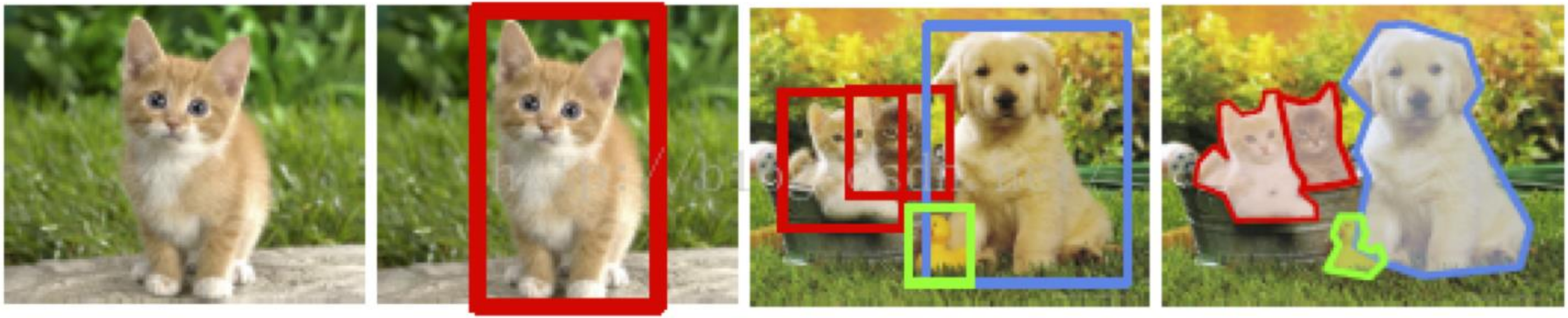
卷积神经网络之目标检测

下一个将被攻克的堡垒？

计算机视觉领域当前主要任务

分类　　定位　　检测　　分割

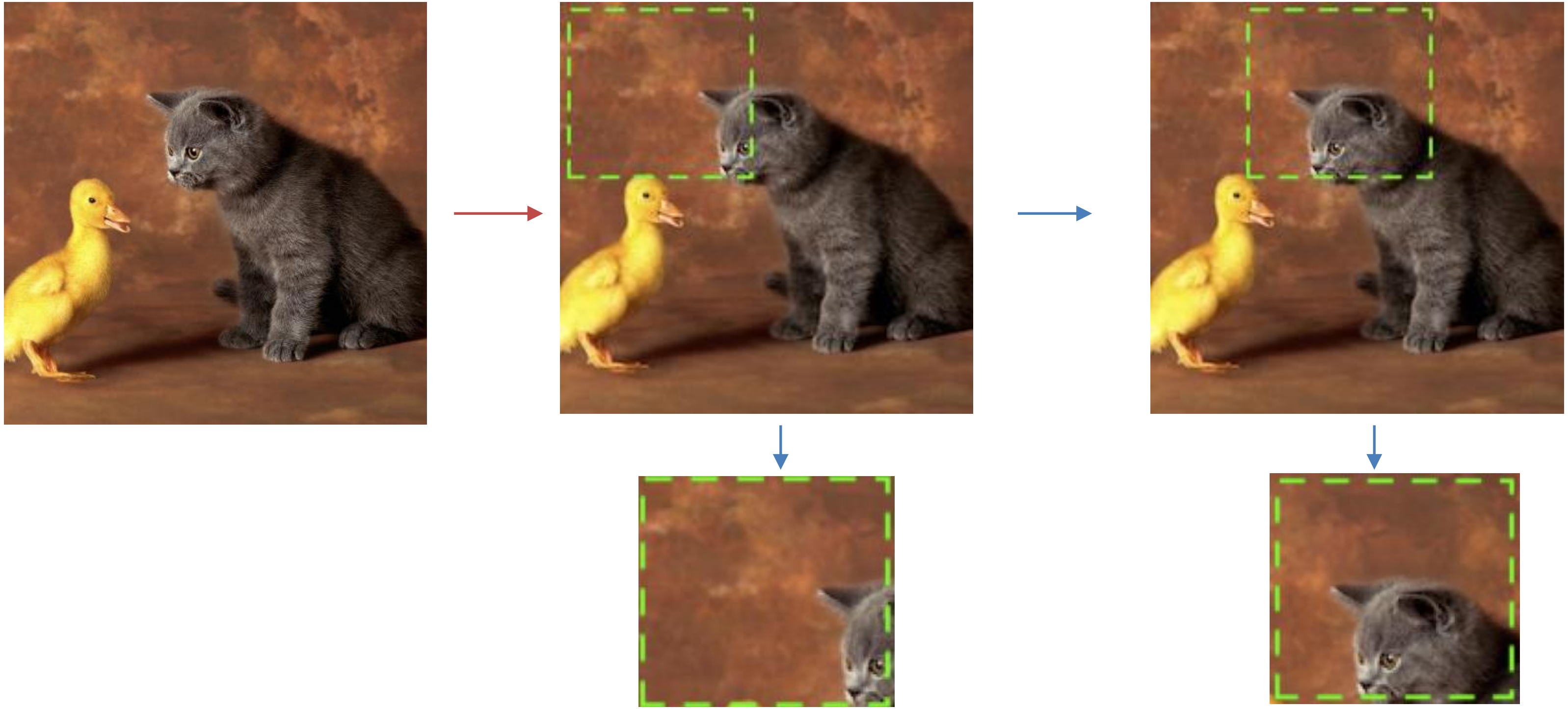


猫　　猫　　猫 狗 鸭子　　猫 狗 鸭子

单目标　　多目标

一个简单的直觉

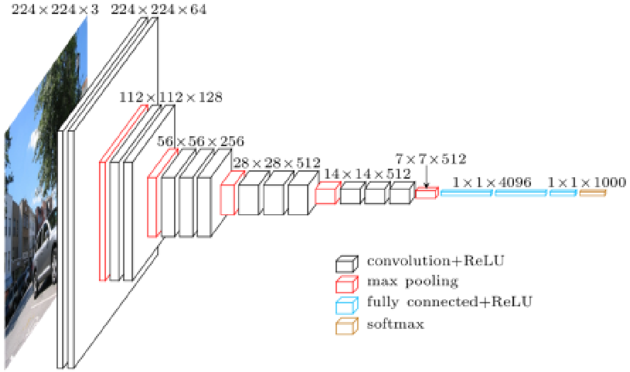
滑动窗体（slide window）



是猫吗？不太像

是猫吗？有点像

▶ 另一个简单的直觉

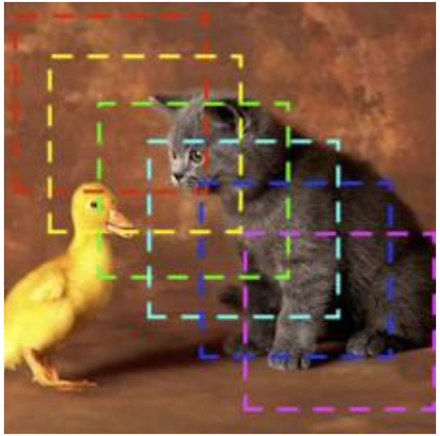


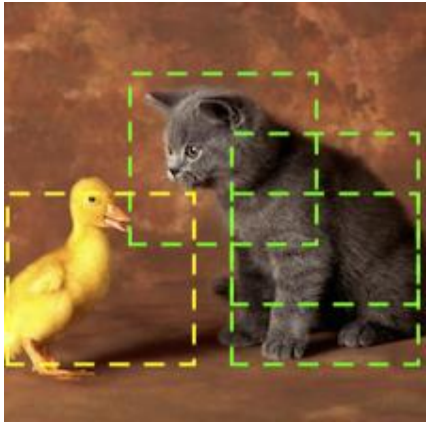
VGG：

7x7x512 $\xrightarrow{\text{flatten}}$ 1x1x4096 $\xrightarrow{\text{fully connection}}$ 1x1xclass number

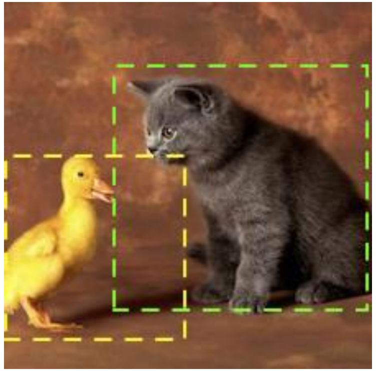
不使用7x7而是使用1x1的卷积核的话

