Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

报告人：周再达

目录
CONTENT

01 摘要
Abstract

02 R-CNN系列
R-CNN

03 网络结构
network structure

04 训练过程
training process

第一部分

摘要

Abstract

摘要

Abstract

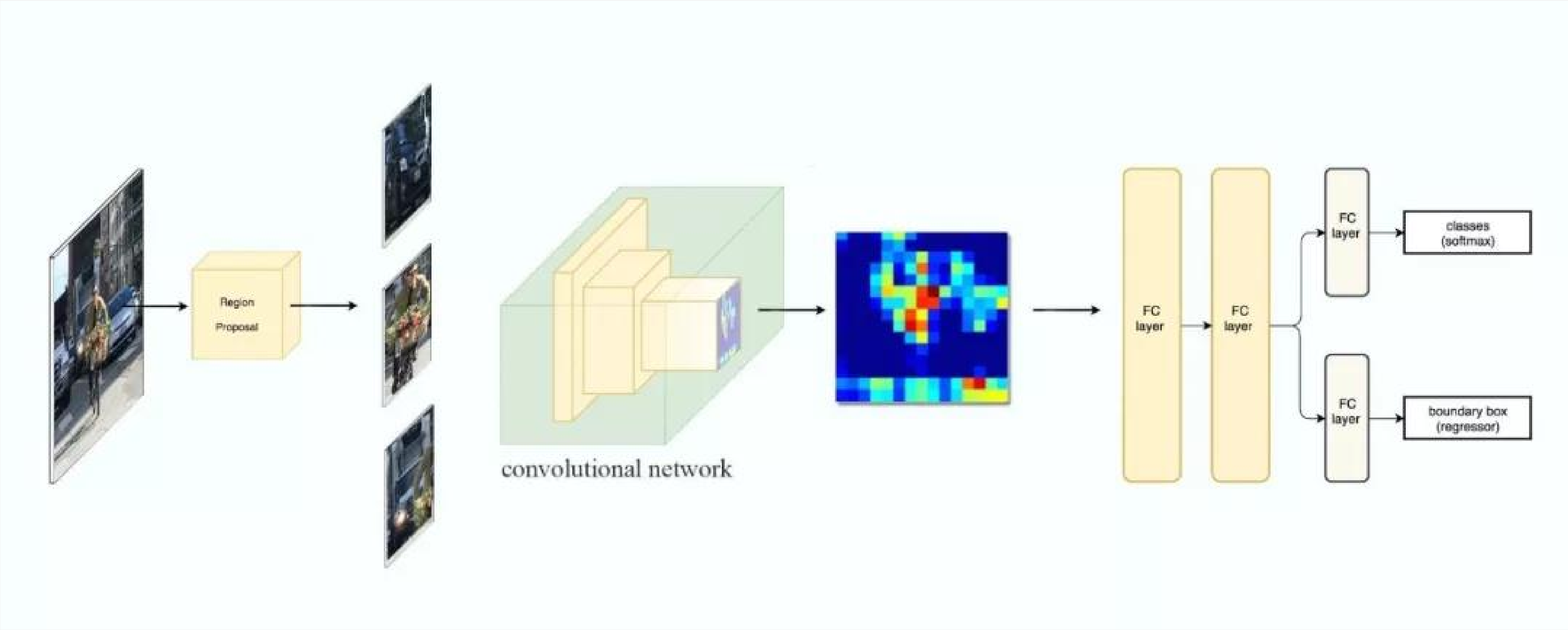
State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet and Fast R-CNN have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. In this work, we introduce a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. We further merge RPN and Fast R-CNN into a single network by sharing their convolutional features—using the recently popular terminology of neural networks with “attention” mechanisms, the RPN component tells the unified network where to look. For the very deep VGG-16 model, our detection system has a frame rate of 5fps (including all steps) on a GPU, while achieving state-of-the-art object detection accuracy on PASCAL VOC 2007, 2012, and MS COCO datasets with only 300 proposals per image. In ILSVRC and COCO 2015 competitions, Faster R-CNN and RPN are the foundations of the 1st-place winning entries in several tracks. Code has been made publicly available

第二部分

R-CNN系列

R-CNN

R-CNN


Region
Proposal
convolutional network
FC
layer
FC
layer
FC
layer
classes
(softmax)
FC
layer
boundary box
(regressor)

