Visual Object Detection

视觉对象检测

智能系统实验室
清华大学基础工业训练中心

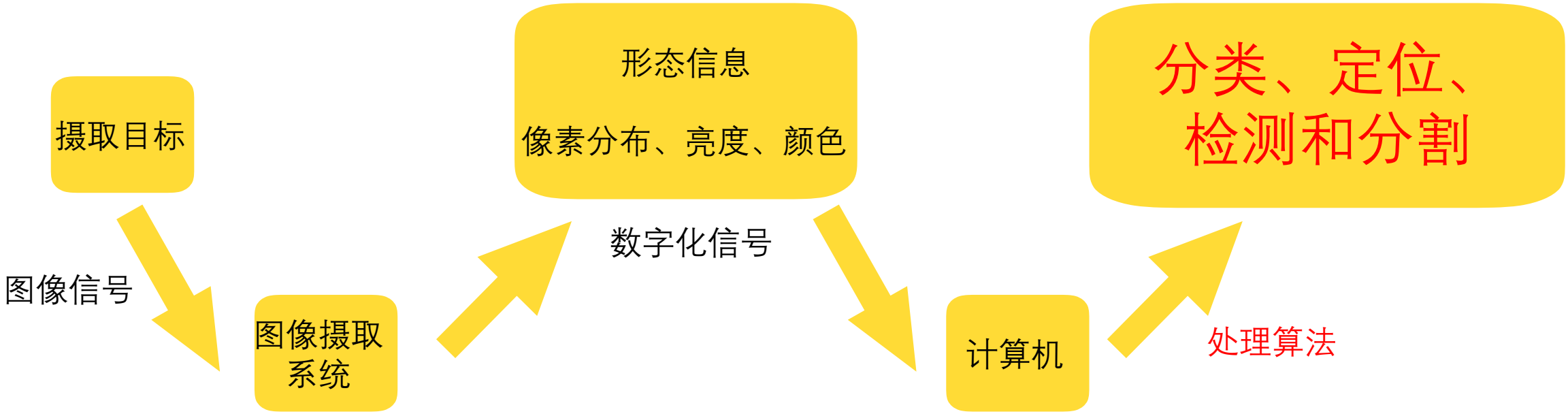
目录

- 计算机视觉的任务
- 计算机视觉的识别指标
- 视觉对象检测的方法
- 图像语义分割的方法

计算机视觉的任务

计算机视觉

- 计算机视觉就是用计算机代替人眼来做测量和判断（简单说来）。
- 计算机视觉是人工智能快速发展的一个分支。
- 计算机视觉的主要任务包括：分类、定位、检测和分割



