法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。



关注 小象学院

第10讲　目标检测（下）

计算机视觉

主讲人：叶梓

上海交通大学博士
主要研究方向：机器学习、深度学习、人工智能

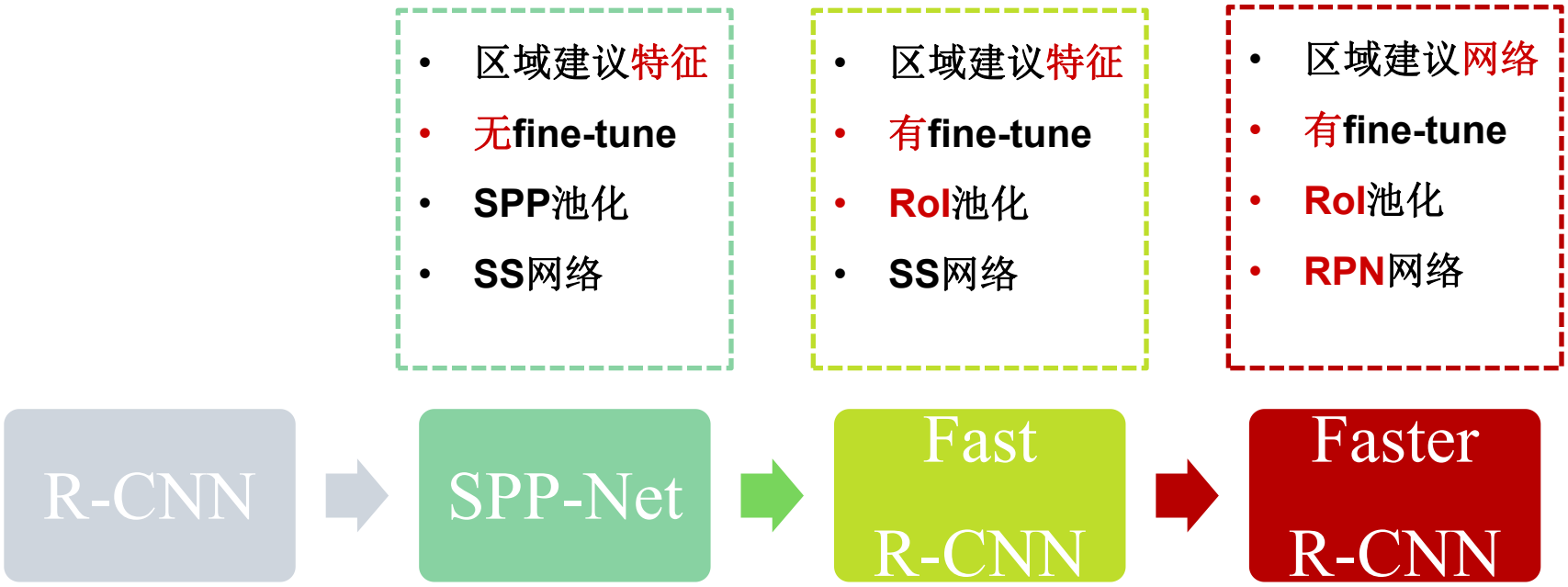
本章结构

- 区域卷积神经网络（R-CNN）系列
  - R-FCN
- YOLO系列
  - YOLO v1
  - YOLO v2/9000
  - YOLO v3
- 人脸检测与行人检测的数据集
- 应用案例：
  - 基于Darknet的YOLO的实现