R-CNN系列

R-CNN

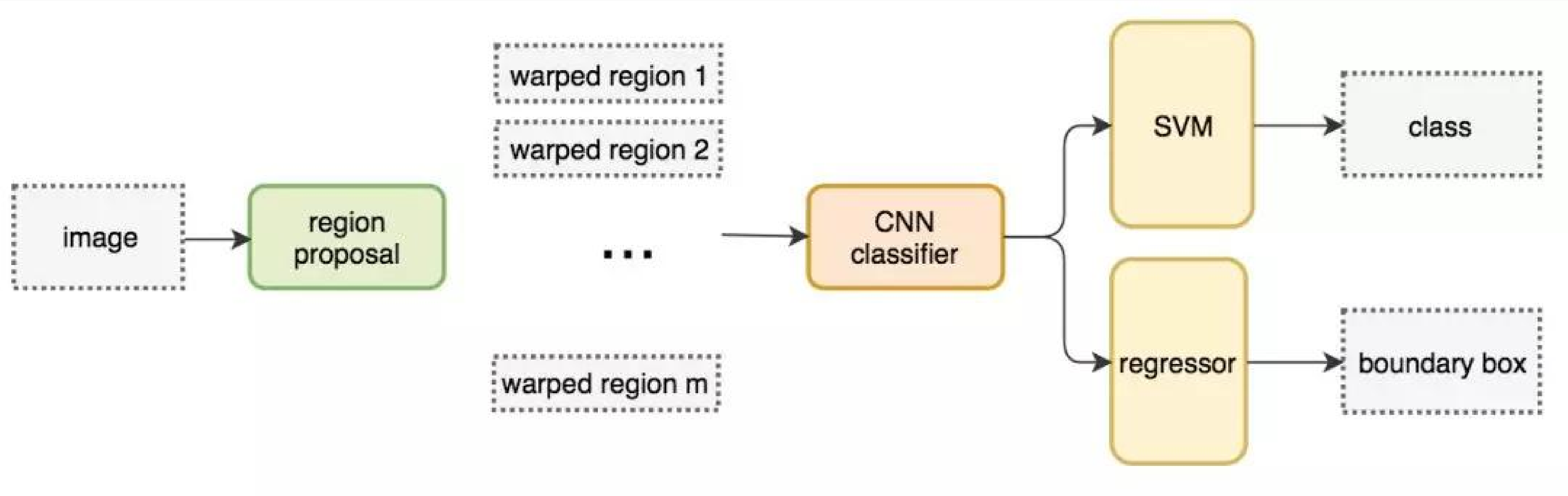


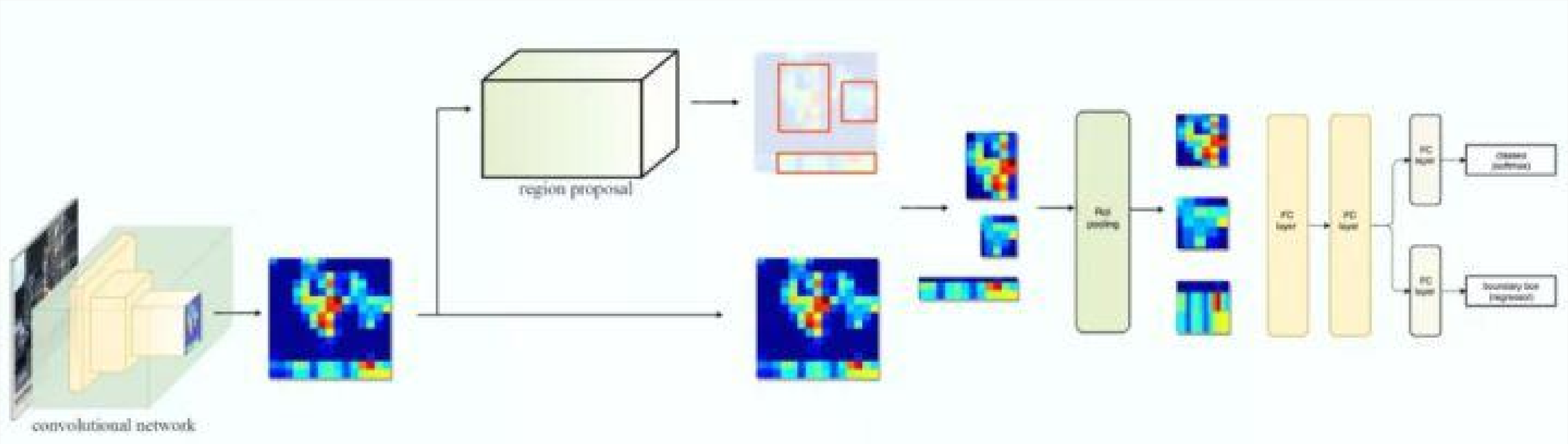
image
region proposal
warped region 1
warped region 2
...
warped region m
CNN classifier
SVM
regressor
class
boundary box

R-CNN系列

R-CNN

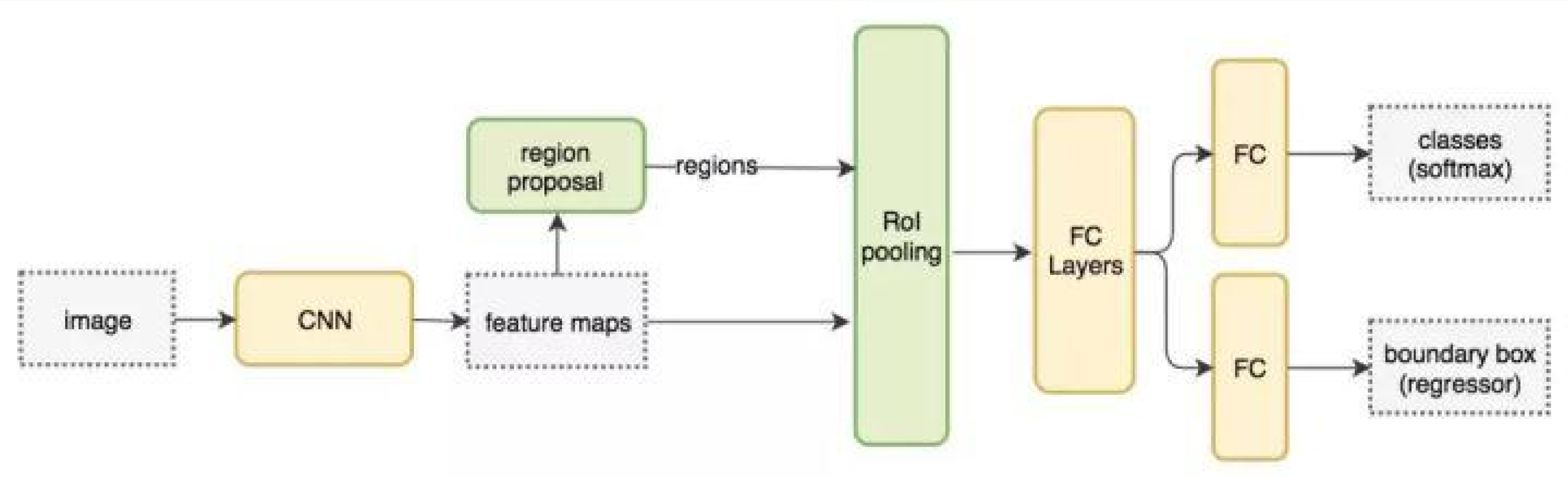
```
ROIs = region_proposal(image)
for ROI in ROIs
    patch = get_patch(image, ROI)
    results = detector(patch)
```

Fast R-CNN


region proposal
convolutional network
RoI pooling
FC layer
FC layer
FC layer
classes (softmax)
FC layer
boundary box (regression)

R-CNN系列

R-CNN

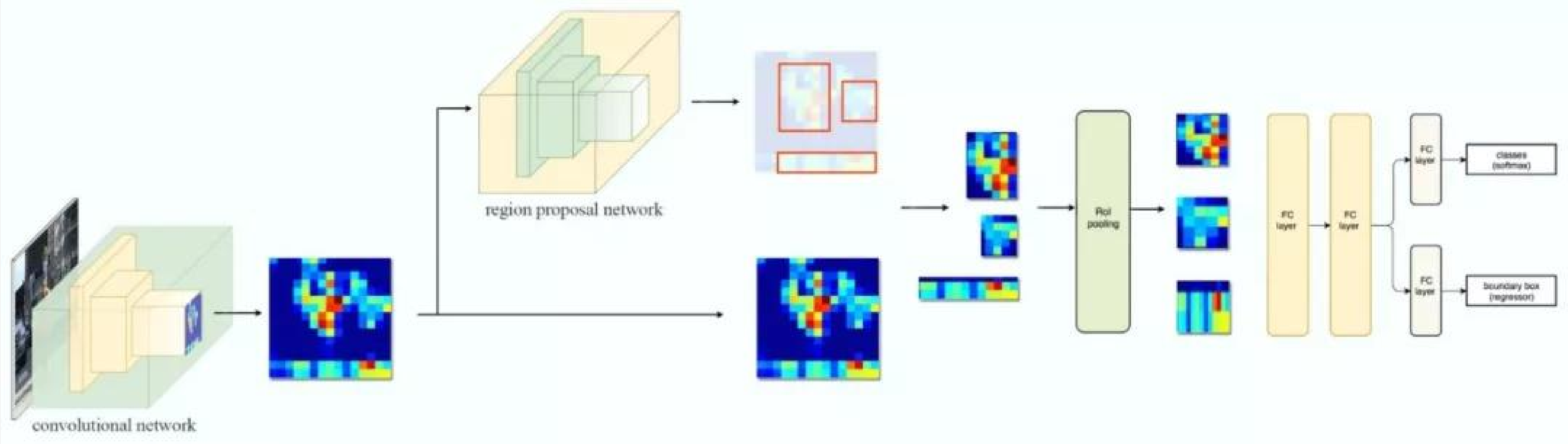

region proposal
regions
image
CNN
feature maps
RoI pooling
FC Layers
FC
classes (softmax)
FC
boundary box (regressor)

R-CNN系列

R-CNN

```
feature_maps = process(image)
ROIs = region_proposal(feature_maps)
for ROI in ROIs
    patch = roi_pooling(feature_maps, ROI)
    results = detector2(patch)
```

Faster R-CNN


region proposal network
convolutional network
RoI pooling
FC layer
FC layer
FC layer
FC layer
classes (softmax)
boundary box (regressor)