7x7x512 $\xrightarrow{\text{conv}}$ 7x7x4096 $\xrightarrow{\text{fully connection}}$ 7x7xclass number



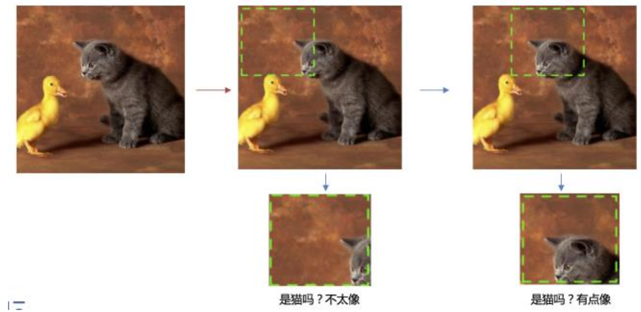


保留高置信度区域

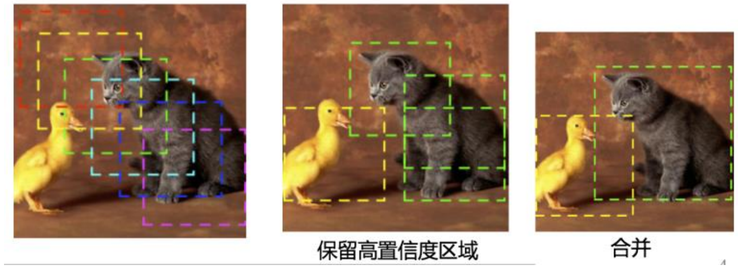


合并

▶问题在哪？



- 从每个小窗体内分析是不是目标物体
  - 方法？
  - 阈值？
  - 可靠性？
  - 窗体大小和间距？
  - 速度？

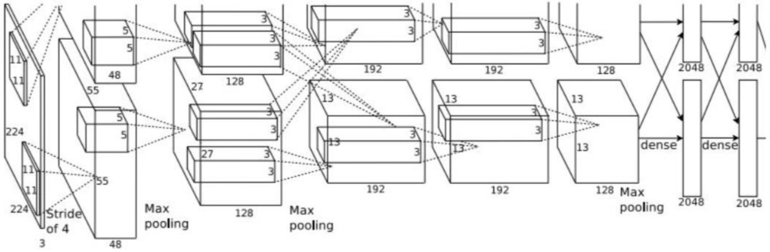


- 改造卷积网络
  - 方法？
  - 阈值？
  - 可靠性？
  - 区域大小和间距？
  - 速度？
- 最主要的问题是：都不太准。

▶定位（localization）

定位

猫



输入：　图片

输出：
- 物体类别
- 位置（坐标）

全连接
softmax

猫：0.91
狗：0.01
鸡：0.0003
鸭：0.0002
鹅：0.0003
…
飞机：$3e^{-7}$

分类

交叉熵损失

+

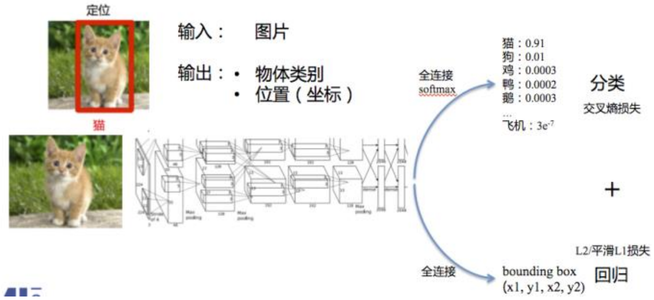
L2/平滑L1损失

全连接

bounding box
(x1, y1, x2, y2)

