Project 2: Mask R-CNN

***Abstract*—This project is about exploring Mask R-CNN for instance segmentation. This network can predict the bounding box and class label as well as the segmentation mask. The average precision of Mask R-CNN at an IoU of 0.5 is computed and the performance of this network on image segmentation is also assessed. For the extension, the Mask R-CNN which is already trained over the COCO dataset is trained on the balloon dataset. Then, the performance of this network is used to detect balloons over the grayscale images and colored video. In addition, a pre-trained COCO Mask R-CNN model is trained over the new dataset including four different shapes (circle, ellipse, triangle, and square) to detect and segment these shapes.**

I. Introduction

Instance segmentation is a challenging task in computer vision since object detection and instance segmentation should be done simultaneously. The purpose of applying instance segmentation is to classify each pixel into specific categories. For this challenging task, [1] proposed a simple and fast method, Mask R-CNN that excels in results compared to prior methods. This method is based on Faster R-CNN [2] with some modifications (Figure 1). A new branch is added to work parallel with the classification and bounding box regression branch to predict segmentation mask in each Region of Interest (ROI). The mask branch is a small Fully Convolutional Network (FCN) that does not exert high computational cost on the system. To address the pixel-to-pixel misalignment of Faster R-CNN, the quantization-free layer, called RoIAlign is proposed to keep the spatial locations. Thus, Mask R-CNN has three outputs, a class label, a bounding box offset (These two outputs are related to the Faster R-CNN), and the object mask.

A. Mask R-CNN

The architecture of Mask R-CNN is similar to the Faster R-CNN, containing two stages. The first stage is the Region Proposal Network (PRN) which proposes bounding boxes for objects and the second stage is extracting features from bounding boxes with ROIPool. A binary mask for each ROI is also predicted in parallel to class prediction in the second stage. Therefore, the loss function (L) in training is a multi-task loss as follows:

$$L = L_{cls} + L_{box} + L_{mask} \tag{1}$$

where L_{cls} is the classification loss and L_{box} is the bounding box loss, which are identical to those defined in [3]). L_{mask} is the segmentation mask loss, which is defined as the average binary cross-entropy loss. If there are K classes, $Km \times m$ is the dimension of mask branch output, $(m \times m)$ shows the mask resolution for each ROI. L_{mask} is defined in a way to generate masks for every class that leads to decoupling mask and class predictions in Mask R-CNN.

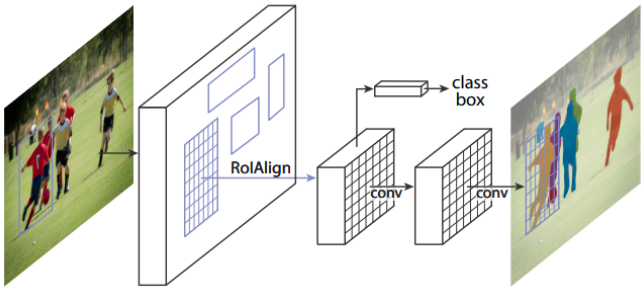


Fig. 1: The Mask R-CNN for instance segmentaion

B. Network Architecture

In the Mask R-CNN structure, feature extraction is done with a convolutional *backbone* architecture and bounding-box recognition and mask prediction which is applied separately to each RoI is done with the network *head*. Two networks are used for the *backbone* architecture, ResNets [4] in which the features extracted from the final convolutional layer of the 4^{th} stage (ResNet-50-C4) and a Feature Pyramid Network (FPN) [5]. The head of the ResNet-C4 backbone is the 5^{th} stage of ResNet (namely, the 9-layer 'res5' [4]). For FPN, the backbone already includes res5 thus the head is more efficient since it uses fewer filters. The details of the head architecture are shown in Figure 2.

The rest of this paper is organized as follows. The performance of Mask-RCNN on image segmentation is assessed in section II and the AP (averaged over IoU thresholds) is reported in this section. In section III, a pre-trained COCO Mask R-CNN model which is trained over balloons is used to detect balloons in grayscale images and colored video. It is also used to train over new datasets to detect and segment the different shapes in images. Conclusions are discussed in Section IV.