R-CNN系列

R-CNN

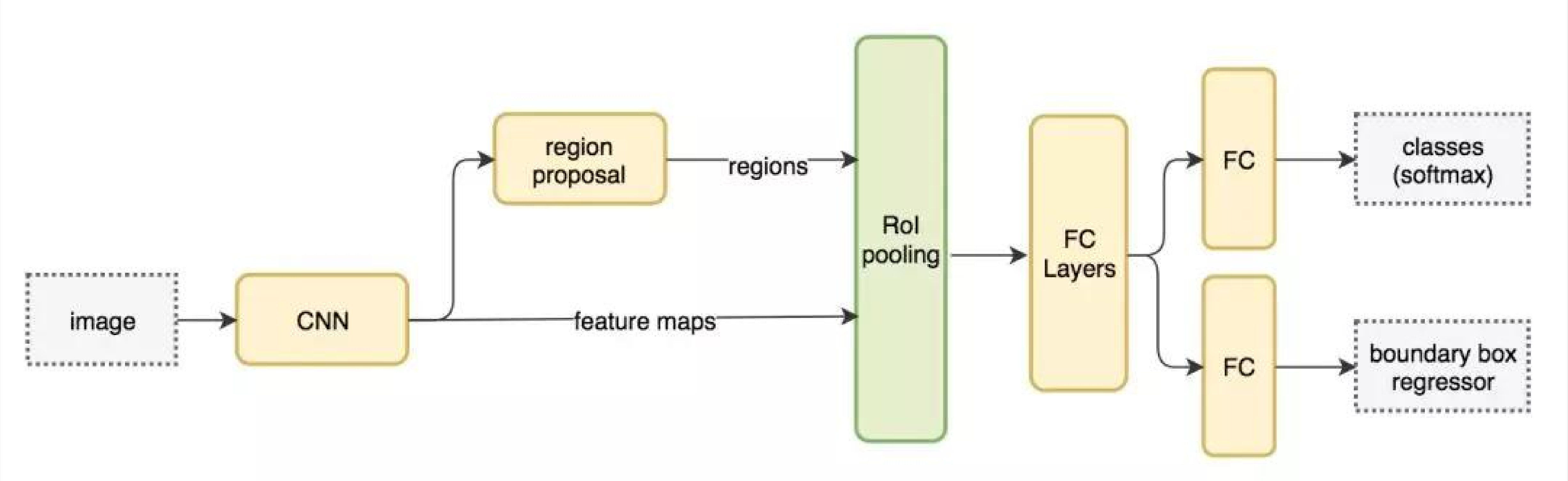


image
CNN
region proposal
regions
feature maps
RoI pooling
FC Layers
FC
FC
classes (softmax)
boundary box regressor

R-CNN系列

R-CNN

```
feature_maps = process(image)
ROIs = region_proposal(feature_maps)
for ROI in ROIs
    patch = roi_pooling(feature_maps, ROI)
    results = detector2(patch)
```

