Simulation and Comparison of Target Detection Algorithm

王源宇 23320211154247,钟杰 23320211154271,周文波 23320211154272,蒋晨 23320211154224

Deep Learning Courses,Xiamen University,Xiamen,China

{wangyuanyu,zhongjie,zhouwenbo,jiangchen}@stu.xmu.edu.cn

Abstract

Target detection is to identify and track the targets in the image or video, determine the category of multiple targets, and give the location and size of these targets in the picture by the way of border marking.

In this paper, Fast R-CNN, YOLOv3 and YOLOv4 are selected to simulate and compare, analyze the advantages and disadvantages of the selected algorithm, and summarize the application fields of each algorithm.

Introduction

Target detection [1-3], as an important branch in the field of computer vision, has developed at the age of 80. The technology of the recording year encourages the training and learning of network equipment in the network and the improvement of various hardware. The detection of target robots can also be developed by drones. Computer vision is actually the human eye plus the brain, which can recognize and analyze the pictures seen, so that the human eye can realize the automatic function. Target detection can be regarded as the realization of computer vision automation. In detail, the basis of the target is recognition, that is, to recognize the categories and borders of all the images to be tested except the background. The target detection technology can be divided into the following according to whether the target to be tested is specified or not: The first type of size is non-target detection, which also belongs to target category detection. The targets can be identified. Of course, these targets are already defined in the data set. The second type is designated target detection, also known as target instance detection. For a given picture, only a specific target needs to be detected, which can be one target or multiple designated targets, such as a designated Husky or a few designated cars. Compared with designated target detection, non-designated target detection is more difficult. Because in reality, there will be different types of objects that are very similar in appearance characteristics, such as a mobile phone and a square mouse, there is only a small difference in the process of extracting features by the neural network. On the other hand, there are also large differences in appearance characteristics of the same type of objects. Target detection refers to separating one or more targets of interest from the background in an image or video, judging whether there is a target to be tested, and determining the target's location, frame range, and category.

Nowadays, with the continuous breakthrough of technology and the continuous expansion of application range, target detection technology has begun to receive more and more attention. As an important branch of artificial intelligence technology, target detection algorithms have been widely used in monitoring security and confronting networks[4], UAV aerial photography [5], face detection [6], autonomous driving, industrial production, aerospace and other fields.

In the past, limited by real-time and other requirements, some applications still have technical barriers. With the advent of the 5G era, for example, the problem of data flooding in the process of autonomous driving will be solved, and target detection technology will definitely play a huge role in the Internet of Vehicles scene. Not only that, in the Internet of Things, intelligent transportation [7], telemedicine, virtual reality, and smart cities, there is also great potential and room for improvement in target detection algorithms. Whether in theory or practice, target detection Research is of great significance as computer vision and even the entire field of deep learning.

Related Work

The detection steps of the traditional target detection algorithm are relatively cumber-some, and it seems that there are many drawbacks today. Its implementation steps are as follows: (1) First, select the region of the picture using a sliding window; (2) rely on manual selection Method for feature extraction, the typical methods are SITF method and HOG method, specifically scale-invariant feature transformation method, direction gradient histogram method; (3) Feature selection in classifica-tion

mainly uses support vector machine (SVM) and AdaBoost And other methods.

Due to the inevitable shortcomings of traditional target detection algorithms, target detection algorithms based on deep learning have gradually become dominant due to the rapid development of deep learning. Dating back to 2012, Hinton et al. introduced deep learning algorithms for the first time in im-age classification. At that time, they used the AlexNet network and took an important step in target detection. In the competition, it was proved that the accuracy of this algorithm was far ahead of other algorithms, and it won the championship of ImageNet competition. Since then, deep learning has been highly valued by researchers. In this method, the convolutional neural network (CNN) is in-troduced to automatically learn target fea-tures instead of manually extracting features. At the same time, regional candidate boxes or direct regression methods are introduced to replace the sliding window selection method, which greatly improves the accu-racy and real-time performance of target detection. , It has improved the two major drawbacks of traditional target detection.

With the continuous development of deep learning, target detection algorithms based on deep learning have begun to be divided into two categories.

