A Literature Review for Object Detection Using CNN

ZHOU YINAN
yinan.zhou@mail.polimi.it

April 10, 2018

Abstract

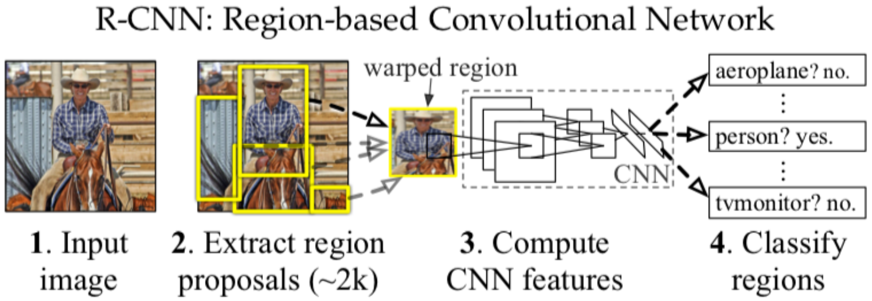
This is my literature review for object detection using deep convolutional network. More specifically, this paper contains my understanding for R-CNN, SPP-net, Fast R-CNN, Faster R-CNN.

1 Introduction

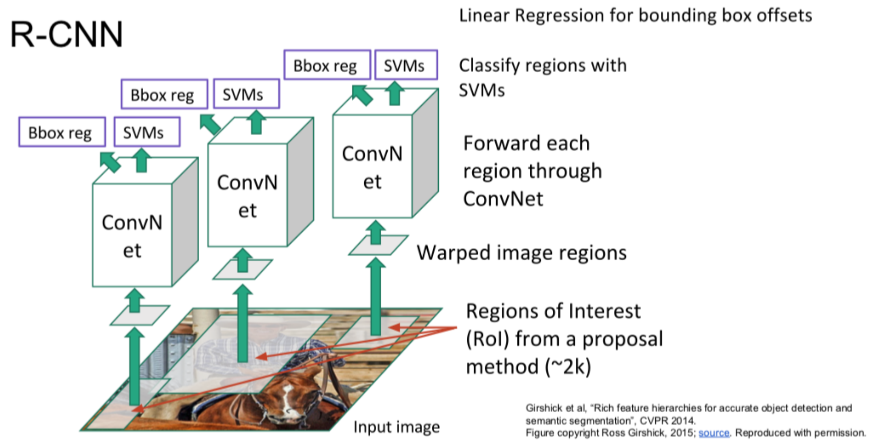
Object detection is a two-task challenge: object localization and object classification. For object localization, the most straight forward idea is to iterate all possible bounding boxes in the image. However it is way too much computation. Therefore people invent smart algorithms for region proposals. After generating proposals, the algorithm should learn how to improve it to match the ground truth bounding box. We treat the bounding box prediction as a regression problem. Each boundng box can be determined by a four tuple (x,y,w,h). As for classification, we use the standard image classification method in CNN. The main difference in the following algorithms lies in **how we extract region proposals** and **how to exploit proposal spatial information efficiently**.

2 R-CNN

R-CNN is the first deep convnets for object detection which achieves state-of-art performance. The original version of R-CNN is accurate but slow and consumes a lot of disk memory. After R-CNN, several modifications are made, such as SPP-net, Fast R-CNN, Faster R-CNN. Thus the original R-CNN is also called "Slow" R-CNN. The pipeline for R-CNN is the following:



R-CNN: Region-based Convolutional Network



R-CNN

Linear Regression for bounding box offsets
Bbox reg
SVMs
Classify regions with SVMs
ConvNet
Forward each region through ConvNet
Warped image regions
Regions of Interest (RoI) from a proposal method (~2k)
Input image

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

1. Given each input image, extracts around 2k proposals. These proposals are called Regions of Interest(RoI). There are many existing methods for extracting RoI. The idea is that RoI extraction is category independent and it is a fixed function. We do not learn how to extract RoI in R-CNN architecture. The authors in the paper uses selection search for RoI extraction.

