

R-CNN

原始论文题目：

Rich feature hierarchies for accurate object detection and semantic segmentation

R-CNN简介

- R-CNN: Regions with CNN features
- 基于神经网络的多物体检测算法

图像分类 VS 物体识别



图像分类 VS 物体识别

- 图像分类：输入图片，输出图片是什么
- 物体识别：输入图片，输出图片里所有对象的位置，以及这些对象分别是什么
- 物体识别包含图像分类，并多了寻找物体位置的过程

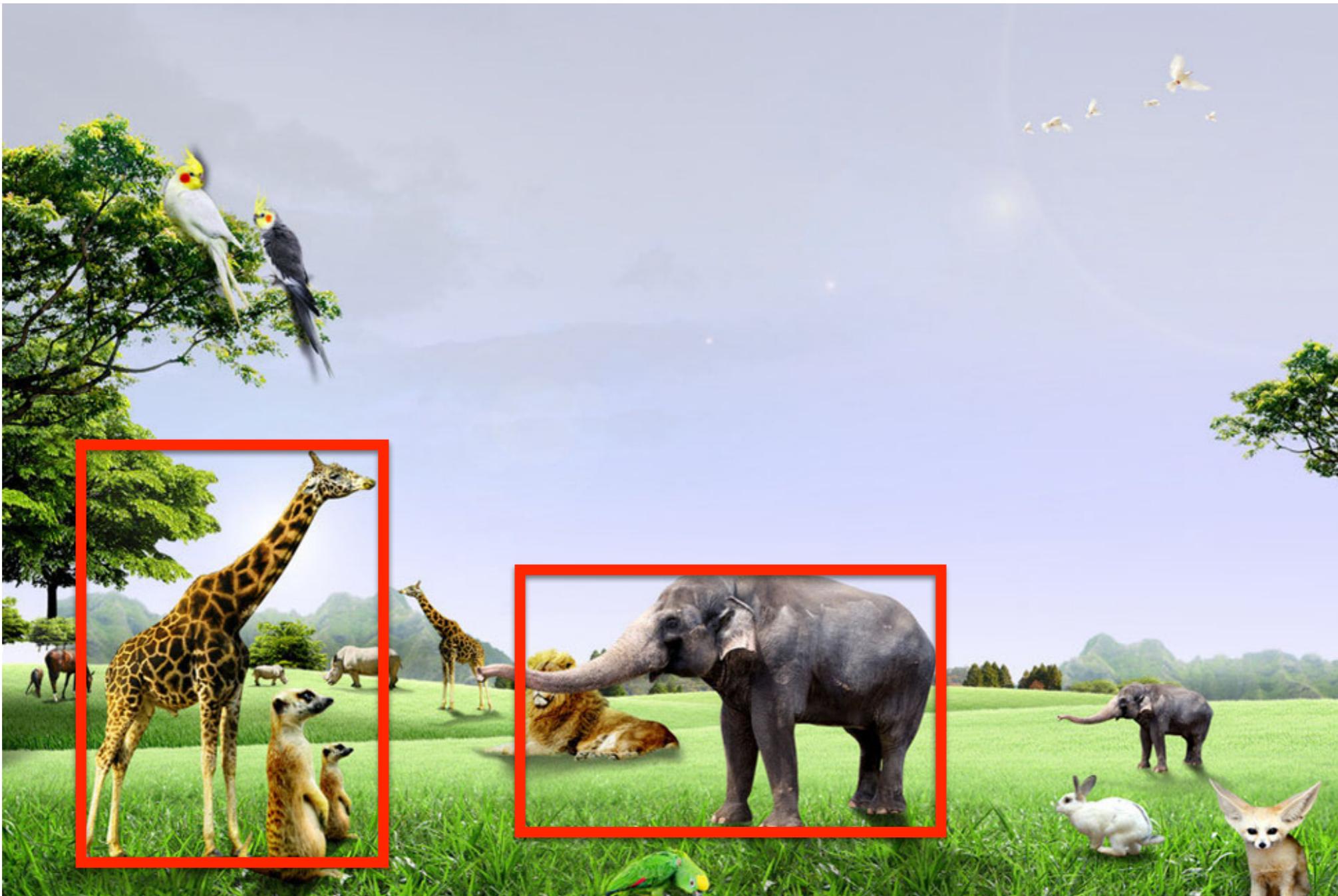
先介绍一些概念..

