深蓝学院
shenlanxueyuan.com

# 深度学习理论与实践

## 第6课：目标检测

主讲人 洪振

中国科学院自动化研究所博士
模式识别国家重点实验室

# 课程内容

深蓝学院
shenlanxueyuan.com

- 目标：
  - 了解目标识别的原理
  - 对比目标识别的传统方法和深度学习优劣
  - 利用Pytorch构建目标识别网络
- 基础储备：
  - 熟悉Pytorch
  - 掌握CNN的基本流程

# 课程内容

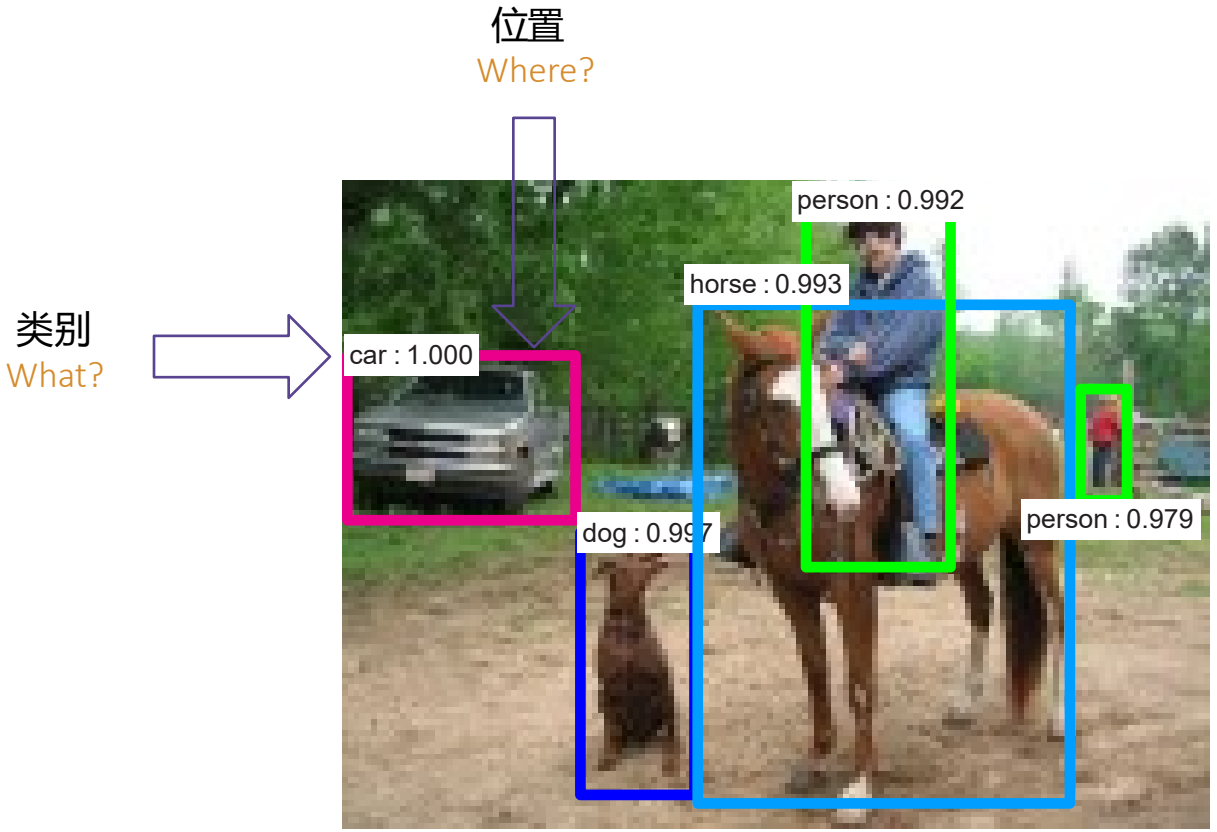
深蓝学院
shenlanxueyuan.com

- ✓**目标识别介绍**
- ✓传统方法和深度学习对比
- ✓Faster RCNN介绍
- ✓Faster RCNN代码详解

# 目标识别—object recognition

深蓝学院 shenlanxueyuan.com

## 定义



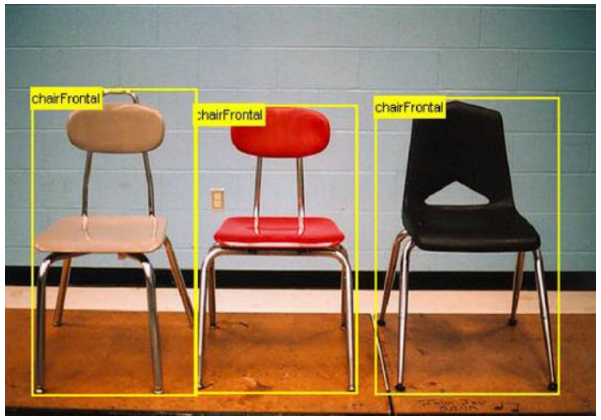

## 常用数据集：

VOC2007 VOC2012: 20 类，简单场景， 11,530 张图像，27,450 物体
COCO：91类，复杂场景， 328,000图像，2,500,000物体

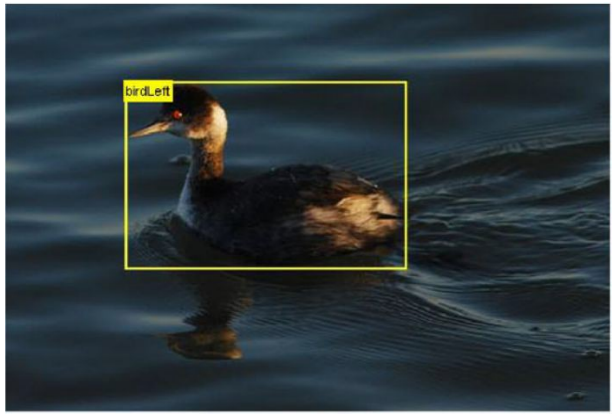
# 目标识别—object recognition

深蓝学院
shenlanxueyuan.com

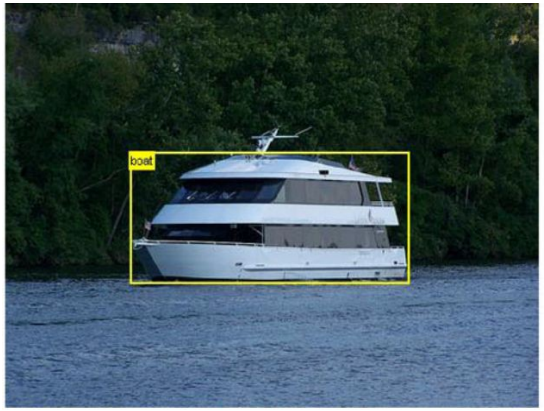
**Pascal Examples**

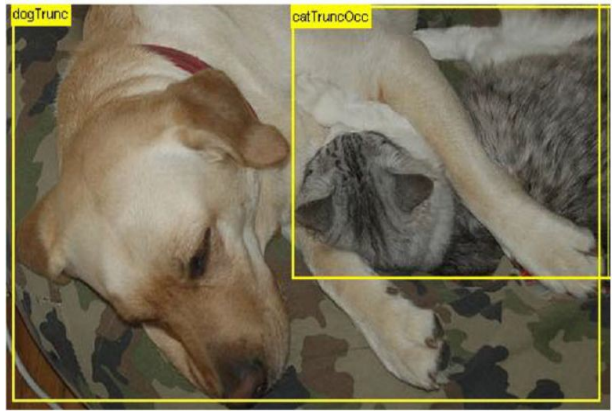








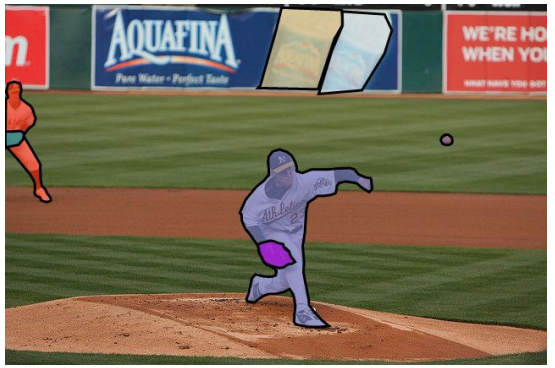


# 目标识别—object recognition

深蓝学院
shenlanxueyuan.com

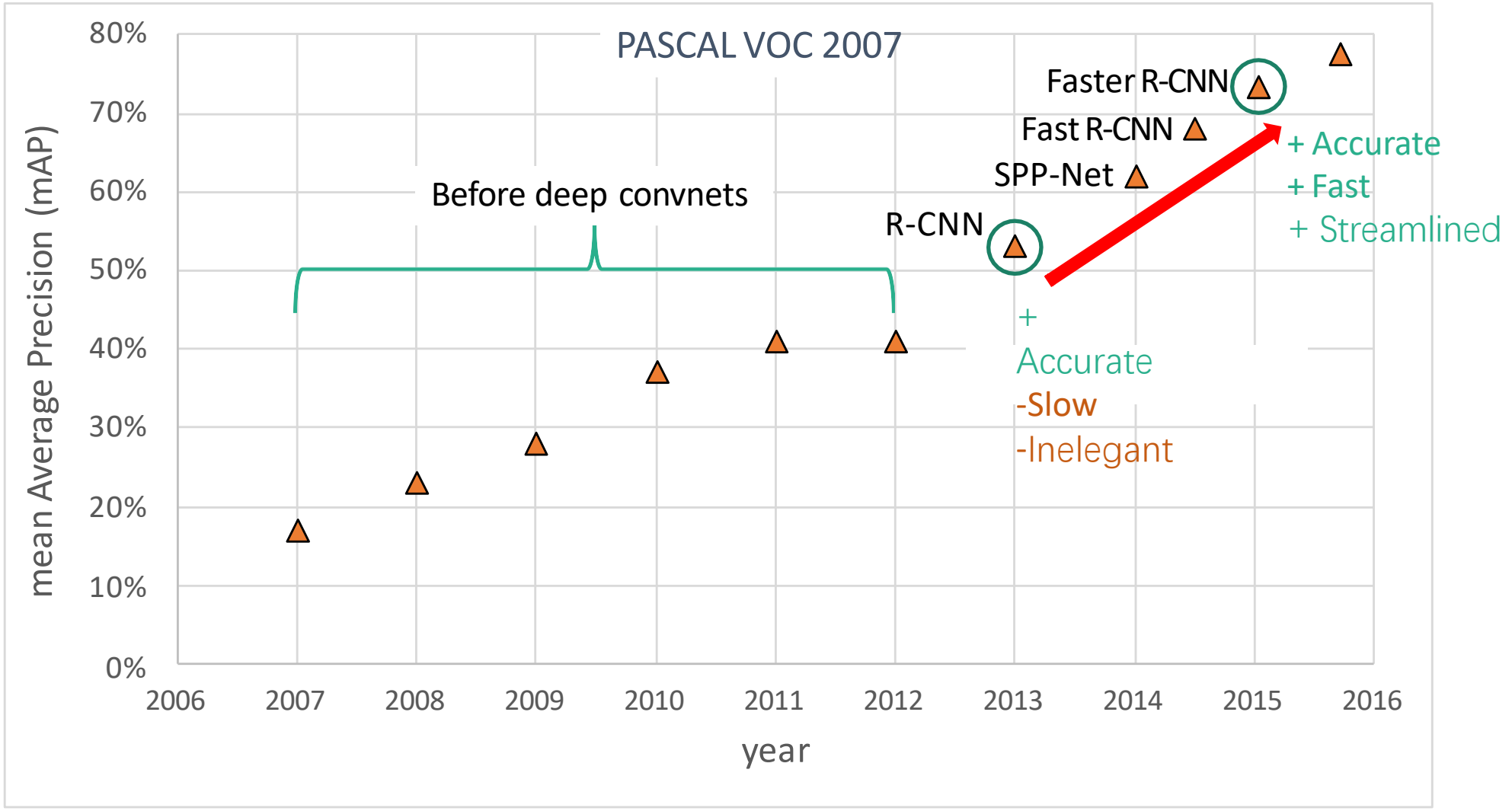
**COCO Examples**


# 目标识别—object recognition

深蓝学院 shenlanxueyuan.com

## 目标识别发展过程



# 目标识别—object recognition

深蓝学院 shenlanxueyuan.com

**3个典型工作: R-CNN　SPP NET Fast R-CNN　Faster RCNN**

| 标题　1–20 | 引用次数 | 发表年份 |
|---|---|---|
| Object detection with discriminatively trained part-based models<br>PF Felzenszwalb, RB Girshick, D McAllester, D Ramanan<br>Pattern Analysis and Machine Intelligence, IEEE Transactions on 32 (9), 1627 ... | 4457 | 2010 |
| Caffe: Convolutional architecture for fast feature embedding<br>Y Jia, E Shelhamer, J Donahue, S Karayev, J Long, R Girshick, ...<br>Proceedings of the 22nd ACM international conference on Multimedia, 675-678 | 2306 | 2014 |
| Rich feature hierarchies for accurate object detection and semantic segmentation<br>R Girshick, J Donahue, T Darrell, J Malik<br>Proceedings of the IEEE conference on computer vision and pattern ... | 1753 | 2014 |
| Cascade object detection with deformable part models<br>PF Felzenszwalb, RB Girshick, D McAllester<br>Computer vision and pattern recognition (CVPR), 2010 IEEE conference on ... | 500 | 2010 |
| Fast r-cnn<br>R Girshick<br>Proceedings of the IEEE International Conference on Computer Vision, 1440-1448 | 344 | 2015 |
| Discriminatively Trained Deformable Part Models (DPM) Code<br>RB Girshick, PF Felzenszwalb, D McAllester<br>http://www.cs.berkeley.edu/~rbg/latent/ | 342 * | 2012 |
| Faster R-CNN: Towards real-time object detection with region proposal networks<br>S Ren, K He, R Girshick, J Sun<br>Advances in neural information processing systems, 91-99 | 286 | 2015 |

Ross Girshick

Research Scientist, Facebook AI Research (FAIR)
computer vision, machine learning
在 eecs.berkeley.edu 的电子邮件经过验证 - 首页

# 课程内容

深蓝学院
shenlanxueyuan.com

- ✓目标识别介绍
- ✓**传统方法和深度学习对比**
- ✓Faster RCNN介绍
- ✓Faster RCNN代码详解

# 目标识别—object recognition

深蓝学院 shenlanxueyuan.com

## 目标识别流程

输入图像 → 目标潜在位置筛选 → 候选框提取 → 候选区域分类 → 结果

SVM
CNN
Adaboost
.....

person

boat

# 相关工作介绍—传统方法

深蓝学院 shenlanxueyuan.com

## 算法流程

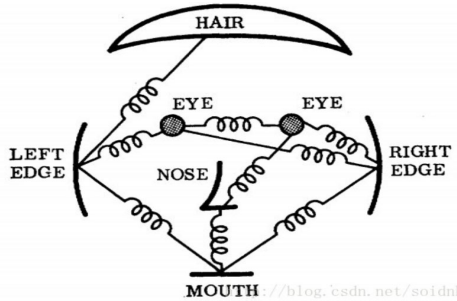
图像 → 候选区域 → 特征提取 → 分类器 → 结果

- 候选区域提取
  - 基于学习：BING，EdgeBoxes
  - 基于区域：海陆分割，机场检测
  - 基于模板：模板匹配
- 特征提取：颜色、纹理、HOG等
- 分类器：SVM、Adaboost等

主流框架：

词袋模型

可变部件模型




模板匹配

# 相关工作介绍—传统方法DPM



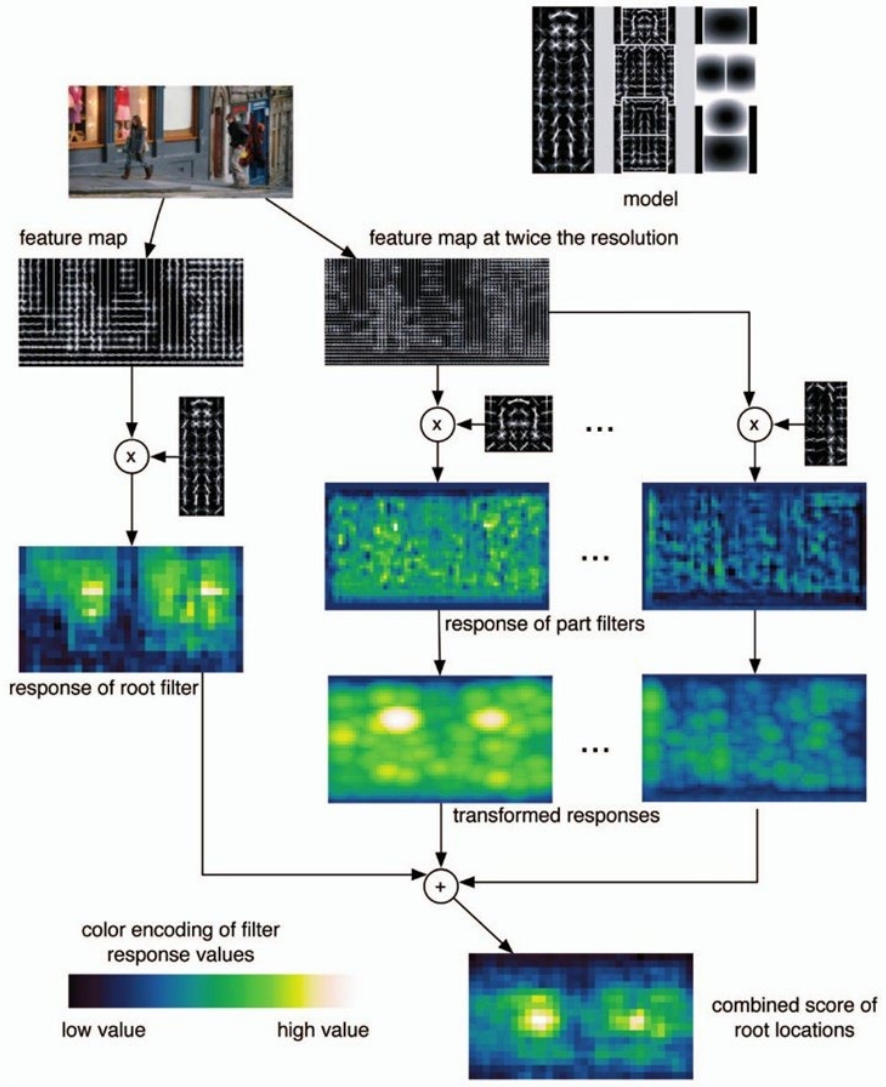
## Deformable Parts Model(DPM):