回归

定位（Localization）



total_loss = classification_loss + α·regression_loss

ground_truth_cls = 分类（分类数的one-hot）
ground_truth_reg = bounding box坐标值（4维向量）

这带来了一些问题和思考：

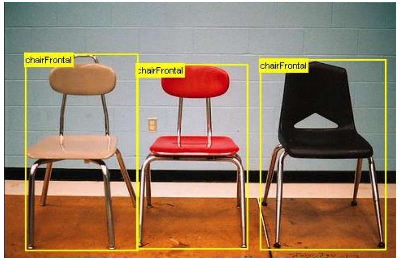
- 训练数据需要给出bounding box的坐标ground truth
- loss中的α是个超参数，需要指定
- 最后得到的特征向量包括空间信息吗？

包含位置信息的公开数据集

PASCAL VOC

http://host.robots.ox.ac.uk/pascal/VOC/voc2012

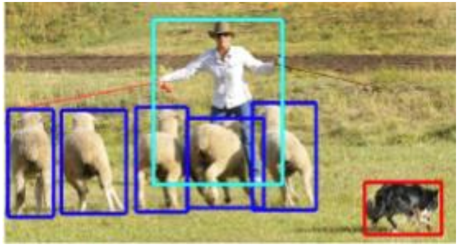
- 20个分类
- 11540张图片
- 27450个物体



COCO

http://cocodataset.org/

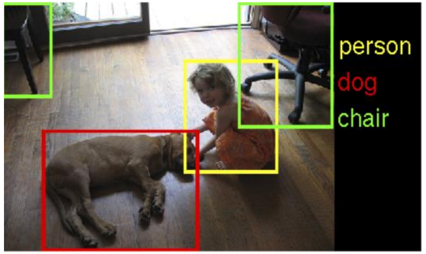
- 80个分类
- 123,287张图片
- 886,284个物体



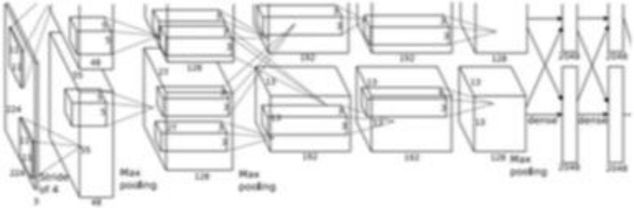
Imagenet

http://www.image-net.org/

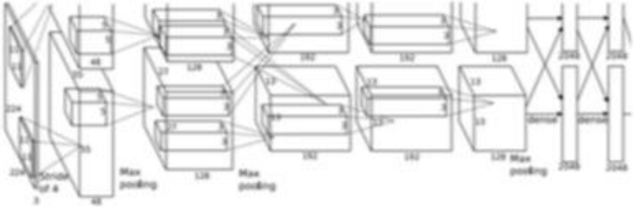
- 200个分类
- 476,688张图片
- 534,309个物体



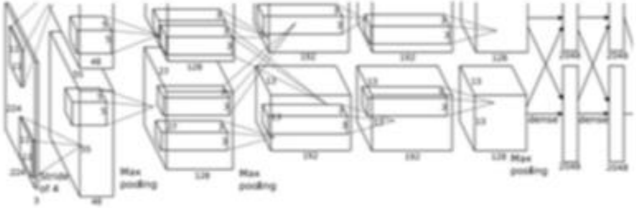
▶ 检测



bounding box
猫：(x1, y1, x2, y2)



bounding box
狗：(x1, y1, x2, y2)
狗：(x1, y1, x2, y2)
猫：(x1, y1, x2, y2)

