CMPE 537
Term Project Report

Oğuzhan Sevim

January 30, 2021

1 INTRODUCTION

In this project, I have experimented with the instance segmentation architecture called Mask R-CNN [1].

Instance segmentation problem can be defined as the combination of semantic segmentation and object detection. In object detection, the goal is to find bounding boxes around each object along with the class labels of these objects. Besides, the semantic segmentation can be defined as a pixel level classification task that aims to detect the boundaries of all objects in a given image. Instance segmentation result of a random image is shown in Figure 1.

Figure 1: Instance segmentation results obtained on a random image.

When it was introduced in 2017, Mask R-CNN was the amongst the state-of-the-art methods for instance segmentation. In [1], authors demonstrate the success of Mask R-CNN by making comparisons with the 2015 and 2016 winners of MS COCO instance segmentation challenge. First of these methods is the Multi-task Network Cascades (MNC) given in [2]. MNC performs instance segmentation by using following 3 steps in a cascaded manner: proposing bounding boxes, finding segmentation masks, and classifying each of these masks. The second method used for the comparison is Fully Convolutional Instance-aware Semantic Segmentation (FCIS) [3]. Figure 2 illustrates the mAP (mean average precision) scores of the 3 architectures on the instance segmentation task on COCO validation set.

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC [10]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [26] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [26] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	ResNeXt-101-FPN	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

Figure 2: Comparison results of Mask R-CNN, MNC, and FCIS networks.

The results shown in Figure 2 are reported from [1]. AP_{50} and AP_{75} refers to the mean average precision scores when the masks with intersection-over-union (IoU) scores higher than 50% and 75% are considered. Also, AP_S, AP_M, and AP_L denote the mAP score for small, medium, and large instances. By small, the instances that have less than 32^2 pixels are considered. Medium instances are the ones that have the pixel count between 32^2 and 96^2 pixels. Likewise, the instances with pixels count higher than 96^2 are considered as large instances. Figure 2 illustrates well the success of Mask R-CNN compared to the previous state-of-the-art instance segmentation architectures.

2 MASK R-CNN METHOD

In this section, some details about the Mask R-CNN network will be presented. The Mask R-CNN network is built incrementally upon its 3 predecessors: R-CNN [4], Fast R-CNN [5], and Faster R-CNN [6]. In order to understand the Mask R-CNN architecture it is better to start with the Faster R-CNN architecture shown in Figure 3(a). By using a region proposal network (RPN) integrated in itself, Faster R-CNN does not require any external region proposal algorithms (e.g., selective search [7]). Since RPN makes use of the same convolutional layers that rest of the network uses, a Faster R-CNN model can be trained faster than its predecessors. During test time, it can process 5 images per second. As illustrated in Figure 3(b), Mask R-CNN architecture differs from Faster R-CNN by the additional branch that performs pixel level classification. This additional branch produces the masks that are used for semantic segmentation. By combining this branch with the Faster R-CNN, which finds the class labels of each masks, Mask R-CNN achieves can perform instance segmentation.

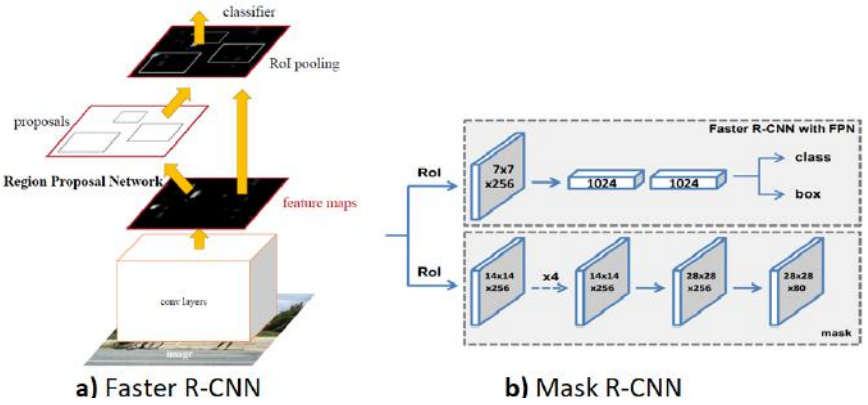


Figure 3: Faster R-CNN and Mask R-CNN architectures.

