the images are scaled. The color formats of the images are converted. Let the zoom ratio be S, the input image size be $W \times H$, and the zoomed image size be $W' \times H'$. Here, $S = W/W' = H/H'$. When the images are scaled, the long side is less than 1000, and the short side is less than 600 (at least one is equal). The target regions in the images are also scaled at the same scale. Since the Caffe frame recognizes the BGR color format, the RGB (Red-Green-Blue) color format of the images is converted to the BGR (Blue-Green-Red) color format.

Step 4 (setting the hyperparameters of the cascaded network): let the maximum number of the iterations in the location subnetwork and the classification subnetwork of MC-CNN-NT be $N_{\mathrm{max}}^{\mathrm{loc}}$ and $N_{\mathrm{max}}^{\mathrm{clc}}$, respectively. Let the initial learning rate in the location subnetwork and the classification subnetwork of MC-CNN-NT be η_{loc} and η_{clc}, respectively.

Step 5 (initializing the weights and the thresholds of the cascaded network): for the location subnetwork of MC-CNN-NT, the VGG-D model parameters are applied to initialize the parameters of the convolution layer shared by RPN and Fast R-CNN. The parameters of the unique layers of RPN and Fast R-CNN are initialized by Gaussian distribution with the mean of 0 and the standard deviation of 0.01. The thresholds of each layer are initialized by the constant 0.

For the classification subnetwork of MC-CNN-NT, the VGG-A model parameters are used to initialize the parameters of the first four convolution layers and the last three full connection layers of VGG-E. The parameters of the other layers are initialized by Gaussian distribution with the mean of 0 and the standard deviation of 0.01. The thresholds of each layer are initialized by the constant 0.

Step 6 (training the location subnetwork): train the location subnetwork using the images coming from the training set. The weights and the thresholds of the location subnetwork are adjusted by cyclic iterations. When the number of the iterations of the location subnetwork is greater than $N_{\mathrm{max}}^{\mathrm{loc}}$, execute step 7.

Step 7 (testing the location subnetwork): test the location subnetwork using the images coming from the testing set. The target prediction boxes on the testing images are output. The position coordinates of the prediction boxes are obtained.

Step 8 (extracting the targets from the location subnetwork): according to the coordinates of the target prediction boxes in the location subnetwork, the target regions are extracted from the original images as the input images of the classification subnetwork.

Step 9 (preprocessing the classification subnetwork images): the size of the images that are input into the classification subnetwork is adjusted to $227 \times 227 \times 3$.

Step 10 (training the classification subnetwork): the weights and the thresholds of the classification network are adjusted by cyclic iterations. When the number of the iterations is greater than the number of the iterations $N_{\max}^{\text{clc}}$, execute step 11.

Step 11 (testing the classification subnetwork): the classification subnetwork is tested with the images coming from the testing set. The confidence scores of each testing image belonging to different categories are obtained. The category with the highest confidence score is the recognized category of the target.

3. Architecture: The Multitask Cascaded Network MC-CNN-ST

The regions of the small target images contain few pixels. If the deep convolution neural network is used to extract information from the deep feature maps, the edge and the detailed information of the images is lost. The recognition accuracy of the small targets is reduced. In this section, in order to solve this problem, the cascaded strategy of the multitask network is employed and a cascaded network MC-CNN-ST is established. This new network is more practical for the attribute recognition of the small targets. Section 3.1 introduces the structure of the location subnetwork in MC-CNN-ST. Section 3.2 demonstrates the structure of the classification subnetwork in MC-CNN-ST. The data augmentation section is described in Section 3.3.

3.1. The Location Subnetwork of MC-CNN-ST. In Figure 3, the convolution layer i is represented as Conv i, the subsampling layer i is represented as Dpool i, and the upsampling layer is represented as Upool i. FC represents the full connection layer. RS is the abbreviation of reshape, and it is a data reorganization layer. Softmax is a classifier.

In the location subnetwork of MC-CNN-ST, the network compression strategy and the feature fusion strategy are proposed to improve the feature extraction quality. The location sub network of MC-CNN-ST inherits RPN and the object location network in the location sub network of MC-CNN-NT. The improved activation function and the loss function in MC-CNN-NT are also applied to MC-CNN-ST. Figure 3 shows the framework of MC-CNN-ST.

In the network of the feature extraction of MC-CNN-ST, the last seven convolution layers of VGG-D in the location sub network of MC-CNN-NT are abandoned. The number of the network layers is compressed to 6. The first six convolution layers of VGG-D use 3×3 size convolution cores. The second convolution layer and the fourth convolution layer connect a 2×2 size subsampling layer, respectively. This forward propagation network structure with six convolution layers and two subsampling layers is called as the backbone network of the feature extraction. In order to extract richer feature information, the lateral connection structure in the feature extraction network is constructed. The first convolution layer, the third convolution layer, and the sixth convolution layer of the backbone network connect two convolution cores of 3×3 size in the lateral connection paths, respectively. The three branches of the network are called Branch 1, Branch 2, and Branch 3, respectively. Branch 1 consists of Conv1, Dpool3, Conv7, and Conv8. Branch 2 consists of Conv2, Dpool1, Conv3, Conv9, and Conv10. Branch 3 consists of Conv4, Dpool2, Conv5, Conv6, Upool1, Conv11, and Conv12. The composition of the three branches is shown in Figure 3.

Branch 1 integrates the information of the first convolution layer into the feature maps, and the footprints of the target locations are well preserved. Branch 2 integrates the information of the third convolution layer into the feature maps, which includes the edge information of the targets and the semantic information of the images. Branch 3 integrates the information of the sixth convolution layer into the feature maps, and the strong semantic information of the targets is incorporated into the feature maps. The extracted features of the three branches include the details of the vehicle edges in the shallow feature maps and the strong semantic information in the high-level feature maps. The implementation of the network fusion strategy enriches the diversity of the extracted features. A 2×2 subsampling layer and a 2×2 upsampling layer are added to Branch 1 and Branch 3, respectively. Using this scheme, the scale of the feature maps output by each branch is consistent. After one subsampling and multiple convolutions, the size of the feature maps becomes a quarter of the original ones. The processes of the size change of the feature maps in three branches are demonstrated in Figure 4. As shown in Figure 4, Branch 1 generates 32 feature maps, Branch 2 generates 64 feature maps, and Branch 3 generates 128 feature maps. The feature extraction network stacks and fuses the output feature maps of three branches and generates 224 feature maps for RPN.

