人工智能导论

李文

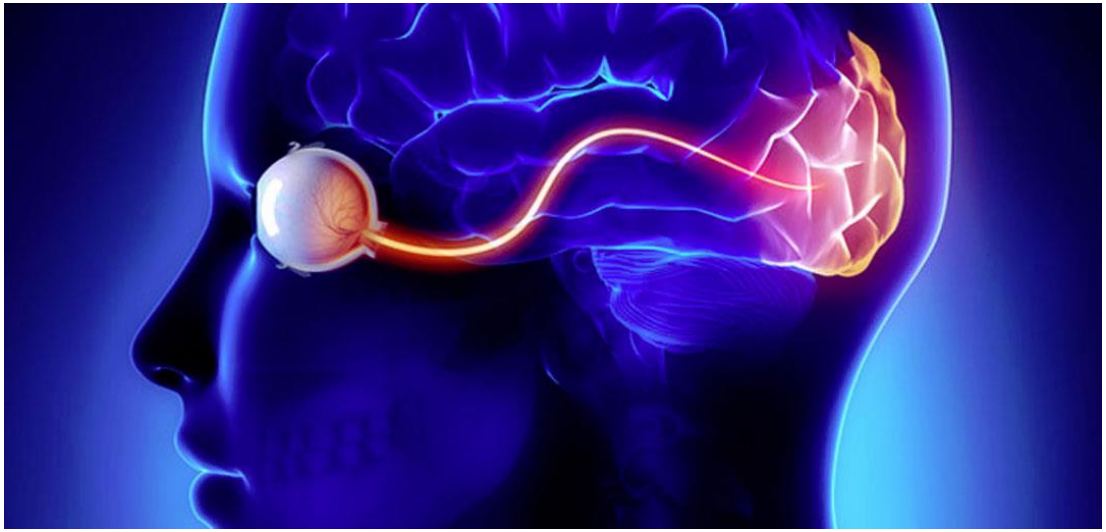
计算机科学与工程学院

数据智能研究组（Data Intelligence Group）：

http://dig.uestc.cn/

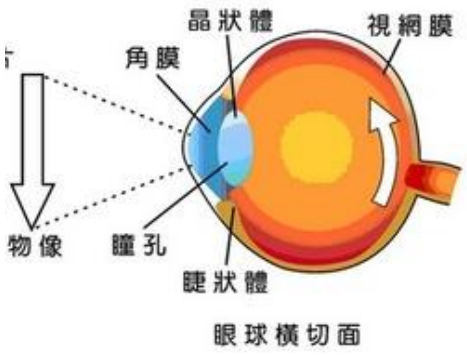
计算机视觉

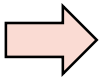
- 给机器以“看”的能力
- 眼睛是人类最重要的传感器，超过一半的大脑都会参与视觉功能

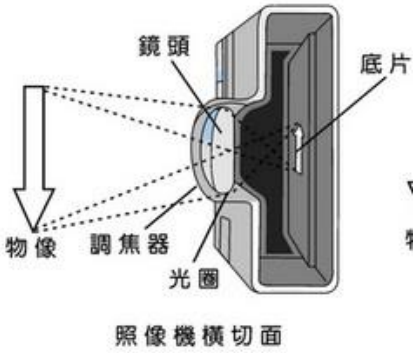


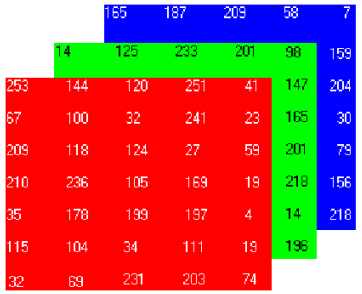
计算机视觉

■成像





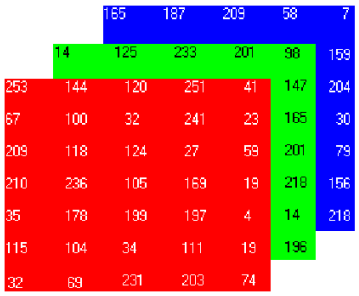




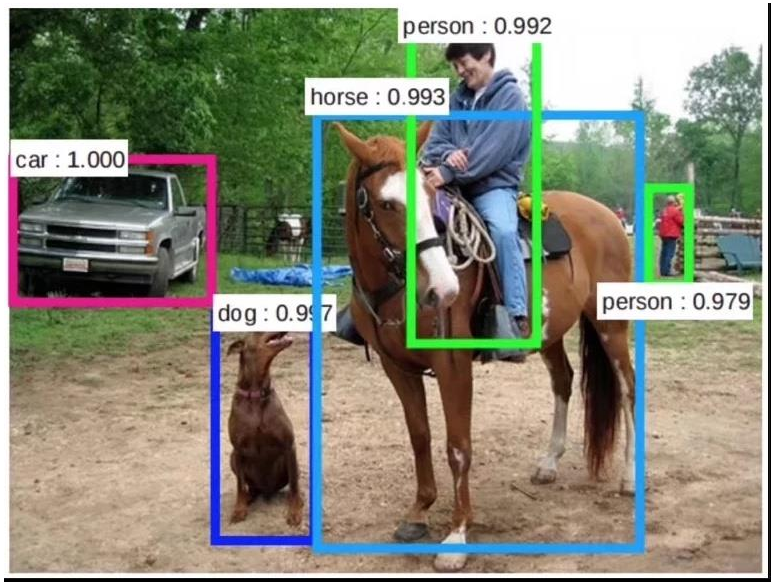
眼睛（人类）　　相机（机器）　　数字图像

计算机视觉

- 从数字化信息中提取信息、识别理解。

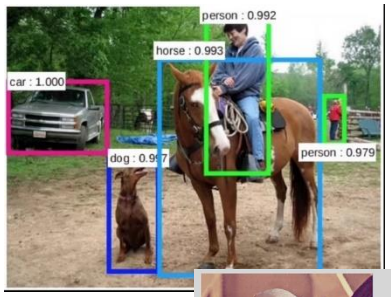


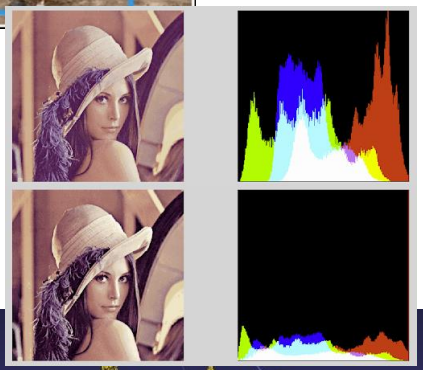
数字图像

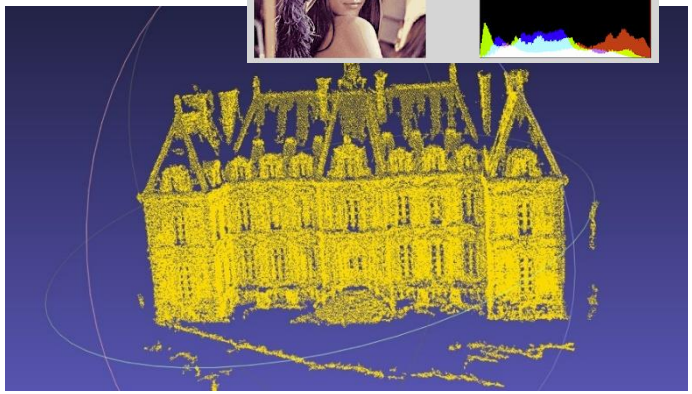


计算机视觉：主要领域

- 计机成像学
  - 图像处理等底层视觉，如图像去噪、图像超分辨、图像增强、风格变换
- 图像理解
  - 语义理解等高层视觉，如图像分类、目标检测、人脸识别、语义分割
- 三维视觉
- 视频识别

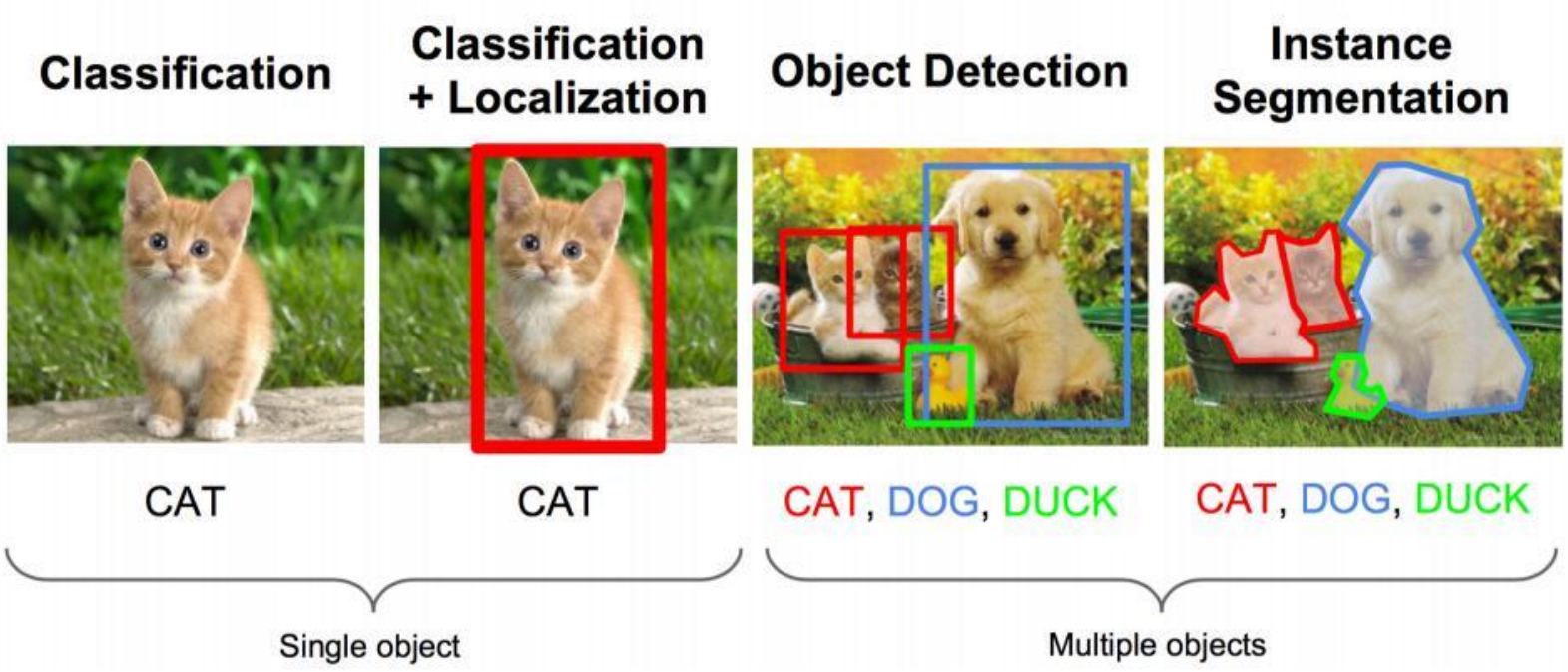






01 图像理解：传统与前沿

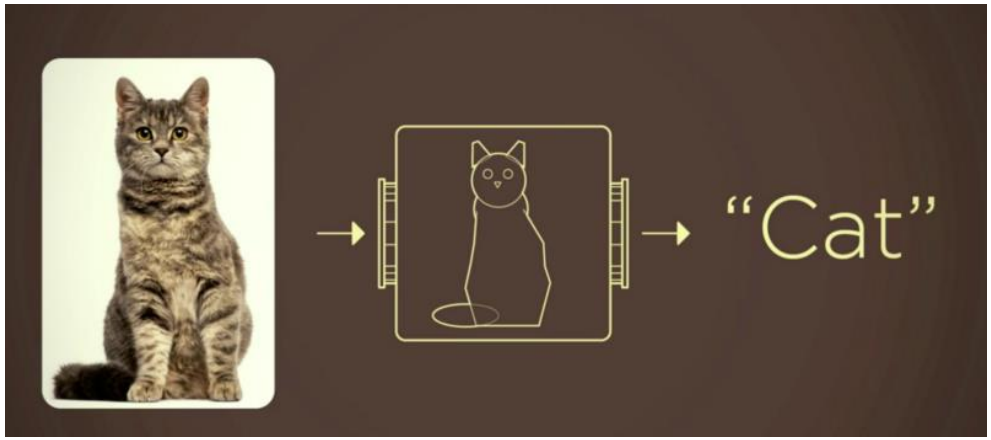
图像理解：传统与前沿


Classification
Classification + Localization
Object Detection
Instance Segmentation
CAT
CAT
CAT, DOG, DUCK
CAT, DOG, DUCK
Single object
Multiple objects

