Coding assignment 2. Based on “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”

by haoqing Ren, Kaiming He, Ross Girshick , Jian Sun.

Report by Golobokov Mikhail

ABSTRACT

The following paper introduce us the neural network Faster R-CNN. And it’s instruments: region proposal networks, translation-invariant anchors, a loss function for learning region proposals, optimization problems and etc. The core Version of R-CNN family is faster R-CNN. This family of the Object Detection Networks was one of the most accurate for the last few years. And even right now the best results achieved on MS COCO belongs to the Cascade Mask R-CNN, one of the members of this family.

1. RCNN – WHAT IS IT?

The R-CNN (Regions with CNNs) architecture was created by a UC Berkley for to applying Convolution Neural Networks to the object detection task. Existing at that time approaches to solving such problems came close to the maximum of their capabilities and significantly improve their performance did not work.

CNN showed representable results in classifying images, and in R-CNN they were essentially applied to the same thing. The input of R-CNN was fed not the entire image, but pre-selected regions in another way, which presumably have some objects. The CNN they were using was CaffeNet (AlexNet). R-CNN was developed for object detection of 20 or 200 classes, that was why the last classification layer of CaffeNet was replaced with a layer with N+1 outputs (with an additional class for the background).

Shortly about the procedure for detecting objects by the network:

a. Selecting candidate regions using Selective Search;
b. Converting a region to a size accepted by CNN CaffeNet;
c. Obtaining a CNN 4096-dimensional feature vector;
d. Conducting N binary classifications of each feature vector using N linear SVM;
e. Linear regression parameters of the frame in the region for more accurate coverage of the object.

2. Fast R-CNN

Despite good results, the performance of R-CNN was low, especially for deeper neural networks. In addition, learning bounding box regressor and SVM required saving a large number of features to disk, so it was expensive in terms of storage size.

Fast R-CNN were boosted by few modifications:

- Pass through CNN not each of the 2000 candidate regions separately, but the entire image as a whole. The proposed regions are then superimposed on the resulting General feature map;
- Take the independent training of three models (CNN, COM, box regressor) and combine all training procedures in one.

The transformation of features that fell into different regions to a fixed size was performed using the RoIPooling procedure. Binary SVM were not used, instead the selected features were passed to a fully connected layer and then to two parallel layers: softmax with K+1 outputs (one for each + 1 class for the background) and bounding box regressor.