- Ground-truth box
- IOU值

Ground-truth box

- Ground-truth: 所谓的“正确答案”
- 在物体识别中，'ground-truth box'指用于物体识别训练的人工标记
- 每个标记包含一个确定位置与宽高的矩形框，和这个矩形框内物体的类别

Ground-truth box



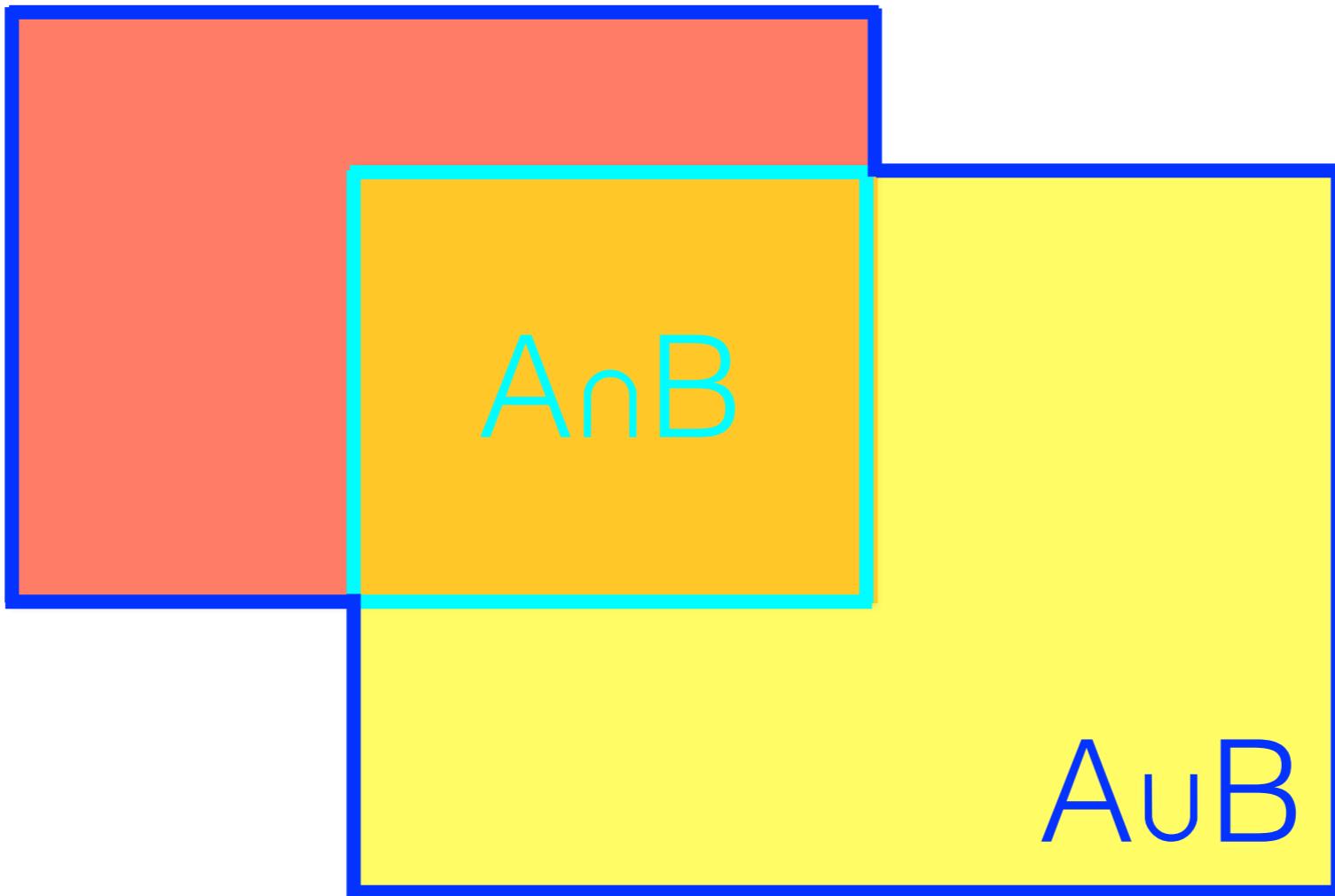
长颈鹿

大象

IOU值

- IOU - Intersection Over Union
- IOU值等于两个矩形框**交集部分**的面积除以**并集部分**的面积，反应两个矩形框的**重合度**