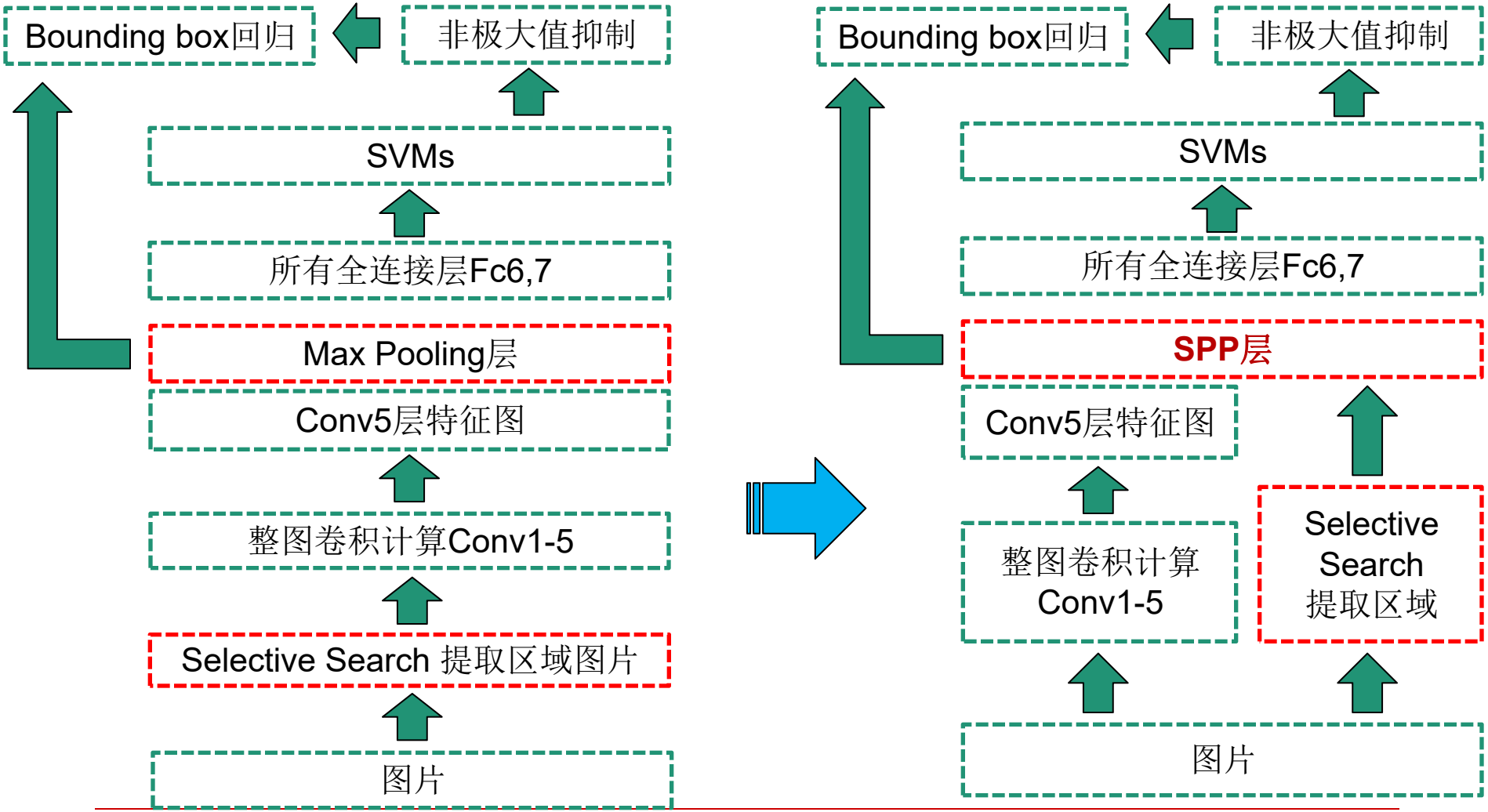
R-CNN回顾

模型进化

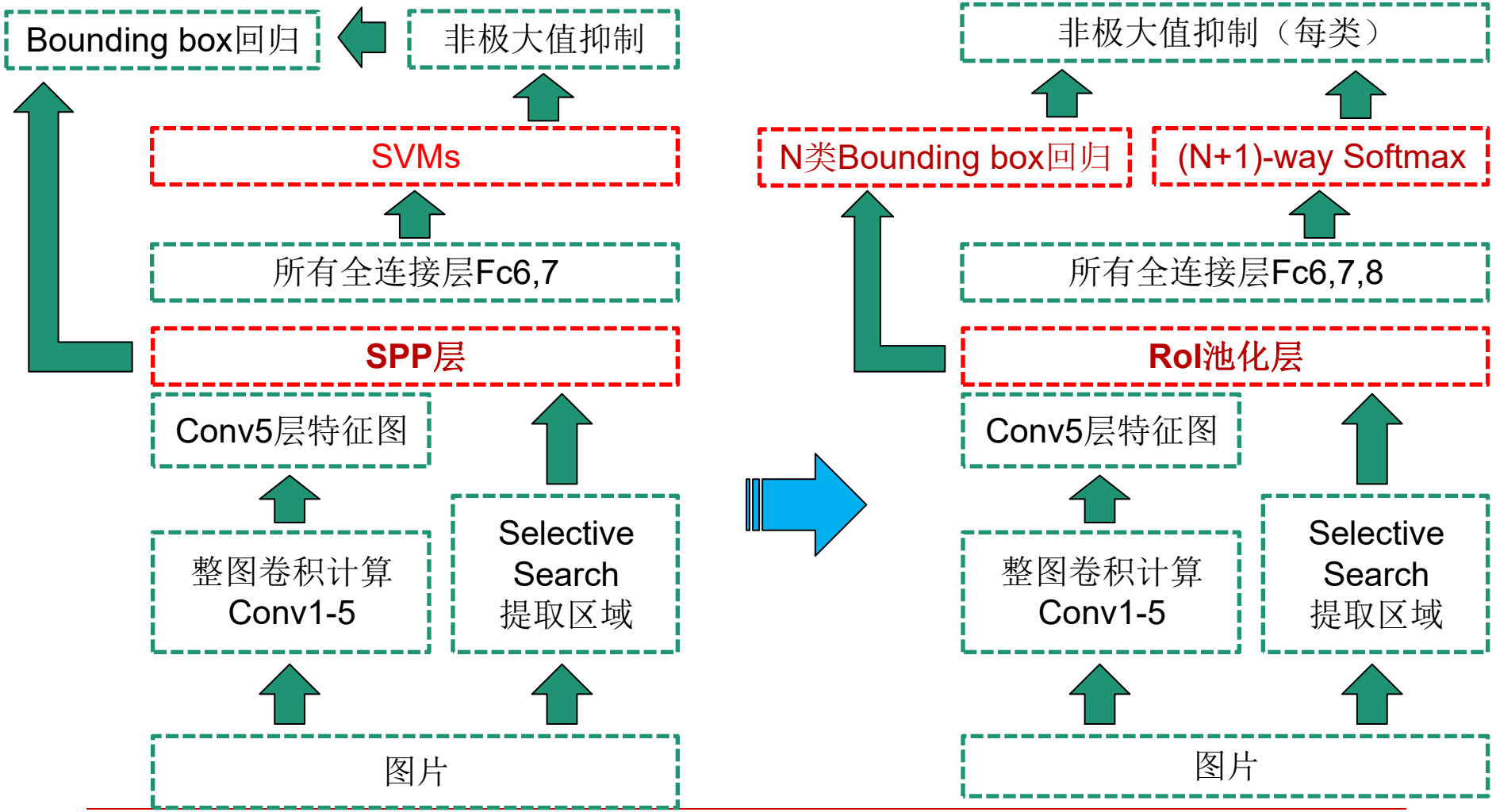
- 共享卷积层计算



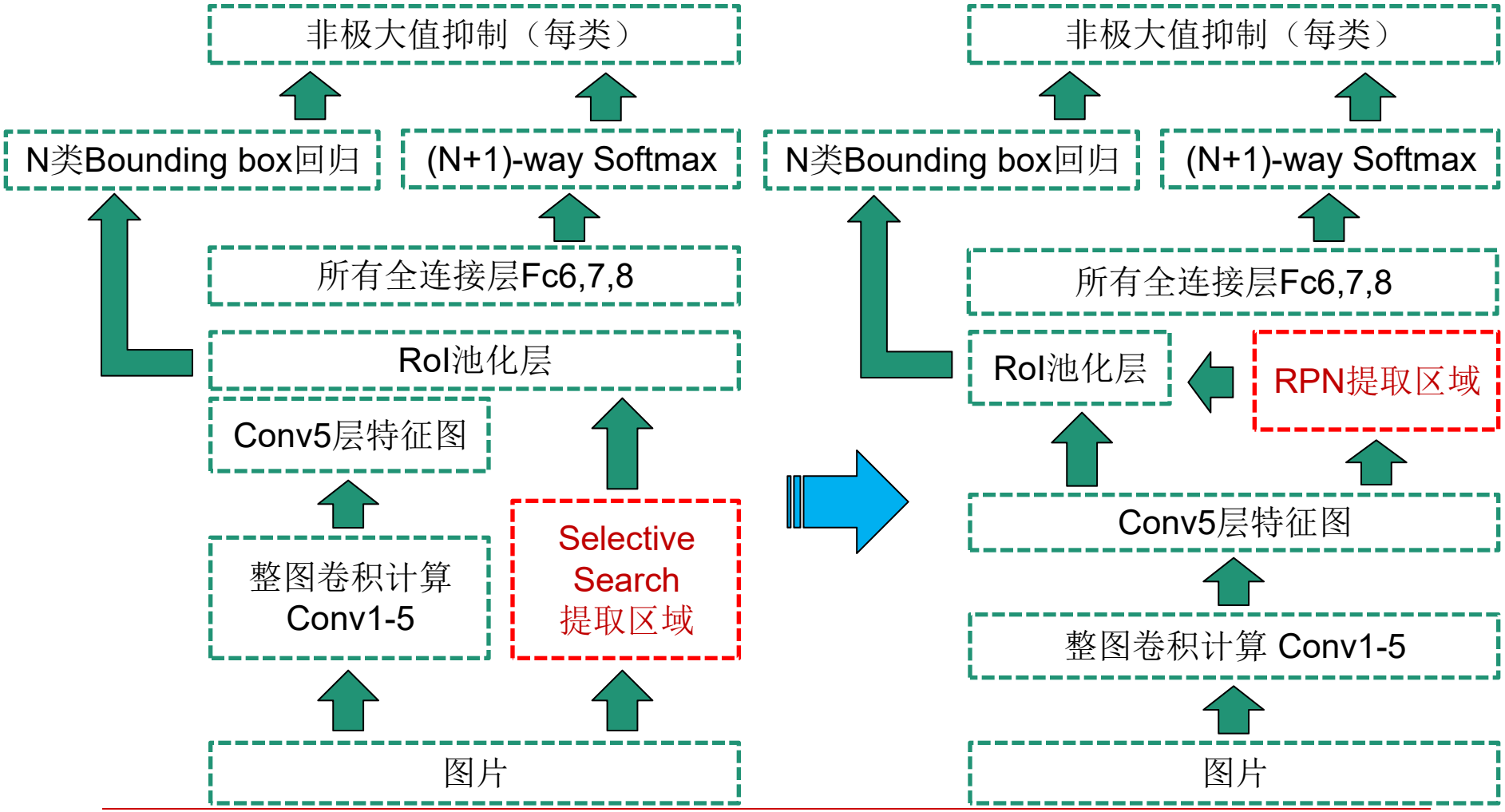
R-CNN vs SPP-Net



SPP-Net vs Fast R-CNN



Fast R-CNN vs Faster R-CNN



R-FCN

R-CNN系列的结构

- 基于旧形态CNN的结构（AlexNet，VGG）
  - 全卷积子网络
  - 全连接子网络
- 相对应的结构设计
  - 全卷积子网络（5层/组）
    - 独立于RoI
    - 计算共享
  - RoI-wise子网络（3层）
    - 计算无法共享

R-FCN

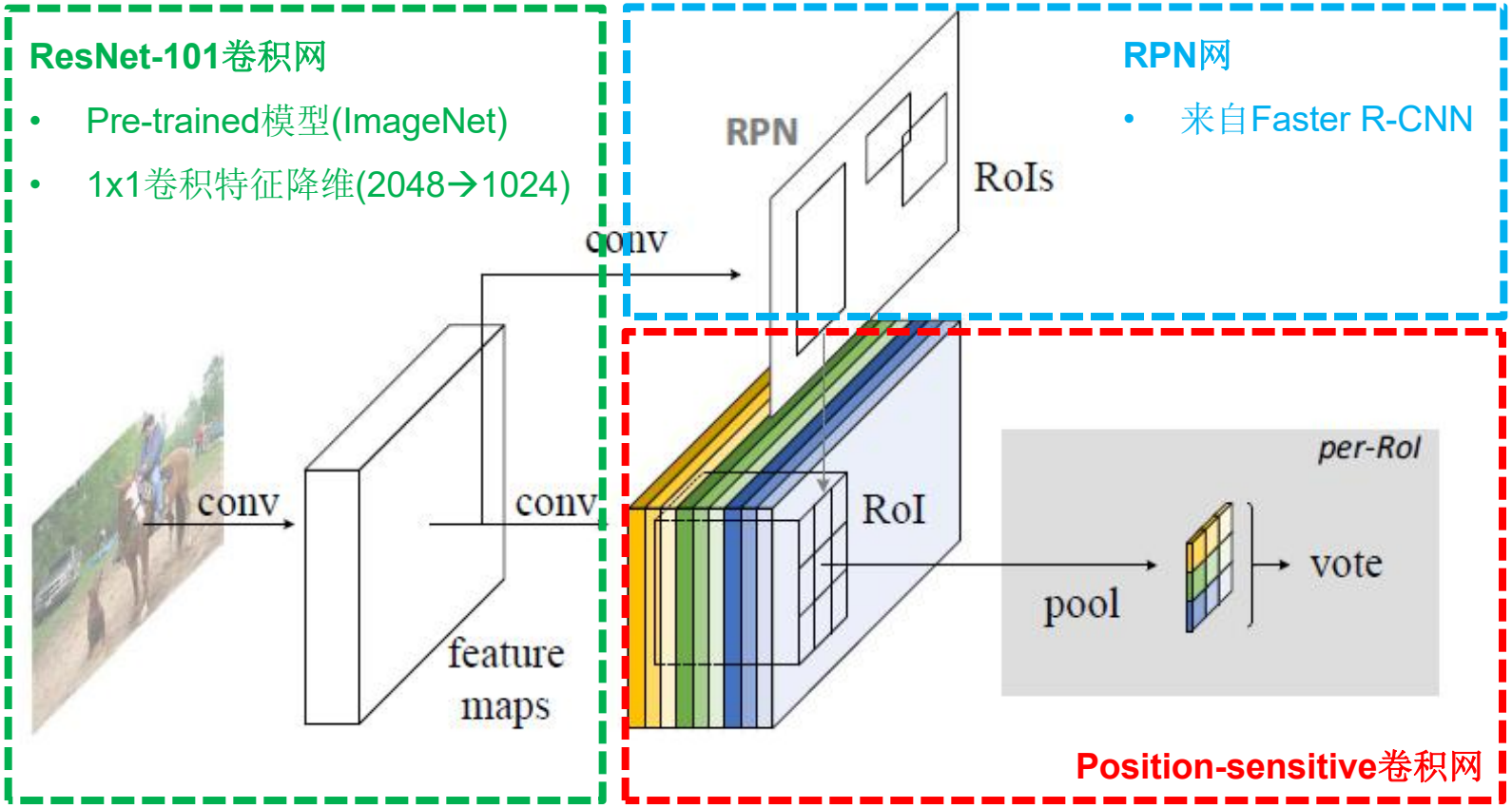
- CNN的全卷积化趋势（ ResNet ， GoogLeNet）
  - 只剩1个全连接层（2048→1000）
- 相应的，基于旧结构设计的R-CNN会出现问题
  - 结构：RoI-wise子网络相当于隐含层
  - 性能：检测性能跟分类性能不一致
  - 应用两难：
    - 检测网络的变换敏感性（Translation variance）
    - 分类网络的变换不变性（Translation invariance）
      - 卷积层越深，不变性越强，对变换越不敏感
  - 不适应设计：ResNet-101→ Conv91 + RoI池化 + Conv10
    - 准确率提升，但速度下降

