Instance Segmentation of Traffic Scene Based on YOLACT

Xingxing Li*, Chao Duan, Yan Zhi, Panpan Yin, Ningxing Wang

Department of electronics and information engineering, Guangzhou College of Technology and Business, FoShan, 528138, China.

*Corresponding author: wslxx@jxstnu.edu.cn

Abstract. As deep learning shines in the field of machine vision, this article uses the YOLACT instance segmentation algorithm to solve the detection and segmentation of pedestrians, vehicles and other objects in traffic scenes. In this paper, the backbone in the YOLACT algorithm is changed from the original resnet50 to the lightweight resnet18 to ensure that the algorithm can be deployed on Jetson-AGX-Xavier, and the processing speed can reach 9.4 frames per second, and the map can reach 81%. In NVIDIA 2070 super, the inference speed on the graphics card can reach 40 frames per second.

1. Introduction

In a natural traffic scene, there are a large number of pedestrians, vehicles and other objects. These targets can be analyzed with the help of artificial intelligence vision algorithms. In the past, target detection algorithms can only give the target frame of the object, but cannot pass the target object through the button. Mapped out. Using the actual segmentation algorithm can solve the problem of object segmentation. Not only can the target frame of the object be obtained, but also the mask image of the target can be obtained. The mask image can better combine the 3D distance information such as laser point cloud and correspond to the original image. On the top, dig out the depth information of the object. Therefore, the instance segmentation algorithm is very important in traffic scenes.

In the design of the target detection model, there are two main routes: Two-stage network pays more attention to accuracy, such as R-FCN, Faster-RCNN, etc [1]. One-stage network pays more attention to speed improvement, and the representative work YOLO [2], SSD 3, etc. Mask R-CNN [4] is a model of instance segmentation model. It is based on the improvement of Faster-RCNN [1] and laid the tone of Two-stage instance segmentation. Similarly, the focus of Mask R-CNN is also on the improvement of accuracy. The YOLACT author hopes to design a One-stage instance segmentation model based on the One-stage target detector. SSD/YOLO [2, 3] etc. accelerate the Faster R- by removing the second stage and making up for the lack of performance in other ways. CNN [1]. However, this idea is not easy to directly apply to instance segmentation tasks, because instance segmentation relies heavily on feature positioning in order to generate masks. In the Two-stage model, re-pooling operations (such as ROI align, ROI pooling) are used to map features to the bounding box. This approach is logically serial, so it is difficult to speed up. Although FCIS parallelizes these operations, it is also difficult to speed up due to too many post-processing steps.

Based on the above considerations, this paper proposes YOLACT [5], which abandons the implicit feature location step and splits the instance segmentation task into two parallel subtasks: (1) Generate a series of prototype masks covering the entire image; (2) For each instance, predict a series of linear
combination coefficients. Finally, during inference, for each instance, use its corresponding predicted mask coefficient, simply multiply and add it with the prototype mask, and then crop and threshold according to the bounding box to obtain the mask corresponding to each instance.

After experimental analysis, YOLACT uses the prototype mask to adaptively learn how to locate the instance target. The number of prototype masks has nothing to do with the number of categories, which means that each prototype mask contains cross-category information. YOLACT aims to learn a distributed representation method so that each instance can perform multiple prototype masks through this representation method Linear combination to get your own mask. Through training, each prototype mask will learn to abstract some details of the input image, such as edge information, position information, or information in response to a specific area.

YOLACT has three significant advantages: (1) Fast speed, because it uses a one-stage network framework, it can meet real-time requirements in practical applications, but the detection speed is also inversely proportional to the number of objects detected, that is, detection The more objects there are, the more time-consuming the post-processing of the algorithm. (2) The mask quality is high because it does not contain re-pooling operations. (3) Strong universality. This idea of generating prototype masks and mask coefficients can be applied to many popular detectors [5].

2. Principle

2.1. YOLACT network

Compared with the human visual system, YOLACT is also very intuitive: linear coefficient combination and detection solve the problem of what the object is, and the generation of the prototype mask seems to be solving the "where" problem. The performance comparison between YOLACT and some other existing instance segmentation models on the COCO dataset is shown in Figure 1:

![Figure 1. Performance of YOLACT real-time instance segmentation algorithm.](image)

In Figure 1, comparing the YOLACT algorithm with other instance segmentation algorithms on COCO data, YOLACT is a compromise between speed and performance in various situations. As can be seen from the figure, YOLACT is the first real-time (over 30 FPS) method, and the mask on the COCO data set can reach 30 mAP.

Analogous to Mask R-CNN to Faster R-CNN, YOLACT aims to add a mask branch to the existing one-stage detector to achieve the purpose of instance segmentation, but it is not desirable to introduce feature positioning steps in this process. YOLACT accomplishes this task by adding two parallel
branches: the first branch uses FCN to generate a series of prototype masks independent of a single instance; the second branch adds additional headers to the detection branch to predict the mask coefficients to use to encode the representation of an instance in the prototype mask space. Finally, after the NMS step, the final prediction result is obtained by linearly combining the output results of the two branches. The network structure of YOLACT is shown in Figure 2:

Because the goal of the segmentation task is to get the mask, and the feature of the mask is that there is a natural spatial connection, YOLACT adopts the above organization form. From the perspective of NN, the Conv layer naturally uses spatial correlation, but the FC layer does not. This leads to a problem, because most One-stage detectors predict box parameters and categories through the FC layer. Two-stage retains spatial information through feature positioning steps such as ROI Align, and uses the Conv layer to output the mask, but these operations must wait for the RPN to complete, which greatly affects efficiency. In YOLACT, the FC layer is responsible for predicting semantic tags, and the Conv layer is responsible for predicting the prototype mask and mask coefficients. The two branches are in parallel, and finally assembled by matrix multiplication, which not only preserves the spatial correlation, but also maintains the One-stage model structure, which is extremely fast.