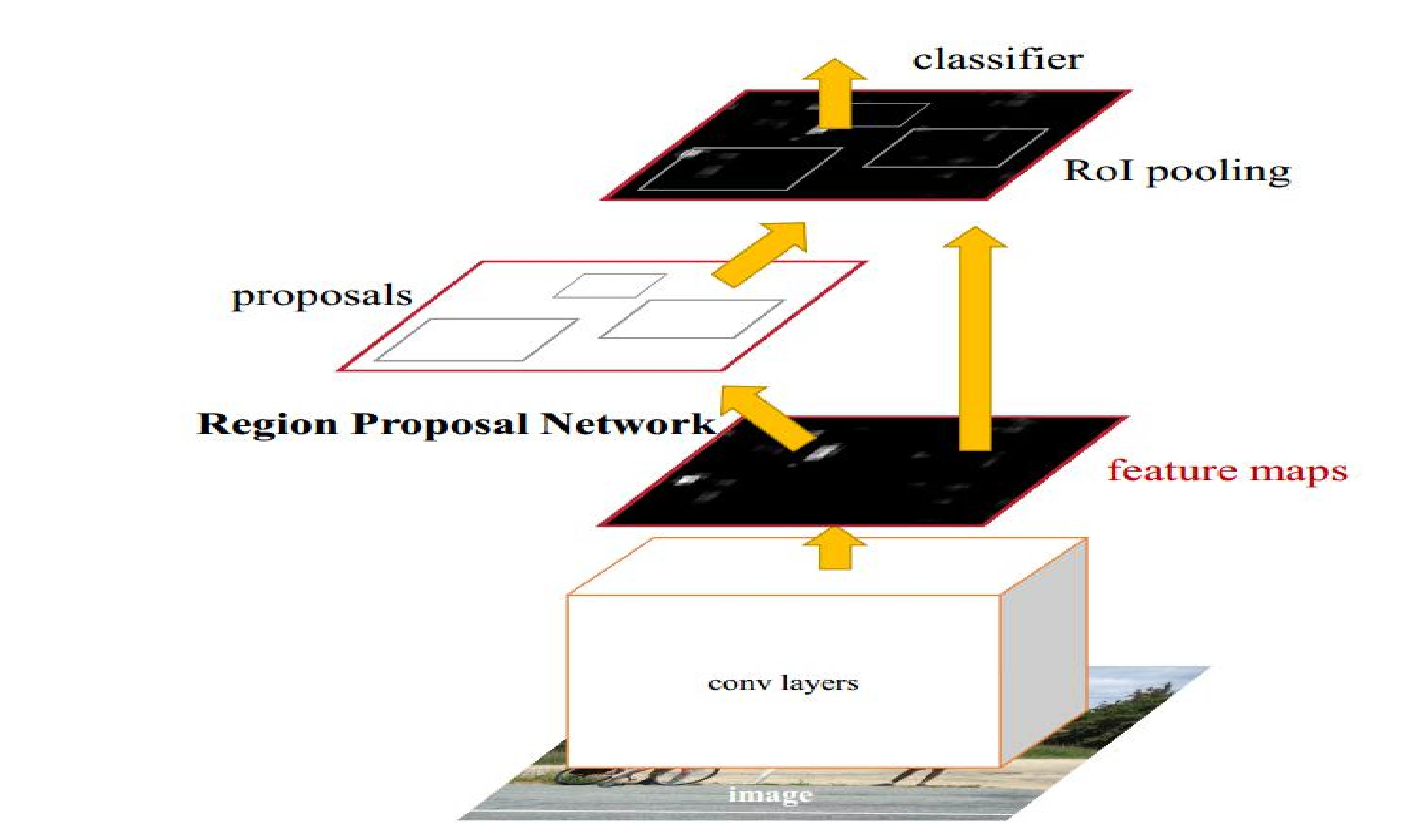
第三部分

网络结构

Network structure

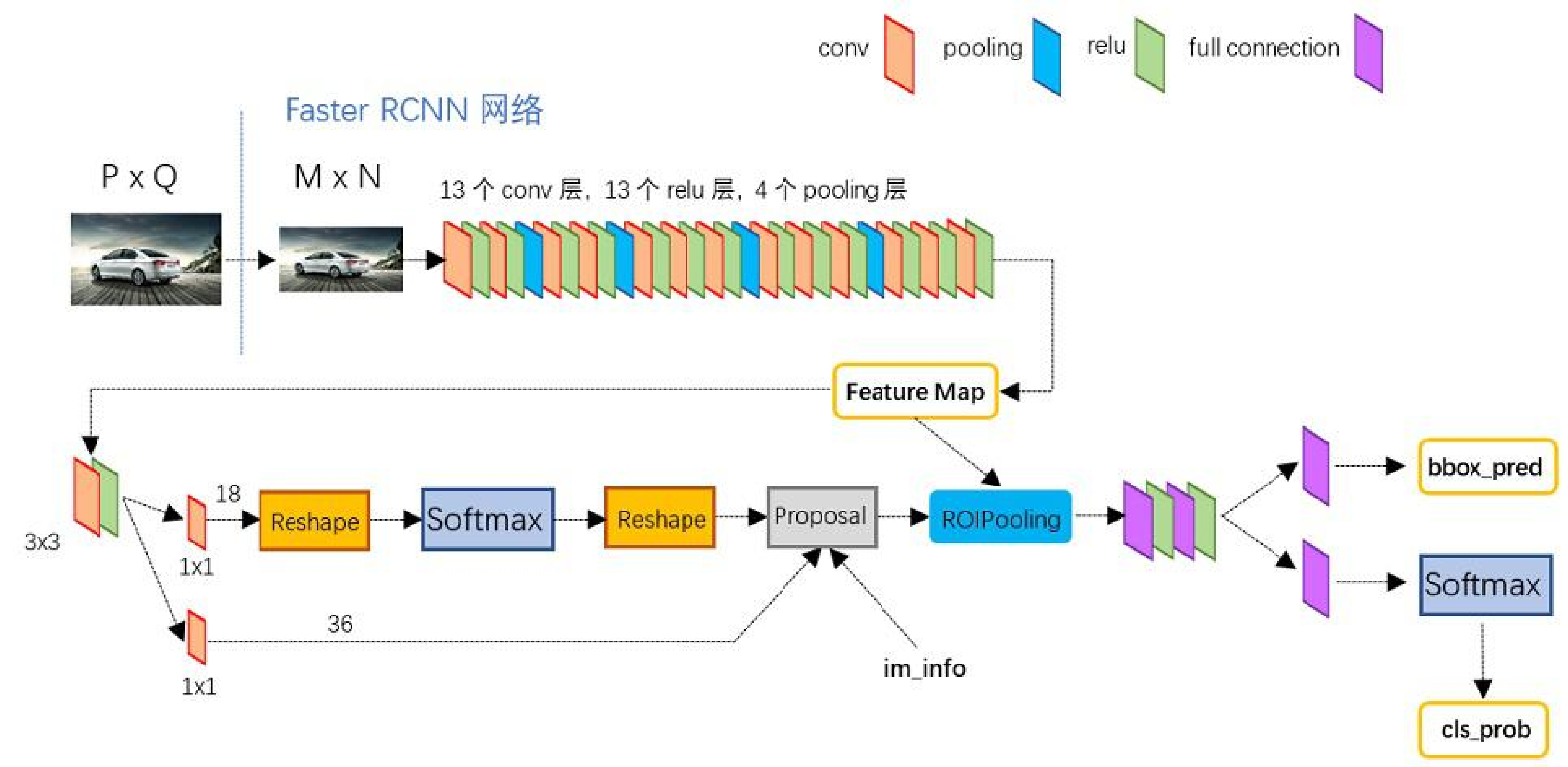
网络结构

Network Structure



网络结构

Network Structure

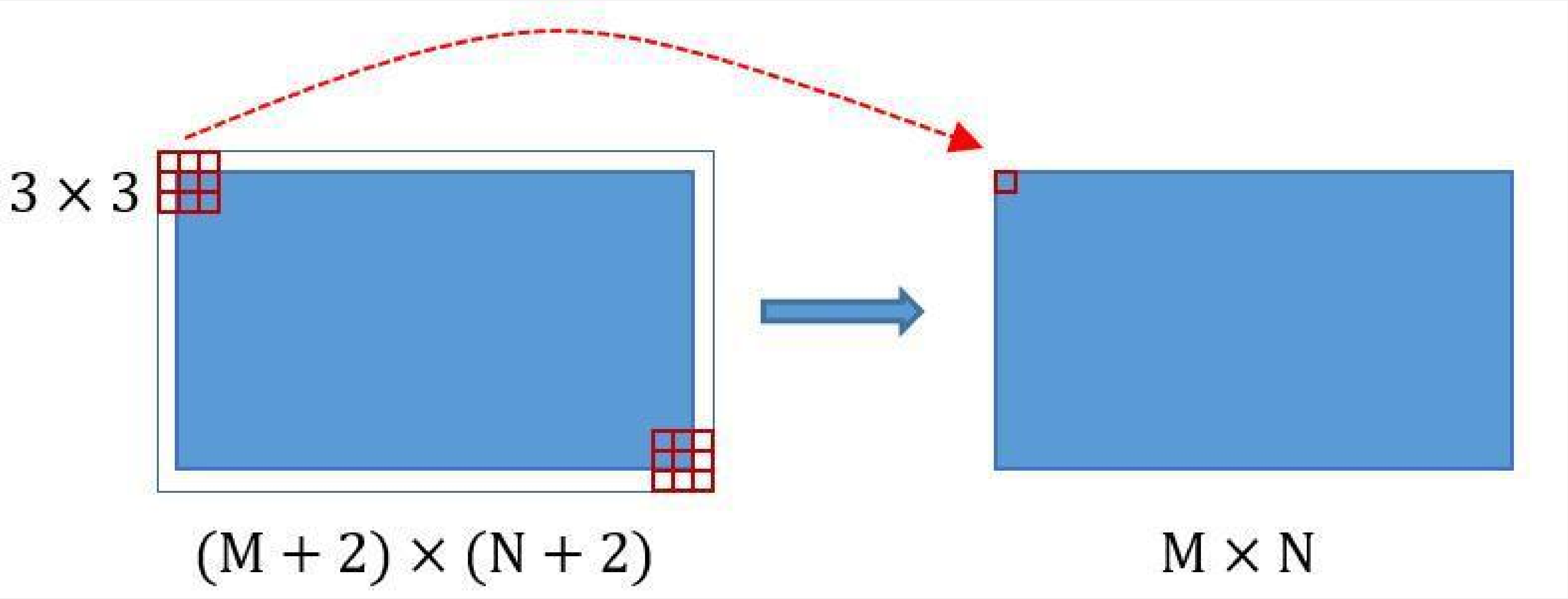


网络结构

Network Structure

- conv layers
- Region Proposal Network
- RoI pooling
- classifier

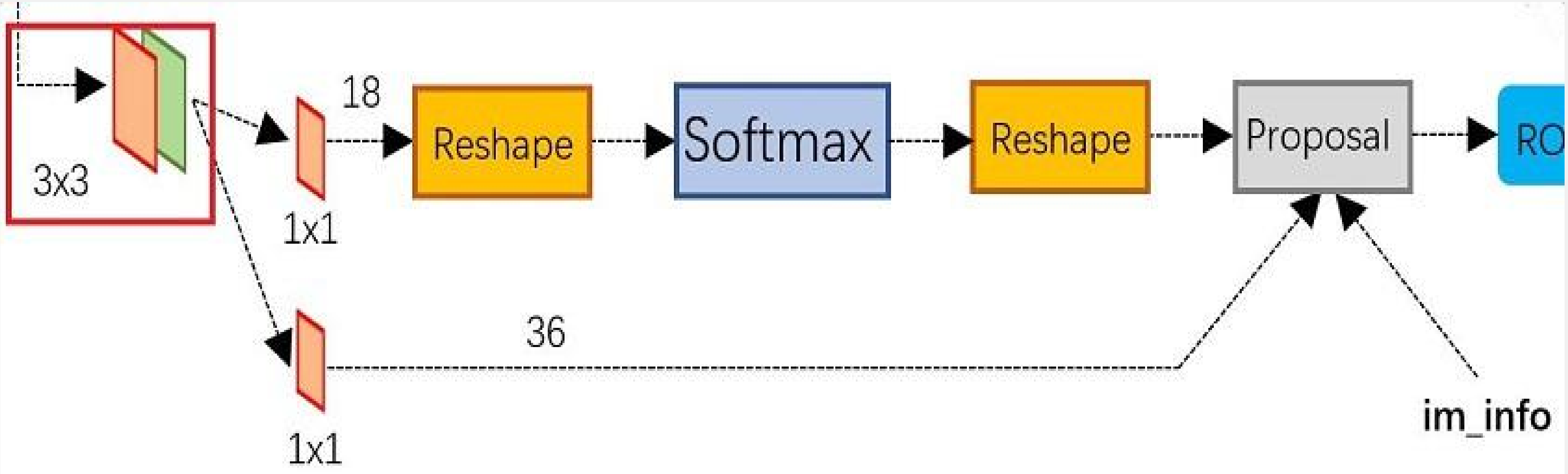
