Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 5590514, 7 pages
https://doi.org/10.1155/2022/5590514

Research Article

Multitarget Detection in Depth-Perception Traffic Scenarios

Qiao Peng[1] and Dengyin Zhang[2]

[1]*College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*
[2]*College of Internet of Things, Nanjing University of Posts and Telecommunications, Jiangsu Key Laboratory of Broadband Wireless Communication and Internet of Things, Nanjing 210003, China*

Correspondence should be addressed to Dengyin Zhang; zhangdy@njupt.edu.cn

Received 15 February 2021; Revised 2 December 2021; Accepted 6 December 2021; Published 4 February 2022

Academic Editor: Nianyin Zeng

Multitarget detection in complex traffic scenarios usually has many problems: missed detection of targets, difficult detection of small targets, etc. In order to solve these problems, this paper proposes a two-step detection model of depth-perception traffic scenarios to improve detection accuracy, mainly for three categories of frequently occurring targets: vehicles, person, and traffic signs. The first step is to use the optimized convolutional neural network (CNN) model to identify the existence of small targets, positioning them with candidate box. The second step is to obtain classification, location, and pixel-level segmentation of multitarget by using mask R-CNN based on the results of the first step. Without significantly reducing the detection speed, the two-step detection model can effectively improve the detection accuracy of complex traffic scenes containing multiple targets, especially small targets. In the actual testing dataset, compared with mask R-CNN, the mean average detection accuracy of multiple targets increased by 4.01% and the average precision of small targets has increased by 5.8%.

1. Introduction

The trend of artificial intelligence has appeared in many fields of scientific research and industrial production. As an application of artificial intelligence, autonomous driving has attracted extensive attention in the industry since the first prototype of autonomous driving demonstrated by Google in 2014. Multitarget detection technology is an important foundation of autonomous driving technology; there have been many studies and improvements for detecting specific targets [1]. However, actual traffic scenarios usually have more complex backgrounds than ordinary scenarios and are often affected by light, occlusion, and weather. Therefore, when the scenarios appear in various categories of targets, such as vehicles [2], persons, and traffic signs [3], it is easy to miss the detection of small targets, which makes it difficult to detect multitarget completely. The missed detection of multitarget will have a huge impact on the next step of autonomous driving, assistance driving [4] and the judgment of actual traffic conditions. Therefore, due to the high requirements of intelligent driving for safety, how to improve the accuracy of multitarget detection and reduce the detection time and solve the fatal problem of missed target detection in multitarget detection seems vital right now.

Traditional target detection methods mainly include frame difference method, background difference method, optical flow method, HOG [5], and SIFT [6]. The main steps generally include extracting target features, training corresponding classifiers, sliding window search, repetition, and false positives filtering. With the development of technology, convolutional neural network (CNN) [7] based on deep learning was proposed. Deep learning is widely used in many fields [8], we propose to apply CNN to avoid complex feature extraction and data reconstruction process in the traditional recognition algorithm, which is a robust and effective method for common detection tasks. According to the algorithm implementation steps classification, the algorithms based on deep learning can also be divided into "one-stage" target detection algorithm and "two-stage" target detection algorithm, mainly as follows: (1) "one-stage" detection

methods are represented by you only look once (YOLO) [9] based on regression, mainly including YOLOv2 [10], YOLOv3 [11], and SSD [12]. They solve the detection task as a regression problem, that is, to directly predict the location, category, and corresponding confidence of each class in real-time detection. This kind of algorithm detects faster, but the precision is not ideal for small targets detection. (2) "Two-stage" detection `methods are represented by regional-CNN (R-CNN) [13] series based on region. They are generally implemented in two stages: producing some potential bounding box, then deploying a classifier to judge the right one according to its probability. Later, based on the R-CNN, other algorithms were improved and extended as fast region-based convolutional network (Fast R-CNN) [14], faster R-CNN [15], and mask R-CNN [16].