3.2. The Classification Subnetwork of MC-CNN-ST. A new shallow convolution network is constructed as a classification sub network of MC-CNN-ST. The shallow network applies the activation function PReLU and the loss function Focal Loss in MC-CNN-NT. The network structure is shown in Figure 3. The network consists of three convolution layers Conv16, Conv17, Conv18, two maximum pooling layers Dpool4, Dpool5, a data reorganization layer reshape, and a softmax classifier. The images to be classified are the single target images extracted by the location sub network of MC-CNN-ST. The image size is normalized to 28×28. The number of the input images for each batch is set to 50. Since each image has three channel charts of blue, green, and red, the number of Conv16 images that input into the classification network is $3 \times 50 = 150$. The size transition processes

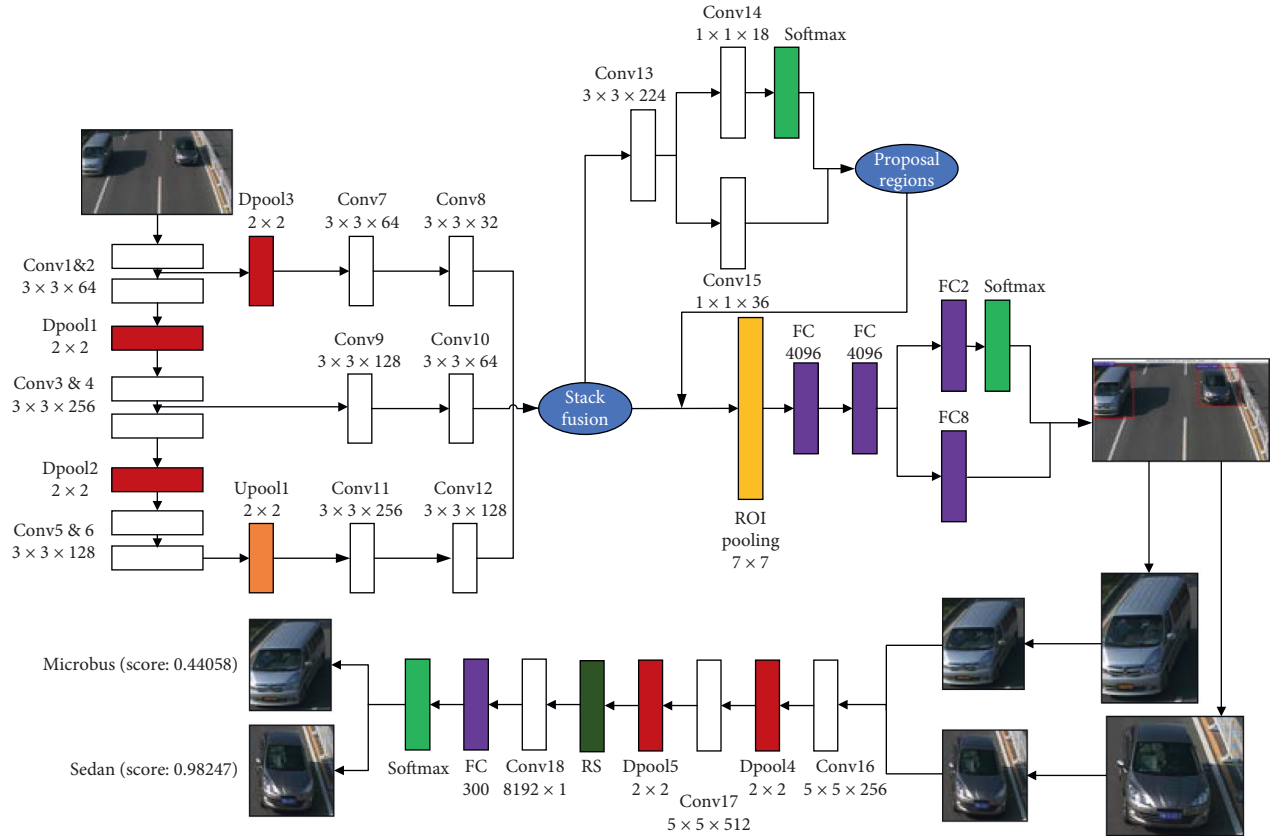


Figure 3: The framework of MC-CNN-ST.