计算机视觉的任务（Visual Task）

Visual Object, 对象，又称物体，目标等

- 分类 Image Classification
- 对象类别

- 定位localization
- 对象位置

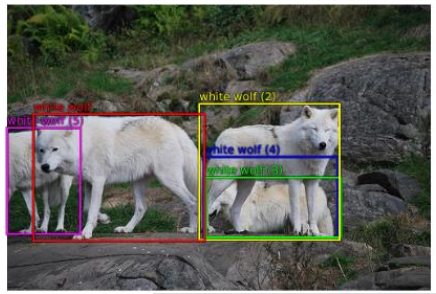
- 检测detection
- 对象类别与位置

- 分割segmentation
- 场景解析与标记

Top 5:
pencil sharpener
pool table
hand blower
oil filter
packet

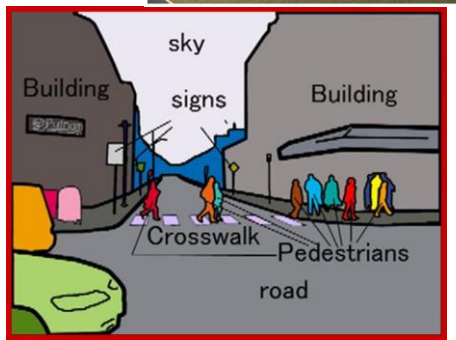
Groundtruth:
pencil sharpener

ILSVRC2012_val_00010000.JPEG



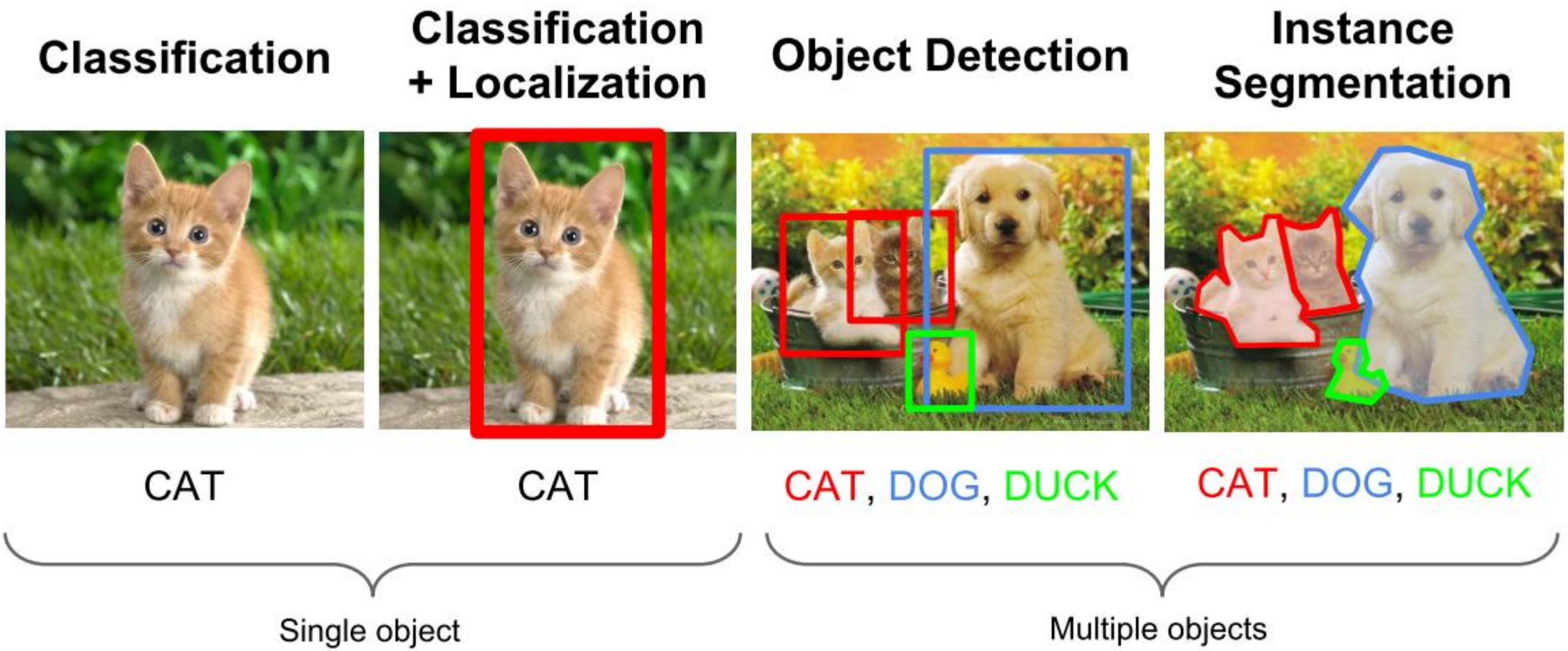
Groundtruth:
white wolf
white wolf (2)
white wolf (3)
white wolf (4)
white wolf (5)

Groundtruth:
tv or monitor
tv or monitor (2)
tv or monitor (3)
person
remote control
remote control (2)



困难

分类、定位、检测、分割


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

计算机视觉识别指标

OD Index

识别的指标

- 精确率（precision）是针对预测结果而言的，它表示的是预测为正的样本中有多少是真正的正样本。预测（分类）为正有两种可能：
  - 一种是把**正类预测为正类(TP)**，
  - 另一种是把**负类预测为正类(FP)**
- 召回率（recall）是针对原来的样本而言的，它表示的是样本中的正例有多少被预测正确了。预测（分类）为负有两种可能：
  - 一种是把**原来的负类预测成负类(TN)**，
  - 另一种是把**原来的正类预测为负类(FN)**
- 准确率(accuracy) 是指对于给定的测试数据集，分类器正确分类的样本数与总样本数之比。（也就是损失函数是0-1损失时测试数据集上的准确率）
- 精确率(precision) = TP/(TP+FP)
- 召回率(recall) = TP/(TP+FN)
- 准确率(accuracy) = (TP+TN)/(TP+FN+FP+TN) = = 预测对的/所有