图像分类

图像分类

- 图像分类是根据图像的语义信息将不同类别图像区分开来，是计算机视觉中重要的基本问题，也是图像检测、图像分割、物体跟踪、行为分析等其他高层视觉任务的基础。

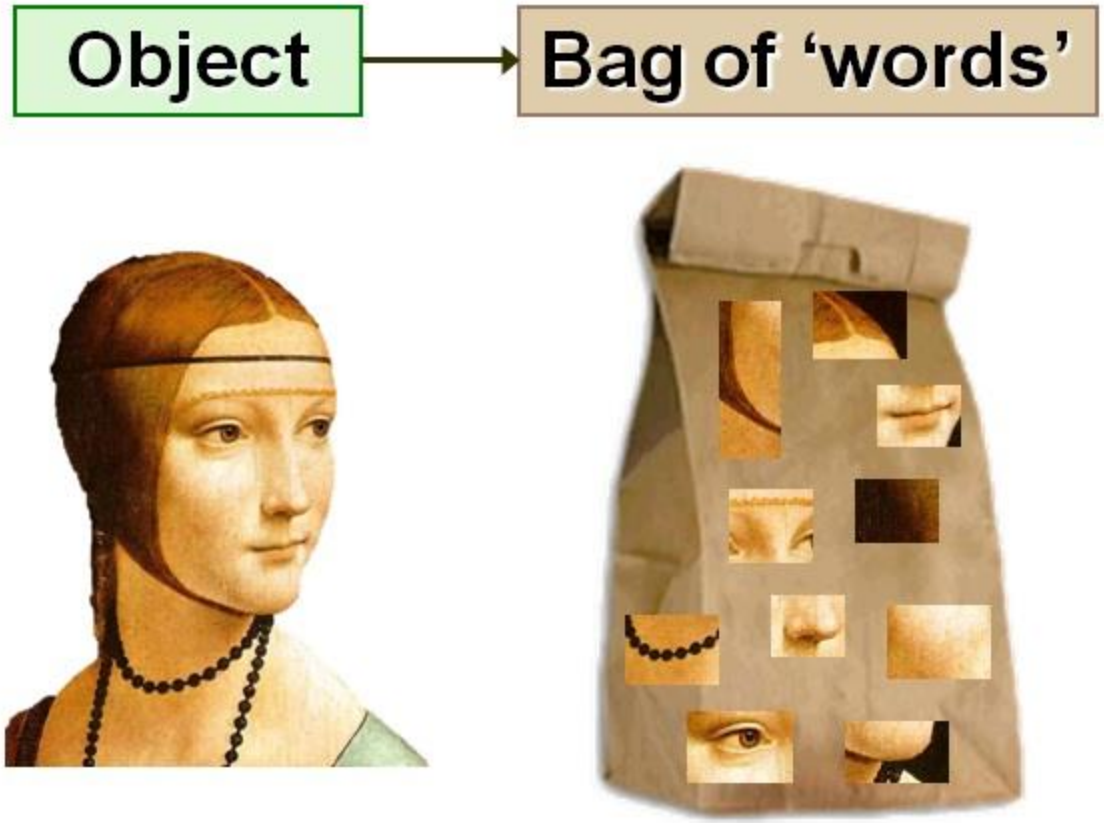


图像分类

- 传统方法：Bag-of-Words模型
  - 特征点提取

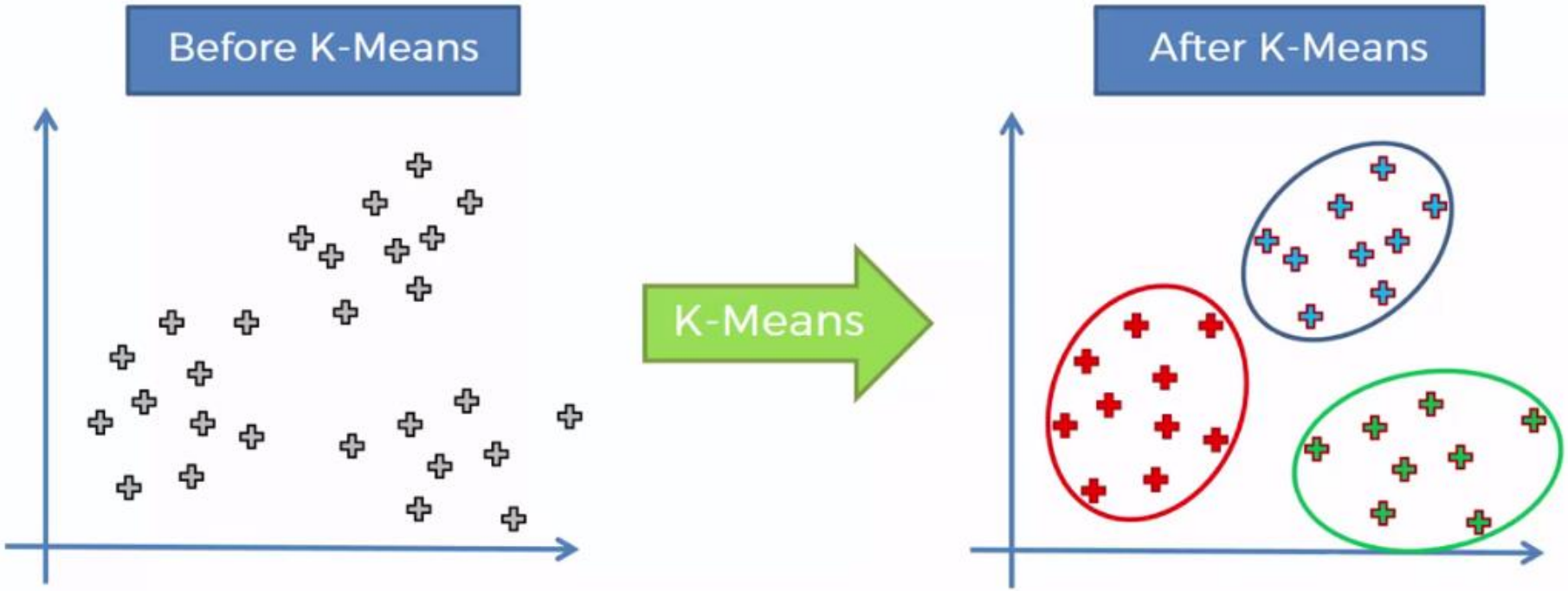
图像分类

- 传统方法：Bag-of-Words模型
  - 每个特征点：128维的向量



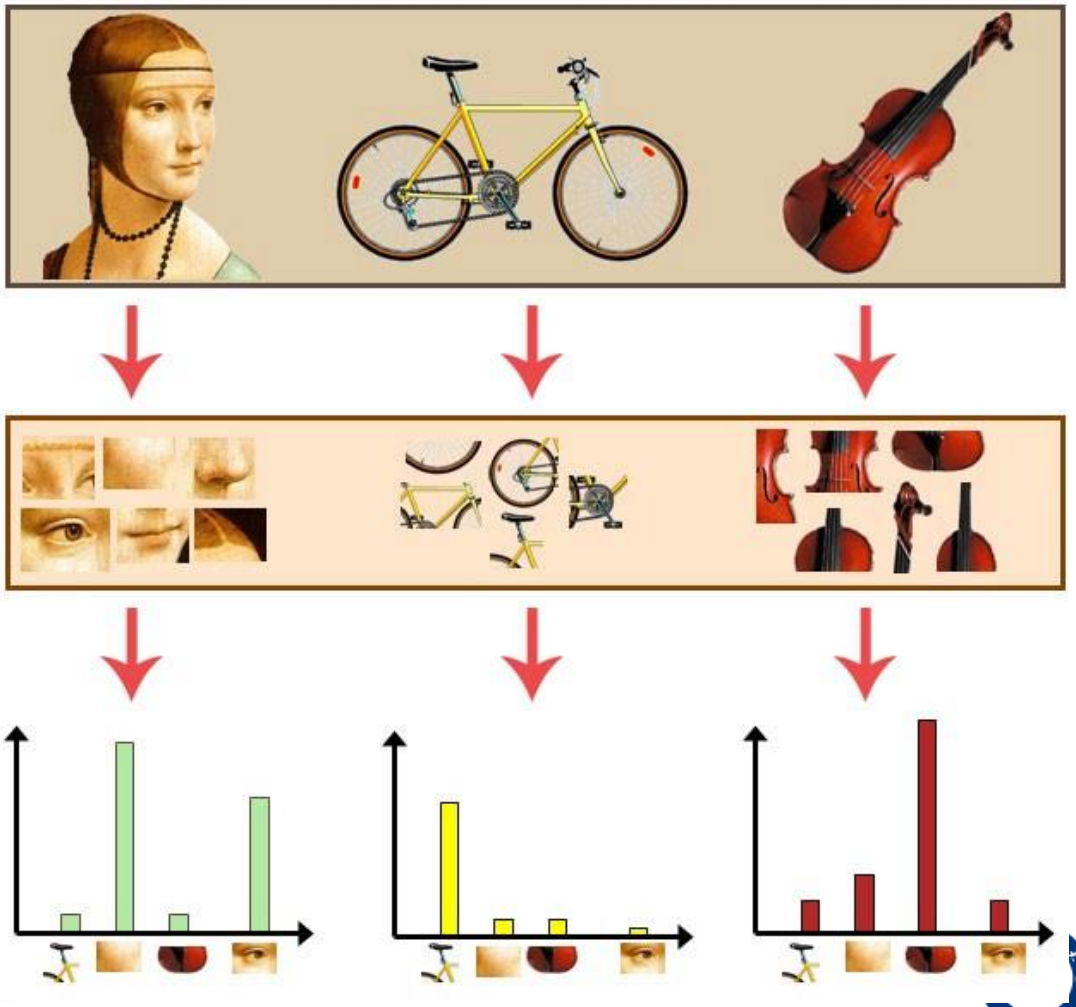
图像分类

- 传统方法：Bag-of-Words模型
  - 如何得到visual words？-> 聚类（如k-means）



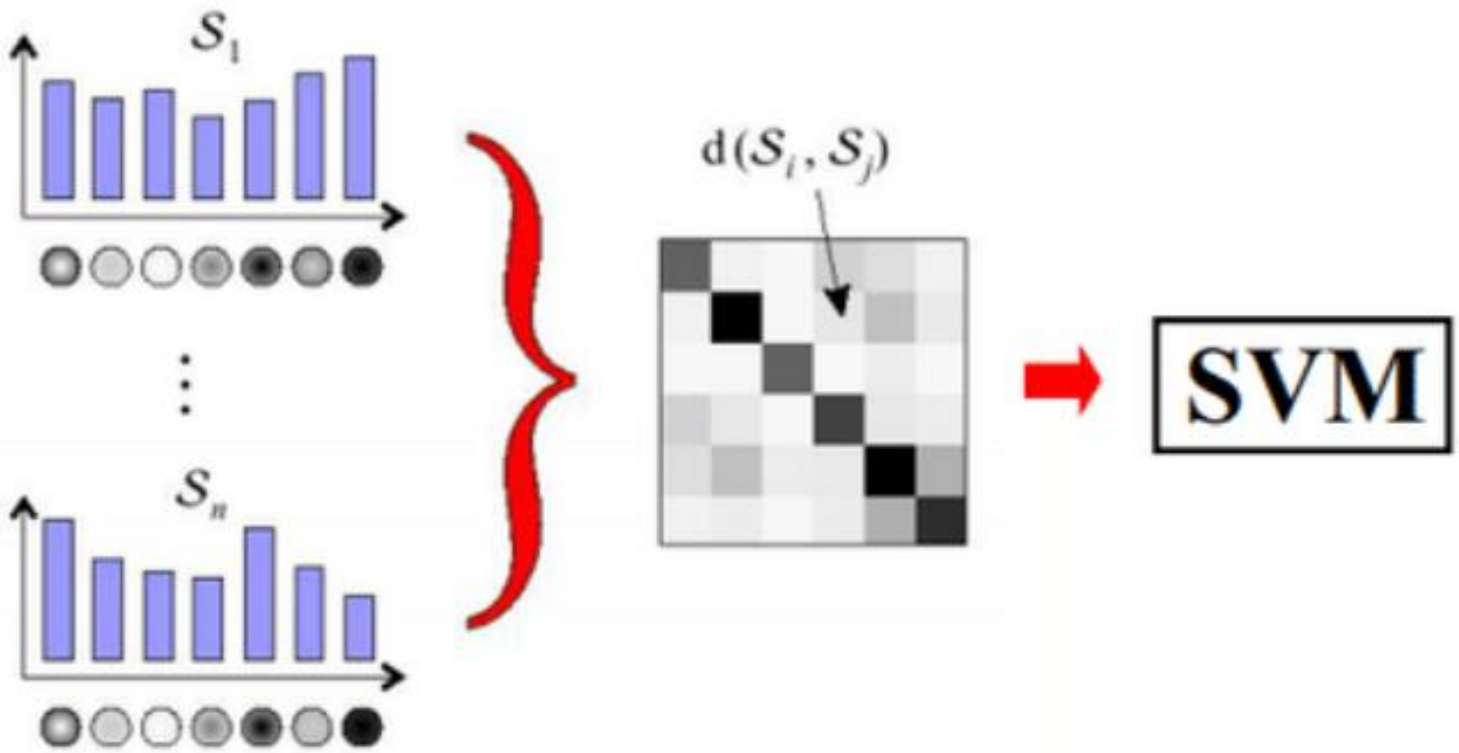
图像分类

- 传统方法：Bag-of-Words模型



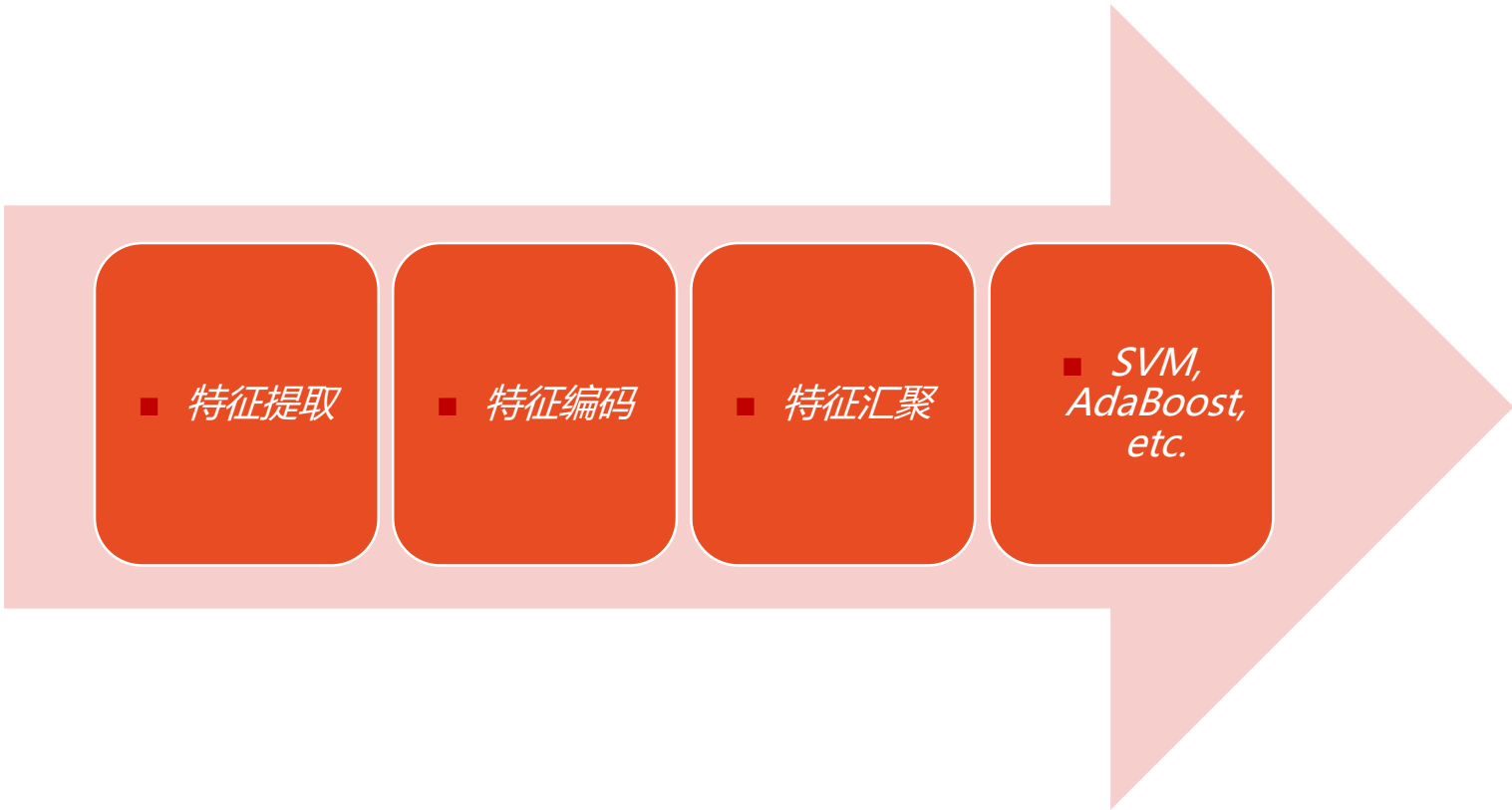
图像分类

- 传统方法：Bag-of-Words模型
  - 一张图片：K维的特征向量（如K=2000）



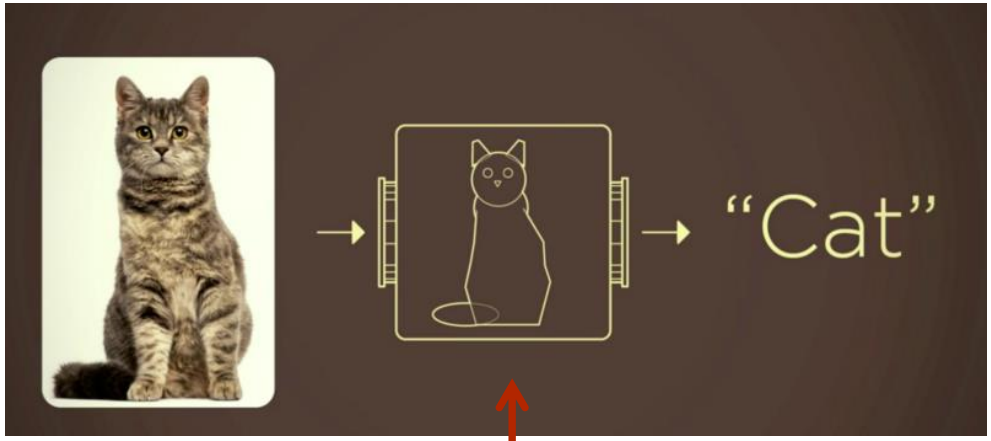
图像分类

- 传统方法：Bag-of-Words模型



图像分类

- 深度学习
  - 端到端学习（End-to-end learning）

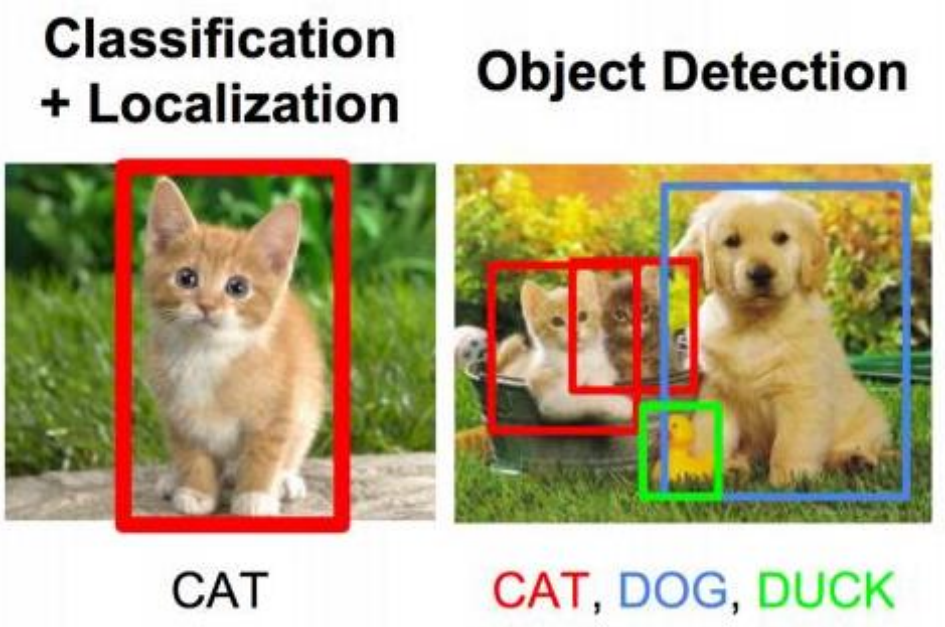


深度学习模型
AlexNet, VGG, etc.

