Understanding Mask R-CNN

Keywords: COCO, Faster R-CNN, object detection, instance segmentation, ROIAlign, bilinear interpolation, keypoint

In computer vision, object detection is concerned with correctly labeling an object and applying a bounding box around it. On the other hand, image segmentation focuses on pixel-to-pixel classification. The intersection of these two tasks can be represented by mask R-CNN. It is practically an instance segmentation.

To understand mask R-CNN, we first need to appreciate the idea behind Faster R-CNN, which serves as the baseline. Faster R-CNN is a region-based model, hence the R prefix. The general pipeline is as follows: it first extracts a feature maps from the input using CNN, which is passed to the next layer to generate region proposals (potential locations where objects may be); each region proposal is then transformed into feature vector of the same dimension, which is done using ROI pooling; these feature vectors are fed to the softmax function and bounding box regression to determine the location and label of objects.

The feature extraction stage utilizes existing neural network backbone such as ResNet. In addition, ROIs are outputted using Region Proposal Network (RPN), which are essentially neural networks that output potentially valid bounding boxes. The process is as follows: it moves a scanning window across the feature map, and generates multiple ROIs using fixed anchors that dictate the scale and aspect ratio of the window. Each of these ROIs is then given an objectness score, which measures the likelihood that an object is present within the region. If the objectness is positive (meaning object), the ROI is outputted. During training, the ground truth objectness score is based on Intersection of Union (IoU), which measures the overlap between the ground truth bounding box and the proposed ROI. The general rule of thumb is that if IoU is greater than 0.5, the objectness score is positive, and vice versa. RPN is fine-tuned until it can predict objectness score reasonably accurately.

Mask R-CNN builds on top of this by adding an extra branch called mask that essentially takes feature vectors as inputs and determines its mask, or what is simply called, pixel-to-pixel classification map. Another difference is that rather than using ROI pooling to obtain these feature vectors, mask R-CNN uses something called ROI Align in order to preserve spatial relationships. Since mask is another type of representation of the original input features, it cannot be outputted using collapsed feature vectors that do not preserve spatial information. We will explore each of these two differences in detail next.

Traditional ROI pooling utilizes quantization to resize the region proposal relative to the new size of the feature map. The dimension of such resized ROI is rounded to the nearest integer, in which case it creates potential misalignment between the original ROI and the resized ROI. It then divides the new ROI into sections (e.g., 2 x 2) and utilizes max pooling on each, which, in my own opinion, creates further loss in spatial information. On the other hand, ROI Align has set out to address these drawbacks. First, it does not perform quantization during resizing: ROI coordinates are simply changed into continuous scale if necessary. After the new ROI is divided into sections, it then performs bilinear interpolation upon each section by taking four points equally spaced out. When the value of each location is found, we can aggregate by taking either the average or the max. Experiments have also shown the key here is no quantization during resizing since ROI warp, which performs bilinear interpolation as well, does not obtain comparable AP (average precision).

With features extracted, the next stage feeds them to the prediction branch. During training, the box prediction branch first processes these features and applies class labels and corresponding bounding boxes. The class label is subsequently used by the mask branch to generate a class-specific mask, meaning that it only applies the K-th class activation function for the mask if the class label is K. This is called decoupling of the mask and box prediction branch, which increases speed since it only performs a single mask prediction. In contrast, older models with coupling prediction branches will have to generate masks for all classes, adding up computation time.

It is worth noting that during inference, mask R-CNN only sends 100 ROI with the highest confidence score to the mask branch, which increases its accuracy and speed. This is not done during training simply because more ROI mask predictions mean more opportunities for fine-tuning hyperparameters.

Mask R-CNN also generalizes to other tasks like human pose estimation. This is achieved by adding an extra keypoint branch alongside the box and mask branch. Similarly, the keypoint branch predicts a mask based on class label prediction from the box branch: if the class label is a person, the keypoint branch goes on predicting a one-hot mask for each of these key points. It is also common to implement a class-agnostic prediction that generates a mask encompassing all keypoint classes.

Mask R-CNN runs each frame for 200ms (5 frames per second) on a single GPU. For a 8-GPU machine, this gives an effective batch of 40 frames per second. Plus, it takes two days to train, which is a dramatic speedup from previous state-of-art models. It also outperforms previous COCO dataset winner.

# Human Pose Estimation

The above mask R-CNN extension to human pose estimation falls under a type of approach called top-down. It usually involves detecting the person instance first and then predicts the keypoint mask within that person bounding box. There is another approach type called bottom-up, which instead directly localizes keypoint within an image and connects them together to form pose estimation. Usually, this association is done based on proximity or some other criteria. One may think of the latter approach as a vanilla object detection model that focuses on keypoint only. Top-down approach should be more efficient since it only works on a smaller region, while the bottom-up approach has to search the entire image, but it is more suited for images where there are lots of human instances, in which case it would be expensive to first detect all persons first.