Mask-YOLO model for fast instance segmentation

Huaiyu Zheng
Northeastern University
zheng.hua@husky.neu.edu

Yaofei Wang
Northeastern University
wang.yaof@husky.neu.edu

Zhiyang Liu
Northeastern University
liu.zhiya@husky.neu.edu

Abstract

We proposed a network combined yolov3 and mask rcnn to do the instance segmentation task. We use darknet as feature map backbone, and yolov3's original approach to get classes and bbox. Then align bbox on the feature maps and go through the mask fcn branch to get mask prediction.Hopefully our network can be faster than mask rcnn and get high AP point. We train our network on COCO [1] dataset.

1. Introduction

The state-of-art method Mask RCNN [2] is really powerful and has elegant precisions for instance segmentation. However, the slow feature map generation process become a bottleneck when we try to increase the speed. Also, it's based on two stages: region proposal network and detection network(classification, detection and mask prediction). We'd like to combine the strength of Mask RCNN with another method, which has faster speed and one-stage end-to-end training.

YOLOv3 as one of the trending framework in object detection, enjoys real-time speed with its unified structure and even higher precision with the pyramid of multi-scale feature maps. With similar performance on COCO dataset, YOLOv3-416 runs two times faster then RetinaNet-50-500 [3]. It's also observed that with spatial pyramid pooling method, the framework would be more robust to the scale change and occlusion [4]. By adding a mask branch, we'd like to incorporate this extreme fast object detection method in our instance segmentation network.

2. Related Works

2.1. Mask RCNN

The proposal-based method is most popular in instance segmentation, which have a strong connection to object detection. Mask R-CNN extends Faster R-CNN [5] by adding a branch for predicting segmentation masks on each Region of Interest (RoI), in parallel with the existing branch for classication and bounding box regression.

They predict an $m \times m$ mask from each RoI using FCN [6], which allows each layer in the mask branch to maintain the explicit $m \times m$ object spatial layout without collapsing it into a vector representation that lacks spatial dimensions. Also, the FCN requires less parameters compared with fc layers.

Before passing the RoI into the FCN, instead of RoIPool, the Mask RCNN applies RoIAlign to properly align the extracted features with the input. In this way, they avoid any quantization of the RoI boundaries or bins.

2.2. YOLO

YOLO is an end-to-end object detection framework that runs in real time. The first version comes out in 2016, which is quite impressive with its real-time performance. Then we have YOLOv2 and YOLOv3, which takes YOLO to a new level with its even higher precision, especially when detecting small objects.

Aside from a structure called Darknet-53, which originally has 53 layer network trained on ImageNet, YOLOv3 also incorporates residual skip connections, upsampling, and makes detections at three different scales, which helps address the issue of detecting small objects.

Our work is based on YOLOv3, and explores several morphing structures like YOLOv3-tiny and YOLOv3-spp. The latter one, YOLOv3-spp is especially robust to objects in different scales with its multiple effective field-of-views. Based on the YOLOv3 part, we add a mask branch to turn the object detection problem into an instance segmentation task.

3. Mask-YOLO Model

Inspired from mask rcnn to build a multi-task learning using multiple fully connected layer. We want to add another branch on yolo to do the instance segmentation task.

A good feature map backbone is base for tasks like classifier, bbox perdition and instance segmentation. In mask rcnn, better backbone alone improves AP by 4 point. And YOLOv3 purposed darknet53 which has deeper networks.