IOU值



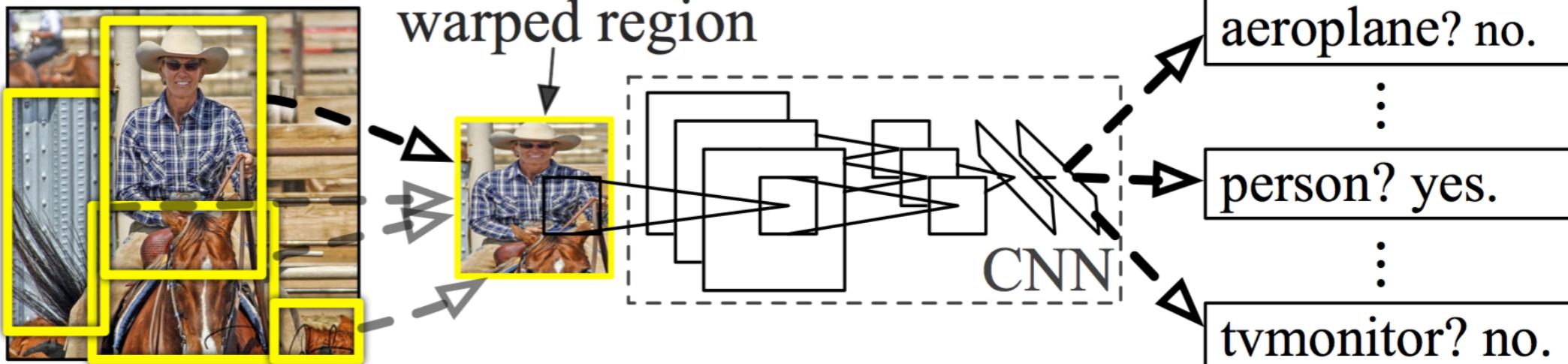
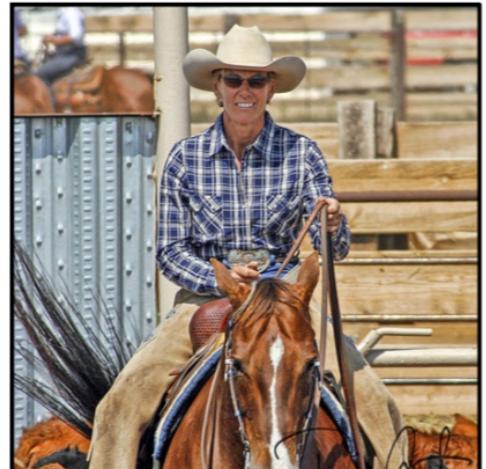
$$\text{IOU} = \frac{A \cap B}{A \cup B}$$

R-CNN算法流程

- 1. 以单张图片作为输入，生成可能的物体候选框 (Region Proposal)
- 2. 以每个物体候选框作为输入，使用CNN计算出特征向量 (Feature Extraction)
- 3. 对于每个特定的类别，以CNN计算的特征向量为输入，使用SVM分类器判断候选框内的物体是否属于这一类别 (Classification)，并对所有属于此类别的候选框进行进一步筛选(使用**non-maximum suppression**方法，后面会讲)，筛选后的候选框即为R-CNN的最终输出

R-CNN算法流程

R-CNN: *Regions with CNN features*



1. Input
image

2. Extract region
proposals (~2k)

3. Compute
CNN features

4. Classify
regions

候选框生成 (Region Proposal)

- 这一步有很多经典方法
- 文中使用的是Selective Search方法，对于任意输入图像，可生成2000个不同的候选框
- 这一步并没有包含神经网络，不必关注其细节

提取特征向量 (Feature Extraction)

- 这一步使用CNN， 输入是由候选框转化而来的224*224的RGB图像， 输出是一个4096维的特征向量
- 所有输入的候选框共享同一个**CNN**
- 由于候选框形状各异， 我们需要将其转化为224*224的图像

不同的转化方法



(A)

(B)

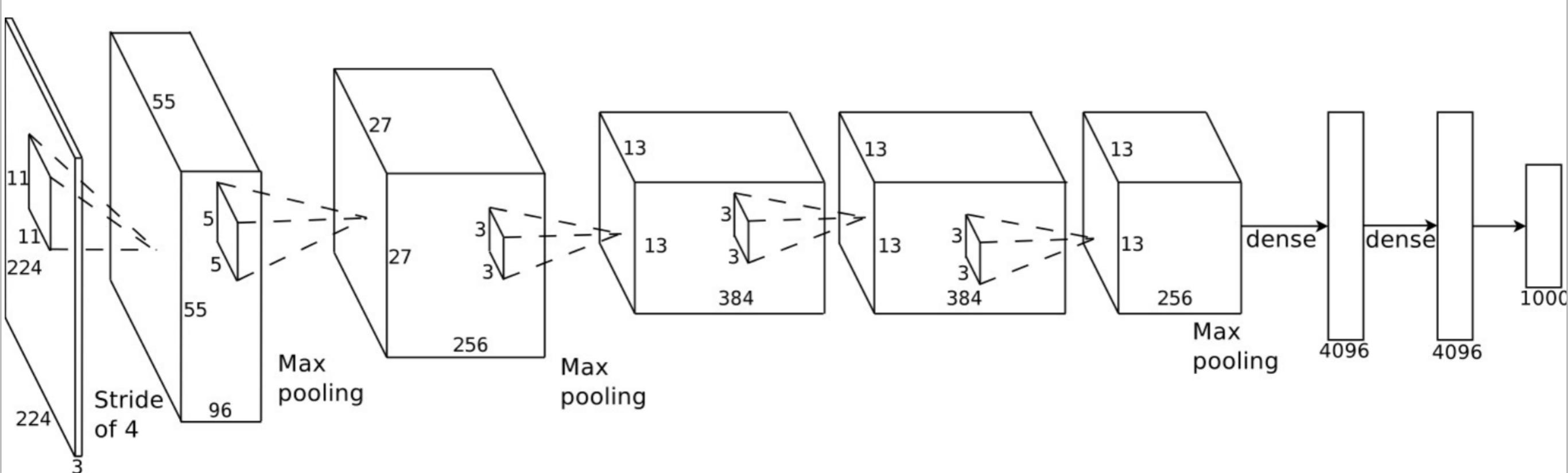
(C)