2. Each RoI has different shape and they can not be forwarded into a normal CNN. Thus we need to warp RoI to fixed size.

3. All the warped image regions are forwarded into ConvNet and outputs a fixed length feature vector.

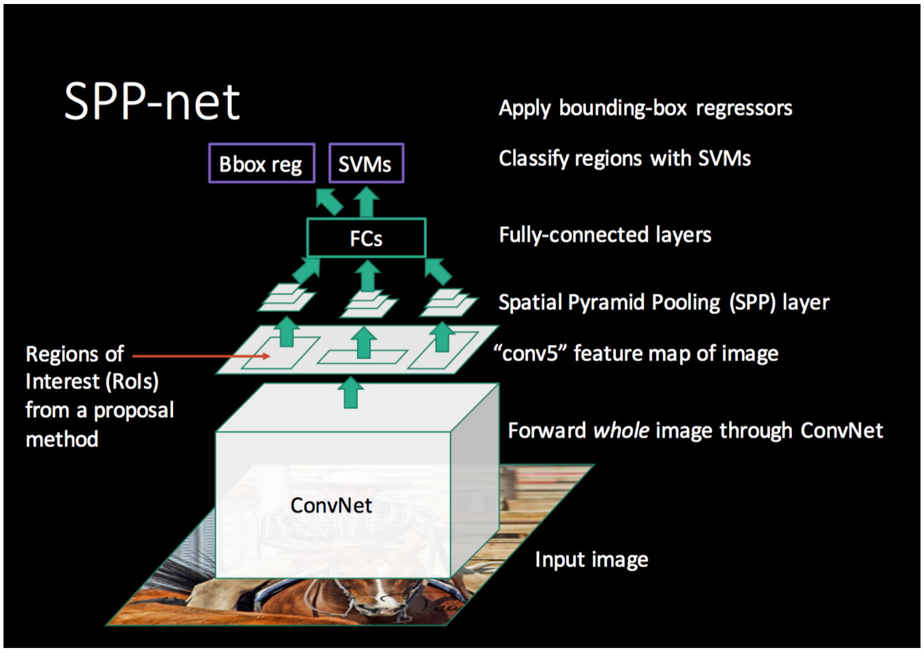
4. We train different SVM for different classes. Each region is scored and classified by these SVM.

5. For object localization, we treat it as a regression problem. After scoring each selective search proposal with a class-specific detection SVM, we predict a new bounding box for the detection using a class-specific bounding-box regressor. Our goal is to learn a transformation that maps a proposed box P to a ground-truth box G.

Despite the start-of-art performance for R-CNN, it has some drawbacks. Training and testing are both slow. The architecture is not end-to-end. We need to perform several training: fine-tune network with softmax classifier, post-hoc linear SVMs, post-hoc bounding-box regressors.

3 SPP-net

R-CNN is slow because given an input image, we need to extract many RoIs. These RoIs most likely to overlap with each other. But the ConvNet needs to re-compute all of them. Thus what we can do to improve the speed is to swap the order of RoI extraction and Convnet. This is exactly what SPP-net does. Also SPP-net invents a ”Spatial Pyramid Pooling” layer to deal with different scales of RoIs. The pipeline for SPP-net is the following:



SPP-net

Apply bounding-box regressors

Classify regions with SVMs

Fully-connected layers

Spatial Pyramid Pooling (SPP) layer

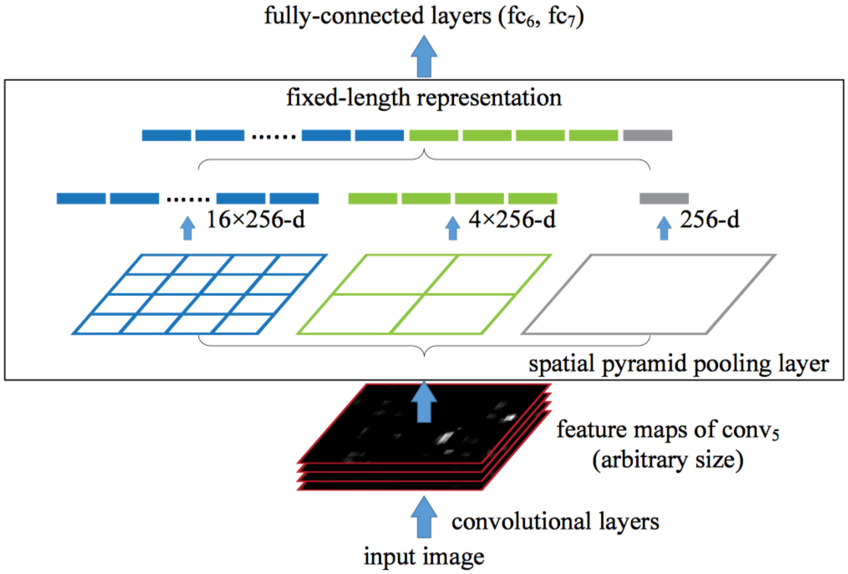
“conv5” feature map of image

Forward *whole* image through ConvNet

Input image

Regions of Interest (RoIs) from a proposal method

1. Forward whole image through ConvNet
2. Use a predefined proposal method to extract RoIs from feature map of image.
3. Use SPP layer to extract fixed length feature vectors
4. Forward the feature vector to fully connected layers
5. Use SVMs for proposal classification and bounding box regressor for localization