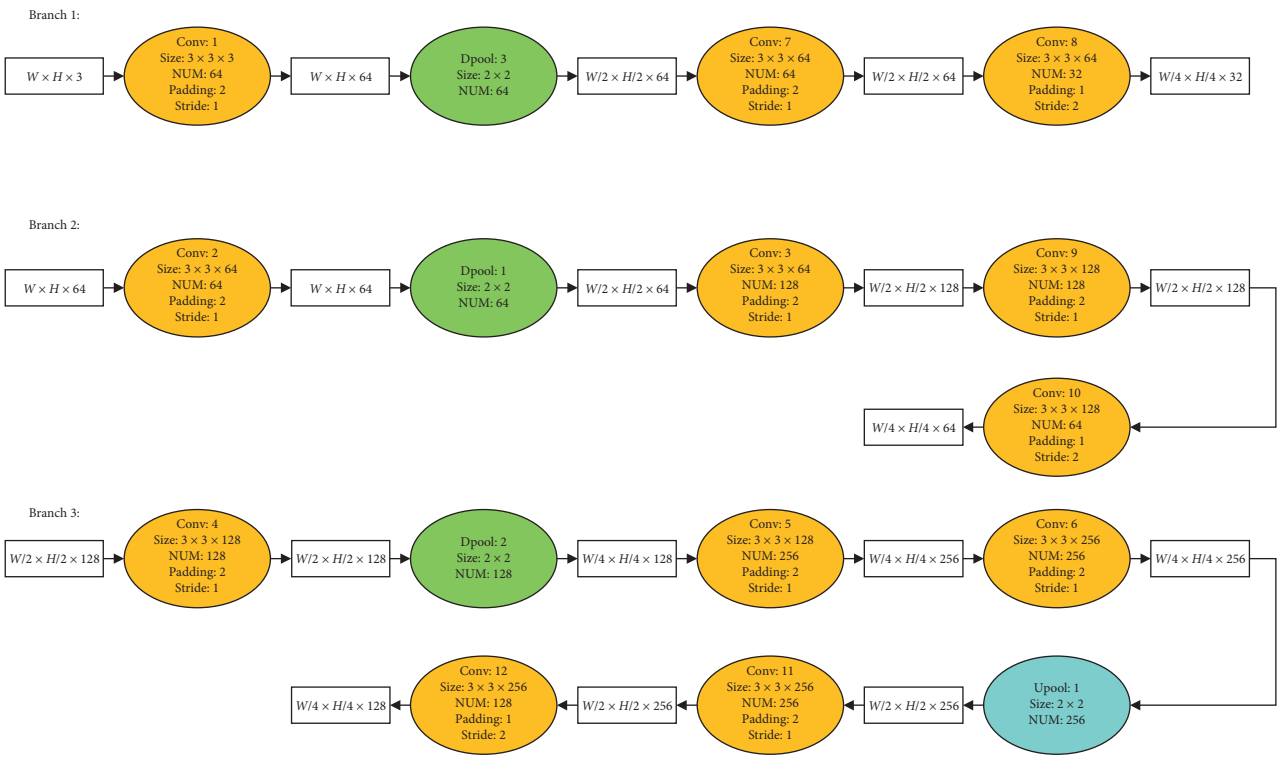


Figure 4: The processes of the size change of the feature maps in three branches.

from Conv16 to Conv18 are shown in Table 3. Here, the reshape layer reorganizes the features coming from Dpool5, the size of the feature map vectors becomes 8192×150. Conv18 uses six convolution cores of 8192×1 size to extract the features. The width of the feature maps output by Conv18 is $(8192 - 8192 + 0)/1 + 1 = 1$. The height of the feature maps output by Conv18 is $(150 - 1 + 0)/1 + 1 = 150$. The full connection layer maps the distributed feature extracted from Conv18 to the sample label space. The number of neural nodes in the full connection layer is set to $50 \times 6 = 300$. Here, 50 is the number of the images input in each batch and 6 is the number of the vehicle categories. Softmax classifier outputs the probability values of each target belonging to six type vehicles. The category that obtains the maximum probability value is the vehicle recognition category. The network adopts the strategies of inputting multiple images in the same batch and the shallow network, the recognition speed of the network is enhanced.

3.3. Data Augmentation. When MC-CNN-ST has robust localization performance for small targets with different sizes, positions, and perspectives, we use random sampling to translate, rotate, flip, and cut the images. The specific operations are presented as follows. (1) Targets are randomly translated $10k$ ($k = \pm 5, \pm 10, \pm 15, \pm 20$) pixels along X or Y axis. (2) The images are rotated 180 degrees. (3) The images are flipped along the axis of the image center. (4) The length of the original images is randomly clipped as the input images. These samples contain at least one central point of the targets. Four data augmentation schemes can effectively avoid overfitting of the model.

4. Experiments

In this section, two groups of experiments are designed to verify the effectiveness of the cascaded network MC-CNN-NT and MC-CNN-ST. Section 4.1 introduces the hardware and software environment of the experiments and the initial setting of the network parameters. Section 4.2 shows the experimental results for the normal targets using MC-CNN-NT. Section 4.3 demonstrates the experimental results for the small targets using MC-CNN-ST.

4.1. Environment and the Initial Value Settings. The experiments use Caffe framework to implement the target detection algorithms. The hardware in the experiments is used as follows: 32 GB RAM, Intel i7 CPU and NVIDIA Geforce GTX1080Ti 11 GB GPU. The software in the experiments is applied as follows: Ubutu 16.04, Python 2.7.14, CUDA8.0 and CUDNN 6.0.

In MC-CNN-NT, some network initial weights are pretrained on the ImageNet dataset. These weights partly locates in the convolution layers shared by PRN and Fast R-CNN in the location sub network, and partly locates in the first four convolution layers and the last three full connection layers of VGG-E in the classification sub network. The other convolution layer initial weights are initialized by Gaussian distribution with the mean value of 0 and the standard deviation of 0.01. The maximum number of the iterations is set to 70,000. The learning rate of the first 50,000 iterations is set to 0.001. The learning rate of the last 20,000 iterations is reduced to 0.0001.

In MC-CNN-ST, the Gaussian distribution with the mean value of 0 and the standard deviation of 0.01 is used to initialize the weights of the whole network randomly. The maximum number of the iterations is set to 100,000. The learning rate of the first 50,000 iterations is set to 0.001. The learning rate of the last 50,000 iterations is reduced to 0.0001.

The other initial parameters of two groups of experiments use the same setting strategy. The thresholds of two cascaded networks are initially set to 0. The focusing parameter of the Focal Loss function γ is set to 2. The momentum term is set to 0.9. The weight-decay coefficient is set to 0.0005. Dropout method is applied to prevent overfitting, and the probability of discarding network neurons is 0.5.

4.2. Experiments of Vehicle Attributes Recognition for the Normal Targets. In the experiments of vehicle attribute recognition for the normal targets, two datasets are selected: the SYIT-Vehicle dataset and BIT-Vehicle dataset [31]. The common feature of the two datasets is that they contain six categories of vehicles: bus, microbus, minivan, sedan, SUV, and truck. The differences of the two datasets are that the location, size, environment, and quantity of the images.