R-FCN

- 适应全卷积化CNN结构，提出全卷积化设计
  - 共享ResNet的所有卷积层
  - 引入变换敏感性（Translation variance）
    1. 位置敏感分值图（Position-sensitive score maps）
       - 特殊设计的卷积层
       - Grid位置信息 + 类别分值
    2. 位置敏感池化（Position-sensitive RoI pooling）
       - 无训练参数
       - 无全连接网络的类别推断

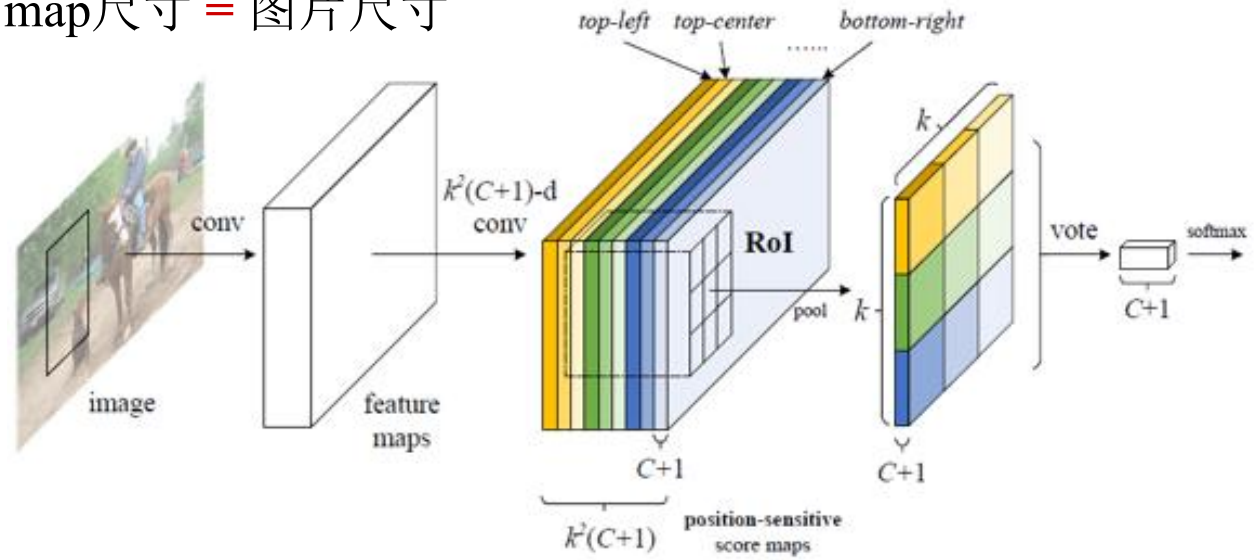
R-FCN

R-FCN结构


ResNet-101卷积网
• Pre-trained模型(ImageNet)
• 1x1卷积特征降维(2048→1024)
RPN网
• 来自Faster R-CNN
RPN
RoIs
conv
conv
conv
feature
maps
RoI
per-RoI
pool
vote
Position-sensitive卷积网

R-FCN的位置敏感卷积层

- 使用$k^2(C+1)$个通道对（位置，类别）组合进行编码
  - 类别：C个物体类+1个背景类
  - 相对位置：k x k个Grid（k=3）
  - 位置敏感分值图（Position-sensitive score maps）
    - 每个分类k^2个score map
    - score map尺寸 = 图片尺寸



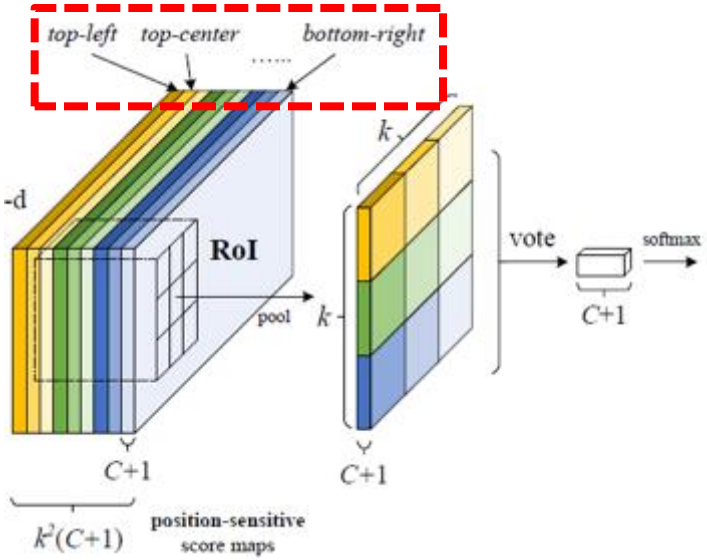
R-FCN的位置敏感RoI池化层

- 显式地编码相对位置信息
  - 将 $w \times h$ 尺寸的RoI拆分成k x k个 $\frac{w}{k} \times \frac{h}{k}$ 尺寸的bin
  - 不同(颜色)bin对应不同(颜色)通道层（score map）
  - Bin内做均值池化
  - 输出尺寸 k x k x (C+1)

$$r_c(i,j \mid \Theta) = \sum_{\substack{(x,y)\in \text{bin}(i,j) \\ (0 \le i,j \le k-1)}} z_{i,j,c}(x + x_0, y + y_0 \mid \Theta)/n$$