举例说明

- 例子:
  - 假设我们手上100张样本图片，有70个正样本（猫图片），30个负样本（狗图片），
  - 计算机视觉的任务要找出所有的正样本（猫图片），
  - 识别系统查找出50个（猫图片），其中只有40个是真正的正样本（猫图片）。
- 计算识别指标:
  - TP: 将正类预测为正类数 40
  - FN: 将正类预测为负类数 30
  - FP: 将负类预测为正类数 10
  - TN: 将负类预测为负类数 20
- 精确率(precision) = TP/(TP+FP) = 80%
- 召回率(recall) = TP/(TP+FN) = 4/7
- 准确率(accuracy) = 预测对的/所有 = (TP+TN)/(TP+FN+FP+TN) = 60%

对象检测的识别精确率指标

- 常用的识别精确率指标：
  - 平均精确率均值mAP
  - PR曲线的覆盖率AUC：P为精确率，R为召回率

平均精确率均值mAP（识别准确率指标之一）

- 平均精确率均值mAP（Mean Average Precision）是对象检测研究中常用数据集VOC 2007所采用的评价指标，被该领域的研究者们广泛使用
- VOC 2007对于mAP的数学定义如下，其中*p*和*r*分别表示模型在取不同的阈值参数时的精确率（Precision）和召回率（Recall）

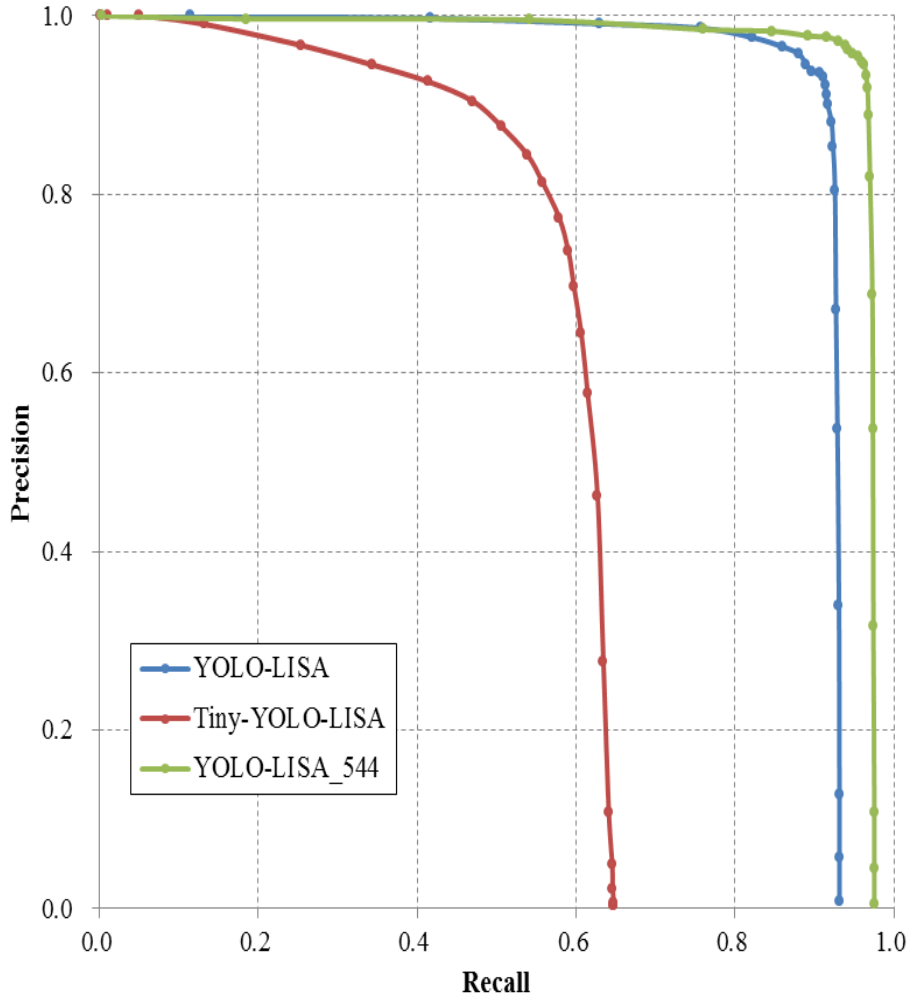
$$\mathrm{AP} = \frac{1}{11} \sum_{r \in \{0, 0.1, \cdots, 1\}} \max_{\tilde{r}:\, \tilde{r} \geq r} p(\tilde{r})$$