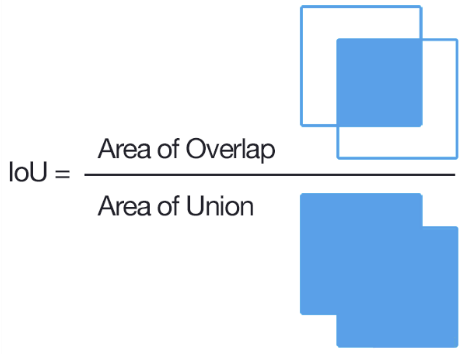


bounding box
鸭：(x1, y1, x2, y2)
鸭：(x1, y1, x2, y2)
鸭：(x1, y1, x2, y2)
…

▶ 目标检测常用名词

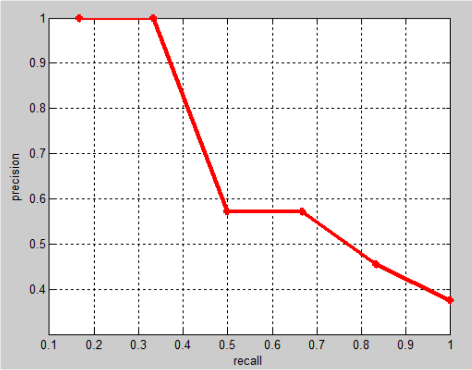
IoU：
Intersection over Union

$$IoU = \frac{Area(b_p \cap b_t)}{Area(b_p \cup b_t)}$$



mAP：
Mean Average Precision

$$mAP = \int_0^1 precision(r)dr$$

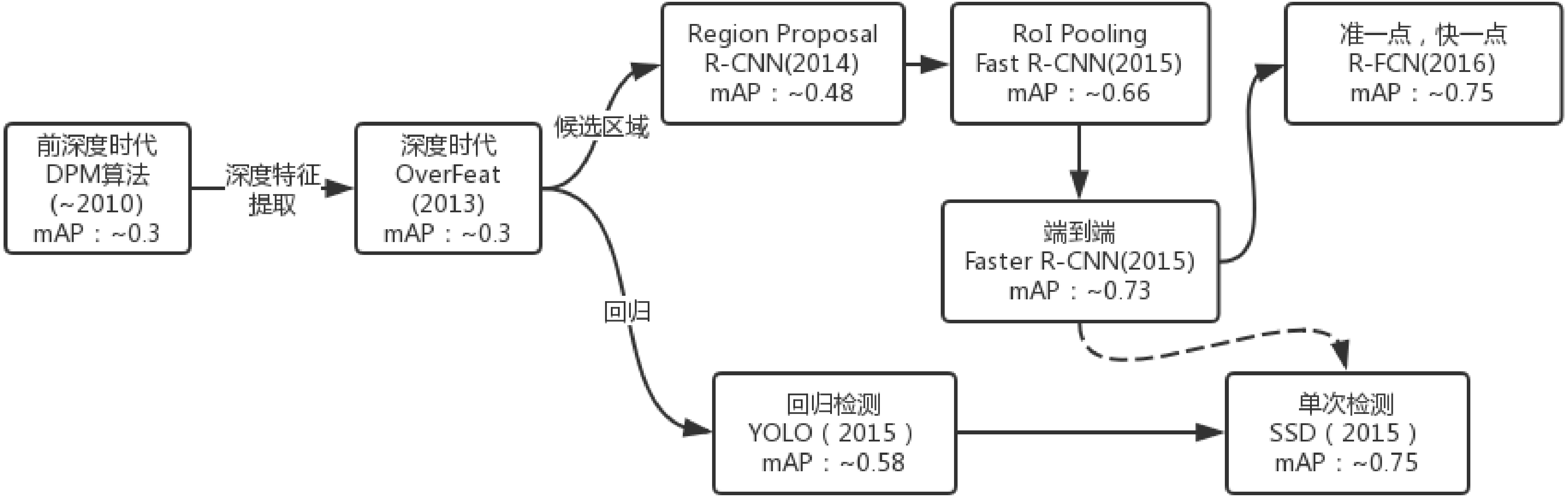


$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

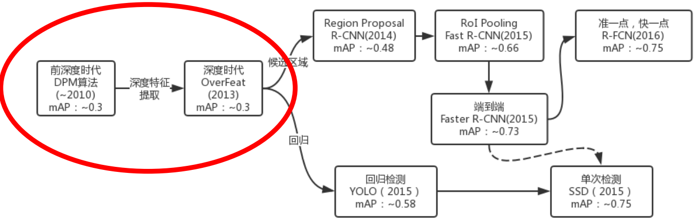
$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

RoI:
Region of Interest

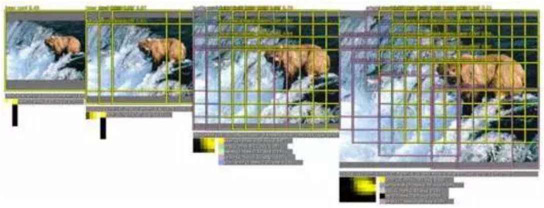
目标检测的发展历程

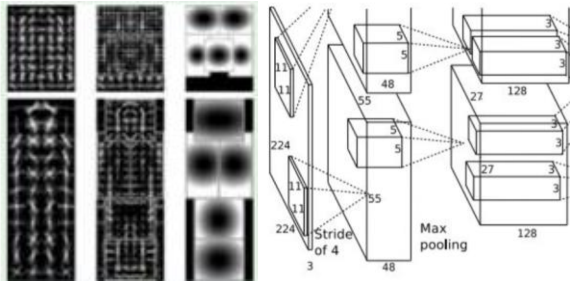

前深度时代
DPM算法
(~2010)
mAP：~0.3
深度特征
提取
深度时代
OverFeat
(2013)
mAP：~0.3
候选区域
Region Proposal
R-CNN(2014)
mAP：~0.48
RoI Pooling
Fast R-CNN(2015)
mAP：~0.66
准一点，快一点
R-FCN(2016)
mAP：~0.75
端到端
Faster R-CNN(2015)
mAP：~0.73
回归
回归检测
YOLO（2015）
mAP：~0.58
单次检测
SSD（2015）
mAP：~0.75

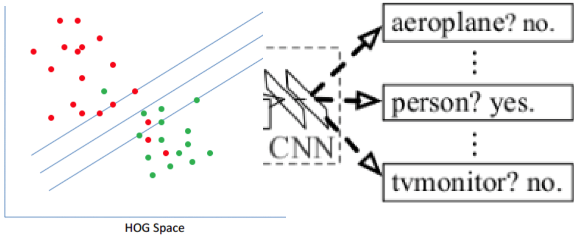
DPM到Overfeat



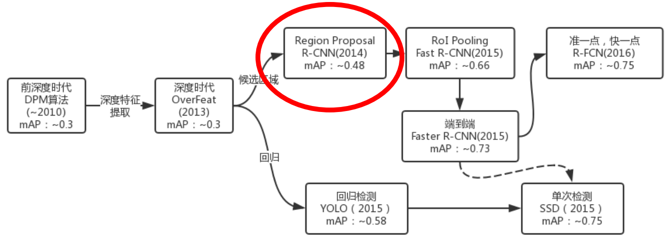
滑动窗口　　提取特征　　分类判断





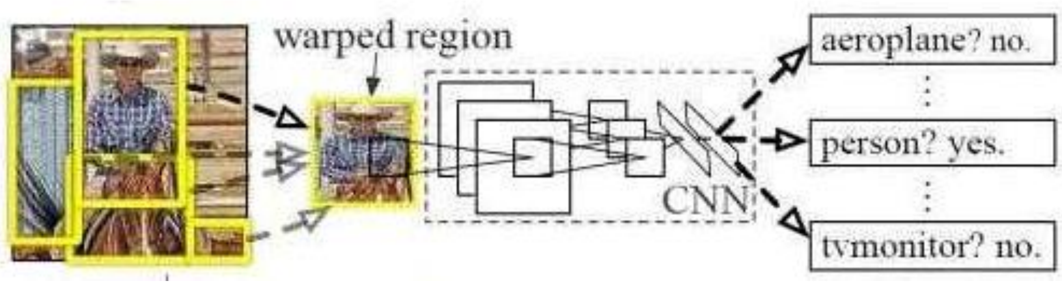


R-CNN



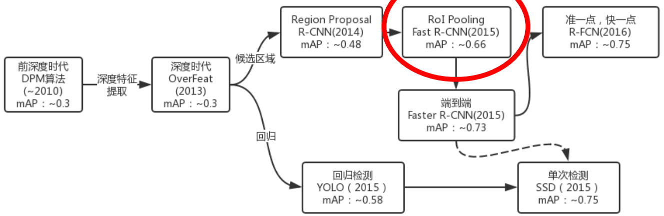
- R-CNN系列开山之作
- By Ross B. Girshick
- arXiv：1311.2524
- 精度提升极为明显（0.3→0.5）
- 速度超级慢
  - Selective search生成阶段很慢
  - 每个Proposal都要CNN做一次inference
  - 每个步骤单独训练，微调困难

