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8课　目标检测

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

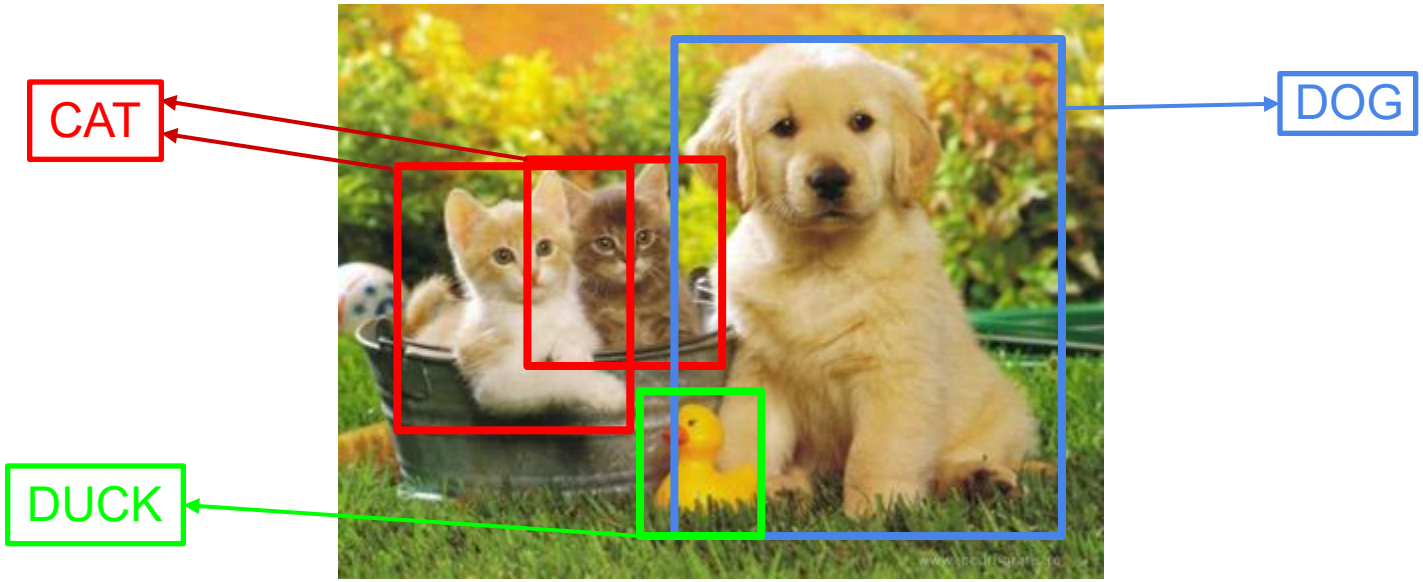
本章结构

- 目标检测（Object detection）
- ILSVRC竞赛
- 区域卷积神经网络（R-CNN）系列
  - R-CNN
  - SPP-NET
  - Fast/faster R-CNN
- 应用案例：
  - Faster R-CNN应用

目标检测

检测图片中所有物体的

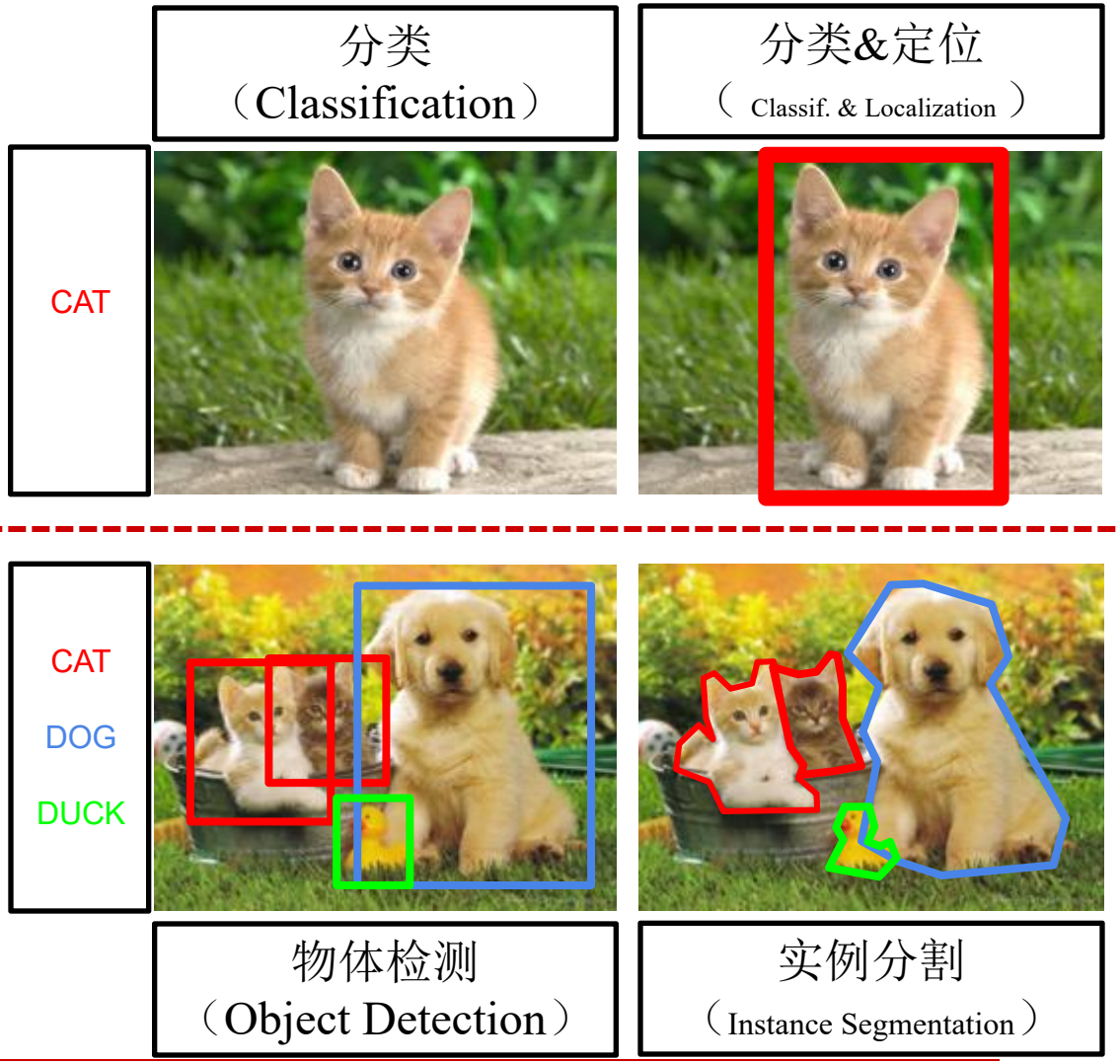
- 类别标签（Category label）
- 位置（最小外接矩形/Bounding box）



目标检测

与其他任务的区别

- 单例任务
  - 分类
  - 分类&定位
- 多例任务
  - 物体检测
  - 实例分割



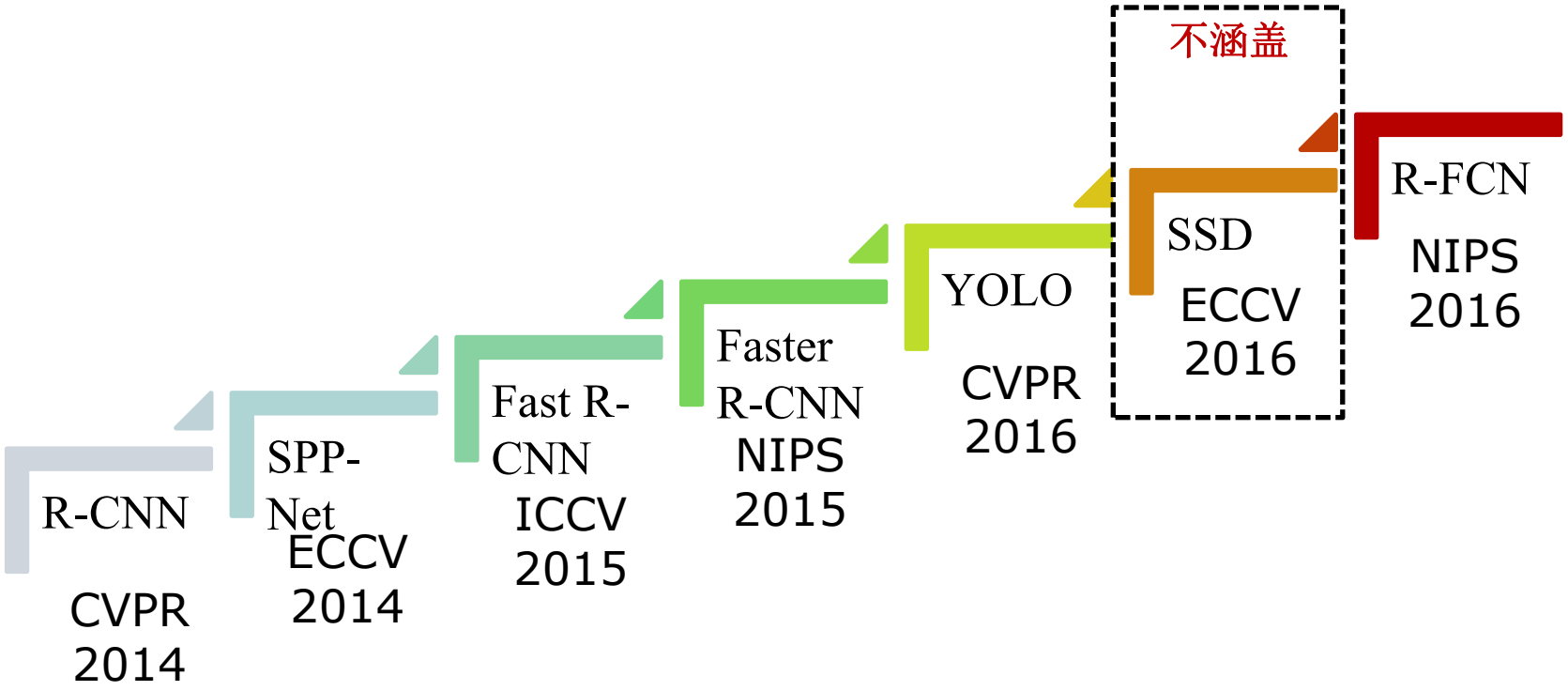
ILSVRC竞赛

Imagenet Large Scale Visual Recognition Challenge

- 物体检测（Object Detection）
  - 竞赛历史：2013-2017
  - 物体类别：200
  - 每个图片多组标签
    - 类别+Bounding box(x, y, w, h)
- 其他任务
  - 图像分类（Image Classification ）
  - 场景分类（Scene Classification）
  - 物体定位（Object Localization）
  - 场景解析（Scene parsing）
- URL： http://image-net.org/challenges/LSVRC/2017/index