$$\mathrm{mAP} = \frac{1}{\#\mathrm{classes}} \sum_{c \in \mathrm{classes}} \mathrm{AP}(c)$$

  - mAP指标度量模型在不同情况下的平均精确率，是对精确率和召回率之间平衡取舍问题的一种有效处理方式。
  - mAP越高，说明模型的检测准确性越好。

PR曲线的AUC指标（识别准确率指标之二）

- AUC=Area under the PR Curve
- 2015年VIVA（Vision for Intelligent Vehicles and Applications）交通标志检测比赛。
- VIVA主办方采用了PR曲线（Precision–Recall Curve）的面积覆盖率AUC（Area under Curve）作为对象检测的识别准确性的评价指标。
- 面积覆盖率（AUC）越高，则对象检测的识别准确性越好。



最佳工作状态

- 针对具体应用场景，对精确率和召回率之间进行一个平衡取舍，从而选择合适的阈值参数，使对象检测器处于最佳的工作状态。

- F_1的数学含义其实就是精确率P和召回率R的调和平均数，综合考虑了二者的影响。

$$F_1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2 \cdot P \cdot R}{P + R}$$

视觉对象检测的算法

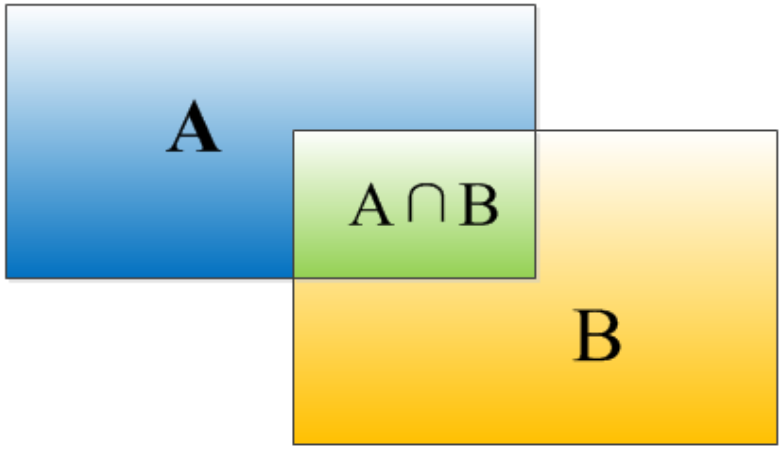
Visual object detection algorithm

IOU（重叠联合比）

- $\mathrm{IOU}_{\mathrm{pred}}^{\mathrm{truth}}$ 表示的是预测框（Prediction）和真实框（Ground Truth）之间的重叠联合比（Intersection over Union）

- IOU定义了2个边界框（ bounding box ）(就是恰好框住对象的矩形框)的重叠度，计算为相交面积（∩）/相并面积（∪）

- $\mathrm{IOU}_{\mathrm{pred}}^{\mathrm{truth}} = \dfrac{\text{Area of Intersection}}{\text{Area of Union}}$



视觉对象检测的错误类型

- 对于模型给出的检测结果，都会根据以下标准，被判定为其中的一种：
- 正确的
  - 正确（Correct）：类别正确，IOU > 0.5
- 错误的
  - 定位错误（Localization）：类别正确，0.1 < IOU < 0.5
  - 相似性错误（Similar）：类别相似，IOU > 0.1
  - 其他错误（Other）：类别错误，IOU > 0.1
  - 背景误认（Background）：IOU < 0.1