获得VOC（Visual Object Class）07,08,09年的检测冠军。

2010年Pedro Felzenszwalb被VOC授予"终身成就奖“

每一类都构建部件模型，以检测人体为例：

- 构建 root 滤波器—人体
- 构建 part 滤波器—头，四肢
- 构建能量函数
- 在大图中滑动检测



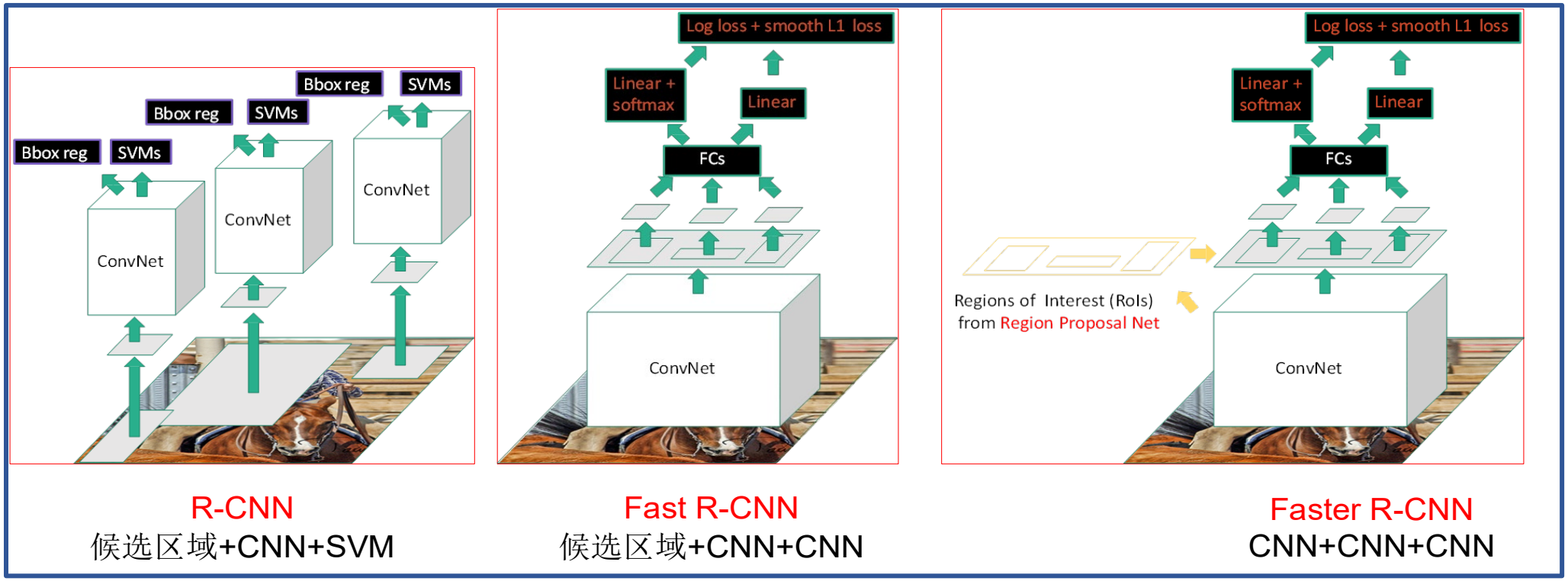
# 相关工作介绍—深度学习

深蓝学院 shenlanxueyuan.com

## 算法流程

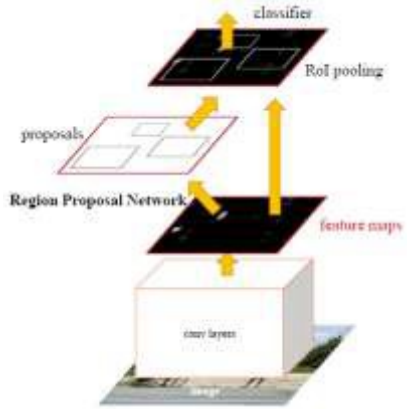

- 基于区域：

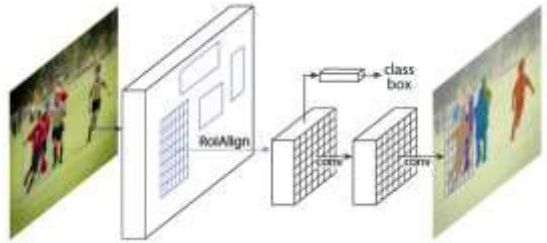
R-CNN CVPR2014
Fast RCNN ECCV2014
Faster RCNN NIPS2015




- 基于回归：YOLO CVPR2016，SSD ECCV2016
- 基于RNN：IONet CVPR2016

# 相关工作介绍—深度学习




RCNN

Faster RCNN

Mask RCNN

2015.4.30

2015.6.8

2015.12.08

2015.12.25

2017.1.23

2013.11.11

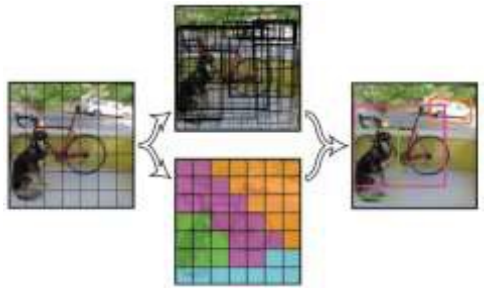
2015.5.14

2017.3.20

Fast RCNN

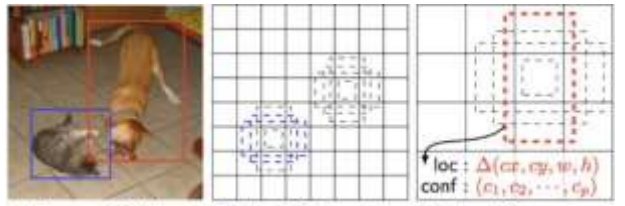
Yolo

Yolo v2



SSD

DSSD

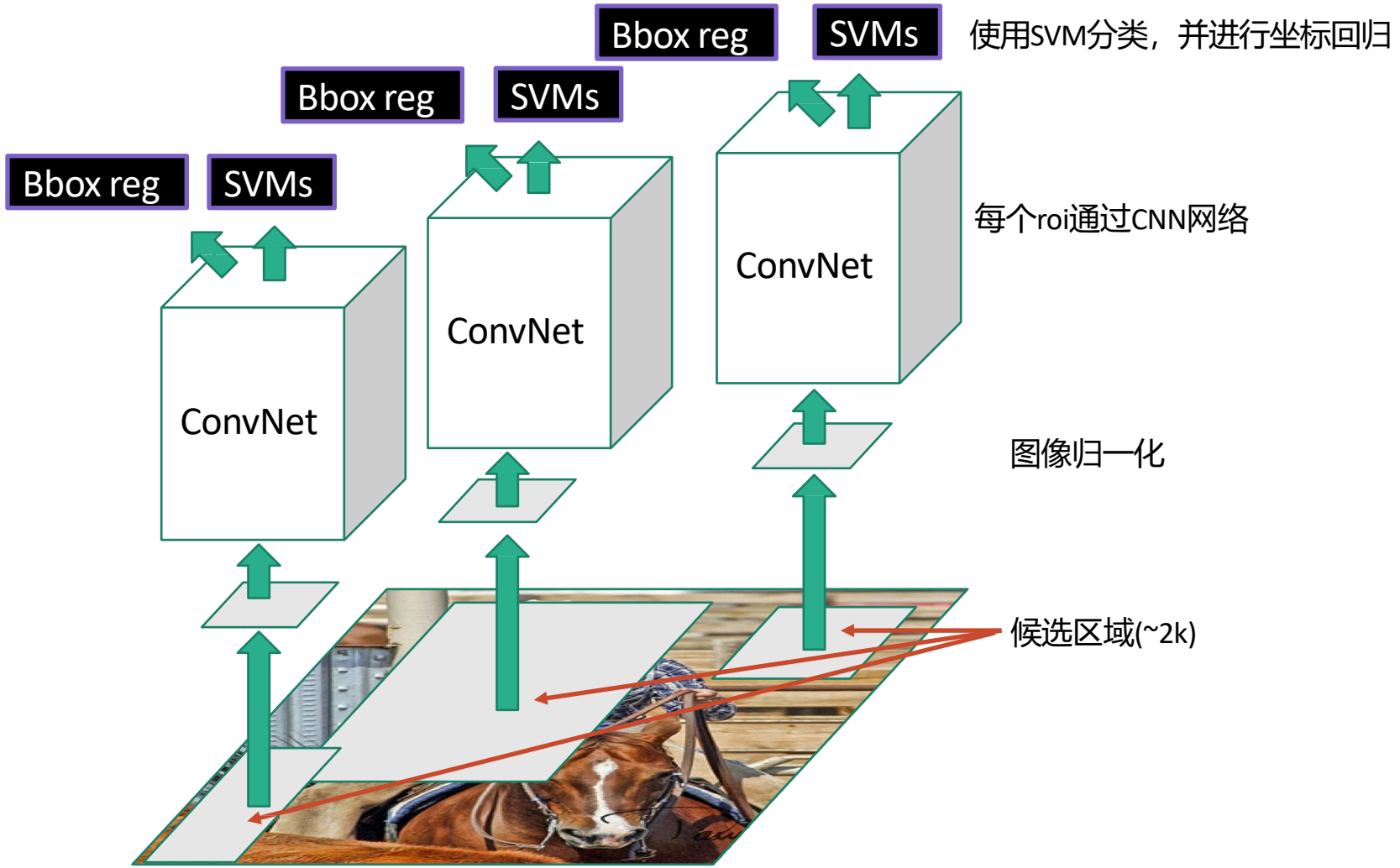



2012.12
**AlexNet**

2014.9
**VggNet &**
**InceptionNet**

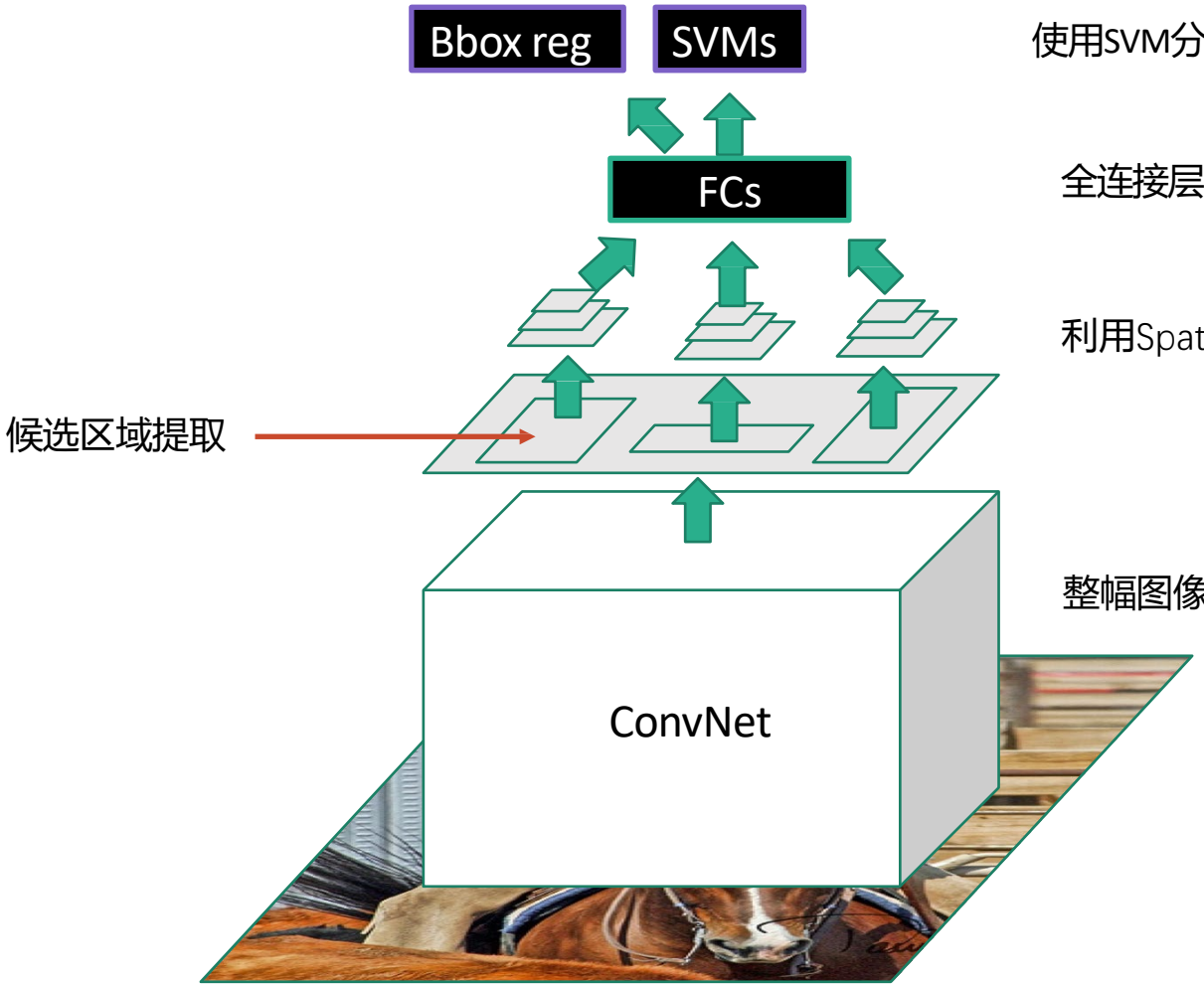
2015.12.10
**ResNet**

# 相关工作介绍--RCNN




“Rich feature hierarchies for accurate object detection and semantic segmentation” Girshick R, Donahue J, Darrell T, et al. **CVPR 2014**

# 相关工作介绍—SPP Net




使用SVM分类，并进行坐标回归

全连接层

利用Spatial Pyramid Pooling (SPP) 池化层提取候选区域部分的特征

整幅图像输入CNN网络，提取特征

“Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition” K. He, X. Zhang, S. Ren, J. Sun. **ECCV 2014**