In summary, due to the low proportion of small targets in an image (usually less than 1%) and poor feature representation limited to size in the complex traffic scenarios, multitarget detection also faces huge challenges. To solve the problems, this paper proposes a two-step detection model based on deep learning for detecting multitarget. For three categories of frequently-occurring targets: traffic signs, vehicles, and persons in the complex traffic scenarios, the first step is to use the optimized CNN model to identify the location of the small targets and mark them with a candidate box. The second step is to obtain multitarget classification, location, and pixel-level segmentation by using mask R-CNN based on the results of the first step. The two-step detection model can effectively improve the overall target recognition algorithm accuracy and build a multitarget detection system. Thus, the difficulty of traffic scenarios information processing and deep learning development can be largely solved.

2. Materials and Methods

2.1. Traditional Work. Based on the consideration of improving detection accuracy, a "two-stage" target detection algorithm is generally implemented: producing some potential bounding box, then deploying a classifier to judge the right one according to its probability. Ross Girshick presented R-CNN [13] by using a deep convolutional network to classify object proposals. For the purpose of the consideration of computing time and space, he employed a region of interest (RoI) pooling layer on fast-RCNN [14] to improve speed and detection accuracy. Later on, a more efficient faster R-CNN [15] inherited from the above was proposed. It introduced a new region proposal network (RPN) directly to obtain candidate areas. Further, mask R-CNN [16] extends faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition [17]. It includes three parts: target positioning, target classification, and segmentation mask prediction, as shown in Figure 1. The structure of mask R-CNN actually includes feature extraction network, feature combination network, region proposal network, RoI align, and functional network. First of all, the feature extraction network constructs a feature pyramid network (FPN) [18] on the basis of residual network (ResNet) [19], performs feature extraction on input images, and generates feature map by CNN. Using FPN + ResNet101 to extract images with low-dimensional semantic features is better than a single feature structure like ResNet. Secondly, the RPN chooses candidate targets in the feature map and Softmax classifier is used to distinguish the candidate target belongs to the background or foreground. Anchor technique is used to calibrate the image of the candidate box position, produce different scale anchor box, and reuse nonmaximum suppression to filter out accurate candidate box and generate candidate target area. Finally, a fully convolutional network (FCN) [20] is used to predict precise target segmentation under the pixel-level correspondingly. Unlike the original FCN, the branches of FCN in mask R-CNN are not classified here, while only distinguishing the front and rear scenes. In addition, mask R-CNN has to use RoI align to replace RoI pooling in faster R-CNN, introducing the bilinear interpolation process of pooling to replace the original RoI pooling quantitative twice and get the final coordinates of the target object. Pooling progress from discrete to continuous can make up for the rough quantitative problems of RoI pooling, effectively reduce the target object's jagged edges, and improve the positioning accuracy of the candidate box.

2.2. Improved Multitarget Detection Model. For traffic scenarios including multiple targets, there are many leak problems caused by mixed and disorderly multitarget, inadequate information characteristics of small targets. In order to solve these problems, this paper proposes a two-step detection model (see Figure 2). The first step is to use the optimized CNN model to identify the location of the small targets according to the pixel definition and mark it with a candidate box. The second step is to obtain multitarget classification, positioning, and pixel-level segmentation by using mask R-CNN based on the results of the first step, mainly for three frequently-occurring targets: vehicles, persons, and traffic signs. The proposed two-step detection model and pixel-level segmentation can effectively improve the recognition accuracy and enhance the representativeness of traffic signs. For the different sizes of targets, we distinguish them according to the pixel definition. Here, we stipulate $\text{pixel}^2 < 32^2$ as small targets, $32^2 < \text{pixel}^2 < 96^2$ as medium targets, and $\text{pixel}^2 > 96^2$ as large targets.

The first step is the optimized CNN network. Among them, the network's overall architecture consists of a feature extraction network and a small target rectangular box detection network. The main components of the optimized CNN model are shown in Figure 3.