候选区域		特征提取		区域分类		边框回归
Region Proposal	→	Feature extraction	→	Region classification	→	Bounding-box regression
Selective search		CNN		SVM		Linear Regression



$$\hat{G}_x = P_w d_x(P) + P_x$$
$$\hat{G}_y = P_h d_y(P) + P_y$$
$$\hat{G}_w = P_w \exp(d_w(P))$$
$$\hat{G}_h = P_h \exp(d_h(P)).$$

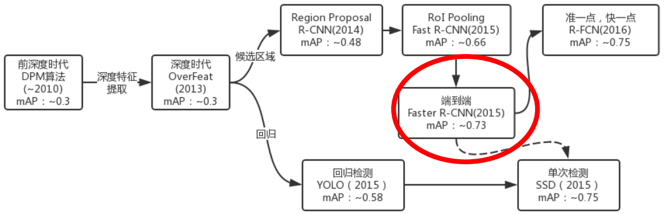
Fast R-CNN



- R-CNN系列进阶
- 仍然By Ross B. Girshick
- arXiv：1504.08083
- 精度进一步提升（0.5->0.6）
- 还有点慢，但是已经好多了
  - 将边框回归纳入到网络结构
  - 加入了RoI pooling，共享了一部分卷积操作
  - Selective search还是很慢

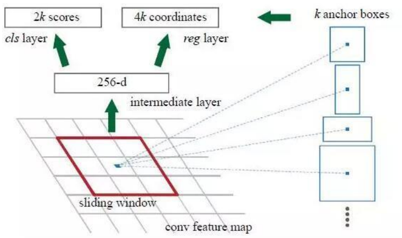
候选区域		特征提取		区域分类		边框回归
Region Proposal	→	Feature extraction	→	Region classification	&	Bounding-box regression
Selective search		CNN(RoI pooling)		Softmax		Linear Regression

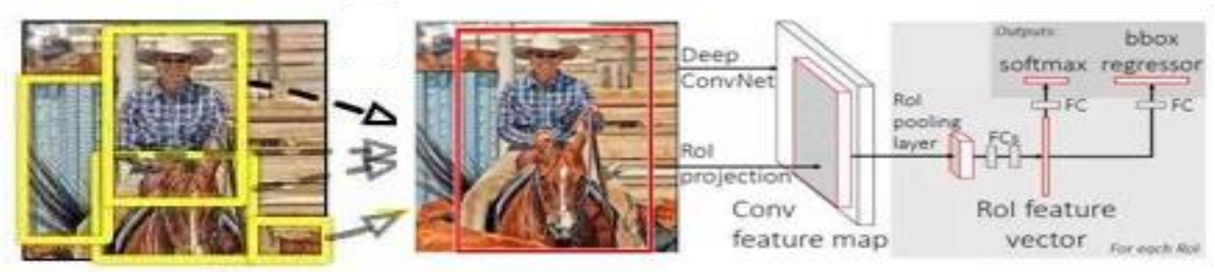
Faster R-CNN



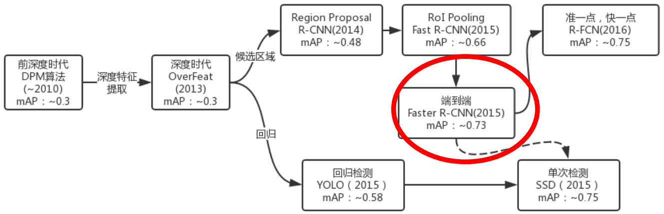
- R-CNN系列再次进阶
- 任少卿，何凯明，Ross B.G.，孙剑
- arXiv：1506.01497
- 精度进一步提升（0.6->0.7）
- 已经很快了，虽然离实时还差点
  - 用神经网络来替代selective search
  - Region的Anchor选择机制

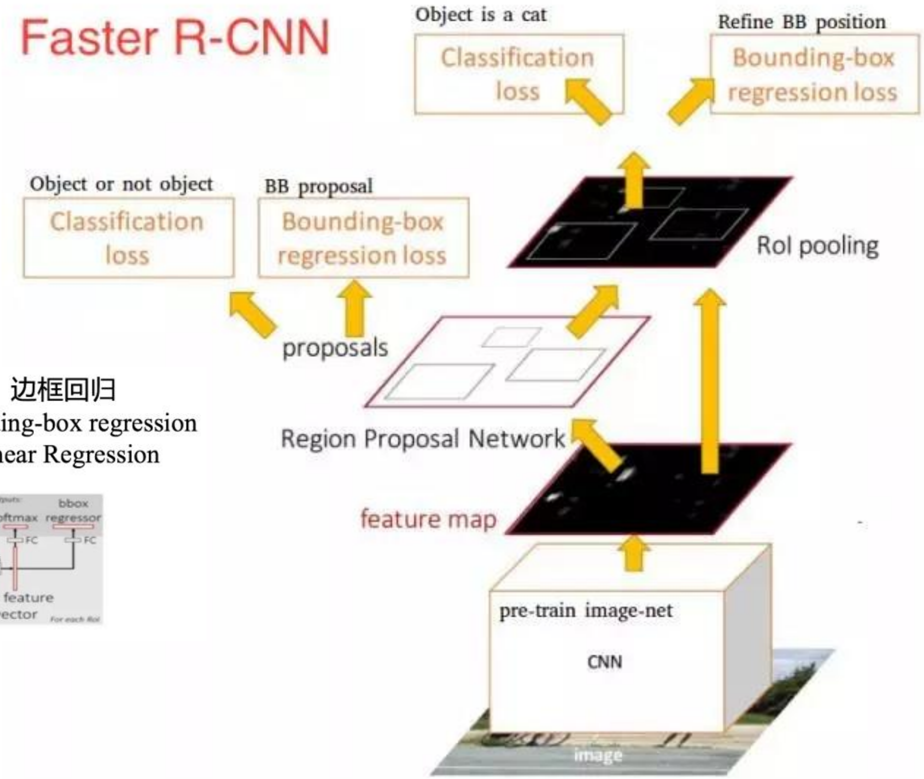
候选区域	&	特征提取	&	区域分类	&	边框回归
Region Proposal		Feature extraction		Region classification		Bounding-box regression
RPN		CNN(RoI pooling)		Softmax		Linear Regression