4.2.1. Datasets. The vehicle images in the BIT-Vehicle dataset are derived from the road monitoring. The images are obtained directly above the road surveillance camera. The sizes of images are 1600×1200 and 1600×1080, respectively. The dataset contains 9850 images with 10053 vehicle targets. Among them, the number of bus, microbus, minivan, sedan, SUV, and truck is 558, 883, 476, 5922, 1392, and 822, respectively. The sample images of the dataset are shown in Figure 5.

The vehicle images in SYIT-Vehicle dataset are derived from the image capturing equipment of the Institute of Optimization Theory and Process Control in Shenyang Ligong University. The dataset contains the vehicle images with multi-region, multi-angle, and multi-illumination. The dataset contains 12000 images with 12161 vehicle targets. Among them, the number of bus, microbus, minivan, sedan, SUV, and truck is 1770, 2174, 1817, 2626, 1891, and 1883, respectively. The sizes of images are not fixed. The background of the vehicle targets of the SYIT-Vehicle dataset is more complex than that of the BIT-Vehicle dataset. The sizes of vehicle targets of the SYIT-Vehicle dataset are more diverse than those of the BIT-Vehicle dataset. The sample images of the SYIT-Vehicle dataset are shown in Figure 6.

In this paper, the SYIT-Vehicle dataset is employed as the training validation set and the testing set. The sample ratio of the training validation set and the testing set is 9 : 1. The sample ratio of the training set and the verification set is set to 9 : 1. The BIT-Vehicle dataset is used as the migration dataset. The robustness of the algorithm is verified by randomly selecting 3600 images from the BIT-Vehicle dataset.

Table 3: Parameters of the classification sub network in MC-CNN-ST.

Position	Feature map input size	Kernel size	Kernel number	Stride	Padding	Feature map output size
Conv16	28×28	5×5	256×150	1	0	24×24
Dpool4	24×24	2×2	256×150	2	0	12×12
Conv17	12×12	5×5	512×150	1	0	8×8
Dpool5	8×8	2×2	512×150	2	0	4×4
Reshape	4×4	—	—	—	—	8192×150
Conv18	8192×150	8192×1	6	1	0	1×150

Figure 5: The samples of the BIT-Vehicle dataset.

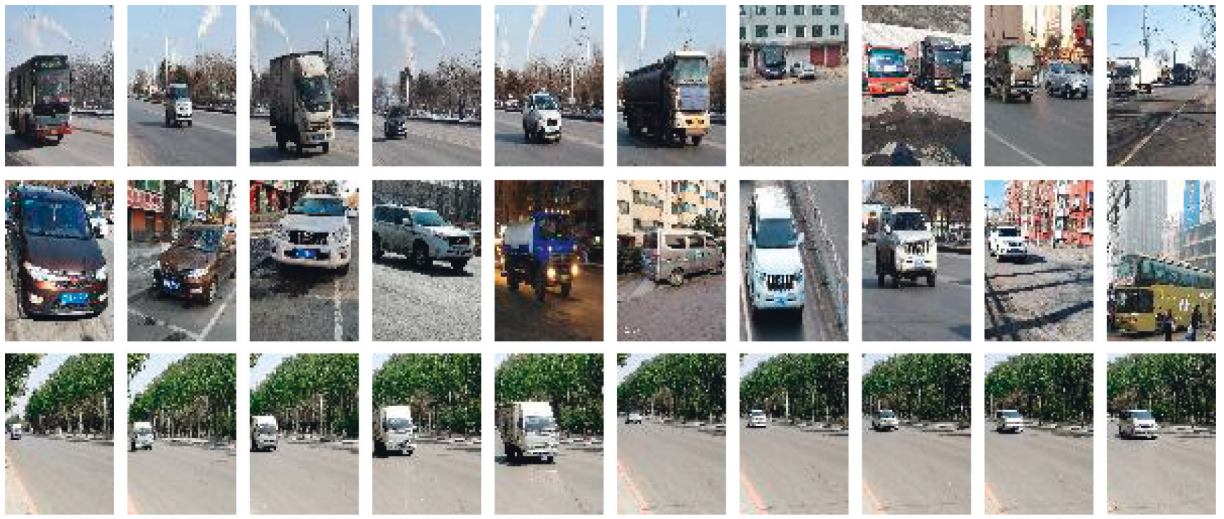


Figure 6: The samples of the SYIT-Vehicle dataset.

4.2.2. Results and Analysis. In this section, the target attribute recognition experiments using MC-CNN-NT are described. Nine networks are selected as the comparison networks. Three noncascaded networks (CNN, Fast R-CNN, and Faster R-CNN) and six cascaded networks (CNN + CNN [3], IFR-CNN + CNN [21], MC-CNN, MC-CNN-P, MC-CNN-F, and MC-CNN-NT) are included in nine networks. Table 4 describes the principal structures of nine networks participating in the comparison.

The results of vehicle attribute recognition using MC-CNN-NT are shown in Tables 5 and 6. The attribute recognition accuracy of the cascaded network MC-CNN is significantly higher than that of the noncascaded network. The recognition accuracy of MC-CNN is 84.24% on the

Table 4: The main structures of nine networks.

Method	Structure
CNN	One stage + exhaustive sliding window + CNN + SVM + single loss
Fast R-CNN	One stage + selective search + CNN + softmax + multi loss
Faster R-CNN	One stage + RPN + CNN + softmax + multi loss
CNN + CNN	Two stages + CNN + CNN
IFR-CNN + CNN	Two stages + IFR-CNN + CNN
MC-CNN	Two stages + faster R-CNN + CNN
MC-CNN-P	MC-CNN + PReLU
MC-CNN-F	MC-CNN + Focal Loss
MC-CNN-NT	MC-CNN + PReLU + Focal Loss

Table 5: The attribute recognition results for the normal targets on the SYIT-Vehicle dataset.