(D)

不同的转化方法

- A: 原始候选框
- B: 沿短边双向扩充为正方形（引入原图背景）
- C: 沿短边双向扩充为正方形（用灰色背景填充空白区）
- D: 沿短边等比例拉伸（warp）
- 第二行相对第一行有向四周小幅度拓展（padding）
- 文中采用方案：**warp + 16-pixel padding**

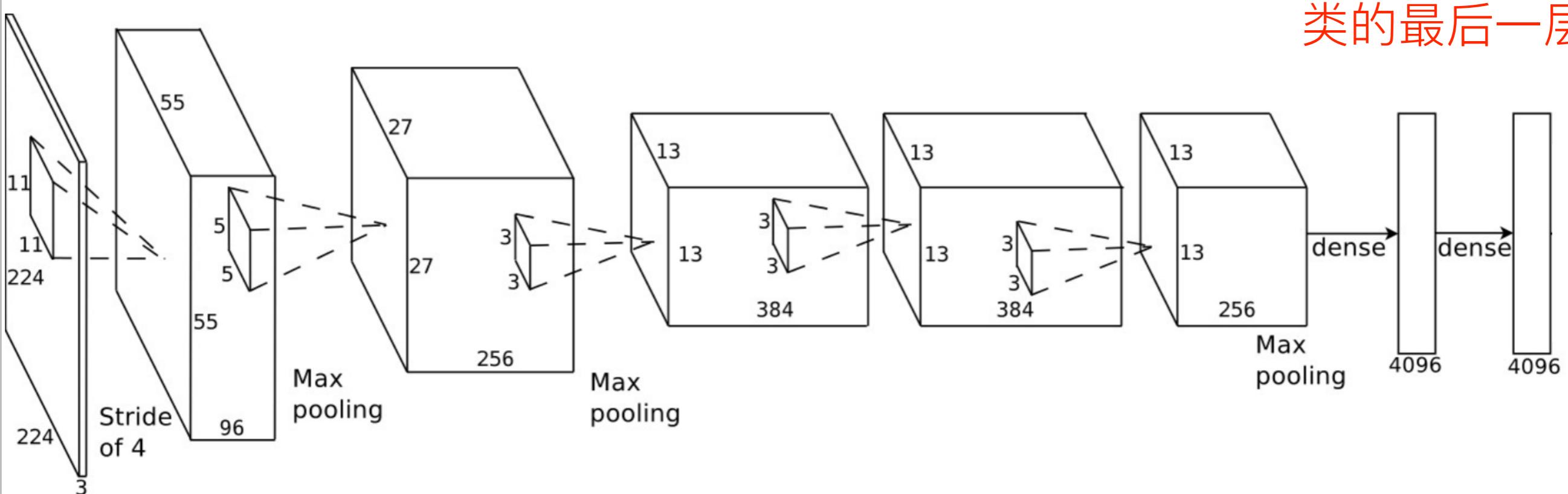
神经网络结构 (第一种)



Krizhevsky's net
(具体参考[2])

