Deep learning based Object Detection and Instance Segmentation using Mask R-CNN in OpenCV

Our previous achievement was Objrct detection using YOLOv3.

The output of an object detector is an array of bounding boxes around objects detected in the image or video frame, but we do not get any clue about the shape of the object inside the bounding box.

Now my attempt is a Convolutional Neural Network (CNN) model called Mask-RCNN (Region based Convolutional Neural Network) for object detection and segmentation. Using Mask-RCNN we not only detect the object, we also obtain a greyscale or binary mask containing the object.

[Mask-RCNN](https://arxiv.org/pdf/1703.06870.pdf) was initially introduced in Nov 2017 by Facebook’s AI Research team using [Python and Caffe2](https://github.com/facebookresearch/Detectron).

## **What is Image Segmentation?**

In computer vision, the term “image segmentation” or simply “segmentation” means dividing the image into groups of pixels based on some criteria. You can do this grouping based on color, texture, or some other criteria that you have decided. These groups are sometimes also called super-pixels.

### **What is instance segmentation?**

In **instance segmentation** the goal is to detect specific objects in an image and create a mask around the object of interest. Instance segmentation can also be thought as object detection where the output is a mask instead of just a bounding box. Unlike **semantic segmentation**, which tries to categorize each pixel in the image, instance segmentation does not aim to label every pixel in the image.

Below we see an example of instance segmentation of two sheep on a very similar colored background.

  
Figure : Instance Segmentation Example

Mask-RCNN is a result of a series of improvements over the original R-CNN paper (by R. Girshick et. al., CVPR 2014) for object detection. [R-CNN](https://arxiv.org/abs/1311.2524)generated region proposals based on selective search and then processed each proposed region, one at time, using Convolutional Networks to output an object label and its bounding box.

The Mask-RCNN network has two major parts.

The first one is the Region Proposal Network which generates around 300 region proposals per image. During training, each of these proposals (ROIs) go through the second part which is the object detection and mask prediction network, as shown above. Note that since the mask prediction branch runs in parallel to the label and box prediction branch, for each given ROI, the network predicts masks belonging to all the classes.

During inference, the region proposals go through Non-Maximum Suppression and only the top scoring 100 detection boxes are processed by the mask prediction branch. So, with 100 ROIs and 90 object classes, the mask prediction part of the network outputs a 4D tensor of size 100x90x15x15, where each mask is of size 15×15.

For the sheep image shown above, the network detected two objects. For each object it outputs an array containing the predicted class score(indicating the probability that the object belongs to the predicted class), left, top, right and bottom locations of the bounding box of the detected object in the frame. The class id from this array is used to extract the corresponding mask from the output of the mask prediction branch. The masks for the two objects detected look like the following:



*Figure : Masks produced by Mask-RCNN*

These masks can then be thresholded to get a completely binary mask. Also Compared to other object detectors like YOLOv3, the network of Mask-RCNN runs on larger images.