目标检测

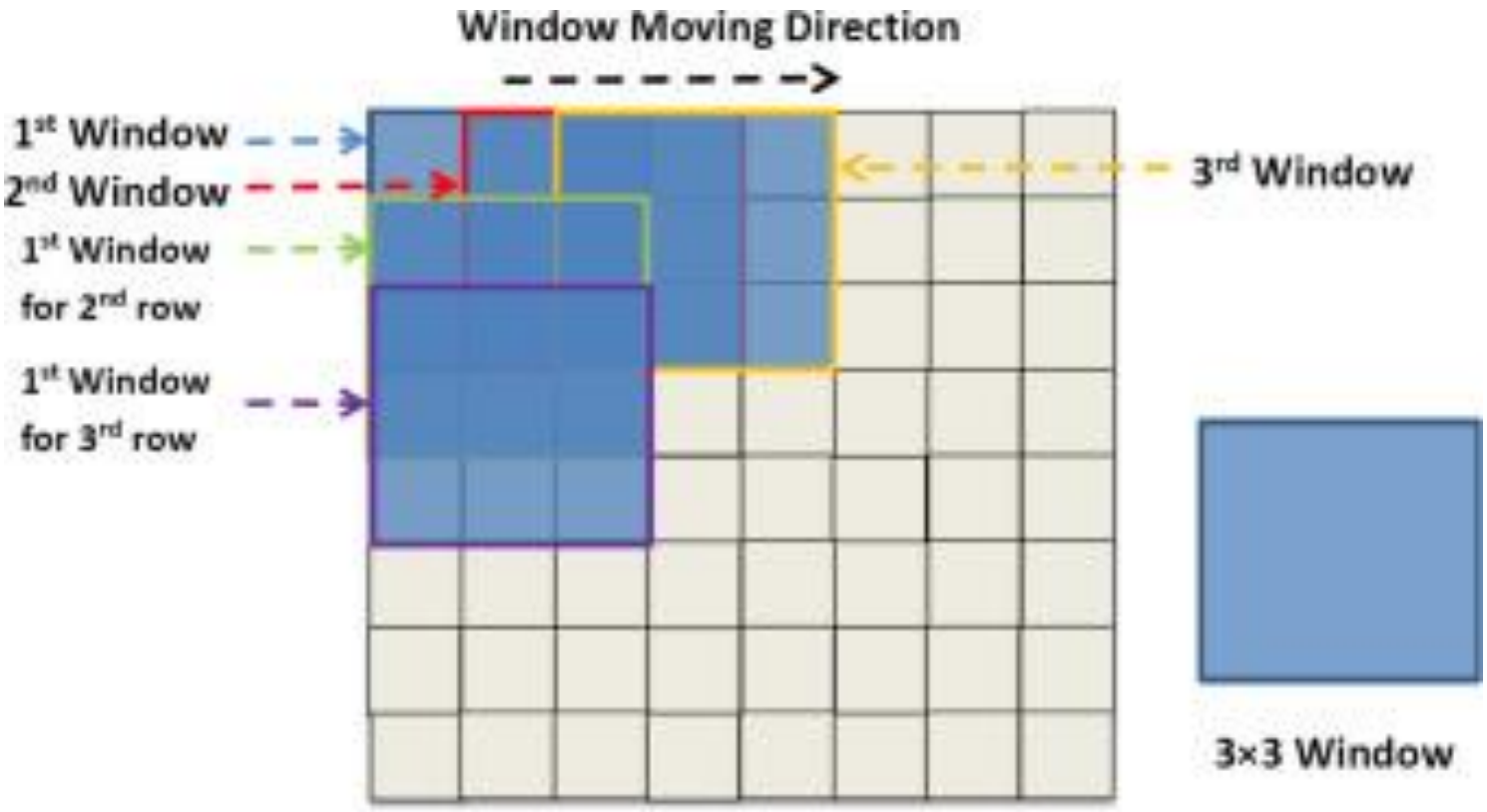
物体检测

- 目标检测主要任务是在给定的图片中精确找到物体所在位置，并标注出物体的类别。其模型可以识别一张图片的多个物体，并可以定位出不同物体（给出边界框）



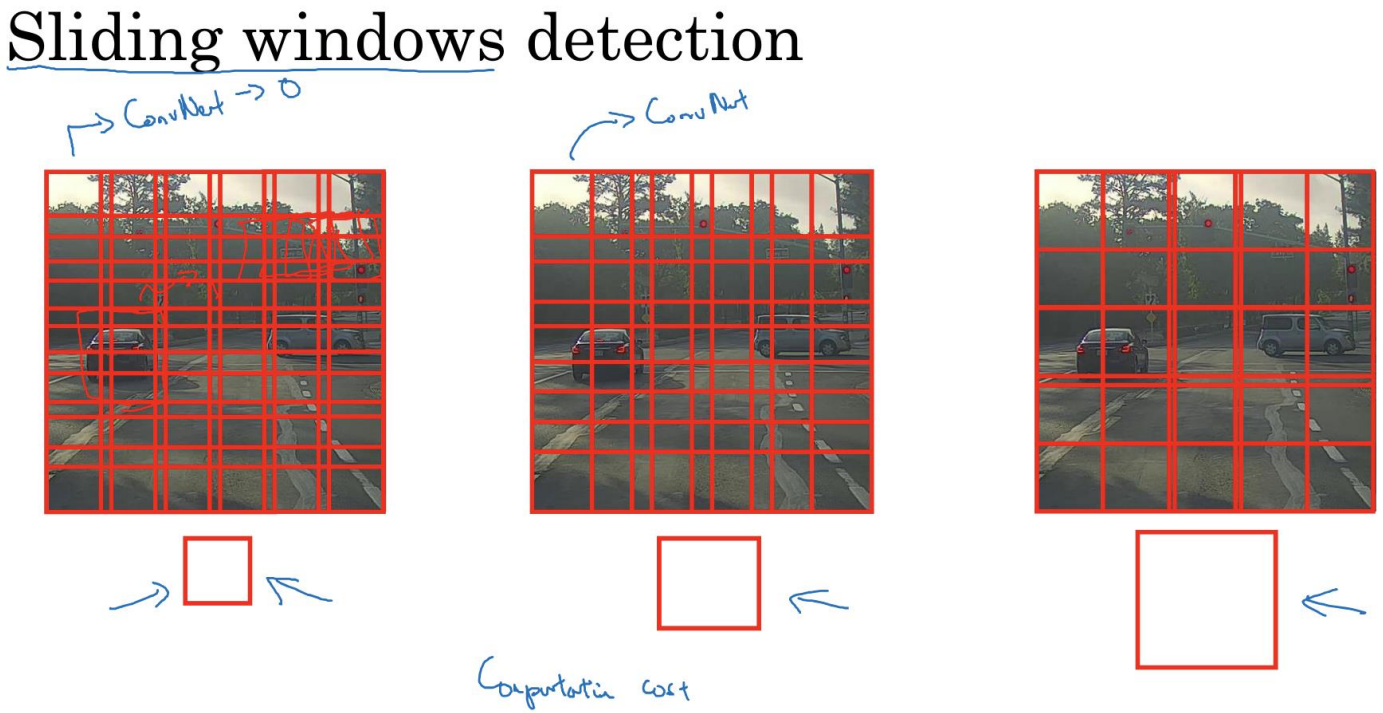
物体检测

- 传统方法：Sliding window



物体检测

- 传统方法：Sliding window
  - 不同尺寸



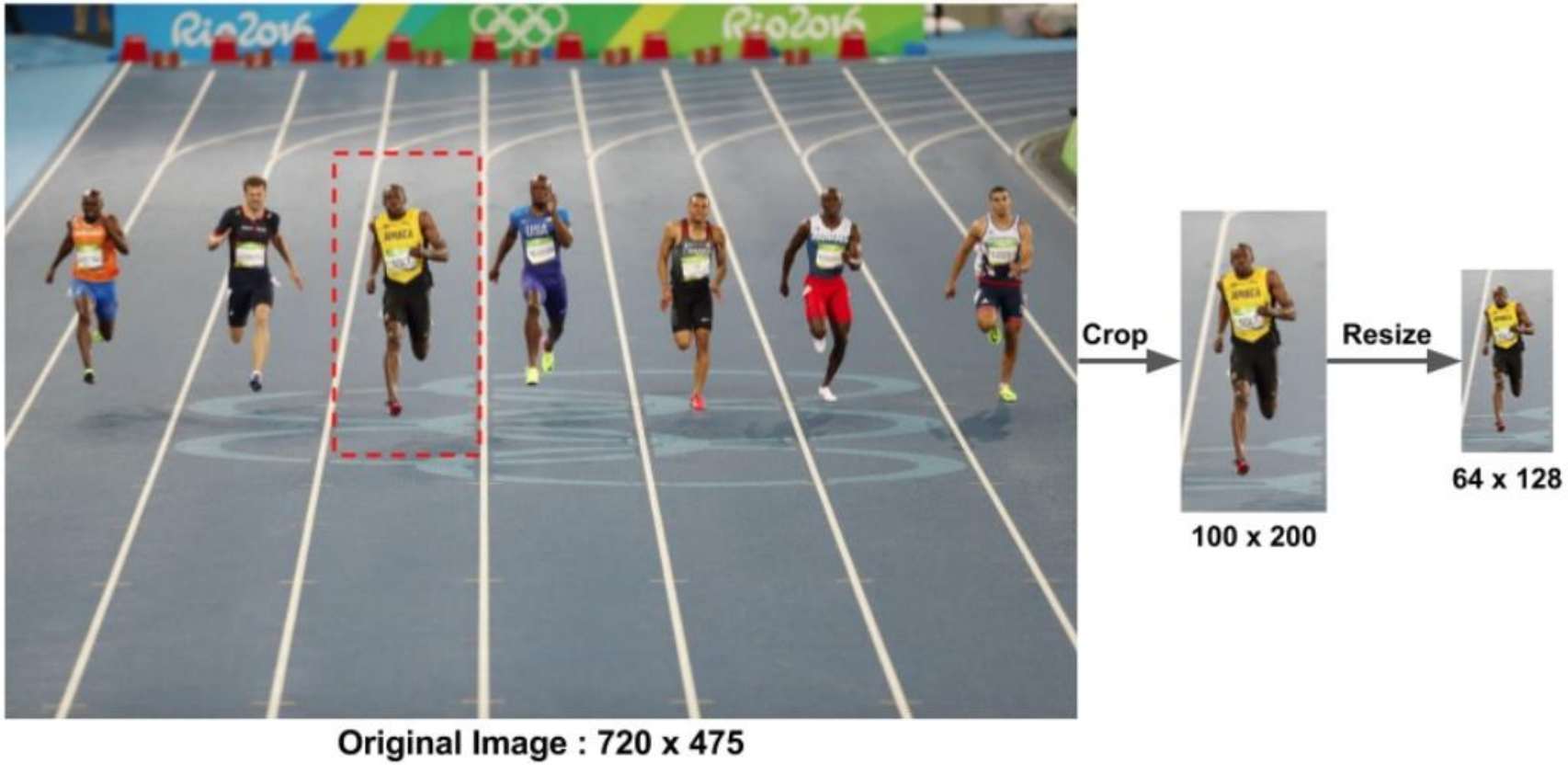
Andrew Ng

物体检测

- 传统方法：Haar特征+Adaboost特征检测 、HOG特征+SVM算法、DPM算法、滑动窗口法

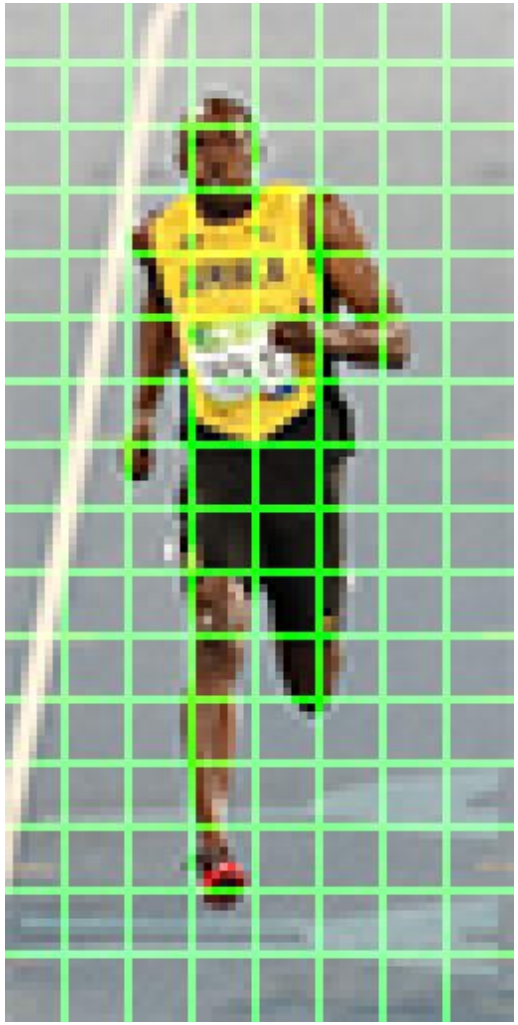
物体检测（HOG + SVM）

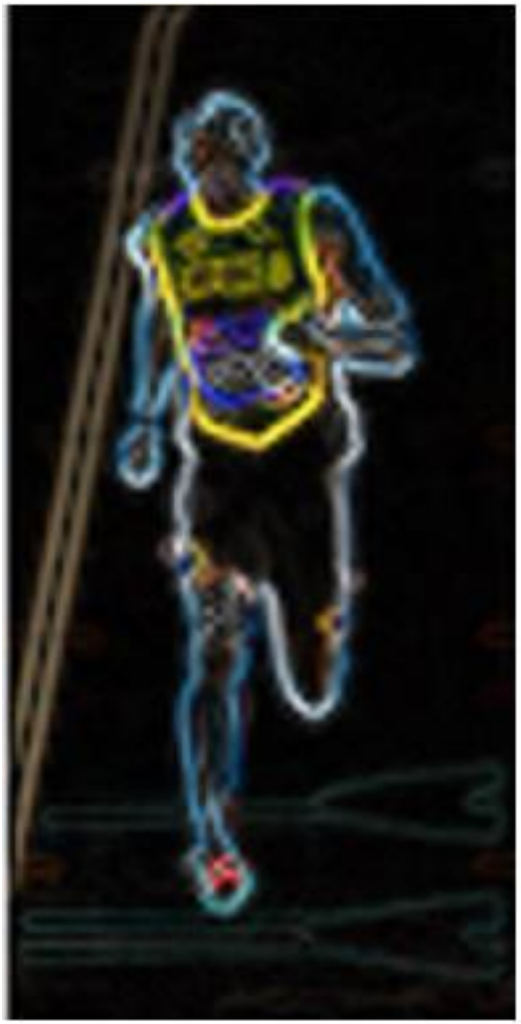
- 预处理

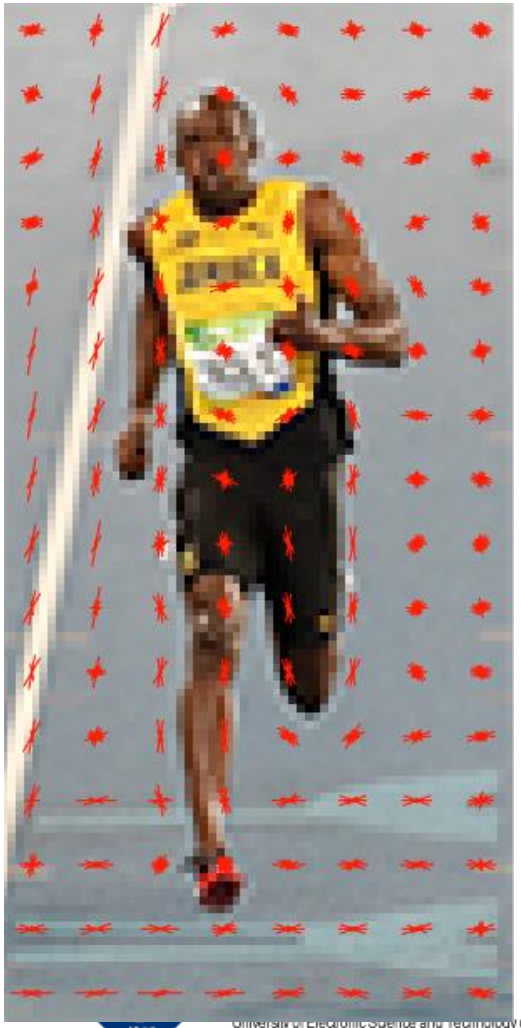


物体检测（HOG + SVM）

- 可视化

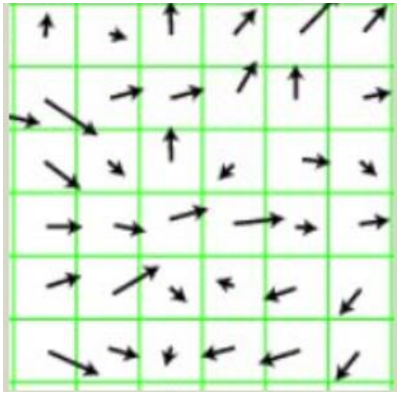


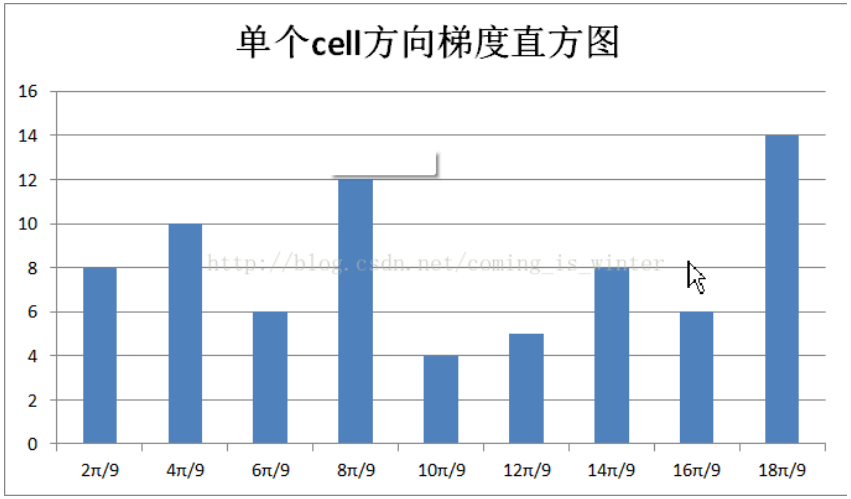




物体检测（HOG + SVM）

- 首先将图像分成小的区域（cell），然后采集细胞单元中各像素点的梯度的或边缘的方向直方图，最后把这些直方图组合起来构成特征。**最终得到3780维的HOG特征向量用来描述一张图**





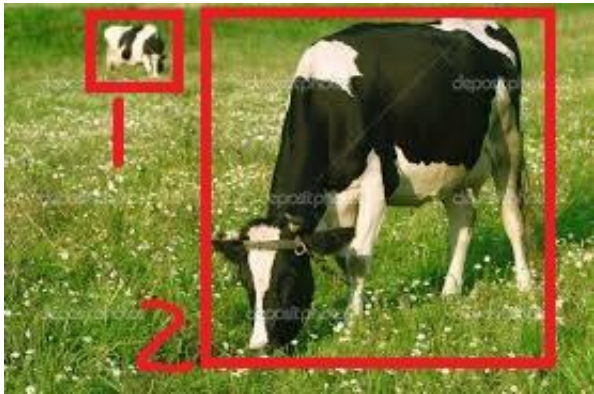
- 优点
  - HOG是在图像的局部方格单元上操作，所以它对图像几何的和光学的形变都能保持很好的不变性
  - 在粗的空域抽样、精细的方向抽样以及较强的局部光学归一化等条件下，效果相对鲁棒
    - 行人检测：只要行人大体上能够保持直立的姿势，可以容许行人有一些细微的肢体动作，而不影响检测效果

物体检测（Selective Search）

- 在目标检测中，进行分类和坐标回归之前，需要进行候选区域提取，selective search方法即是一种常用的提取候选区域的方法
- 要实现候选区域提取，需要进行相似度计算以及区域合并算法
- 如何定位一张图像上的目标(比如“牛”)？处理的流程可以是这样的：

  1、 将图像划分成很多的小区域(regions);

  2、判定每个区域是属于“牛”的还是“非牛”，将属于“牛”的区域进行合并，就定位到牛了！



物体检测（Selective Search）

区域合并算法：

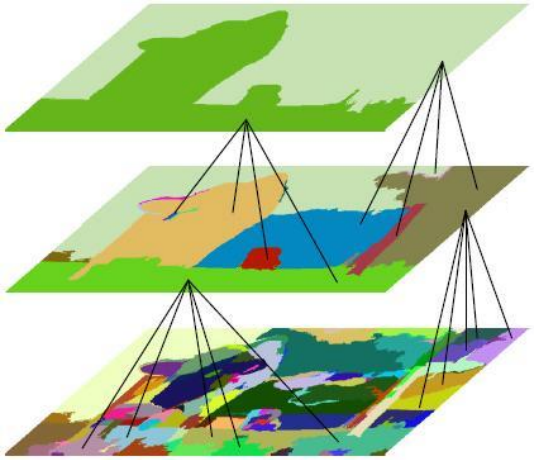
1. 使用 Efficient Graph-Based Image Segmentation的方法获取原始分割区域R={r1,r2,…,rn}

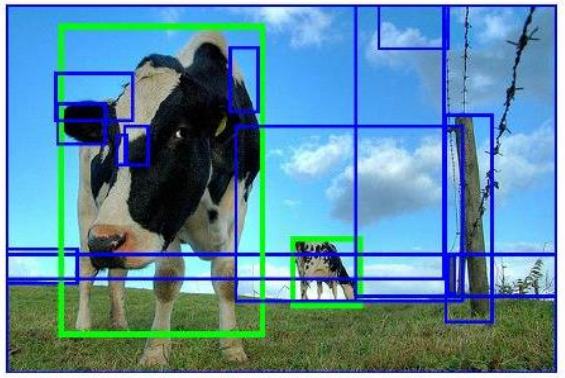
2. 初始化相似度集合S=ø

3. 计算两两**相邻区域**之间的**相似度**，将其添加到相似度集合S中

4. 从相似度集合S中找出，**相似度最大**的两个区域 ri 和rj，将其合并成为一个区域 rt，从相似度集合中除去原先与ri和rj相邻区域之间计算的相似度，计算rt与其相邻区域（原先与ri或rj相邻的区域）的相似度，将其结果添加的到相似度集合S中。同时将新区域 rt 添加到 区域集合R中。

5. 获取每个区域的Bounding Boxes，这个结果就是物体位置的可能结果L。





物体检测（Selective Search）

相似度计算：相似度计算方法有四种

1、颜色（color）相似度：计算其颜色直方图，并进行相似度计算。

2、纹理（texture）相似度：这里的纹理采用SIFT-Like特征。

3、大小（size）相似度：这里的大小是指区域中包含像素点的个数。使用大小的相似度计算，主要是为了尽量让小的区域先合并

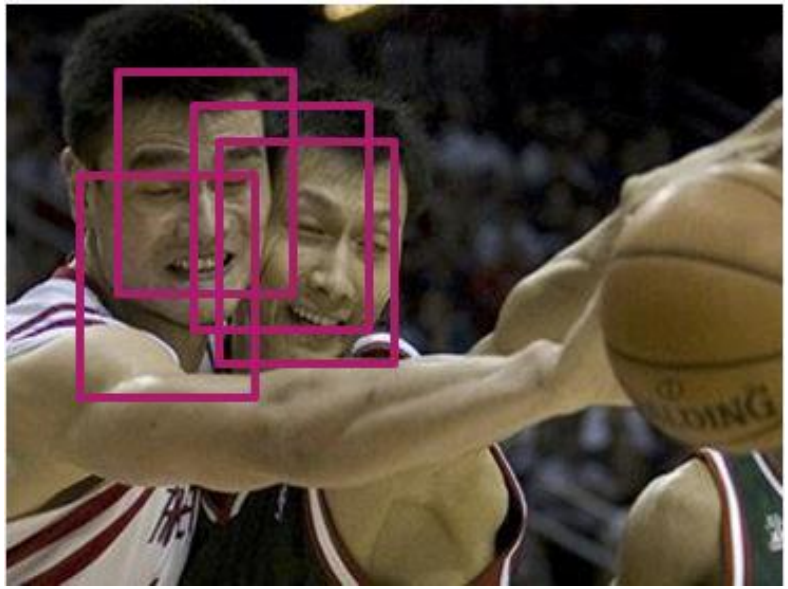
4、吻合（fit）相似度：这里主要是为了衡量两个区域是否更加“吻合”，其指标是合并后的区域的Bounding Box越小，其吻合度越高。

最后将四种度量方法合并成一种策略：S = a*S(color) + b* S(texture) + c*S(size) + d*S(fit)

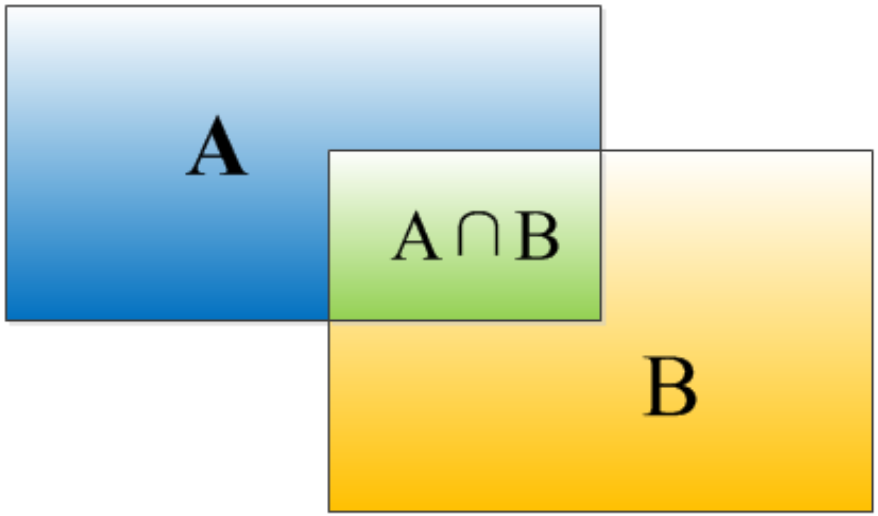
物体检测（Selective Search）

物体检测（NMS）

- **非极大抑制（Non-Maximum Suppression, NMS）：**对滑动窗口或者其它的object proposals方法产生大量的候选窗口进行融合。