Faster R-CNN

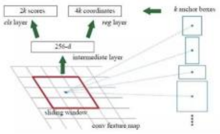


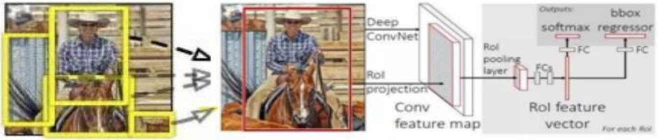


候选区域 & 特征提取 & 区域分类 & 边框回归

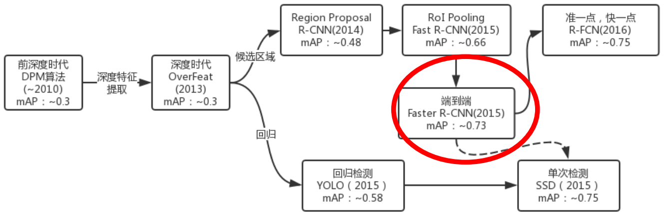
Region Proposal & Feature extraction & Region classification & Bounding-box regression

RPN | CNN(RoI pooling) | Softmax | Linear Regression



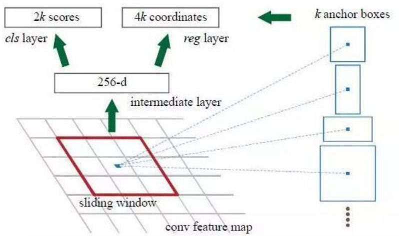


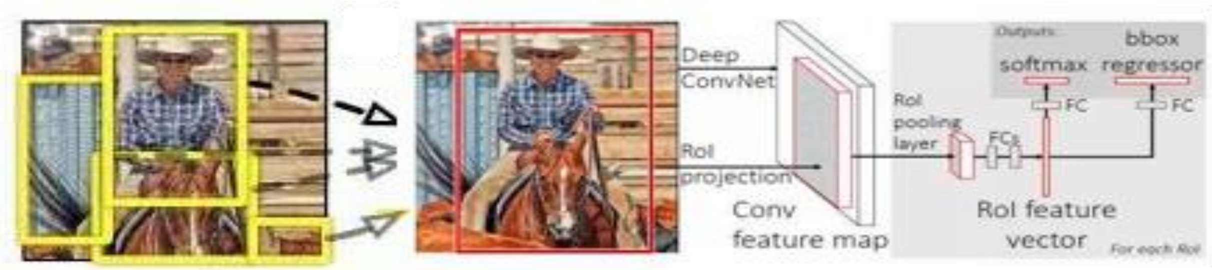
Faster R-CNN



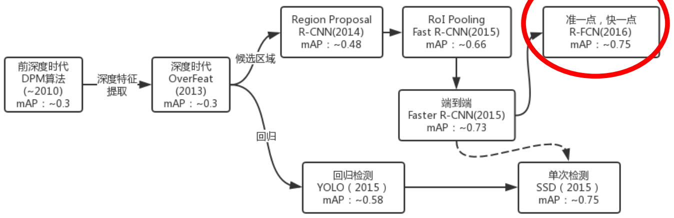
- R-CNN系列再次进阶
- 任少卿，何凯明，Ross B.G.，孙剑
- arXiv：1506.01497
- 精度进一步提升（0.6->0.7）
- 已经很快了，虽然离实时还差点
  - 用神经网络来替代selective search
  - Region的Anchor选择机制

候选区域	&	特征提取	&	区域分类	&	边框回归
Region Proposal		Feature extraction		Region classification		Bounding-box regression
RPN		CNN(RoI pooling)		Softmax		Linear Regression



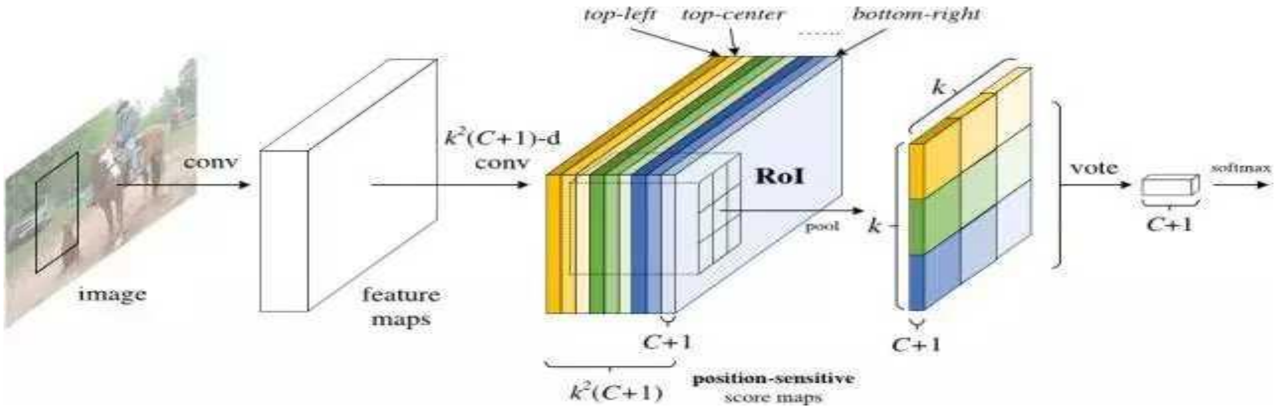


R-FCN

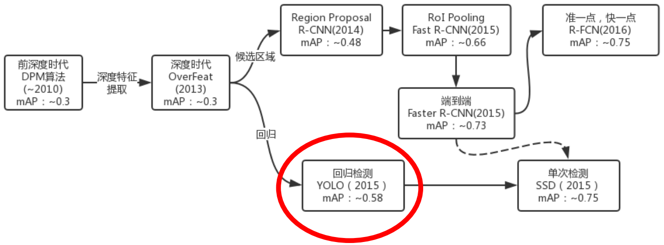


- 全卷积模型
- 代季峰，Yi Li，何凯明，孙剑
- arXiv：1605.06409
- 精度进一步提升（0.7->0.8）
- 又快了一点
  - 用全卷积结构代替全连接