视觉对象检测方法

- **R-CNN**
  - Region based convolutional networks for accurate object detection and segmentation, TPAMI, 2015.
  - Rich feature hierarchies for accurate object detection and semantic segmentation, CVPR 2014.
- **Fast R-CNN**
  - Fast R-CNN, ICCV 2015.
- **Faster R-CNN**
  - Faster R-CNN, NIPS, 2015.
- **YOLOv1-->YOLOv3**
  - You Only Look Once: Unified, Real-Time Object Detection, CVPR 2016.
- **SSD**
  - SSD: Single Shot MultiBox Detector, ECCV 2016.

http://wiki.icenter.tsinghua.edu.cn/icenterwiki/index.php/计算机视觉

参考资料

- [1] R. Girshick et al., "Rich feature hierarchies for accurate object detection and semantic segmentation", Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 580-587, 2014.
- Region based convolutional networks for accurate object detection and segmentation, TPAMI 2016.
- [2] Girshick R. Fast R-CNN[C]. IEEE International Conference on Computer Vision, 2015: 1440-1448.
- [3] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2016, 39(6):1137-1149.

R-CNN

- R-CNN: 全名叫Regions with CNN features / Region-based Convolutional Neural Networks
- 将卷积神经网络应用region proposal的策略，自底下上训练可以用来定位目标物和图像分割
- 当标注数据是比较稀疏的时候，在有监督的数据集上训练之后到特定任务的数据集上fine-tuning（微调参数，总体网络架构不变了）可以得到较好的性能。
- 用ImageNet上训练好的模型，在需要训练的数据上fine-tuning一下，检测效果很好。
- 突破性：当时在Pascal VOC数据集上测试性能最好，达到的效果比当时最好的DPM方法 mAP还要高上20点。
  https://www.rossgirshick.info/

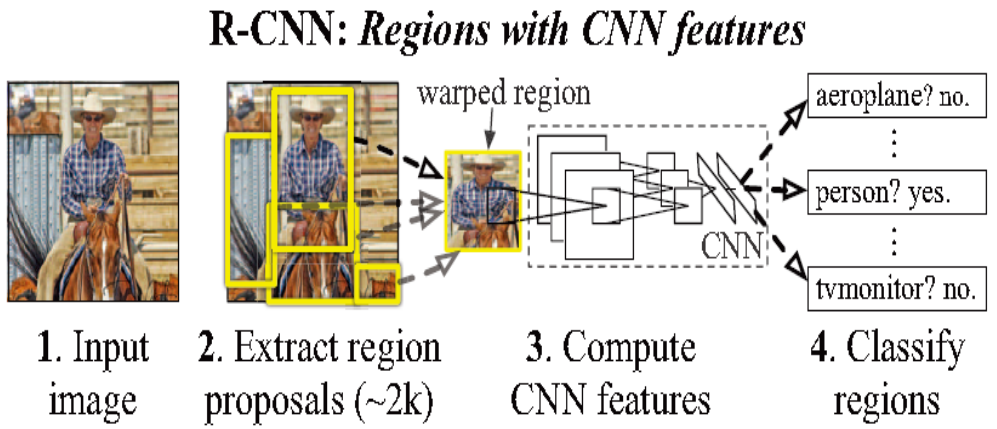


Figure 1: Object detection system overview. Our system (1) takes an input image, (2) extracts around 2000 bottom-up region proposals, (3) computes features for each proposal using a large convolutional neural network (CNN), and then (4) classifies each region using class-specific linear SVMs. R-CNN achieves a mean average precision (mAP) of **53.7% on PASCAL VOC 2010**. For comparison, [34] reports 35.1% mAP using the same region proposals, but with a spatial pyramid and bag-of-visual-words approach. The popular deformable part models perform at 33.4%.