区域卷积神经网络R-CNN

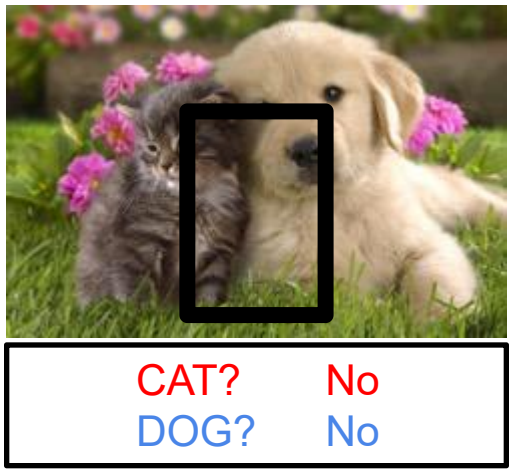
模型进化史

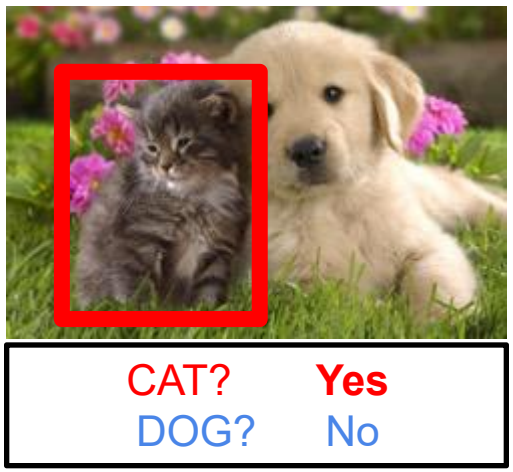


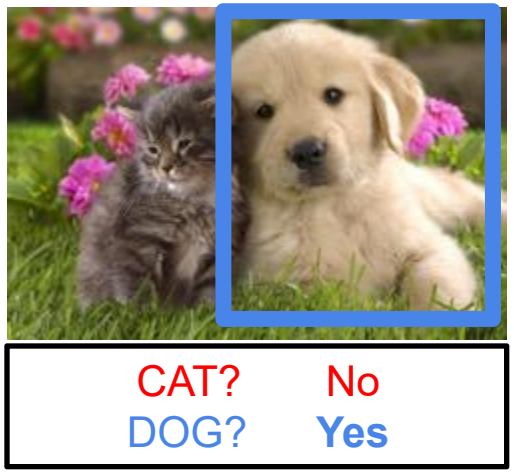
区域卷积神经网络R-CNN

模型结构

- 按分类问题对待
  - 模块1：提取物体区域（Region proposal）
    - 不同位置，不同尺寸，数量很多
  - 模块2：对区域进行分类识别（Classification）
    - CNN分类器，计算量大



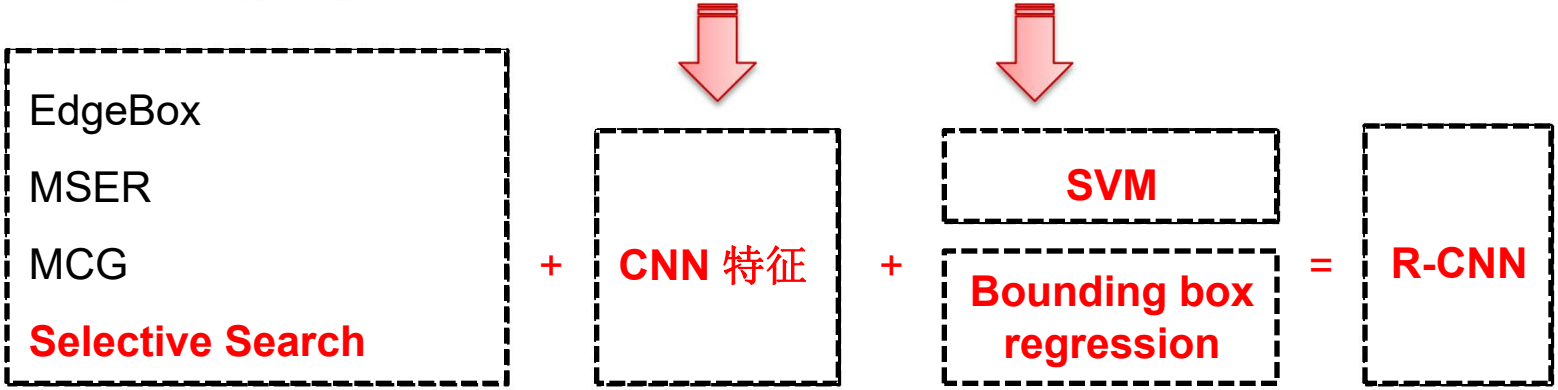


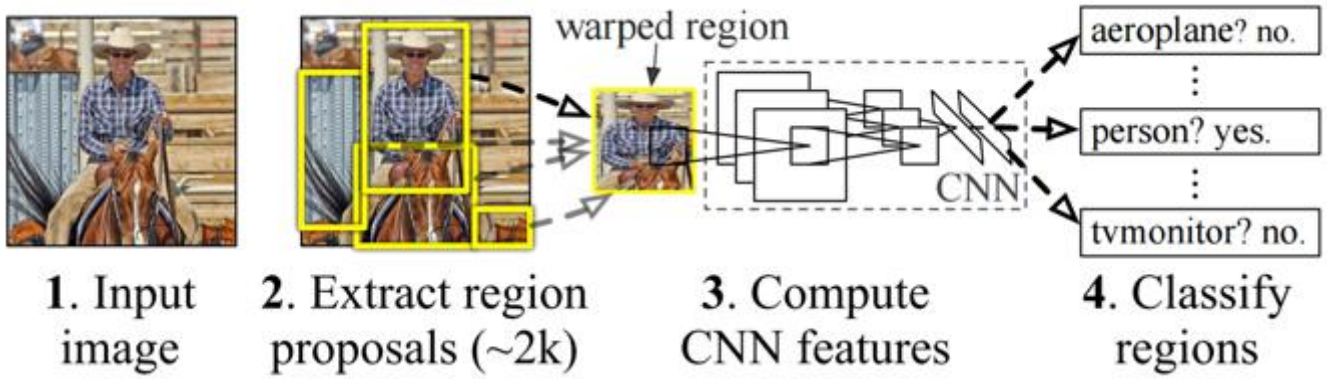


区域卷积神经网络R-CNN

传统方法 → R-CNN

- Region proposals + 手工特征 + 分类器





RCNN： selective search

- 0、按一定的规则生成区域集R
- 1、计算区域集R里每个相邻区域的相似度S={s1,s2,…}
- 2、找出相似度最高的两个区域，将其合并为新集，添加进R
- 3、从S中移除所有与2中有关的子集
- 4、计算新集与所有子集的相似度
- 5、跳至2，直至S为空

优先合并四种区域：

- 颜色（颜色直方图）相近的
- 纹理（梯度直方图）相近的
- 合并后总面积小的
- 合并后，总面积在其BBOX中所占比例大的



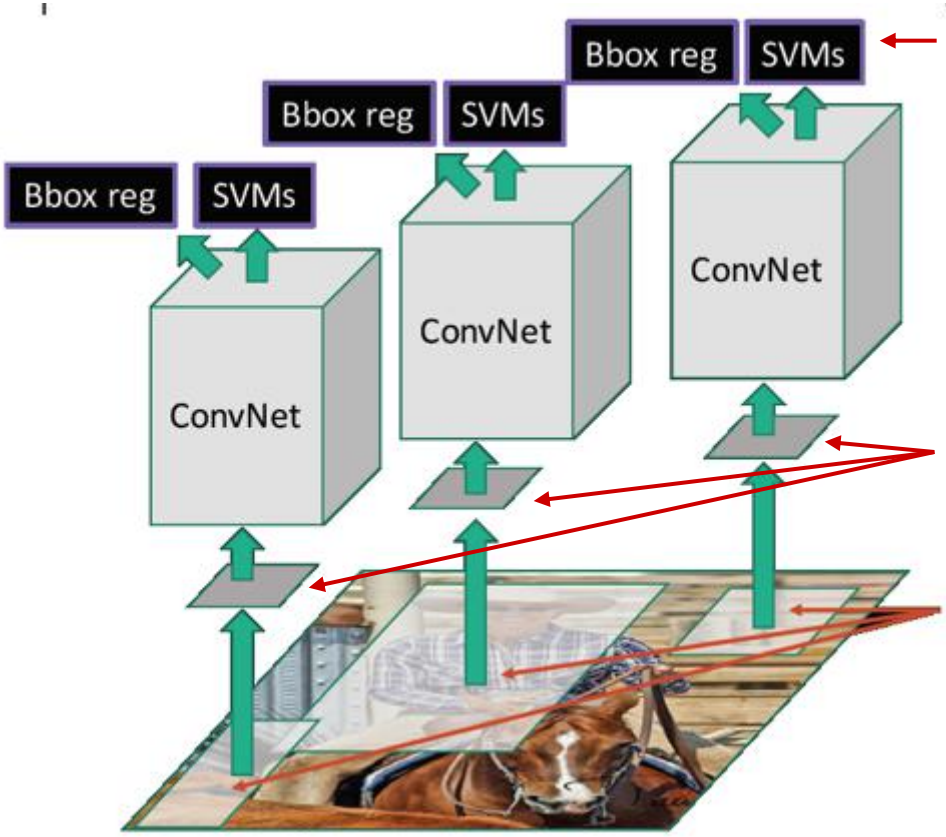
RCNN：warp

- 一、在原始区域目标周围取一块区域进行等比缩放到CNN需要的图片大小；
- 二、去除原始目标区域然后对目标区域进行填充，再等比缩放到CNN需要的图片大小；
- 三、直接将原始目标区域非等比缩放到CNN需要的图片大小



