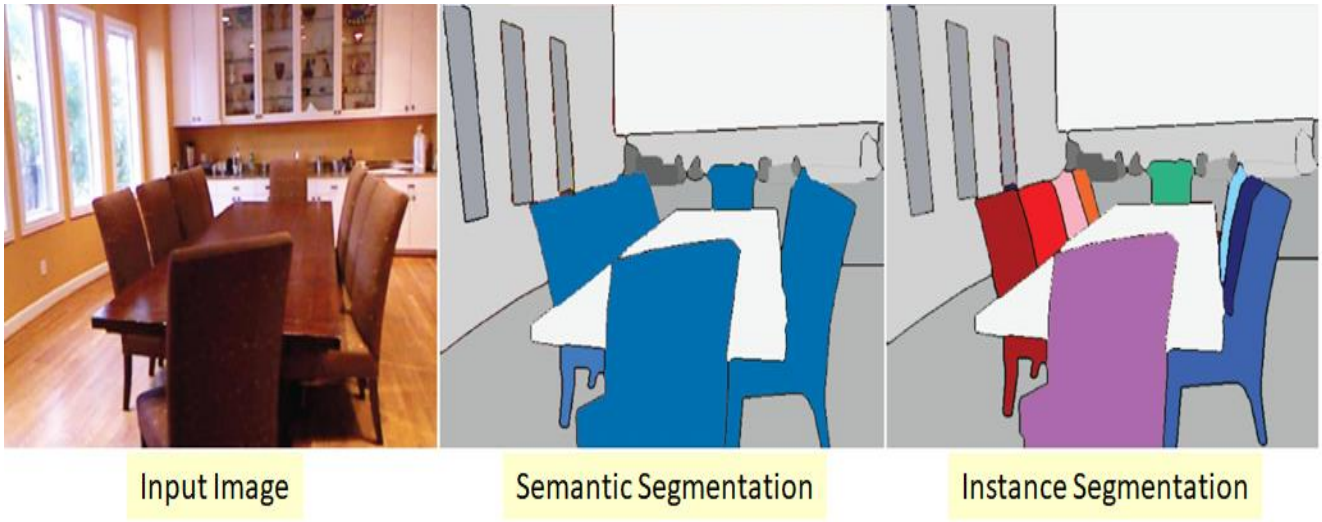
Image Segmentation

Topics Covered

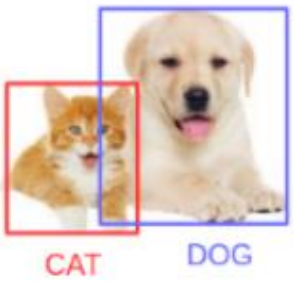
1. Mask RCNN Introduction
2. Architecture of Mask RCNN
3. Summary