R-CNN

- 输入图像，提取提炼区域（region）：
  - 用选择性搜索（selective search）的算法去搜索一个‘fast mode’(快速模式)，对每一个提出的可能有对象的图像区域提取出一个4096维的特征向量。
  - 对于不是标准227*227像素的正方形的区域，使其标准化。最简单的方法是膨胀（dilate，形态学算法）其最小外边框（设宽度=16 pixels），使整幅图像大小合适。
- 计算CNN特征：
  - CNN网络架构： 5个卷积层（Convolution Layers），2个全连接(Fully Connected Layers), 正如Yann Le Cun之前提出的LeNet算法。
- 区域分类：
  - 对每一个类预先训练好一个支持向量机（SVM），然后对之前提炼出来的特征向量（feature vector）用对应类的SVM去“打分”。
  - 贪心思想的“非极大值抑制”(non-maximum suppression)算法：如果一个区域和一个有更高打分的区域有交集（Intersection-over-Union （IoU））并且IoU的值>某个阈值，那么这个区域（得分相对低的）将被舍弃。

R-CNN的缺点

- 训练分为3个步骤的流水线（对候选区提取特征的微调卷积网络，训练线性SVM作为对象探测器，处理proposal计算卷积特征，边界框（BBOX）回归运算）；
- 训练时间和空间开销大。要从每一张图像上提取大量proposal，还要从每个proposal中提取特征，并存到磁盘中；
- 测试时间开销大。要从每个测试图像上，提取大量proposal，再从每个proposal中提取特征来进行检测过程；
- 速度慢。一个原因是在前向运算时对每一个候选区域的对象分别计算，并没有用共享权值或共享模型参数的方法加快。

Fast R-CNN改进R-CNN

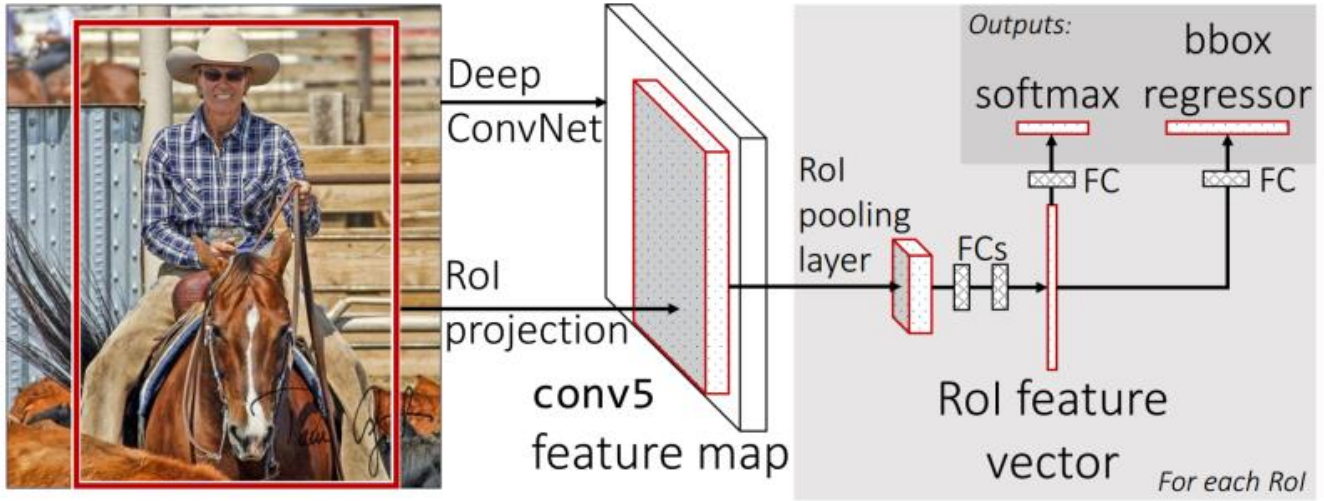
- 1. 比R-CNN更高的检测质量（mAP）；
- 2. 把多个任务的损失函数写到一起，实现单级的训练过程;
- 3. 在训练时可更新所有的层;
- 4. 不需要在磁盘中存储特征。