R-CNN的架构

R-CNN模块划分



模块4：Bounding box回归模型
- 对SS提供的区域进行精化
- 基于CNN特征
- 每个分类一个SVM

模块3：线性SVMs分类器
- 对CNN特征（4096）进行分类
- 每个分类一个SVM

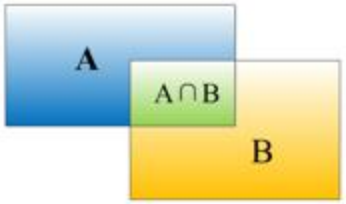
模块2：AlexNet网络
- 对所有区域进行特征提取（Fc7）
- fine-tune

区域预处理
- Bounding box膨胀（16p）
- 尺寸变换成227x227

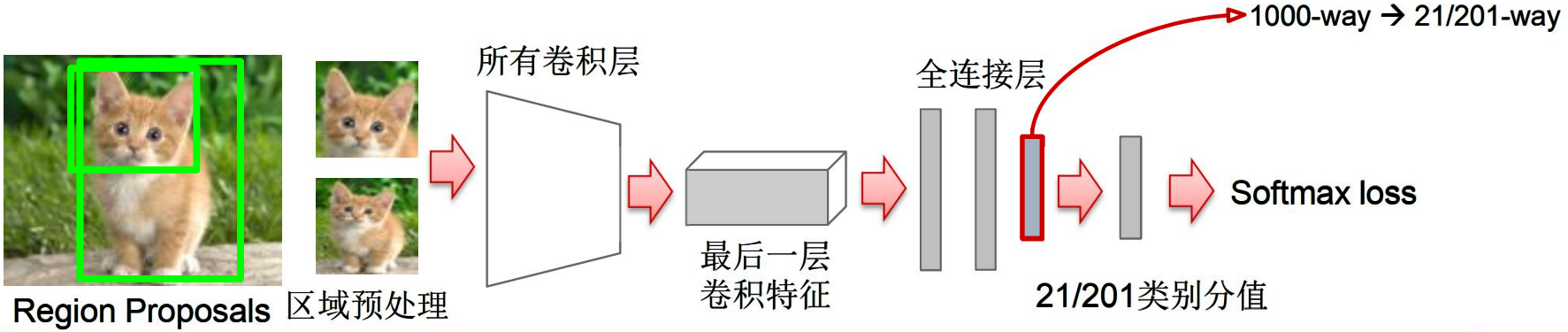
模块1：Selective Search(SS)获取区域
- ~2000个区域Region proposals
- 跟分类无关，包含物体

R-CNN：微调

IoU = (A∩B)/(A∪B)

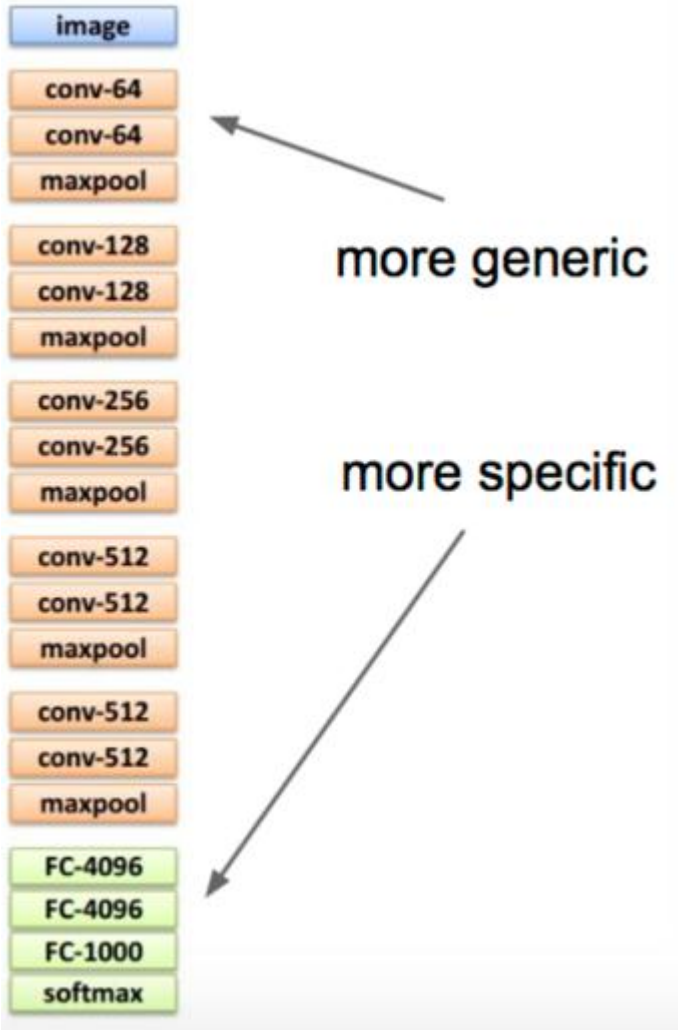


- M ← 在ImageNet上对CNN模型进行pre-train
- M'← 在SS生成的所有区域上对M进行fine-tune
  - Log loss
  - Softmax层改成(N+1)-way，其余不变
  - 正样本（N类）：跟Ground-truth重合IoU >= 0.5
  - 负样本（1 类）：IoU < 0.5

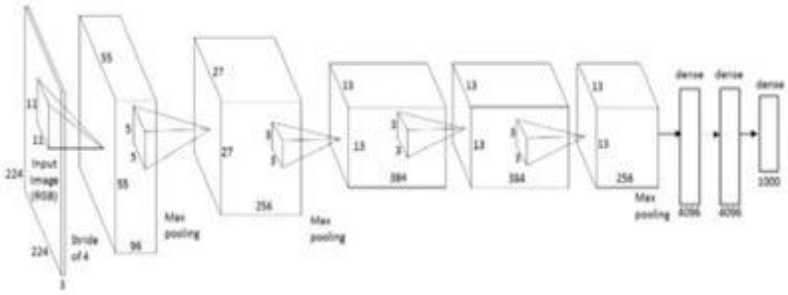


R-CNN： fine tuning

- 当数据量较小时，运用深度学习很可能会出现严重的过拟合的现象。
- 可用Alexnet、VGG等网络在ImageNet等训练集上预训练模型，然后只对于网络最后面的几层进行重新训练。
- 对于神经网络的底层，因为它充分的在大数据集上进行了基础特征的提取如：颜色、边框等，依旧可以在小数据集上进行运用。

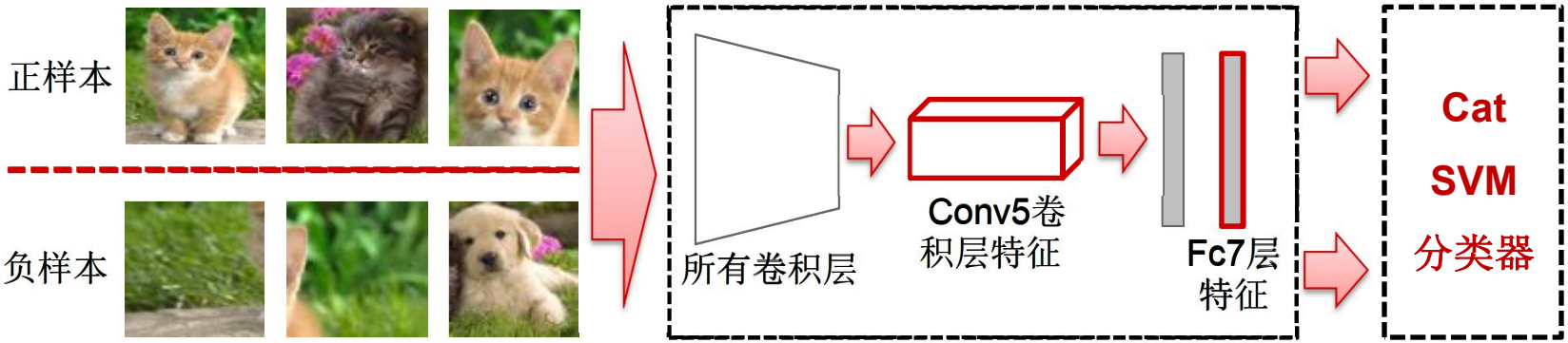


R-CNN：分类部分



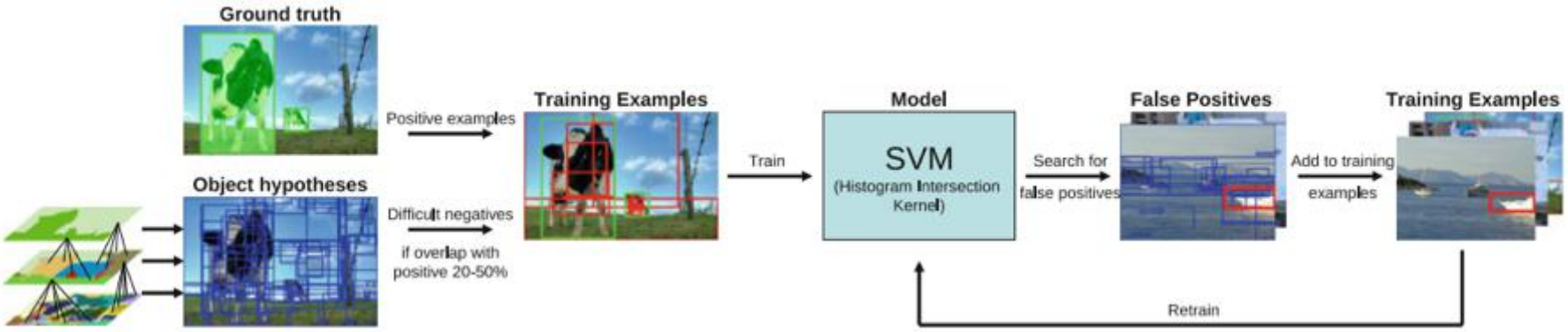
R-CNN训练流程

- C ← 在M’的Fc7特征上训练线性SVMs分类器
  - Hinge loss：SVM特有的
  - 每个类别（N类）对应一个SVM分类器
  - 正样本：所有Ground-truth区域
  - 负样本：跟Ground-truth重合IoU < 0.3的SS区域