The first is a two-stage target detection algorithm based on candidate regions. As the name implies, the two-stage detection pro-cess is divided into two steps, namely, do-main selection and detection. In detail, the candidate area is selected first, and then the candidate area is identified and detected. The candidate area here is the same as the concept of area selection in the traditional target detection algorithm. The area to be detected is first selected for a given picture, which can greatly reduce the amount of repeated calculations. However, the specific methods for selecting candidate regions are different, and they are mainly divided into two categories. The first is the Region Proposal method adopted by the R-CNN series of algorithms. This method is much simpler and faster than the sliding window traversal method of traditional target detection algo-rithms. It uses typical information such as the edge information and color characteris-tics of the picture, and the selected area greatly improves the accuracy of the target information. We call it high recall rate, and at the same time, the candidate area gener-ated will be much less. Theoretically, it is 2000 areas.

Although the effect of the Region Pro-posal method is much better than that of the sliding window method, it still needs to generate 2000 candidate regions, which will still cause a lot of repeated calculations. Therefore, the Faster R-CNN[8] algorithm proposed an improvement. It adopted the idea of assigning the work of selecting can-didate regions to the neural network to do it, and designed a region selection network (Region Proposal Network, PRN)[9]. One step is very innovative, and it is also one of the important reasons for choosing Faster R-CNN algorithm as the experimental object after this article. It combines the selected candidate area with the subsequent target classification, and is designed as a network, that is, the RPN and the underlying neural network are shared, which greatly improves the detection speed.

The second is a single-stage target detec-tion algorithm based on regression. From the above introduction, it can be known that the detection speed of the two-stage algorithm has been greatly improved, but it still cannot meet the real-time requirements. Therefore, a single-stage algorithm represented by YOLO (You Only Look Once)[10] and YO-LO9000 [11] was proposed. This algorithm eliminates the process of generating candi-date regions and uses the idea of regression analysis to analyze the characteristic in-formation of the input image. Perform re-gression operations directly to obtain target classification and location information di-rectly, simplifying the whole process into an end-to-end regression problem. In recent years, with the iterative upgrade of this series of algorithms, such as YOLOv3[12], YOLOv4[13] and other targets The detection algorithm can already achieve a high bal-ance of accuracy and speed, which can theoretically be applied to practical scenes with high real-time performance.

Proposed Solution

Faster R-CNN

The biggest innovation of Faster R-CNN is the introduction of RPN network for edge extraction. The specific method is as follows: input the image to be tested into the convolutional layer to obtain a feature image, that is, a feature map. The repositioning of the frame position (also called regression) and the classification of objects are merged through the RPN network. Then compare the approximate position information of the target provided by the window selected by sliding on the feature map with the object position information obtained by the regression to obtain a more accurate position.

The entire process of the Faster R-CNN algorithm can be divided into four modules, as shown in Figure 1:

(1) ***Convolutional layer***. As a deep learning algorithm, the core is still the CNN network. A convolution process of Faster R-CNN mainly consists of three parts: convolutional network, activation function and pooling layer. The feature maps of the input image are extracted through the above three operations. This feature map can be used not only in the fully connected layer, but also shared with the RPN network.

(2) ***RPN (Region Proposal Networks).*** The main function of the RPN network is to generate candidate regions. At the same time, the selected frame will also be judged and calibrated by the softmax layer to obtain a more accurate area.

(3) ***ROI layer***. The purpose of this layer is to process the region selected in the previous step and the features extracted by convolution to obtain the feature information of the candidate

region, and then pass the information to the subsequent classifier for target recognition.

(4) ***Classifier***. The function of the classifier is to calculate the confidence of the target category in the candidate frame from the feature information of the candidate area in the upper layer, and to obtain the final frame size and position coordinates through feedback calibration.

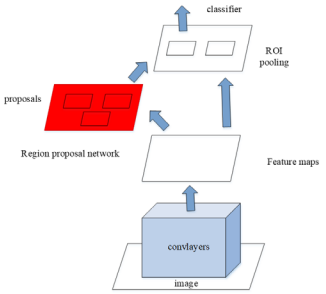

Figure 1: The Structure of Faster R-CNN

YOLO v3

The representative of the single-stage target detection algorithm based on regression thinking is the YOLO series algorithm, and the YOLOv3 algorithm is directly selected for research here. YOLOv3 has the following characteristics and its advantages: the feature extraction network adopts a ResNet-like residual block structure, the frame extraction part adopts a FPN-like multi-scale prediction structure, and a better feature extraction network (Darknet-53), etc. The problem of poor generalization ability of YOLO series algorithms on small objects is well improved.