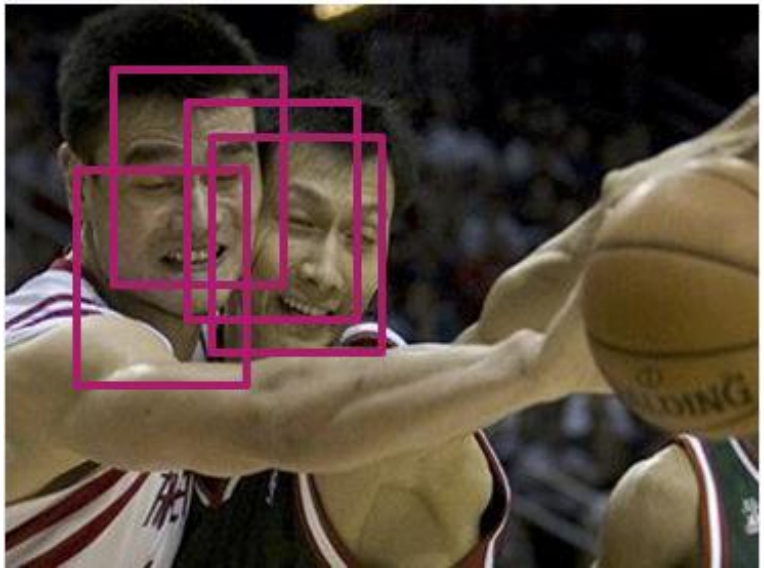


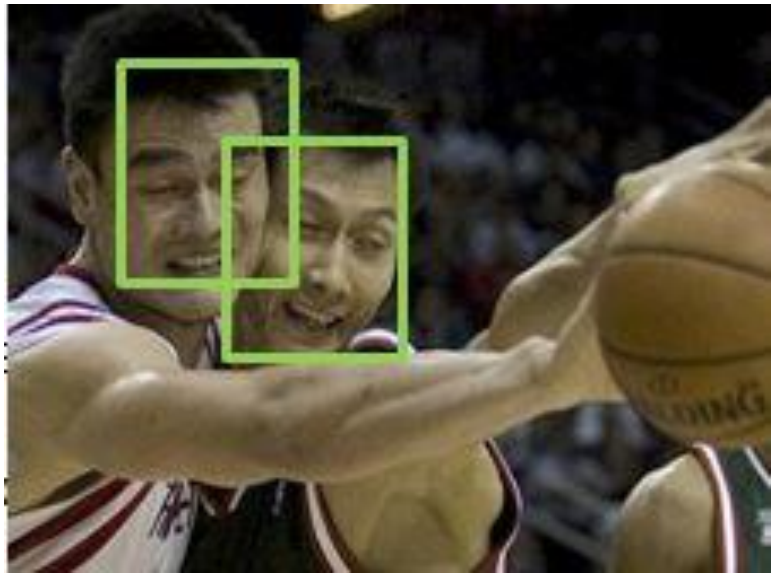
- IOU：IOU表示了bounding box 与 ground truth 的重叠度，IOU=(A∩B)/(A∪B)



物体检测（NMS）

- 具体步骤：首先对所有窗口的分数（基于classifier的预测值）进行从小到大排序取出最高分数的序号。循环计算到次高分数窗口与最高分数窗口A的IOU，若超过一定的threshold，就把A保留，把相关窗口删掉。





物体检测（NMS）

Object Detection

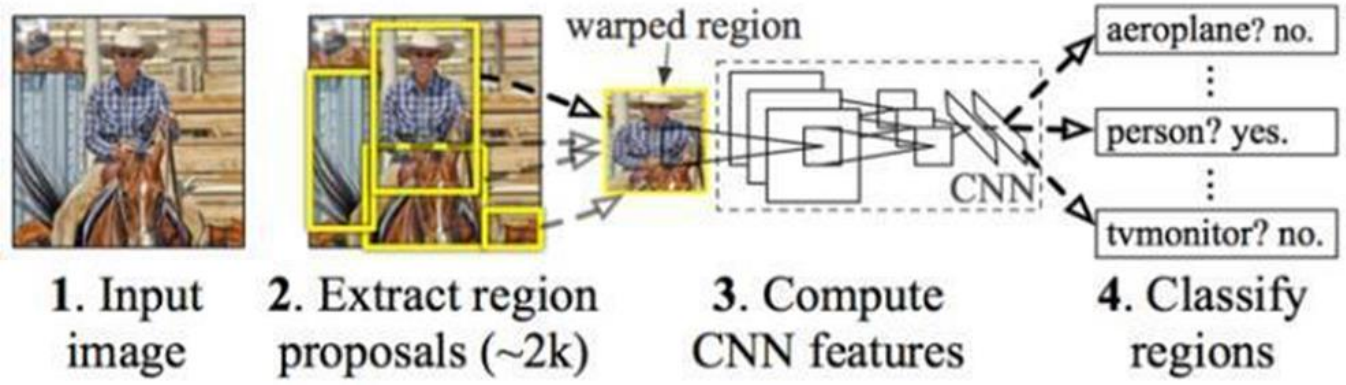
Non-max suppression

物体检测：深度方法

- **目前主流的目标检测算法主要是基于深度学习模型**，可以分成两大类：
  - Two-stage检测算法
    - 类似传统方法，首先产生候选区域（region proposals），然后对候选区域分类（一般还需要对位置精修）
    - 这类算法的典型代表是基于region proposal的R-CNN系算法，如R-CNN，Fast R-CNN，Faster R-CNN等
  - One-stage检测算法
    - 不需要region proposal阶段，直接产生物体的类别概率和位置坐标值，比较典型的算法如YOLO、SSD等

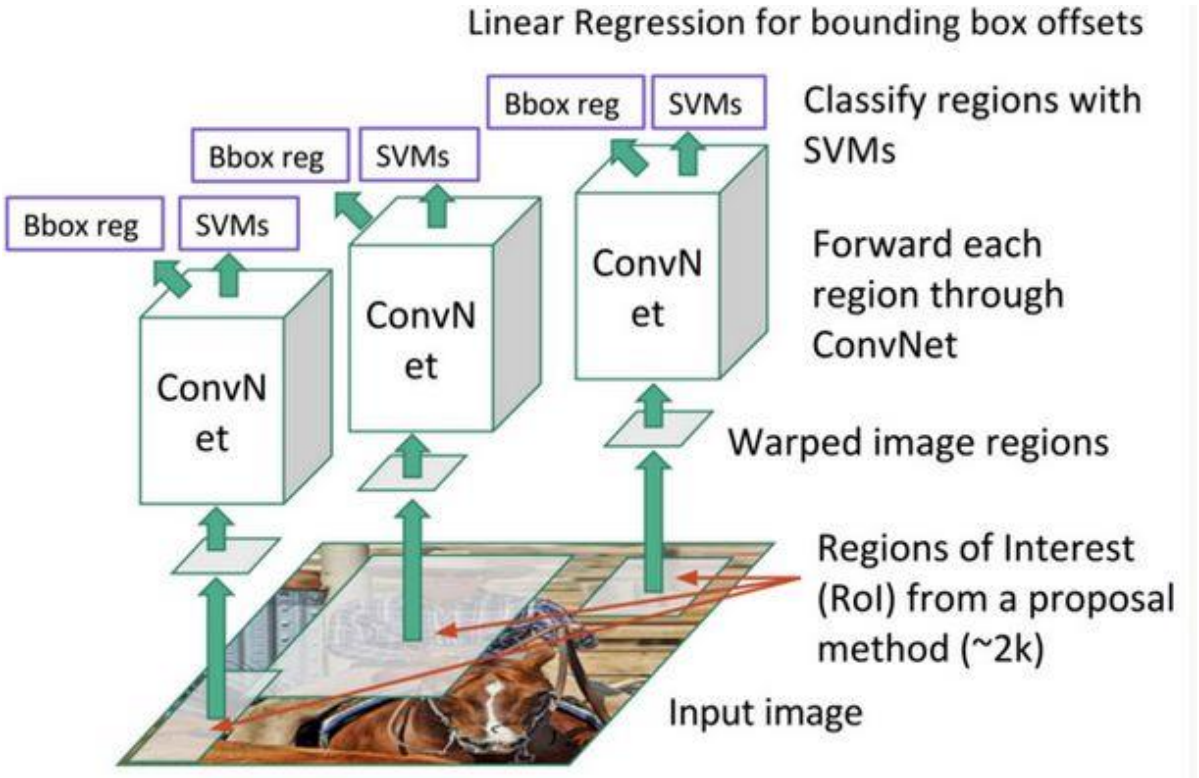
物体检测（R-CNN）

- R-CNN是基于region proposal方法的目标检测算法系列开山之作，其先进行区域搜索，然后再对候选区域进行分类。其选用Selective search方法来生成候选区域。
- 候选区域（Region Proposal）：是预先找出图中目标可能出现的位置。它利用了图像中的纹理、边缘、颜色等信息，可以保证在选取较少窗口(几千甚至几百)的情况下保持较高的召回率（Recall）



物体检测（R-CNN）

R-CNN模型的训练是多管道的，CNN模型首先使用2012 ImageNet中的图像分类竞赛数据集进行预训练。然后在检测数据集上对CNN模型进行finetuning，其中那些与真实框的IoU大于0.5的候选区域作为正样本，剩余的候选区域是负样本（背景）



物体检测（R-CNN）

- 总体来看，R-CNN是非常直观的，就是把检测问题转化为了分类问题，并且采用了CNN模型进行分类，但是效果却很好
  - 2012 PASCAL VOC数据集的mAP为62.4%（比第二名高出22%）
  - 在2013 ImageNet上的mAP为31.4%（比第二名高出7.1%）
- R-CNN缺点：

(1) 训练分为多个阶段，微调网络+训练SVM+训练边框回归器

(2) 训练耗时，占用磁盘空间大：5000张图像产生几百G的特征文件

(3) 速度慢: 使用GPU, VGG16模型处理一张图像需要47s

(4) 测试速度慢：每个候选区域需要运行整个前向CNN计算

(5) SVM和bounding box回归是事后操作：在SVM和回归过程中CNN特征没有被学习更新

(6) 输入图片的大小必须要固定（224x224、227x227等）

物体检测（SPPNet)

- **何恺明：SPP-NET (Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition) 2014年**
- R-CNN对图像提完Region Proposal（2000个左右）之后将每个Proposal当成一张图像进行后续处理(CNN提特征+SVM分类)，实际上对一张图像进行了2000次提特征和分类的过程
- 完全可以对图像只提取一次卷积层特征，然后只需要将Region Proposal在原图的位置映射到卷积层特征图上，这样对于一张图像我们只需要提一次卷积层特征，然后将每个Region Proposal的卷积层特征输入到全连接层做后续操作

物体检测（SPPNet)

- **存在的问题：每个Region Proposal的尺度不一样，并不能直接这样输入全连接层**
- 而SPP-NET恰好可以解决这个问题。做到的效果为：不管输入的图片是什么尺度都可以（强行对特征图进行分割）

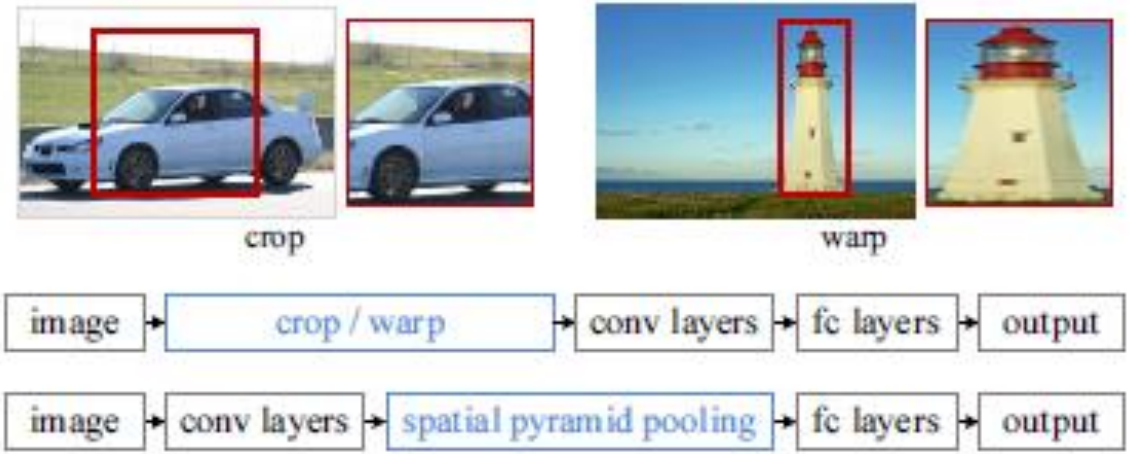
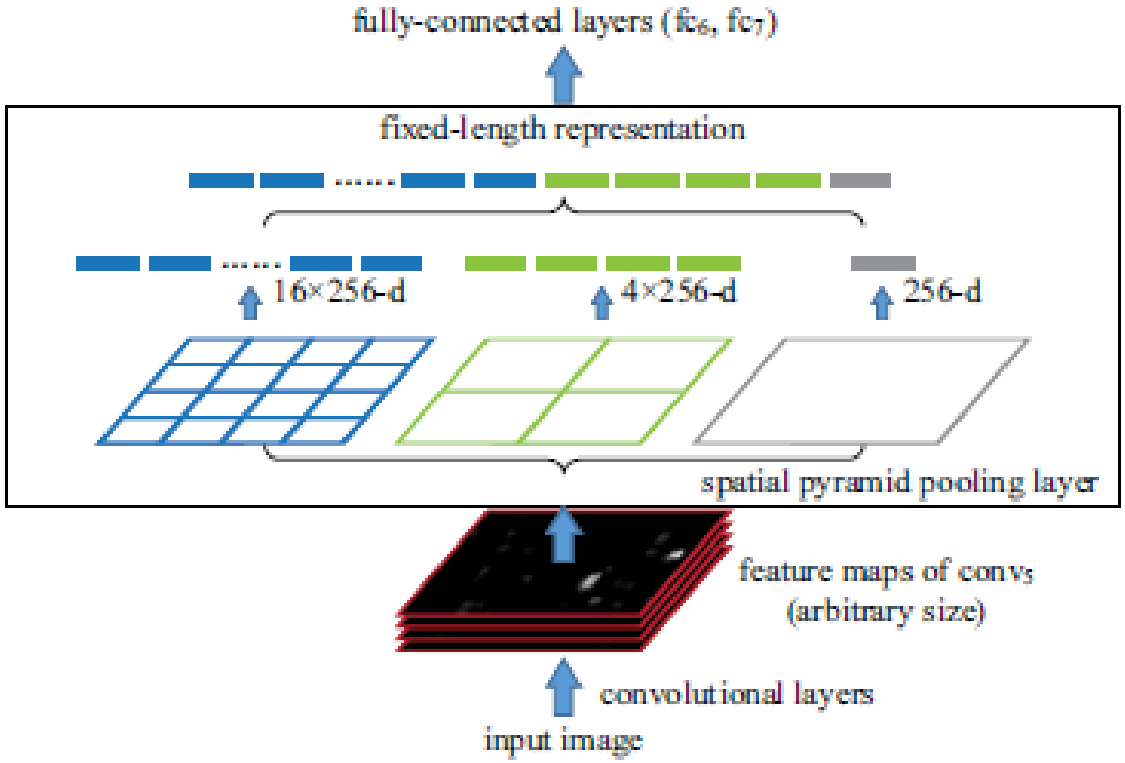


Figure 1: Top: cropping or warping to fit a fixed size. Middle: a conventional CNN. Bottom: our spatial pyramid pooling network structure.

物体检测（SPPNet)

- **存在的问题：每个Region Proposal的尺度不一样，并不能直接这样输入全连接层**
- 而SPP-NET恰好可以解决这个问题。做到的效果为：不管输入的图片是什么尺度都可以（强行对特征图进行分割）



物体检测（SPPNet)

- 使用**SPP-NET**相比于**R-CNN**可以大大加快目标检测的速度，但是依然存在着很多问题：

  (1) 训练分为多个阶段，步骤繁琐: 微调网络+训练SVM+训练训练边框回归器

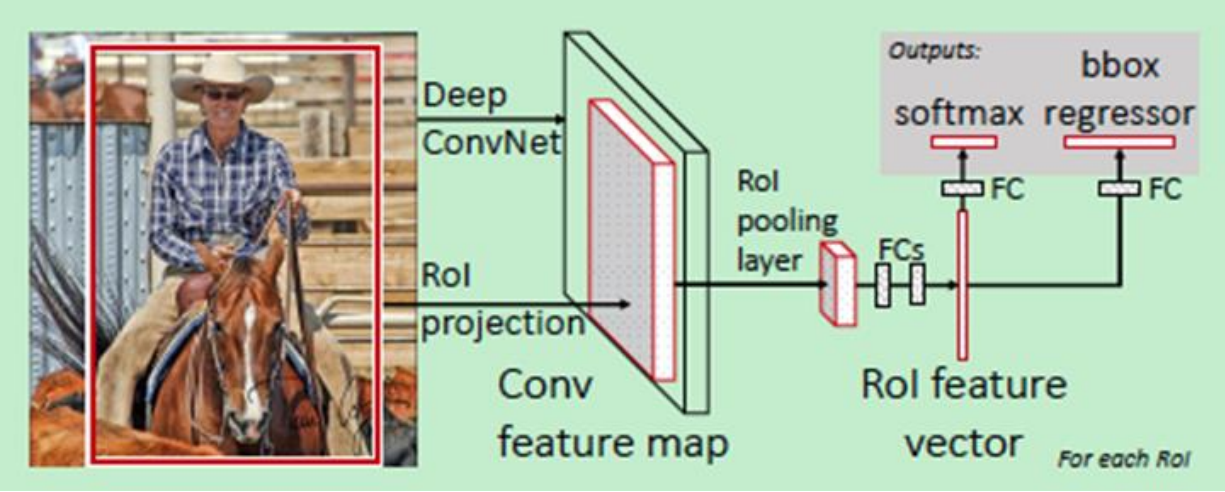
  (2) SPP-NET在微调网络的时候固定了卷积层，只对全连接层进行微调，而对于一个新的任务，有必要对卷积层也进行微调。（分类的模型提取的特征更注重高层语义，而目标检测任务除了语义信息还需要目标的位置信息）

物体检测（Fast R-CNN）

- Ross Girshick, ICCV 2015提出
- Fast R-CNN的提出主要是为了减少候选区域使用CNN模型提取特征向量所消耗的时间，其主要借鉴了SPP-net的思想。
- 而对于Fast R-CNN，其CNN模型的输入是整张图片，然后结合RoIs（Region of Interests）pooling和Selective Search方法从CNN得到的特征图中提取各个候选区域的所对应的特征。采用了softmax分类器而不是SVM分类器。

物体检测（Fast R-CNN）

- ROI pooling layer实际上是SPP-NET的一个精简版，SPP-NET对每个proposal使用了不同大小的金字塔映射，而ROI pooling layer只需要下采样到一个7x7的特征图



物体检测（Fast R-CNN）

- 性能对比

	R-CNN	Fast R-CNN
Training Time:	84 hours	**9.5 hours**
(Speedup)	1x	**8.8x**
Test time per image	47 seconds	**0.32 seconds**
(Speedup)	1x	**146x**
mAP (VOC 2007)	66.0	**66.9**

Faster!

FASTER!

Better!