Fast R-CNN

1. 使用外部算法（选择性搜索SS ）来找出候选区域（2000个object proposal），找出感兴趣的区域（Regions of Interest, RoI），映射到特征空间里；
2. 缩放图片的scale得到图片金字塔，得到conv5的特征金字塔；
3. 对于每个scale的每个ROI，求取映射关系，在conv5中crop出对应的patch；并用一个单层的空间金字塔池化层（SPP） layer（称为RoI pooling layer）来统一到一样的尺度，因为后续的全连接层输入的所有向量有同样的大小；
4. 连续续经过两个全连接层得到特征，特征又分别共享到到两个新的全连接层，分别对应两个优化目标
   - 第一个优化目标是分类，使用softmax，
   - 第二个优化目标是边界框回归（bbox regression），使用了一个smooth的L1-loss（一次函数和小量时二次函数的结合）。

Fast R-CNN优点

- Fast R-CNN 实现了端到端的联合训练（end-to-end joint training） (single stage)
- R-CNN用SVM训练特征时需要中间大量的磁盘空间存放特征，Fast RCNN没有了SVM这一步，所有的特征都暂存在显存中，不需要额外的磁盘空间。
- R-CNN中因为ROI-centric的原因，测试时间开销大，Fast R-CNN进一步通过single scale(pooling->spp just for one scale) testing和SVD（奇异值分解）(降维)分解全连接来提速。