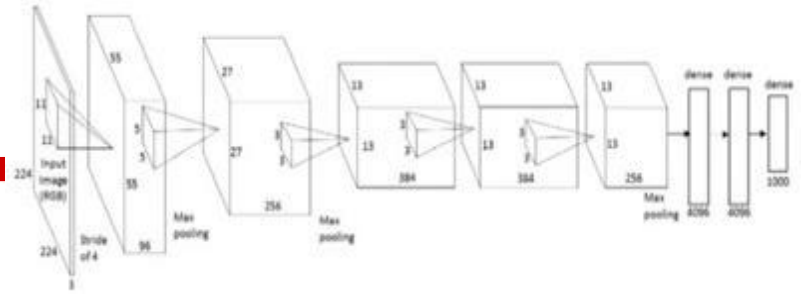


R-CNN：分类部分

- SVM训练完成后，如果完全分类正确，所有正样本的输出概率都大于0.5，而所有负样本的输出概率都小于0.5。
- 但常见的情况是有一部分的负样本的输出概率也大于0.5的, 这些样本就称之为“False Positives”。
- 如果把这些“False Positives”收集起来，对SVM进行二次训练，经过二次训练的SVM的分类准确度一般会有一定的提升。



R-CNN：回归部分



R-CNN训练流程

- R ← 在M’的Fc7特征上训练Bounding box回归模型
  - 提升定位性能（Bounding box的准确性）
  - 每个类别（N类）训练一个回归模型
    - 将SS提供的Bounding box做重新映射P→G
    - 训练输入
      - Bounding box对 $\{(P^i, G^i)\}_{i=1,\ldots,N}$
        - 中心位置（x, y）
        - 宽高尺寸（w, h）
      - CNN的Conv5特征 $\phi_5(P)$

$$P^i = (P^i_x, P^i_y, P^i_w, P^i_h)$$
$$G = (G_x, G_y, G_w, G_h)$$

$$\hat{G}_x = P_w d_x(P) + P_x$$
$$\hat{G}_y = P_h d_y(P) + P_y$$
$$\hat{G}_w = P_w \exp(d_w(P))$$
$$\hat{G}_h = P_h \exp(d_h(P))$$
$$d_\star(P) = \mathbf{w}_\star^{\mathrm{T}} \phi_5(P)$$

    - P的IoU > 0.6
    - Squared loss

$$\mathbf{w}_\star = \underset{\hat{\mathbf{w}}_\star}{\operatorname{argmin}} \sum_i^N (t^i_\star - \hat{\mathbf{w}}_\star^{\mathrm{T}} \phi_5(P^i))^2 + \lambda \|\hat{\mathbf{w}}_\star\|^2$$

$$t_x = (G_x - P_x)/P_w$$
$$t_y = (G_y - P_y)/P_h$$
$$t_w = \log(G_w/P_w)$$
$$t_h = \log(G_h/P_h).$$

  - 测试阶段
    - 参数w已经训练好

R-CNN：测试

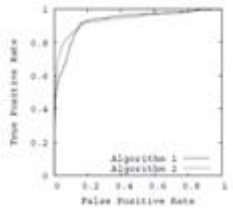
R-CNN测试阶段

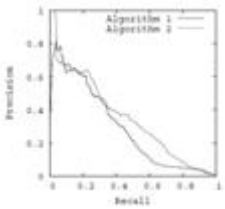
- Selective Search（fast mode）提取~2000区域/图片
- 将所有区域膨胀+缩放到227x227
- 使用fine-tune过的AlexNet计算2套特征
  - 为每个类别执行
    - Fc7特征 → SVM分类器 → 类别分值
    - 使用非极大值抑制（IoU>=0.5）获取无冗余的区域子集
      - 所有区域按分值从大到小排序
      - 剔除冗余：与最大分值区域IoU>=0.5的所有区域
      - 保留该最大分值区域，剩余区域作为新候选集
    - Conv5特征 → Bounding box回归模型 → Bbox偏差
    - 使用Bbox偏差修正区域子集

R-CNN性能评价



- True Positive区域：IoU >= 0.5
- False Positive区域：IoU < 0.5
- False Negative区域：遗漏的Ground truth区域
- 准确率precision: TP/(TP+FP)
- 召回率recall: TP/(TP+FN)





- 第C类的平均精度(AP)：PR曲线之下的面积，是Precision对于Recall的积分。
- mAP：所有类别的平均精度求和除以所有类别，即数据集中所有类的平均精度的平均值。

R-CNN性能评价

用于SVM、b-box 回归的特征需要存储到磁盘中，这将需要占用大量的磁盘空间，而且提取这些特征也会耗费好多时间。

R-CNN性能

- mAP大幅提升

数据集	Best baseline	R-CNN
PASCAL VOC 2010	35.1%	53.7%
ILSVRC	24.3%	31.4%

- 问题
  - 训练时间很长（84小时）
    - Fine-tune（18）+特征提取（63）+SVM/Bbox训练（3）
  - 测试阶段很慢：VGG16一张图片47s
  - 复杂的多阶段训练

SPP-Net

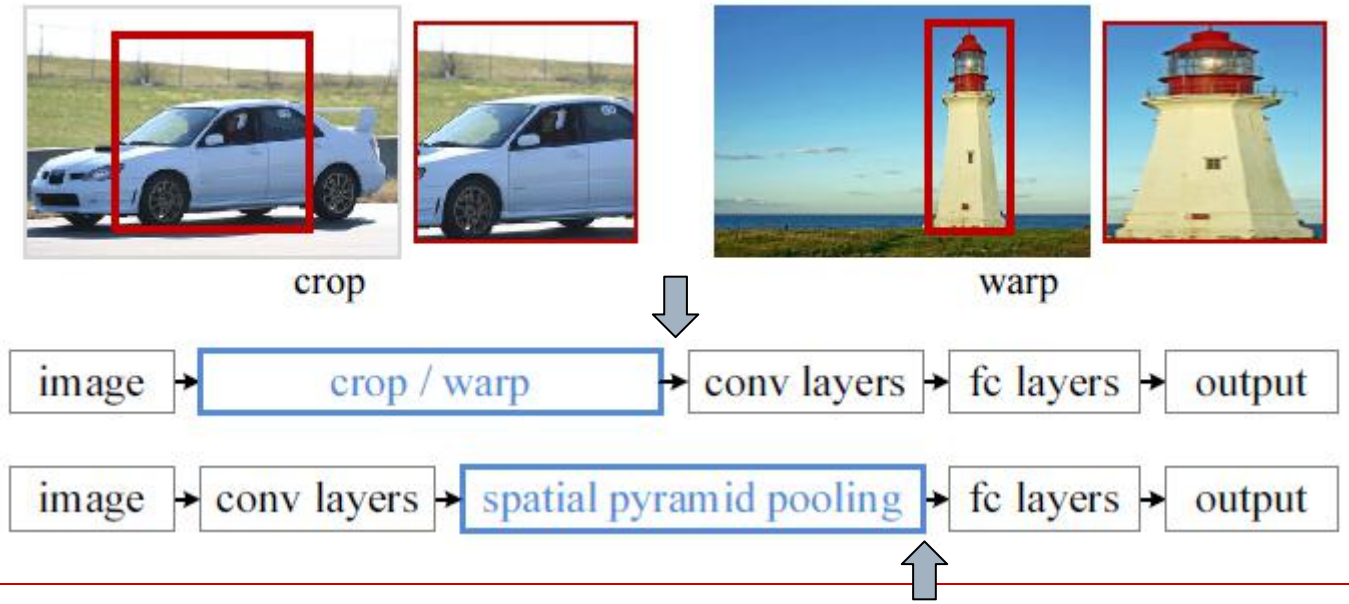
- R-CNN速度慢的原因之一
  - 卷积特征重复计算量太大
    - 每张图片的~2000区域都会计算CNN特征
- 两大改进
  - 直接输入整图，所有区域共享卷积计算（一遍）
    - 在Conv5层输出上提取所有区域的特征
  - 引入空间金字塔池化（Spatial Pyramid Pooling）
    - 为不同尺寸的区域，在Conv5输出上提取特征
    - 映射到尺寸固定的全连接层上