3 EXPERIMENTS

The experiments on Mask R-CNN are done by using the Keras and Tensorflow implementation given in [8]. Since ResNet-101 is used as the backbone, the number of parameters in the network is quite high (more than $44M$). Thus, transfer learning is used by utilizing network weights that are trained on MS COCO training set.

Different experiments are conducted by using Mask R-CNN architecture. First, some of the scores shown in Figure 2 are tried to be reproduced. In the second part, the network is changed in such a way that it performs instance segmentation only on cat class. The details of these experiments are discussed in Sections 3.1, 3.2, and 3.3.

3.1 Attempts to Replicate Original Results

By using the weights trained on COCO training set, I have tried to replicate AP_{50} and AP_{75} results shown at the 5^{th} row of Figure 2. COCO *minival* test set, which consists of approximately $5k$ images are used for this purpose. The test results obtained in the paper and in my simulations are shown in Table 1. The results of the paper and my simulations are quite different from each other. A possible reason for this difference may be that COCO *minival* test set are different than the one used in the original paper [1].

Table 1: Comparison of test results.

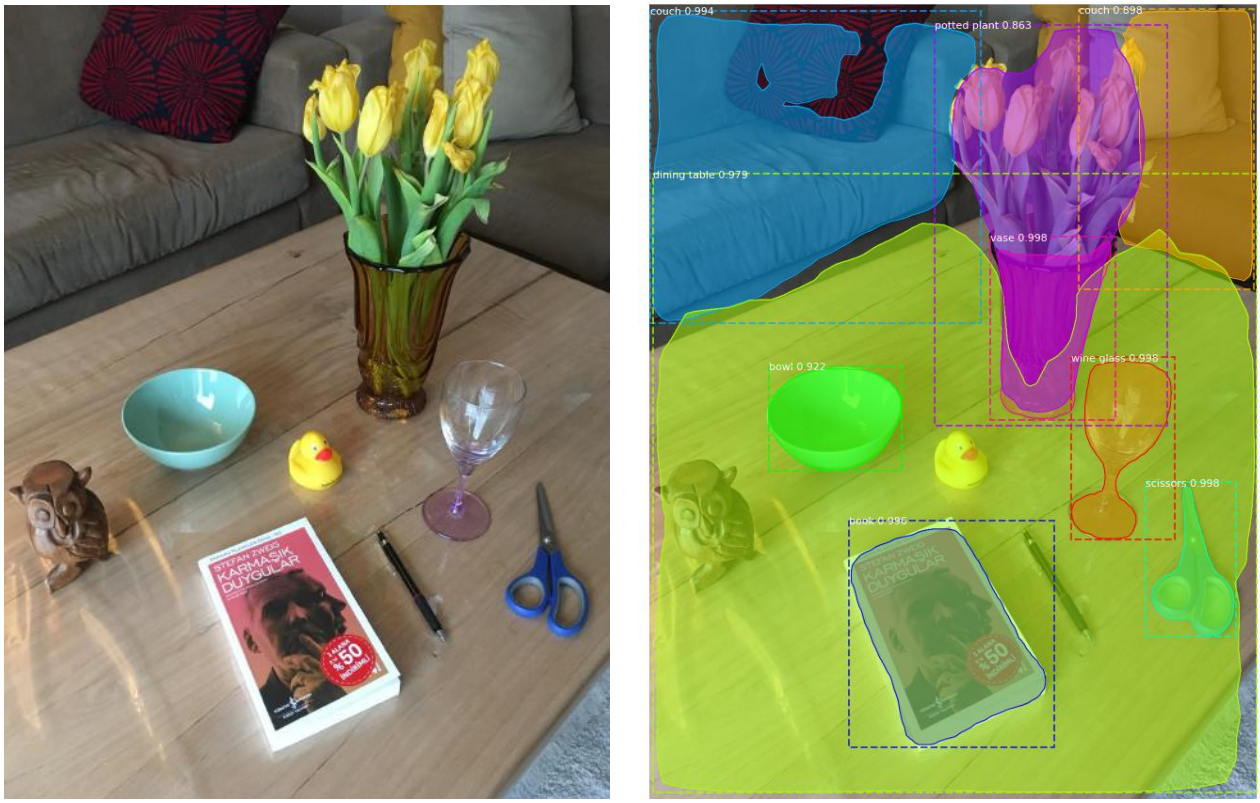
	AP_{50}	AP_{75}
In paper	58	37.8
My trial	65.5	46.2