Using VGG-16 CNN on Pascal VOC 2007 dataset

物体检测（Fast R-CNN）

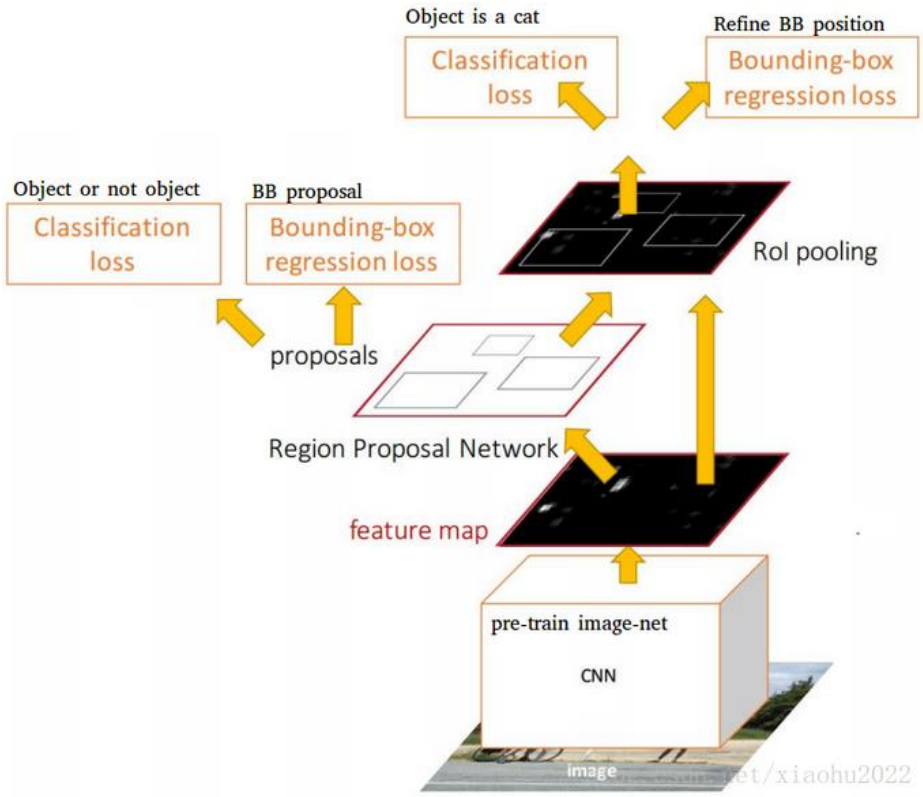
- 优点：
  - Fast R-CNN融合了R-CNN和SPP-NET的精髓，并且引入多任务损失函数，使整个网络的训练和测试变得十分方便。在Pascal VOC2007训练集上训练，在VOC2007测试的结果为66.9%(mAP)，如果使用VOC2007+2012训练集训练，在VOC2007上测试结果为70%（数据集的扩充能大幅提高目标检测性能）
- 缺点：
  - Region Proposal的提取使用selective search，消耗了目标检测过程大部分的时间（提Region Proposal 2~3s，而提特征分类只需0.32s），无法满足实时应用，而且并没有实现真正意义上的端到端训练测试（region proposal使用selective search先提取出来）
- 端到端：Faster R-CNN

物体检测（Faster R-CNN）

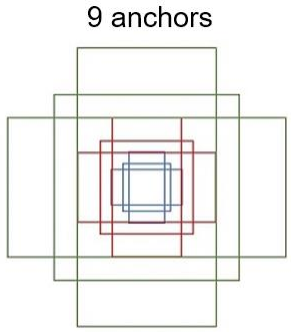
对于Fast R-CNN，其仍然需要selective search方法来生产候选区域，这是非常费时的

Faster R-CNN模型引入了RPN (Region Proposal Network)，直接产生候选区域

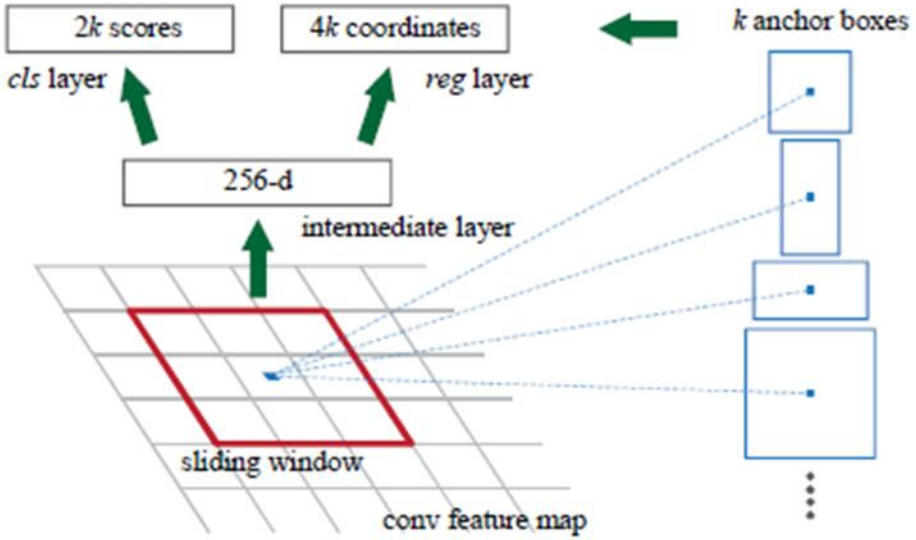
RPN采用任意大小的的图像作为输入，并输出一组候选的矩形，每个矩形都有一个对象分数。Faster R-CNN将一直以来分离的region proposal和CNN分类融合到了一起，使用端到端的网络进行目标检测，无论在速度上还是精度上都得到了不错的提高



物体检测（Faster R-CNN）



- Region Proposal Network (RPN)
  - Anchor是滑动窗口的中心，它与尺度和长宽比相关，默认采3种尺度（128,256,512），3种长宽比（1:1,1:2,2:1），则在每一个滑动位置k=9 anchors。最后一层卷积层，后边接cls layer(box-classification layer)和reg layer(box-regression layer)分别用于分类和边框回归



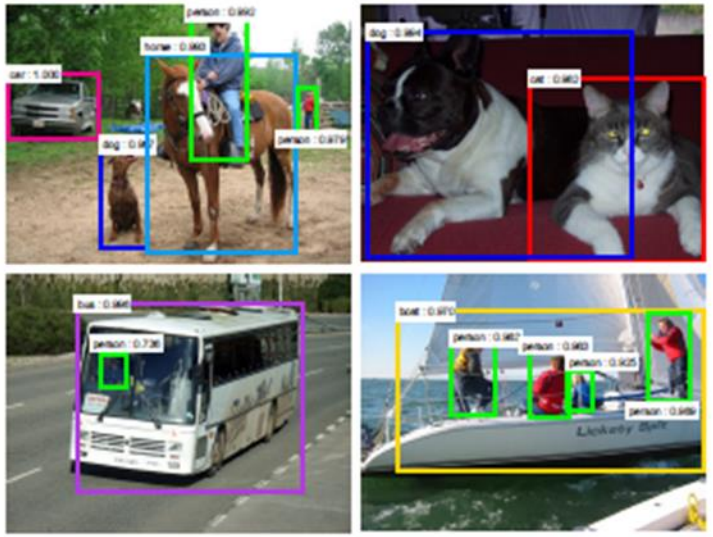


Figure 3: **Left**: Region Proposal Network (RPN). **Right**: Example detections using RPN proposals on PASCAL VOC 2007 test. Our method detects objects in a wide range of scales and aspect ratios.

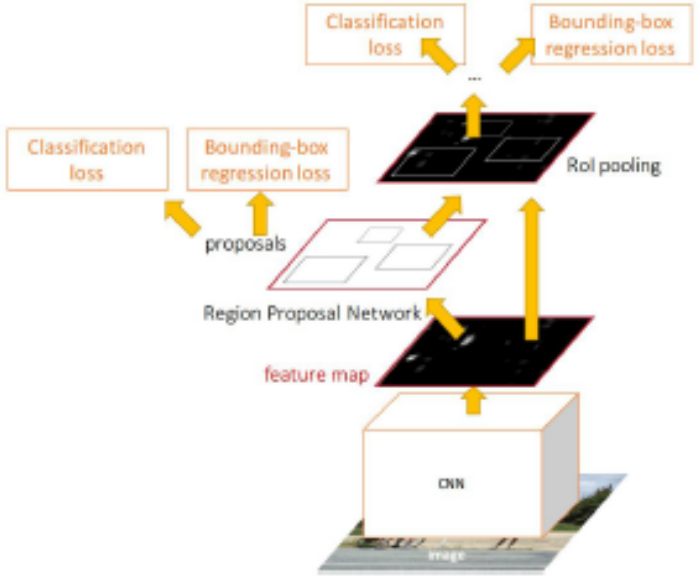
物体检测（Faster R-CNN）

- Faster R-CNN模型采用一种4步迭代的训练策略，使得Faster R-CNN可以与RPN有机融合在一起，形成一个统一的网络

Faster R-CNN: Training

In the paper: Ugly pipeline
- Use alternating optimization to train RPN, then Fast R-CNN with RPN proposals, etc.
- More complex than it has to be

Since publication: Joint training!
One network, four losses
- RPN classification (anchor good / bad)
- RPN regression (anchor -> proposal)
- Fast R-CNN classification (over classes)
- Fast R-CNN regression (proposal -> box)



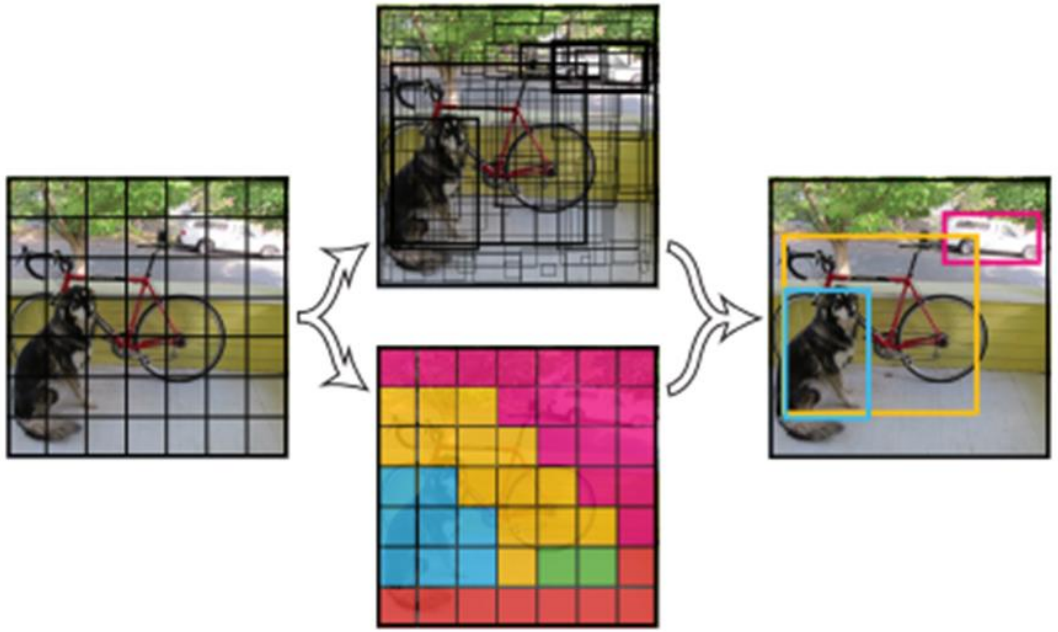
物体检测

- 以上介绍的都是Two-stage方法，这种方法都有一个共同的特征，即第一个阶段都要产生候选区域(region proposal)，之后再对候选区域进行分类和坐标回归等操作。虽然不同算法产生后选取区域的方法不尽相同，但是其目的都是一致的。
- 但是two-stage方法有其固有缺点，即是第一阶段都要进行候选区域提取会消耗很多时间，使目标检测的耗时增加，效率降低。
- 因此在此基础上，我们对目标检测方法进行了改进，产生了one-stage方法，可以直接对物体的坐标和类别进行回归而不需要进行显式的提取候选区域的过程。

物体检测（YOLO）

- YOLO（You Only Look Once）是一个全新的方法，把一整张图片一下子应用到一个神经网络中去。网络把图片分成不同的区域，然后给出每个区域的边框预测和概率，并依据概率大小对所有边框分配权重。最后，设置阈值，只输出得分（概率值）超过阈值的检测结果

直接把图片分成不同区域，
不用先找**proposals**



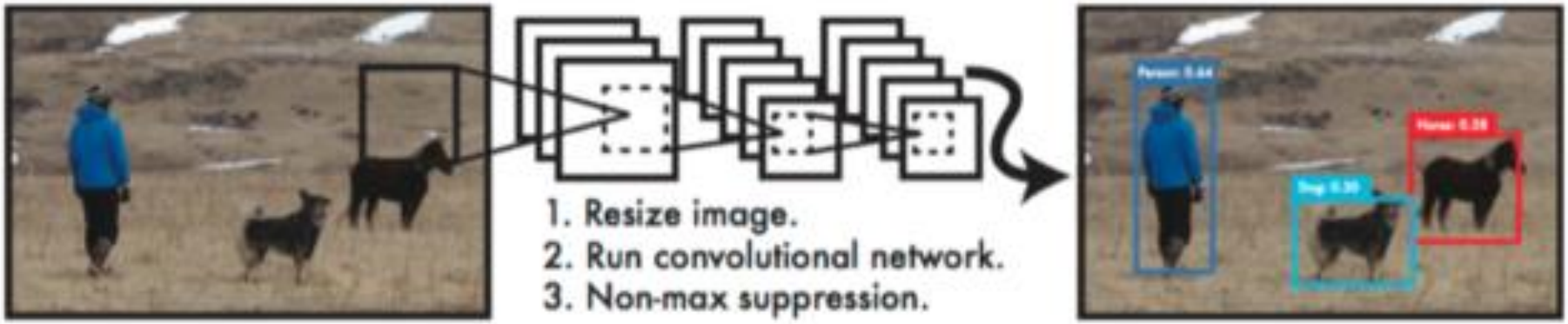
物体检测（YOLO）

- YOLO的目标检测的流程：

(1) 给个一个输入图像，resize图像到448x448，将图像划分成7x7的网格

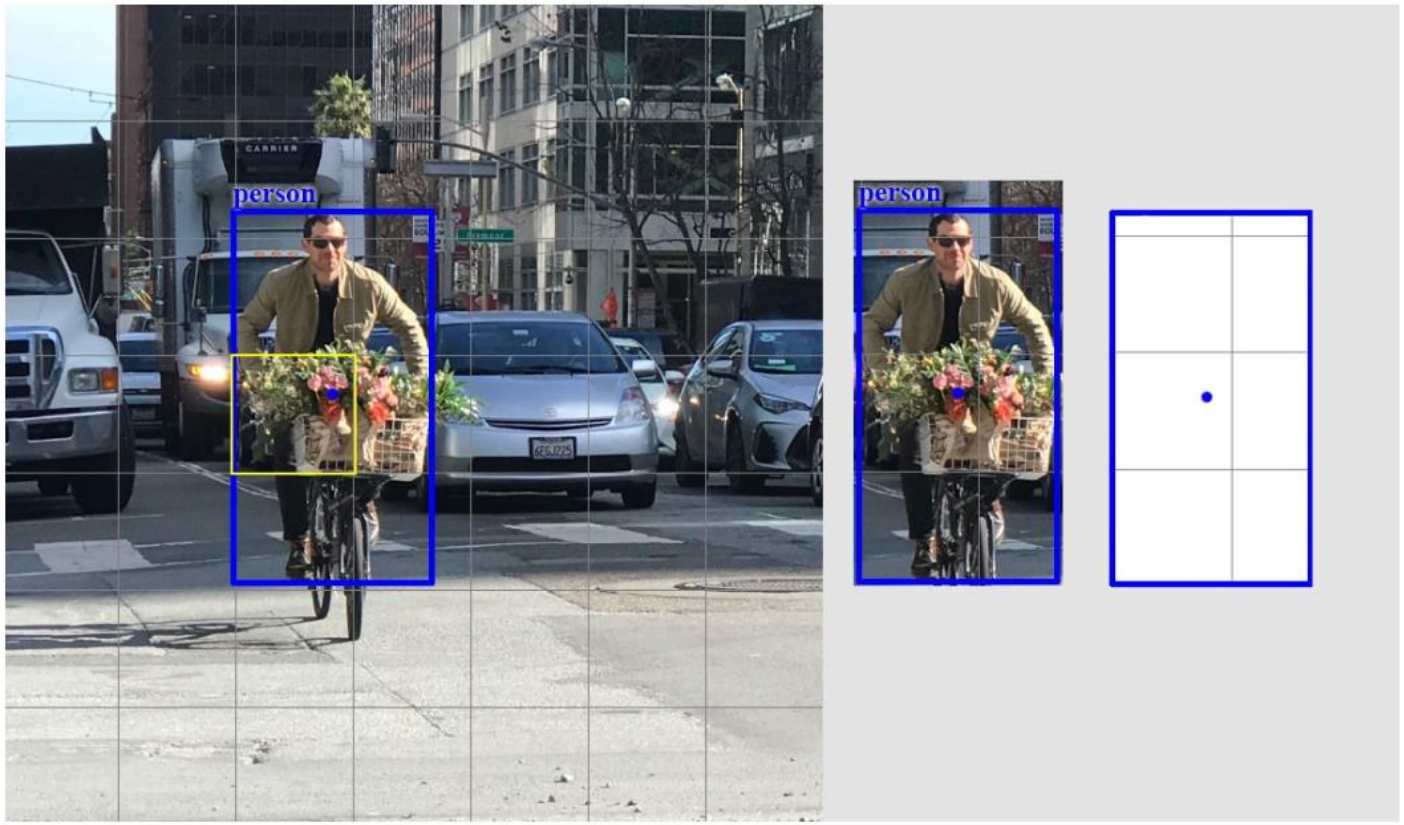
(2) 对于每个网格，我们都预测2个边框（包括每个边框是目标的置信度以及每个边框区域在多个类别上的概率）