Here, the feature extraction network is an optimization network that integrates different convolutional layers, max-pooling layers, and local batch normalization (LBN) layers. As shown in Table 1, the optimized CNN network gradually deepens from Conv1, and parameter increments are carried out for the representation of small targets. The Conv1 firstly runs a large kernel (11×11) in the input images to preserve the low level but rich detail. Then the obtained features are transmitted to two 3×3 convolutional layers (Conv2 and

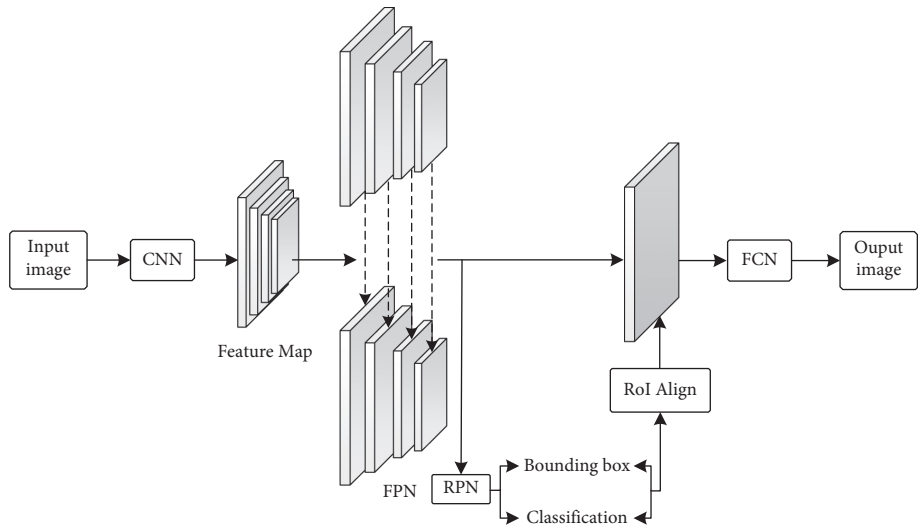

Figure 1: The network architecture diagram of mask R-CNN.

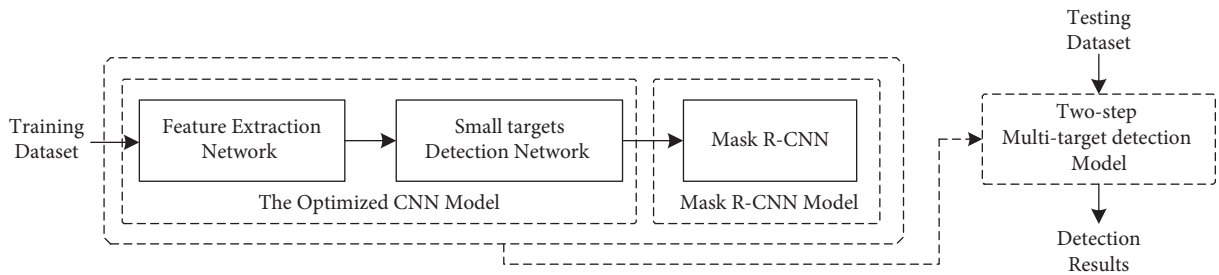

Figure 2: Two-step multitarget detection model.