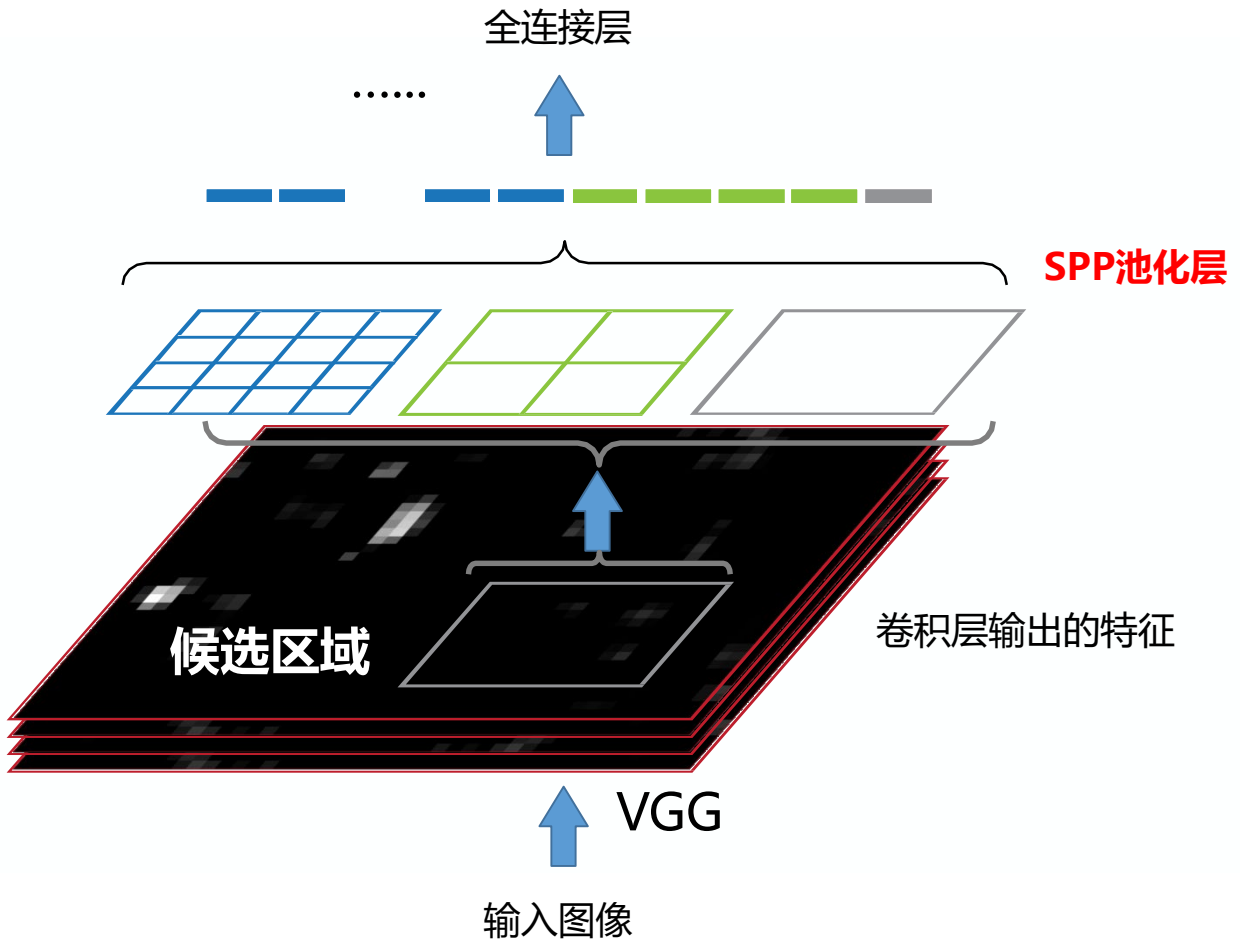
# 相关工作介绍—SPP Net

深蓝学院 shenlanxueyuan.com

## SPP池化层（Spatial Pyramid Pooling）

输入特征大小：w*h*c

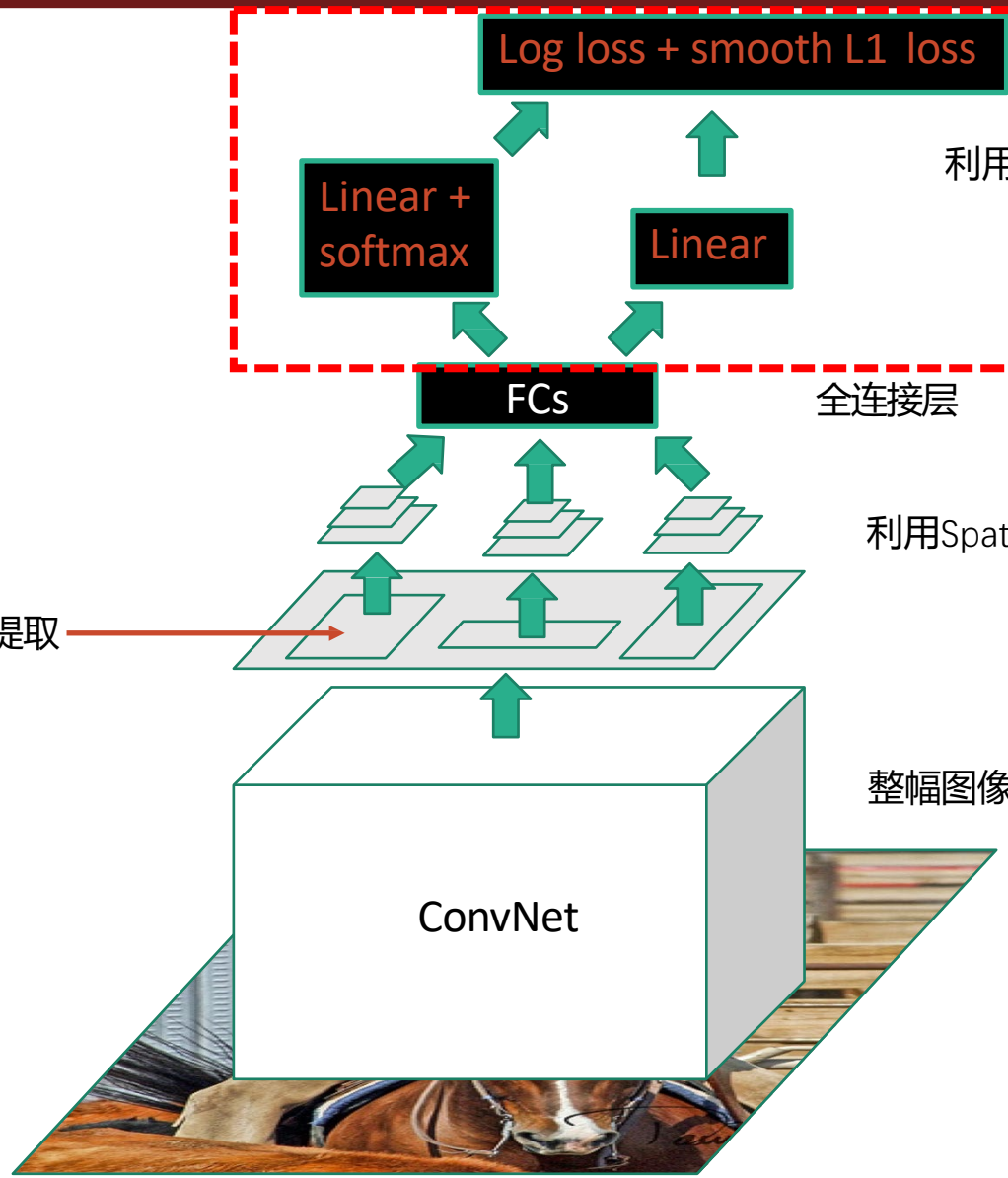
提取后特征尺寸：4*4*c+2*2*c+1*1*c




不同尺寸候选区域提取到同样维度的特征，便于输入全连接层

“Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition” K. He, X. Zhang, S. Ren, J. Sun. **ECCV 2014**

# 相关工作介绍—Fast RCNN




Multi-task loss

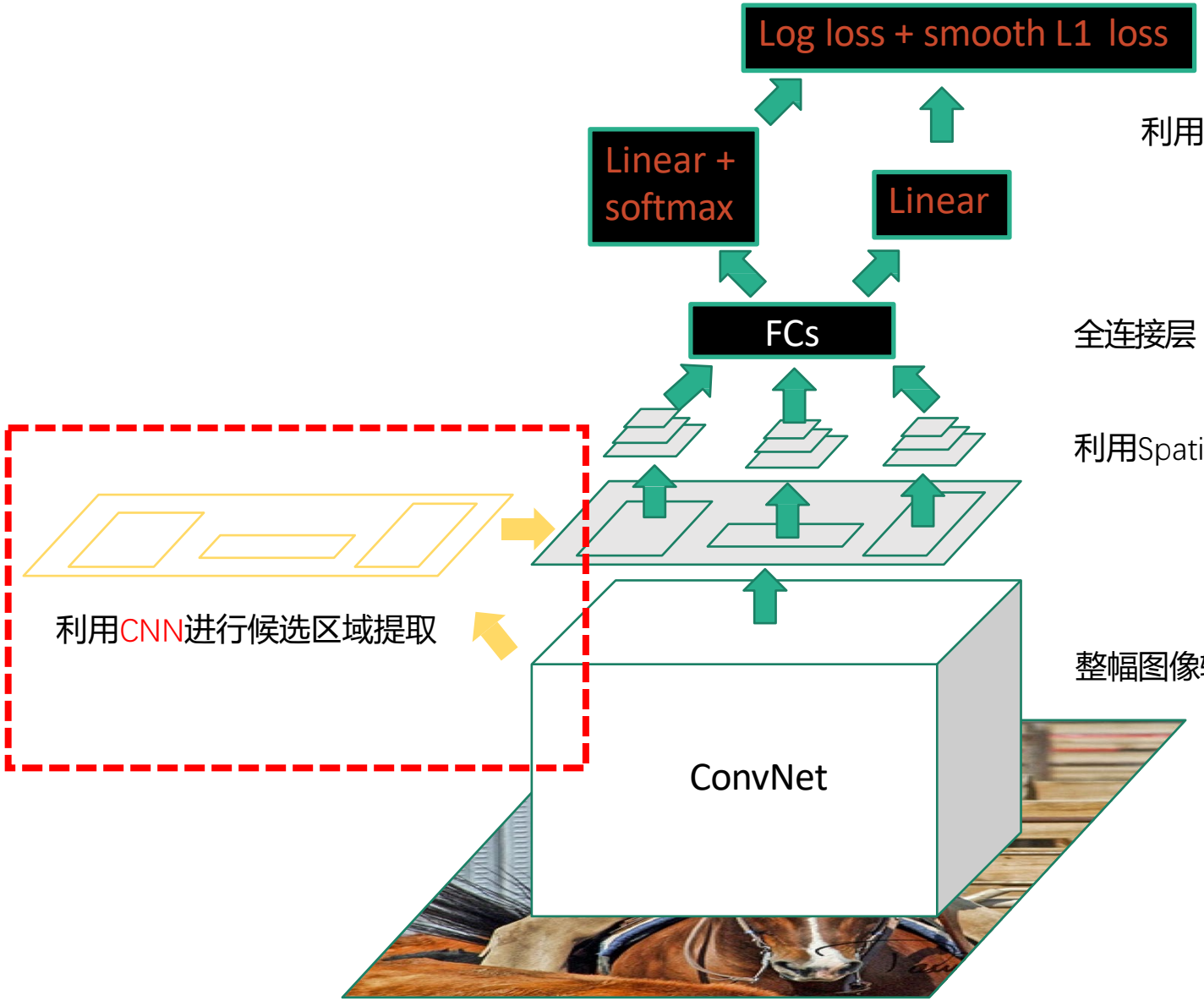
利用CNN进行分类和坐标回归

全连接层

利用Spatial Pyramid Pooling (SPP) 池化层提取候选区域部分的特征

整幅图像输入CNN网络，提取特征

# 相关工作介绍—Fast RCNN




Multi-task loss

利用CNN进行分类和坐标回归

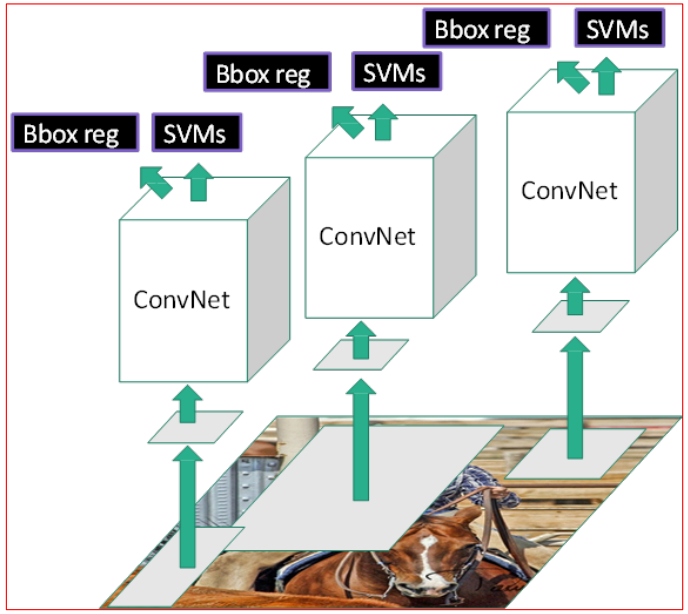
全连接层

利用Spatial Pyramid Pooling (SPP) 池化层提取候选区域部分的特征

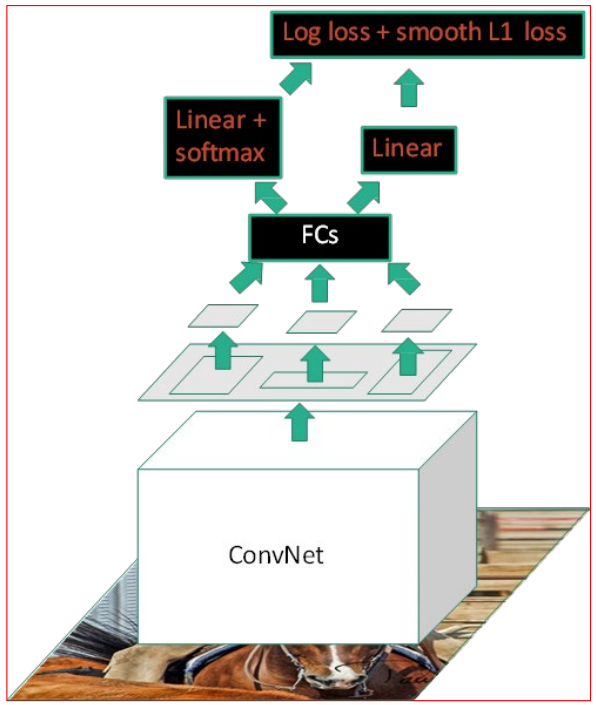
整幅图像输入CNN网络，提取特征

# 相关工作介绍—RCNN 系列

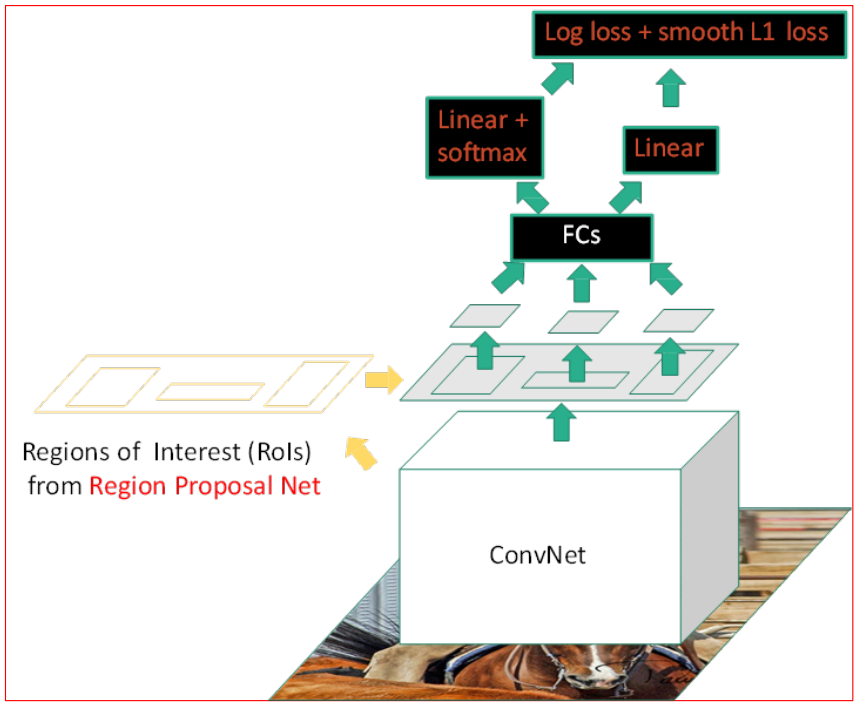





R-CNN
proposals+CNN+SVM




Fast R-CNN
proposals+CNN+CNN



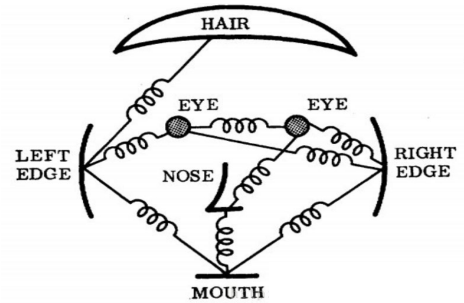

Faster R-CNN
CNN+CNN+CNN

# 传统方法和深度学习目标识别流程对比



词袋模型

可变部件模型



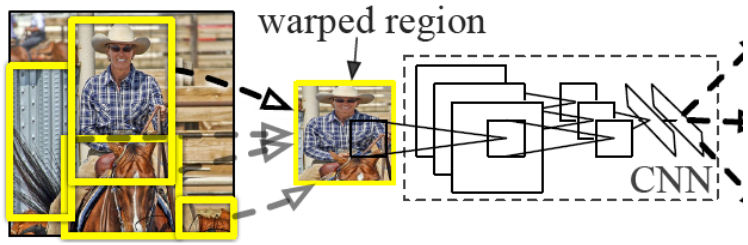

**传统算法：**

图像 → 候选区域 → 特征提取 → 分类器 → 结果

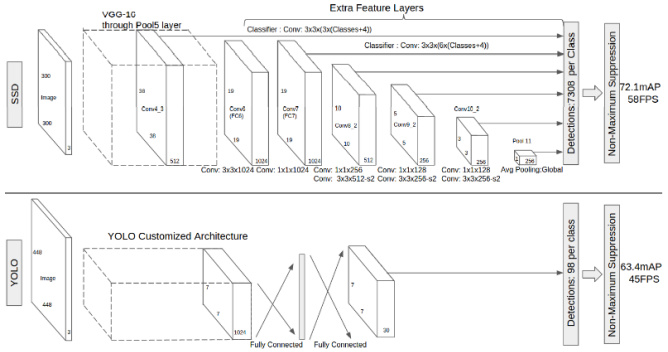
**深度学习：**

图像 → 神经网络 → 结果

优点：
端到端
可塑性
普适性




R-CNN




SSD，YOLO

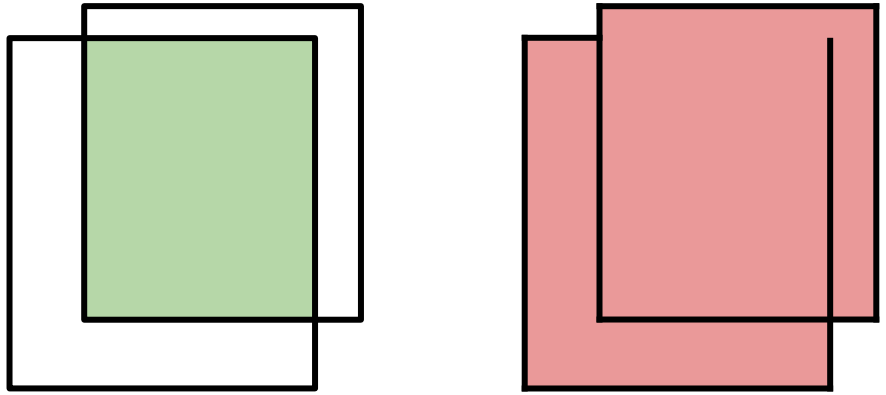
# 一些基础知识

深蓝学院
shenlanxueyuan.com

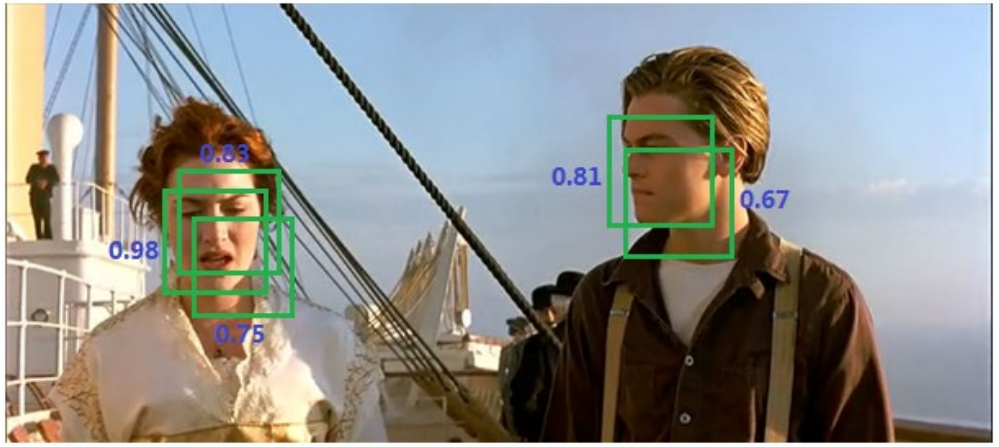
## Intersection Over Union (IoU)

