Faster RCNN

Summary

One of the most popular deep learning-based object detection algorithms is the family of R-CNN algorithms, introduced by Girshick. He provides a basic framework for object detection with deep learning. Since then, many variants advent as improvements. The faster R-CNN algorithm and its components include anchors, the base network, the region proposal network, and region of interest pooling. The building blocks help us understand the core algorithm, how it works, and how end-to-end deep learning object detection is possible. Hence, we will go along with the logic step by step.

Object Detection and Deep Learning

Given an input image, we wish to obtain:
1. A list of bounding boxes
2. A class label
3. The probability / confidence score

The traditional object detection pipeline
1. Sliding windows to localize objects at different locations
2. Image pyramids, used to detect objects at varying scales
3. Classification via a pretrained CNN

Disadvantages:
Slow and tedious
Lack of aspect ratio
Error prone

Measure

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

Ps: the mainstream algorithms are Faster/Mask R-CNN, SSD, YOLO
In order to apply IoU to evaluate an arbitrary object detector, we need:
1. The ground-truth bounding boxes
2. The predicted bounding boxes from our model
3. Class label

Where do the ground-truth examples come from?
Hand-labeled
To summarize, the goal is to take the training images + bounding boxes,

construct an object detector, and then evaluate its performance on the testing set. An IoU score > 0.5 is normally considered a good prediction.

The (Faster) R-CNN Architecture
Evolution of R-CNN
R-CNN: Regions with CNN features
1. Input image
2. Extract region proposals (selective search)
3. Compute CNN features
4. Classify regions

Here the core step is how to generate region proposals.

Fast R-CNN
A novel contribution was made: Region of Interest (ROI) Pooling
The network is effectively end-to-end trainable:
1. We input an image and associated ground-truth bounding boxes
2. Extract the feature map
3. Apply ROI pooling and obtain the ROI feature vector
4. And finally use two sets of fully-connected layers to obtain (1) the class label predictions and (2) the bounding box locations for each proposals.
The performance suffers dramatically at inference.

Faster R-CNN
Replace the Selective Search with Region Proposal Network(RPN).
The first component, the RPN, is used to determine where in an image a potential object could be. Just there is potentially an object at a certain location in the image.

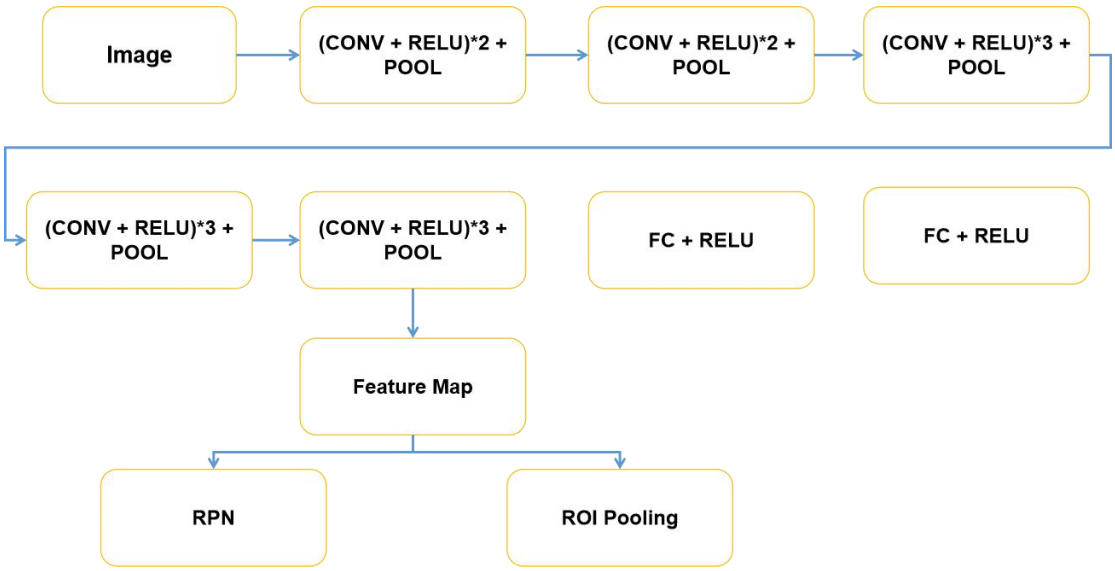
The proposed bounding box ROIs are based on the Region of Interest (ROI) pooling module of the network along with the extracted features from the previous step.

Note that the RPN is not actually labeling the ROI. It is computing its objectness score. It is just like a binary classifier. Background or foreground

The complete pipeline is
1. Region proposal
2. Feature extraction
3. Computing the bounding box coordinates of the objects
4. Providing class labels for each bounding box

The base network is a pretrained CNN to extract feature

A fully-convolutional nerual network enjoys two primary benefits
1. Fast, due to all convolution operations
2. Able to accept images of any spatial resolution, provided that the image and network can fit into memory.



Anchors

In traditional object detection pipelines we would use either
1. A combination of a sliding window + image pyramid or
2. A selective search-like algorithm to generate proposals for our classifier.

The core separation between classification and object detection is the prediction of bounding boxes, or (x, y)-coordinates surrounding an object.

How do we handle a network predicting values outside the boundaries of the image?
How do we encode restrictions such as xmin < xmax and ymin < ymax?

Instead of trying to predict the raw (x, y)-coordinates of the bounding boxes, we can instead learn to predict their offsets from the reference boxes, the delta values allow us to obtain a better fit to our reference box without having to predict the actual raw (x, y)-coordinates, enabling us to bypass the potentially impossible problem of encoding bounding box “rules” into the network.

Hence, we need to generate the anchors ourselves without utilizing a selective search algorithm. To accomplish this process, we first need to uniformly sample points across an input image.
The nest step is to create a set of anchors at each of the sampled points. We will generate nine anchors with varying sizes and aspect ratios surrounding a

given sampled point.

The colors of the bounding boxes are our scales/sizes: 64 x 64, 128 x 128, 256 x 256. For each scale we also ave the aspect ratio, 1: 1, 1: 2, 2: 1. Then, we can get nine total anchors.

Region Proposal Network
The goal of generating anchors is to obtain good coverage over all possible scales and size of objects in an image, the goal of the Region Proposal Network is to prune the number of generated bounding boxes to a more manageable size.

Provided that our foreground probability is sufficiently large, we the apply
Non-maxima suppression to suppress overlapping
Proposal selection

Training the RPN
The RPN module has two loss functions associated with it. The first one is for classification which measures the accuracy of the RPN predicting foreground vs background.

The second one is for our bounding box regression. This loss function only operates on the foreground anchors as background anchors would have no sense of a bounding box.

Region of Interest (ROI) Pooling
The goal of the ROI pooling module is to accept all N proposal locations from the RPN module and crop out feature vectors from the convolutional feature map.
Using array slicing to extract the corresponding patch from the feature map
Resize it to 14x14xD where D is the depth of the feature map
Applying a max pooling operation with 2x2 strides, yielding a 7x7xD feature vector
The final feature vector obtained from the ROI pooling module can be fed into the region-based convolutional neural network which we will use to obtain the final bounding box coordinates.

Region-based Convolutional Neural Network
1. Obtain the final class label predictions for each bounding box location based on the cropped feature map from the ROI Pooling module.
2. Further refine the bounding box prediction (x, y)-coordinates for better prediction accuracy

The novel idea is IoU and RPN.