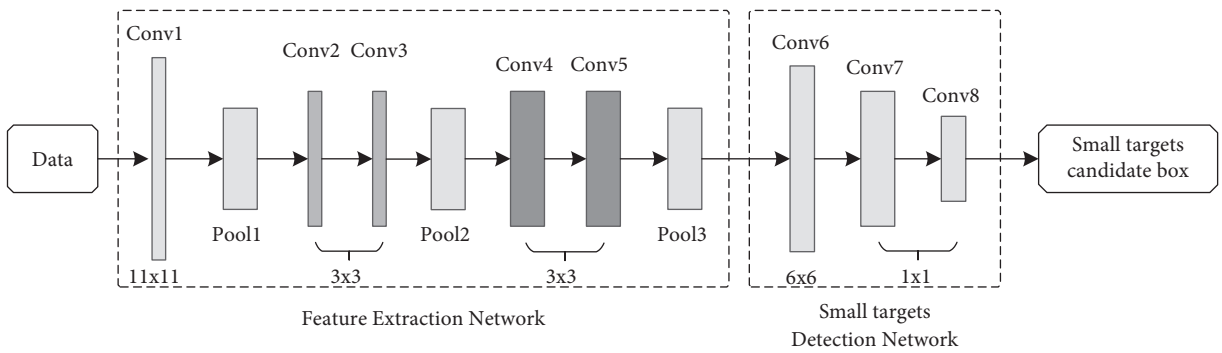

Figure 3: Main building blocks of the optimized CNN model.

Conv3), and the two convolutional layers of Conv2 and Conv3 are connected sequentially to replace the 5×5 kernel in VGGNet [21]. This decomposition has two advantages: (1) by activating multiple convolutional layers of the nonlinear function, it is helpful to enhance the nonlinear expansion capability and extract more in-depth and better features than a single layer; (2) it can reduce the calculation of parameters. Assuming that the input channel and output channel convolutional layer, C and D, is a 5×5 kernel, respectively. Hence, the parameters will reach $5 \times 5 \times C \times D = 25 \times C \times D$ with 5×5 kernel, while only $2 \times (3 \times 3 \times C \times D) = 18 \times C \times D$ when combined with two 3×3 convolutional layers instead. Obviously, the reduction of parameters is $25/18 = 1.4$ times. Visually, the convolution decomposition method introduces fewer parameters, is easy to overfit and more powerful features are expressed. Then, the LBN layer, ReLU activation layer, and max-pooling layer are adopted to simplify the network computational complexity, compress the input feature map, and achieve better feature extraction.

The small target rectangular box detection network inputs the feature map obtained by the feature extraction network to obtain the minimum rectangular box. The full

Table 1: The structure and parameters of the optimized CNN model.

Layer name	Conv1	Conv2	Conv3	Conv4	Conv5	Conv6	Conv7	Conv8
Input size(c, *h*, w)	3,1280,1280	96,159,159	256,159,159	256,79,79	384,79,79	384,39,39	4096,40,40	4096,40,40
Output size (c, *h*, w)	96,318,318	256,159,159	256,159,159	384,79,79	384,79,79	4096,40,40	4096,40,40	256,40,40
Kernel(size, stride, pad)	11,4,0	**3,1,1**	**3,1,1**	**3,1,1**	**3,1,1**	6,1,3	**1,1,0**	**1,1,0**
Pooling(size, stride)	3,2		3,2		3,2			
Remarks	LBN layer		LBN layer			Dropout0.5	Dropout0.5	

connection layer is commonly used in traditional networks, such as VGGNet, which is replaced by two convolutional layers (Conv7 and Conv8) here, as shown in Table 1. These two convolutional layers adopt a 1×1 convolution kernel. It can predict object boundaries and output the differential value between the predicted bounding boxes and ground truth. The core of our approach is to adopt a 1×1 convolutional kernel replacing a common fully-connected layer. As the convolution kernels own the local receptive domain, they can slide across a larger input image and obtain multiple outputs regardless of the size of the input images, thus improving the efficiency of neural network forward propagation, enhancing the learning ability of CNN, and reducing the amount of computation. Through this detection network, the input feature map can output multiple positioning candidate boxes for small targets. We save them as a new dataset.

