Detection of Traffic Signs Based on Combination of GAN and Faster-RCNN

To cite this article: Wenqing Huang et al 2018 J. Phys.: Conf. Ser. 1069 012159

View the article online for updates and enhancements.
Detection of Traffic Signs Based on Combination of GAN and Faster-RCNN

Wenqing Huang¹, Mingzhu Huang¹ and Yuting Zhang¹

¹School of Information Science and Technology, Zhejiang Sci-Tech University,
No.928, No.2 Street, Xiasha High Education Park, Hangzhou City, Zhejiang Province,
China. Email: 851489278@qq.com

Abstract. Object detection has been widely researched and a series of algorithms have been proposed. These algorithms also have satisfactory verification results in various public datasets. However, the detection effect of small objects is still not satisfactory. Existing object detection algorithms usually achieve the detection of small objects by learning Multi-scale features, but this brings a large amount of calculation. In this paper, we solve the problem that is difficult to detect with small objects by mapping the non-obvious features of the small object into large object Features with the same feature distribution. Then the super-resolution feature obtained by mapping can be used to significant improve the detection performance in the process of training small objects detection.

1. Introduction

With the development of urbanization, traffic is getting more and more crowded and traffic accidents are frequent. Therefore, driverless based on computer vision will be an important measure to solve the problem. The detection of traffic signs will be one of the keys to this research. As the traffic signs in the road environment are generally small, it belongs to the category of small object detection.

Object detection needs to identify the object category and find the exact location of the object, which is more challenging than the classification problem. The small object has always been a difficult problem in object detection because the characteristics are not obvious. In the proposed algorithms including RCNN [1], Fast-RCNN [2] and Faster-RCNN [3], regional proposal methods are used to find candidate bounding boxes. Although the speed is indeed accelerated, real-time detection cannot be achieved, and it is more suitable for medium scale object detection. Of course, there are also some algorithms that have abandoned the process of regional proposals and have located objects directly on the whole feature map, such as You Only Look Once(YOLO) [4] and Single Shot MultiBox Detector(SSD) [5]. But they are only suitable for detecting images with high resolution and large objects, the effect of small object detection is usually not ideal. Many efforts [6, 7, 8] have attempted to improve the performance of small object detection through data augmentation or increased feature dimensions. However, these greatly reduce the detection speed, and there is no guarantee that the extracted features are sufficient to distinguish small object detections.

In this paper, we combine convolutional neural networks and Generative Adversarial Networks (GAN) [9] to achieve small object detection. For the problem that the small object feature is not obvious, we solve the small-object detection problem by the generative network to generate super-resolution features with the same feature distribution as the small object.
2. Related Work

2.1. Object Detection
Deep convolutional neural network (CNN) is widely applied in object detection. And it improves the detection accuracy enormously than the previous works such as HOG-based deformable part model (DPM) [10]. DPM proposes a direct optimization through the latent variable model and uses dynamic programming for rapid reasoning. However, before the RCNN was proposed, its performance was still unsatisfactory and reached a bottleneck. To address the location problem, RCNN relies on regional proposal technology (called selective search) to find bounding boxes containing high probability objects. Then a convolutional layer is used to gradually extract high-dimensional features for each candidate bounding box. Finally, the Support Vector Machines (SVM) is used as a classifier to determine the object class in the box. Obviously, RCNN has a lot of duplicate calculations because each candidate bounding box has some overlap. To alleviate this problem, Fast-RCNN introduces a region of interest (ROI) pooling layer to share the calculation of features. This network uses the entire image as input to extract global feature map and then the ROI pooling layer extracts a fixed length feature of each region proposal from them. Nevertheless, Fast-RCNN exploits selective search technology which limits the speed of detection to achieve further increase. Therefore, the Region Proposal Network (RPN) is proposed to further improve the detection speed. The corresponding method is called Faster-RCNN. Moreover, the bounding box regression is generally applied in detection method mentioned above to improve the detection accuracy.

In addition to the methods mentioned above, there is also a class of object detection methods that discards the regional proposal technology to find candidate boxes. The two representative algorithms are YOLO and SSD, which can quickly detect objects. YOLO directly predicts bounding box and class probability from the complete image. It uses global context information, rarely errors in the classification of background classes, and has good generalization ability. However, YOLO is very sensitive to the proportion of objects. It predicts the object candidate box by sliding on each unit. It can not increase the speed if the unit is set too small, and it is difficult to detect the object if the unit is set too large. To solve these weaknesses. SSD adds pooling operations after each layer of convolution so that feature maps are reduced layer by layer to form feature map pyramids. Then, the candidate boxes are extracted in the sliding cells of each level feature map, which not only improves the limitations of the YOLO candidate box selection, but also guarantees the speed. Based on the previous YOLO foundation, YOLOv2 [11] has also been improved to achieve higher detection accuracy and faster detection speeds.