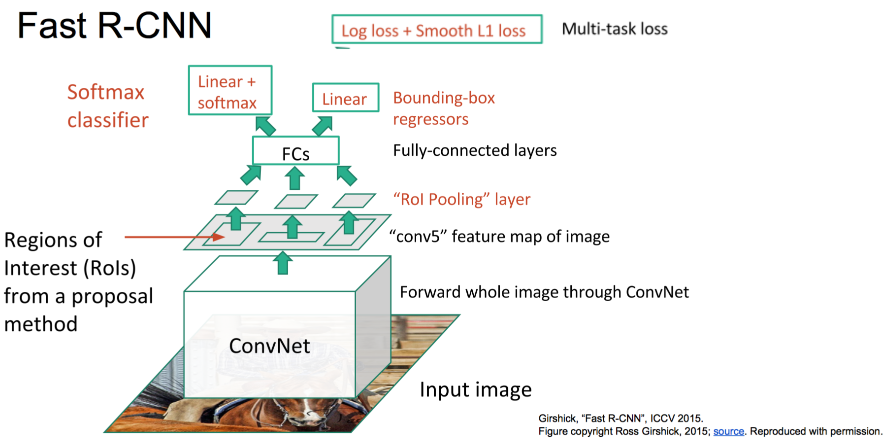

fully-connected layers (fc_6, fc_7)
fixed-length representation
16×256-d
4×256-d
256-d
spatial pyramid pooling layer
feature maps of $conv_5$
(arbitrary size)
convolutional layers
input image

Here I want to say more about the SPP later. ”Spatial Pyramid Pooling” is actually a general technique which can not only be applied to object detection, but also general cases of image classification using ConvNet. SPP-layer can generate a fixed length representation of image size/scale regardless of the input size. So by using SPP layer, we can release the ”fixed size input” constraint for ConvNet.

SPP-net makes testing fast but also introduces new problems. The original design for SPP- net does not define a differentiable function for SPP-layer. Thus, it can not update parameters below SPP layer during training. So the ConvNet is frozen which affects the performance.

4 Fast R-CNN

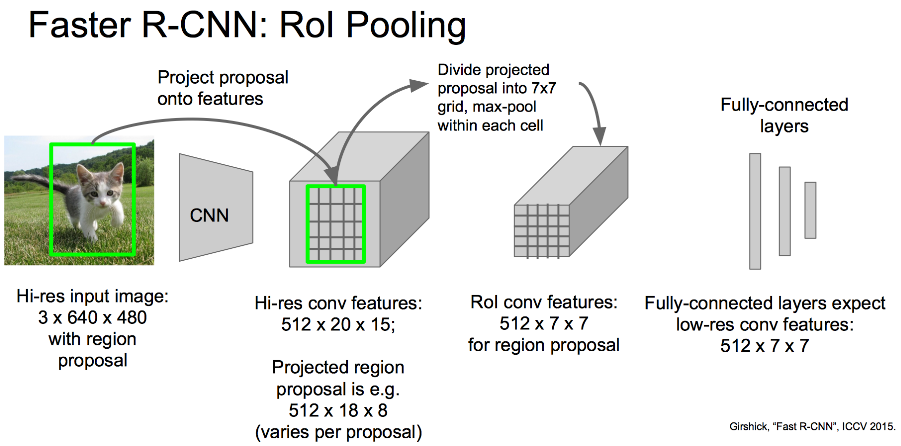
Fast R-CNN deals with the problem of frozen ConvNet layers in SPP-net. Instead of using ”Saptial Pyramid Pooling”, it uses ”RoI Pooling”. RoI pooling layer is differentiable and thus the whole network can be trained. Also notice that in R-CNN and SPP-net, we train SVMs and Bounding box regressors seperately which is time consuming. Here in Fast R-CNN, we have one network, and can be trained end- to-end. We do not train classifier and bounding box predictor seperately. Instead, we use a ”multi-task loss” to combine the classification and regression loss. The pipeline for Fast R-CNN is the following:



Fast R-CNN

Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

1. Forward the whole image through ConvNet
2. Extract RoIs using pre-defined proposal method.
3. RoI pooling to extract feature vectors
4. Forward feature vectors to fully connected layers
5. multi-task loss calculator