SPP-Net

- 使用SPP技术实现了
  - 共享计算
  - 适应不同输入尺寸

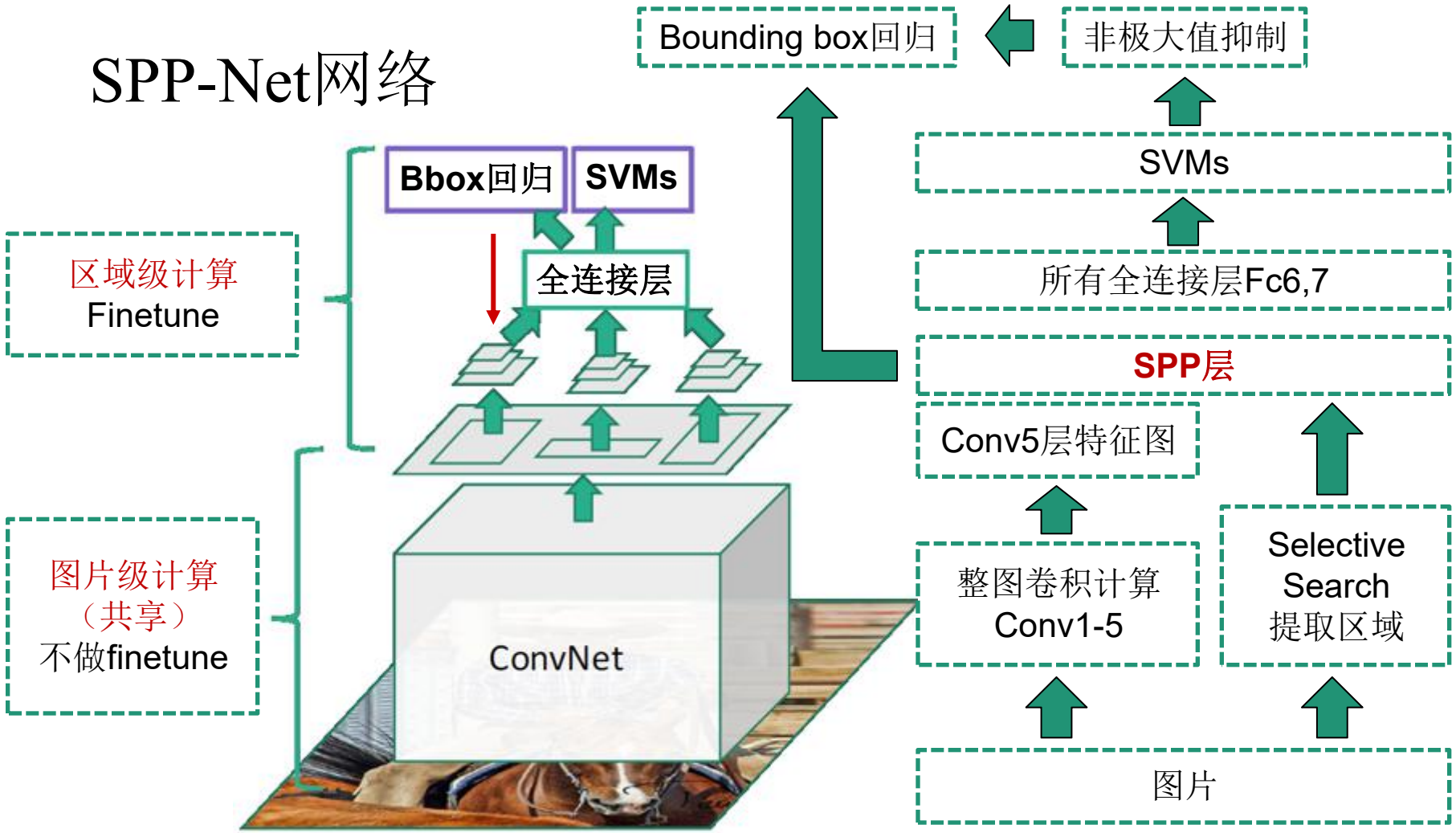
为什么CNN需要固定输入大小？

卷积层和池化层的输出尺寸都是和输入尺寸相关的，它们的输入是不需要固定图片尺寸的，真正需要固定尺寸的是最后的全连接层。



SPP-Net

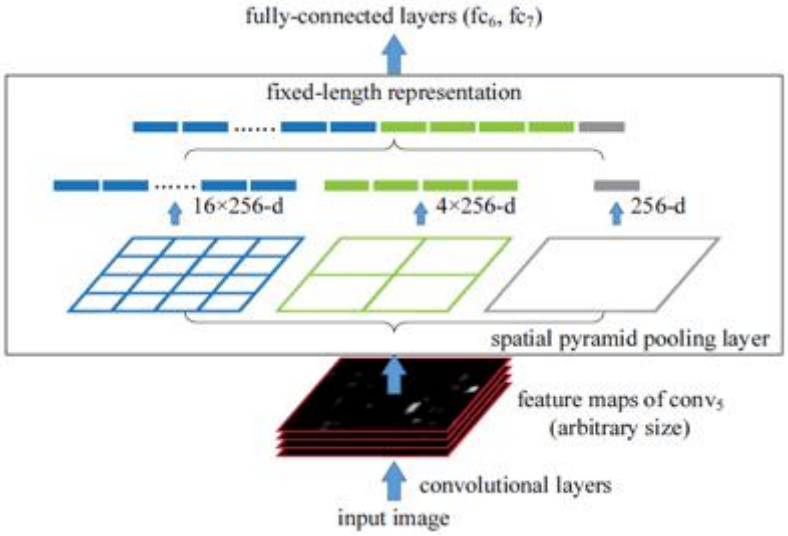
SPP-Net网络

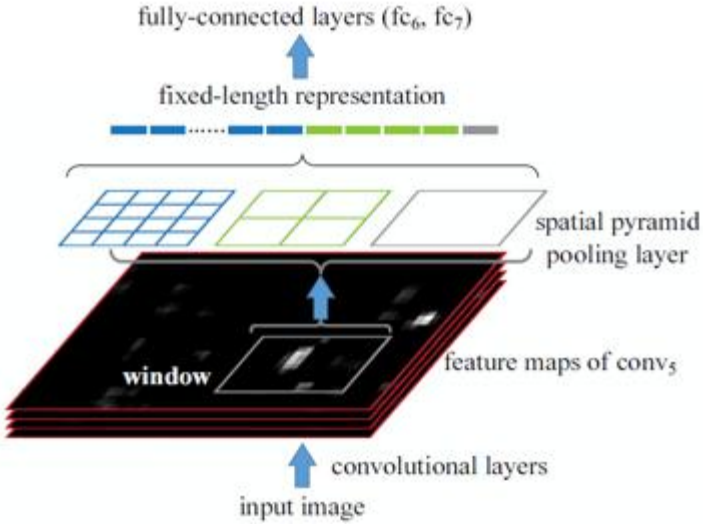


SPP-Net

- 空间金字塔池化
  - 替换Conv5的Pooling层
  - 3个level和21个Bin： 1x1, 2x2, 4x4
  - Bin内使用Max pooling

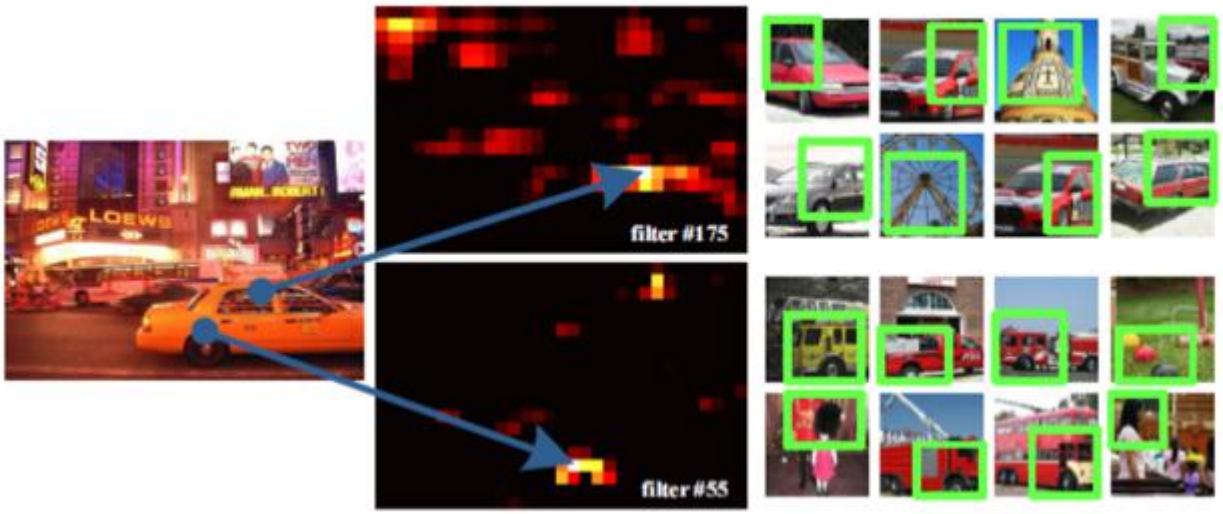
这里仅仅是分块，而无需考虑原始尺寸





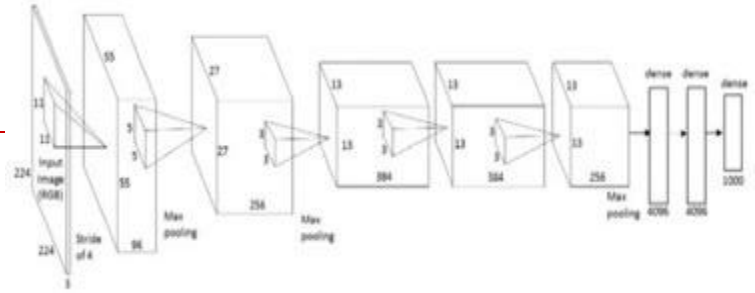
SPP-Net

- 对卷积层可视化发现：输入图片的某个位置的特征反应在特征图上也是在相同位置。
- 基于此，对某个ROI区域的特征提取只需要在特征图上的相应位置提取就可以了。



(a) image (b) feature maps (c) strongest activations

SPP-Net训练流程



SPP-Net训练流程

- M ← 在ImageNet上对CNN模型进行pre-train
- F ← 计算所有SS区域的SPP特征
- M’ ← 使用F特征fine tune新fc6→fc7→fc8层

与R-CNN区别
- SPP特征←Pool5特征
- 只finetune全连接层

- F’ ← 计算M’的fc7特征
- C ← 使用F ’特征训练线性SVM分类器
- R ← 使用F特征训练Bounding box回归模型

SPP-Net问题

SPP-Net问题

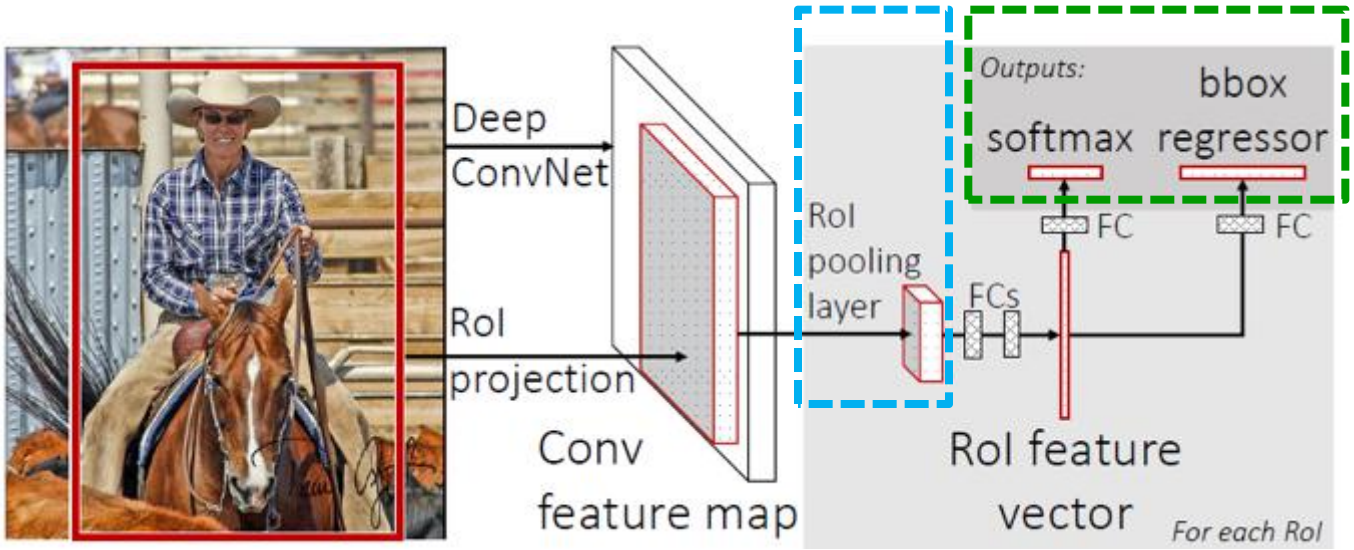
- 继承了R-CNN的剩余问题
  - 需要存储大量特征
  - 复杂的多阶段训练
  - 训练时间仍然长（25.5小时）
    - Fine-tune（16）+ 特征提取（5.5）+ SVM/Bbox训练（4）
- 带来了新问题
  - SPP层之前的所有卷积层参数不能fine tune