The second step is mask R-CNN model. We input the new dataset with the positioning rectangle box of small targets into the mask R-CNN model. FPN + RestNet101 [18, 19] is used for feature extraction. RPN selects candidate target in the feature map and uses Softmax classifier to distinguish candidate target belongs to the background or foreground. The anchor technique is used to calibrate the image of the candidate box position and produce different scale anchor boxes. Then we reuse NMS to filter out accurate candidate box. Combined with the optimized CNN network model generated by the small target rectangle area, the candidate target area of multi-target can be generated. Here, FCN [20] in mask R-CNN is used to distinguish front and rear scenes, predict corresponding targets, and perform pixel-level target segmentation, which improves the accuracy of small target's location and multitarget detection. Finally, the two-step detection model is to realize our multitarget detection and pixel-level segmentation.

2.3. Loss Function. In the first step of this paper, the optimized CNN model uses the bounding box loss function, L_{reg} to locate small targets and mark them with candidate box.

$$L_{\text{reg}}(t_i, t_i^*) = \sum_{i\in\{x,y,w,h\}} \text{smooth}_{L1}(t_i - t_i^*),$$
$$\text{smooth}_{L1}(x) = \begin{cases} 0.5x^2, & |x| < 1, \\ |x| - 0.5, & \text{otherwise}, \end{cases} \tag{1}$$

where i is the index of the anchor point, t_i is a vector representing the four parameterized coordinates, x, y, w, and h of the predicted bounding box, and t_i^* is that of the ground-truth box associated with a positive anchor.

The second step, the mask R-CNN model uses the multitask loss function for training. Due to the introduction of mask, the loss function is composed of the classification loss function, bounding box loss function, and mask loss function:

$$L(\{p_i\},\{t_i\},\{m_i\}) = \frac{1}{N_{\text{cls}}}\sum_i L_{\text{cls}}(p_i, p_i^*) + \lambda\frac{1}{N_{\text{reg}}}\sum_i p_i^* L_{\text{reg}}(t_i, t_i^*) + \frac{1}{N_{\text{mask}}}\sum_i L_{\text{mask}}(m_i), \tag{2}$$

where λ is the balance weight, p_i represents the predicted probability that anchor i is recognized as the target. If the intersection over union (IoU) value is not less than 0.5, the anchor i could be marked as a positive sample. Otherwise, it is labeled as a negative sample. The ground-truth label p_i^* is the nominal value, a function of 0 and 1. It is 1 if the anchor is positive and is 0 if the anchor is negative. N_{cls} is the total number of anchor points, N_{reg} is the number of positive samples, N_{mask} is the number of the mask, m_i represents the confidence with which the object is predicted to be the target, and m_i^* represents the output after the sigmoid function passes through the first mask layer pixel by pixel.

L_{cls} is log loss over two classes (object vs not object); L_{mask} is mask loss:

$$L_{\text{cls}}(p_i, p_i^*) = -\log[p_i p_i^* + (1 - p_i^*)(1 - p_i)], \tag{3}$$

$$L_{\text{mask}}(m_i) = -[m_i \log m_i^* + (1 - m_i)\log(1 - m_i^*)]. \tag{4}$$

3. Results and Discussion

3.1. Training Settings. The experiment is implemented in an Ubuntu 16.04.5 system with NVIDIA TITAN Xp, 12 GB GDDR5. CPU we utilized is Intel Xeon E5-1620 v4 and memory is 64 GB DDR4. This convolutional neural network is based on the Caffe platform. The training was conducted using the deep learning framework *PyTorch* 1.1 and *Python* 3.6.

Table 2: Comparison of the algorithms.

Method	AP/%			mAP (%)	FPS/(frame/s)
	Vehicles	Persons	Signs		
YOLOv3	56.33	51.74	48.28	52.12	56.74
Faster R-CNN	80.29	67.81	66.90	71.67	26.15
Mask R-CNN	83.95	73.56	69.51	75.67	23.27
Our approach	**85.62**	**78.12**	**75.31**	**79.68**	**17.58**

Figure 4: Partial detection result.