Figure 2 shows the network flow chart of YOLOv3. The input is specified as 416×416. DBL is conv+BN+leaky relu, conv refers to the convolutional network layer, BN refers to the normalization layer, leaky relu refers to the activation function layer; resn is the residual network, and n refers to the residual block The number of; concat is a tensor stitching layer that combines the sampled values of the two-layer network.

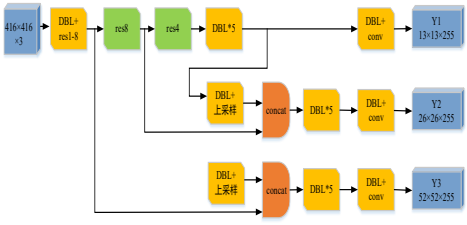

Figure 2: The Network Structure in YOLO v3

(1) ***Backbone***. The backbone network of YOLOv3 is the Darknet-53 network, which has 53 convolutional layers as the name suggests. Although there are 53 convolutional layers, there are a large number of 3×3 and 1×1 convolutional layers. The advantage of this approach is that the size of the convolution kernel can be changed to limit the tensor size. The network introduces the residual neural network structure (Resnet). The idea of this structure is to selectively extract features through a conversion function, instead of inputting the features obtained from all links to the next layer, which can effectively solve deep learning Due to the increase in the number of network layers, the gradient explosion phenomenon is caused.

(2)*FPN-Like Detection*. YOLOv3 draws on the idea of FPN (mentioned above), on the idea of FPN (mentioned above), and proposes FPN-like detection in three sizes of 13×13, 16×16, and 52×52. The three sizes are output as y1, y2, and y3, corresponding to the output result on the far right of Figure 2.5. The specific method is to first divide the entire input image into S×S grids, use clustering to obtain 9 boxes to be tested, and then divide the 9 boxes into three sizes for detection, and finally return to a tensor In terms of value, the specific size is S×S×[3×(4+1)+N], where S usually takes the value 7, 3 refers to three sizes, and 4 refers to the center coordinate of the target frame (x, y) The width w and height h of the frame, 1 refers to the score value of the target category, also known as the confidence level, and N is the number of defined categories. The FPN-like method effectively improves the detection performance of small targets.

(3)*Softmax*. YOLOv3 uses multiple logistic classifiers to replace the softmax classifier. The logistic classifier uses the simplest two-category, that is, it only judges whether it is a certain type of target. Through multiple two-category detection, it can solve the multi-target classification well and has a higher accuracy.

YOLO v4

The last algorithm chosen was YOLOv4. As a derivative algorithm of the YOLO series, its accuracy and real-time performance have reached a high balance, and training can be completed on a GPU, which is very suitable for application in actual scenes.

YOLOv4 can be regarded as an upgraded version of YOLOv3. The algorithm schematic diagram is shown in Figure 3. The YOLOv4 algorithm structure is mainly divided into four parts: Input, Backbone backbone network, Neck, and Head. The innovation and improvement of each part are introduced in detail below.

(1)*Input Module*. The biggest innovation of Yolov4's input terminal is the design of Mosaic data enhancement method. Mosaic can randomly crop, scale, and position 4 images. Many types of small target objects can also be randomly added through scaling, which reduces the work that requires a large number of data sets and makes the network more robust. At the same time, the requirements for hardware facilities are also reduced, so that the same training effect can be achieved through a relatively small batch value, so that training can be completed on one GPU.

(2)*The Backbone*. Yolov4 uses the CSPDarknet53 network structure in the backbone network Backbone, which can enhance the learning ability of the neural network. The main method is to use the concat structure instead of the add structure. The learning process

of the neural network is mainly to extract the target features through the convolution operation. The size and depth of the add operation remain the same, while the size of the concat operation remains the same, but the depth will increase. Compared with the two, it is obviously more. The feature information of the depth value strengthens the learning ability of the network and improves the accuracy.