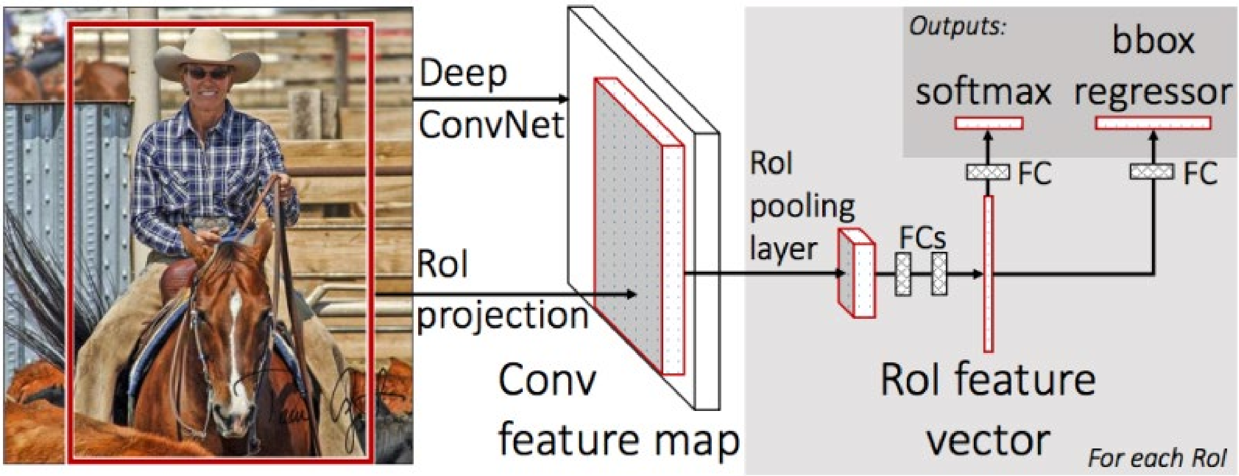

Figure 1 - the network architecture

3. Faster R-CNN

After the improvements made in Fast R-CNN, the narrowest point of the neural network was the mechanism for generating candidate regions. In 2015, a team from Microsoft Research was able to make this stage significantly faster. They proposed to calculate regions not from the original image, but again from the map of features obtained from CNN. This module called Region Proposal Network (RPN) was added to do this. We can see it in Fig.2.

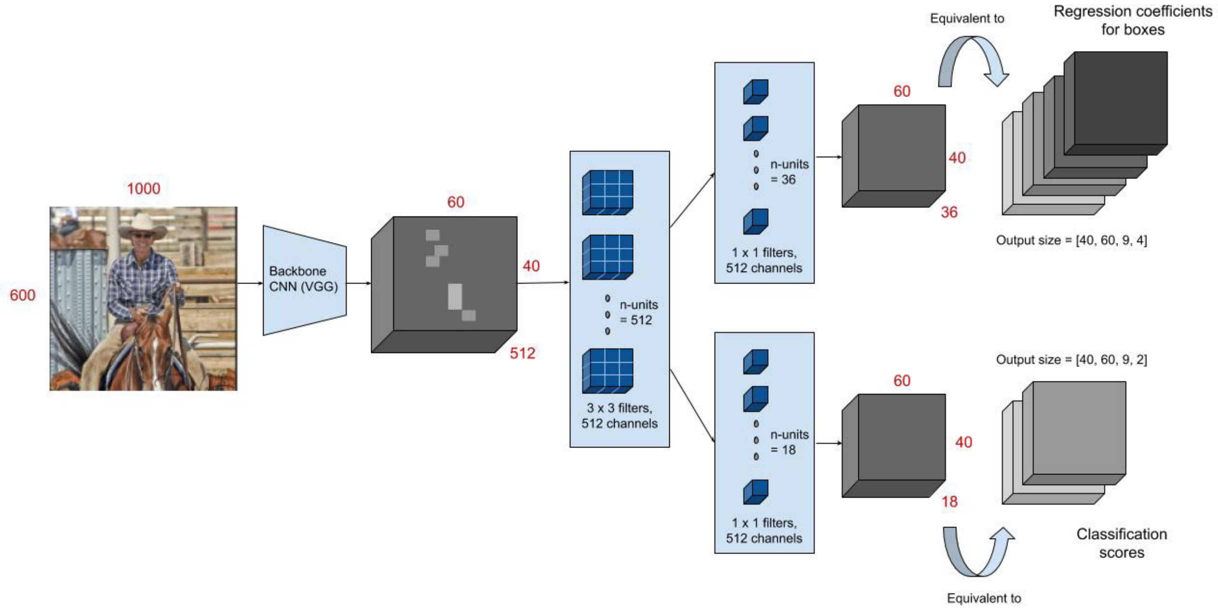
1000
600
Backbone CNN (VGG)
60
40
512
n-units = 512
3 x 3 filters, 512 channels
n-units = 36
1 x 1 filters, 512 channels
n-units = 18
1 x 1 filters, 512 channels
60
40
36
60
40
18
Equivalent to
Regression coefficients for boxes
Output size = [40, 60, 9, 4]
Output size = [40, 60, 9, 2]
Classification scores
Equivalent to