候选区域 Region Proposal RPN	&	特征提取 Feature extraction CNN(RoI pooling)	&	区域分类 Region classification Softmax	&	边框回归 Bounding-box regression Linear Regression



YOLO（you only look once）

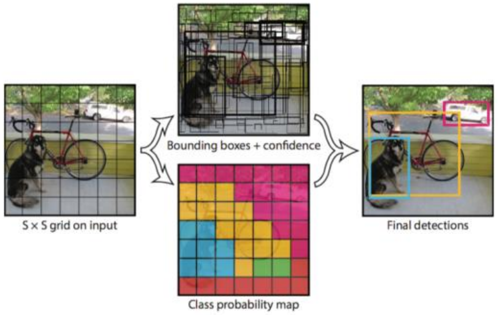


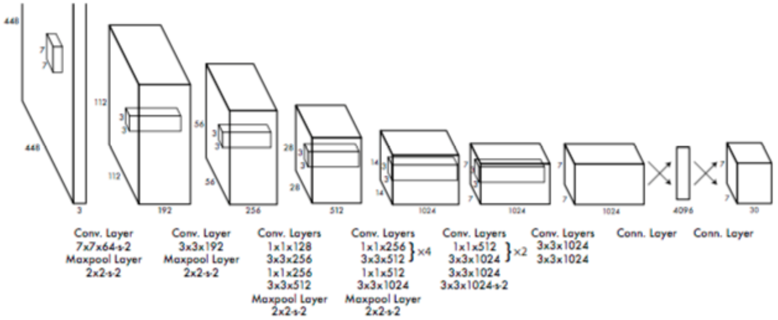
- 回归系的开创者
- 仍然有RBG的影子
- arXiv：1506.02640
- 精度和同期的fast rcnn差不多，比faster差
- 快得不是一点半点
  - 把region方法转化成了一个回归方法
  - 只跑一次卷积
- 每个小块里面只能给出一个分类的物体，不超过2个

特征提取
Feature extraction
CNN

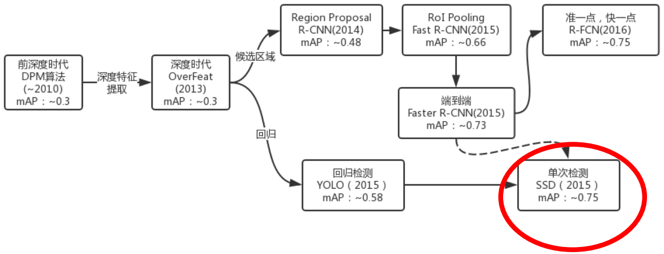
&

边框、分类回归
Bounding-box, class regression
Linear Regression





SSD（Single Shot multibox Detector）

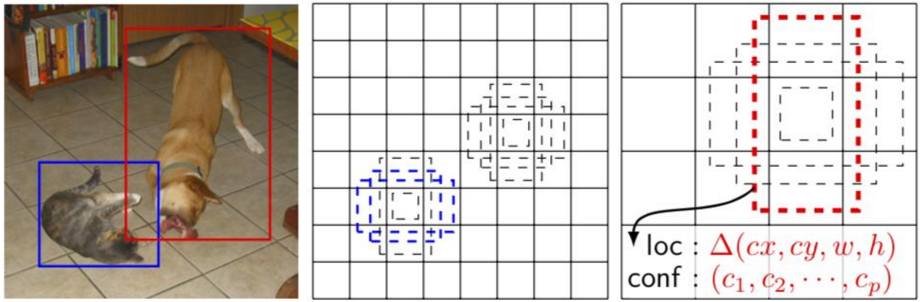


- 总算不是那几个人了
- arXiv：1512.02325
- 比YOLO准得多，和faster差不多
  - 不是每个分块预测一个类两个框
  - 吸收了faster rcnn的anchor思想
  - 多尺度feature map上提取特征
- 仍然是回归，仍然非常快

特征提取
Feature extraction
CNN

&

边框、分类回归
Bounding-box, class regression
Linear Regression



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

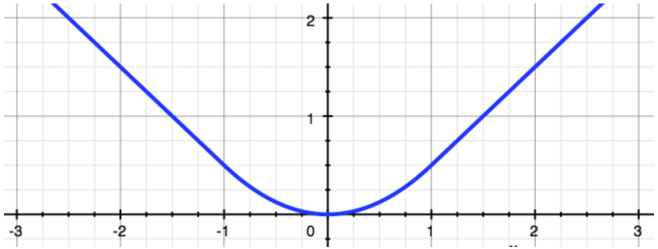
总结

	Fps	VOC 2007
Overfeat	0.5	
R-CNN	0.077	48-66%
SPP-net		63.1-82.4%
Fast R-CNN		66.9%-70%
Faster R-CNN	15(ZF Model)	73.2%-85.6%
R-FCN	6	83.6%
YOLO	45-150	58.8%
SSD	58-72	75.1%

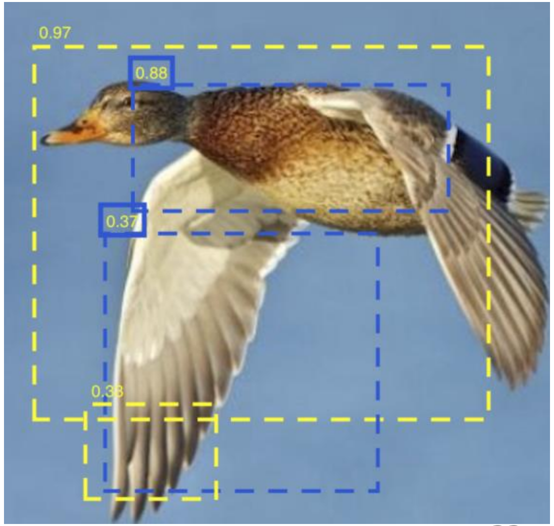
一些细节

- B-Box regression的损失函数：平滑L1损失

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & |x| \geqslant 1 \end{cases}$$