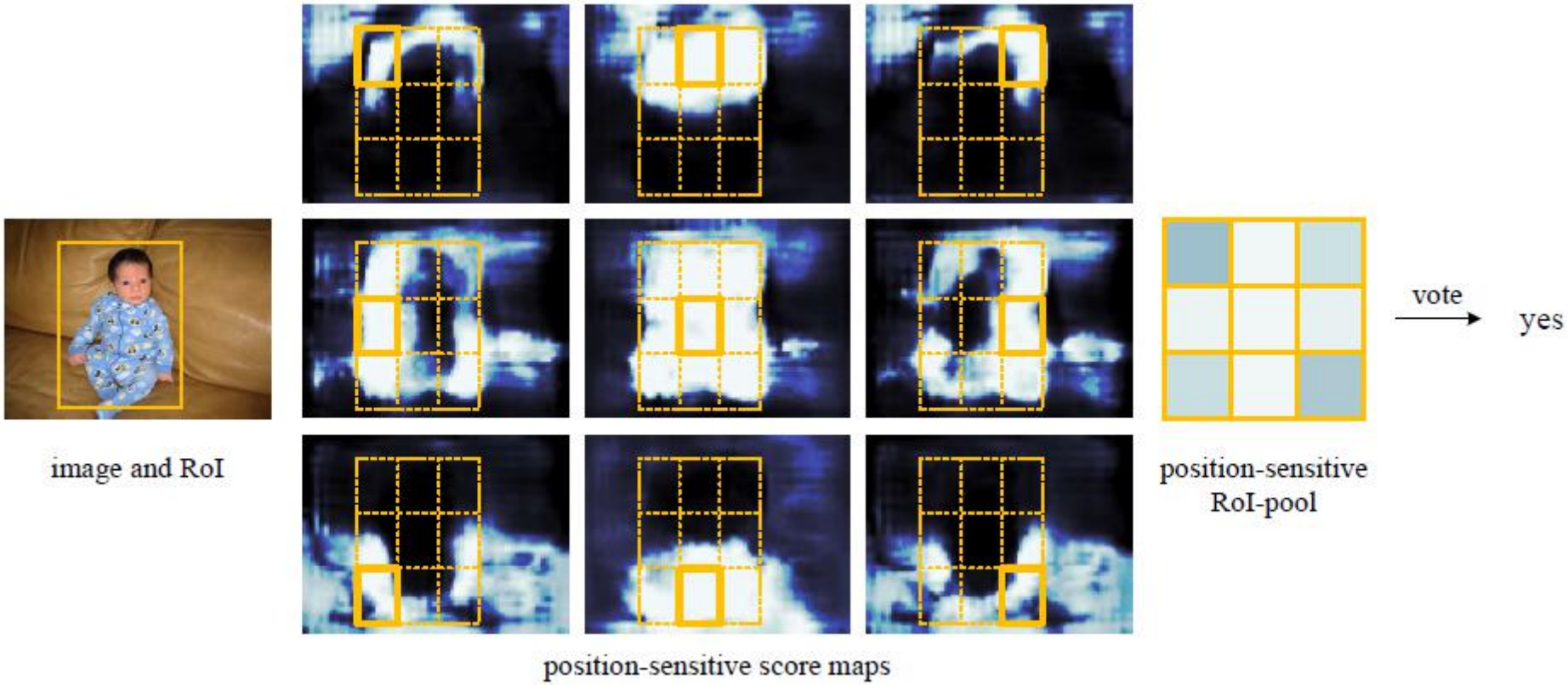
(x_0, y_0) 为RoI左上角坐标

$$\lfloor i\tfrac{w}{k} \rfloor \le x < \lceil (i+1)\tfrac{w}{k} \rceil \text{ and } \lfloor j\tfrac{h}{k} \rfloor \le y < \lceil (j+1)\tfrac{h}{k} \rceil$$



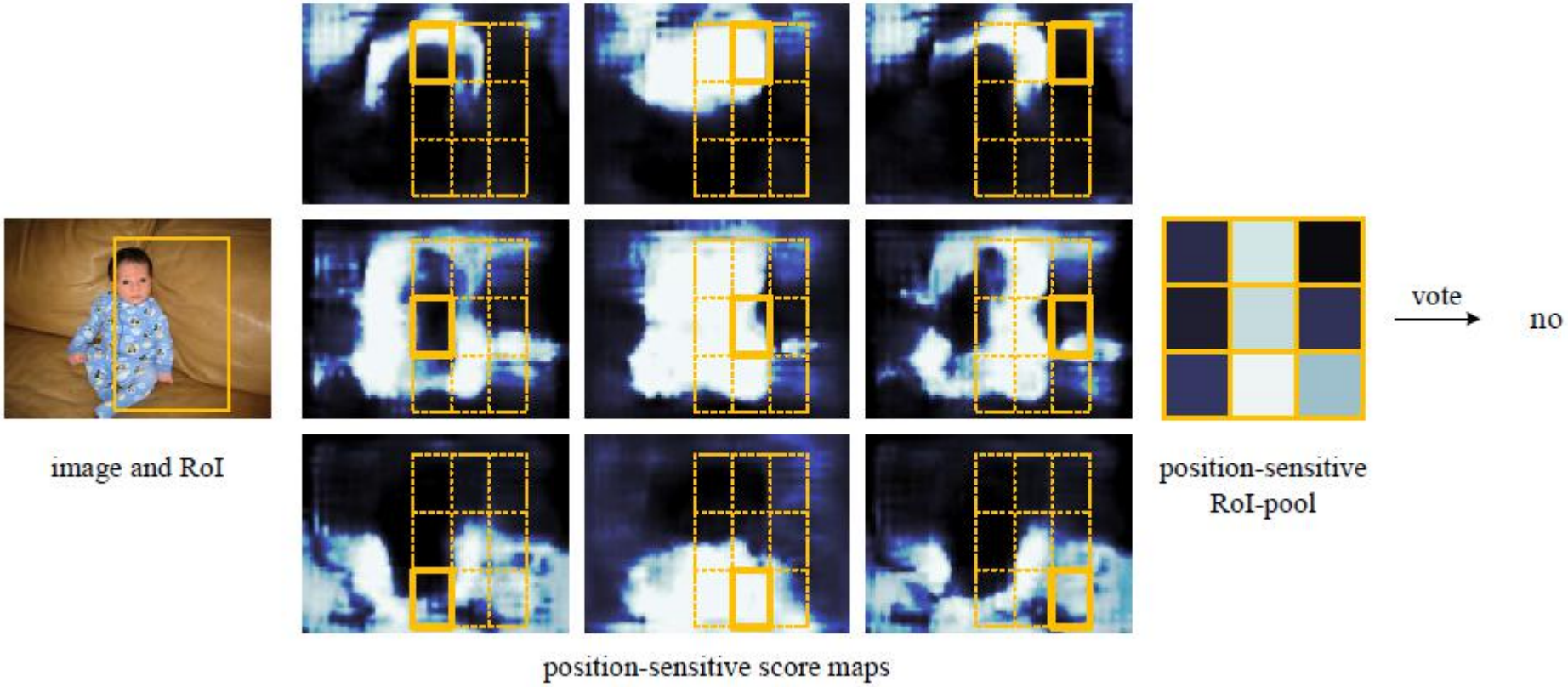
R-FCN

R-FCN的Score map可视化（person类别）



R-FCN

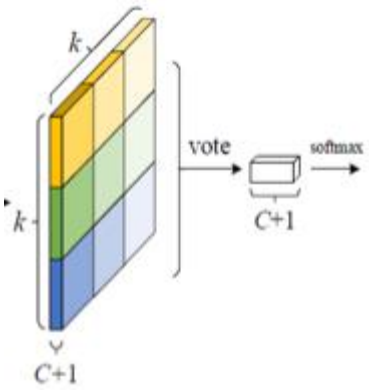
R-FCN的Score map可视化（person类别）



R-FCN的多任务损失函数

ROI 区域的真实标记

- 总损失： $L(s, t_{x,y,w,h}) = L_{cls}(s_{c^*}) + \lambda[c^* > 0]L_{reg}(t, t^*)$
- 分类损失函数 $L_{cls}(s_{c^*}) = -\log(s_{c^*})$
  - 对池化输出计算k^2区域上的均值投票 $r_c(\Theta) = \sum_{i,j} r_c(i, j \mid \Theta)$
  - Softmax归一化 $s_c(\Theta) = e^{r_c(\Theta)} / \sum_{c'=0}^{C} e^{r_{c'}(\Theta)}$
- Bounding box回归损失函数
  - $4k^2$通道的位置敏感卷积层 $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$
  - 使用位置敏感RoI池化获取4 x k x k输出
  - 在k x k区域上均值投票 $t = (t_x, t_y, t_w, t_h)$
    - x 对应预测框
    - x_a 对应anchor框
    - x^* 对应ground truth框
    - y, w, h同理
  - 每个分类一个回归模型



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

R-FCN的训练

- OHEM（Online Hard Example Mining）
  - 首先对RPN获得的候选ROI（正负样本分别进行排序）进行排序操作；
  - 然后在含有正样本（目标）的ROI中选择前N个ROI，将正负样本的比例维持在1:3的范围内，保证每次抽取的样本中都会含有一定的正样本
  - 这样训练可以提高网络的分类能力。
- Faster R-CNN的4步训练法
  - 2轮：RPN跟R-FCN交替训练

R-FCN

R-FCN位置敏感的性能

- 基于ResNet-101
- 位置敏感性带来大幅提升

method	RoI output size ($k \times k$)	mAP on VOC 07 (%)
naïve Faster R-CNN	1×1	61.7
	7×7	68.9
R-FCN (w/o position-sensitivity)	1×1	*fail*
R-FCN	3×3	75.5
	7×7	**76.6**

