0.1 Faster R-CNN

- *Algorithm:* Faster R-CNN
- *Input:* A RGB 3 channel picture
- *Complexity:* None
- *Data structure compatibility:* None
- *Common applications:* Artificial intelligence

Problem. Faster R-CNN

Object detection is the task of detecting instances of objects of a certain class within an image. A sample of object detection is shown in Figure 1. Faster R-CNN[**fasterrcnn**], stading for Faster Region-based Convolutional

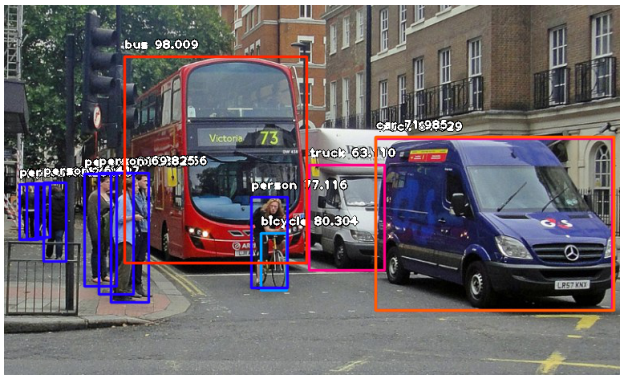

Figure 1: The result of object detection

Network, is a great milestone in the object detection. Its predecessor R-CNN[**rcnn**] first raise some region of interest by using selective search algorithm whose speed is far from practice. Faster R-CNN raises a brand new architecture to perform object detection. The sketch of its architecture is shown in Figure 2

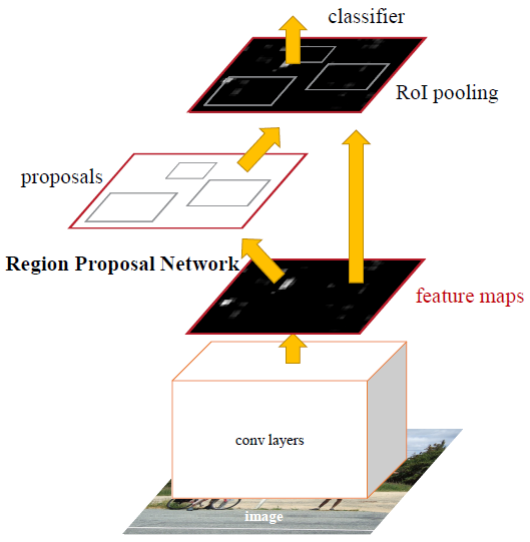

Figure 2: The structure of Fast R-CNN

Description

Conv Layers

Faster R-CNN use VGG network[**vgg**] to generate feature. It's a common backbone used in computer vission, so we don't talk about it here.

Region Proposal Network(RPN)

Talking about RPN, anchors are the most import mechanism. After we got feature maps by VGG net. We generate 9 anchors boxes shown in Figure 3. After that, it will use regression to get 4 parameters x_1, y_1, x_2, y_2

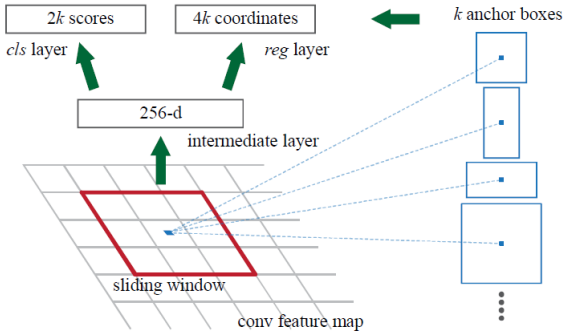

Figure 3: Region Proposal Network(RPN)

for the bounding box of the target and 1 score for whether the bounding box includes a target.

The RoI pooling layer

After we crop the image from the region proposed by RPN, the cropped image cannot be directly send to VGG net soon. Because the image size is not match with the input size of VGG. So, Faster R-CNN first project the region to $H/16 \times W/16$ which is the scale of the feature map. Then, it divided the feature map to $H_{pool} \times W_{pool}$ grid where it is the scale of our target size. At last we do max pooling.

Classification

Classification use VGG net as common. The structure is shown in Figure 4.

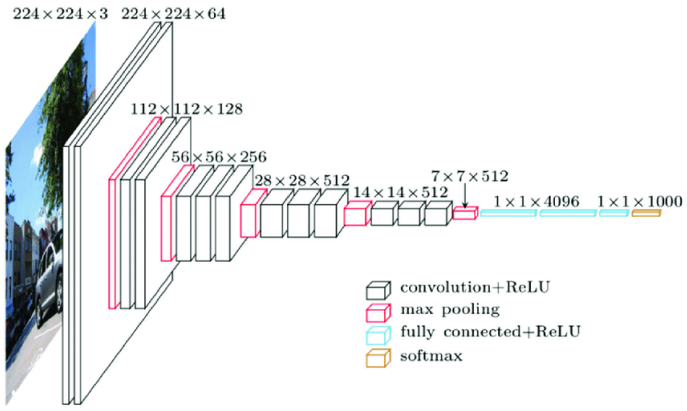

Figure 4: VGG net)