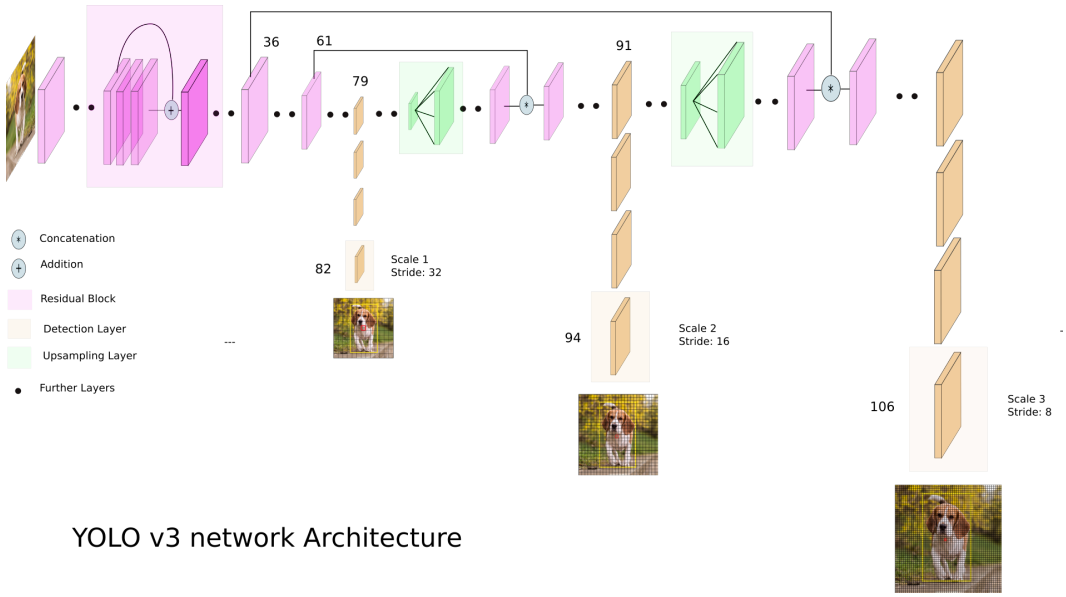

Figure 1. YOLOv3 network Architecture

It has multi scale layer structure like feature pyramid network. Compare with mask rcnn, YOLOv3 can provide more accurate bbox proposal and higher speed. Hopefully more accuracy bbox can give us higher AP point on instance segmentation task.

3.1. Feature Map

We use YOLOv3's darknet59 as our feature map backbone. The yolov3 framework looks like figure 1.

Darknet59 introduced the idea of ResidualNetIt, which inclues 3×3 convolution kernel, 1×1 convolution kernel and short cut connection. It can extract feature in the image faster than ResNet. The multi-scale feature map is merged element-wide by adding. In this way, we can get more high-level semantic information from the later layers, and also get fine-grained information from the pervious layer.

3.2. FCN branch

For the network, we use YOLOv3's original method to predict the classes and bbox. YOLOv3 makes predictions on three different scales. We predict 3 boxes per scale.. Applying convolution layers on each scales feature map shrinks it to the size of predition feature map which is $N \times N \times [B \times (4+1+80)]$. The shape of the detection kernel is $1 \times 1 \times (B \times (5+C))$. Here B is the number of bounding boxes a cell on the feature map can predict, "5" is for the 4 bounding box attributes and one object confidence, and C is the number of classes. In YOLOv3 trained on COCO, $B = 3$ and $C = 80$, so the kernel size is $1 \times 1 \times 255$.

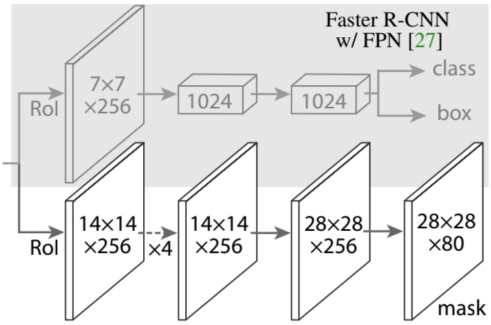

Figure 2. Mask branch prediction

And we add a fully connected branch on the layer before upsampling which used as our mask prediction feature map. We can know which layer we should use based on YOLOv3 classes and bbox prediction. We align predicted bbox on that feature map layer. Then put that feature map into mask branch to get mask prediction. The structure of mask branch also followed the design of mask rcnn shows in figure 2. RoIAlingn generates a feature map with size of 14x14. Then make a deconvolution to increase the feature map to 28×28, and take sigmoid for each classes. The output is a binary mask with size of 28×28. At last, transform the 28×28 mask using bilinear interpolation to the size of the box in the original image.

4. Experiment

TBD

5. Conclusion

TBD

References

[1] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[3] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[4] Kwang-Ju Kim, Pyong-Kun Kim, Yun-Su Chung, and Doo-Hyun Choi. Performance enhancement of yolov3 by adding prediction layers with spatial pyramid pooling for vehicle detection. pages 1–6, 11 2018.

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[6] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.