R-FCN

性能对比R-FCN vs Faster R-CNN

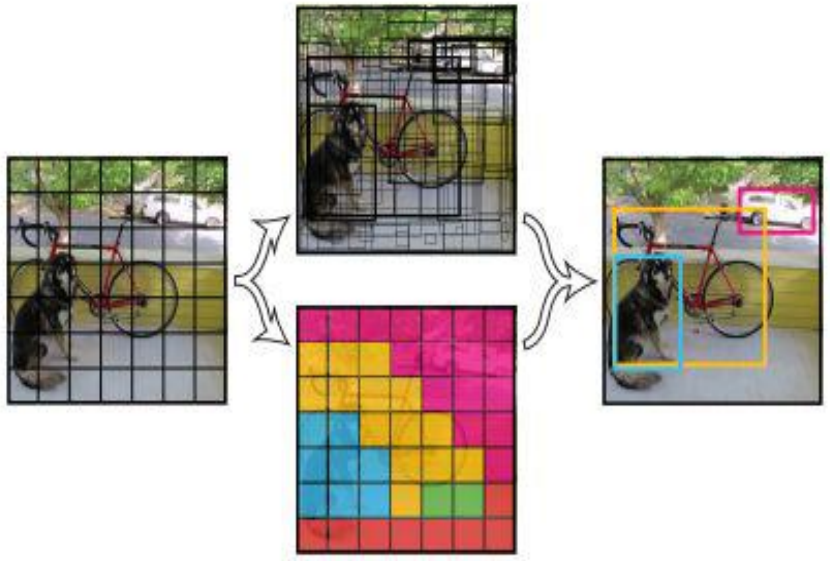
- 基于ResNet-101
- OHEM没有带来时间开销
- k x k = 7 x 7
- Test时间提升2.5x

	depth of per-RoI subnetwork	training w/ OHEM?	train time (sec/img)	test time (sec/img)	mAP (%) on VOC07
Faster R-CNN	10		1.2	0.42	76.4
R-FCN	0		0.45	0.17	76.6
Faster R-CNN	10	✓ (300 RoIs)	1.5	0.42	79.3
R-FCN	0	✓ (300 RoIs)	0.45	0.17	**79.5**
Faster R-CNN	10	✓ (2000 RoIs)	2.9	0.42	*N/A*
R-FCN	0	✓ (2000 RoIs)	0.46	0.17	79.3

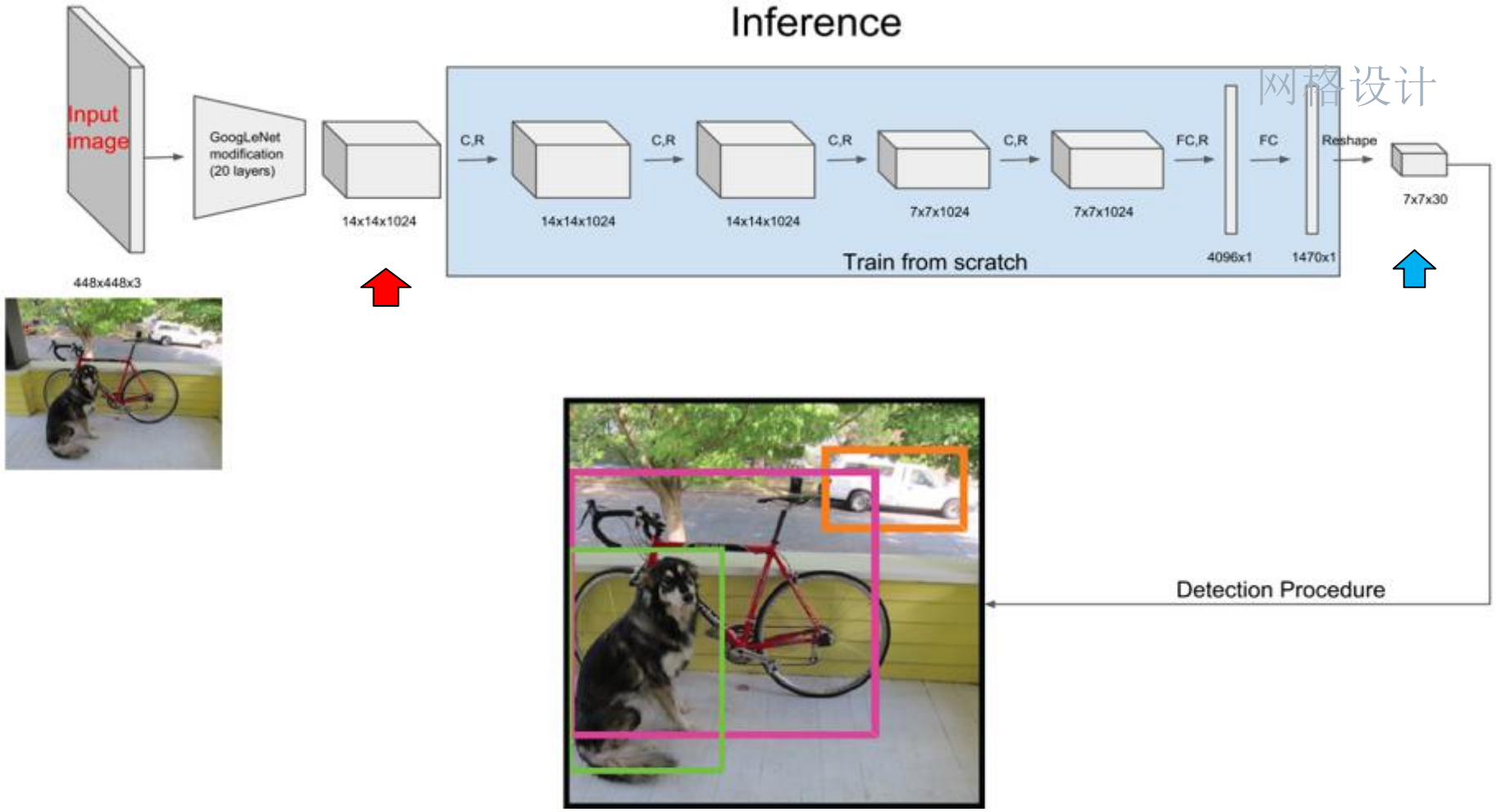
YOLO v1

- YOLO将物体检测任务当做一个regression问题来处理，通过YOLO，每张图像只需要"看一眼"就能得出图像中都有哪些物体和这些物体的位置。

- 将图像resize到448*448作为神经网络的输入，
- 使用一个神经网络，直接从一整张图像来预测出bbox的坐标、box中包含物体的置信度和物体的可能性，
- 然后进行非极大值抑制，筛选Boxes。



YOLO v1


Inference
Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Train from scratch
Detection Procedure

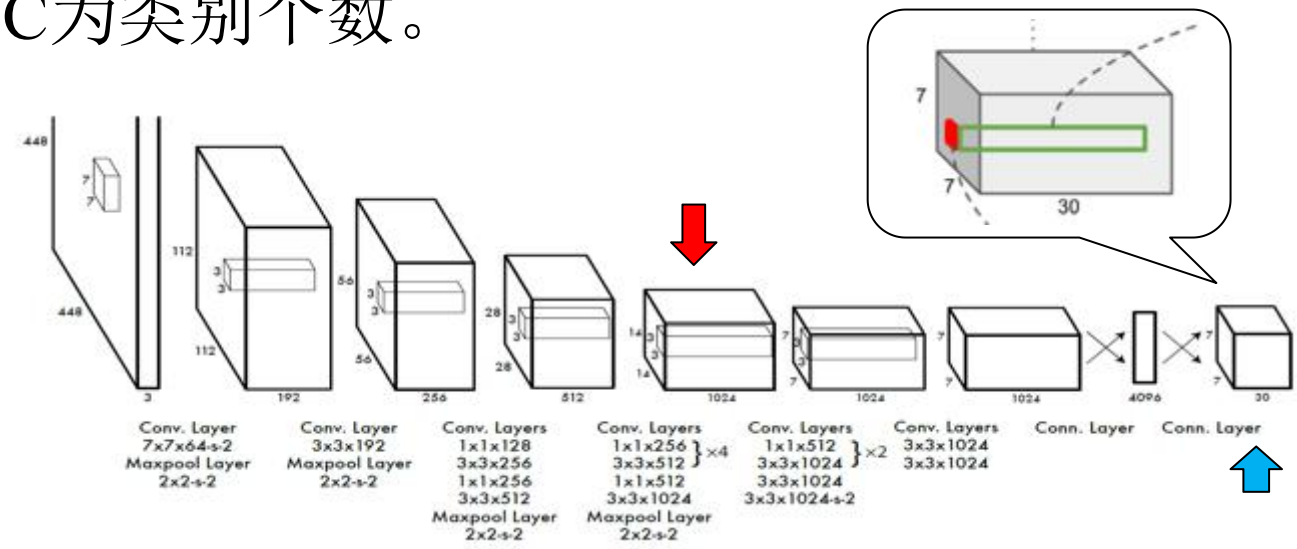
网格设计

YOLO v1

- 首先利用ImageNet 1000-class的分类任务数据集Pretrain卷积层。使用上述网络中的前20 个卷积层，加上一个 average-pooling layer，最后加一个全连接层，作为 Pretrain 的网络。
- 将Pretrain的结果的前20层卷积层应用到Detection中，并加入剩下的4个卷积层及2个全连接。
- 同时为了获取更精细化的结果，将输入图像的分辨率由 224* 224 提升到 448* 448。
- 将所有的预测结果都归一化到 0~1, 使用 Leaky RELU 作为激活函数。
- 为了防止过拟合，在第一个全连接层后面接了一个 ratio=0.5 的 Dropout 层。