conv layers



网络结构

Network Structure

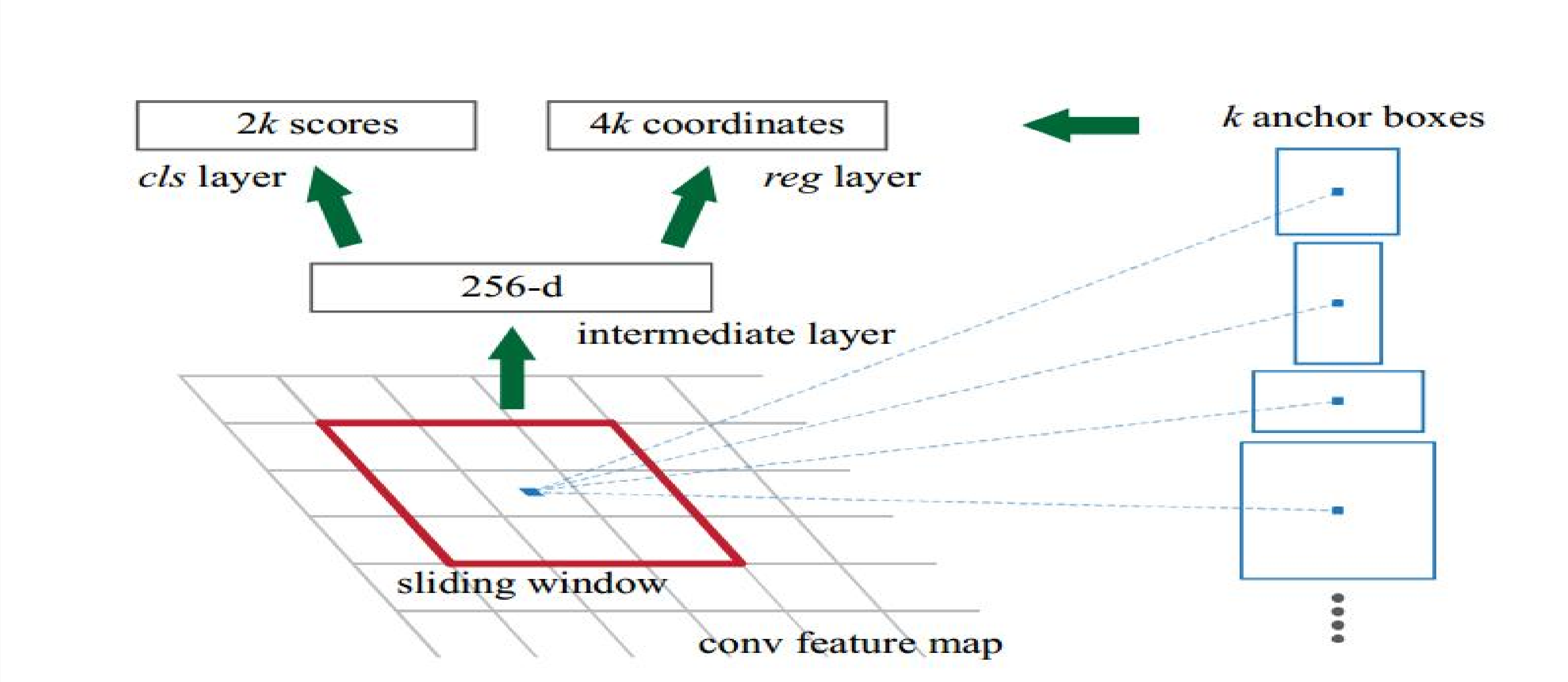
RPN



网络结构

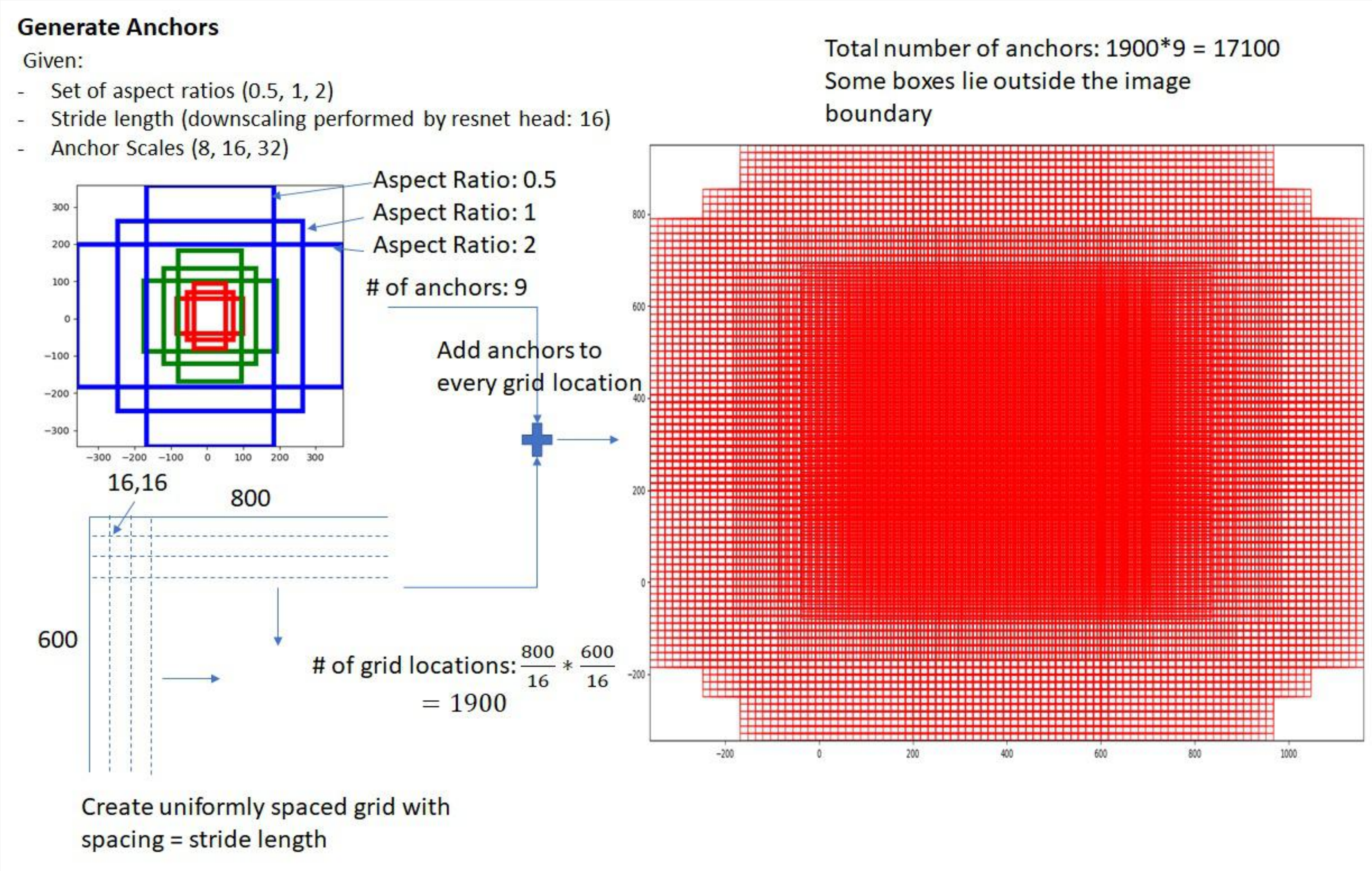
Network Structure

RPN



RPN

Anchor



Generate Anchors

Given:

- Set of aspect ratios (0.5, 1, 2)
- Stride length (downscaling performed by resnet head: 16)
- Anchor Scales (8, 16, 32)