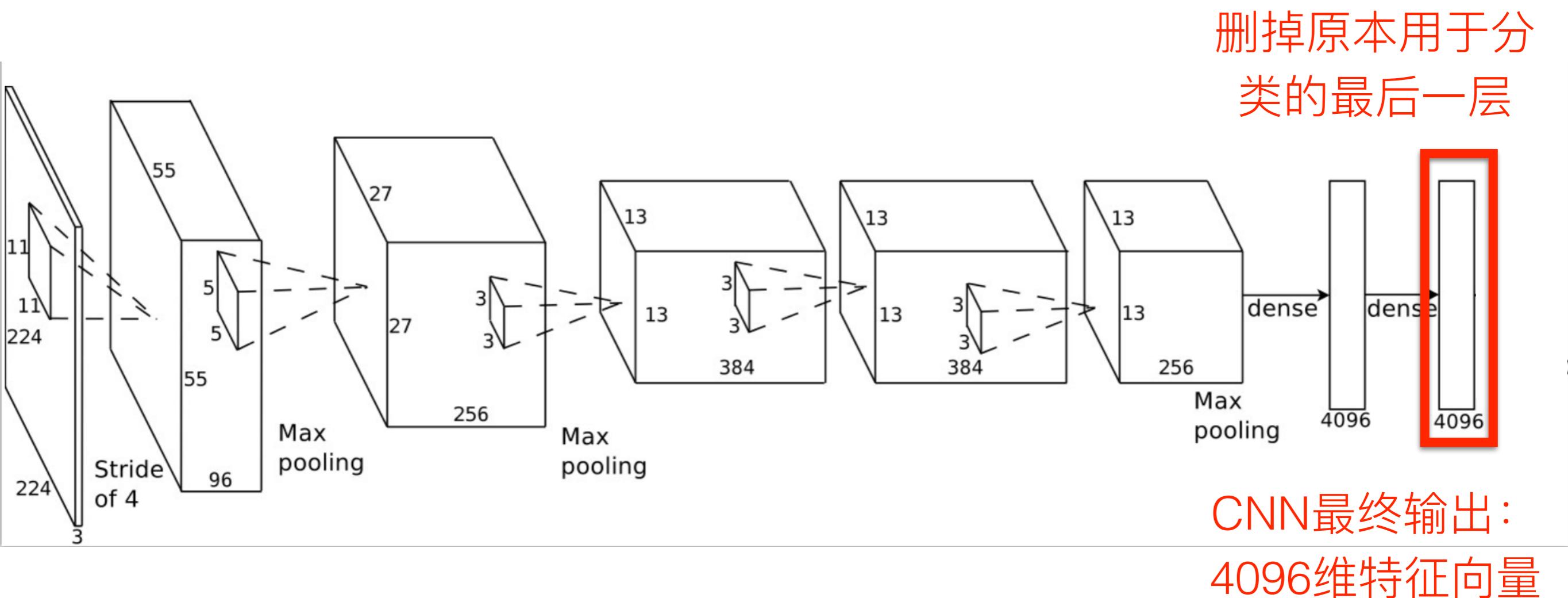
神经网络结构 (第一种)