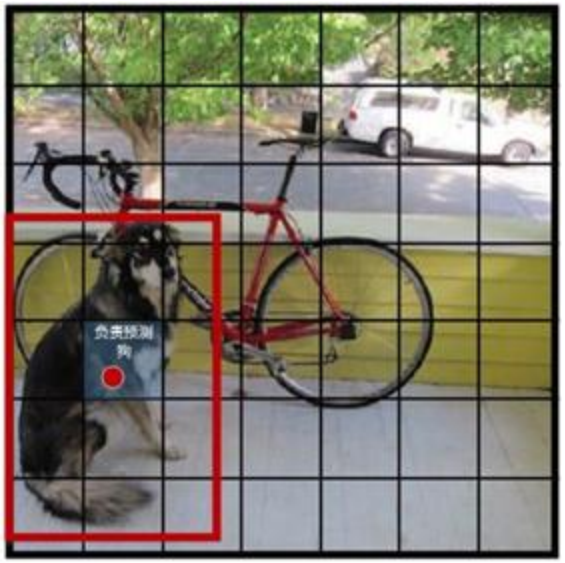
YOLO v1

- YOLO网络结构由24个卷积层与2个全连接层构成，网络入口为448x448，图片进入网络先经过resize，网络的输出结果为一个张量。
  - 输出的维度为：S* S* (B* 5+C)，
  - 在 Pascal VOC 上，预测的结果维度是 7*7*(2*5+20)。
- 其中，S为划分网格数，B为每个网格负责目标个数，C为类别个数。



YOLO v1

- 将一幅图像分成SxS个网格(grid cell)，如图所示。右图中物体"狗"的中心点（红色原点）落入第5行、第2列的格子内，所以这个格子负责预测图像中的狗。
- 每一个栅格预测B（B=2）个bboxes，以及这些bboxes的confidence scores。



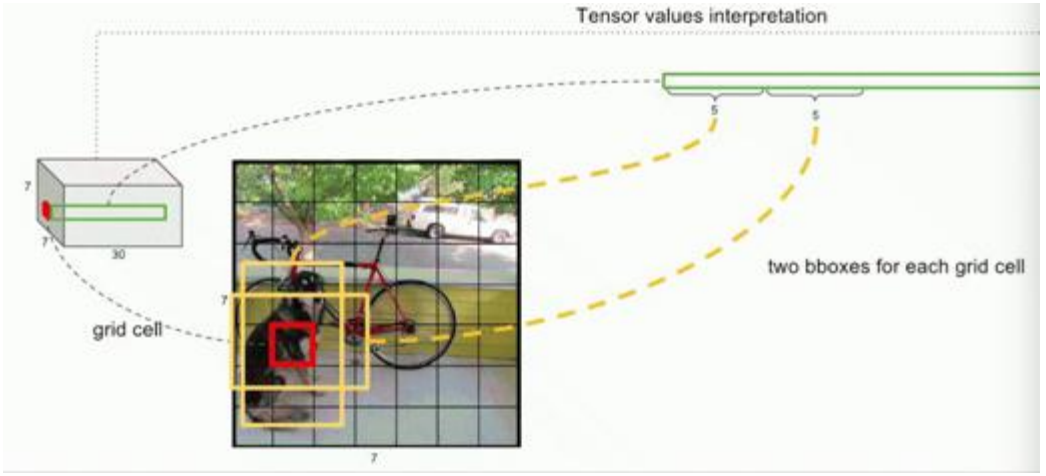
- confidence scores反映了模型对于这个栅格的预测：该栅格是否含有物体，以及这个box的坐标预测的有多准。
- 公式定义如下：

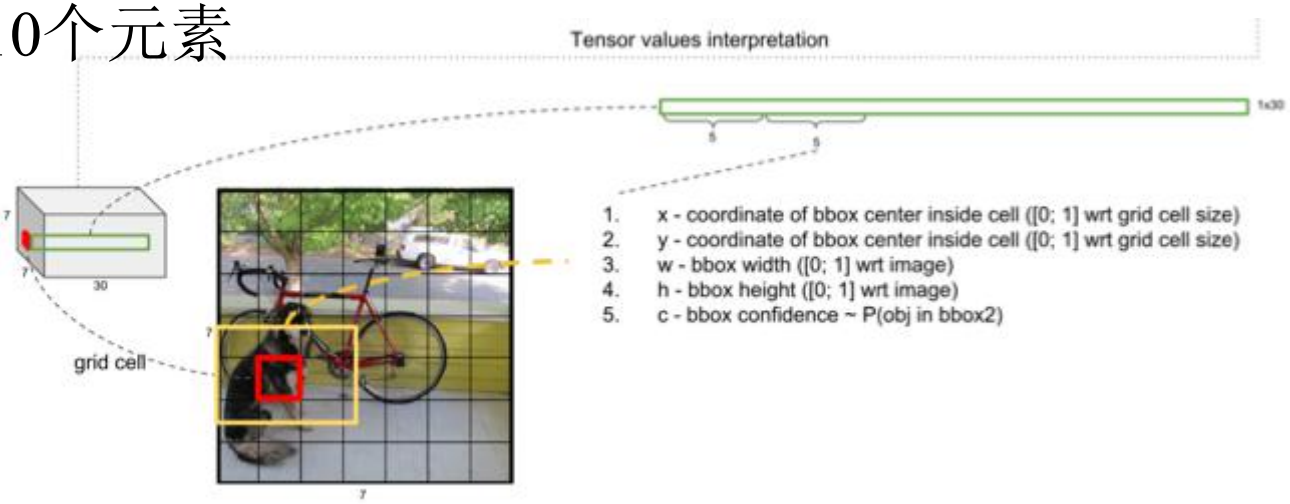
$$confidence = Pr(Object) * IOU_{pred}^{truth}$$

- 如果这个栅格中不存在object，则confidence score应该为0；否则的话，confidence score则为 predicted bounding box与 ground truth box之间的 IOU（intersection over union）。

YOLO v1

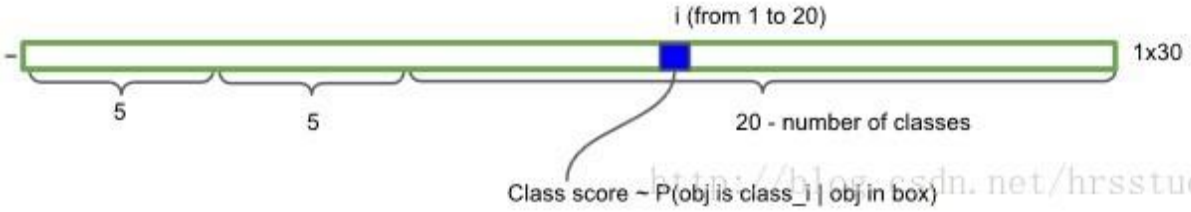
- YOLO v1：每个栅格两个bbox。
- 对每个bbox有5个预测值：x, y, w, h, confidence。
- 作为预测的结果每一行中前10个元素。