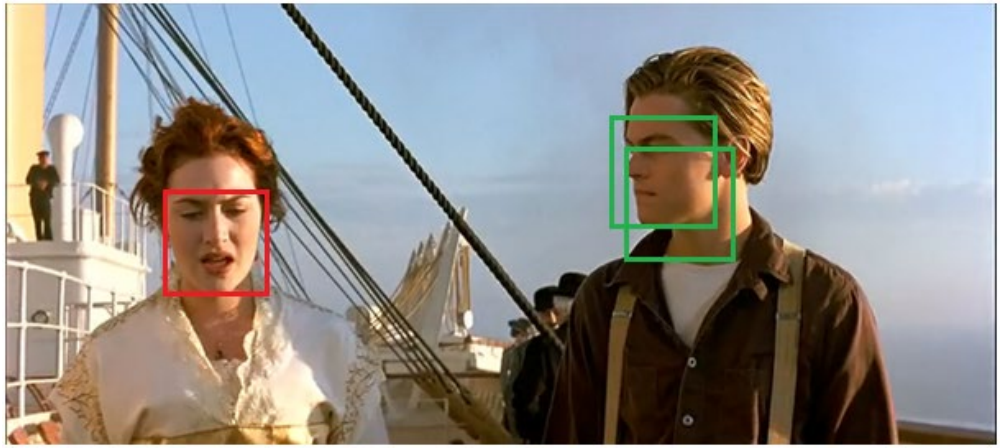
测量物体定位是否精准的测量方法

$$IoU(A,B) = \frac{Intersection(A,B)}{Union(A,B)}$$



## non maximum suppression(NMS)



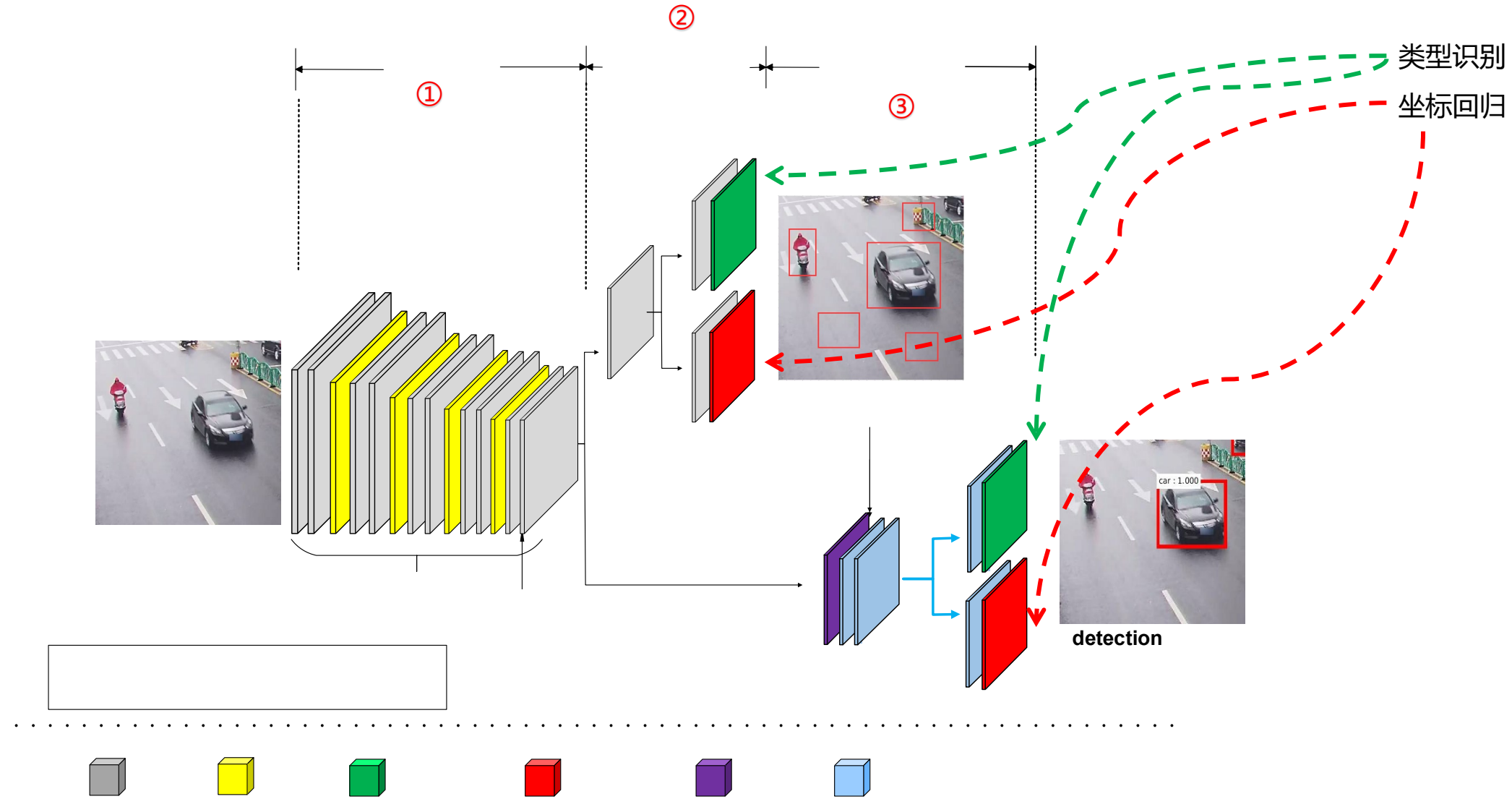



# 课程内容

深蓝学院
shenlanxueyuan.com

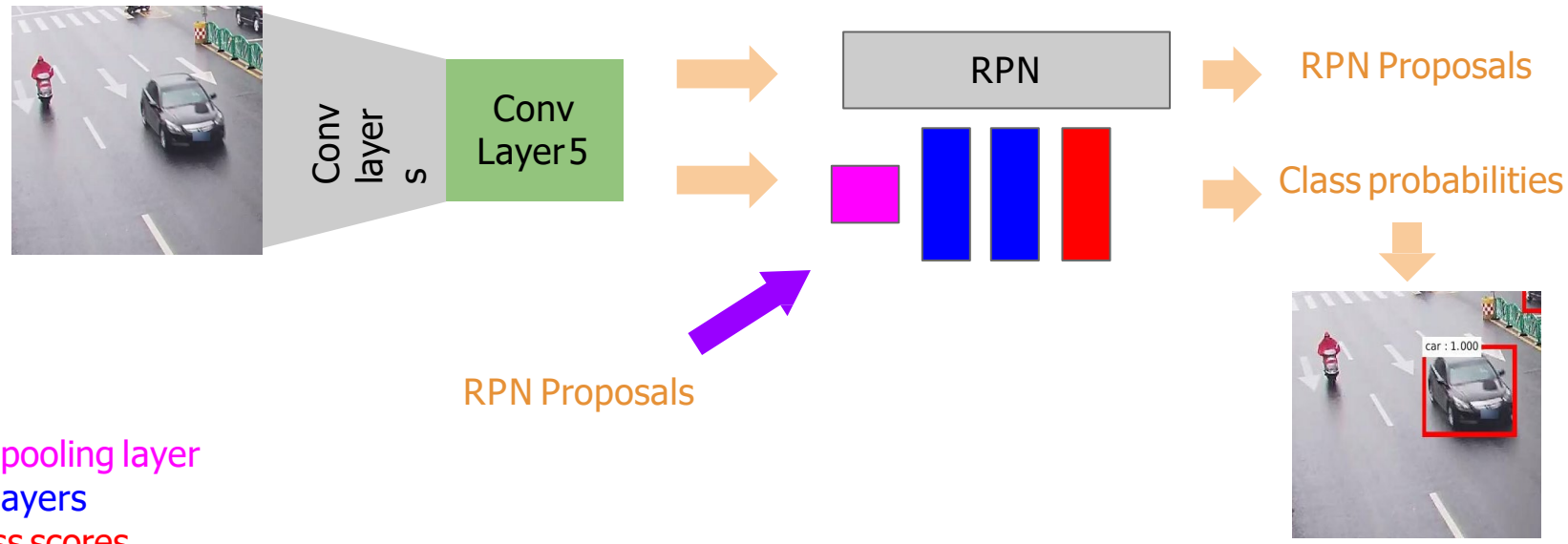
- ✓目标识别介绍
- ✓传统方法和深度学习对比
- ✓**Faster RCNN介绍**
- ✓Faster RCNN代码详解

# Faster RCNN 网络结构



# Faster RCNN 网络结构

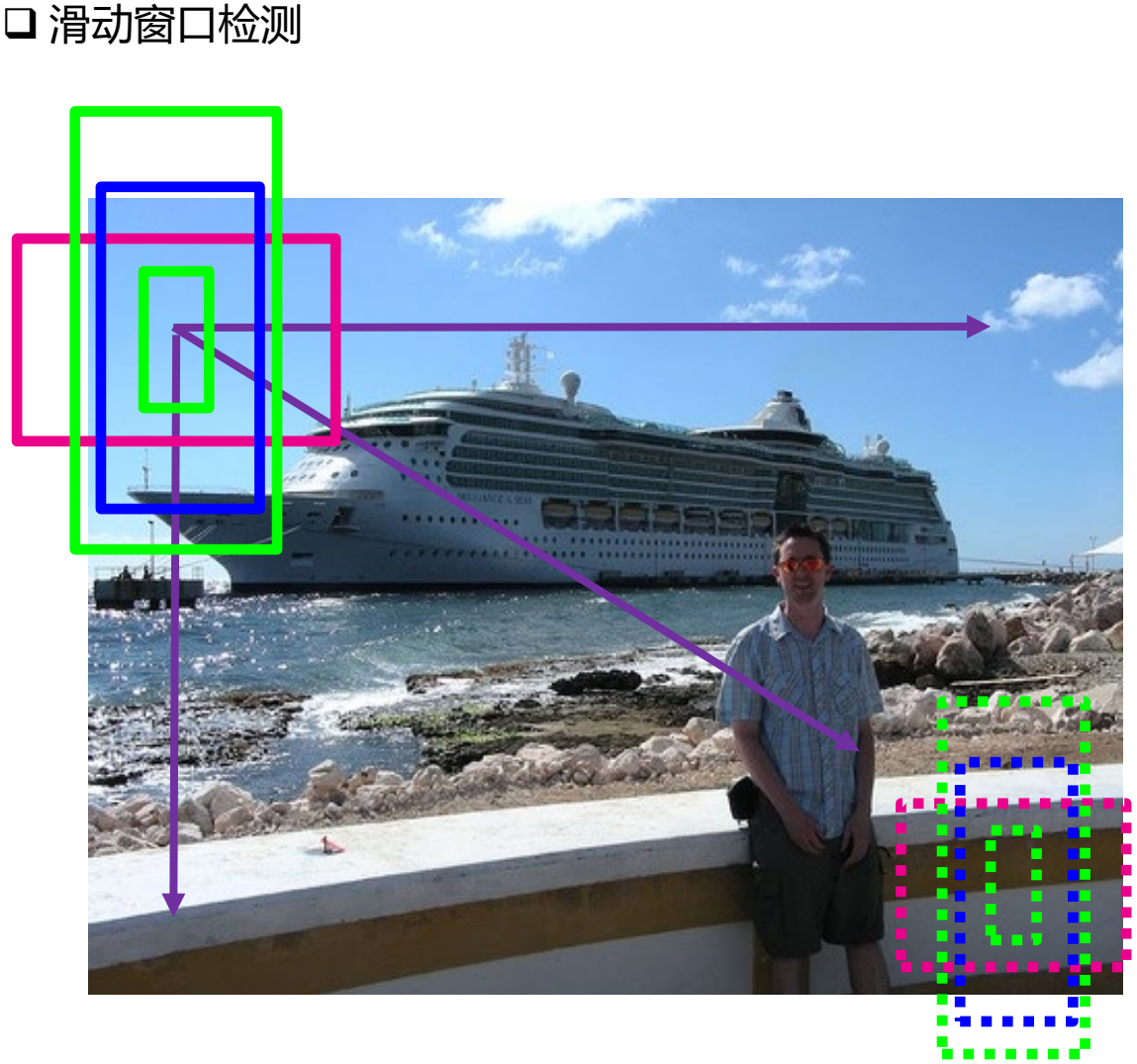
- 特征提取网络：VGG、ResNet(去掉最后pooling层，下采样均为16)
- 候选区域提取网络（RPN），粗略获取物体潜在位置
- 候选区域分类网络：对RPN输出的候选区域进行多类分类以及位置精确调整
- 候选区域提取和分类网络共享特征提取部分

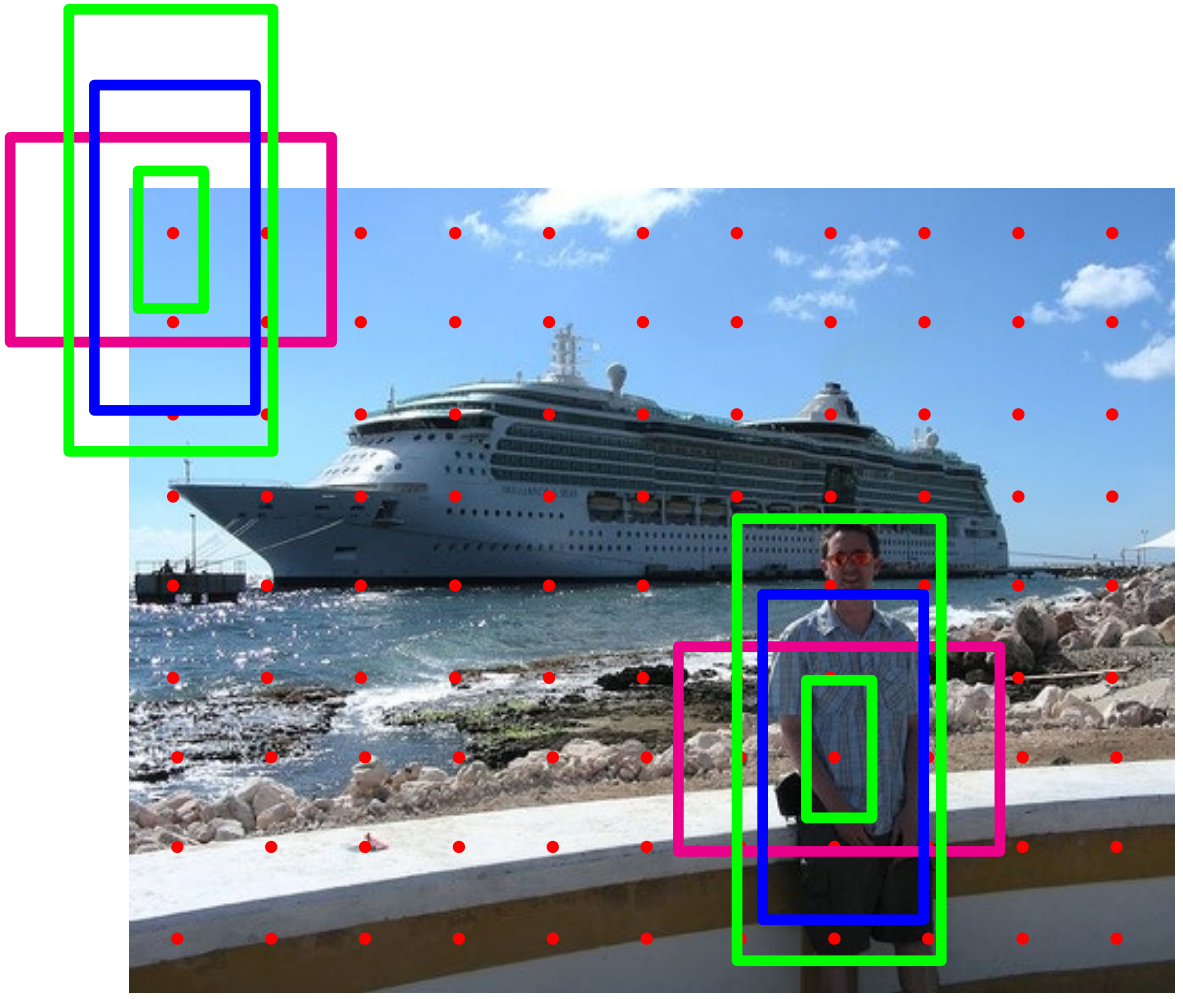


# 候选区域提取

深蓝学院
shenlanxueyuan.com

- 滑动窗口检测





# 候选区域提取

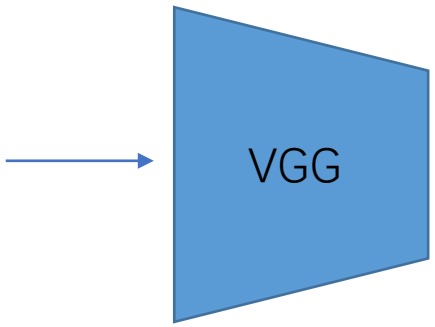
深蓝学院 shenlanxueyuan.com

❑ Region Proposal Network（RPN）

- Anchor：9种尺度（3种尺度，3种长宽比）
- 锚点数量为最后feature map大小，位置为原图中感受野的中心
- RPN的训练过程可以看做图像分割过程，对feature map上每个点进行分类（预测以当前点为中心，哪些anchor覆盖物体，哪些anchor属于背景）