删掉原本用于分
类的最后一层



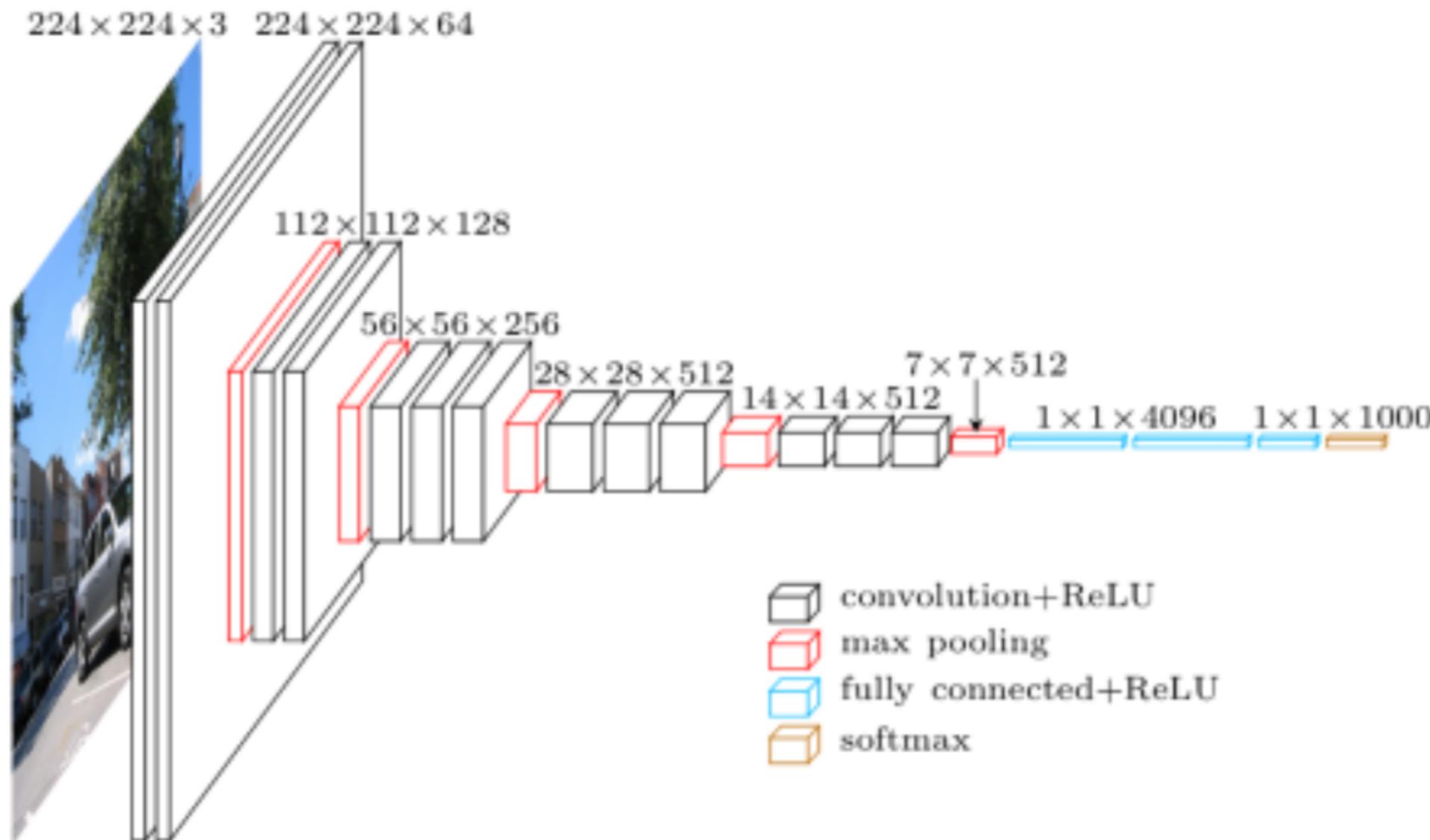
Krizhevsky's net

神经网络结构 (第一种)



Krizhevsky's net

神经网络结构 (第二种)



VGG-16 net

神经网络结构

- Krizhevsky: 5个卷积层， 2个全连接层（除最后一层，下同）
- VGG-16: 13个卷积层， 2个全连接层
- VGG-16准确率更高，但是训练和检测所需的时间更长

SVM分类器

- SVM - Support Vector Machine
- 一个**特定类别**的SVM分类器以CNN计算得来的4096维特征向量作为输入，来判断此特征向量代表的图片区域是否属于这一特定类别
- R-CNN算法通过训练多个SVM分类器来实现对候选框的分类
- 目前无需了解SVM的具体细节

对候选框的筛选

- 由于候选框有很多，最终同一个物体可能被多个候选框选中，所以我们需要选出其中最有代表性的候选框
- 这里我们使用**非极大值抑制方法(Non-maximum suppression)**

非极大值抑制 (Non-maximum suppression)

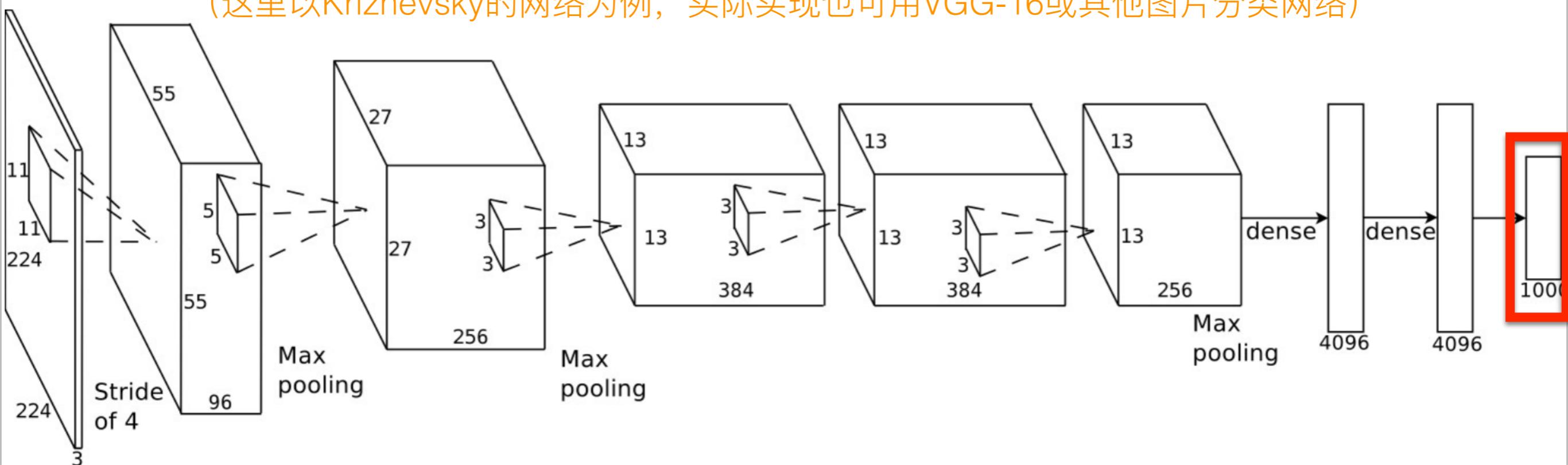
1. 对于一个特定类别，将属于此类别所有候选框按照SVM的得分从大到小排列，得到一个序列M
2. 取出M中第一个元素(即得分最高的元素)，放入序列N(初始为空)
3. 对于M中每一个元素，计算其与N中所有元素的IOU值并取最大值，若最大值超过某一预设阈值，则直接删除此元素
4. 若序列M不为空，则重复步骤2-3
5. 序列N中的元素即为此类别的最终输出

如何训练CNN?