The classic Anchor-based detection model predicts 4 values for each Anchor to represent the box information, and C values to represent the category score, a total of (4+C) values. YOLACT predicts (4+C+k) values for each Anchor, and the additional k values are the mask coefficients. In order for YOLACT to obtain the final desired mask through linear combination, it is important to be able to subtract the prototype mask from the final mask. In other words, the mask coefficient must be positive and negative. Therefore, the $\tanh$ function is used for nonlinear activation in mask coefficient prediction, because the value range of the $\tanh$ function is (-1, 1). The output of the two branches is processed through the basic matrix multiplication with the sigmoid function to synthesize the mask.

$$M = \sigma(PC^T)$$

Among them, $P$ is the prototype mask set of $h \times w \times k$, and $C$ is the coefficient set of $n \times k$, representing $n$ instances that pass the NMS and threshold filtering, and each instance corresponds to $k$ mask coefficients.

Loss is composed of classification loss, box regression loss and mask loss. The classification loss and box regression loss are the same as SSD, and the mask loss is the pixel-by-pixel binary cross entropy of the predicted mask and ground truth mask. In order to improve the segmentation effect of small targets, YOLACT cuts the Mask. During inference, it will first cut according to the detection frame and then threshold. During training, the ground truth box is used for cropping, and the loss scale is balanced by dividing by the area of the corresponding ground truth box.
2.2. Backbone Detector

Because predicting a set of prototype masks and mask coefficients is a relatively difficult task and requires richer and more advanced features, YOLACT hopes to balance speed and feature richness in network design. Therefore, the design of YOLACT's backbone detector follows the idea of Retina-Net, while paying more attention to speed. YOLACT uses ResNet-101 combined with FPN as the default backbone network, and the default input image size is 550×550. Compared with the original Retina-Net, the design of the YOLACT detection head (as shown below) is lighter and faster.

The detection head uses smooth $l_1$ loss to train bounding box parameters, and uses the same bounding box parameter encoding method as in SSD 3]. Use softmax cross entropy to train the classification part, a total of $(C+1)$ categories. At the same time, use the OHEM method to select training samples, and the ratio of positive and negative samples is set to 1:3. This article is based on the Jetson-AGX-Xavier embedded board development environment. Using the backbone of renset101 and the input of 550×550 size images can no longer meet the real-time requirements. Therefore, in the case of doing multiple experiments, deploy in Jetson-AGX-Xavier uses resnet18 as the backbone of the network, and uses the input size of the 450×450 network.

In the training phase, the auxiliary loss is added. The specific method is to follow the $1\times1$ convolution after the largest feature map, and output a feature map with the number of channels $c$. Its corresponding ground truth is obtained from the annotation of instance segmentation, so there is no mandatory requirement that each pixel can only belong to a single category. Therefore, the loss calculation method used here is to run sigmoid on $c$ channels, similar to the loss design of YOLO-v3 multi-label classification.

3. Experiments

The data set in this paper is taken by traffic surveillance cameras. The data set is divided into training set and test set according to the ratio of 8 to 2, including 1000 training set and 200 test set.
It can be seen from Figure 4 that in the traffic scene, most of the pedestrians, vehicles, and people riding electric vehicles appear in the image. So this article divides the target into 4 categories, the first category is the background, and the others are pedestrians, vehicles, pedestrians and vehicles. This article uses YOLACT as an example segmentation algorithm for two main purposes. One is to ensure that the above four types of targets can be accurately identified and segmented in traffic scenarios, and the other is to deploy and apply the algorithm to Jetson-AGX-Xavier. This article uses different resnet18 lightweight networks as the backbone on Jetson-AGX-Xavier. The mAP performance on the test set is shown in the following table:

|          | .50  | .55  | .60  | .65  | .70  | .75  | .80  | .85  | .90  | .95  |
|----------|------|------|------|------|------|------|------|------|------|------|
| Box      | 51.27| 81.45| 78.48| 74.34| 70.05| 63.44| 54.96| 43.61| 27.75| 14.70|
| Mask     | 21.11| 46.17| 39.09| 33.26| 26.54| 21.81| 16.39| 12.66| 9.50 | 5.45 |

It can be seen from Table 1 that in a traffic scene, YOLACT has an accuracy of 81.45% for the mAP of the bounding box above 0.5, and an accuracy of 46.17% for the mask. The basic algorithm can detect most targets while accurately segmenting objects shape.
4. Conclusion
This article uses the YOLACT one-stage instance segmentation algorithm to complete the detection and segmentation of pedestrians and vehicles in traffic scenes. In order to be deployed on Jetson-AGX-Xavier, this article uses resnet18 as the backbone of YOLACT, and the processing speed can reach 9.4 frames per second. Map reaches 81%, Mask can reach 46% accuracy, and the inference speed can reach 40 frames per second on the NVIDIA 2070super graphics card, which is practical in engineering.

Acknowledgments
This work was financially supported by fund project, that is, Young Talents in Higher Education of Guangdong, China, (No. 2019KQNCX231).

References
[1] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1137-1149.
[2] Bochkovskiy, Alexey, C. Y. Wang, and H. Y. M. Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection." (2020).
[3] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector[C].European Conference on Computer Vision. Springer International Publishing, 2016.
[4] Kaiming H, Georgia G, Piotr D, et al. Mask R-CNN[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, PP:1-1.
[5] Bolya D, Zhou C, Xiao F, et al. YOLACT: Real-time Instance Segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, PP(99):1-1.