Method	Bus (%)	Microbus (%)	Minivan (%)	Sedan (%)	SUV (%)	Truck (%)	Total (%)
CNN	80.00	68.65	75.14	63.25	64.33	68.65	70.00
Fast R-CNN	78.92	70.81	73.51	68.33	70.27	76.22	72.97
Faster R-CNN	87.03	76.76	78.92	69.73	81.62	77.30	79.82
CNN + CNN	88.94	79.31	80.14	75.23	83.20	79.41	81.89
IFR-CNN + CNN	92.49	84.23	83.74	82.23	84.02	81.41	84.96
MC-CNN	91.35	83.70	83.70	80.92	84.78	81.00	84.24
MC-CNN-P	94.05	84.67	83.62	83.16	84.78	80.46	85.12
MC-CNN-F	94.60	84.62	84.24	84.08	87.57	83.62	86.46
MC-CNN-NT	97.30	91.11	90.57	88.41	90.73	89.49	91.27

Table 6: The attribute recognition results for the normal targets on the BIT-Vehicle dataset.

Method	Bus (%)	Microbus (%)	Minivan (%)	Sedan (%)	SUV (%)	Truck (%)	Total (%)
CNN	60.00	41.08	48.11	44.87	52.44	50.27	52.88
Fast R-CNN	78.92	42.17	48.11	45.95	54.60	58.92	57.30
Faster R-CNN	81.24	64.33	64.33	62.17	66.49	65.95	69.46
CNN + CNN	82.24	66.03	67.00	65.57	67.3	68.03	71.44
IFR-CNN + CNN	85.24	70.43	78.01	70.21	73.68	76.24	75.64
MC-CNN	83.70	70.19	76.68	69.11	70.19	76.14	74.33
MC-CNN-P	88.03	71.77	82.08	70.73	73.98	75.60	77.03
MC-CNN-F	89.11	73.52	83.70	72.35	75.22	77.00	78.48
MC-CNN-NT	90.19	77.76	86.41	75.60	79.38	86.41	82.63

SYIT-Vehicle dataset, which is 4.42% higher than that of Faster R-CNN. The recognition accuracy of MC-CNN is 74.33% on the BIT-Vehicle dataset, which is 4.87% higher than that of Faster R-CNN. The multitask decomposition strategy of the cascaded network MC-CNN improves the recognition accuracy of the network. The cascaded network MC-CNN-P enhances the recognition accuracy by 0.88% and 2.7% compared with MC-CNN in two datasets, respectively. The cascaded network MC-CNN-F enhances the recognition accuracy by 2.22% and 4.15% compared with MC-CNN in two datasets, respectively. The adaptive learning strategy for the negative values of the activation function PReLU and the hard-to-detect sample reinforcement learning strategy of the Focal loss function both improve the quality of the cascaded network target recognition. MC-CNN-NT gets 91.27% recognition accuracy on the SYIT-Vehicle dataset. The increases of MC-CNN-NT accuracy are 11.45%, 9.38%, 6.31%, 7.03%, 6.15%, and 4.81% than Faster R-CNN, CNN + CNN, IFR-CNN + CNN, MC-CNN, MC-CNN-P, and MC-CNN-F, respectively. MC-CNN-NT gets 82.63% recognition accuracy on the BIT-Vehicle dataset. The increases of MC-CNN-NT accuracy are 13.17%, 11.19%, 6.99%, 8.3%, 5.6%, and 4.15% than Faster R-CNN, CNN + CNN, IFR-CNN + CNN, MC-CNN, MC-CNN-P, and MC-CNN-F, respectively. MC-CNN-NT that combines the PReLU function with the Focal loss function achieves the best performance of target attribute recognition. Simultaneously, MC-CNN-NT achieves high accuracy in the migration datasets, which verifies that the model has good robustness.

4.3. Experiments of Vehicle Attributes Recognition for the Small Targets

4.3.1. Datasets. The COCO-Vehicle dataset is used to carry out the experiments of vehicle attribute recognition for the small targets. The COCO-Vehicle dataset contains seven category targets: person, bus, microbus, minivan, sedan, SUV, and truck. The dataset is manually annotated according to the file of the COCO dataset [32]. The person, car, and bus in the COCO dataset are extracted, and the car in COCO dataset is subdivided into sedan, minivan, microbus, SUV, and truck. Since the number of trucks in the COCO dataset is few, 300 trucks from the VOC2007 dataset are selected to supplement them. The sample images of the COCO-Vehicle dataset are shown in Figure 7.

In this paper, the definition of small target, medium target, and large target is based on the standard of reference [33]: P_{pix} represents the percentage of ROIS pixels in the whole image. The targets with $P_{pix} \le 2.4\%$ are defined as the small targets. The targets with $2.4\% \le P_{pix} \le 47.2\%$ are defined as the medium targets. The targets with $P_{pix} \ge 47.2\%$ are defined as the large targets. In the COCO-Vehicle dataset, the number of small targets accounts for 63.10%, the number of medium targets accounts for 32.86%, and the number of large targets accounts for 4.04%. The statistical results of the COCO-Vehicle dataset are shown in Table 7.

4.3.2. Experiments of the Location for the Small Targets. In order to verify the small target location performance of MC-CNN-ST, the contrast experiments of six networks are designed in this paper. The main structures of the six networks are shown in Table 8. Net-A network is an original Faster R-CNN Network. Net-B network is based on the original Faster R-CNN, which compresses the number of convolution layers in the feature extraction network to 6. Net-C network and Net-D network fuse the second, fourth, and sixth convolution layers of VGG-E. Net-C network adds a 5×5 convolution layer to the three branches of the fusion network. Net-D network adds two 3×3 convolution layers