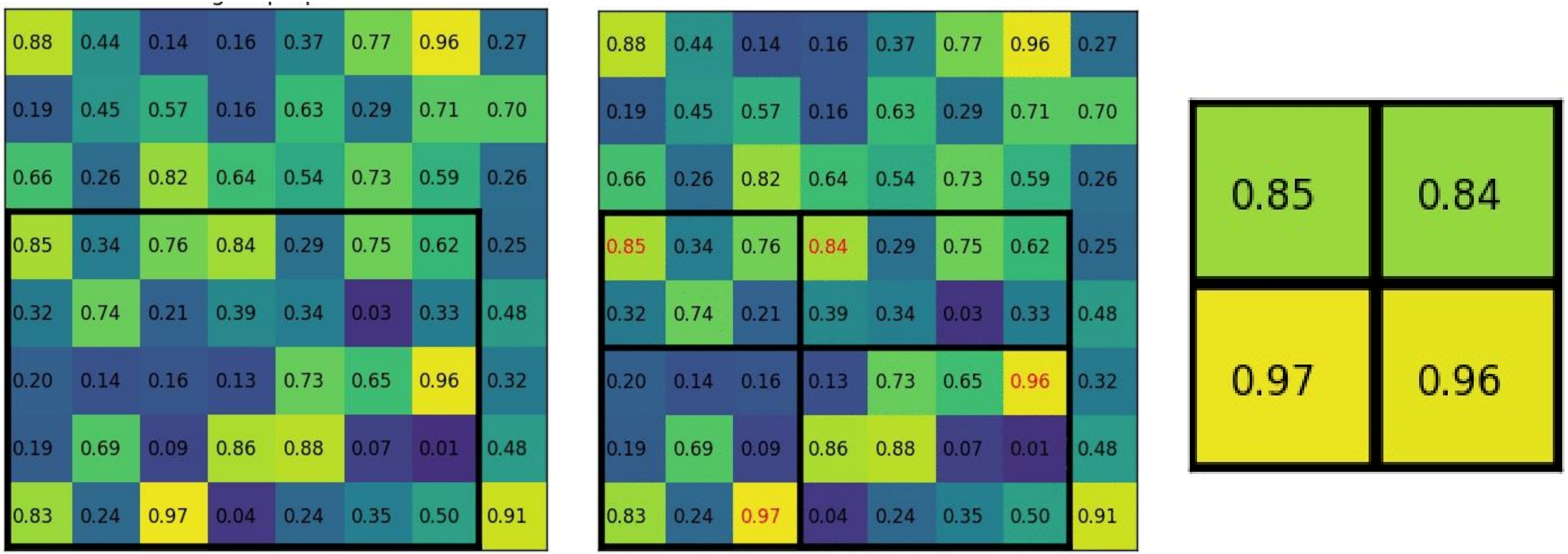


- Non-max suppression：抑制较小置信度的bounding box
  A.将所有框按置信度排序，选择最高的一个
  B.剩余所有框中删掉与当前选中框IoU>threshold的
  C.将当前选择框记录，重复A-C，直到
  - 没有剩余 或者
  - 达到最大值



一些细节

- ROI pooling：将输入的roi features调整为相同大小



▶一些细节

- RPN的实现：

  输入：conv5-3的feature map(h,w,512)

  结构：3x3 conv/relu→1x1conv/sigmoid(cls)

  →1x1conv/linear(reg)

  输出：　k=9(3 scale * 3 aspect ratio)

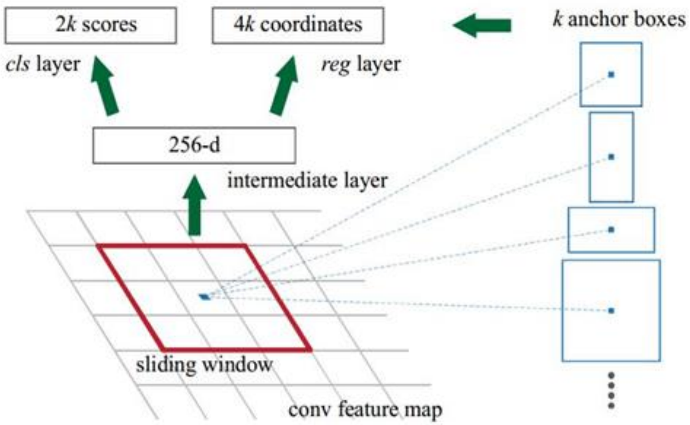
  cls：(h,w,2k)

  reg：(h,w,4k)

  Loss：

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i,p_i^*) \quad \text{交叉熵}$$

$$+\lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i^*). \quad \text{L1}$$



  Gound truth:

  对每个anchor进行标注（由原图ground truth计算得出）

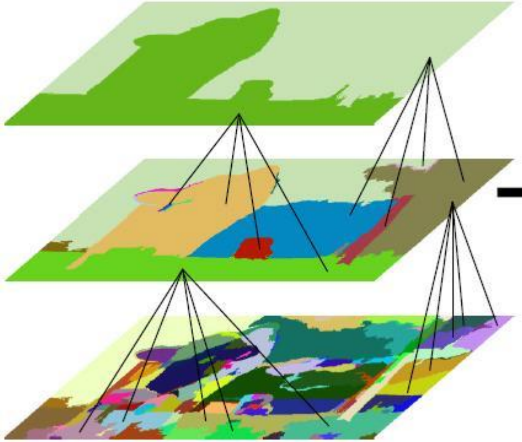
  IoU最大或者>threshold(0.7)则为正样本

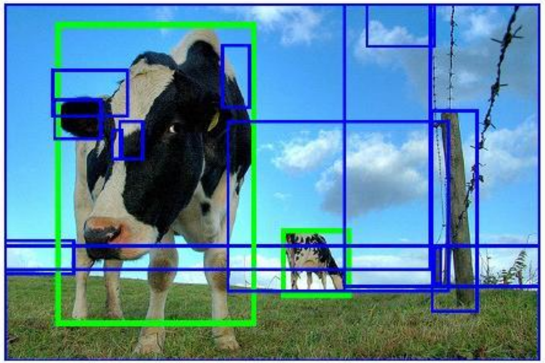
  IoU<0.3为负样本，0.3~0.7的不用于训练

  均衡采样

▶一些细节

- Selective search：
  - 在颜色、纹理等空间中映射特征
  - 计算不同区域中的特征距离
  - 合并相似区域
    - 小区域优先
    - 外接矩形的重合度高





THANK YOU