We select 3000 higher resolution images from the Microsoft COCO dataset [22] and save 3 classes: vehicles, persons, and traffic signs among the multiple kinds of labels. Preprocessing methods such as rotation, scaling, and little distortion are utilized to strengthen the data as our experiment dataset. Among 3000 images, 2500 of them were used to train the model, and 500 were used to test the model in this paper. The experiment utilizes the multitask loss function for training, including L_{reg}, L_{cls}, and L_{mask}. Referred on the configuration of faster R-CNN [15], the RoI is considered positive if it has IoU with a ground-truth box of at least 0.5 and negative otherwise. The mask loss L_{mask} is defined only on positive RoIs. We follow the previous steps to train alternately until the network converges.

The evaluation metrics of detection performance are the same as the Microsoft COCO [22] benchmark. We adopt average precision (AP), mean average precision (mAP), and frame per second (FPS) to conduct a quantitative evaluation on the detection results of vehicles, persons, and traffic signs. AP represents the average accuracy of the single-class target, mAP represents the overall situation of achieving correct positioning across the testing dataset, and FPS represents the speed and frame rate of the algorithm.

3.2. Performance Comparisons and Analysis. To comprehensively evaluate the multitarget detection performance of our framework, we use the same testing dataset to experiment under different models. For three types of frequently-occurring targets: vehicles, persons, and traffic signs, we calculate AP, mAP, and FPS of the experiment. Table 2 provides the comparison of our results in the same testing dataset with YOLOv3 [11], faster R-CNN [15], and mask R-CNN [16].

As can be seen from Table 2, in the complex traffic scenarios containing multiple targets, the average detection accuracy of the two-step detection model has been improved overall, especially the average detection accuracy of small targets. Compared to mask R-CNN, the mean average detection accuracy of multiple targets increased by 4.01%. For three types of frequently-occurring targets, the average precision of vehicles has increased by 1.67%, the average precision of a person has increased by 4.56%, and the average precision of small traffic signs has significantly increased by 5.8%. Without significantly reducing the detection speed, the two-step detection model can effectively improve the average detection accuracy of complex traffic scenes containing multiple targets, especially small targets.

Figure 4 is a partial detection effect diagram of the algorithm. In the first step, after the optimized CNN model, the candidate frame of the small target (traffic sign) can be obtained, and then the accurate position of all the instance targets can be obtained through the mask R-CNN model. The network layer of the optimized CNN model here is deepened step by step. Through the decomposition of the convolutional layer, rich information is learned from the fine-grained details of the bottom layer, and the dimensionality and downsampling of the feature map are reasonably increased to enrich the small objects. This indicates that the detection part is trained to make full use of the convolutional layer to activate multiple convolutional layers of nonlinear functions, which helps to enhance the expansion ability of nonlinear feature information, and can obtain deeper and better features than a single layer extraction, and then can classify and detect small traffic signs well; then input into the mask R-CNN model to obtain the location and classification of all the targets (vehicles, pedestrians, and traffic signs) contained in the picture. It can be seen from Figure 4 that the image will be occluded by the target and the background is dim, but this two-step model can effectively detect them.

Because the first step of the optimized CNN detection model has a prepositioning for small targets, then detection and pixel-level segmentation are effectively performed in the detection process of the second step, which usefully avoids the problem of missed detection of small targets. In addition, pixel-level segmentation by using mask R-CNN effectively improves the detection accuracy. Building a multitarget detection system can have good robustness to complex traffic scenarios. It may further solve the difficulties of information processing and deep learning development in the traffic scenes.

4. Conclusions

In this paper, we propose a two-step detection model based on deep learning to detect multitarget contained in complex traffic scenes. Small targets can be predetected through the optimized CNN model in the first step, and then input into the mask R-CNN model for classification, location, and pixel-level segmentation. Experimental results show that the model proposed in this paper has greatly improved the detection accuracy of multitarget in complex traffic scenes, has good robustness to scenes of different target sizes, and significantly improves the performance of target missing detection. For future work, we want to further improve the speed of multitarget detection by algorithm improvement that combines the power of deep CNN and traditional computer vision methods in complex traffic scenarios.