3.2 Occlusion Experiments

The effect of occlusion is investigated in this section. 3 images along with their corresponding segmented versions are used for this purposed. The first image and its segmentation result are shown in Figure 4. This pair of images are used for demonstrating how well the segmentation masks are generated when there is no occlusion among the small objects.

The first occlusion experiment is done by using opaque objects where the results are illustrated in Figure 5. When the scissor is placed on top of the book, the book can not be detected by the model. Similarly, the occlusion reduces the mask quality of the scissors. The decrease in the mask quality can be seen by looking at the scissor mask shown in Figure 4(b). Furthermore, since the edges of table do not present in the image, the model can not detect the table just by looking at the texture.

The second occlusion experiment is done by using transparent objects wine glass and vase. As shown in Figure 6(b), when the wine glass is placed in front of the bowl, boundaries of the bowl mask get highly distorted and its confidence score decreases from 0.922 to 0.819. When the vase is occluded by the wine glass, only the part of the vase gets detected. However, the flowers inside the transparent vase are still detectable.



(a) Test results (b) Segmented image

Figure 4: Results when there is no occlusion among the following small objects: bowl, book, vase, wine glass, and scissors.

(a) Original image (b) Segmented image

Figure 5: Results when there exists occlusion between opaque objects scissors and book.

(a) Original image (b) Segmented image

Figure 6: Results when the occlusions are caused by transparent objects of wine glass and vase.

3.3 A Cat Detector Application

For the last part of the experiments, I have built a cat detector system. In repository [8], there already exists a balloon detection script. It works by changing the last layers of the Mask R-CNN such that it performs detection only for balloon class instead of many. The network starts with pretrained weights and newly added layers are fine-tuned by using a small data set of balloons.

In order to convert this balloon detector into a cat detector, I needed to create a small data set of cat images. The data set is created by annotating 30 cat images. As the annotation tool, [9] is used. It offers very friendly interface where annotations can be saved in different formats. For me, annotating a single cat object took approximately $1-2$ minutes.

After training the last layers of the cat detector Mask R-CNN for 30 epochs, the system was ready to use. I have tested it on a video of my cat. The resulting video can be seen by clicking this link.

4 CONCLUSIONS

When it was introduced, Mask R-CNN was among the state-of-the-art instance segmentation architectures. It is ability to process 5 frames per second and its relative simplicity still make it a very useful tool.

In this project, I have experimented with the Mask R-CNN model by using a publicly available implementation. First, I have tried to replicate 2 of the test results reported in the paper. I have failed to reproduce the same results most probably because of the data set differences.

Furthermore, I have tried to observe the effects of occlusion caused by opaque and transparent objects. We saw that existence of occlusion depreciates the produced masks. As a natural result of light refraction ability of transparent objects, the distortion of mask region can outreach outer areas of the objects itself. However, I should admit that I could do more comprehensive tests about the occlusion issue.

Lastly, I have converted a balloon detector implementation into a cat detector. I have created my own data set by annotating 30 images by hand. It was a nice experience. This application shows how easily a pretrained Mask R-CNN can be converted for a more specific application.

References

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[2] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3150–3158.

[3] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2359–2367.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

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

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[7] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[8] W. Abdulla, "Mask R-CNN for object detection and instance segmentation on Keras and Tensorflow," https://github.com/matterport/Mask_RCNN, 2017.

[9] A. Dutta, A. Gupta, and A. Zissermann, "VGG image annotator (VIA)," http://www.robots.ox.ac.uk/ vgg/software/via/, 2016, version: 2.0.10.