(3) 根据上一步可以预测出7*7*2个目标窗口，然后根据阈值去除可能性比较低的目标窗口，最后NMS去除冗余窗口即可。



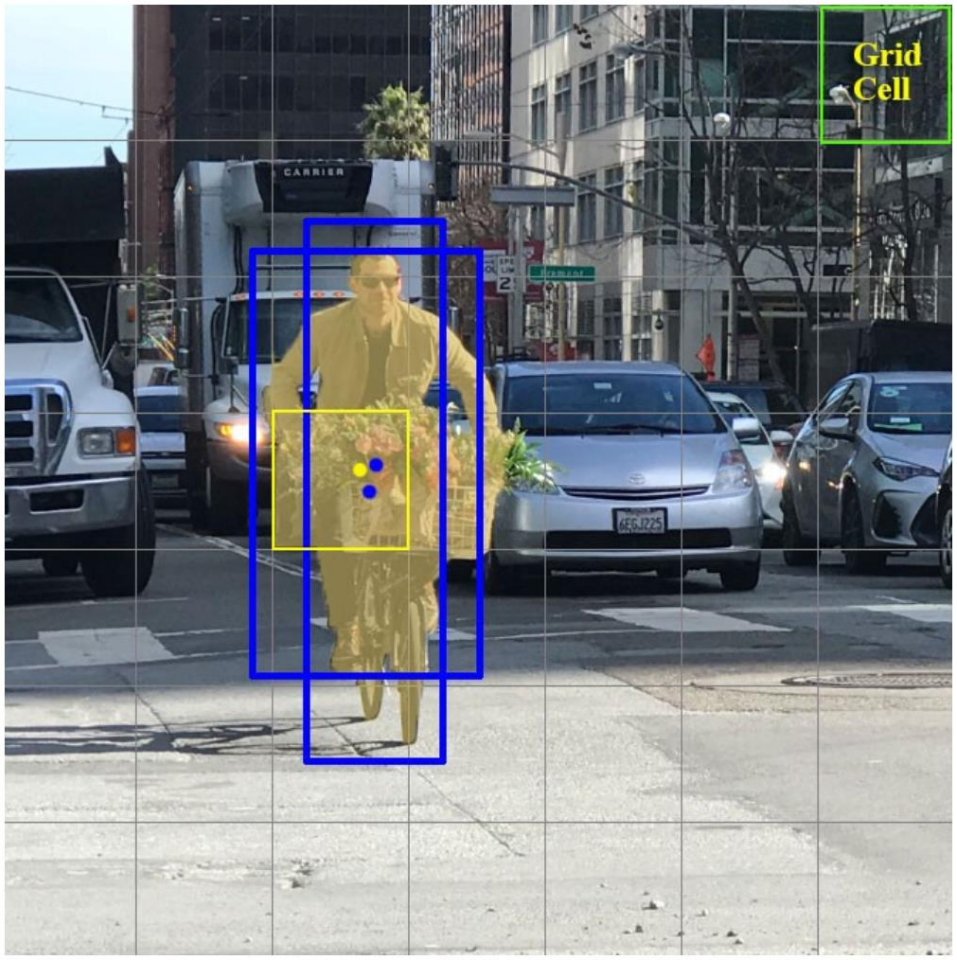
物体检测（YOLO）

- 输入图片分为S x S个grid cells
- 每个cell只负责预测一个物体



物体检测（YOLO）

- 输入图片分为S x S个grid cells
- 每个cell只负责预测一个物体（2个bounding boxes）

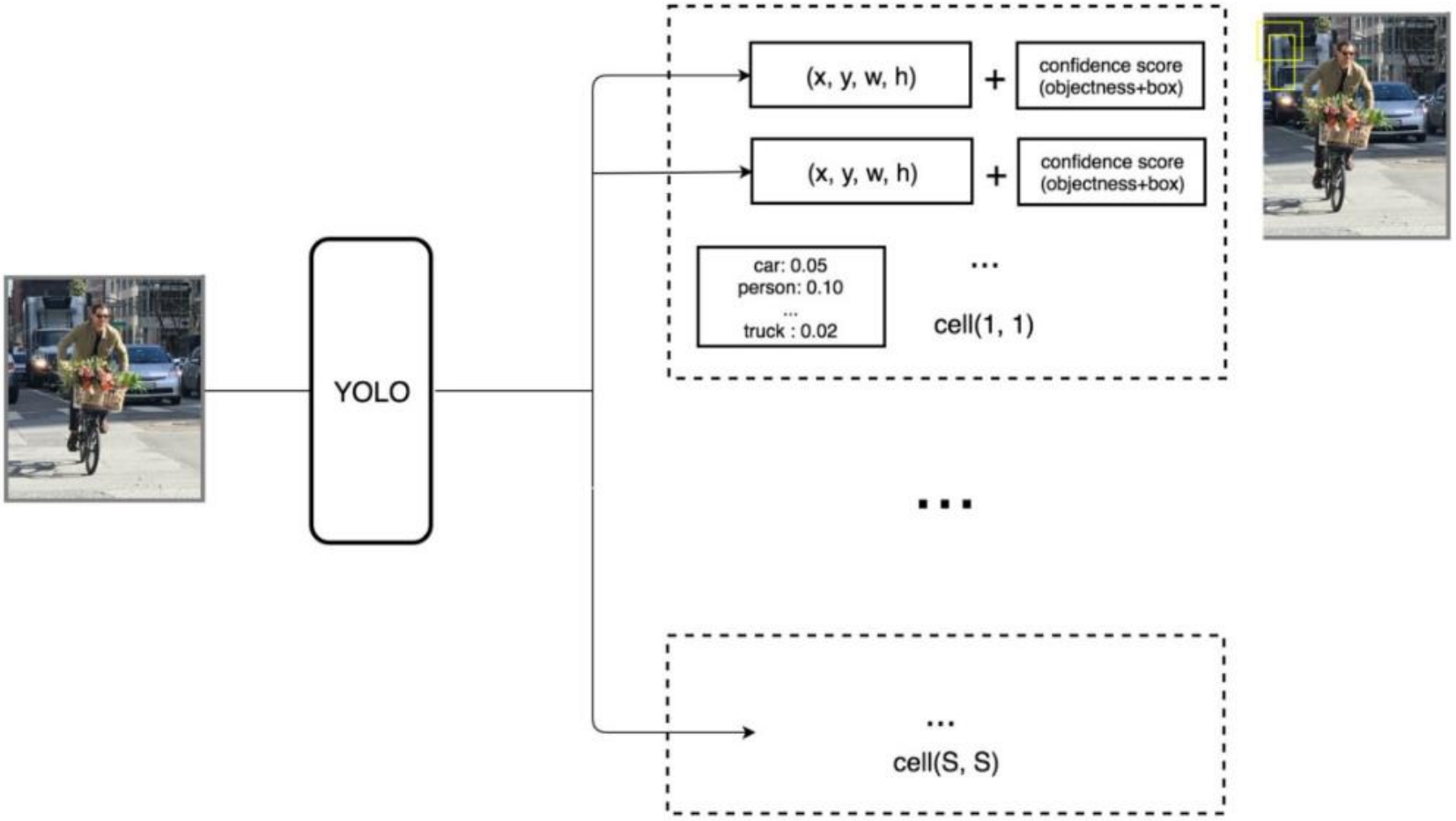


物体检测（YOLO）

- 对于每一个cell
  - 预测B个bounding boxes，每个bounding box有一个confidence score (objectness)
  - 只预测一个物体
  - 假设有C个类别，预测分别属于每个类别的概率
  - 对于PASCAL VOC 2007，有7x7 grid cells, 2 bounding boxes, 以及20个类别

物体检测（YOLO）

- 对于PASCAL VOC 2007，有7x7 grid cells, 2 bounding boxes, 以及20个类别



物体检测（YOLO）

YOLO网络结构如右图。主要采用了AlexNet。卷积层主要用来提取特征，全连接层主要用来预测类别概率和坐标

卷积层之后接了一个4096维的全连接层，然后后边又全连接到一个7*7*30维的张量上。7*7就是划分的网格数，在每个网格上预测目标两个可能的位置以及这个位置的目标置信度和类别，也就是每个网格预测两个目标，每个目标的信息有4维坐标信息(中心点坐标+长宽)，1个是目标的置信度，还有类别数20(VOC上20个类别)，总共就是(4+1)*2+20 = 30维的向量

30 = 2*(4+1) + 20

2个bounding boxes
4个坐标值
1个confidence score
20个类别

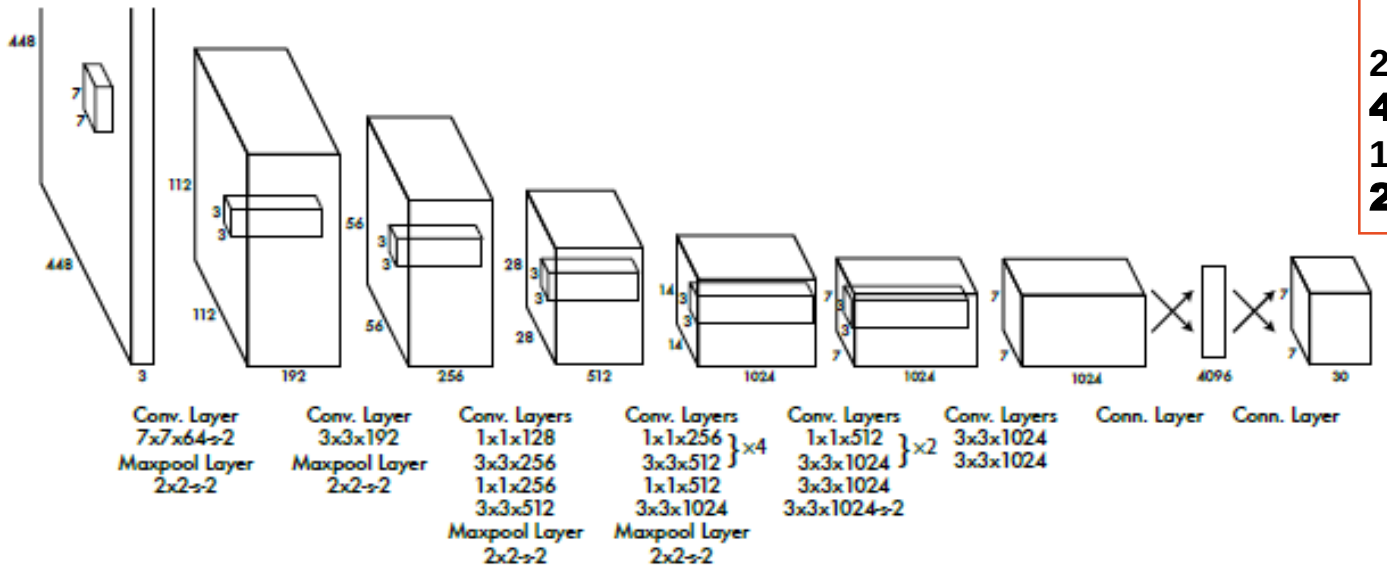


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

物体检测（YOLO）

- Loss function
  - Localization
  - Confidence
  - Classification

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

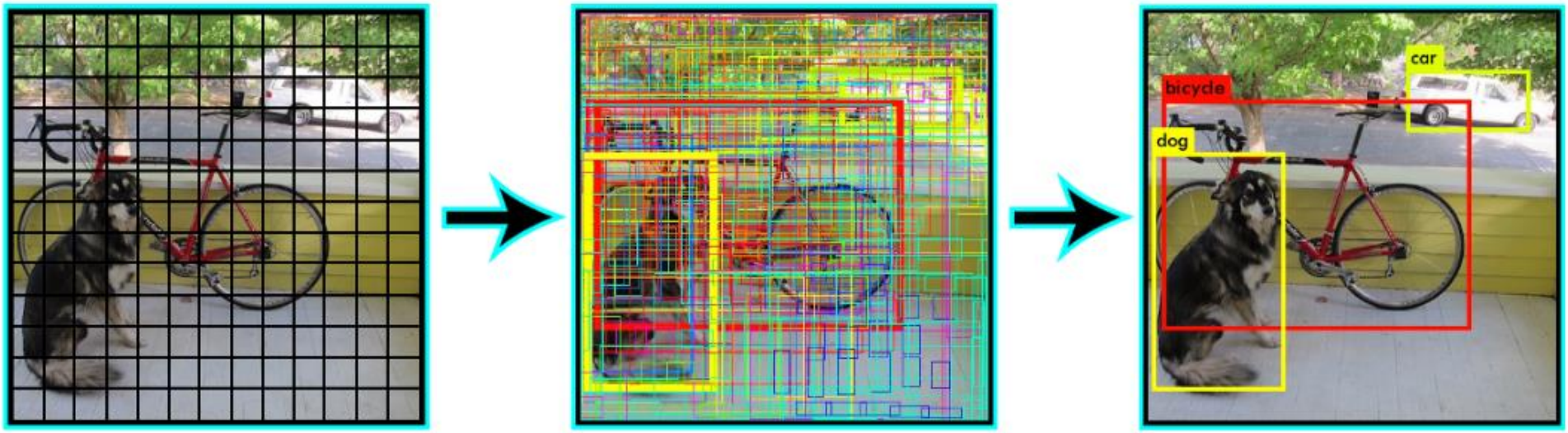
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2$$

物体检测（YOLO）

- 保留confidence score大于0.25的bounding boxes
- 最后用Non-maximum suppression (NMS)



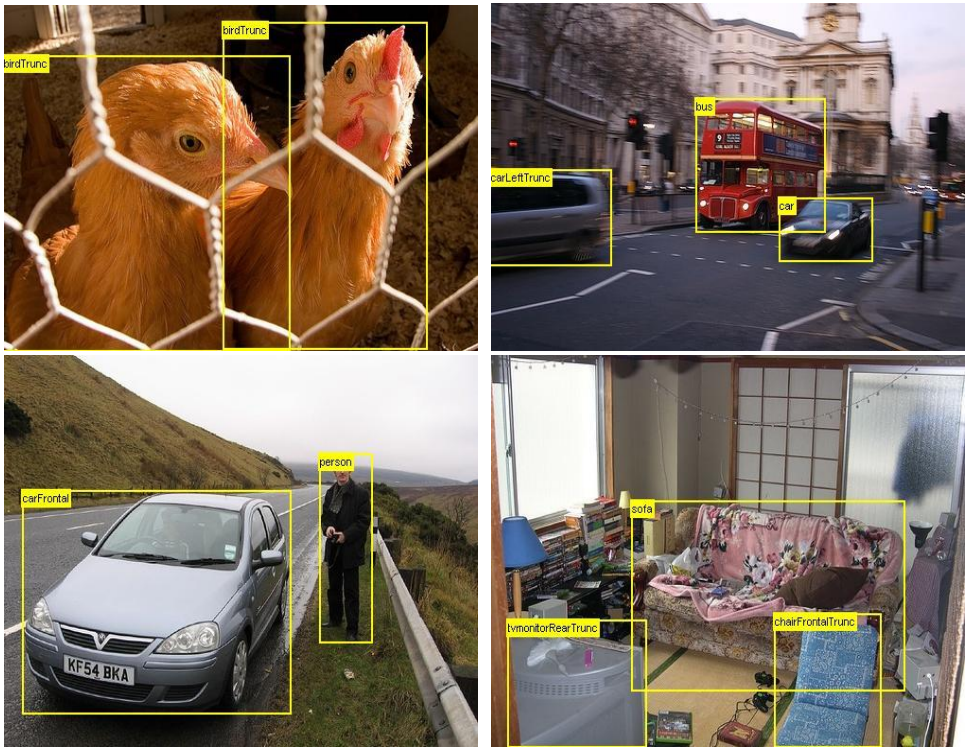
物体检测（YOLO）

- 预测速度快
  - 可以每秒处理45张图像
- 定位+识别都是在同一个网络中完成
  - 增加信息的耦合，提升模型精度
- YOLO基于整张图片的信息来预测bounding boxes
  - Region proposal的方法受限于固定的区域

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [30]	2007	16.0	100
30Hz DPM [30]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [37]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[27]	2007+2012	73.2	7
Faster R-CNN ZF [27]	2007+2012	62.1	18

物体检测（YOLO）

- YOLO具有较强的泛化能力
  - PASCAL VOC 2007（20个类）训练

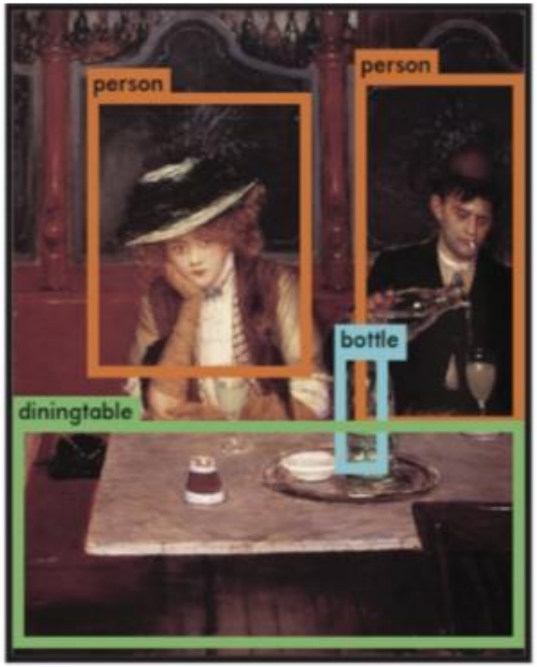


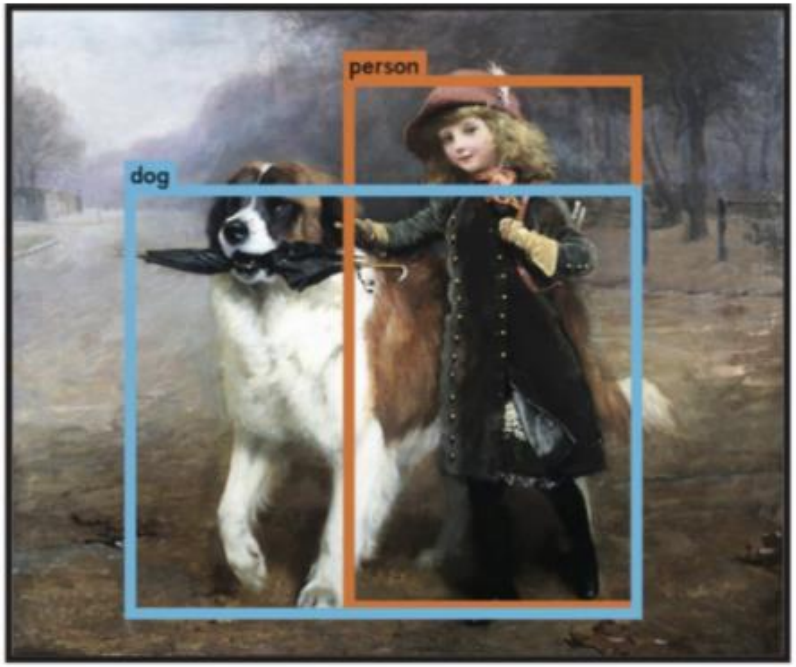
PASCAL VOC 2007

- *Person:* person
- *Animal:* bird, cat, cow, dog, horse, sheep
- *Vehicle:* aeroplane, bicycle, boat, bus, car, motorbike, train
- *Indoor:* bottle, chair, dining table, potted plant, sofa, tv/monitor

物体检测（YOLO）

- YOLO具有较强的泛化能力
  - 艺术图像中的物体检测





物体检测（YOLOv2）

- Redmon和Farhadi在2016年提出
- CVPR 2017 Best Paper Honorable Mention
- 跟YOLO相比的改动：
  - Batch normalization [1]
    - 在每一个conv layer后面加BN（提升2%左右的acc）
    - BN：对每一层的输入进行0均值、1方差的归一化，保证每层的输入数据分布是稳定的

$$\hat{x}^{(k)} = \frac{x^k - E[x^k]}{\sqrt{Var[x^k]}}$$

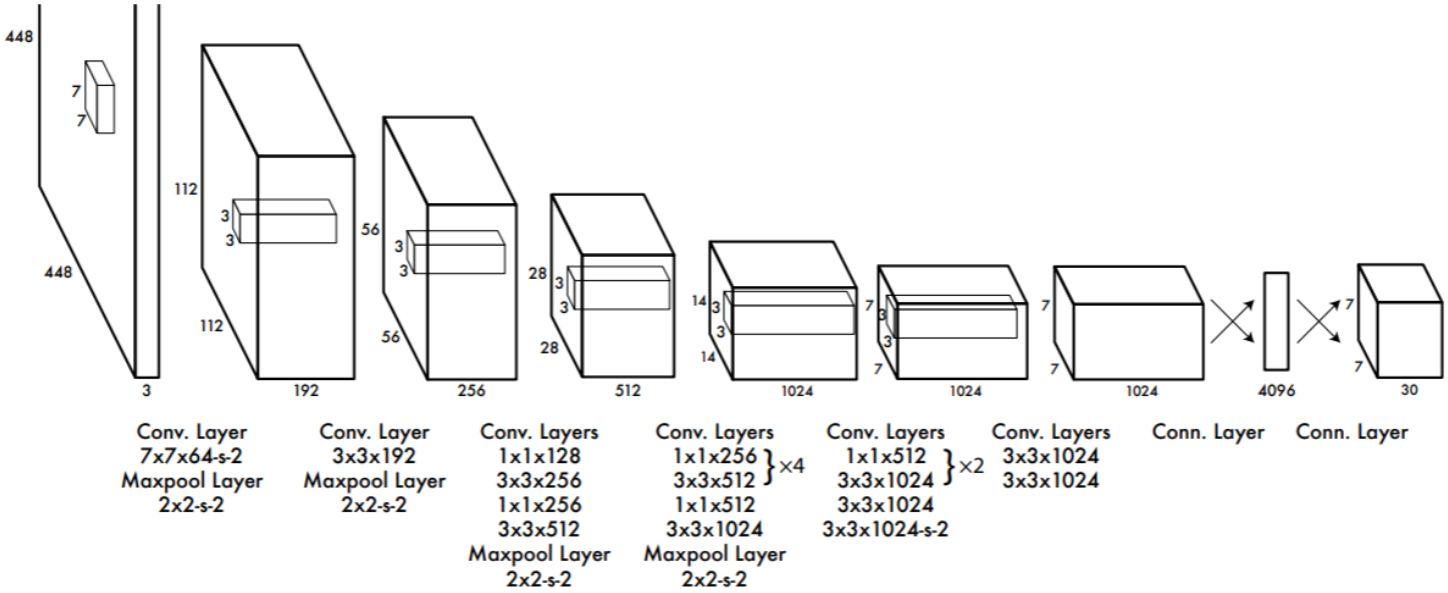
缺点：每层的数据分布被固定，使得这种分布可能不一定是前面一层的要学习到的数据分布，这样强行归一化就会破坏掉刚刚学习到的特征

解决办法：学习γ, β，还原上一层应该学到的数据分布 $y^{(k)} = \gamma^k \hat{x}^{(k)} + \beta^{(k)}$

[1] Ioffe and Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ArXiv, 2015.