YOLO v1

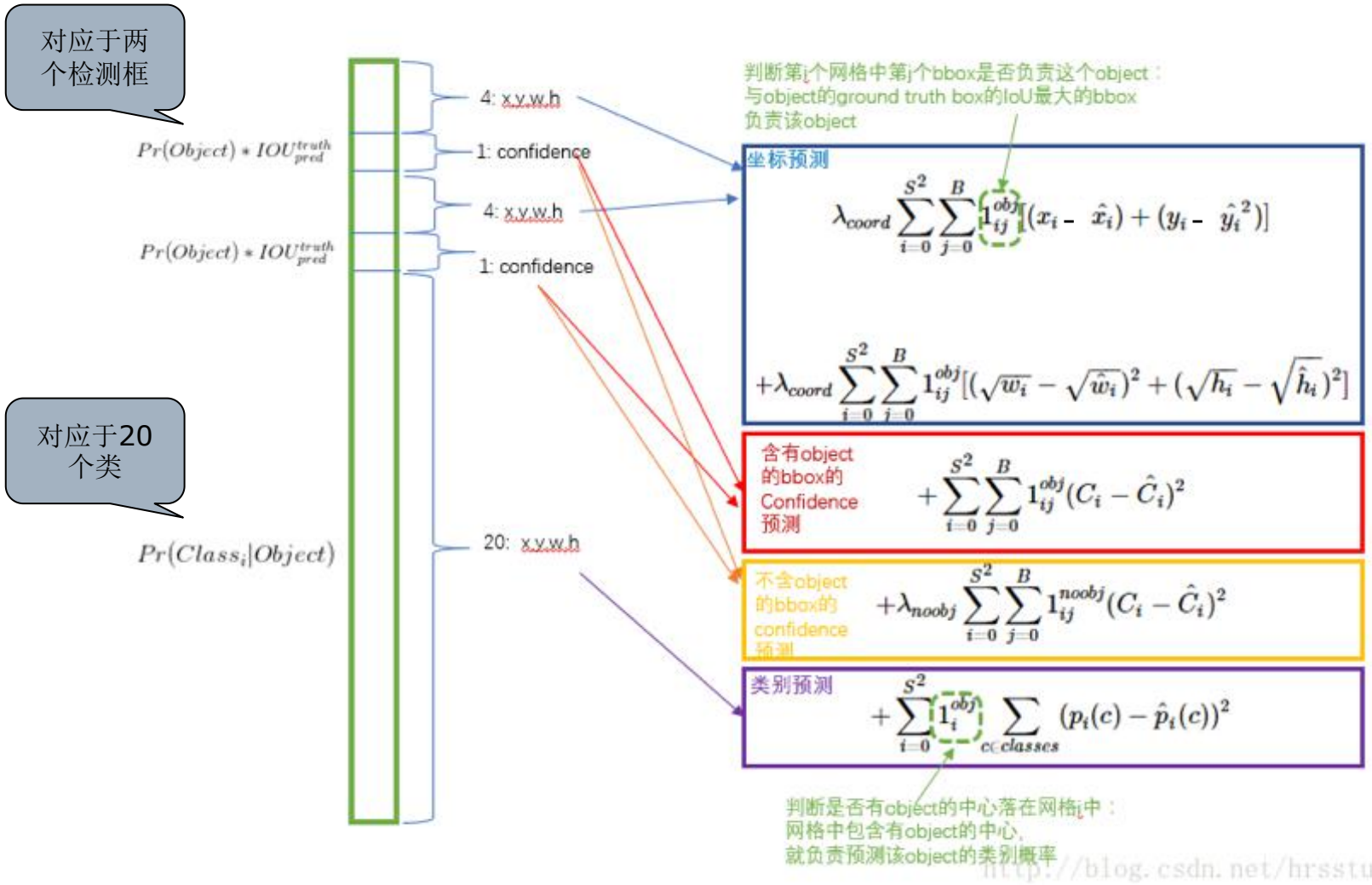
- 每一个栅格还要预测C条件类别概率(conditional class probability)：$Pr(Class_i|Object)$。即在一个栅格包含一个Object的前提下，它属于某个类的概率



  - 条件类别概率：是针对每个栅格的。
  - Confidence：是针对每个bbox的。
- 在测试阶段，将每个栅格的条件类别概率与每个 bbox的confidence相乘：

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

YOLO v1

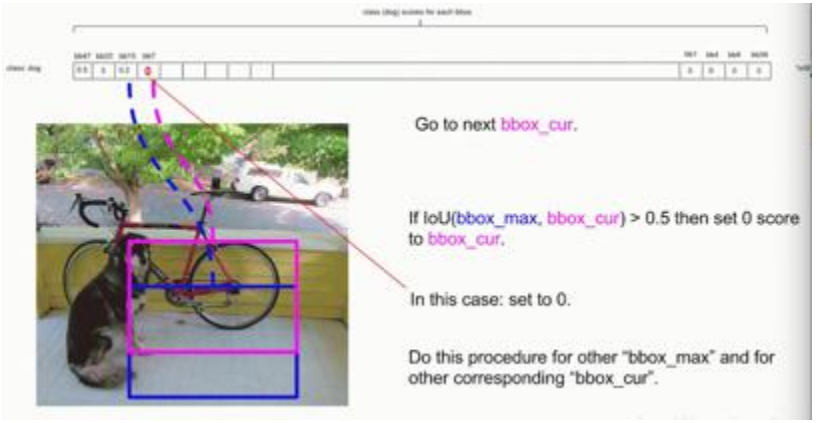


http://blog.csdn.net/hrsstudy

YOLO v1

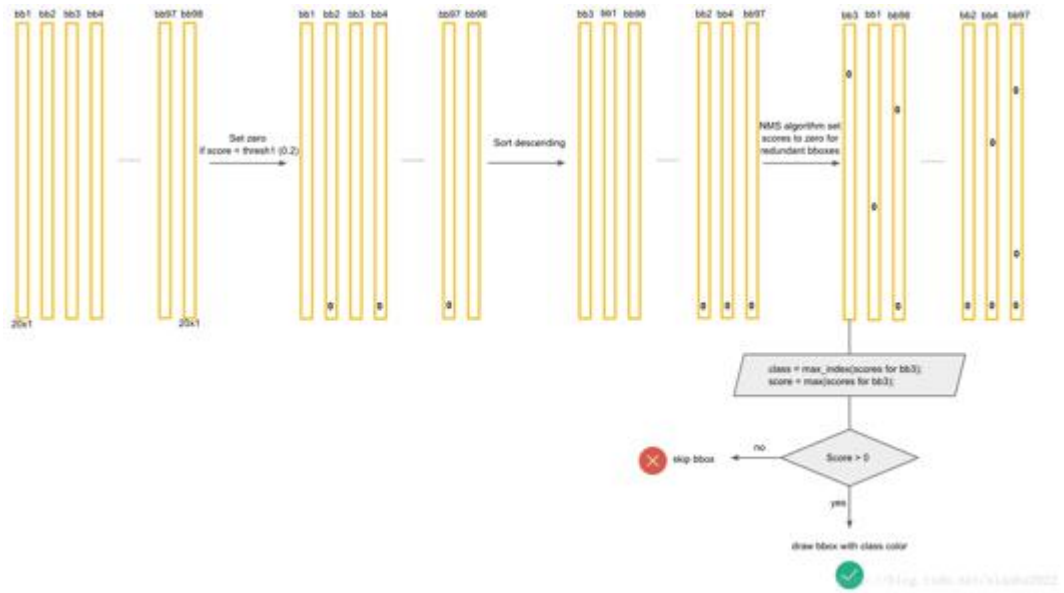
- NMS（非极大值抑制）：
  - 首先从所有的检测框中找到置信度最大的那个框，
  - 然后挨个计算其与剩余框的IOU，如果其值大于一定阈值（重合度过高），那么就将该框剔除；
  - 之后对剩余的检测框重复上述过程，直到处理完所有的检测框。





YOLO v1

- Yolo的实际策略是先使用NMS，然后再确定各个box的类别，其基本过程如图所示：
- 对于98个boxes，首先将小于置信度阈值的值归0，然后分类别地对置信度值采用NMS。
- 这里NMS处理结果不是剔除，而是将其置信度值归为0。
- 最后才是确定各个box的类别，当其置信度值不为0时才做出检测结果输出。



YOLO v1

- YOLO v1的优点：
  - 检测物体的速度很快；
  - 假阳性率低；
  - 能够学到更加抽象的物体的特征。
- YOLO v1的不足：
  - YOLO的物体检测精度低于其他state-of-the-art的物体检测系统。
  - YOLO容易产生物体的定位错误。
  - YOLO对小物体的检测效果不好（尤其是密集的小物体，因为一个栅格只能预测2个物体）。

YOLO v2

Batch Normalization

- Batch Normalization可以提高模型收敛速度，减少过拟合。在所有卷积层都应用了BN，使结果提升了2%。

High Resolution Classifier

- 基于ImageNet数据集预训练模型，大部分类器输入图像尺寸小于256×256。
- 在YOLOv2中，首先采用448×448分辨率的ImageNet数据fine tune使网络适应高分辨率输入；
- 然后将该网络用于目标检测任务finetune。高分辨率输入使结果提升了4%mAP。

YOLO v2

- **Convolutional With Anchor Boxes**
  - 在这一版本中借鉴了Faster R-CNN中的anchor思想，用anchor boxes来预测bounding boxes。准确率只有小幅度的下降，而召回率则提升了7%
- **Dimension Clusters**
  - 使用了K-means聚类方法类训练bounding boxes，可以自动找到更好的boxes宽高维度。
- **Direct location prediction**
  - 在YOLO模型上采用anchor boxes的是模型不稳定性，大部分不稳定因素来源于预测boxes位置（x，y）。
  - 将预测偏移量改变为YOLO的预测grid cell的位置匹配性（location coordinate），将预测值限定在0-1范围内，增强稳定性。

YOLO v2

Fine-Grained Features

- 改进后的YOLO对13×13的feature map进行目标检测。更精确的特征（finer grained features）可以提高对于小目标的检测。
- 加入passtrough层以增加特征。passthrough类似于ResNet，将高分辨率特征和低分辨率特征结合，使26×26×512的特征图转化为13×13×2048的特征图。该改进增加了1%的性能。