2.2. Generative Adversarial Network
Generation of confrontation networks (GAN) is a powerful subclass of the generative model and has been successfully applied to image generation and editing, semi-supervised learning and domain adaptation [12, 13]. Various flavors of GANs have been recently proposed, both purely unsupervised [14~17] as well as conditional [18, 19].

In the GAN framework, the model learning a deterministic transformation \(G\) of a simple distribution \(P_z\), with the goal of matching the data distribution \(P_d\). This learning problem can be seen as a two-player game between generators and discriminator. Generator learning how to generate samples similar to real data, and discriminator learning how to distinguish real data from fake data. The goal of both parties is to minimize their own costs, and the solution to the game is the Nash equilibrium. Both parties cannot raise costs unilaterally [14].
3. Our Network

3.1. Overview

We embed the GAN model into a Faster-RCNN framework, as shown in figure 1. We add a generative network above the convolutional layer of the Faster-RCNN and change the fully connected layer to the upper and lower structure. The upper layer is the discriminator, which is used to distinguish whether the input features are true data or fake data. And the lower level is the detector, which is used to output the class and location of the object. The feature map generated by the generator will be superimposed with the feature map extracted by the following five-layer convolution and then sent to the discriminator for distinction. the lower-level feature maps pass through two cascaded convolutional layers of sizes 3 and 1, respectively, in order to match their size to the five-layer convolutional output of the Faster-RCNN. Multiple residual blocks with 3×3 convolution filter, batch normalization layer and Relu activation layer are introduced to form the generation network.

3.2. Training Function

Let \( f_s \) and \( f_l \) denote the characteristics of small and large objects, respectively. Our goal is to train a generator \( G \) that converts \( f_s \) to \( f_l \) to deceive discriminator \( D \). However, the generator may have difficulty learning from the limited information contained in \( f_s \). Therefore, low-level features \( f \) are introduced to produce residual representations between small and large object representations. Finally, the goal can be described as

\[
\min_G \max_D L(D,G) = E_{f_s \sim p_{data}(f_s)} \log D(f_s) + E_{f_l \sim p_{data}(f_l)} (\log (1 - D(f_s + G(f_l | f))))
\]

To solve the minimum and maximum problems, we take the following strategies. In our design, we aim to map the characteristics of small objects to large objects represented by similar features. Therefore, the generative network generates auxiliary bias, setting non-zero values for small objects, and setting zero values for large objects. To achieve this goal, we must train the network to obtain appropriate parameters. For the discriminator, we take the original large object feature map as input, and obtain suitable parameters through training. The loss function is defined as follows,

\[
L_{dis} = -\log D(f_l) - \log (1 - D(f_s + G(f_l | f)))
\]

Obviously, when the \( L_{dis} \) is minimized, the discriminator would have the ability to distinguish the difference between the generated large objects and the original large objects.

As for the generator, its purpose is to reduce the gap of characteristics between the generated object and the original object through continuous training, so that the generated similar features can deceive
the discriminator. In order to optimize the parameters of the generator, the corresponding loss function is defined as follows.

\[
L_g = -\log D(f_s + G(f_s | f)) \tag{3}
\]

In the end, when the adversarial training reaches balance, that is, when the discrimination network cannot distinguish whether the input is a real feature or a generated feature, We will conduct the object detection training. In the object detection training process, each predictive box extracted from the feature map is finally sent to the fully connected layer, and the fully connected layer has two output layers. The first output layer calculates the probability of each predictive box in \(n+1\) classes, \(p = (p_0,p_1,...,p_n)\) is calculated by softmax regression. The second output layer is responsible for calculating the predictive box coordinates \(r^n = (r^n_x, r^n_y, r^n_w, r^n_h)\). Then, a multitasking loss function is used to perform regression calculation on the type and the coordinates of the predictive box, and the offset of the coordinate of the predictive box is also calculated. Each training predictive box will be marked with a true value category and the object position of the truth box. The next multitasking loss is introduced to adjust the training direction to keep convergence, so that the detection effect is continuously improved.

\[
L_c = L_{obs}(p, g) + \lambda [g \geq 1] L_{loc}(r^n, r) \tag{4}
\]

Where \(L_{obs}(p, g) = -\log p_g\) is the log loss of the real category \(g\). \(L_{loc}\) is the border regression loss function, the detailed calculation is as follows,

\[
L_{loc}(r^n, r) = \sum_{i\in\{x, y, w, h\}} S_{L_1}(r^n_i - r_i) \tag{5}
\]

where the \(S_{L_1}\) is a smooth \(L_1\) loss which is defined as equation (6).

\[
S_{L_1} = \begin{cases} 
0.5x^2 & |x| < 1 \\
|x| - 0.5, & otherwise 
\end{cases} \tag{6}
\]

Up to the present, we already have the corresponding loss function and can easily form the complete loss of the whole network,

\[
L = w_1 L_{dis} + w_2 L_c \tag{7}
\]

where the \(w_1, w_2\) are two weights. Here we just set them to 1.