II. Validation

A. Training

In this section, the Mask R-CNN is applied to some images that the network has not seen before (test images of the dataset) to assess the validity of the proposed network in predicting the class label with related bounding boxes and

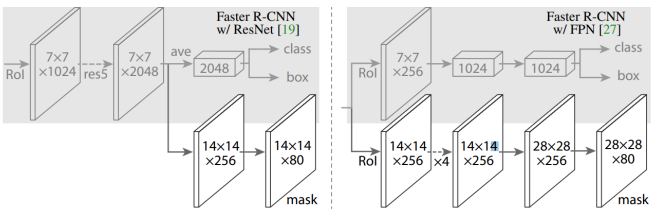


Fig. 2: Head architecture of two different backbone networks

Fig. 3: The structure of JSON annotation of COCO dataset file

segmenting different objects. Mask R-CNN with ResNet-101-FPN backbone is trained on the COCO [6] dataset while using the model pretrained on a 5k-class subset of ImageNet [7]. As the GitHub repository used for duplicating the result of this paper differs from the one presented in the paper, some differences are explained in the following. To support training multiple images per batch, all images are resized to the same size by preserving the aspect ratio. For example, 1024x1024px on MS COCO. So if an image is not square we pad it with zeros. The paper uses a learning rate of 0.02 which is too high, and often causes the weights to explode, especially when using a small batch size. It is found that smaller learning rates converge faster anyway.

Mask R-CNN should be trained on a dataset that has the class label, bounding box, and its associated mask for each object in an image as a ground-truth. The COCO dataset has these features in its annotations file. The annotation of each object in the COCO dataset is shown in Figure 3.

Thus, the network is trained on the COCO dataset and the computed weights are saved and used for detection and segmentation. As shown in Figure 4, Mask R-CNN achieves good results even under challenging conditions. The results show that the objects are detected and segmented correctly. Each object is shown with a bounding box around the object with class prediction. In addition, each object is segmented with a specific color to distinguish among the detected objects in the image.

B. AP

AP (Average Precision) is a popular performance metric in measuring the accuracy of object detection algorithms. Average precision computes the average precision value for recall value over 0 to 1. The general definition for the AP is finding the area under the precision-recall curve.

Precision measures how accurate is the predictions or the percentage of correct positive predictions among all predictions made.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

where $TP = truepositive$, $FP = falsepositive$.

Recall measures how good the algorithm is to find all the positive or the percentage of correct positive predictions among all positive cases in reality.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

where $FN = falsenegative$.

AP_{50} is an average precision at an IoU of 0.5. IoU measures the overlap between 2 boundaries. It is used to measure how much our predicted boundary overlaps with the ground truth (the real object boundary). Here, AP_{50} is computed for all the "minival" datasets which is a small part of the validation dataset that is used for assessing the model performance. The number of images in the "minival" datasets is 4952. The value of AP_{50} for the "minival" datasets is about 0.65 (Figure 5. which differs from the value reported in the paper. The reason is that AP_{50} is calculated on the small part of the dataset, not the whole.

III. Extension

A. Instance segmentation over grayscale images

In this section, it is tried to extend the Mask R-CNN. One of the extension projects is the balloon detection. In this project, some RGB balloon images are collected and then annotated with VIA (VGG Image Annotator) to have the bounding box and segmentation mask for each object. Here, transfer learning is applied which means that instead of training a model from scratch, the weights of trained Mask R-CNN on the COCO dataset are used as initial weights to train the network on the balloon dataset. The COCO dataset does not contain a balloon class, but it contains a lot of other images so the trained weights have already learned a lot of the features common in natural images, which really helps.