物体检测（YOLOv2）

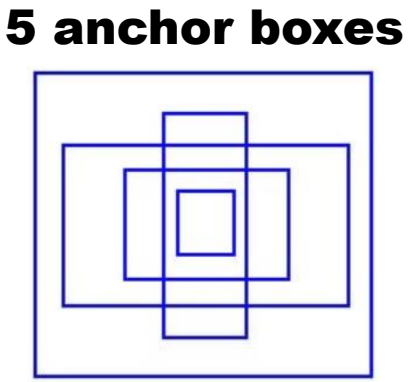
- 跟YOLO相比的改动：
  - High-resolution classifier（提升4%左右的acc）
    - YOLO的input先用224x224，然后增加到448x448来fine-tuning

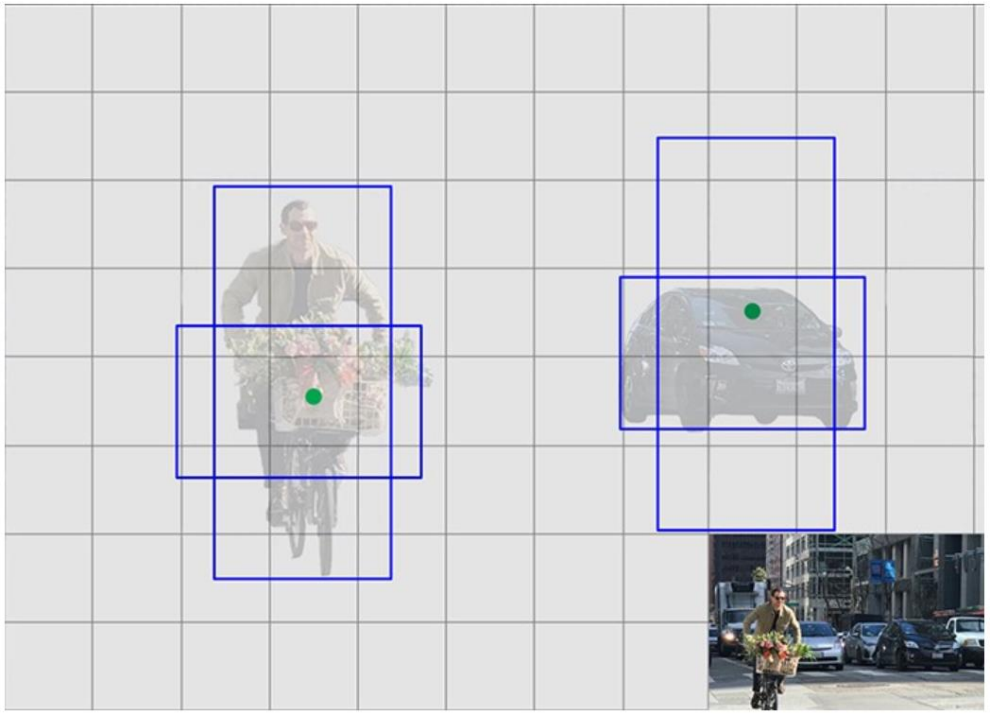


    - YOLOv2直接用448x448来训练base model（基于ImageNet），然后fine-tuning detection model

物体检测（YOLOv2）

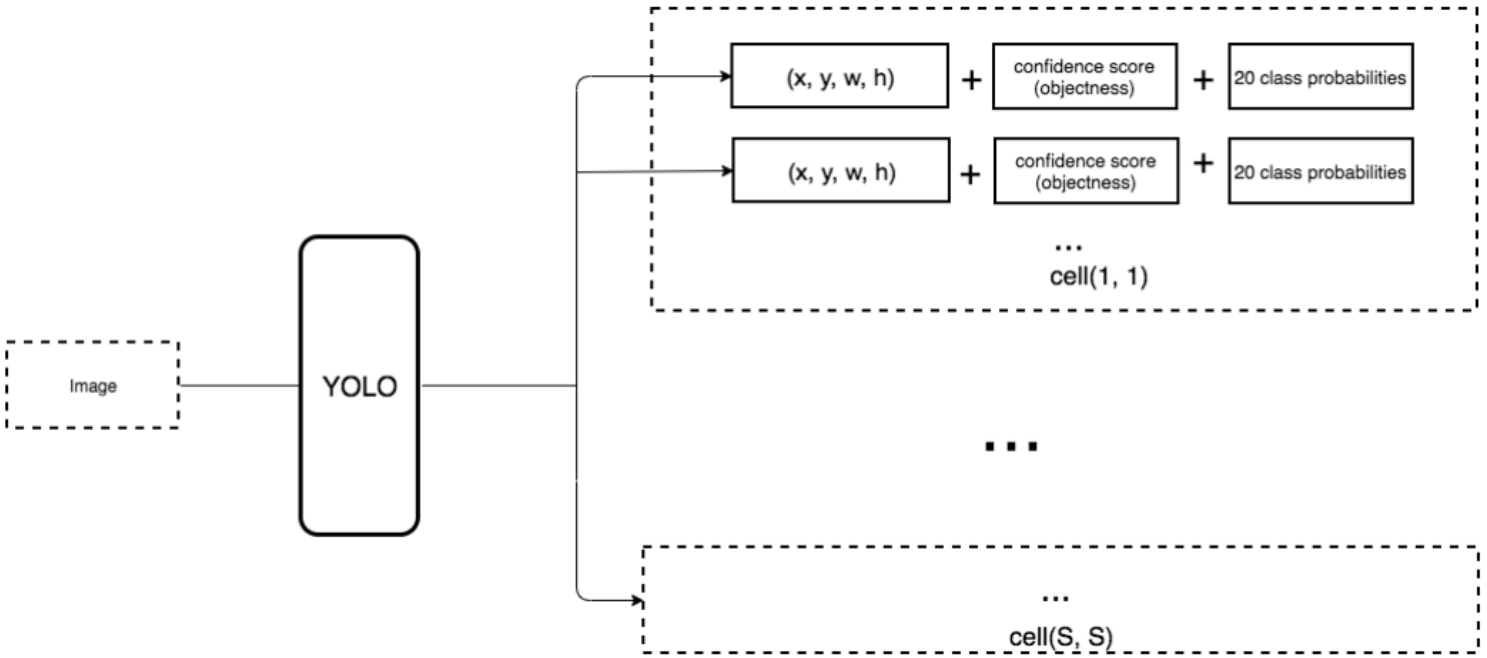
- 跟YOLO相比的改动：
  - Convolutional with Anchor Boxes
    - YOLO一开始随机生成bounding boxes，不稳定
    - YOLOv2定义大小不同的anchor boxes来降低随机性
    - 基于RPN的思想，优化location offsets，而不是box本身的坐标





物体检测（YOLOv2）

- 跟YOLO相比的改动：
  - Convolutional with Anchor Boxes
    - Classification从对grid cell到对bounding box
    - 对每一个bounding box的prediction，4个坐标值，1个confidence score，20个类别的概率，一共25个值。若一个grid cell有5个bounding box，则有125个值。



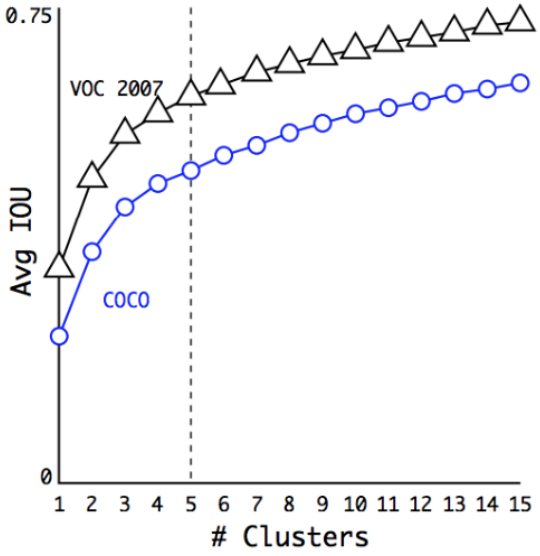
物体检测（YOLOv2）

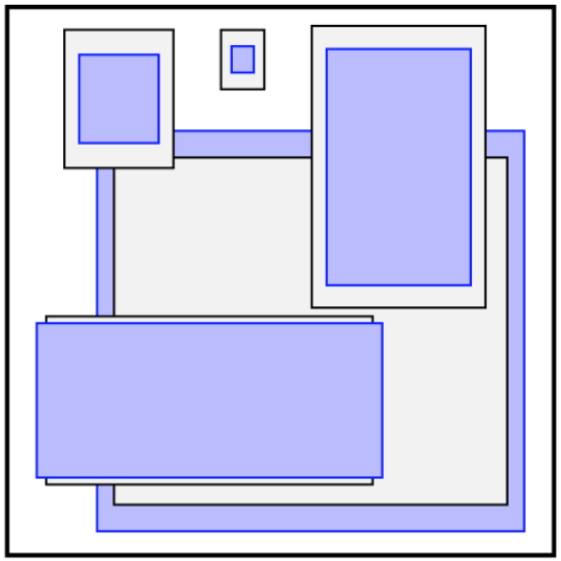
- 跟YOLO相比的改动：
  - Convolutional with Anchor Boxes
    - 把input image大小从448 x 448改为416 x 416
    - 把grid cells从7 x 7变成13 x 13
    - Anchor boxes个数为5（对比之前的random选取）
    - Accuracy 从69.5%减少到69.2%
    - Recall 从81%提升到88%

物体检测（YOLOv2）

- 跟YOLO相比的改动：
  - Dimension Clusters
    - Faster RCNN是预先人为定义好anchor box（或称为prior）的尺寸
    - YOLOv2不是手动选择priors，而是在训练集中的ground truth bounding boxes上运行K-means聚类，以自动找到好的先验
    - K-means的距离准则（distance metric）基于IoU来计算（假设所有的ground truth bounding boxes的中心点重合在一起）

$$d(box, centroid) = 1 - IOU(box, centroid)$$





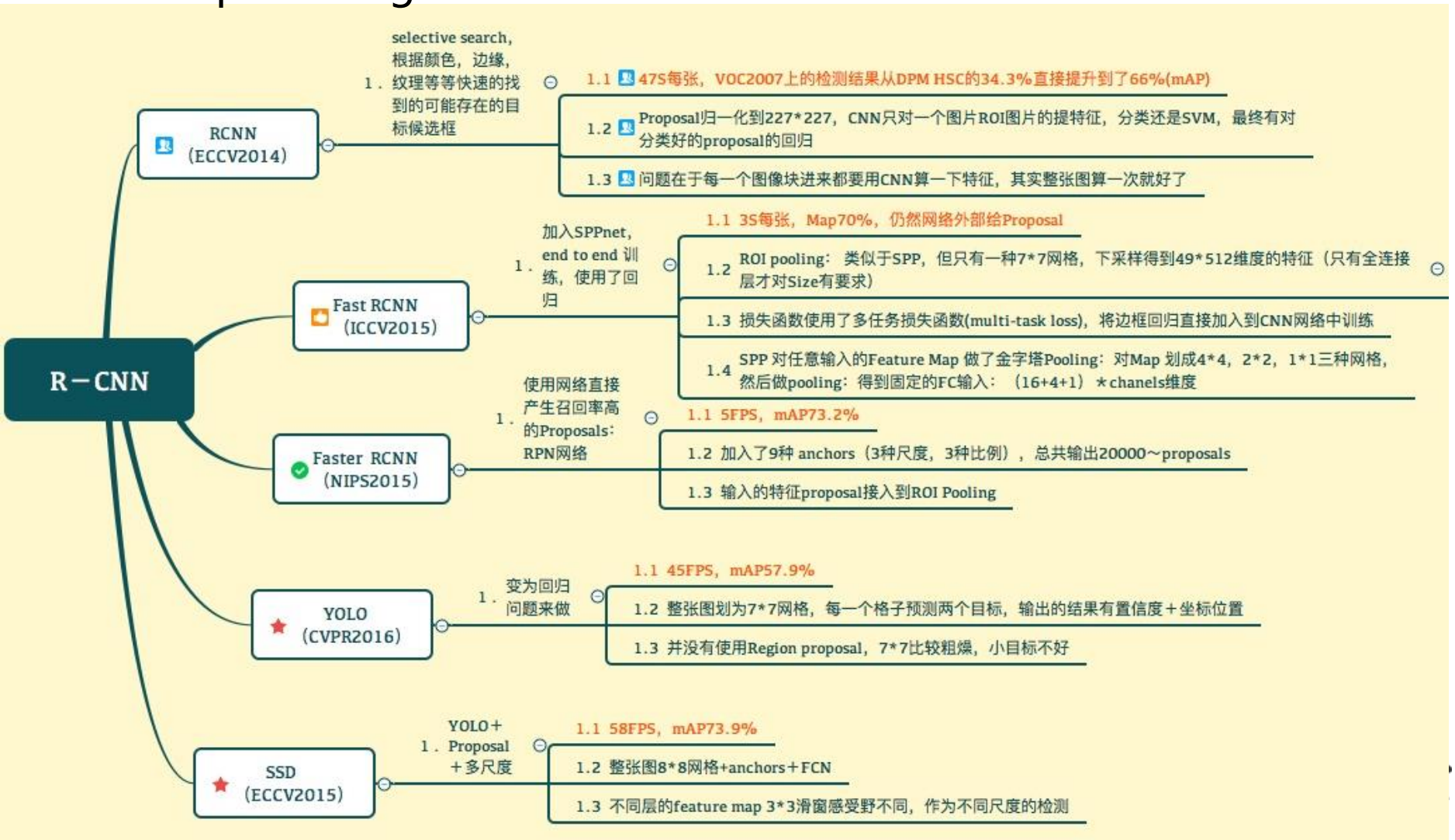
物体检测（YOLOv2）

- YOLOv2 vs. YOLO：加入了更多的tricks

	YOLO							YOLOv2
batch norm?		✓	✓	✓	✓	✓	✓	✓
hi-res classifier?			✓	✓	✓	✓	✓	✓
convolutional?				✓	✓	✓	✓	✓
anchor boxes?				✓	✓			
new network?					✓	✓	✓	✓
dimension priors?						✓	✓	✓
location prediction?						✓	✓	✓
passthrough?							✓	✓
multi-scale?								✓
hi-res detector?								✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	**78.6**

物体检测

- Deep learning based methods



语义分割

图像/物体分割

- 图像/物体分割是把图图像分成若干个特定的、具有独特性质的区域并提出感兴趣目标的技术和过程

CAT, DOG, DUCK

物体分割（传统方法）

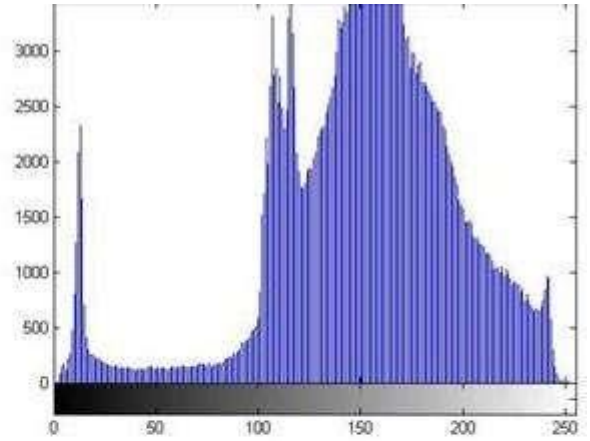
- 阈值法
- 边缘检测法
- 主动轮廓模型
- 分水岭算法
- 区域生长法
- 随机决策森林
- 基于图论的图像分割
- 马尔可夫随机场与条件随机场

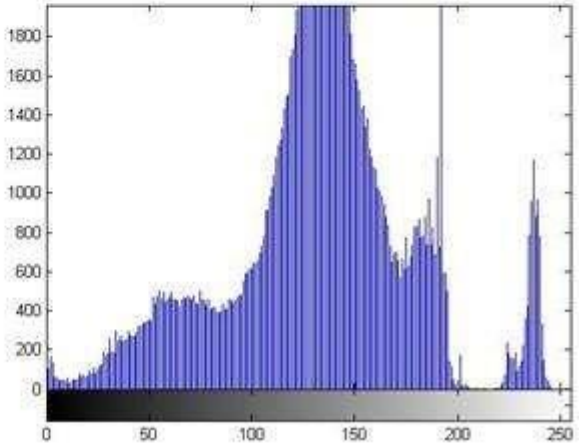
物体分割（阈值法）

阈值法：阈值分割方法作为一种常见的区域并行技术，就是用一个或几个阈值将图像的灰度直方图分成几个类，认为图像中灰度值在同一类中的像素属于同一物体。

阈值分割方法的关键和难点是如何取得一个合适的阈值。而实际应用中，阈值设定易受噪声和光亮度影响。

阈值分割的优点是计算简单、运算效率较高、速度快。



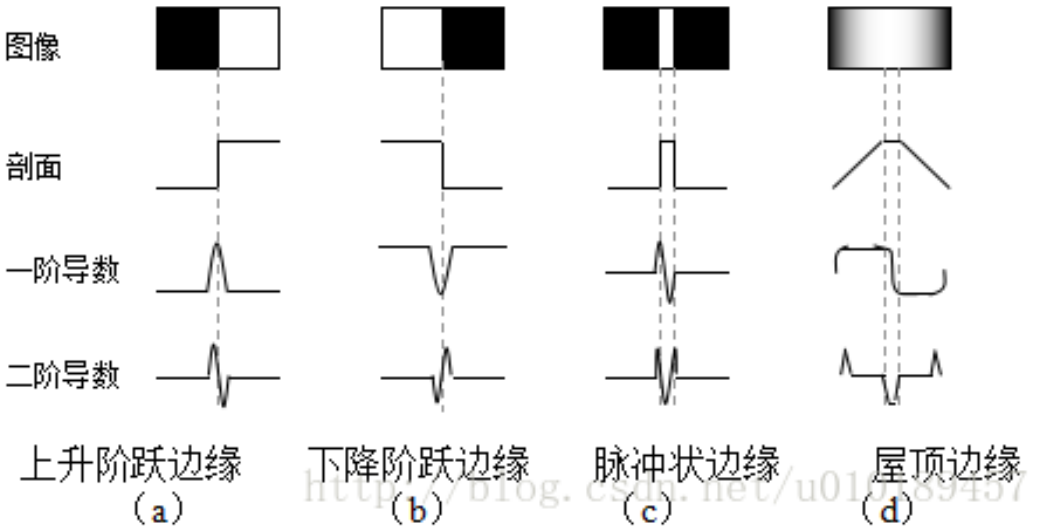


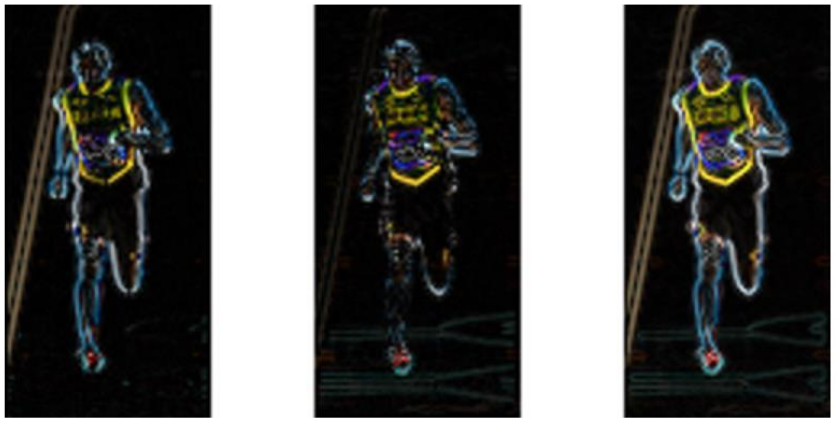
物体分割（边缘检测法）

基于边缘的分割方法：基于边缘检测的分割方法试图通过检测包含不同区域的边缘来解决分割问题，是最常用的方法之一。通常不同的区域之间的边缘上像素灰度值的变化往往比较剧烈，这是边缘检测得以实现的主要假设之一。

常用灰度的一阶或者二阶微分算子进行边缘检测。常用的微分算子有一次微分(sobel算子，Robert算子等)，二次微分(拉普拉斯算子等)和模板操作(Prewit算子，Kirsch算子等)。

基于边缘的分割方法其难点在于边缘检测时抗噪性和检测精度之间的矛盾

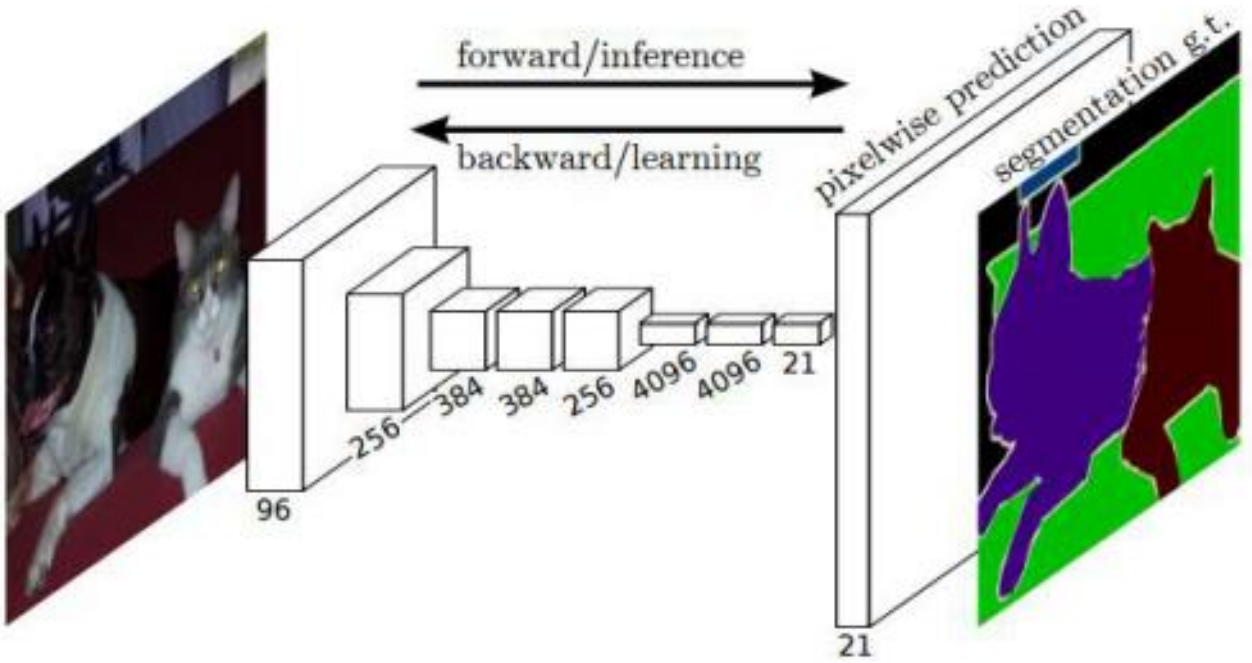




Left : Absolute value of x-gradient. Center : Absolute value of y-gradient.
Right : Magnitude of gradient.