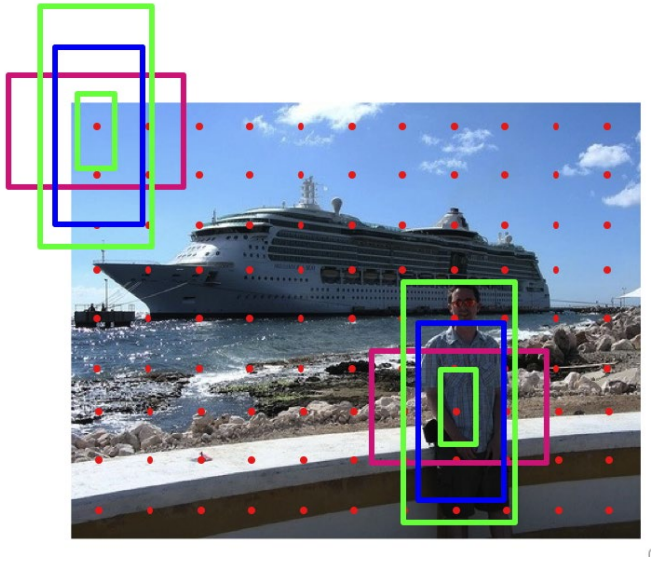
| anchor | $128^2$, 2:1 | $128^2$, 1:1 | $128^2$, 1:2 | $256^2$, 2:1 | $256^2$, 1:1 | $256^2$, 1:2 | $512^2$, 2:1 | $512^2$, 1:1 | $512^2$, 1:2 |
|---|---|---|---|---|---|---|---|---|---|
| proposal | 188×111 | 113×114 | 70×92 | 416×229 | 261×284 | 174×332 | 768×437 | 499×501 | 355×715 |