Figure 2 - The architecture of the region proposal network or RPN

There is a little neural network with a small "window" 3x3 which are used to go over the features extracted by CNN. The results, which were given by this network, are passed to two parallel fully connected layers: box-regression layer (reg) and box-classification layer (cls). The outputs of these layers were based on anchor: k frames for each position of the sliding window, having different sizes and aspect ratios. The reg layers for each anchor gave 4 coordinates, which adjust the position of the covering frame. Cls layers gave 2 values – the probability that the frame contains at least some object or that it does not contain. The example of this one we can take from our paper in Fig.3:

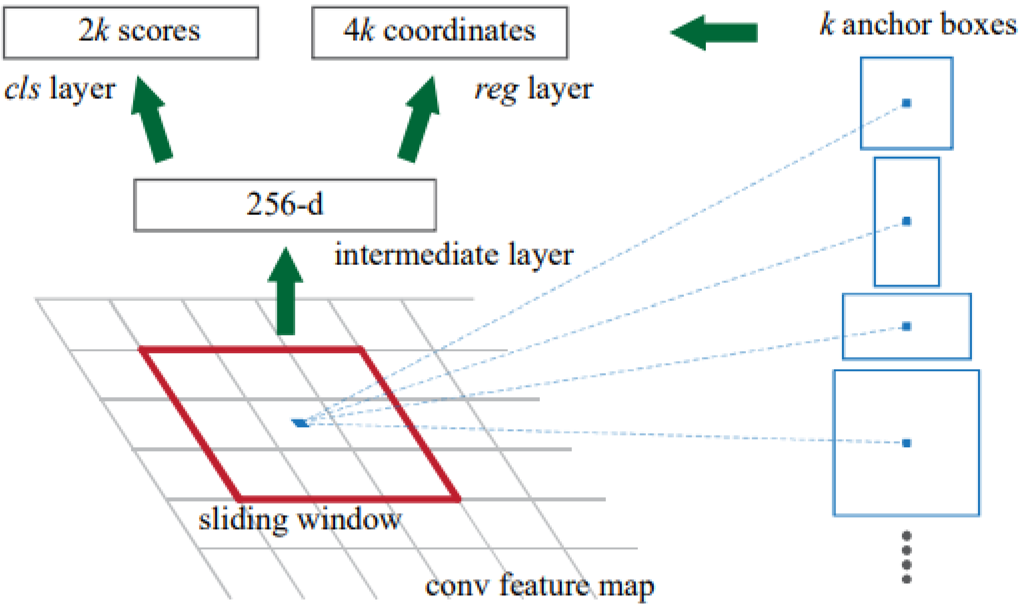

Figure 3 - Region Proposal Network (RPN)

Reg and cls layers combined learning process, the had general loss-function, which representing the sum of the loss functions of each of them, with a balancing coefficient.

Both RPN layers gave only offers for candidate regions. Those ones, which have a high probability of containing an object, are passed on to the object detection module and the specification of the scope.

In order to separate the features obtained in CNN between the RPN and the detection module, the entire network learning process is built iteratively, using several steps:

1. Initialized and trained to identify candidate regions RPN-part;
2. Using the proposed RPN regions, the Fast R-CNN part is re-trained;
3. A trained detection network is used to initialize weights for RPN. General convolution layers, however, are fixed and only RPN-specific layers are reconfigured;
4. With fixed convolution layers, Fast R-CNN is finally adjusted.

4. Experiments

In this paper we train the network using PASCAL VOC 2007 dataset.

The results based on the basic settings are the following:

Table 1 – Test results from 20 epochs using pre-trained model

Classes	AP per category, mAp	Mean AP, mAp
Airplane	76.89	66.41
Bicycle	71.12	
Bird	70.73	
Boat	52.21	
Bottle	48.67	
Bus	64.11	
Car	76.28	
Cat	77.87	
Chair	40.87	
Cow	76.67	
Dining table	59.39	
Dog	75.20	
Motorbike	81.35	
Person	78.71	
Potted plant	70.41	
Sheep	43.33	
Sofa	64.89	
Train	58.41	
Tv monitor	78.24	