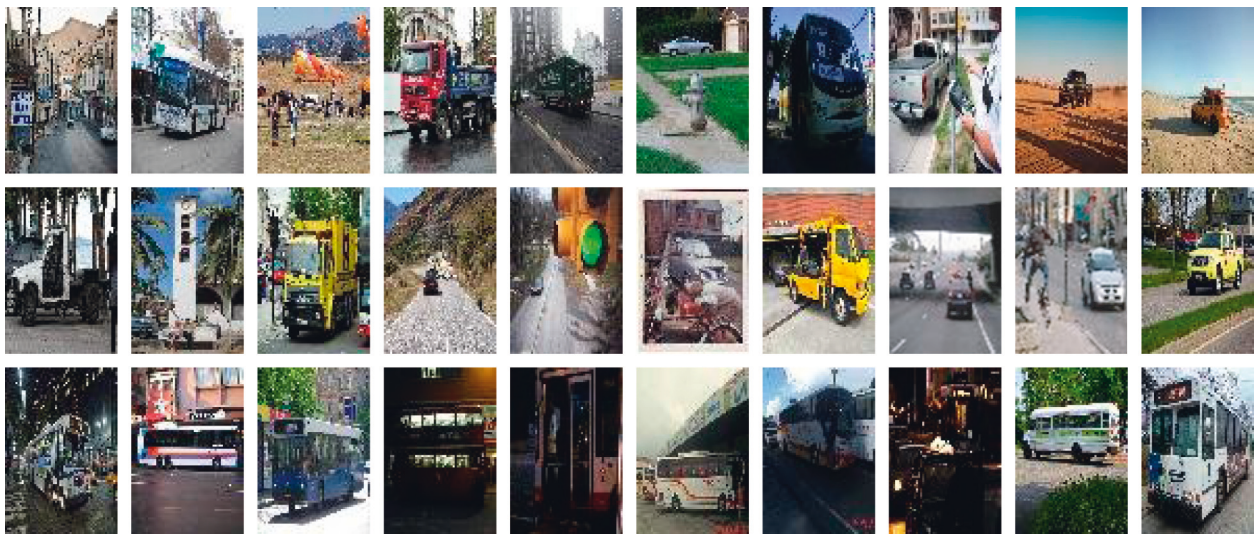


Figure 7: The samples of the COCO-Vehicle dataset.

Table 7: The sample statistics of the COCO-Vehicle datasets.

Target type	Person	Bus	Microbus	Minivan	Sedan	SUV	Truck	Total
Small	5083	240	356	261	3249	900	40	10129
Middle	1783	796	176	235	1430	483	372	5275
Big	16	240	9	38	273	40	32	648
Total	6882	1276	541	534	4952	1423	444	16052

Table 8: The structures of six location networks.

Method	Description of comparative experiments
Net-A	The original faster R-CNN
Net-B	Faster R-CNN + structure compression
Net-C	Structure compression + the fusion of 2, 4, 6 feature map + using one 5×5
Net-D	Structure compression + the fusion of 2, 4, 6 feature map + using two 3×3
Net-E	Structure compression + the fusion of 1, 3, 6 feature map + using one 5×5
Net-F	Structure compression + the fusion of 1, 3, 6 feature map + using two 3×3

to the three branches of the fusion network. Net-E network and Net-F network fuse the first, third, and sixth convolution layers of VGG-E. Net-F network is the location subnetwork of the cascaded network MC-CNN-ST. Different from Net-F network, Net-E network adds a 5×5 convolution layer to the three branches of the network.

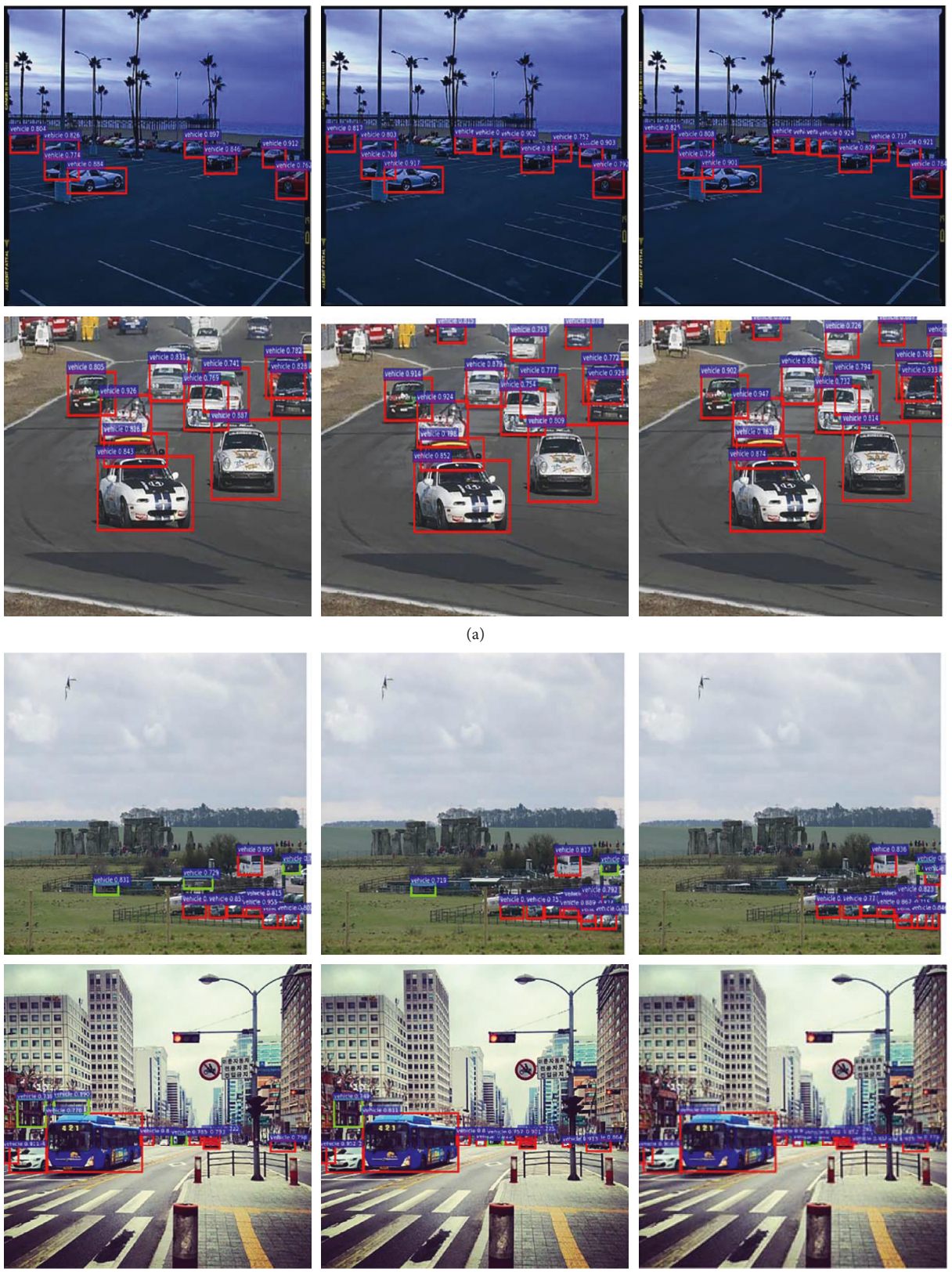
The location results of six networks are shown in Table 9 and Figure 8. As shown in Table 9, the MC-CNN-ST location network (Net-F) using the network compression strategy has better recognition ability for the background region of the suspected targets. The number of false acceptances is reduced. The increase of Net-F network recognition precision is 21.23%, 8.55%, 5.03%, 2.42%, and 2.7% more than Net-A, Net-B, Net-C, Net-D, and Net-E, respectively. Net-F network using the network fusion strategy enhances the expressive ability for the target features and reduces the