Under the influence of the loss function, the network continuously adjusts parameters through backward propagation, and finally completes the detection and training process.

4. Experiments

4.1. Datasets

Tsinghua-Tencent 100K [20] is a large traffic sign benchmark set containing 30,000 examples of traffic signs. The resolution of the image is 2048x2048. As in [21], classes with less than 100 instances will be ignored and the final 45 classes will be retained. The semantic definition of the object size is the same as the Microsoft COCO benchmark. Small objects (area <32x32 pixels), medium objects (32x32 <area <96x96), and large objects (area > 96x96).

4.2. Experimental Parameters

For traffic sign detection, the Faster-RCNN with pre-trained ZF-net is used. For the generator and discriminator, the parameters of the newly added convolutional and fully connected layers are initialized with "Xavier" [21]. We also adjusted the maximum size of the image to 1600 pixels as input. The entire network received random gradient descent (SGD) training with a momentum of 0.9 and a weight loss of 0.0005. The number of residual blocks is set to 5. During training, 25% of the object...
proposal is the foreground, which overlaps with the truth ground box at least 0.5 Intersection-over-Union (IOU), and the rest is the background. In addition, our tests were conducted on the Faster-RCNN framework implemented on the NVIDIA GeForce GTX 1070 GPU and Caffe platform.

4.3. Evaluation Standard

According to different performance focuses, there are many evaluation indicators for object detection, such as detection accuracy, detection efficiency, positioning accuracy. This article uses accuracy rate and recall rate as evaluation criteria.

a) The accuracy rate is for our prediction results. It shows how many positive samples are correctly detected in the test results. Then there are two possibilities for a positive prediction. One is to predict the positive class as a positive class \( TP \), and the other is to predict the negative class as a positive class \( FP \), that is,

\[
P = \frac{TP}{TP + FP}
\]  

(8)

b) The recall rate is for our original sample. It shows how many positive examples in the samples were correctly predicted. There are also two possibilities. One is to predict the original positive class as a positive class \( TP \), and the other is to predict the original positive class as a negative class \( FN \).

\[
R = \frac{TP}{TP + FN}
\]  

(9)

In the experimental results of object detection, we certainly hope that the higher the accuracy of the test results is, the better the recall rate is. However, in reality, the two are contradictory in certain circumstances. For example, in extreme cases, we only searched one result. It is accurate, then the precision is 100%, but the recall rate is very low. If we return all the results, the Recall is 100%, but the accuracy will be very low. Therefore, just looking at the accuracy rate or recall rate is not accurate. In different situations, you need to determine if you are more concerned with accuracy or recall. Therefore, in order to avoid the confusion of data analysis by a single detection standard, it is necessary to analyze the experimental results using Accuracy-Recall curves.

4.4. Results and Performance

Table 1 shows a comparison of the accuracy and recall between our method and the other state-of-the methods on the traffic sign detection. It is evident from the data that our method is better than the Faster R-CNN in terms of detection performance. And Fast R-CNN method which is better at detecting small objects still has no obvious advantage over the detection result.

| Methods       | Fast R-CNN | Faster R-CNN | Our method |
|---------------|------------|--------------|------------|
| Accuracy      | 60.91      | 70.18        | 89.65      |
| Recall        | 78.53      | 49.39        | 87.27      |

In order to analyze the performance of the three methods at different scales. We divide the object into three types of scales, and compare their detection performance in traffic signs by the Accuracy-Recall curve, as shown in figure 2. Overall, our method is superior to other methods, especially in the detection of small objects, which proves the effectiveness of our strategy. As shown in sub-graph (a) of figure 2, Fast-RCNN outperforms Faster-RCNN in small object detection. However, the Faster-RCNN achieves better performance than the Fast-RCNN method when introducing the GAN into the Faster-RCNN. It proves that the generator creates a proper bias for small objects and makes the characteristics of small objects similar to large objects, so as to achieve good performance. For large and medium-sized object detection, the three methods shown in figure 2 are slightly different. Objectively speaking, the performance of Fast-RCNN is weaker than the other two methods. When
testing medium and large objects, the Faster-RCNN is slightly weaker than Ours.