Fast R-CNN网络

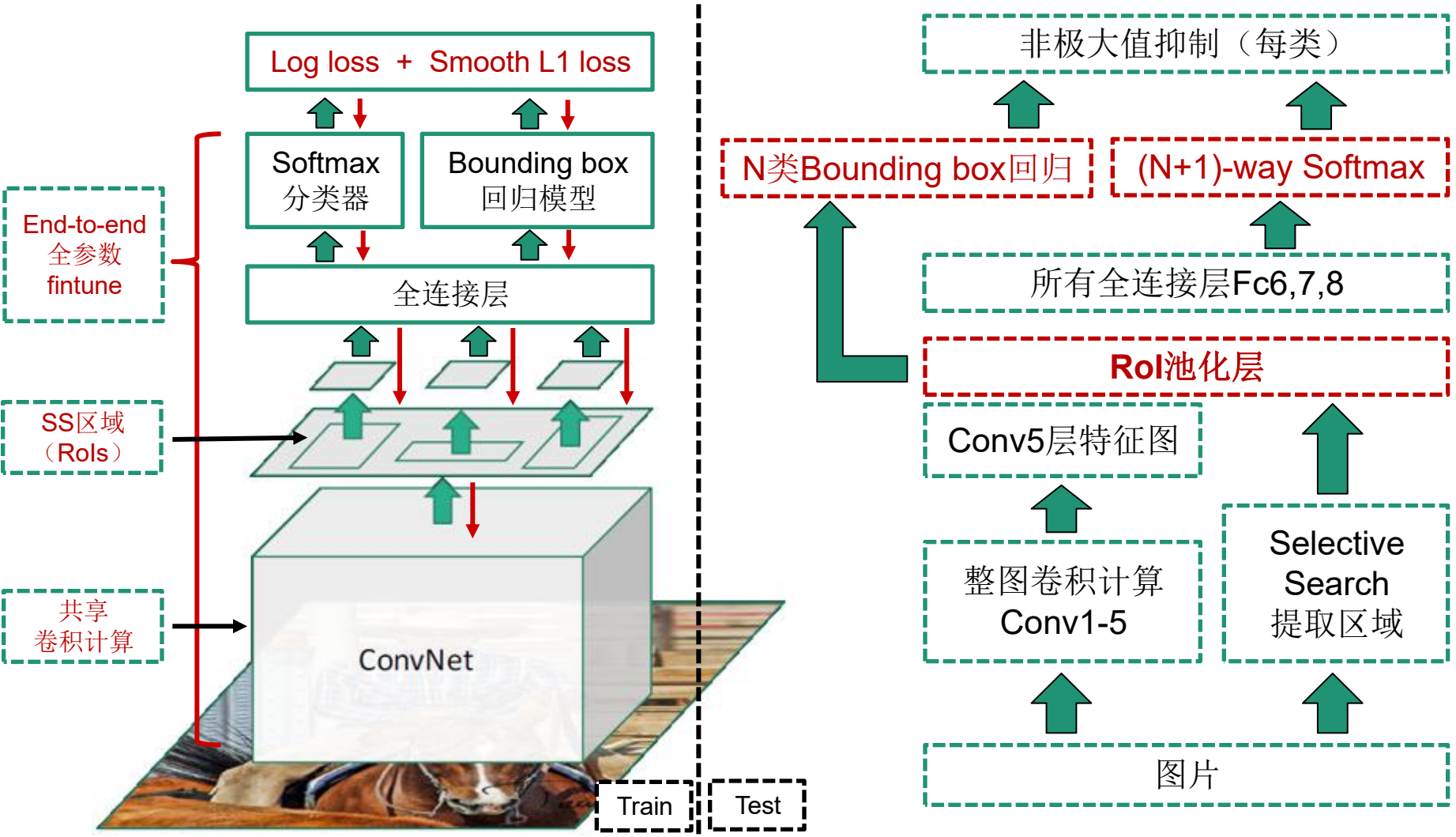
- 改进
  - 比R-CNN，SPP-Net更快的trainng/test
  - 更高的mAP
  - 实现end-to-end（端对端）单阶段训练
    - 多任务损失函数（Multi-task loss）
  - 所有层的参数都可以fine tune
  - 不需要离线存储特征文件

Fast R-CNN

- 在SPP-Net基础引入2个新技术
  - 感兴趣区域池化层（RoI pooling layer）
  - 多任务损失函数（Multi-task loss）

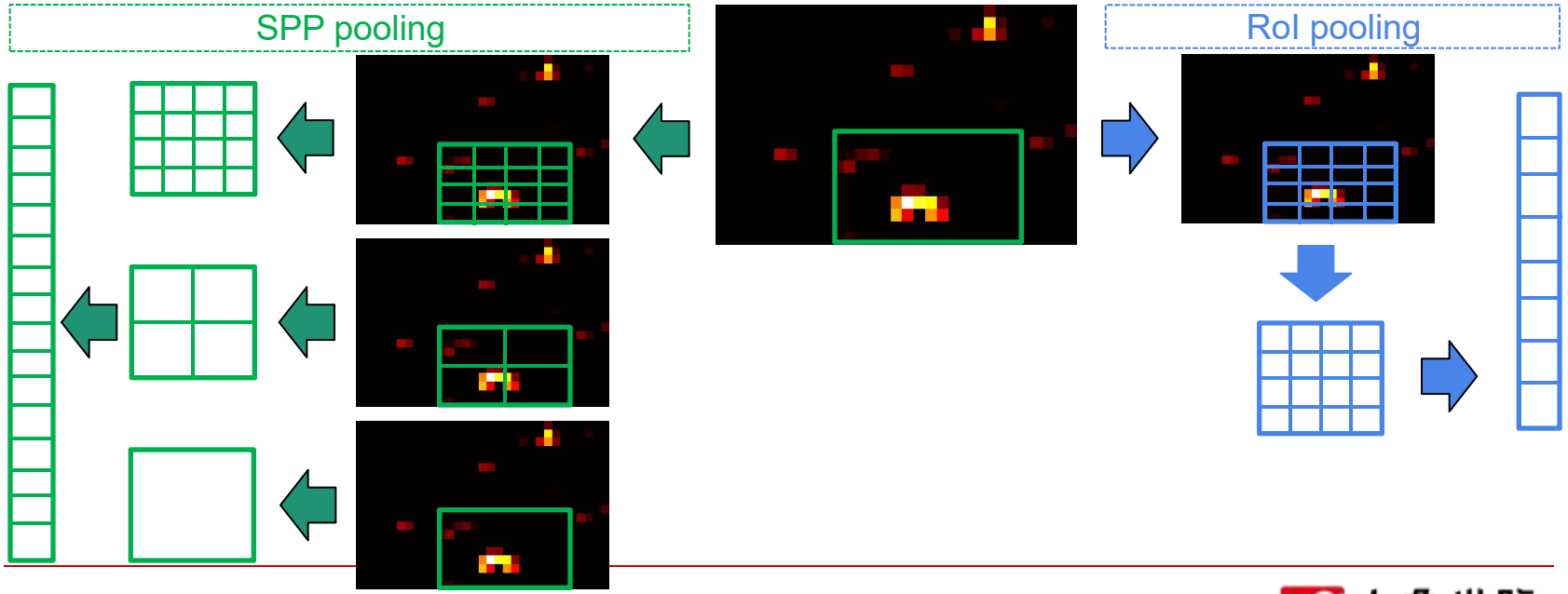


Fast R-CNN


Log loss + Smooth L1 loss
Softmax
分类器
Bounding box
回归模型
全连接层
End-to-end
全参数
fintune
SS区域
（RoIs）
共享
卷积计算
ConvNet
Train
Test
非极大值抑制（每类）
N类Bounding box回归
(N+1)-way Softmax
所有全连接层Fc6,7,8
RoI池化层
Conv5层特征图
整图卷积计算
Conv1-5
Selective
Search
提取区域
图片

Fast R-CNN

- 感兴趣区域池化（RoI pooling）
  - 空间金字塔池化（SPP pooling）的单层特例
  - 将RoI区域的卷积特征拆分成H x W网格（7x7 for VGG）
  - 每个Bin内的所有特征进行Max pooling



Fast R-CNN

- 感兴趣区域池化（RoI pooling）
  - 非重叠区域：类似Max pooling
  - 重叠区域
    - 多个区域的偏导之和

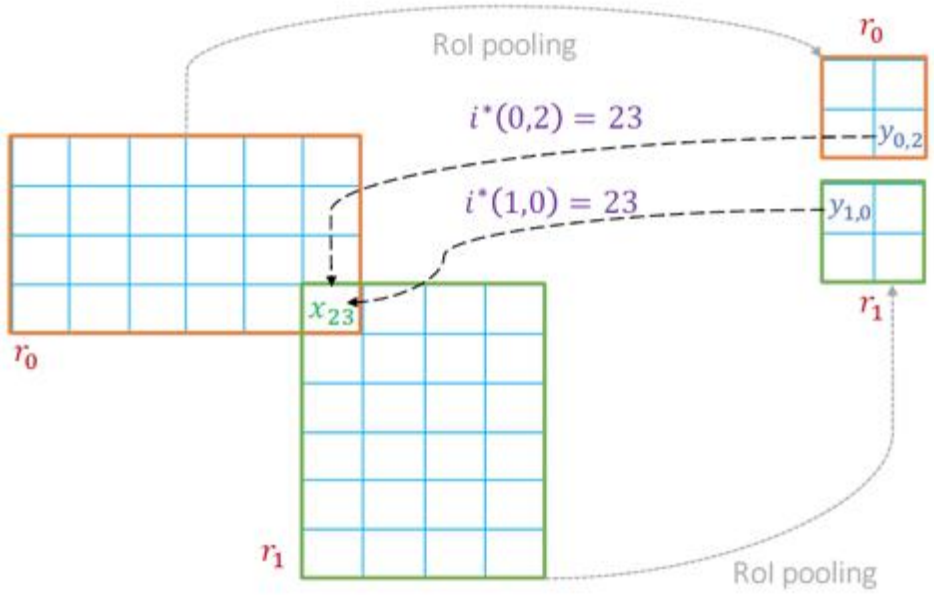
$$\frac{\partial L}{\partial x_i} = \sum_r \sum_j [i = i^*(r,j)] \frac{\partial L}{\partial y_{rj}}$$