600*1000




下采样16倍


约40*60

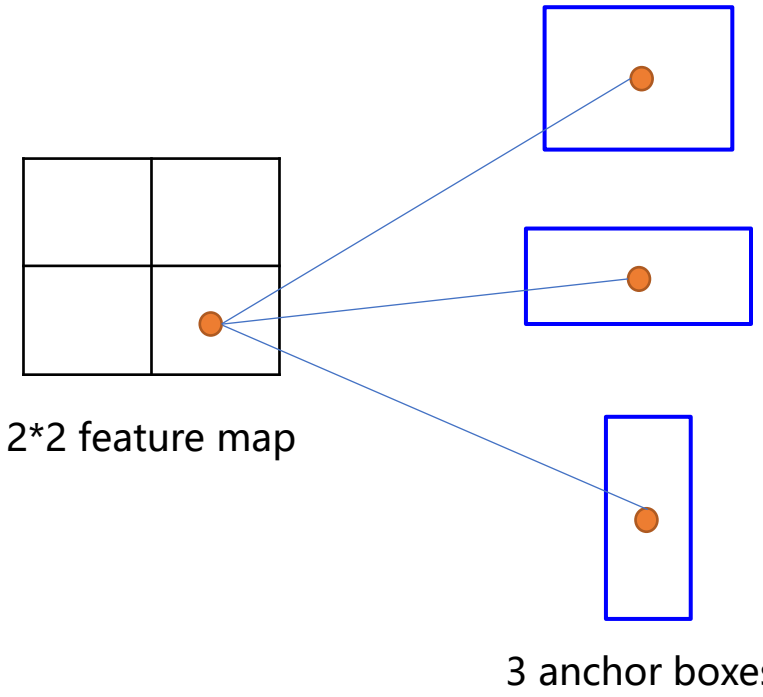


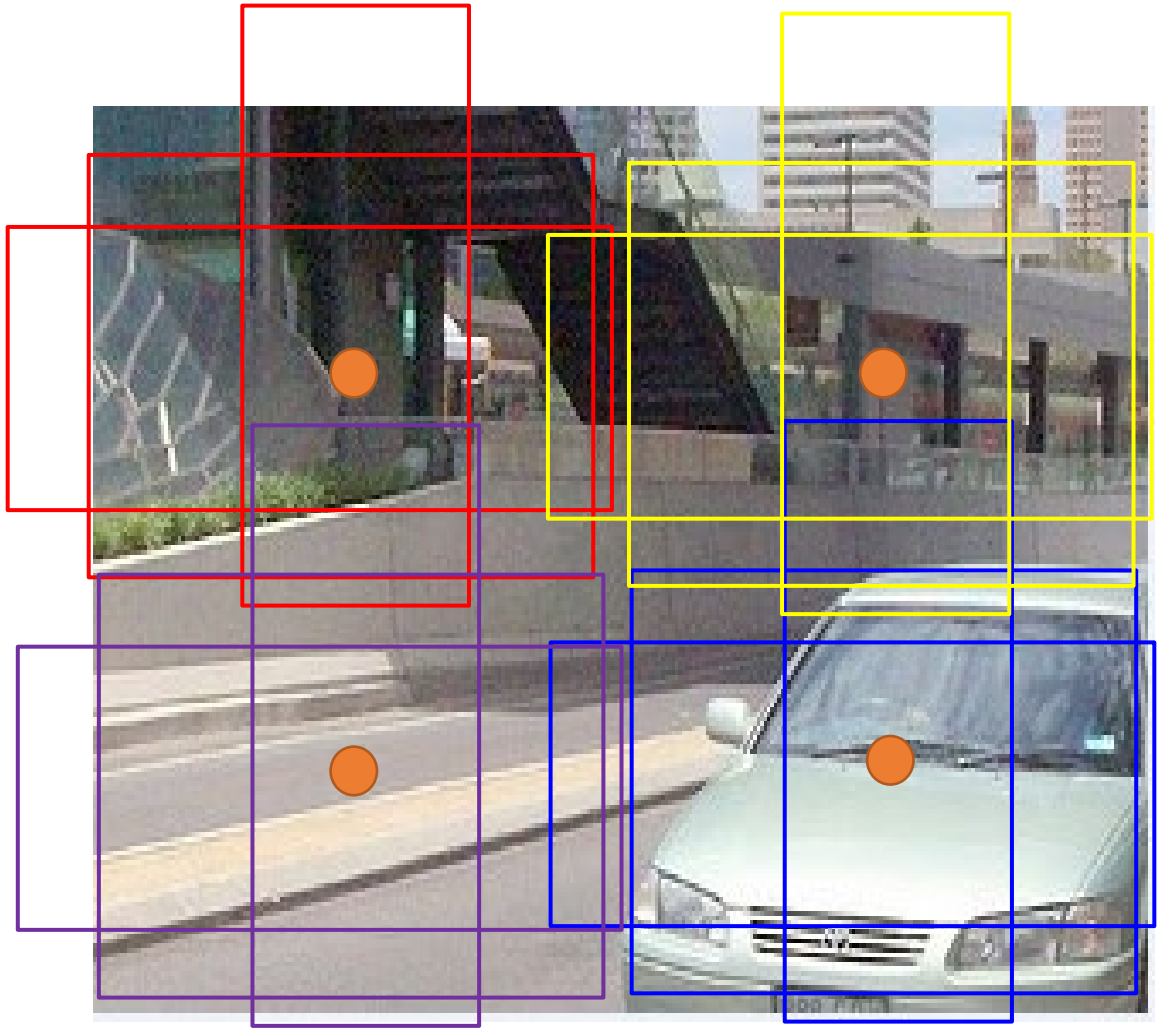
# 候选区域提取RPN



举例：

Feature map大小为2*2， 3种预设矩形框（anchor box）



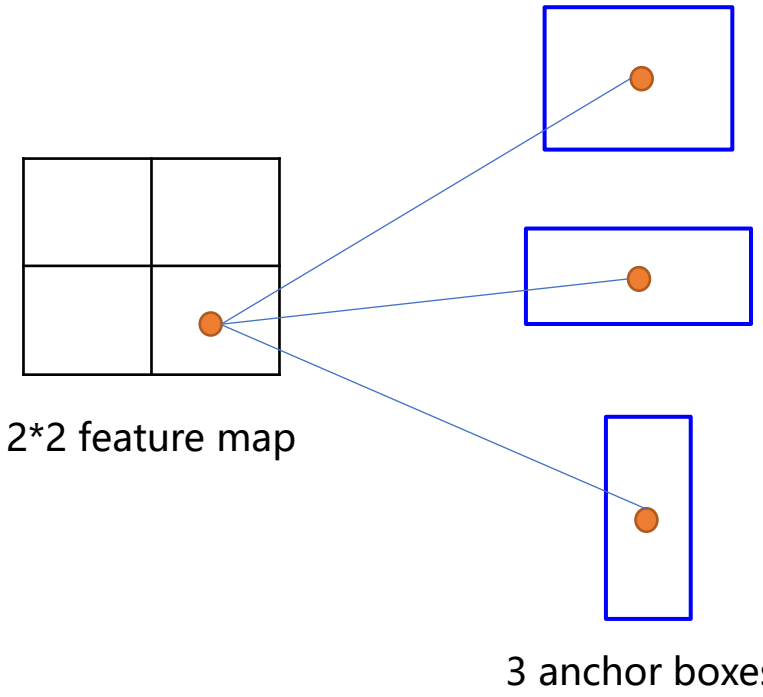



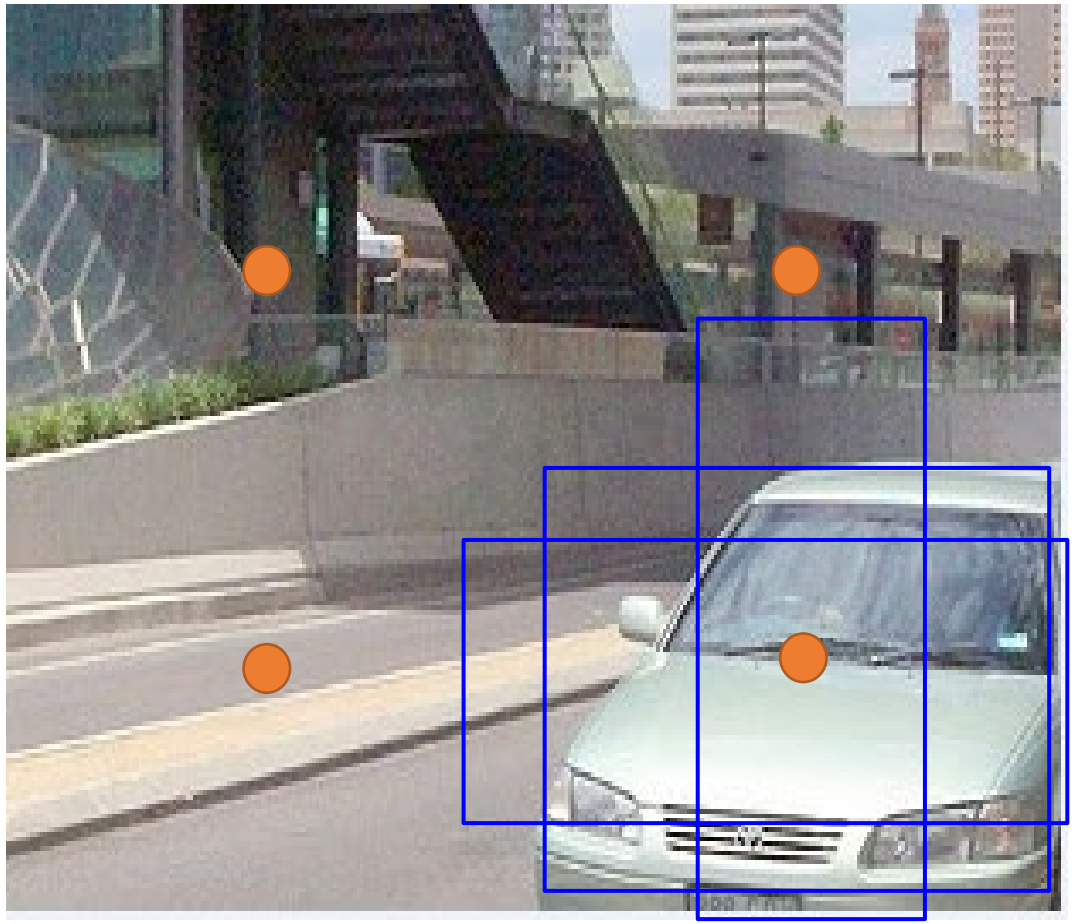
# 候选区域提取RPN



举例：

Feature map大小为2*2， 3种预设矩形框（anchor box）



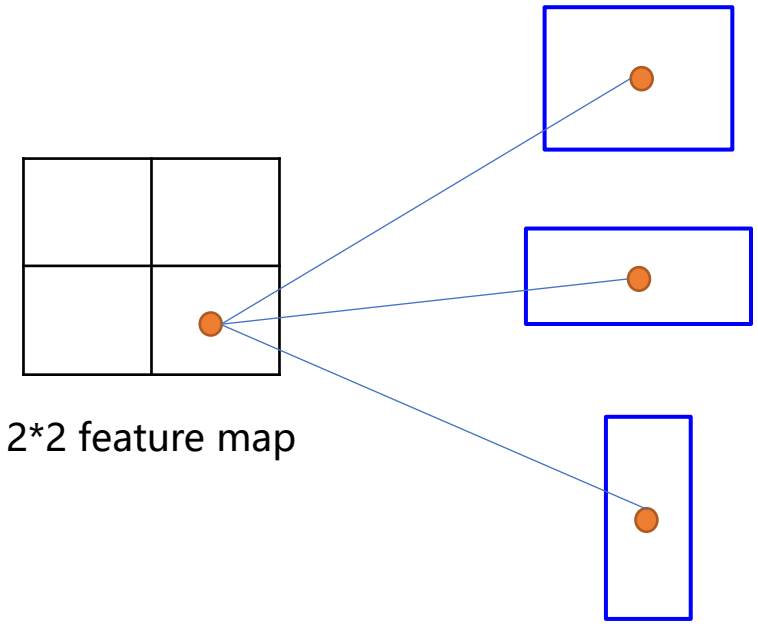



# 候选区域提取RPN



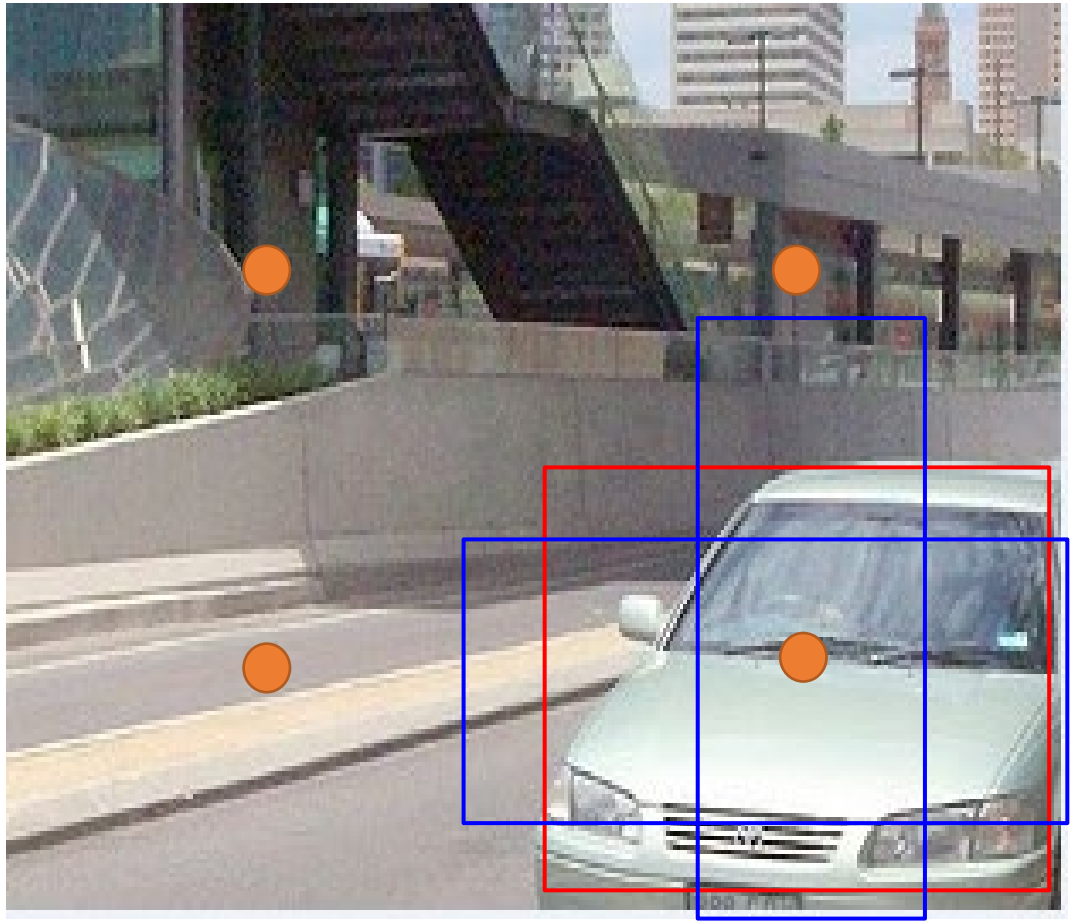
举例：

Feature map大小为2*2， 3种预设矩形框（anchor box）



2*2 feature map

3 anchor boxes



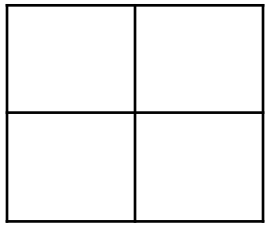
IOU > 0.7

# 候选区域提取RPN



举例：

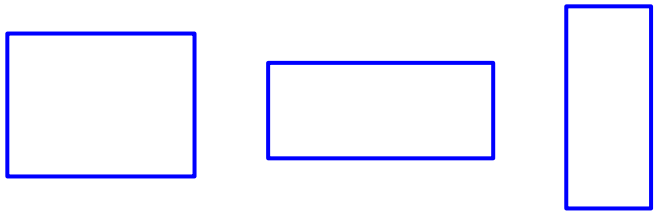
Feature map大小为2*2， 3种预设矩形框（anchor box）



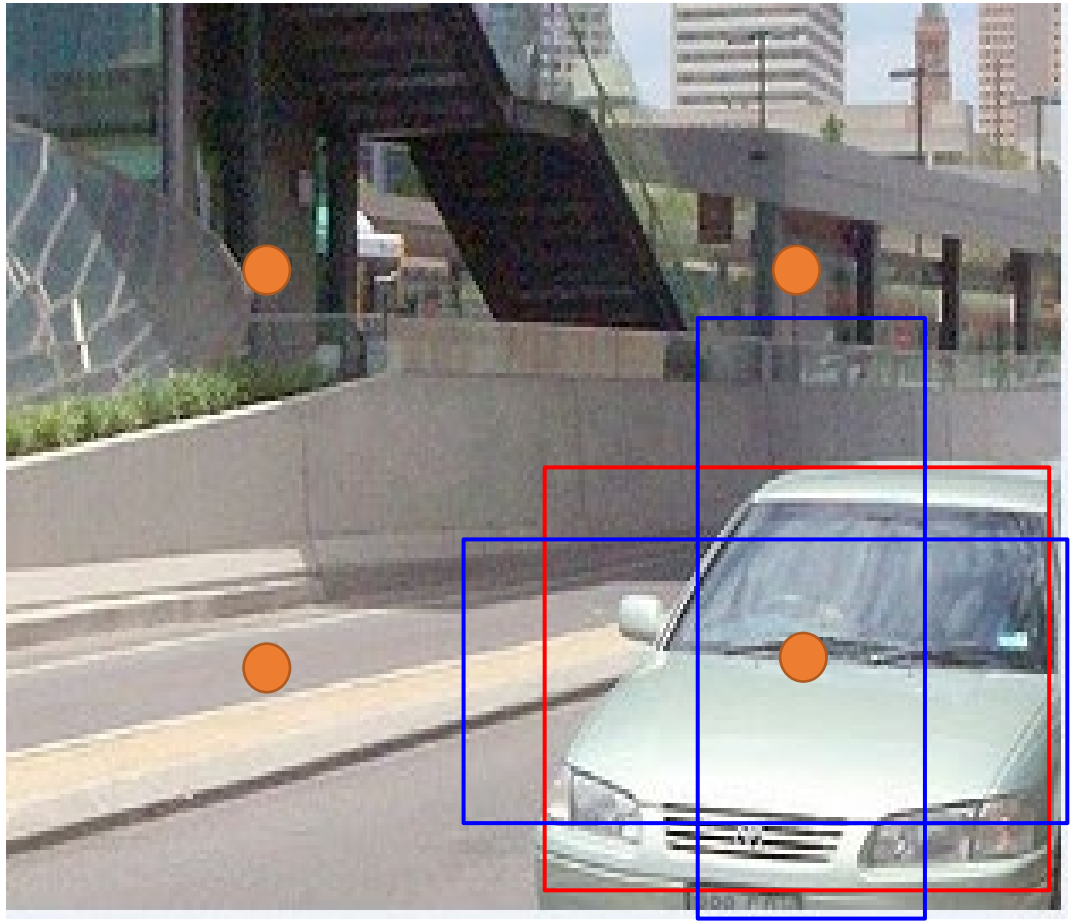
2*2 feature map

| | |
|---|---|
| 0 | 0 |
| 0 | 1 |

标签



0-背景　　1　　2　　3

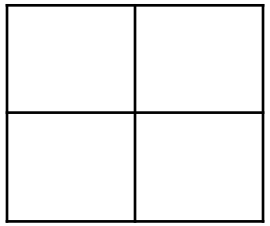


# 候选区域提取RPN



举例：

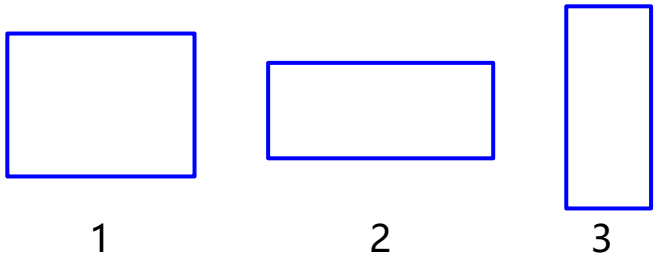
Feature map大小为2*2， 3种预设矩形框（anchor box）



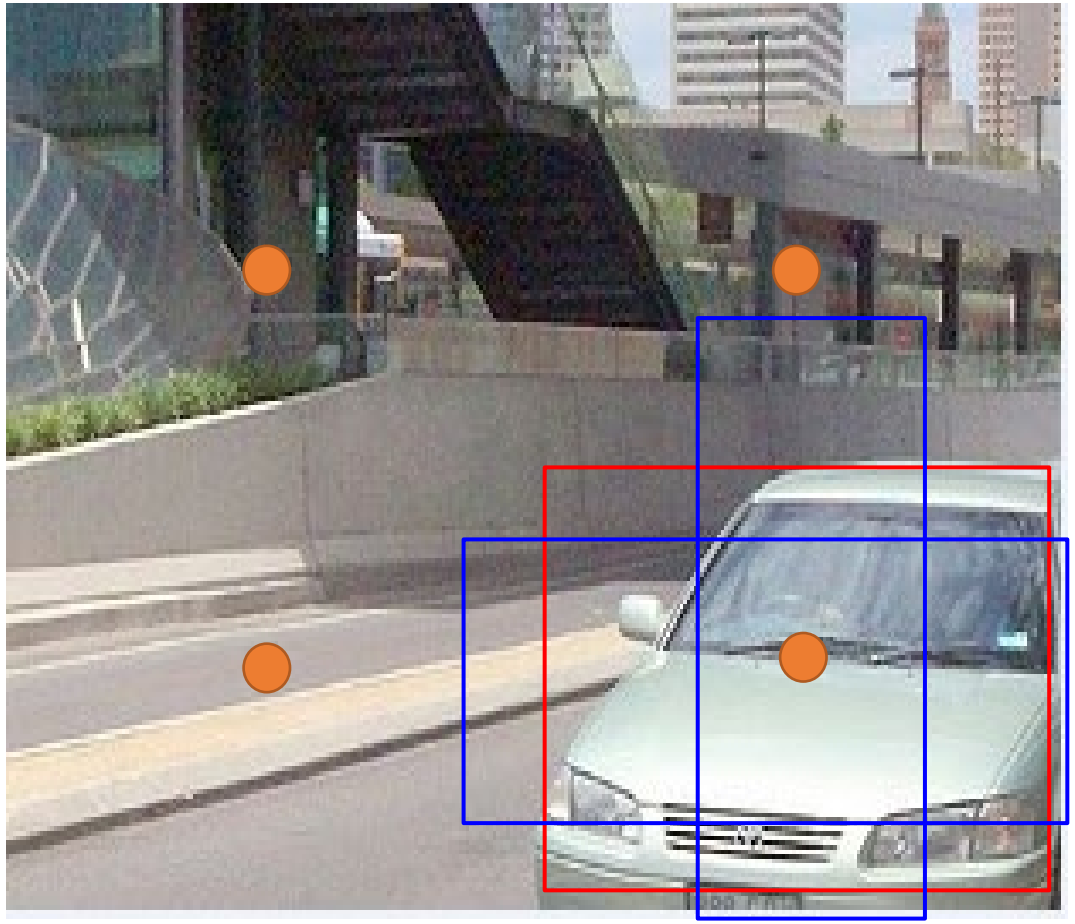
2*2 feature map

| | |
|---|---|
| 000 | 000 |
| 000 | 100 |

标签



1 2 3



为了降低学习难度，并产生更多的候选区域，采取二分类方式，并非one-hot的编码方式。

# RPN网络结构

深蓝学院 shenlanxueyuan.com

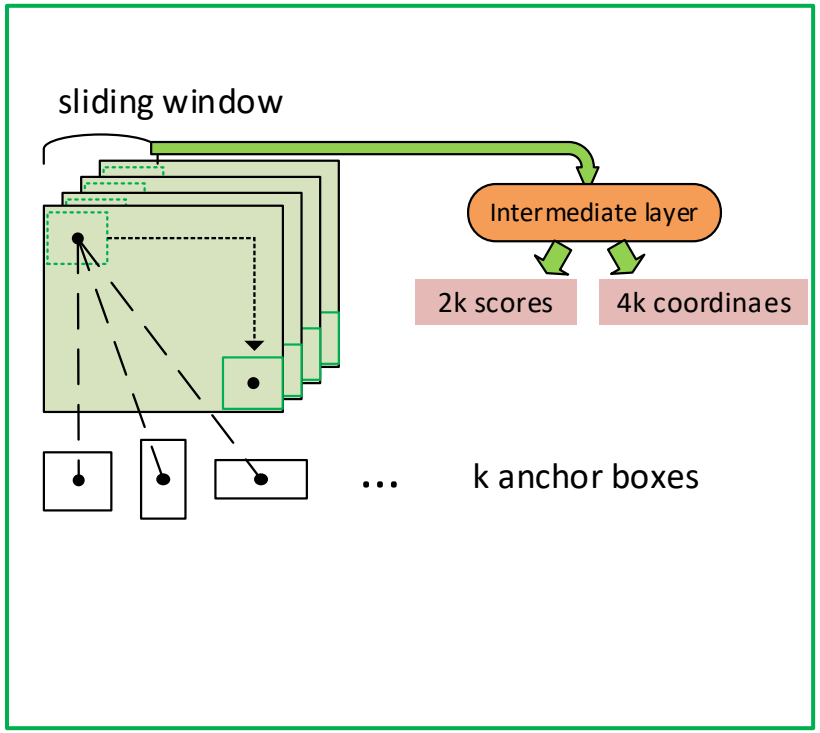
## 前向传播

输入图像大小为$H \times W$，经过VGG得到$\frac{H}{16} \times \frac{W}{16}$的特征，输入到RPN网络得到：

分类的feature map：$N \times 2K \times \frac{H}{16} \times \frac{W}{16}$

坐标回归的feature map：$N \times 4K \times \frac{H}{16} \times \frac{W}{16}$

每个锚点预设$K$种anchor box， faster RCNN中$K = 9$




## Loss计算

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

分类损失　　坐标回归损失

# RPN网络结构

深蓝学院 shenlanxueyuan.com

- 分类损失计算：

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

- Label生成：

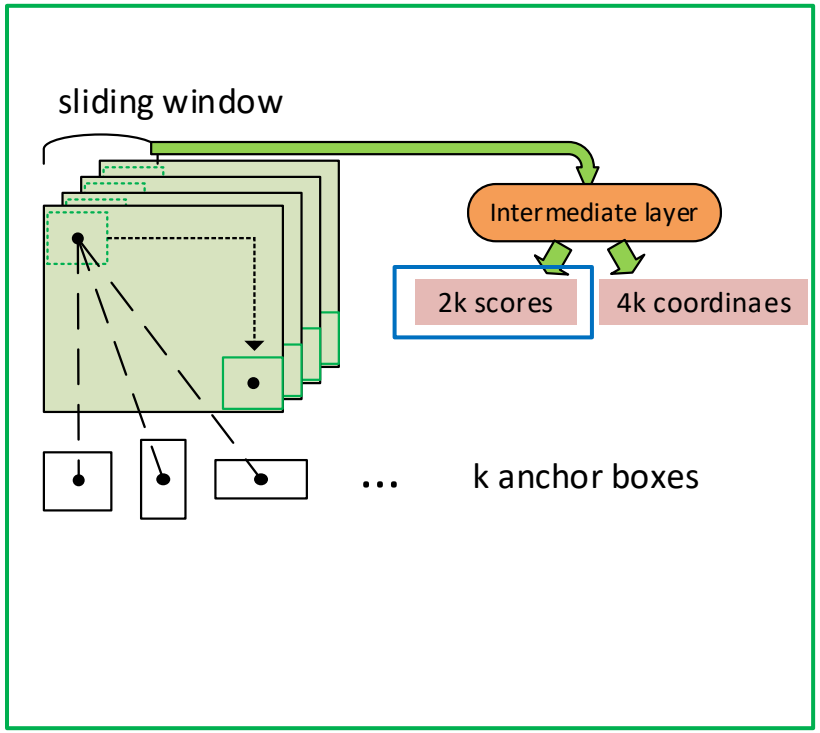
$p_i, p_i^*$：预测的label和真实的label

$L_{cls}$：交叉熵损失

- pi* = 1 if IoU > 0.7
- pi* = 0 if IoU < 0.3
- otherwise, do not contribute to loss

忽略IoU阈值在0.3-0.7之间的anchor box

(loss weight乘0即可实现)



# RPN网络结构

深蓝学院 shenlanxueyuan.com

- 坐标回归损失计算：

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

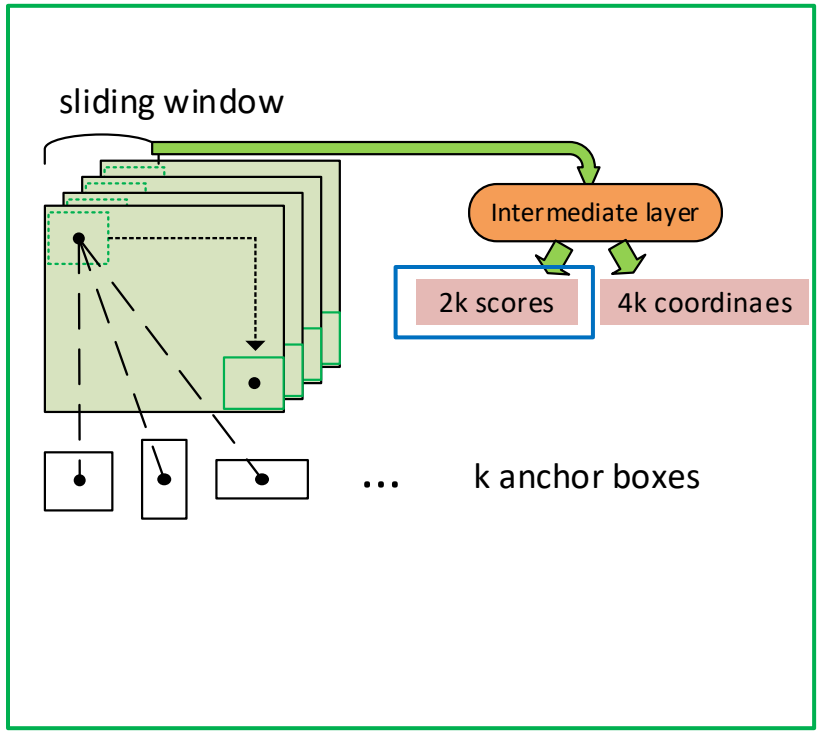
$$L_{loc}(t,t^*) = \sum_{i\in\{x,y,w,h\}} smooth_{L_1}(t_i,t_i^*)$$

其中，

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & if\ |x|<1 \\ |x|-0.5 & otherwise \end{cases}$$

$t_i\ t_i^*$：预测位置偏差和真实的位置偏差

$t = \{x\ y\ w\ h\}$ （左上角坐标$xy$，宽高$wh$）




- Label生成

$$t_x^* = (x^* - x_a)/w_a$$
$$t_y^* = (y^* - y_a)/h_a$$
$$t_w^* = \log(w^*/w_a)$$
$$t_h^* = \log(h^*/h_a)$$

预测阶段：

$$x = t_x * w_a + x_a$$
$$y = t_y * h_a + h_a$$
$$w = \exp(t_w) * w_a$$
$$h = \exp(t_h) * h_a$$

# 候选区域分类网络—Fast RCNN

深蓝学院 shenlanxueyuan.com

- 分类网络

  作用：对RPN输出的候选区域进行分类（多类），并再次进行位置微调

  Loss：与RPN阶段类似

真实位置偏差

预测位置偏差

$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1]L_{\text{loc}}(t^u, v),$$