Aspect Ratio: 0.5
Aspect Ratio: 1
Aspect Ratio: 2

# of anchors: 9

Add anchors to every grid location

16,16

800

600

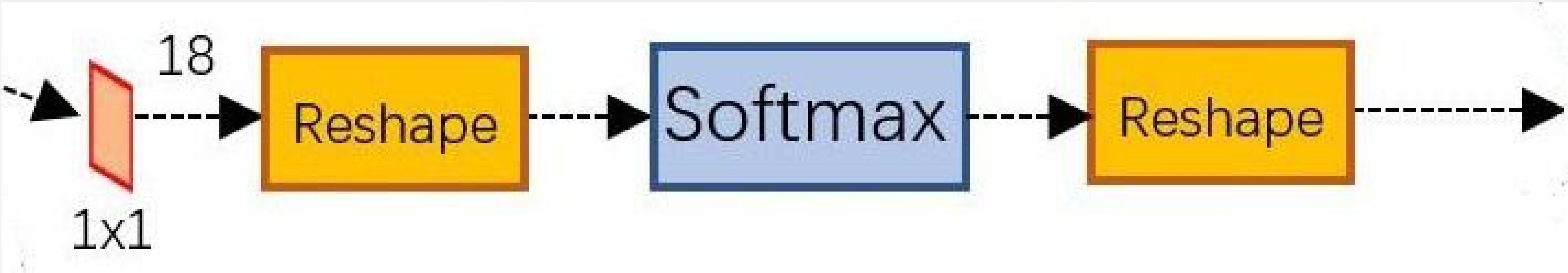
$$\text{\# of grid locations: } \frac{800}{16} * \frac{600}{16} = 1900$$

Create uniformly spaced grid with spacing = stride length

Total number of anchors: 1900*9 = 17100
Some boxes lie outside the image boundary