- 第一阶段：有监督预训练 (Supervised pre-training)
- 第二阶段：特定域微调 (Domain-specific fine tuning)
- 第三阶段：训练SVM分类器

第一阶段：有监督预训练 (Supervised pre-training)

(这里以Krizhevsky的网络为例，实际实现也可用VGG-16或其他图片分类网络)



这一阶段保留原始的
最后一层 (分类层)

第一阶段：有监督预训练 (Supervised pre-training)

- 使用纯图片分类的大数据集来训练CNN，使其具有**图片分类**的能力
- 训练完成后，**将最后一层删去**，保留剩余的网络结构和参数到下一阶段

第二阶段：特定域微调

(Domain-specific fine tuning)

- 这一阶段我们训练CNN(已经预训练好)判断候选框类别的能力，输入为候选框(图片中的特定域)，输出为此候选框所属的类别(某种物体或背景)
- 正负样本的定义：对于特定图片里的特定候选框，计算它与图中**所有类别物体的所有ground-truth**的IOU值，取其中最大值并记下对应的物体类别。如果这个值超过某一设定的阈值(threshold，文中为0.5)，则此候选框为**这个类别的正样本**(某种物体)，反之则为**所有类别的负样本**(背景)

第三阶段：训练SVM分类器 (SVM Classifier)

- 这一阶段我们训练多个SVM分类器，每个SVM分类器用于判断输入是否属于某一特定类别
- 对于特定的SVM分类器：输入是CNN计算出的特征向量，输出是此候选框内的物体是否属于这一类别
- 正负样本的定义(与第二阶段不同)：对于一个特定类别的SVM，正样本为**所有此类别的ground-truth**，负样本为与ground-truth的IOU值低于0.3的所有候选框。
- 对于IOU值高于0.3但不是ground-truth的候选框，我们忽略不计，也即它们**既不是正样本，也不是负样本**

进一步改进

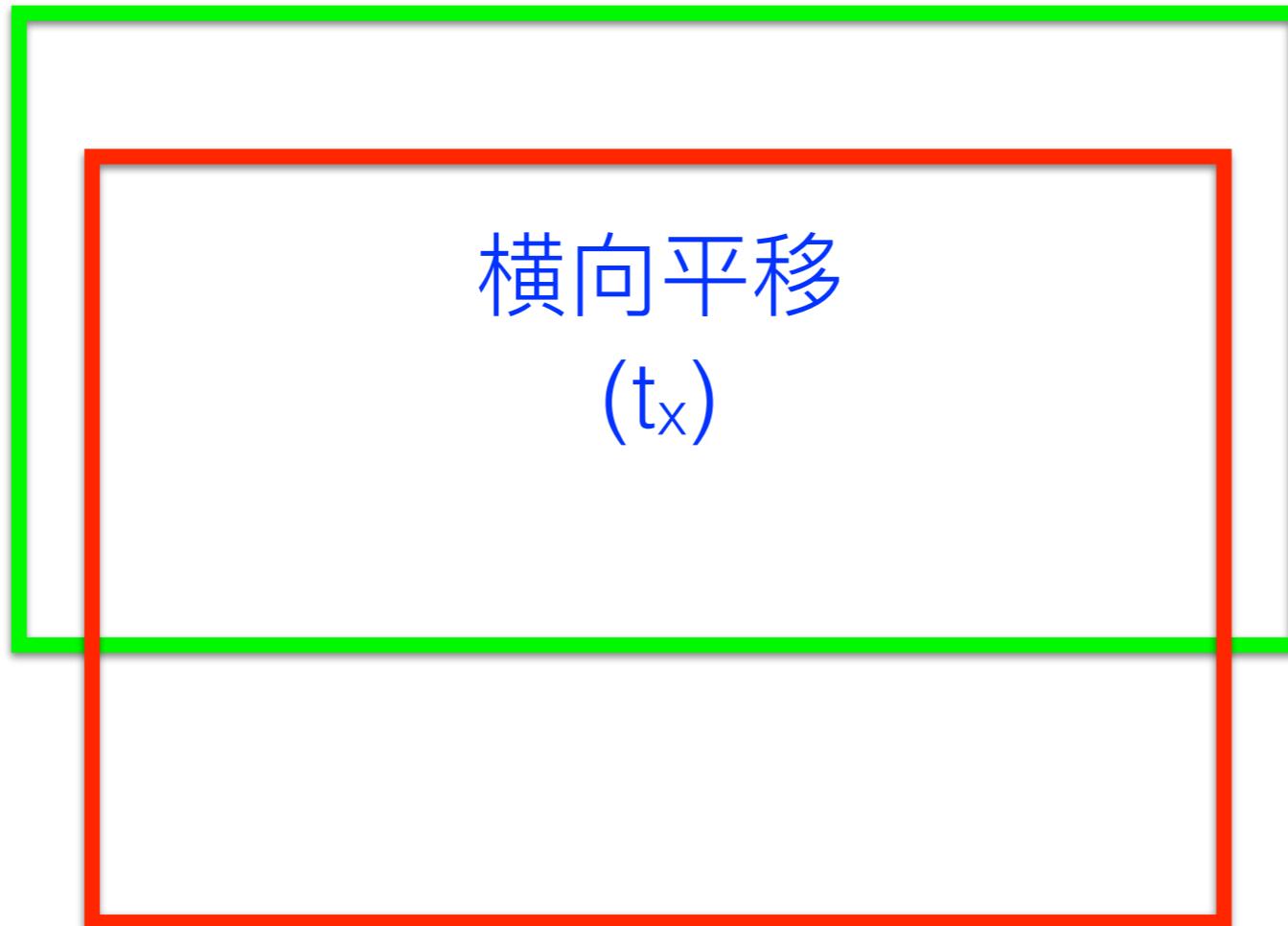
- 经测试证明， 测试中的false-positive主要源于候选框定位不准确， 需要提高候选框定位精度
- 这里我们用一个Bounding-box regressor来对候选框的位置进行精修