Mask R-CNN

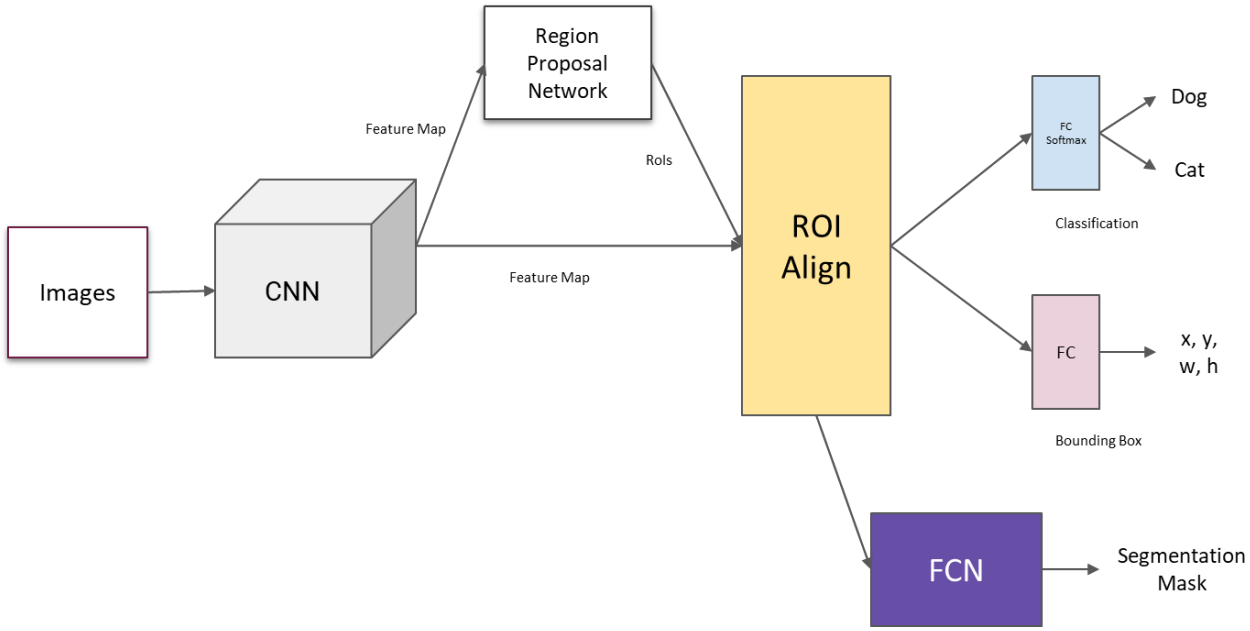


Mask R-CNN is basically an extension of Faster R-CNN. Faster R-CNN is widely used for object detection tasks. For a given image, it returns the class label and bounding box coordinates for each object in the image. So, let's say you pass the following image:

The Fast R-CNN model will return something like this:



The Mask R-CNN framework is built on top of Faster R-CNN. So, for a given image, Mask R-CNN, in addition to the class label and bounding box coordinates for each object, will also return the object mask.



Let's first quickly understand how Faster R-CNN works again. This will help us grasp the intuition behind Mask R-CNN as well.

- Faster R-CNN first uses a ConvNet to extract feature maps from the images
- These feature maps are then passed through a Region Proposal Network (RPN) which returns the candidate bounding boxes

- We then apply an RoI pooling layer on these candidate bounding boxes to bring all the candidates to the same size
- And finally, the proposals are passed to a fully connected layer to classify and output the bounding boxes for objects

Once you understand how Faster R-CNN works, understanding Mask R-CNN will be very easy. So, let's understand it step-by-step starting from the input to predicting the class label, bounding box, and object mask.

Backbone Model

Similar to the ConvNet that we use in Faster R-CNN to extract feature maps from the image, we use the ResNet 101 architecture to extract features from the images in Mask R-CNN. So, the first step is to take an image and extract features using the ResNet 101 architecture. These features act as an input for the next layer.

Region Proposal Network (RPN)

Now, we take the feature maps obtained in the previous step and apply a region proposal network (RPM). This basically predicts if an object is present in that region (or not). In this step, we get those regions or feature maps which the model predicts contain some object.

Region of Interest (RoI)

The regions obtained from the RPN might be of different shapes, right? Hence, we apply a pooling layer and convert all the regions to the same shape. Next, these regions are passed through a fully connected network so that the class label and bounding boxes are predicted.

Till this point, the steps are almost similar to how Faster R-CNN works. Now comes the difference between the two frameworks. In addition to this, **Mask R-CNN also generates the segmentation mask.**