Table 9: The location results for the small targets on the COCO-Vehicle dataset.

Method	Vehicle	TP	FP	FN	Precision (%)	Recall (%)
Net-A	1913	1237	546	676	69.38	64.66
Net-B	1913	1459	319	454	82.06	76.27
Net-C	1913	1555	262	358	85.58	81.29
Net-D	1913	1576	211	337	88.19	82.38
Net-E	1913	1614	222	299	87.91	84.37
Net-F	1913	1651	171	262	90.61	86.30

number of false rejection. The increase of the recall rate of Net-F network is 21.64%, 10.03%, 5.01%, 3.92%, and 1.93% than Net-A, Net-B, Net-C, Net-D, and Net-E, respectively.

The location results of Net-A, Net-B, and Net-F correspond to the images of the first, second, and third columns of Figures 8(a) and 8(b), respectively. As shown in Figure 8(a), Net-A network recognizes 8 and 10 targets in two different images, respectively. Net-B network recognizes 11 and 13 targets in two images, respectively. Net-F network recognizes 12 and 14 targets in two images, respectively. Net-F network recognizes the largest number of the targets. Longitudinal network compression strategy and horizontal network fuse strategy of Net-F network reduce the number of false rejection in the model and improve the recall rate of the network. In Figure 8(b), the vehicles marked with the red borders are correctly identified as the foreground. Buildings, billboards, and other backgrounds are misidentified as vehicles marked with the green borders. As shown in Figure 8(b), the number of false detections in Net-F network



(a)

(b)

Figure 8: The results of the comparative experiments. (a) The reduction of false negative. (b) The reduction of false positive.

Table 10: The attribute recognition results for the small targets on the COCO-Vehicle dataset.

Method	Bus (%)	Microbus (%)	Minivan (%)	Sedan (%)	SUV (%)	Truck (%)	Total (%)
CNN	65.25	37.70	38.36	43.93	38.36	51.14	45.79
Fast R-CNN	70.50	49.18	53.77	44.59	54.09	57.70	54.97
Faster R-CNN	74.10	53.12	59.67	51.15	55.41	60.62	59.34
CNN + CNN	74.99	54.89	59.80	52.53	56.99	61.39	60.10
IFR-CNN + CNN	75.70	55.12	59.97	53.45	57.41	62.51	60.71
MC-CNN-NT	77.70	57.04	60.00	54.75	58.36	62.62	61.75
MC-CNN-ST	88.27	79.46	75.67	75.10	77.52	79.95	79.33

Figure 9: The classification results for the small targets using MC-CNN-ST.

is 1, which is less than the number of false detections in Net-A and Net-B.

4.3.3. Experiments of Attribute Recognition for the Small Targets. In order to verify the small target recognition performance of MC-CNN-ST, the contrast experiments of seven networks are designed in this paper. These seven networks include three noncascaded networks (CNN, Fast R-CNN, and Faster R-CNN) and four cascaded networks (CNN + CNN, IFR-CNN + CNN, MC-CNN-NT, and MC-CNN-ST). The results of attribute recognition are demonstrated in Table 10 and Figure 9.

As shown in Table 10, MC-CNN-ST achieves the superior recognition results in seven networks. The increases of the accuracy of MC-CNN-ST are 33.54%, 24.36%, 19.99%, 19.23%, 18.62%, and 17.58 than CNN, Fast R-CNN, Faster R-CNN, CNN + CNN, IFR-CNN + CNN, and MC-CNN-NT, respectively. As shown in Figure 9, MC-CNN-ST has good attribute recognition ability for six category vehicles.

5. Conclusions

This paper is devoted to solving the problem of vehicle attribute recognition. The multitask cascaded networks MC-CNN-NT and MC-CNN-ST are established to recognize vehicle attributes with normal size and small size, respectively. The cascaded multitask networks improve the recognition effect of one-stage networks in the complex background. The implementation of the network compression strategy and the feature fusion strategy reduces the false acceptance rate and improves the recall rate for the small targets. The use of the activation function PReLU and the loss function Focal Loss improves the nonlinear mapping ability of the networks and the mining ability for the hard-to-detect samples. The experimental results show that the

increase of the recognition accuracy of MC-CNN-NT for the normal targets is 18.3%, 11.45%, 9.38%, 6.31%, and 7.03% more than Fast R-CNN, Faster R-CNN, CNN + CNN, IFR-CNN + CNN, and MC-CNN, respectively. The increase of the recognition accuracy of MC-CNN-ST for the small targets is 24.36%, 19.99%, 19.23%, 18.62%, and 17.58% than Fast R-CNN, Faster R-CNN, CNN + CNN, IFR-CNN + CNN, and MC-CNN-NT, respectively. In the future research, we consider fusing the infrared image features with the visible image features to enhance the recognition accuracy for the small target vehicles.