Data Availability

All data generated or analyzed during this study are included in this article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors thank the MDPI English Editing Team. This work was financially supported by National Natural Science Foundation of China (No. 61872423), Industry Prospective Primary Research & Development Plan of Jiangsu Province (No. BE2017111), the Scientific Research Foundation of the Higher Education Institutions of Jiangsu Province (No. 19KJA180006), and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (No. KYCX18_0912).

References

[1] X. Chen, K. Kundu, Y. Zhu, G. Berneshawi, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," *News in Physiological Sciences*, pp. 424–432, 2015.

[2] Z. Gao, H. Ji, T. Mei, B. Ramesh, and X. Liu, "Eovnet: earth-observation image-based vehicle detection network," *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 9, pp. 3552–3561, 2019.

[3] C. Wang, "Research and application of traffic sign detection and recognition based on deep learning," in *Proceedings of the International Conference on Robots & Intelligent System*, May 2018.

[4] J. Wei, J. He, Y. Zhou, K. Chen, Z. Tang, and Z. Xiong, "Enhanced object detection with deep convolutional neural networks for advanced driving assistance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1572–1583, 2020.

[5] N. Laopracha and K. Sunat, "Comparative study of computational time that HOG-based features used for vehicle detection," in *Proceedings of the International Conference on Computing and Information Technology*, pp. 275–284, Bangkok, Thailand, July 2017.

[6] X. Gao, Y. Wu, K. Yang, and J. Li, "Vehicle bottom anomaly detection algorithm based on SIFT," *Optik*, vol. 126, no. 23, pp. 3562–3566, 2015.

[7] H.-C. Shin, H. R. Roth, M. Gao et al., "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[8] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.

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

[10] J. Redmon and A. Farhadi, "YOLO9000: better. faster, stronger," in *Proceedings of the 2017IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525, Honolulu, HI, USA, July 2017.

[11] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," in *Proceedings of the 2018IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 2767–2773, Salt Lake, UT, USA, June 2018.

[12] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: single shot multibox detector," in *Proceedings of the European conference on computer vision*, pp. 21–37, Amsterdam, The Netherlands, October 2016.

[13] Z. Zhigang, L. Huan, D. Pengcheng, Z. Guangbing, W. Nan, and Z. Wei-Kun, "Vehicle target detection based on R-FCN," in *Proceedings of the 2018 Chinese Control And Decision Conference (CCDC)*, pp. 5739–5743, Shenyang, China, June 2018.

[14] R. Girshick, "Fast R.-CNN," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, Santiago, Chile, December 2015.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R.-Cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2015.

[16] E. J. Piedad, T.-T. Le, K. Aying, F. K. Pama, and I. Tabale, "Vehicle count system based on time interval image capture method and deep learning mask R-CNN," in *Proceedings of the TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, pp. 2675–2679, Kochi, India, October 2019.

[17] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, "Instance-sensitive fully convolutional networks," in *Proceedings of the European Conference on Computer Vision ECCV*, Amsterdam, The Netherlands, October 2016.

[18] Y. Zhao, R. Han, and Y. Rao, "A new feature pyramid network for object detection," in *Proceedings of the 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, pp. 428–431, Jishou, China, September 2019.

[19] K. Zhang, L. Guo, and C. Gao, "Optimization method of residual networks of residual networks for image classification," in *Proceedings of the 2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 321–325, Shanghai, China, January 2018.

[20] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of*

the International Conference on Learning Representations, San Diego, CA, USA, May 2015.

[22] J. Pont-Tuset and L. V. Gool, "Boosting object proposals: from pascal to COCO," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1546–1554, Santiago, Chile, December 2015.