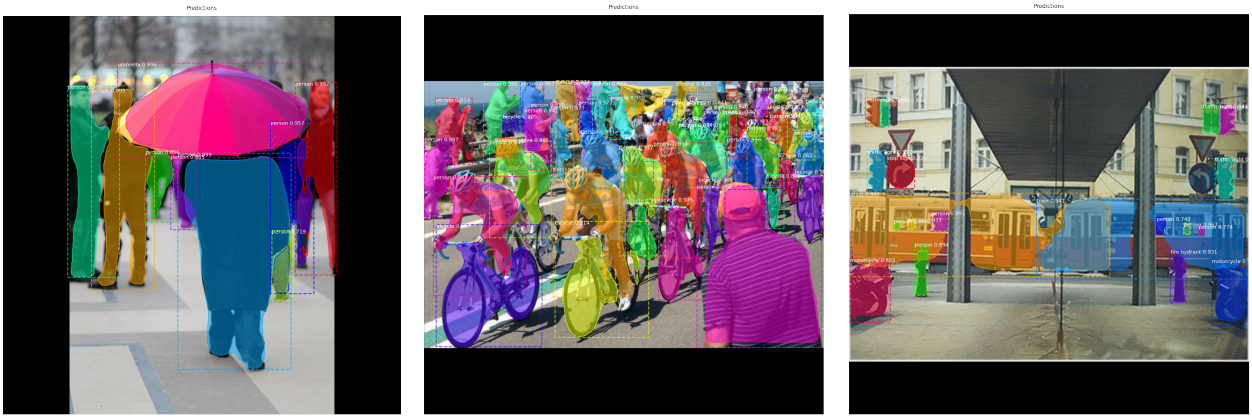
These obtained weights are used to detect the ballon in RGB images. For the extension, I changed some of the images to grayscale to see that the Mask R-CNN can detect balloons in these kinds of images.

As shown in Figure 6, Mask R-CNN is not able to detect all balloons in grayscale images. In the grayscale images, if the intensities of balloons and backgrounds are close to each other so that it is hard to distinguish between them in grayscale images, this network is unable to detect the balloon in this challenging condition. It is shown that detecting the small size of balloons is also difficult for Mask R-CNN.

B. Instance segmentation over video

Mask R-CNN is also can applied to the video. For this purpose, I have tried to find a video having balloons in it. Then, the fps (frame per second) of the video is computed, and then it is converted to a list of images. The pre-trained Mask R-CNN over balloon dataset is used over this list and the segmentation mask is applied to them. Then, images with object masks are obtained. Thus, the color splash effect can be applied to them in such a way that a grayscale version of the image is created, and areas marked by the object mask are copied back to the color pixels from the original image. The video that I made by applying Mask R-CNN segmentation and color splash effect is available in https://youtu.be/fqo2Y44vT_g. The original video is in https://youtu.be/QvaRmEZaUuU. The one frame of this video is shown in Figure 7.

(a) class labels: Zebra and Elephent (b) class labels: Sheep (c) class labels: Person, Skis



(d) class labels: Person, Umbrella (e) class labels: Bicycle, Person (f) class labels: Train, Traffic light, Stop sign, Motorcycle, Fire hydrant

Fig. 4: Mask R-CNN results on the COCO test set. These results are based on ResNet-101-FPN

Compute mAP @ IoU=50 on Batch of Images

```
# Compute VOC-style Average Precision
def compute_batch_ap(image_ids):
    APs = []
    for image_id in image_ids:
        # Load image
        image, image_meta, gt_class_id, gt_bbox, gt_mask =\
            modellib.load_image_gt(dataset, config,
                                   image_id)
        # Run object detection
        results = model.detect([image], verbose=0)
        # Compute AP
        r = results[0]
        AP, precisions, recalls, overlaps =\
            utils.compute_ap(gt_bbox, gt_class_id, gt_mask,
                             r['rois'], r['class_ids'], r['scores'], r['masks'])
        APs.append(AP)
    return APs

# Pick a set of random images
# print(len(dataset))
image_ids = np.random.choice(dataset.image_ids, len(dataset.image_ids))
APs = compute_batch_ap(image_ids)
print("mAP @ IoU=50: ", np.mean(APs))
```

```
mAP @ IoU=50:  0.6498105340244482
```

Fig. 5: The AP_{50} over the batch of images

C. Mask R-CNN trained on shapes dataset

Mask R-CNN which is already trained on the COCO dataset is used to train on the shapes dataset. For this purpose, the weights trained on MS COCO are used to train a model on the synthetic shapes dataset (squares, triangles, circles and ellipses) which enables fast training. For generating the shapes synthetic dataset, it uses OpenCV to draw different geometric shapes within the given height and width boundaries. This method generates images on the fly and no file access is required.