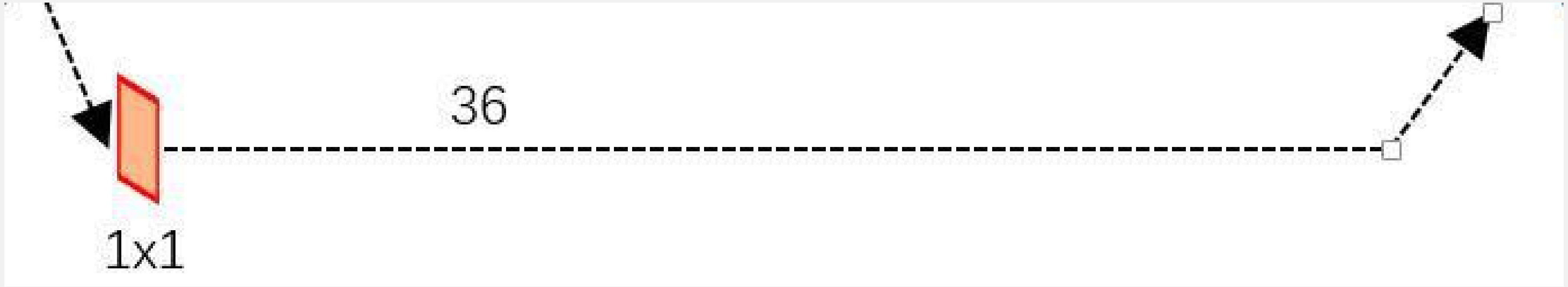
RPN

softmax判定foreground与background



RPN

对proposals进行bounding box regression

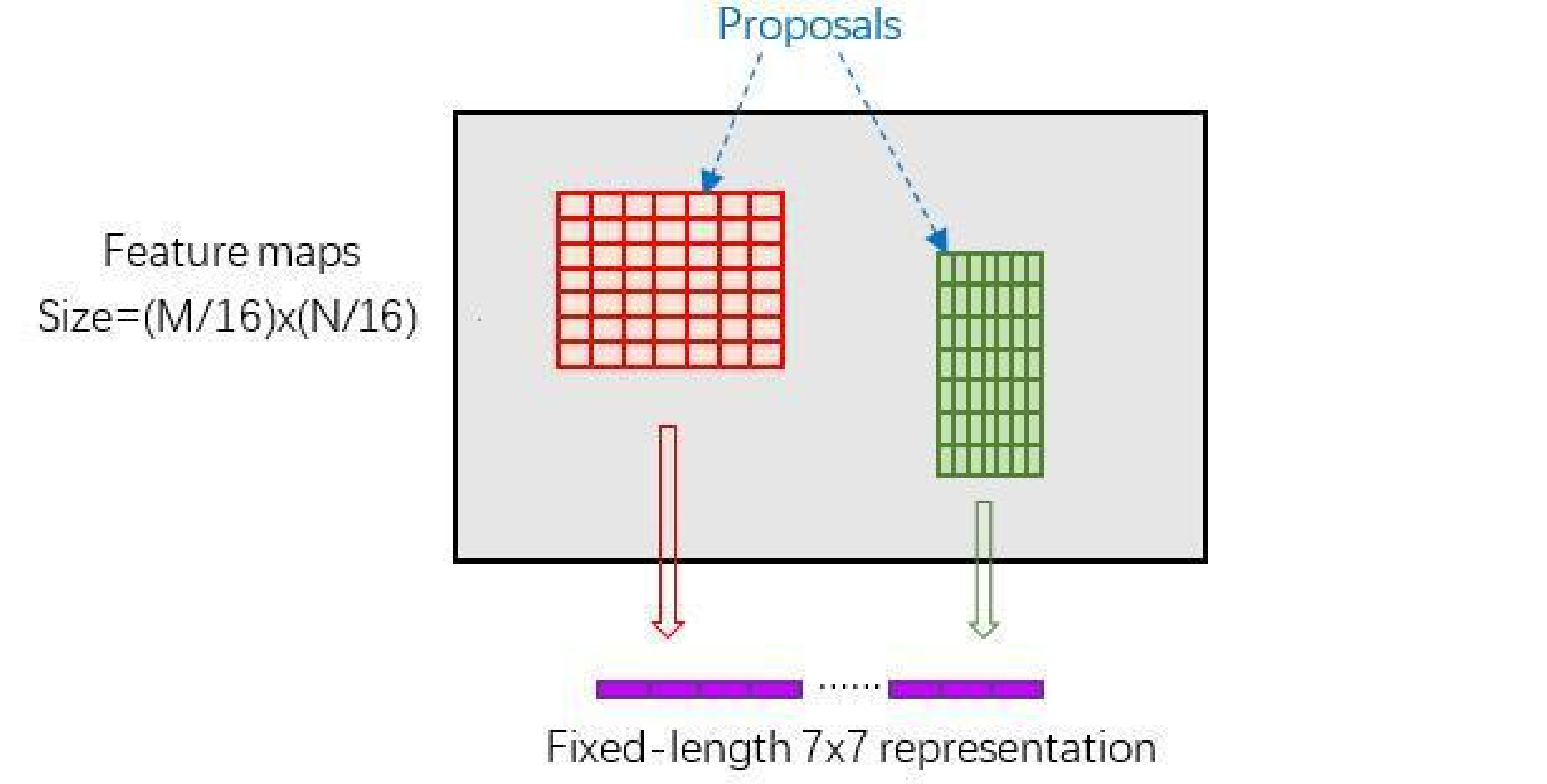


$$\left[d_x(A), d_y(A), d_w(A), d_h(A)\right]$$

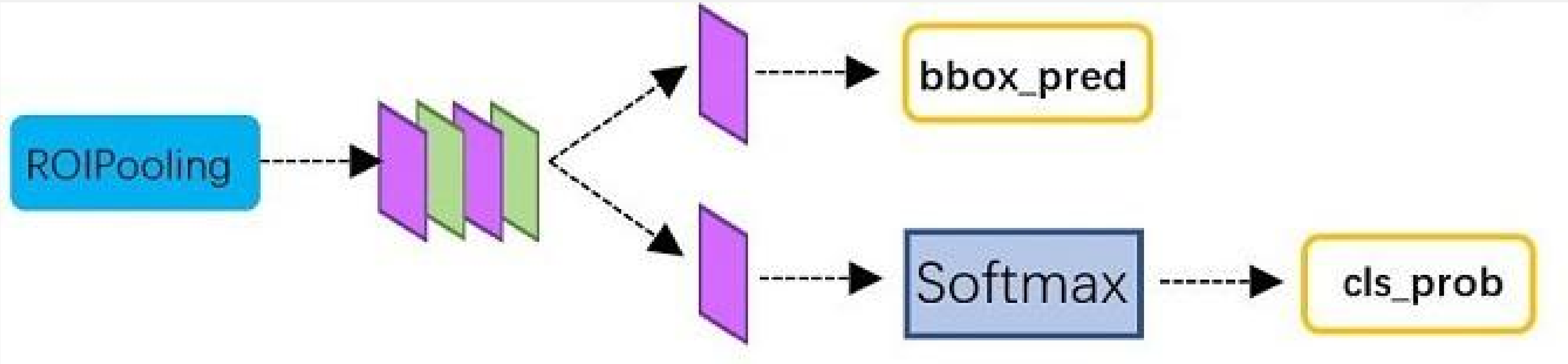
RPN

Proposal Layer

RoI pooling


Proposals
Feature maps
Size=(M/16)x(N/16)
......
Fixed-length 7x7 representation

Classifier



第四部分

训练过程

training process

训练过程

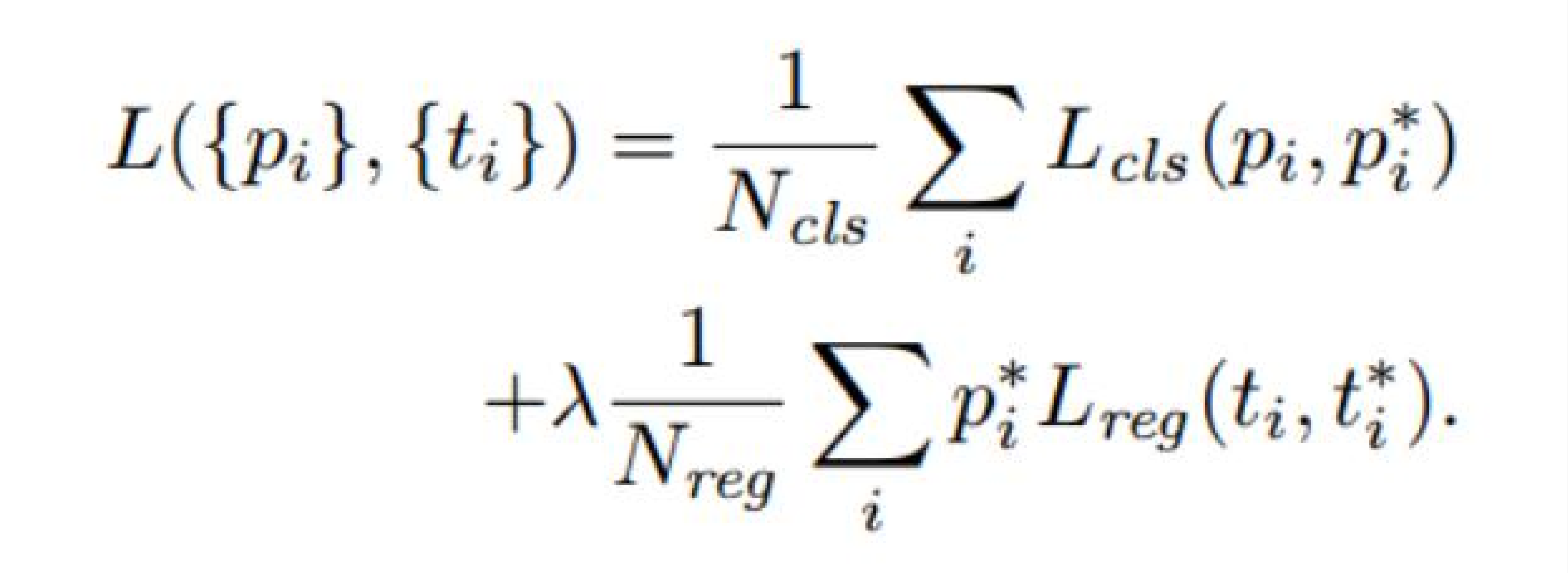
training process

- Alternating training
- Approximate joint training
- Non-approximate joint training

训练过程

training process

Loss Function



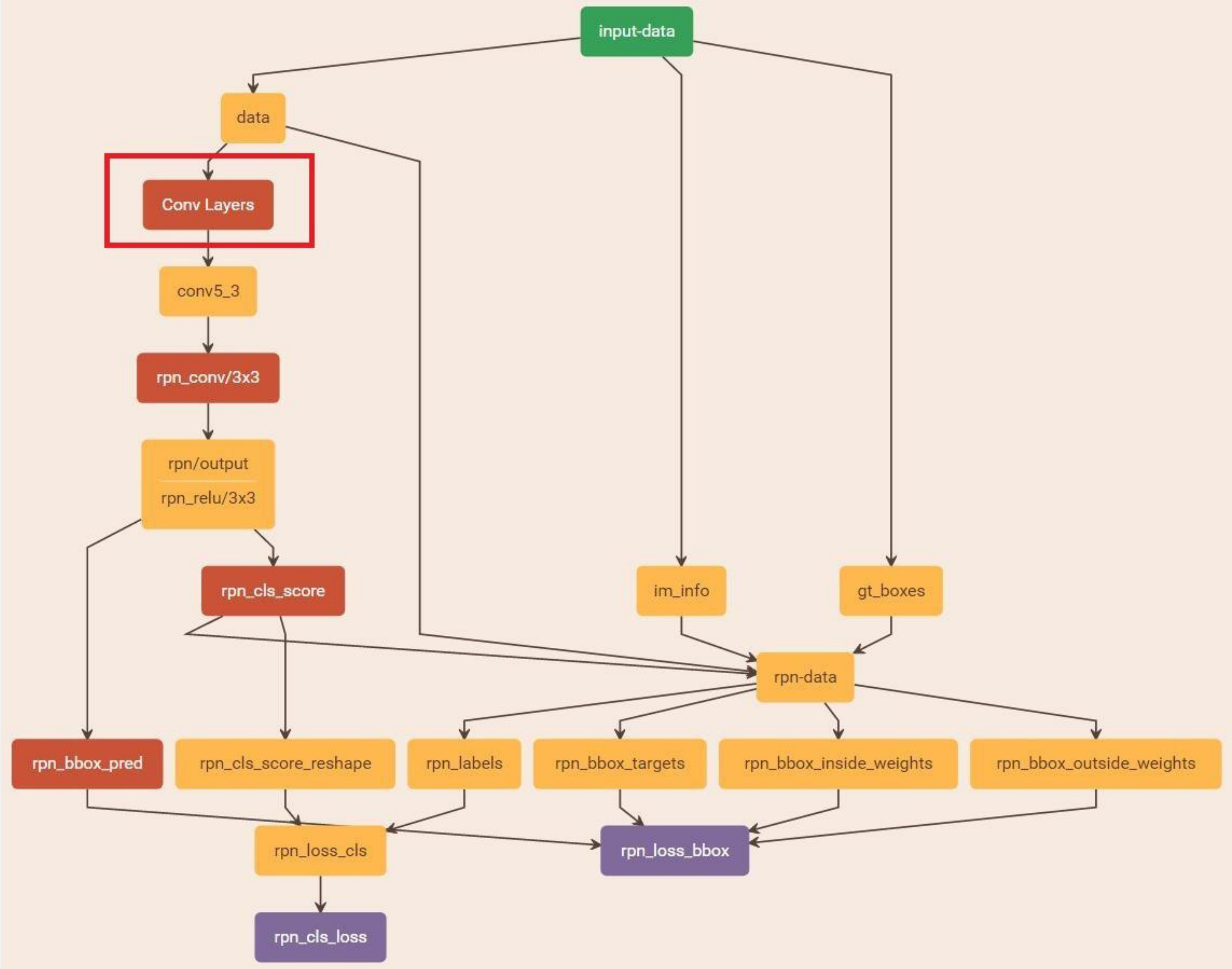
$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*).$$