1 if r, j "pooled" input i; 0 o/w

Partial for x_i

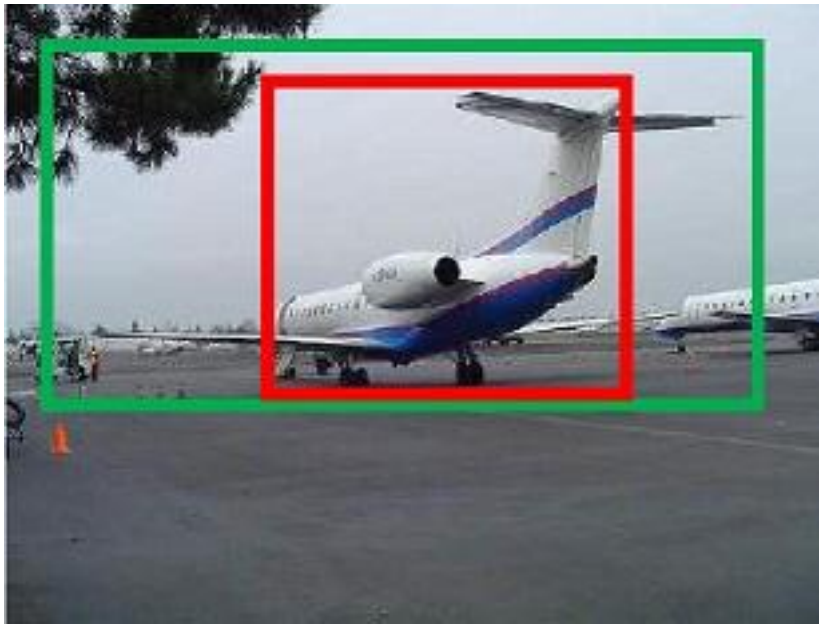
Over regions r, locations j

Partial from next layer



Fast R-CNN网络

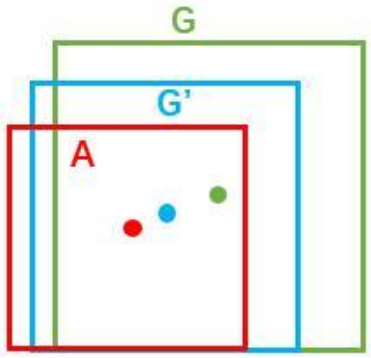
- 绿色框为飞机的Ground Truth(GT)，红色为提取的foreground anchors，那么即便红色的框被分类器识别为飞机，
- 但由于红色的框定位不准，这张图相当于没有正确的检测出飞机。



- 所以要采用一种方法对红色的框进行微调，使得foreground anchors和GT更加接近。

Fast R-CNN网络

- 寻找一种映射f，使得f(A_x, A_y, A_w, A_h)=(G'_x, G'_y, G'_w, G'_h)，其中(G'_x, G'_y, G'_w, G'_h)≈(G_x, G_y, G_w, G_h)
- 先做平移；后做缩放



$$G'_x = A_w \cdot d_x(A) + A_x \qquad G'_w = A_w \cdot \exp(d_w(A))$$
$$G'_y = A_h \cdot d_y(A) + A_y \qquad G'_h = A_h \cdot \exp(d_h(A))$$

- 需要学习的是$d_x(A)$，$d_y(A)$，$d_w(A)$，$d_h(A)$这四个变换，例如：

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

Fast R-CNN

Fast R-CNN网络

- 多任务损失（Multi-task loss）

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

  - 分类器loss： $L_{\text{cls}}(p, u) = -\log p_u$
    - 每个RoI的概率分布 $p = (p_0, \ldots, p_K)$
    - Ground truth类别 u
  - Bounding box回归L1 loss: $L_{\text{loc}}(t^u, v) = \sum_{i \in \{x,y,w,h\}} \text{smooth}_{L_1}(t_i^u - v_i)$
    - 每个RoI共有N个loss（per-class）
    - 偏差目标 $v = (v_x, v_y, v_w, v_h)$
    - 预测偏差 $t^u = (t_x^u, t_y^u, t_w^u, t_h^u)$
    - 指示函数 $[u \geq 1]$
      - 物体类别：1，有回归loss
      - 背景类别：0，没有回归loss

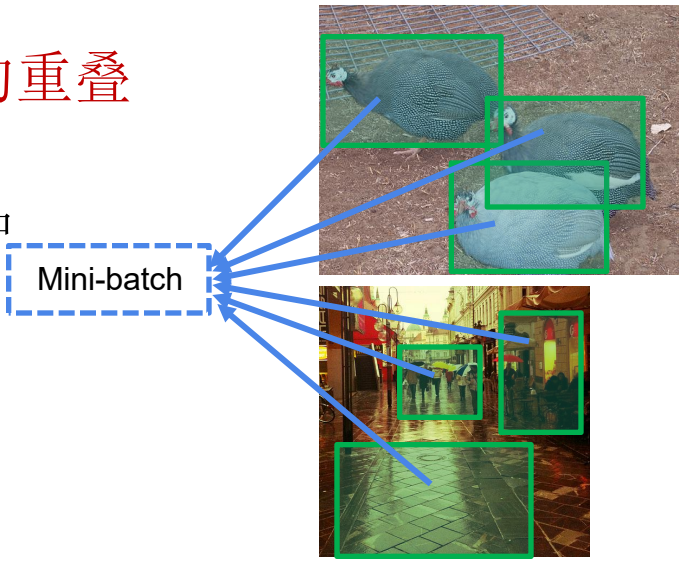
高斯（0,1）归一化

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

$$\begin{aligned} t_x &= (G_x - P_x)/P_w \\ t_y &= (G_y - P_y)/P_h \\ t_w &= \log(G_w/P_w) \\ t_h &= \log(G_h/P_h). \end{aligned}$$

Fast R-CNN

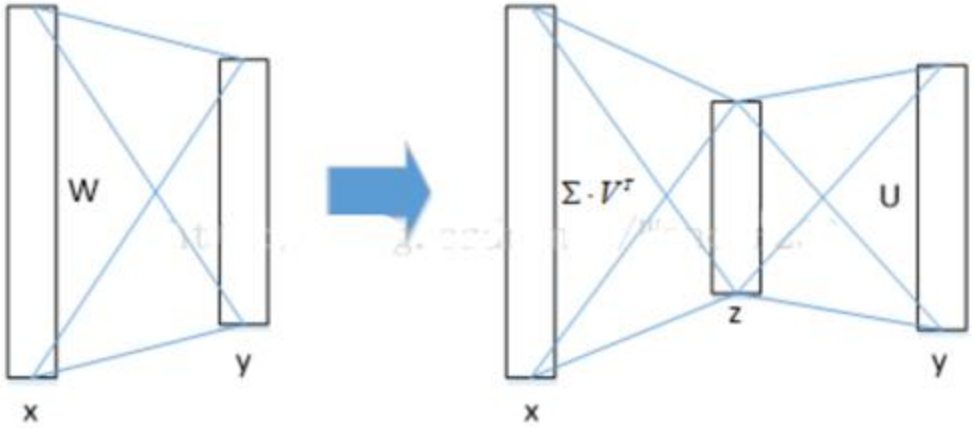
- 训练：在pre-trained模型上做finetune
- Mini-batch sampling抽样
  - 分级抽样法（ Hierarchical sampling）
    - Batch尺寸(128) = 每个batch的图片数量(2) x 每个图片的RoI数量(64)
    - 充分利用卷积层的共享计算
  - RoI分类基于与Ground truth的重叠
    - 物体占25%：IoU >= 0.5
    - 背景占75%：IoU在[0.1, 0.5)中
- 初始模型VGG16



Fast R-CNN

- 在目标检测任务中，要处理的RoI数量比较多，几乎一半的前向计算时间被用于全连接层。
- 就Fast R-CNN而言，RoI池化层后的全连接层需要进行约2k次，因此在Fast R-CNN中可以采用SVD分解加速全连接层计算；
- 全连接层输入数据为X，输出数据为Y，全连接层权值矩阵为W，尺寸为 u × v ，那么该层全连接计算为：Y = W × X。

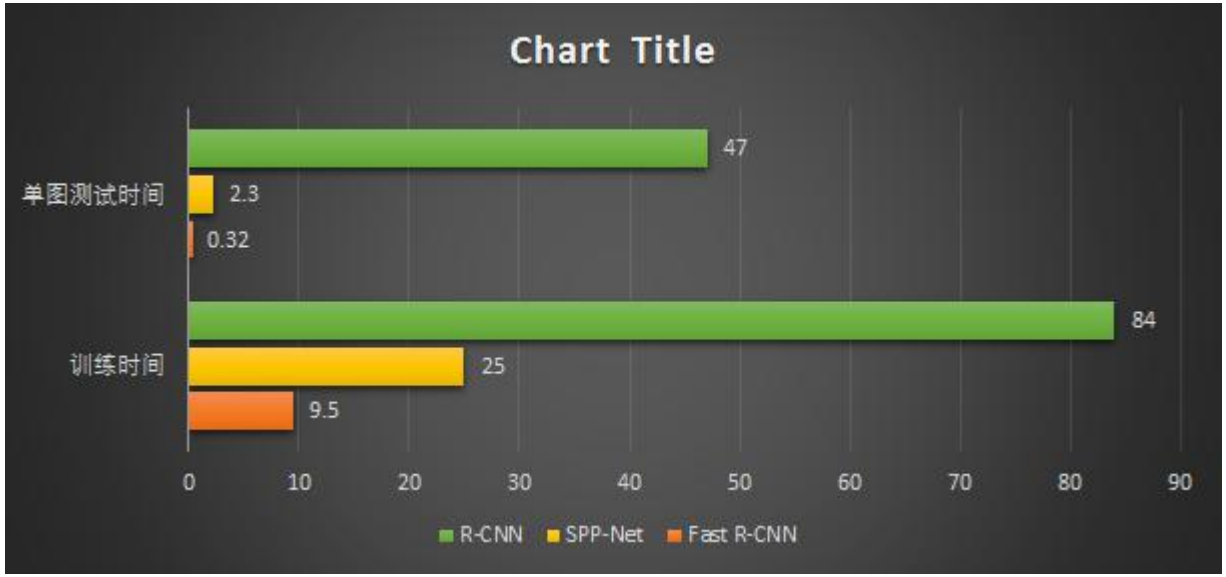
$$W \approx U\Sigma_t V^T$$



Fast R-CNN

Fast R-CNN性能提升

	Fast R-CNN	SPP-Net	R-CNN
训练时间	9.5（8.8x）	25（3.4x）	84
单图测试时间	0.32s（146x）	2.3s（20x）	47.0s
mAP	66.9%	63.1%	66.0%