The codes for generating the different shapes in OpenCV just need the corner points of the rectangle and triangle, the center point of the circle and ellipse with axes lengths of ellipse, and the radius of the circle. This model is trained on the new shapes dataset to detect and segment these shapes. The performance of this network is illustrated in figure 8. As Mask R-CNN has already trained the features of different objects in the MS COCO dataset and these shapes datasets are simple, it is just enough to train the head layers and freeze all the backbone layers.

IV. Conclusion

In this paper, it is tried to use Mask R-CNN in new images to check the validity of this network performance. The average precision of this network on the small part of the test dataset is also computed. The proposed network which is trained on the balloons dataset is applied to detect balloons on grayscale images and colored video. The result shows that it has a good performance. However, This network has difficulty in detecting balloons in some conditions where the intensities of the balloon and background are close to each other in grayscale images. In addition, Mask R-CNN is also trained

(a) RGB image (b) Grayscale image

Fig. 6: Instance segmentation of Mask R-CNN over RGB vs Grayscale images

Fig. 7: One frame of video that is generated with instance segmentation of Mask R-CNN and color splash method

on the synthetic shapes dataset to detect four different shapes including squares, triangles, circles and ellipses in images. The accuracy of this trained network is also assessed. This network is powerful and if it is trained on a sufficient number of new objects it can detect them.

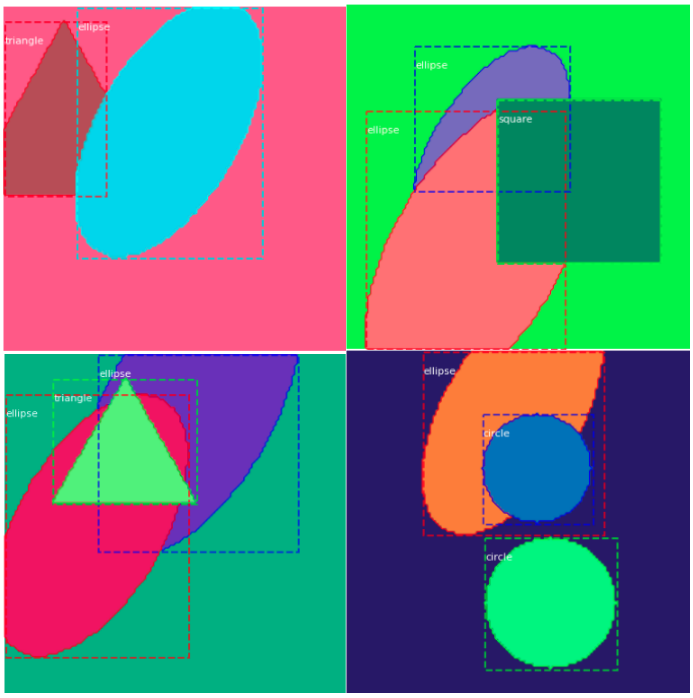


Fig. 8: Instance segmentation of Mask R-CNN on synthetic shapes dataset

REFERENCES

[1] K. He, G. Georgia, D. Piotr, and G. Ross. "Mask r-cnn." In Proceedings of the IEEE International Conference on Computer Vision, pp. 2961-2969. 2017.
[2] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.
[3] R. Girshick. Fast R-CNN. In ICCV, 2015.
[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016.
[5] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In CVPR, 2017.
[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In ECCV, 2014.
[7] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, 2012.