Multi-Scale Training

- 模型只包含卷积层和pooling 层，因此可以随时改变输入尺寸。每经过10次训练，就会随机选择新的图片尺寸进行训练。

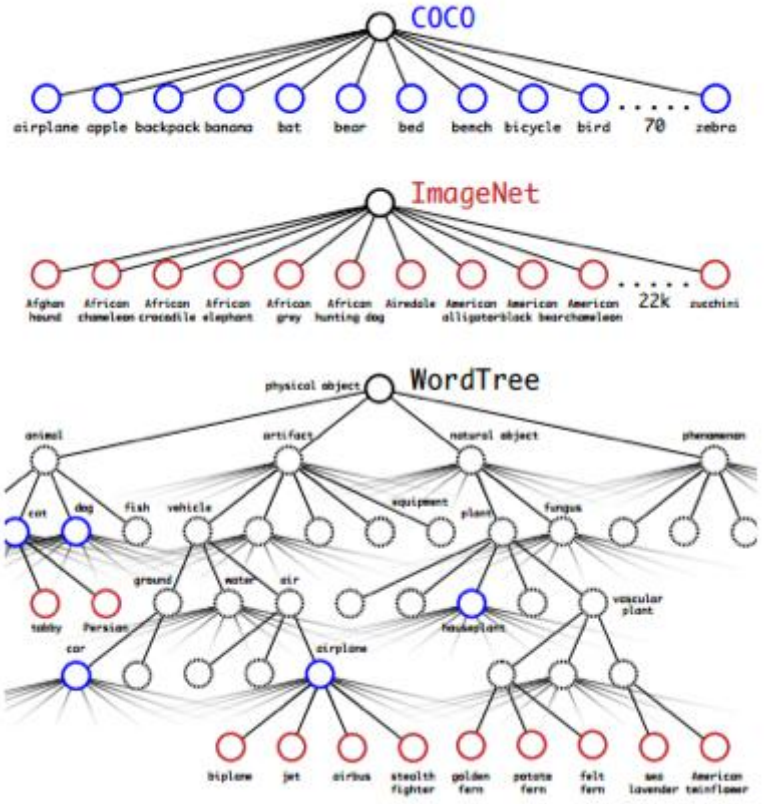
YOLO v2

- YOLO v2的基础模型是Darknet-19，其结构如表所示。
  - 使用较多的3 * 3卷积核，在每一次池化操作后把通道数翻倍。
  - 网络使用了全局平均池化，把1 * 1的卷积核置于3 * 3的卷积核之间，用来压缩特征。
  - 也用了BN稳定模型训练。

Type	Filters	Size/Stride	Output
Convolutional	32	3 × 3	224 × 224
Maxpool		2 × 2/2	112 × 112
Convolutional	64	3 × 3	112 × 112
Maxpool		2 × 2/2	56 × 56
Convolutional	128	3 × 3	56 × 56
Convolutional	64	1 × 1	56 × 56
Convolutional	128	3 × 3	56 × 56
Maxpool		2 × 2/2	28 × 28
Convolutional	256	3 × 3	28 × 28
Convolutional	128	1 × 1	28 × 28
Convolutional	256	3 × 3	28 × 28
Maxpool		2 × 2/2	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Maxpool		2 × 2/2	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	1000	1 × 1	7 × 7
Avgpool		Global	1000
Softmax			

YOLO v2/9000

- ImageNet的数据标签来源于WordNet，具有一定层次结构。
- 作者根据WordNet建立了ImageNet标签的树（WordTree）。
- 作者采用wordtree来综合ImageNet和COCO数据集

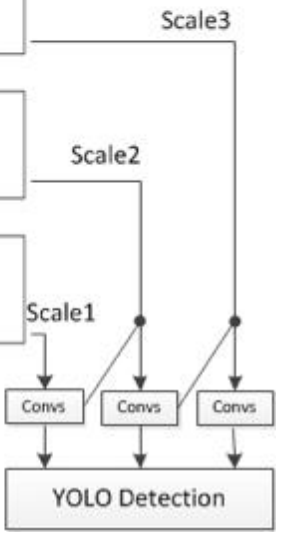


YOLO v3

Darknet-53

- YOLO v3中使用了一个53层的卷积网络，这个网络由残差单元叠加而成。
- YOLO v3使用逻辑回归预测每个边界框的分数。
- 为了实现多标签分类，模型不再使用softmax函数作为最终的分类器，而是使用 binary cross-entropy作为损失函数。

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			



YOLO v3

- 多尺度预测：YOLO v3从三种不同尺度的特征图谱上进行预测任务。
  - 在Darknet-53得到的特征图的基础上，经过7个卷积得到第一个特征图谱，在这个特征图谱上做第一次预测。
  - 然后从后向前获得倒数第3个卷积层的输出，进行一次卷积一次x2上采样，将上采样特征与第43个卷积特征连接，经过7个卷积得到第二个特征图谱，在这个特征图谱上做第二次预测。
  - 然后从后向前获得倒数第3个卷积层的输出，进行一次卷积一次x2上采样，将上采样特征与第26个卷积特征连接，经过7个卷积得到第三个特征图谱，在这个特征图谱上做第三次预测。

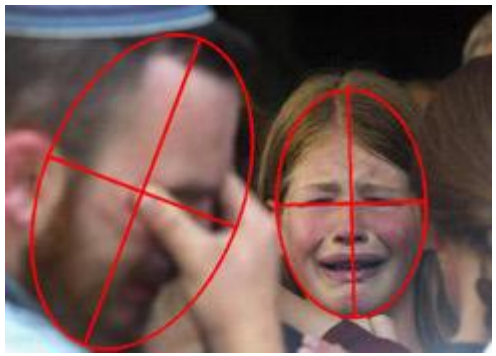
人脸检测

FDDB - Face Detection Data set and Benchmark

- 2845张图片/5171张人脸
- 椭圆标注
- 灰度图/彩色图
- 检测难点
  - 遮挡（Occlutions）
  - 不同姿态（Different poses）
  - 低分辨率（Low resolution）
  - 失焦（Out-of-focus）
- Url
  - http://vis-www.cs.umass.edu/fddb/







人脸检测

IJB-A - IARPA Janus Benchmark A

- 5,712张图片/2,085个视频/49,759张人脸
- 识别
  - 500个人
  - Meta data: 性别、肤色、姿态等
- 无约束的人脸数据集
- Url
  - https://www.nist.gov/itl/iad/image-group/ijba-dataset-request-form

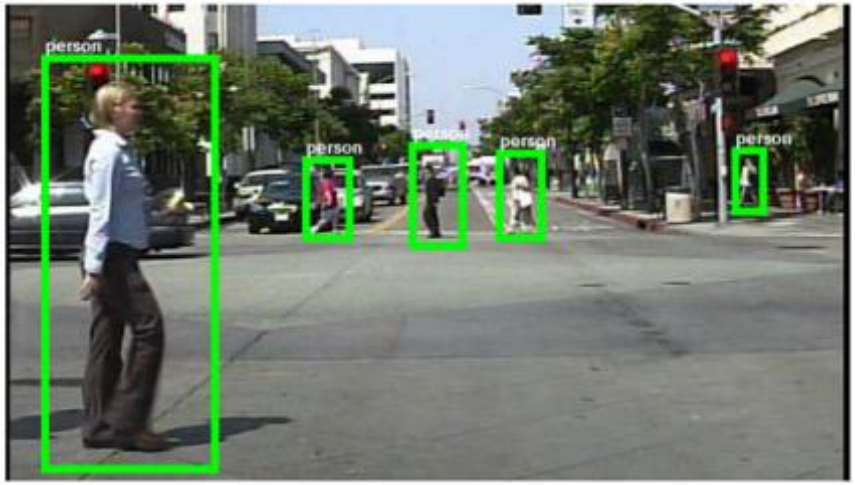


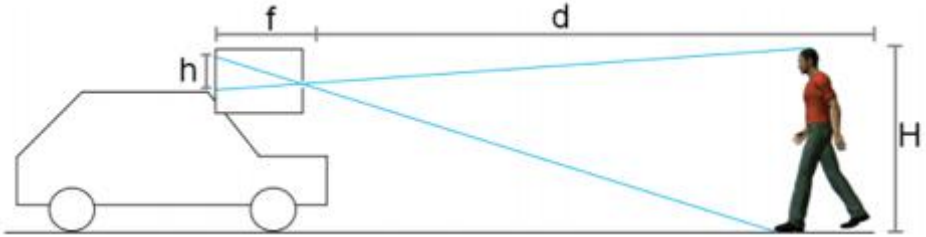
行人检测

Caltech

- 10小时640x380视频/ 250,000帧
- 车载摄像头拍摄
- 350,000行人bounding box
- Url
  - http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/







参考资料

- https://blog.csdn.net/tinyzhao/article/details/53717136
- https://www.cnblogs.com/dudumiaomiao/p/6560841.html
- https://blog.csdn.net/hrsstudy/article/details/70305791
- http://www.360doc.com/content/17/0809/10/10408243_677742029.shtml
- https://blog.csdn.net/qq_34784753/article/details/78918625

参考资料

- https://blog.csdn.net/WZZ18191171661/article/details/79481135
- https://blog.csdn.net/hrsstudy/article/details/70305791
- http://www.360doc.com/content/17/0810/10/10408243_678094588.shtml
- https://www.cnblogs.com/makefile/p/YOLOv3.html

问答互动

在所报课的课程页面，

1、点击“全部问题”显示本课程所有学员提问的问题。

2、点击“提问”即可向该课程的老师和助教提问问题。



联系我们

小象学院：互联网新技术在线教育领航者

– 微信公众号：**小象学院**