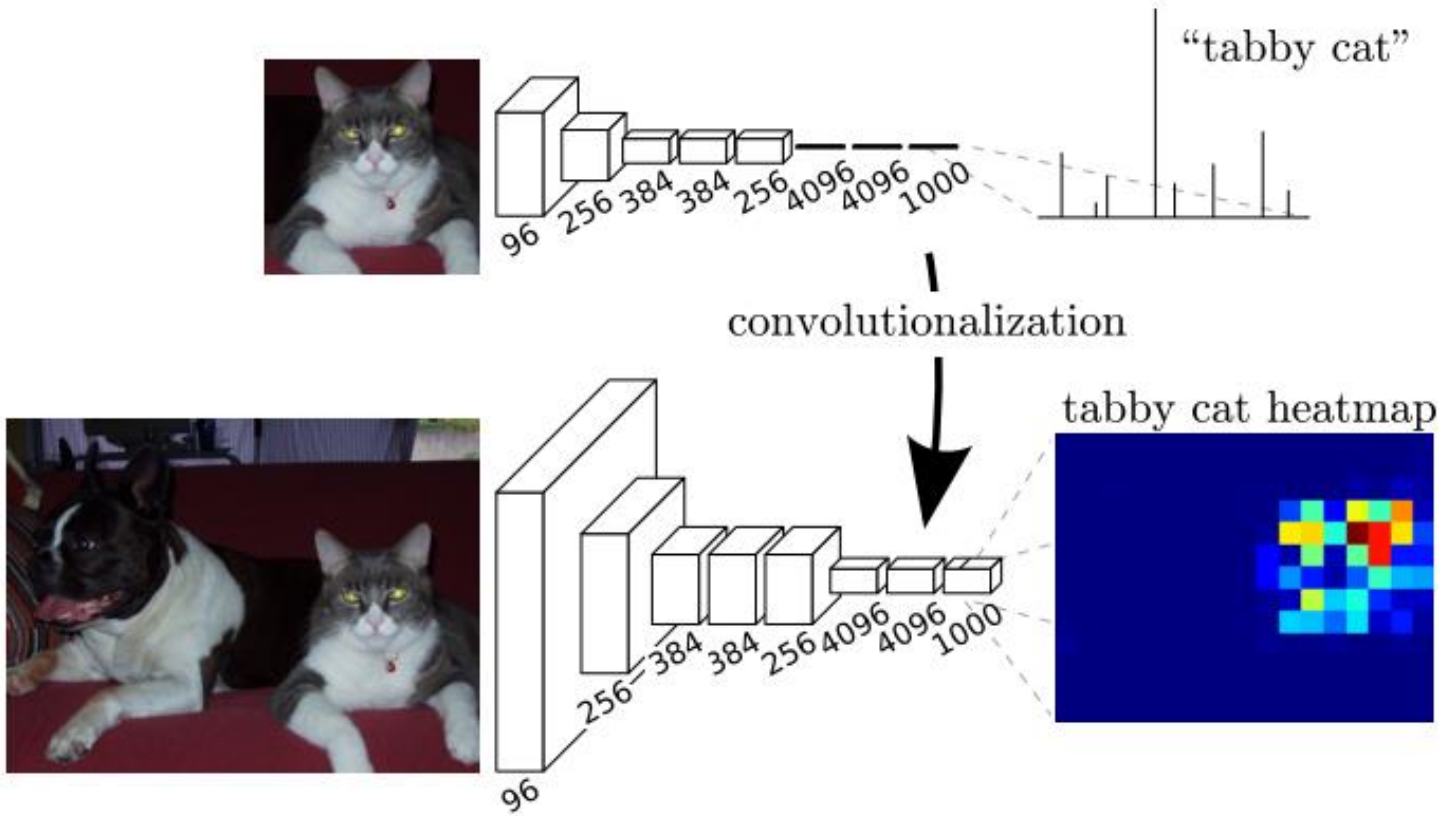
物体分割（FCN）

- Jonathan Long, et al. CVPR 2015
- Fully Convolutional Networks (FCN) 追求的是，输入是一张图片，输出也是一张图片，学习像素到像素的映射



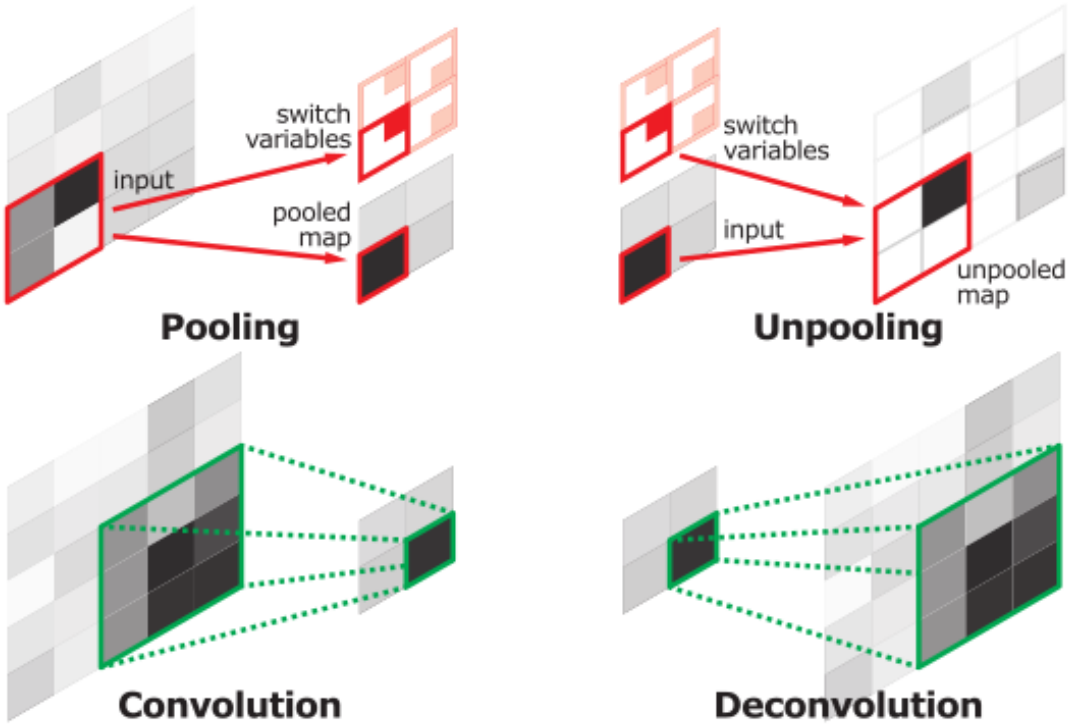
物体分割（FCN）

- FCN将传统CNN中的全连接层转化成一个个的卷积层。如下图所示，在传统的CNN结构中，前5层是卷积层，第6层和第7层分别是一个长度为4096的一维向量，第8层是长度为1000的一维向量。FCN将这3层表示为卷积层。所有的层都是卷积层，故称为**全卷积网络**。



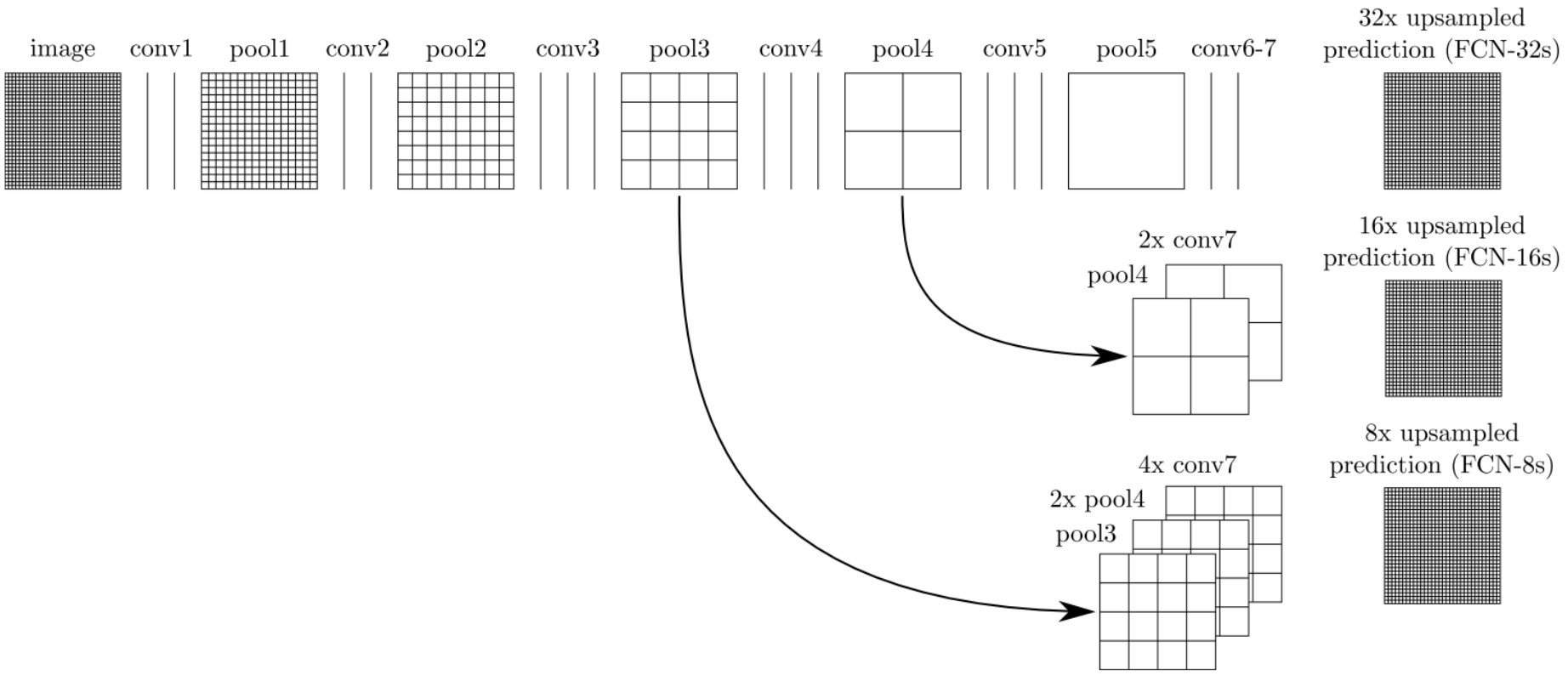
物体分割（FCN）

- 经过多次卷积以后，得到的图像越来越小,分辨率越来越低，为了从这个分辨率低的粗略图像恢复到原图的分辨率，FCN使用了上采样
- 例如经过5次卷积(以及pooling)以后，图像的分辨率依次缩小了2，4，8，16，32倍。对于最后一层的输出图像，需要进行32倍的上采样，以得到原图一样的大小。



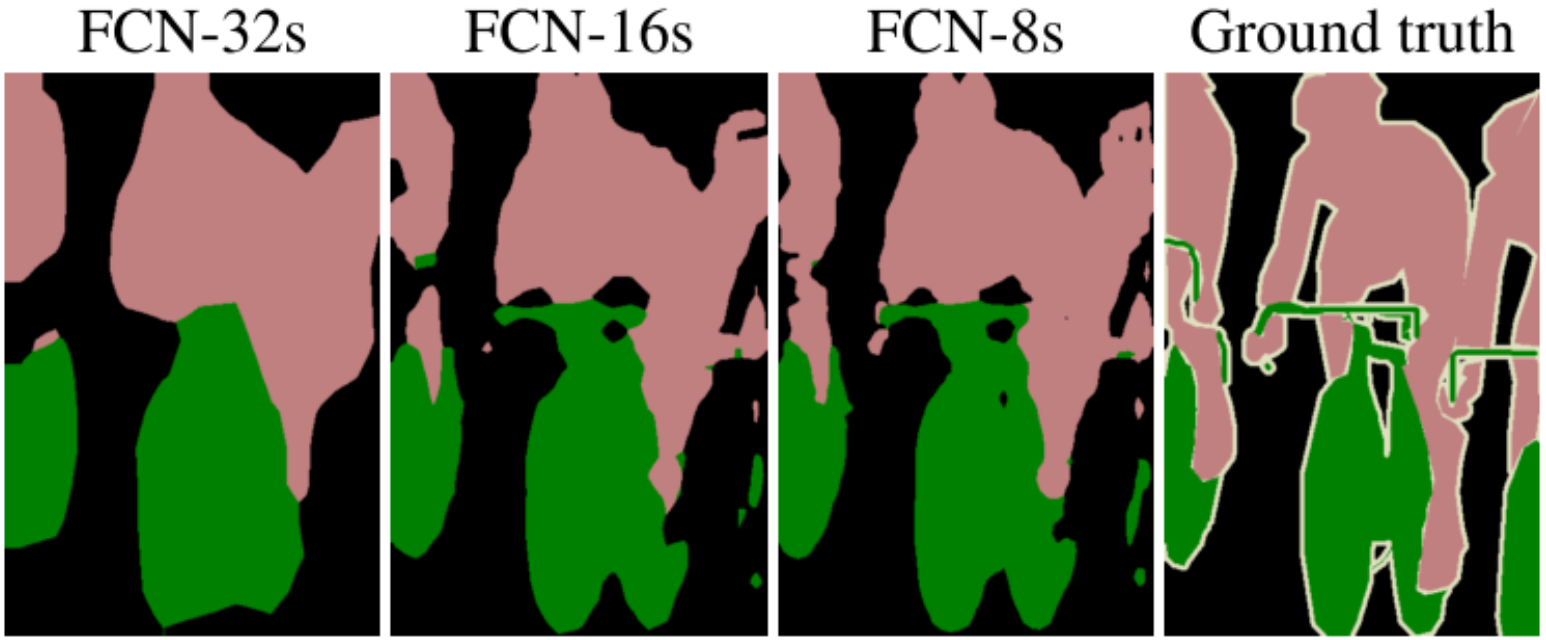
物体分割（FCN）

- 对第5层的输出（32倍放大）反卷积到原图大小，得到的结果还是不够精确，一些细节无法恢复。于是可以将第4层的输出和第3层的输出也依次反卷积，分别需要16倍和8倍上采样，结果就精细一些

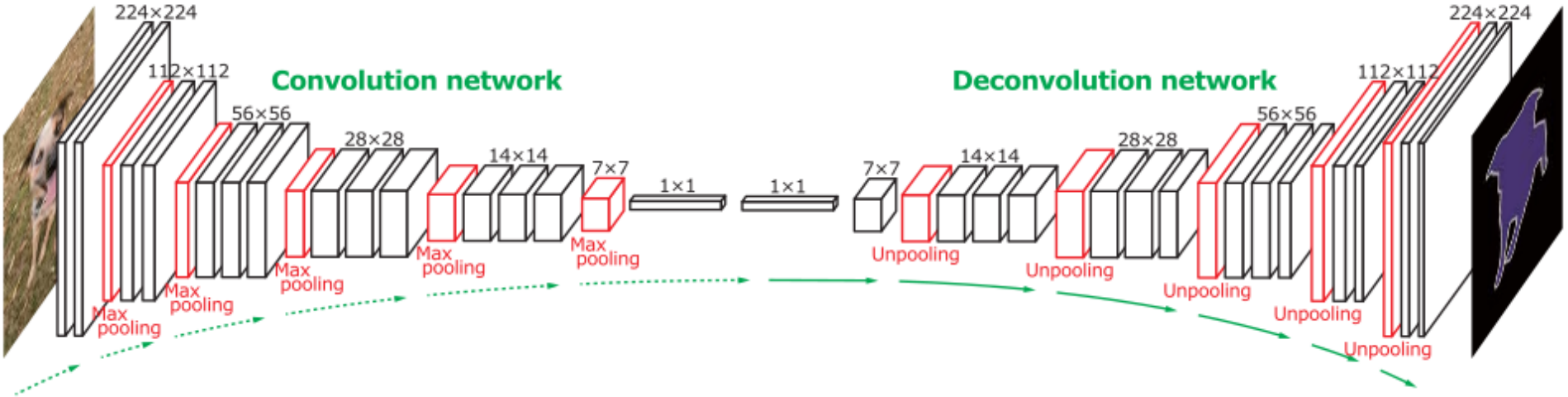


物体分割（FCN）

- 下图是32倍，16倍和8倍上采样得到的结果的对比，可以看到它们得到的结果越来越精确：



物体分割（FCN）



物体分割（FCN）

- 跟传统方法相比的优点：
  - 可以接受任意大小的输入图像，而不用要求所有的训练图像和测试图像具有同样的尺寸。
  - 更加高效，因为避免了由于使用像素块而带来的重复存储和计算卷积的问题
- 缺点：
  - 得到的结果还是不够精细。进行8倍上采样虽然比32倍的效果好了很多，但是上采样的结果还是比较模糊和平滑，对图像中的细节不敏感。
  - 对各个像素进行分类，没有充分考虑像素与像素之间的关系，忽略了在通常的基于像素分类的分割方法中使用的空间规整（spatial regularization）步骤，缺乏空间一致性

回顾

- 计算机视觉
  - 图像处理
  - 三维视觉
  - 图像理解
    - 图像分类
    - 目标检测：R-CNN，Fast R-CNN, Faster RCNN, YOLO-v1, YOLO-v2
    - 图像分割：FCN