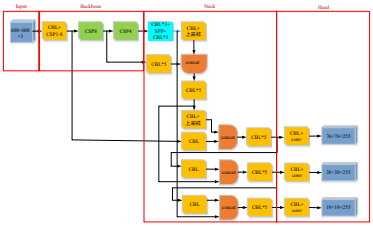

Figure 3: The Modules in YOLO v4

(3)*The Neck. The Neck.* Neck, as the name suggests, is equivalent to the neck of the deep convolutional network. It is a key part of connecting the backbone network and the output layer, which can better extract fusion features. Yolov4 uses the SPP module and FPN+PAN to work together to form the Neck structure.

(4)*The Head.* CIOU Loss is used in YOLOv4 to replace IOU Loss. The calculation method of CIOU Loss is as follows: First, introduce the third box, take the minimum value outside the original two boxes, and calculate the diagonal distance of the third box as the distance between the original two boxes. Then, by calculating the Euclidean distance between the center points of the original two boxes, it can be used as a measure of the ratio of the two. This method can effectively solve the problem of IOU Loss.

MS COCO Dataset

This article chooses the MS COCO data set as the standard, mainly because of the following advantages:

(1) First, it is a very large data set with more than 330,000 pictures, of which 200,000 pictures can be used for training with annotated pictures. , Can provide a wealth of training samples for the algorithm. In terms of object categories, it provides 91 types of objects, which can be said to be very challenging.

(2) The number of individually labeled individuals in the 200,000 labeled samples is as high as 1.5 million. The target detection and positioning provided during the training process are very accurate and more suitable for practical applications.

(3) Compared with other data sets, in the scene understanding problem, it provides a wider range of object recognition and more accurate contextual semantic relations, making a great contribution to the development of object recognition.

Therefore, this article uses the MS COCO data set as the main basis for algorithm comparison.

Experiment

The Device

The implementation of the algorithms in this paper uses the Pytorch framework, and the experimental configuration is shown in Table 1.

The Results and Comparison

The experimental results of the three algorithms, i.e. Faster R-CNN, YOLO v3, YOLO v4 on the COCO data set are shown in Figures 4, 5, and 6 respectively (here, the accuracy value AP and the speed value FPS are mainly intercepted):

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.394
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.609
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.422
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.216
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.438
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.539
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.322
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.508
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.533
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.325
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.581
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.706
fps:27.83: 100%|
```

Figure 4: The test result of Faster R-CNN, running on the COCO dataset

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.433
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.630
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.470
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.284
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.485
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.538
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.346
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.581
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.634
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.474
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.687
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.766
Speed: 11.8/14.8/26.5 ms inference/NMS/total per 640x640 image at batch-size 32
```