真实label

预测label

多类交叉熵损失

Smooth L1 loss

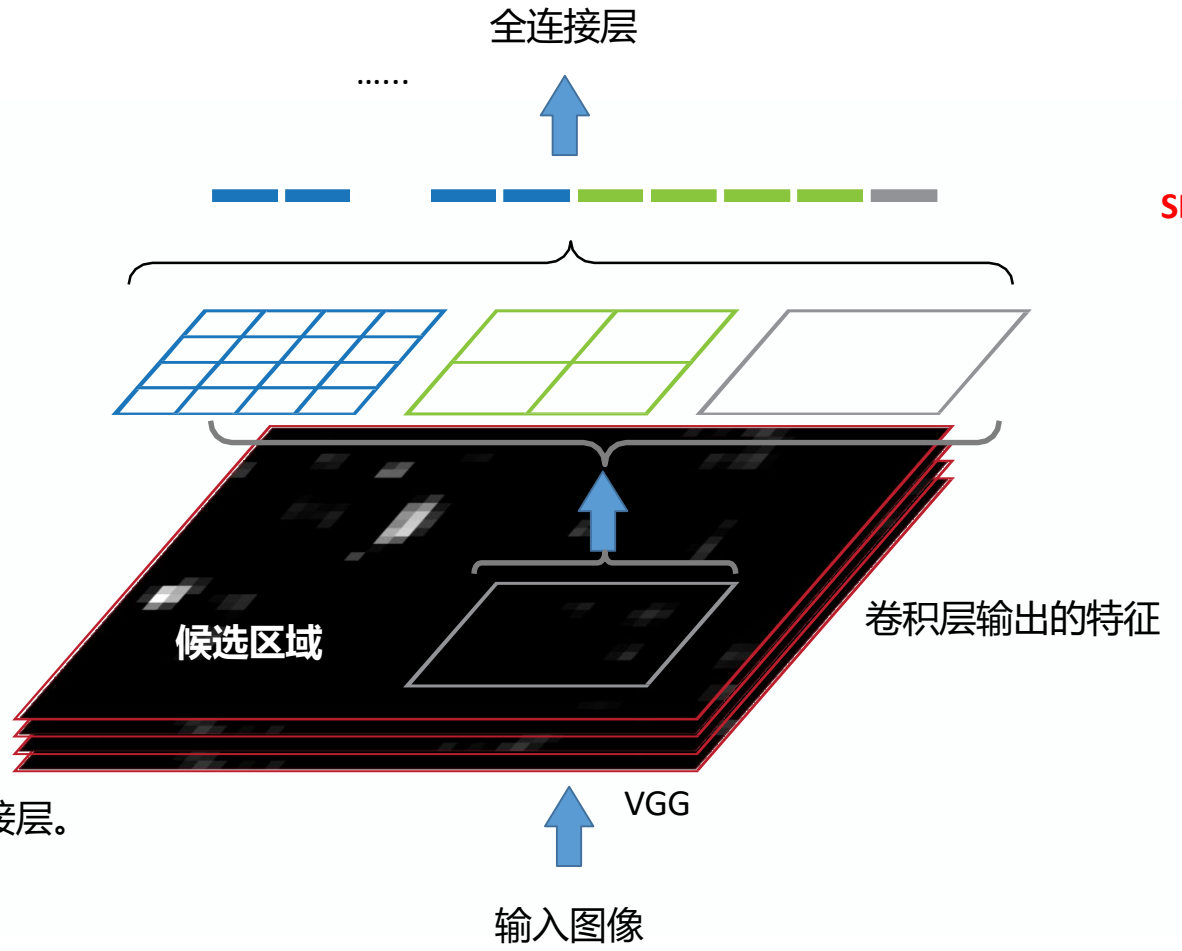
# Faster RCNN—分类阶段

深蓝学院 shenlanxueyuan.com

- SPP池化层（Spatial Pyramid Pooling）

输入特征大小：$w \times h \times c$

提取后特征尺寸：$4 \times 4 \times c + 2 \times 2 \times c + 1 \times 1 \times c$



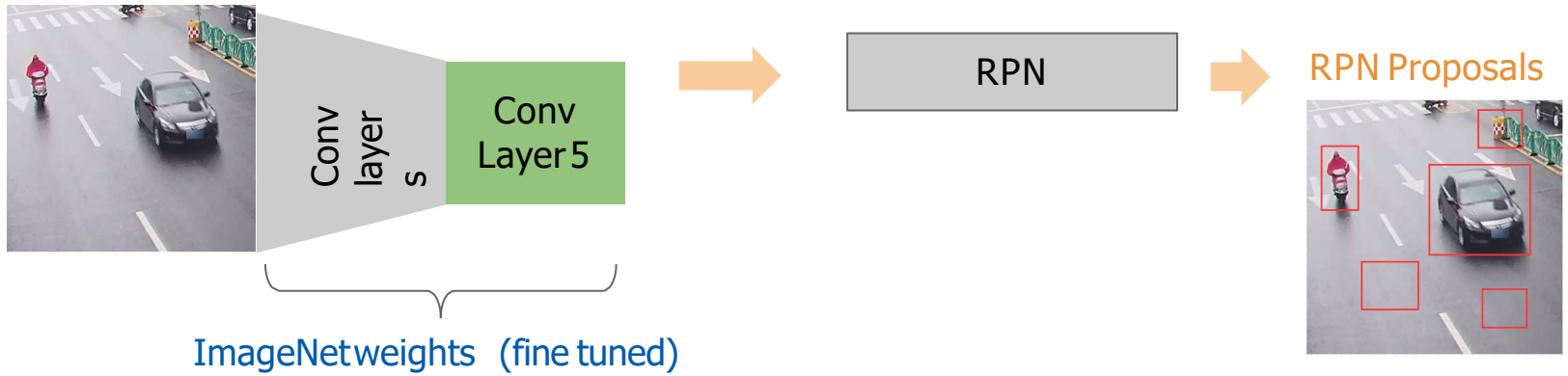

不同尺寸候选区域提取到同样维度的特征，便于输入全连接层。

# Faster RCNN 训练细节

深蓝学院 shenlanxueyuan.com

- 四步骤训练

Step 1: 利用ImageNet模型初始化VGG，训练RPN网络.

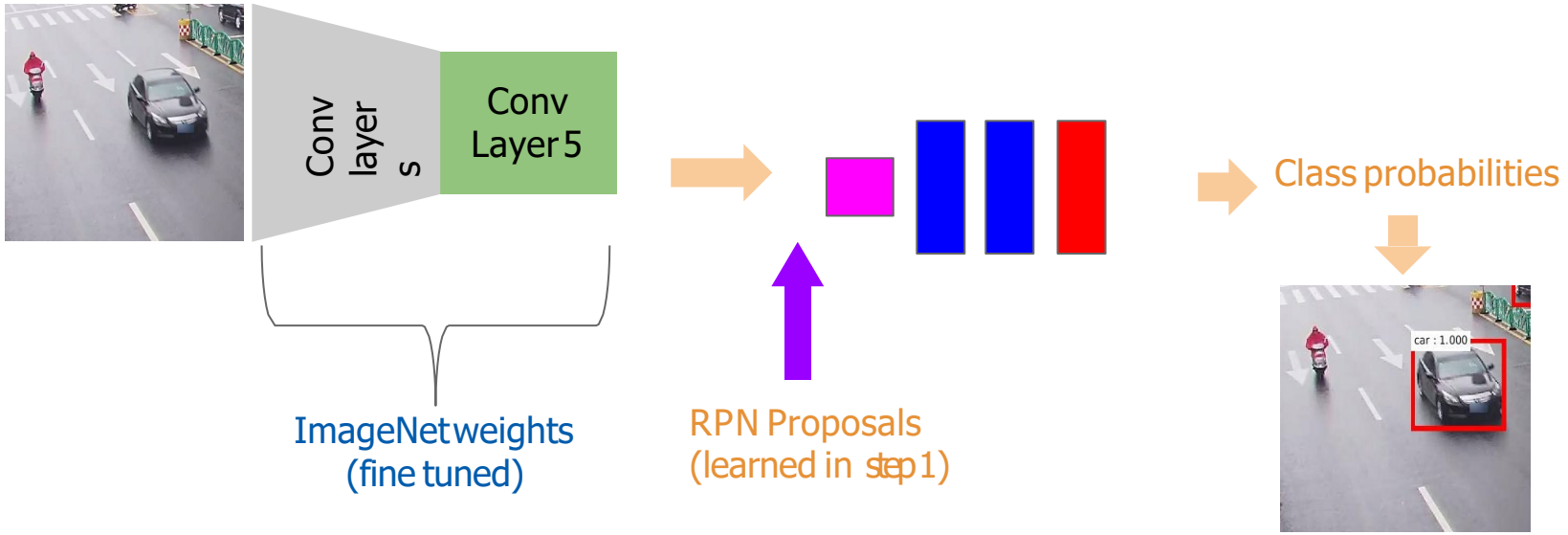


# Faster RCNN 训练细节

深蓝学院 shenlanxueyuan.com

- 四步骤训练

Step 2:利用ImageNet模型初始化VGG，训练分类网络（在此过程中调用RPN的输出）

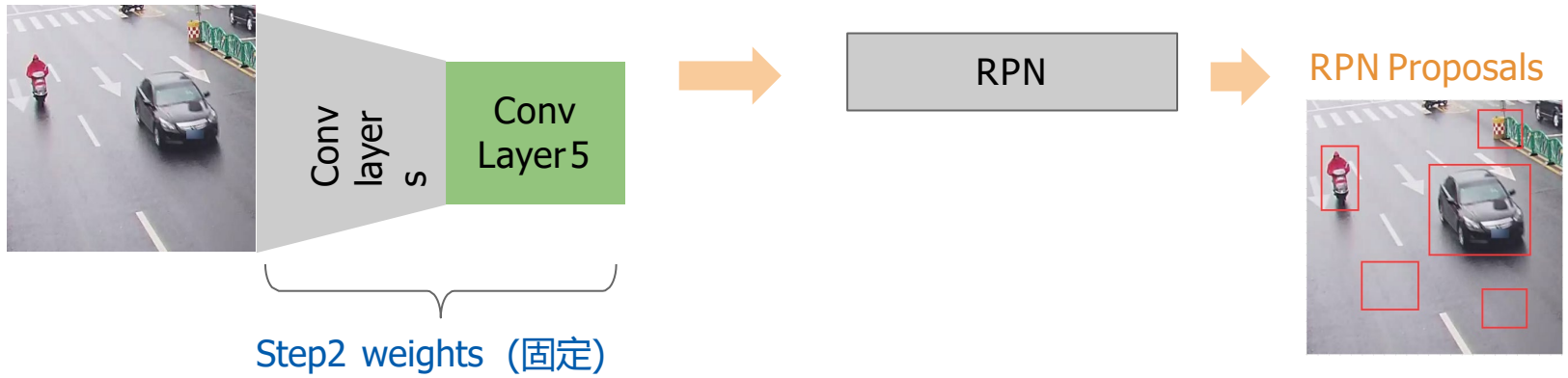


# Faster RCNN 训练细节

深蓝学院 shenlanxueyuan.com

- 四步骤训练

Step 3: 利用step 2模型初始化VGG，训练RPN网络.

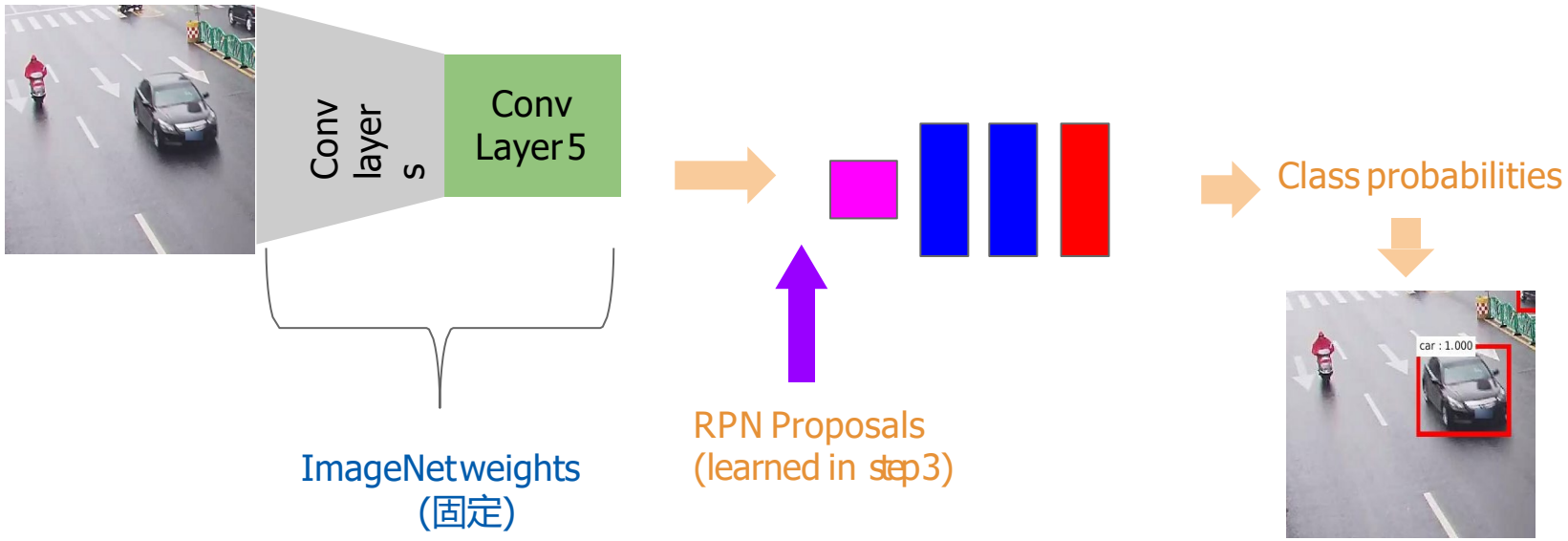


# Faster RCNN 训练细节

深蓝学院 shenlanxueyuan.com

- 四步骤训练

Step 4: 利用step 2&3模型初始化VGG，训练分类网络（在此过程中调用RPN的输出）



# 课程内容

深蓝学院
shenlanxueyuan.com

- ✓目标识别介绍
- ✓传统方法和深度学习对比
- ✓Faster RCNN介绍
- ✓**Faster RCNN代码详解**

# 前期准备

深蓝学院 shenlanxueyuan.com

- 运行环境: Linux, python 2.7/3.6, pytorch 0.4.0/0.4.1, CUDA 8.0+
- 安装必要的python包: pip install -r requirements.txt
- 下载并解压VOC数据集:

  cd faster-rcnn.pytorch && mkdir data

  ln -s /your/dataset VOCdevkit2007
- 下载vgg16 pretrain model:

  (https://www.dropbox.com/s/s3brpk0bdq60nyb/vgg16_caffe.pth?dl=0)

  cd data && mkdir pretrained_model

  ln -s /your/model vgg16_caffe.pth

1. Download the training, validation, test data and VOCdevkit

```
wget http://host.robots.ox.ac.uk/pascal/VOC/voc2007/VOCtrainval_06-Nov-2007.tar
wget http://host.robots.ox.ac.uk/pascal/VOC/voc2007/VOCtest_06-Nov-2007.tar
wget http://host.robots.ox.ac.uk/pascal/VOC/voc2007/VOCdevkit_08-Jun-2007.tar
```

2. Extract all of these tars into one directory named `VOCdevkit`

```
tar xvf VOCtrainval_06-Nov-2007.tar
tar xvf VOCtest_06-Nov-2007.tar
tar xvf VOCdevkit_08-Jun-2007.tar
```

3. It should have this basic structure

```
$VOCdevkit/                           # development kit
$VOCdevkit/VOCcode/                   # VOC utility code
$VOCdevkit/VOC2007                    # image sets, annotations, etc.
# ... and several other directories ...
```

4. Create symlinks for the PASCAL VOC dataset

```
cd $FRCN_ROOT/data
ln -s $VOCdevkit VOCdevkit2007
```

https://github.com/jwyang/faster-rcnn.pytorch

# 前期准备



- 编译一些辅助代码/层：

  cd lib && sh make.sh

- 目的：

  python setup.py build_ext --inplace

  - model/utils/bbox.pyx　　#计算box之间IoU
  - pycocotools/maskApi.c　　#读取coco数据集中二值掩膜

  model/nms/src/*　　#NMS gpu代码

  model/roi_pooling/src　　#RoI pooling 层（SPP空间金字塔池化的简化版）

  model/roi_align/src　　# RoI align 层

  model/roi_crop/src　　# RoI crop 层

# 添加新的层

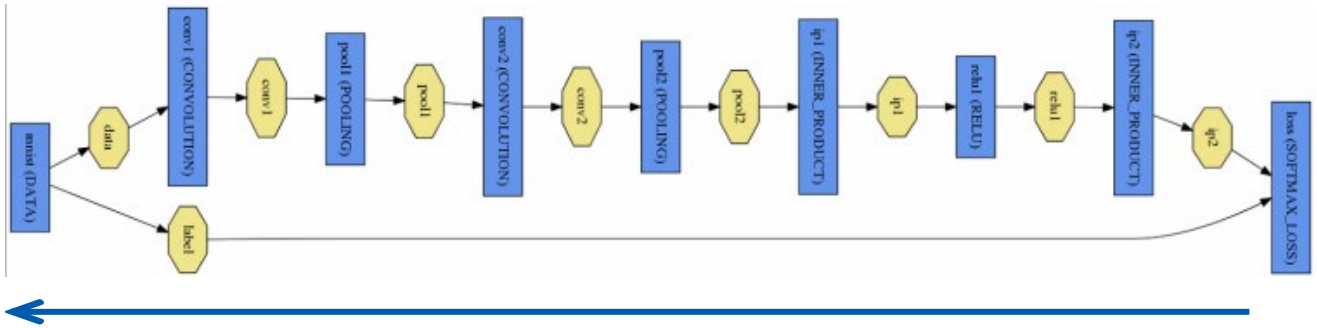
深蓝学院
shenlanxueyuan.com

- 用途：
  - 实现新的功能，但是pytorch不支持
  - 代码运行速度有要求
- 途径—C拓展：
  - 准备.c文件实现
  - 利用工具编译为python可以调用的模块
  - 嵌入到pytorch网络中
- 实例展示

# 代码主框架

设计网络(前向传播)

准备数据 →




设计损失函数（计算loss）

反向传播，更新参数

主函数：trainval_net.py

- 准备数据：Dataset+DataLoader
- 网络设计：vgg16+faster rcnn
- 损失计算： label准备，调用loss计算
- 参数更新：for循环迭代

# 数据准备



- 读取数据集中全部图像和gt boxes 信息到roidb, lib/roi_data_layer/roidb.py::combined_roidb(args.imdb_name)
- 利用roidb初始化dataset类，lib/roi_data_layer/roidb.py:: class roibatchLoader(data.Dataset)
- 用dataset初始化dataloader

dataset中getitem函数返回内容：data, im_info, gt_boxes, num_boxes，

其中利用lib/roi_data_layer/minibatch.py:: get_minibatch获取原始图像和gt boxes

注：

数据集图像的短边resize到600

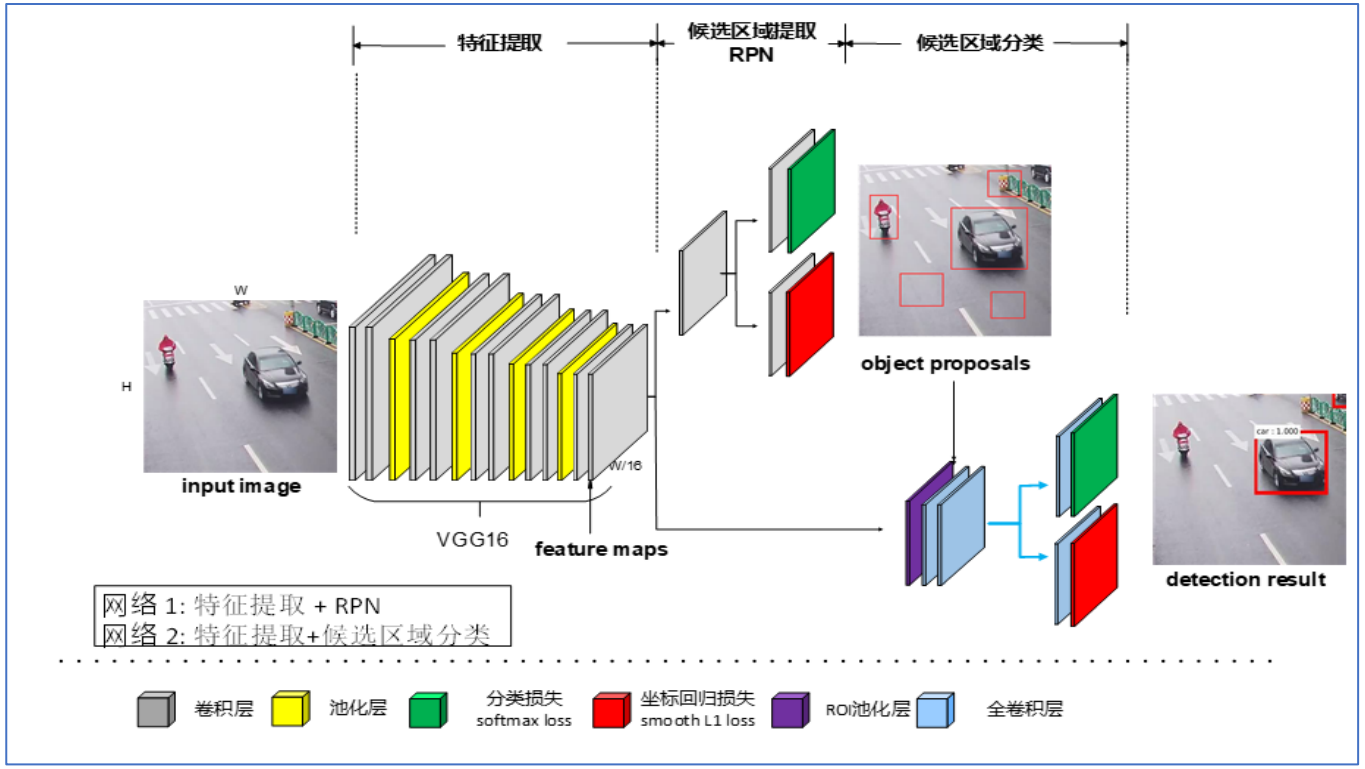
同一个batch中图像的长宽比保持一致

# 网络结构设计

深蓝学院 shenlanxueyuan.com

- lib/model/faster_rcnn
  - vgg16.py
  - resnet.py
  - 全部继承_fasterRCNN(lib/model/faster_rcnn/faster_rcnn.py)
- 主要的类成员:

```
self.RCNN_base #vgg网络部分
self.RCNN_rpn       #RPN部分
self.RCNN_cls_score # fast rcnn 分类
self.RCNN_bbox_pred #fast rcnn 坐标回归
```