Alternating training

- 在已经训练好的model上，训练RPN网络
- 利用上一个步骤中训练好的RPN网络，收集proposals
- 第一次训练Faster RCNN网络
- 第二训练RPN网络
- 再次利用上个步骤中训练好的RPN网络，收集proposals
- 第二次训练Fast RCNN网络

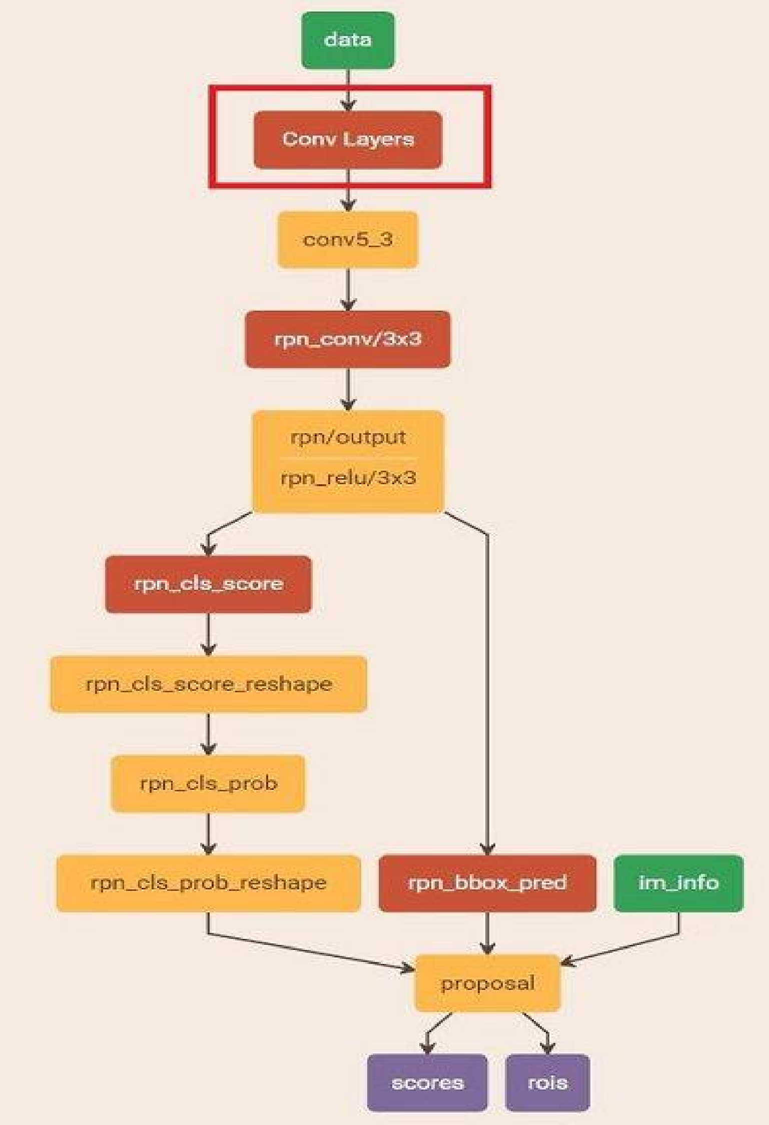
Alternating training

训练RPN网络



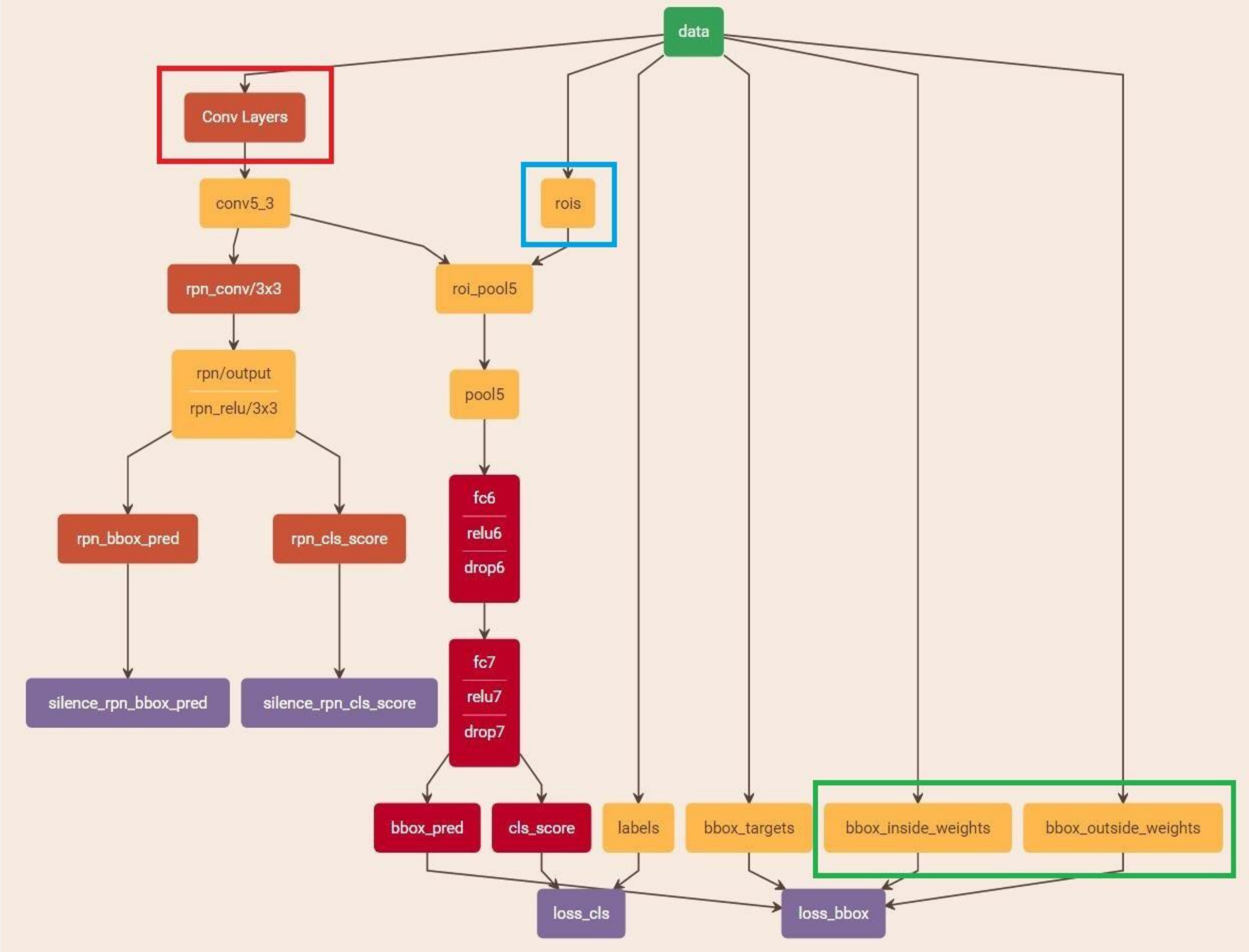
Alternating training

通过训练好的RPN网络收集proposals



Alternating training

训练Faster RCNN网络



感谢观看