Figure 5: The test result of YOLO v3, running on the COCO dataset

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.474
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.662
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.515
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.297
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.524
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.615
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.367
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.599
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.652
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.470
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.703
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.797
Speed: 5.3/1.1/6.4 ms inference/NMS/total per 640x640 image at batch-size 32
```

Figure 6: The test result of YOLO v4, running on the COCO dataset

The Comparison are listed in table 2, table 3:

Table 1: The experimental configuration

Unit	CPU	RAM	GPU	OS
Model	Intel i7-9800X	64G	GeForce RTX 2080ti	Ubuntu18.04

Table 2: The average FPS and (AP)IOU are listed below

Algorithm	Backbone	FPS	（AP）@IoU=0.50:0.95 area=all
Faster R-CNN	ResNet-101	27.83	0.394
YOLOv3	Darknet-53	37.74	0.433
YOLOv4	DSPDarknet-53	156.25	0.474

Table 3: The mAP on three different areas, small, medium and large are listed below

Algorithm	area=small	area=medium	area=large
Faster R-CNN	0.261	0.438	0.539
YOLOv3	0.284	0.485	0.538
YOLOv4	0.297	0.524	0.615

The Results and Analysis

The following conclusions can be drawn from the experimental results:

First, as shown in Table 4.2, the three algorithms are compared vertically. In terms of detection accuracy, in the case of area=all, YOLOv4 has a higher recognition rate of 0.474 than the other two algorithms. At the same time, in the other three cases (small targets, medium targets, and large targets), the recognition accuracy of YOLOv4 is higher than The other two algorithms. Accuracy level behavior: YOLOv4>YOLOv3>Faster R-CNN.

As shown in Table 4.3, compare the detection accuracy of one of the algorithms for small targets, medium targets and large targets. Taking the YOLOv4 algorithm as an example, the accuracy of small targets, medium targets and large targets are 0.297, 0.524, 0.615 respectively. It can be seen that the detection accuracy of the YOLOv4 algorithm on large targets is significantly higher than that of small targets, and the other two algorithms have similar conclusions.

According to the analysis and comparison of the experimental results, Faster R-CNN is a representative of the two-stage target detection algorithm. Compared with the previous generations of R-CNN algorithms, in order to improve the detection accuracy, a higher number of ResNet-101 is introduced. As the backbone network. Although the detection accuracy has been improved to a certain extent, the expansion of the algorithm model has caused an increase in the amount of calculation, and the detection speed is only 27.83FPS, which cannot meet the real-time requirements. For this reason, YOLOv3 and YOLOv4 are the representatives of single-stage target detection algorithms. The idea of transforming target detection into regression problems can be said to have made a breakthrough simplification in the algorithm model, which not only improves the detection accuracy, but also The contribution is to improve the detection speed. Although the YOLO series algorithm adopts the regression idea to improve the detection speed to a certain extent, it also has a good generalization ability for large-scale targets, but the detection accuracy for small targets is low. For the single-stage detection algorithm, the backbone network is DarkNet-53, CSPDarkNet-53 and other networks. The detection speed is continuously improving, and the detection accuracy is also continuously improving and exceeds the two-stage target detection algorithm. YOLOv4 achieves a high balance between speed and accuracy.

Conclusion

The main work of this paper is to simulate and compare the three algorithms. First, the paper propose the network architecture of these algorithms. Then, obtain the experimental results of related algorithms on various data sets and some areas of current target detection by consulting information, and propose the advantages and disadvantages of the three algorithms and applicable scenarios.

In general, as an important research field of computer vision, target detection has achieved many excellent results. As one of the prospects in the field of target detection, with the development of technologies such as neural networks, it is believed that target detection technology will have better performance in various practical applications in the future.

References

[1] 黄健，张钢．深度卷积神经网络的目标检测算法综述[J]．计算机工程与应用，2020，56(17):12-23.

[2] 许德刚，王露，李凡．深度学习的典型目标检测算法研究综述[J/OL]．计算机工程与应用，2021，57(08)：10-25．

[3] 梁鸿，王庆玮，张千，等．小目标检测技术研究综述[J]．计算机工程与应用，2021，57(01)：17-28．

[4] 李诚，张羽，黄初华．改进的生成对抗网络图像超分辨率重建[J]．计算机工程与应用，2020，56(04)：191-196．

[5] 魏玮，蒲玮，刘依．改进 YOLOv3 在航拍目标检测中的应用[J]．计算机工程与应用，2020，56(07)：17-23．

[6] 罗明柱，肖业伟．全卷积神经网络的多尺度人脸检测的研究[J]．计算机工程与应用，2019，55(05)：124-128+165．

[7] 王为，姚明海．基于计算机视觉的智能交通监控系统[J]．浙江工业大学学报，2010，38(05)：574-579．

[8] 伍伟明．基于 Faster R-CNN 的目标检测算法的研究[D]．广州：华南理工大学，2018．

[9] Ren Shaoqing, He Kaiming, Girshick Ross, et al. Faster R-CNN:Towards Real-Time Object Detection with Region Proposal Networks[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, 39(6).

[10] Redmon J, Divvala S, Girshick R, et al. You only look once：unified, real-time object detection[C]. *IEEE conference on Computer Vision and Pattern Recognition*, 2016.

[11] Redmon J, Farhadi A. YOLO9000：better, faster, stronger[J]. arXiv:1612. 08242v1, 2016.

[12] Redmon J, Farhadi A. YOLOv3：an incremental improvement[J]. arXiv:1804. 02767v1, 2018.

[13] Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4:Optimal Speed and Accuracy of Object Detection[C]. *IEEE conference on Computer Vision and Pattern Recognition*. 2020. arXiv:2004.10934v1 [cs.CV].