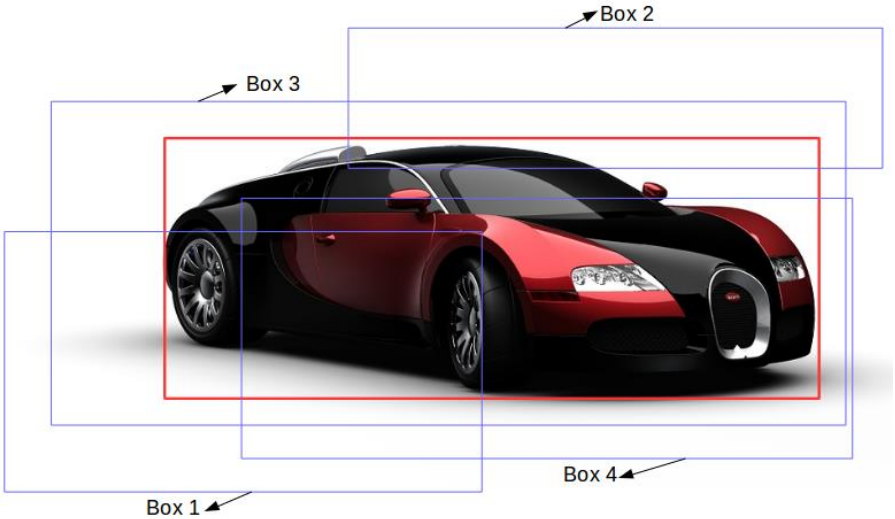
For that, we first compute the region of interest so that the computation time can be reduced. For all the predicted regions, we compute the Intersection over Union (IoU) with the ground truth boxes. We can computer IoU like this:

IoU = Area of the intersection / Area of the union

Now, only if the IoU is greater than or equal to 0.5, we consider that as a region of interest. Otherwise, we neglect that particular region. We do this for all the regions and then select only a set of regions for which the IoU is greater than 0.5.

Let’s understand it using an example. Consider this image:

Here, the red box is the ground truth box for this image. Now, let’s say we got 4 regions from the RPN as shown below:



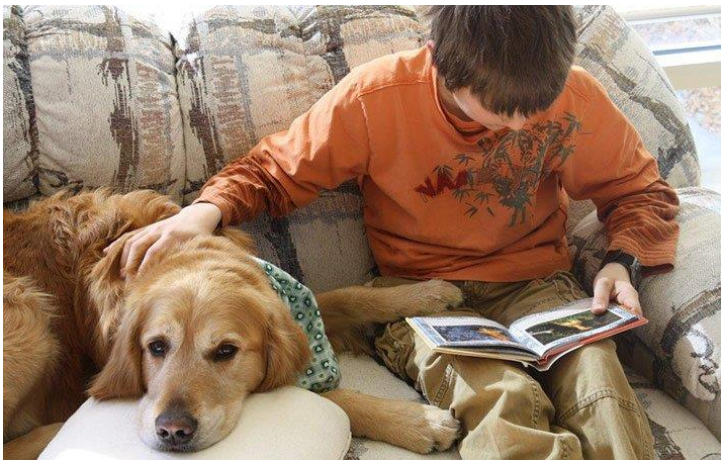
Here, the IoU of Box 1 and Box 2 is possibly less than 0.5, whereas the IoU of Box 3 and Box 4 is approximately greater than 0.5. Hence. we can say that Box 3 and Box 4 are the region of interest for this particular image whereas Box 1 and Box 2 will be neglected.

Next, let's see the final step of Mask R-CNN.

Segmentation Mask

Once we have the RoIs based on the IoU values, we can add a mask branch to the existing architecture. This returns the segmentation mask for each region that contains an object. It returns a mask of size 28 X 28 for each region which is then scaled up for inference.

Again, let's understand this visually. Consider the following image:



The segmentation mask for this image would look something like this:

Here, our model has segmented all the objects in the image. This is the final step in Mask R-CNN where we predict the masks for all the objects in the image.

Summary

- Mask R-CNN is basically an extension of Faster R-CNN. Faster R-CNN is widely used for object detection tasks. For a given image, it returns the class label and bounding box coordinates for each object in the image
- Mask RCNN uses a decoder network addition to faster RCNN to construct masks
- Mask RCNN uses ROI Align instead of ROI Pooling.