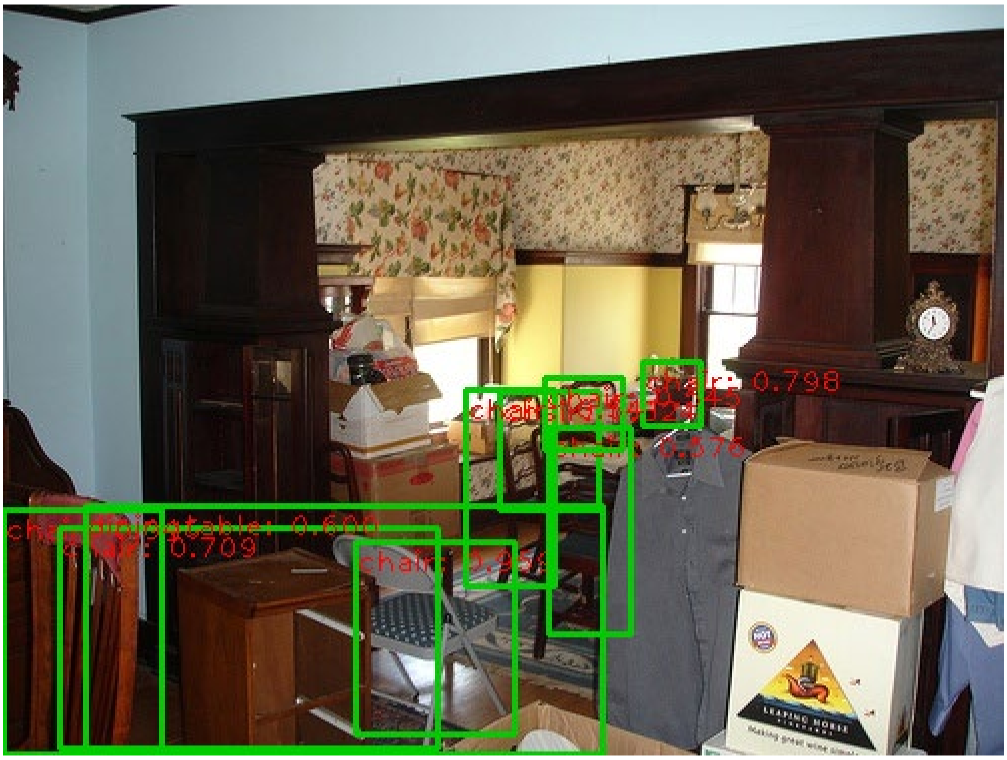

Figure 4 - The result of image detection

The training time was about 12-14 hours for 20 epochs. We think that's not enough, we need more epochs for training for getting better results.

Anchors are sampled local windows with different scales / aspect ratio. The Region Proposal Network (RPN) classifies each local anchor (window) to be either foreground or background, and stretch the foreground windows to fit ground truth object bounding boxes.

Table 2 – the influence of Scale and Ratio on performance

Scale value	Ratio value	Mean mAp (%)
1	1	61.4
1	3	62.1
3	1	65.2
3	3	65.5

More Scale value and more Ratio value can help mean AP to increase. Scale value gives a greater increase in accuracy than ratio value. We can increase few more accuracy if we'll change both values.

If we don't have pre-trained model, we'll train the network from scratch and get the follow results for 20 epochs like in Table 3.

Table 3 – Test results for 20 epochs from scratch

Classes	AP per category, mAp	Mean AP, mAp
Airplane	0.03	0.16
Bicycle	0.03	
Bird	0.04	
Boat	0.00	
Bottle	0.00	
Bus	0.31	
Car	0.29	
Cat	0.73	
Chair	0.02	
Cow	0.02	
Dining table	0.01	
Dog	0.13	
Motorbike	0.18	
Person	1.06	
Potted plant	0.01	
Sheep	0.00	
Sofa	0.02	
Train	0.10	
Tv monitor	0.01	