Figure 2. The objects in the data set are divided into three levels: the small object, the medium object, and the large object. The method proposed in this paper is compared with the object detection results in this case with Faster R-CNN and Fast R-CNN, and show the comparison data with the Accuracy-Recall curve.
Figure 3. Examples of detection results from the Tsinghua-Tencent 100K dataset. (a) shows the detection effect of small objects under simple background conditions. (b) shows the detection effect of small objects in a complex background. It can be clearly seen that we can detect some traffic signs that are difficult to be seen by the naked eye.

5. Conclusion
This paper introduces GAN into the Faster-RCNN framework. We use the generative network to generate a residual representation that makes the characteristics of small objects similar to the characteristics of large objects, thus deceiving the discriminator. With this strategy, our method effectively improves the performance of small object detection compared to Faster-RCNN. Although our method is based on traffic signs, the application of the method is not limited to the recognition of traffic signs, but can also be used for small object recognition in other situations. Of course, there are many areas where we need to improve. The combination of the two models makes the network structure more complex. Training also needs to be completed in two stages. The increase in time and calculation is unavoidable.

6. References
[1] Girshick R, Donahue J et al 2014 Rich feature hierarchies for accurate object detection and semantic segmentation Proc. Int. Conf. On CVPR (Columbus: IEEE) pp 580-587.
[2] Girshick R 2015 Fast R-CNN Proc. Int. Conf. On CVPR (Santiago: IEEE) pp 1440-1448.
[3] Ren S, He K, Girshick R, et al 2017 Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks TPAMI 39 p 1137.
[4] Redmon J, Divvala S et al 2016 You Only Look Once: Unified, Real-Time Object Detection Proc. Int. Conf. On CVPR (Las Vegas: IEEE) pp 779-788.
[5] Liu W, Anguelov D et al 2015 SSD: Single Shot MultiBox Detector Proc Int. Conf. On ECCV (Amsterdam: Springer) pp 21-37.
[6] Bell S, Zitnick C L et al 2015 Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks Proc. Int. Conf. On CVPR (Las Vegas: IEEE) pp 2874-2883.
[7] Li H, Lin Z, Shen X et al 2015 A convolutional neural network cascade for face detection Proc. Int. Conf. On CVPR (Boston: IEEE) pp 5325-5334.
[8] Yang F, Choi W, Lin Y 2016 Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers Proc. Int. Conf. On CVPR (Las Vegas: IEEE) pp 2129-2137.

[9] Goodfellow I J, Pougetabadjie J et al 2014 Generative Adversarial Networks Advances in Neural Information Processing Systems pp 2672-2680.

[10] Felzenszwalb P F, Girshick R B, et al 2014 Object detection with discriminatively trained part-based models TPAMI 47 pp 6-7.

[11] Redmon J, Farhadi A 2016 YOLO9000: Better, Faster, Stronger Preprint gr-qc /arXiv: 1612.08242.

[12] Radford A, Metz L et al 2015 Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks Computer Science 2015.

[13] Zhang H, Xu T, et al 2016 StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks Preprint gr-qc/arXiv: 1612.03242.

[14] Goodfellow I J, Pouget-Abadie J et al 2014 Generative adversarial nets Proc. Int. Conf. On CVPR (Columbus: IEEE) pp 2672-2680.

[15] Arjovsky M, Chintala S et al 2017 Wasserstein GAN Preprint gr-qc/arXiv: 1701.07875.

[16] Gulrajani I, Ahmed F et al 2017 Improved Training of Wasserstein GANs Preprint gr-qc/arXiv: 1704.00028.

[17] Berthelot D, Schumm T et al. 2017 BEGAN: Boundary Equilibrium Generative Adversarial Networks Preprint gr-qc/arXiv: 1703.10717.

[18] Mirza M, Osindero S 2014 Conditional Generative Adversarial Nets Computer Science pp 2672-2680.

[19] Odena A, Olah C et al 2016 Conditional Image Synthesis with Auxiliary Classifier GANs Preprint gr-qc/arXiv: 1610.09585.

[20] Zhu Z, Liang D, Zhang S et al 2016 Traffic-Sign Detection and Classification in the Wild Proc. Int. Conf. On CVPR (Las Vegas: IEEE) pp 2110-2118.

[21] Glorot X, Bengio Y 2010 Understanding the difficulty of training deep feed forward neural networks Journal of Machine Learning Research 9 pp 249 -256.