Bounding-box Regressor

- 对于每一个候选框， 我们将其映射到与之IOU值最大的ground-truth box
- B-Box Regressor最终输出四个值：横向平移距离，纵向平移距离，横向缩放比例，纵向缩放比例
- 四个值记为 t_x, t_y, t_w, t_h



红色 - 候选矩形框
绿色 - 正确矩形框



红色 - 候选矩形框

绿色 - 正确矩形框



红色 - 候选矩形框

绿色 - 正确矩形框



红色 - 候选矩形框

绿色 - 正确矩形框



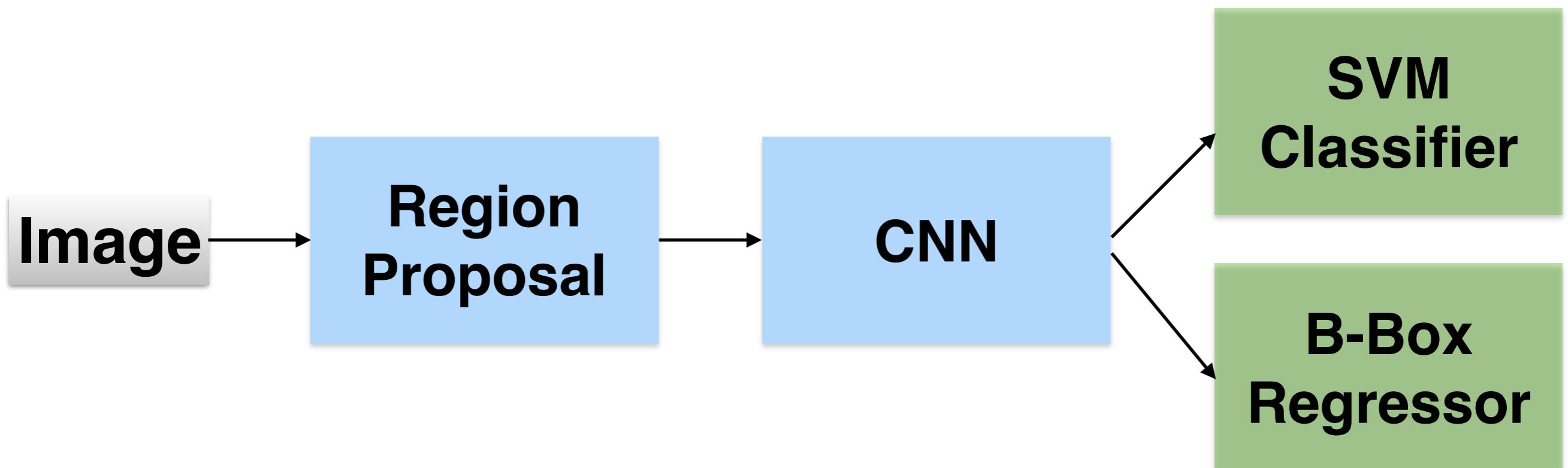
纵向缩放
(t_h)

红色 - 候选矩形框
绿色 - 正确矩形框

Bounding-box Regressor

- 加入B-Box Regressor后对原流程的修改：Selective-Search方法生成候选框之后，先使用B-Box Regressor生成每个候选框的修正参数，并对候选框进行修正，再将修正后的候选框输入到CNN中进行分类和筛选
- 具体细节请参考[1]的附录C

R-CNN 算法流程(完整)



参考文献

- (1) Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition* (pp.580-587). IEEE Computer Society.
- (2) Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *International Conference on Neural Information Processing Systems* (Vol.25, pp.1097-1105). Curran Associates Inc.
- (3) 基于R-CNN的物体检测 - <http://blog.csdn.net/hjimce/article/details/50187029>