Fast R-CNN

Fast R-CNN准确性提升

- 端对端（End-to-end）训练

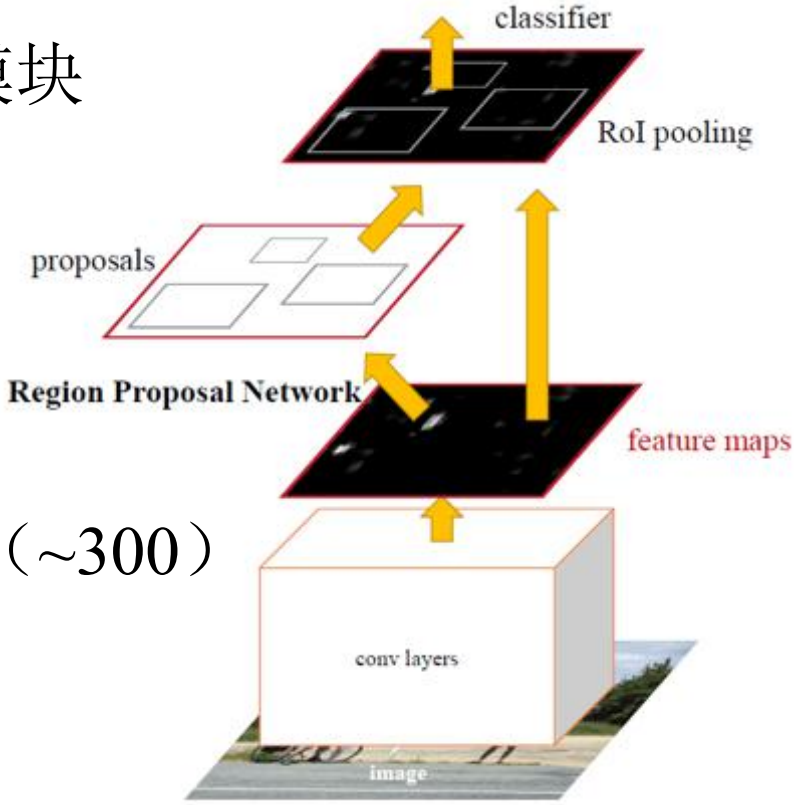
Finetune层数	Fc6层之后	Conv3层之后	Conv2层之后
VOC 2007 mAP	61.4%	66.9%	67.2%
单图测试时间	0.32s	0.32s	0.32s

- 多任务（Multi-task）训练

训练类型	Fast R-CNN （VGG16）			
多阶段训练			Y	
Mutil-task训练		Y		Y
测试阶段Bbox回归			Y	Y
VOC 2007 mAP	62.6%	63.4%	64.0%	66.9%

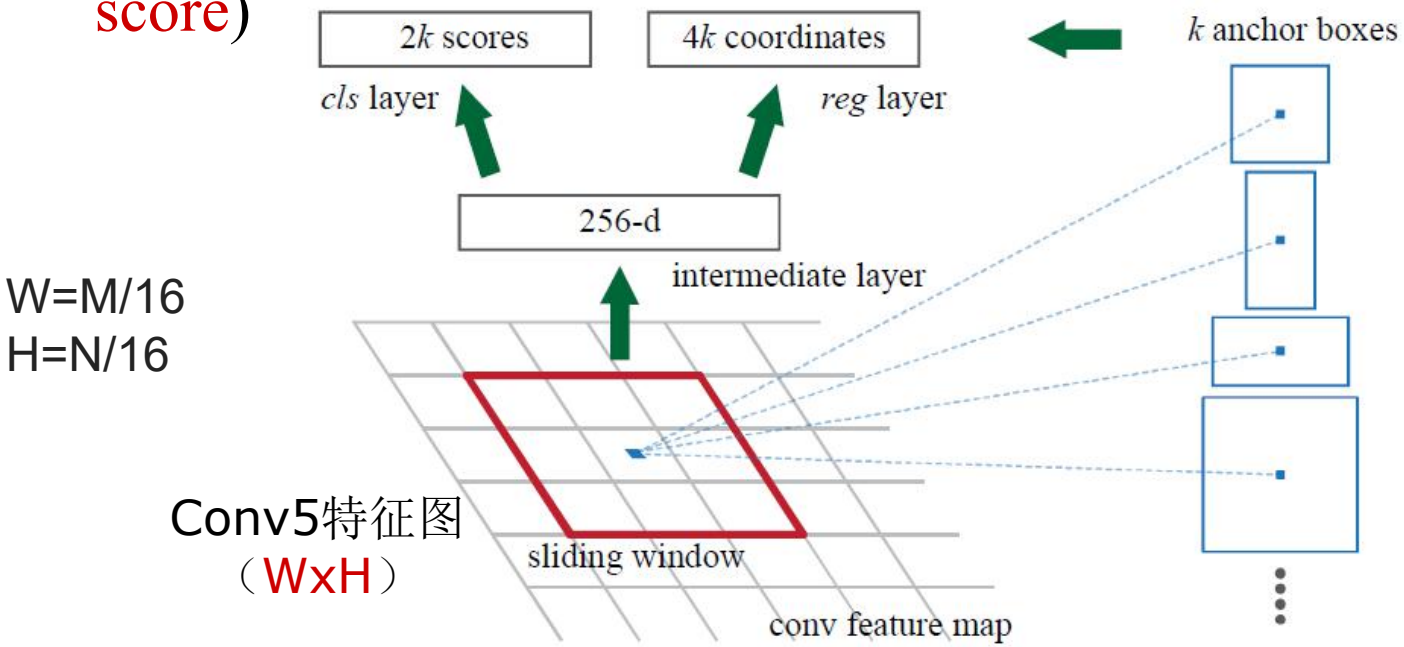
Faster R-CNN

- 集成Region Proposal Network（RPN）网络
  - Faster R-CNN = Fast R-CNN + RPN
  - 取代离线Selective Search模块
    - 解决性能瓶颈
  - 进一步共享卷积层计算
  - 基于Attention注意机制
    - 引导Fast R-CNN关注区域
  - Region proposals量少质优（~300）
    - 高precision，高recall



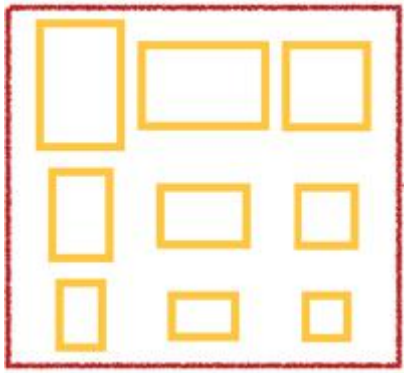
Faster R-CNN网络

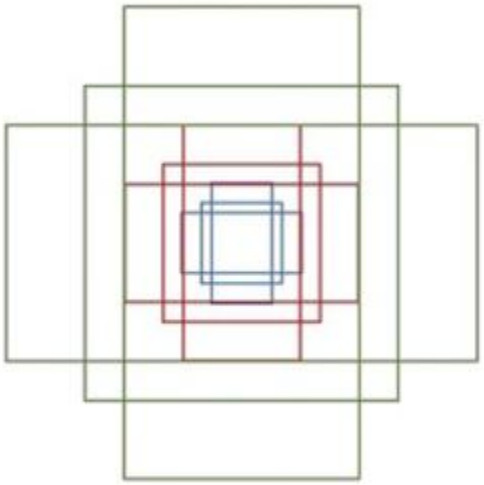
- Region Proposal Network（RPN）网络 全卷积网络
  - 3x3，256-d卷积层 + ReLU←输入图片的Conv5特征
  - 1x1，4k-d卷积层→输出k组proposal的offsets（r, c, w, h）
  - 1x1，2k-d卷积层→输出k组（object score, non-object score)



Faster R-CNN网络

- Anchor box类型k=9
  - 3个尺度scale（128, 256, 512）
  - 3个宽高比ratio（1:1, 1:2, 2:1）
- Anchor总数量W×H×k
  - conv5特征图每个点上有k个anchor







Faster R-CNN网络

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

- RPN网络的loss
  - 分类 L_{cls} 覆盖2类：object & non-object
  - 回归 $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$ 使用smooth L1
    - x, x_a, x^* 分别是对应预测框、anchor框、ground truth框
    - y, w, h类似
  - 指示函数 p_i^*
  - 训练样本mini-batch
    - 单个图片
    - 128个正样本：IoU > 0.7的anchor框 （或最大IoU）
    - 128个负样本：IoU < 0.3的anchor框

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

Faster R-CNN网络

- 4步训练流程
  - Step1 – 训练RPN网络
    - 卷积层初始化←ImageNet上pretrained模型参数
  - Step2 - 训练Fast R-CNN网络
    - 卷积层初始化←ImageNet上pretrained模型参数
    - Region proposals由Step1的RPN生成
  - Step3 - 调优RPN
    - 卷积层初始化←Fast R-CNN的卷积层参数
    - 固定卷积层，finetune剩余层
  - Step4 - 调优Fast R-CNN
    - 固定卷积层，finetune剩余层
    - Region proposals由Step3的RPN生成

卷积层无共享

卷积层共享

Faster R-CNN网络

Faster R-CNN性能提升

	Faster R-CNN	Fast R-CNN	R-CNN
单图测试时间	0.198s	2.0s	50.0s
PASCAL VOC 07 mAP	66.9%	66.9%	66.0%

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**