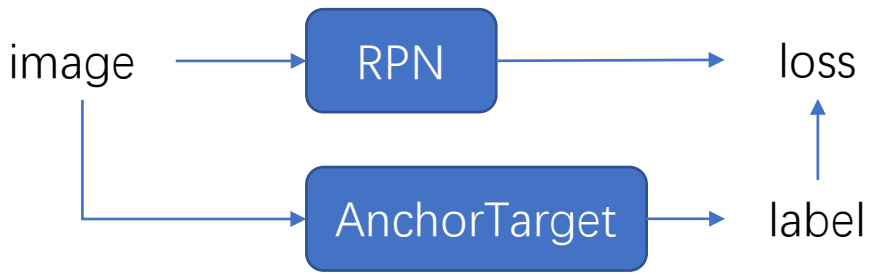
# 损失计算—生成各种label

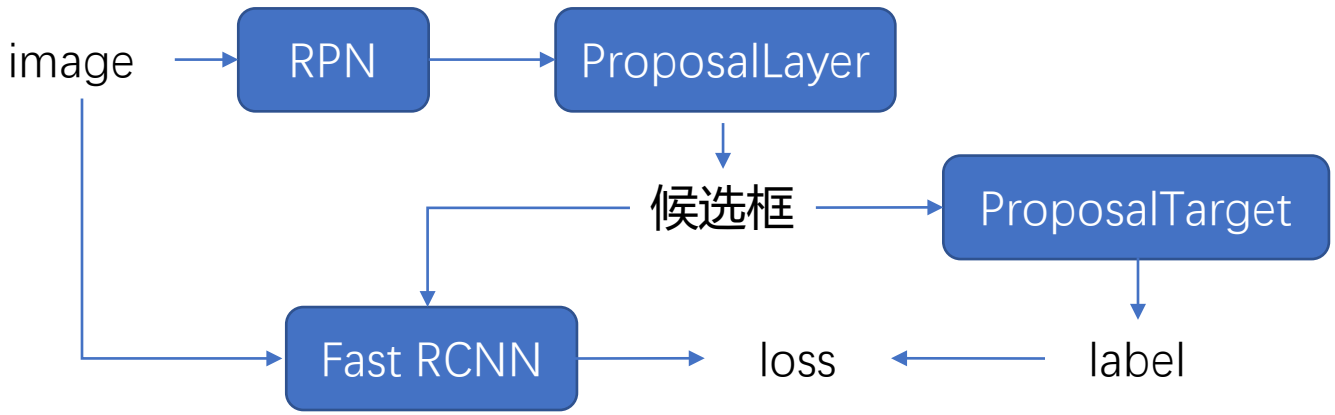
深蓝学院 shenlanxueyuan.com

❑ 三个layer

- AnchorTargetLayer： 负责在训练RPN的时候，从上万个anchor中选择一些(比如256)进行训练，以使得正负样本比例大概是1:1. 同时给出训练的位置参数目标。 即返回gt_rpn_loc和gt_rpn_label。
- ProposalLayer： 在RPN输出的候选区域，选取得分较高的N个候选框，然后进行NMS，然后选择一定数目（2000或者300)，用以候选区域分类训练或者测试。
- ProposalTargetLayer： 负责在训练候选区域分类的时候，从RoIs选择一部分(比如128个)用以训练。同时给定训练目标, 返回（sample_RoI, gt_RoI_loc, gt_RoI_label)

AnchorTargetCreator和ProposalTargetCreator是为了生成训练的目标，只在训练阶段用到，ProposalLayer是RPN为第二个分类网络生成RoIs，在训练和测试阶段都会用到。三个共同点在于他们都不需要考虑反向传播。



# Faster RCNN 训练细节



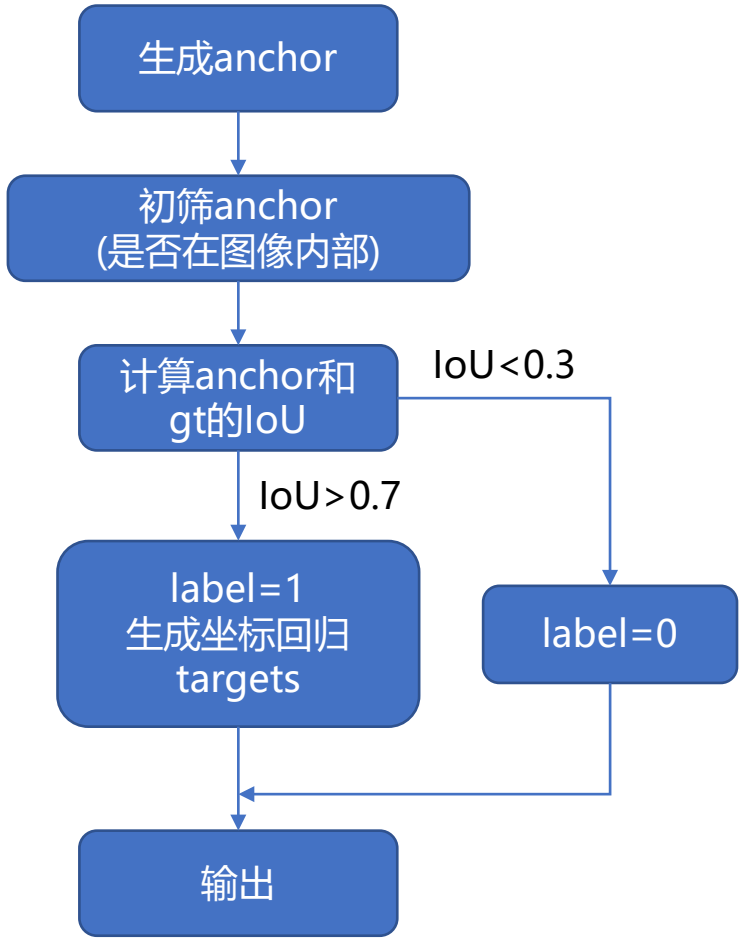
## ❑关于矩形框的描述：

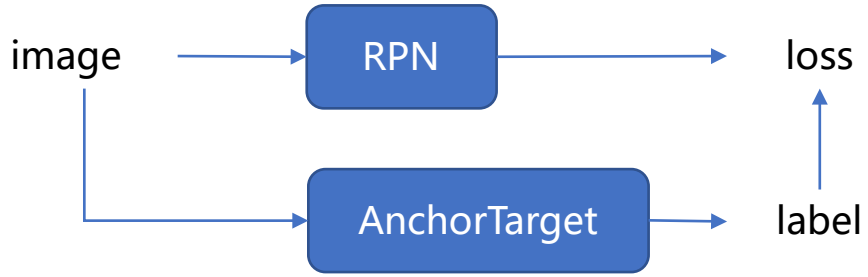
- Bbox: bounding box，边界框。其中Ground Truth Bounding Box是每一张图中人工标注的框的位置。一张图中有几个目标，就有几个框(一般小于10个框)。Faster R-CNN的预测结果也可以叫bounding box，不过一般叫 Predict Bounding Box.
- Anchor：人为在图像中假想的具有一定尺度、比例的框。一个feature map的锚的数目有上万个（比如 20000）。
- RoI：region of interest，候选矩形框。Faster R-CNN之前传统的做法是利用selective search从一张图上大概2000个候选框框。现在利用RPN进行候选框提取。
- loc： bbox，anchor和RoI，本质上都是一个框，可以用四个数（y_min, x_min, y_max, x_max）表示框的位置，还可以用（y，x，h，w）表示，即框的中心座标和长宽。Ground truth中记录的是前一种方式，在训练中进行位置回归的时候，用的是后一种的表示。

# AnchorTargetLayer



负责在训练RPN的时候，从上万个anchor中选择一些(比如256)进行训练，以使得正负样本比例大概是1:1. 同时给出训练的位置参数目标。即返回gt_rpn_loc和gt_rpn_label。



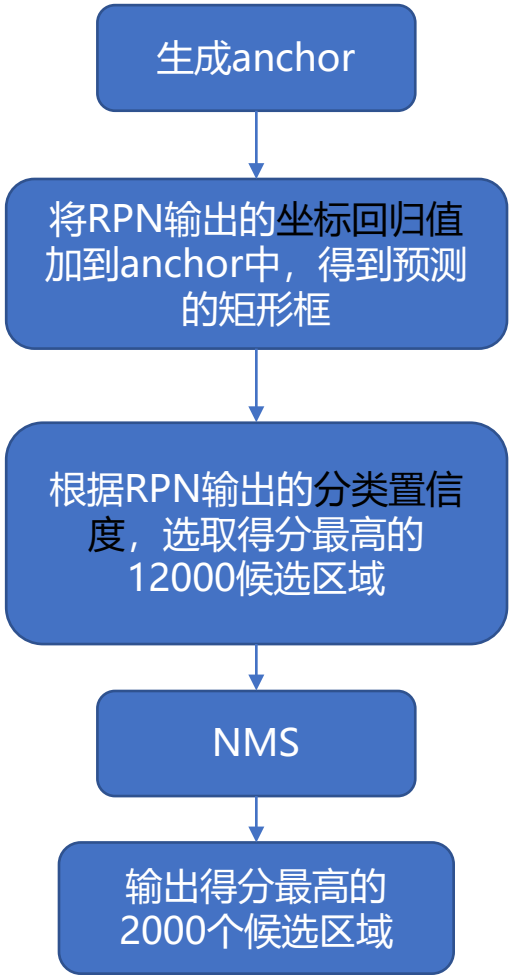



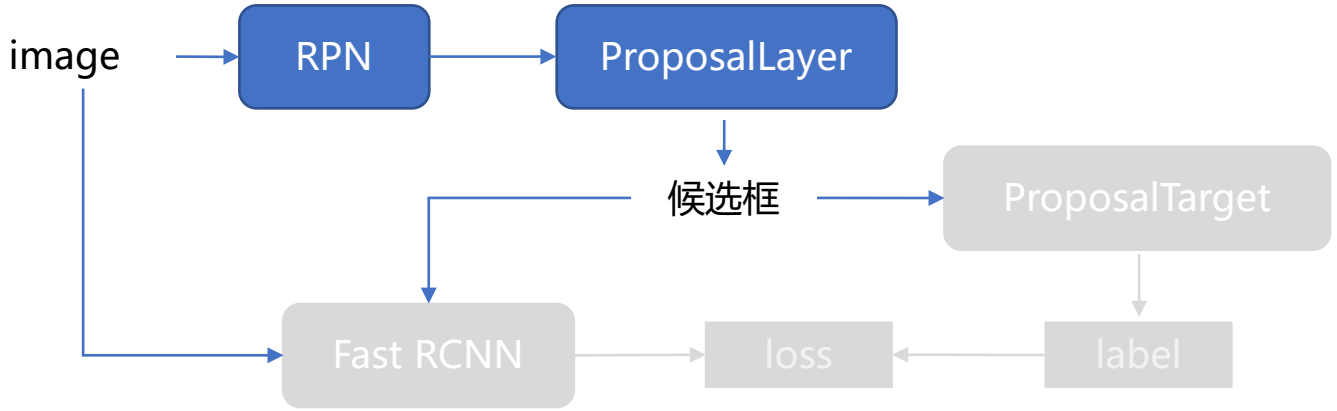

并非所有符合条件的anchor都参与计算，每张图像选取随机选取256个anchor参与计算，其中正负样本比例为1:1。

# ProposalLayer



在RPN输出的候选区域，选取得分较高的N个候选框，然后进行NMS，然后选择一定数目（2000或者300），用以候选区域分类训练或者测试

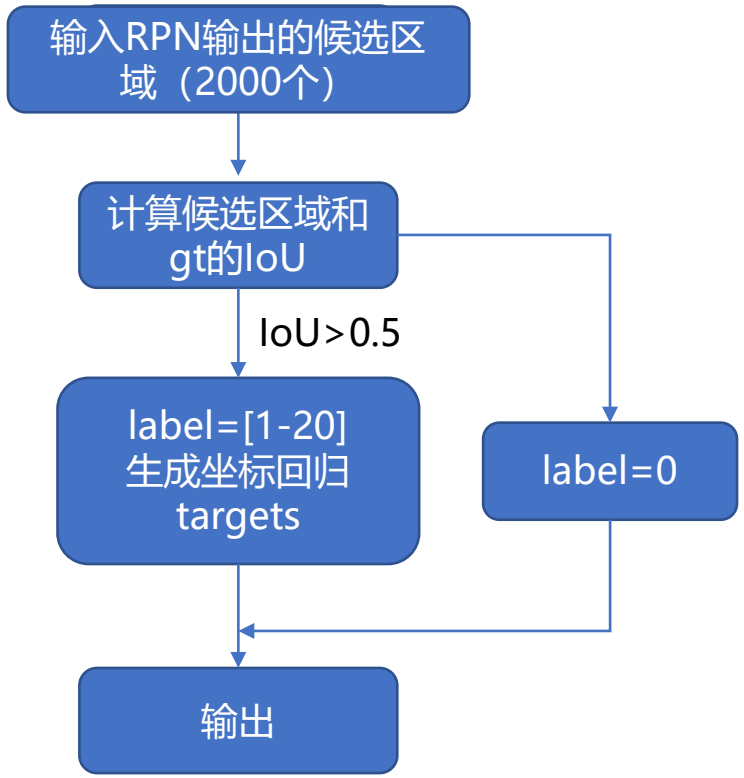


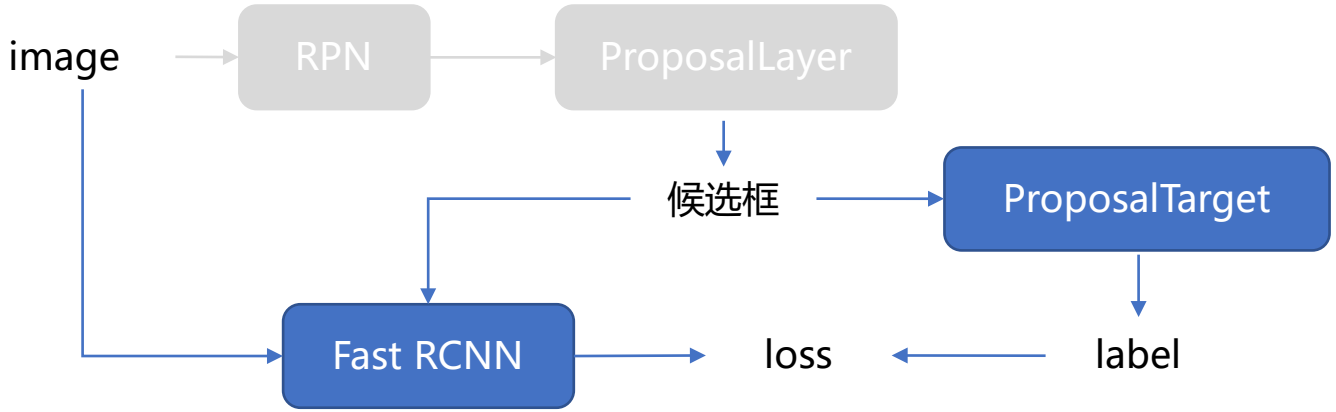



# ProposalTargetLayer



负责在训练候选区域分类的时候，从RoIs选择一部分(比如128个)用以训练。同时给定训练目标, 返回（sample_RoI, gt_RoI_loc, gt_RoI_label）




并非所有符合条件的候选区域都参与计算。每张图像，选取随机选取128个参与计算，其中正负样本比例为1:3。

# Faster RCNN 训练细节



## ❑四个损失

- RPN 分类损失：anchor是否为前景（二分类）
- RPN位置回归损失：anchor位置微调
- RoI 分类损失：RoI所属类别（21分类，多了一个类作为背景）
- RoI位置回归损失：继续对RoI位置微调

  四个损失相加作为最后的损失，反向传播，更新参数。

# 作业

深蓝学院
shenlanxueyuan.com

- 对比可视化FasterRCNN两个阶段
  - RPN阶段得分最高的20个框（即proposal layer的输出top-20）
  - 该top-20精细化分类后的结果

# 感谢各位聆听

## Thanks for Listening