Faster R-CNN: RoI Pooling

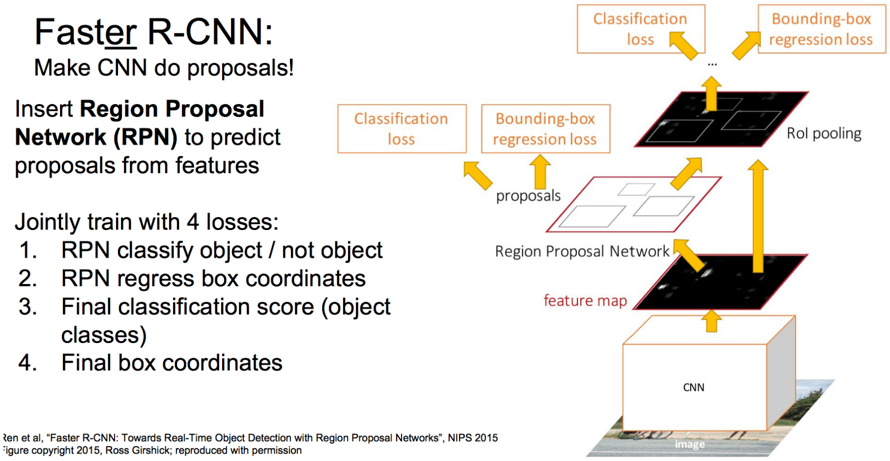
Project proposal onto features
Divide projected proposal into 7x7 grid, max-pool within each cell
Fully-connected layers
CNN
Hi-res input image: 3 x 640 x 480 with region proposal
Hi-res conv features: 512 x 20 x 15;
Projected region proposal is e.g. 512 x 18 x 8 (varies per proposal)
RoI conv features: 512 x 7 x 7 for region proposal
Fully-connected layers expect low-res conv features: 512 x 7 x 7

Girshick, "Fast R-CNN", ICCV 2015.

RoI pooling layer is actually a single SPP layer. It uses max pooling to convert the features inside any valid region of interest into a small feature map with fixed spatial extent.

5 Faster R-CNN

Fast R-CNN actually works so well that the bottle neck is region proposals. All the three methods above: R-CNN, SPP-net, Fast R-CNN use a predefined fixed function for RoI extraction. Faster R-CNN suggests that we could also learn how to extract RoIs using deep learning. Faster R-CNN inserts a Regional Proposal Network(RPN) after the last convolutional layer. RPN trained to produce region proposals directly, thus there is no need for external region proposals. After RPN, use RoI pooling , an upstream classifier and bbox regressor just like Fast R-CNN. The pipeline for Faster R-CNN is the following:



Faster R-CNN:

Make CNN do proposals!

Insert **Region Proposal Network (RPN)** to predict proposals from features

Jointly train with 4 losses:

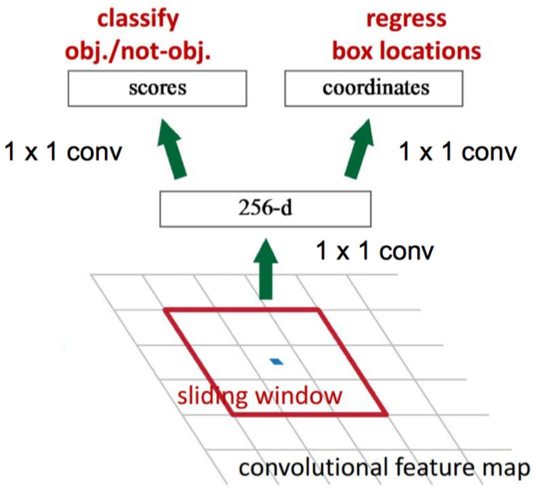
1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates

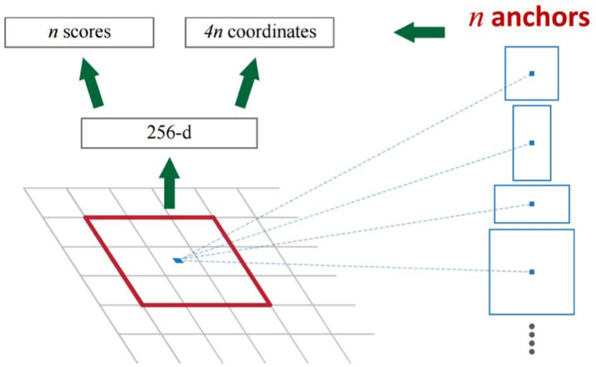
Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

1. Forward input image through ConvNet
2. Feature maps pass through Region Proposal Network(RPN) to extract RoIs
3. RoI pooling layer
4. classifier

5.1 RPN

Slide a small window on the feature map. This window is 1x1 convolution. We build a small network for classifying object or not-object and regressing bbox locations. The center of the sliding window provides localization information with reference to the original input image. More precisely, we extract N anchors in the original image.


classify
obj./not-obj.
regress
box locations
scores
coordinates
1 x 1 conv
1 x 1 conv
256-d
1 x 1 conv
sliding window
convolutional feature map


n scores
4n coordinates
n anchors
256-d