Data Availability

The three datasets (SYIT-Vehicle dataset, BIT-Vehicle dataset, and COCO-Vehicle dataset) used in the paper can be obtained through e-mail (liufang5208@sylu.edu.cn and xuke@sylu.edu.cn).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Open Foundation of Science and Technology on Electro-Optical Information Security Control Laboratory (Grant no. 61421070104), Science and Technology Project of Educational Department of Liaoning Province (Grant no. LG201715), Natural Science Foundation of Liaoning Province (Grant no. 20170540790), and (Grant no. 2019-ZD-0256).

References

[1] Q. W. Li, H. S. Cheng, Y. Zhou, and G. Y. Huo, "Road vehicle monitoring system based on intelligent visual internet of things," *Journal of Sensors*, vol. 2015, Article ID 720308, 16 pages, 2015.

[2] Q. W. Xue, K. Wang, J. J. Lu, and Y. J. Liu, "Rapid driving style recognition in car-following using machine learning and vehicle trajectory data," *Journal of Advanced Transportation*, vol. 2019, Article ID 9085238, 11 pages, 2019.

[3] B. Hu, J.-H. Lai, and C.-C. Guo, "Location-aware fine-grained vehicle type recognition using multi-task deep networks," *Neurocomputing*, vol. 243, pp. 60–68, 2017.

[4] R. Feris, B. Siddiquie, Y. Zhai, J. Petterson, L. Brown, and S. Pankanti, "Attribute-based Vehicle Search in Crowded Surveillance videos," in *Proceedings of the 1st International Conference on Multimedia Retrieval*, Trento, Italy, April 2011.

[5] J. Prokaj and G. Medioni, "3-D model based vehicle recognition," in *Proceedings of the 2009 IEEE Workshop on Applications of Computer Vision (WACV)*, pp. 1–7, Salt Lake County, UT, USA, January 2009.

[6] K. Ramnath, S. N. Sinha, R. Szeliski, and E. Hsiao, "Car make and model recognition using 3D curve alignment," in *in Proceedings of the 2014 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 285–292, Steamboat Springs, CO, USA, March 2014.

[7] T. N. Tan and K. D. Baker, "Efficient image gradient based vehicle localization," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1343–1356, 2000.

[8] P. Negri, X. Clady, S. M. Hanif, and L. Prevost, "A cascade of boosted generative and discriminative classifiers for vehicle detection," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, Article ID 782432, 2008.

[9] X. X. Ma and W. E. L. Grimson, "Edge-based rich representation for vehicle classification," in *Proceedings of the Tenth IEEE International Conference on Computer Vision*, pp. 1185–1192, Beijing, China, October 2005.

[10] S. S. Teoh and T. Bräunl, "Symmetry-based monocular vehicle detection system," *Machine Vision and Applications*, vol. 23, no. 5, pp. 831–842, 2012.

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

[12] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Santiago, Chile, December 2015.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 1137–1149, Istanbul, Turkey, 2015.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2014.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, CVPR, Las Vegas, NV, USA, June 2016.

[16] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multiBox detector," in *Proceedings of the 14th European Conference on Computer Vision*, pp. 21–37, Amsterdam, Netherlands, October 2016.

[17] Z. Q. Huo, Y. Z. Xia, and B. L. Zhang, "Vehicle type classification and attribute prediction using multi-task rcnn," in *Proceedings of the International Congress on Image & Signal Processing*, pp. 564–569, IEEE, Shanghai, China, October 2017.

[18] T. Y. Tang, S. L. Zhou, Z. P. Deng, H. X. Zou, and L. Lei, "Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining," *Sensors*, vol. 17, no. 2, p. 336, 2017.

[19] Q. Meng, H. S. Song, G. Li, Y. A. Zhang, and X. Q. Zhang, "A block object detection method based on feature fusion networks for autonomous vehicles," *Complexity*, vol. 2019, Article ID 4042624, 14 pages, 2019.

[20] Z. Wang, P. Chen, and J. X. Pan, "Target detection in complex background based on depth learning," *Journal of Chongqing University of Technology: Natural Science*, vol. 32, no. 4, pp. 171–176, 2018.

[21] H. Gong, Y. Zhang, F. Liu, and K. Xu, "Vehicle recognition using multi-task cascaded network," in *Proceedings of Fifth Symposium on Novel Optoelectronic Detection Technology and Application*, pp. 1–8, Xian, China, October 2018.

[22] P. Fu and S. P. Xie, "License plate location based on cascaded convolution neural network," *Computer Technology and Development*, vol. 28, no. 1, pp. 1362–1451, 2018.

[23] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the 2017 IEEE Conference Computer Vision*

and Pattern Recognition (CVPR), pp. 936–944, Honolulu, HI, USA, July 2017.

[24] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Conference Computer Vision and Pattern Recognition (CVPR)*, no. 8, pp. 886–893, Diego, CA, USA, 2005.

[25] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[26] S. Karungaru, M. Fukumi, and N. Akamatsu, "Human face detection in visual scenes using neural networks," *IEEJ Transactions on Electronics Information and Systems*, vol. 122, no. 6, pp. 995–1000, 2008.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on ImageNet classification," in *Proceedings of the 2015 IEEE International Conference on Computer Vision*, pp. 1026–1034, Santiago, Chile, December 2015.

[28] F. Agostinelli, M. Hoffman, P. Sadowski, and P. Baldi, "Learning activation functions to improve deep neural networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Banff, Canada, April 2014.

[29] V. Nair and G. E. Hintion, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML)*, pp. 807–814, Haifa, Israel, June 2010.

[30] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587, Colorado Springs, CO, USA, June 2014.

[31] Z. Dong, Y. Wu, M. Pei, and Y. Jia, "Vehicle type classification using a semisupervised convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2247–2256, 2015.

[32] T. Y. Lin, M. Maire, S. Belongie et al., "Microsoft coco: common objects in context," in *Proceedings of the 13th European Conference On Computer Vision*, pp. 740–755, Zurich, Switzerland, September 2014.

[33] B. Singh and L. S. Davis, "An analysis of scale invariance in object detection – SNIP," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3578–3587, Salt Lake City, USA, June 2018.