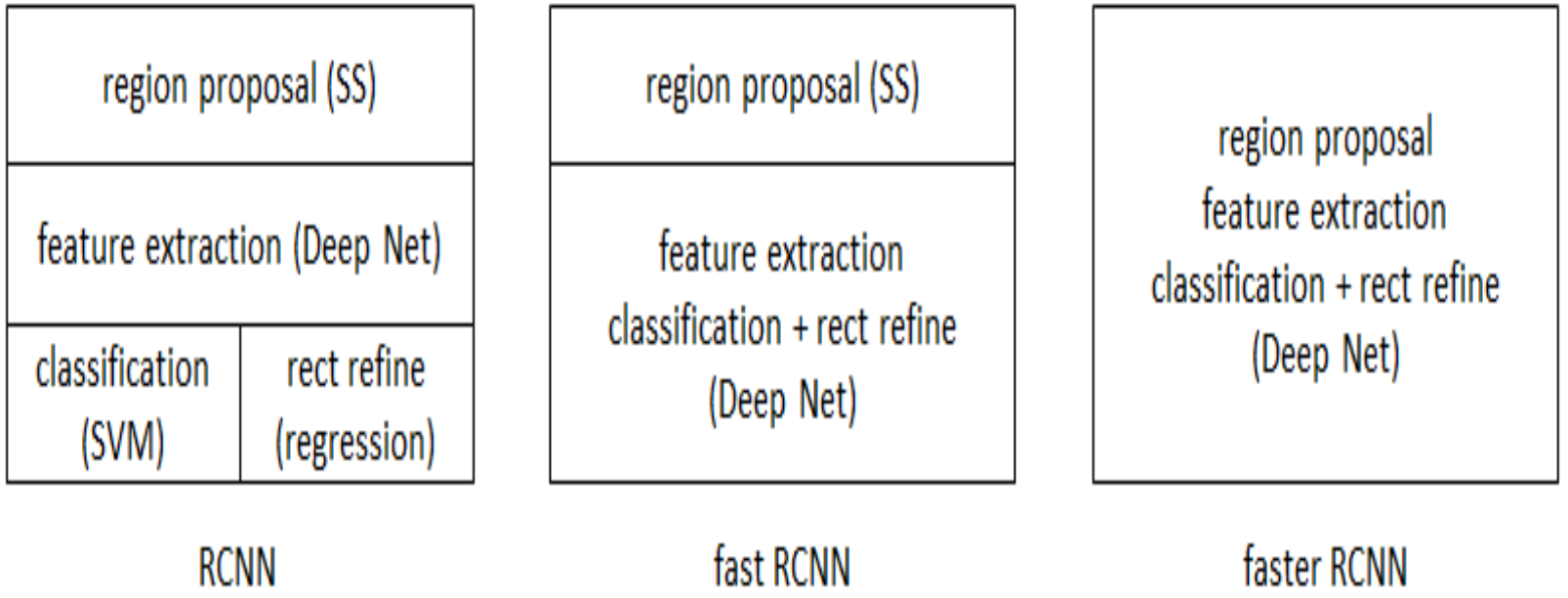


Faster R-CNN改进Fast R-CNN

- Faster R-CNN速度更快，精确度更高。
- Faster R-CNN中，每个网络可以独立训练或联合训练。
- 模型有4个损失函数：
  - RPN（区域生成网络）分类是否对象；
  - RPN 边界框提议；
  - Fast R-CNN 对象分类；
  - Fast R-CNN 边界框回归 。

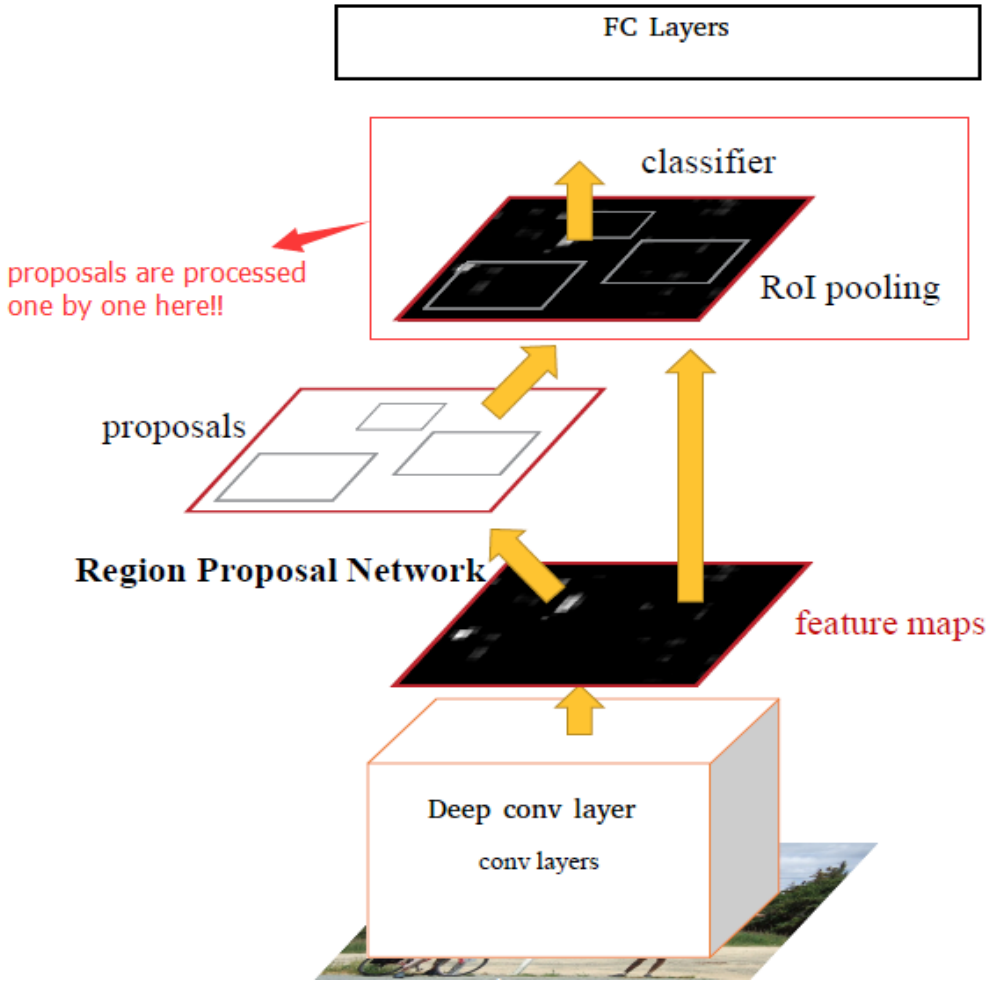
Faster R-CNN

- Faster RCNN可以简单地看做“RPN+fast R-CNN“的系统，用RPN代替fast R-CNN中的Selective Search方法。
- RPN区域生成网络



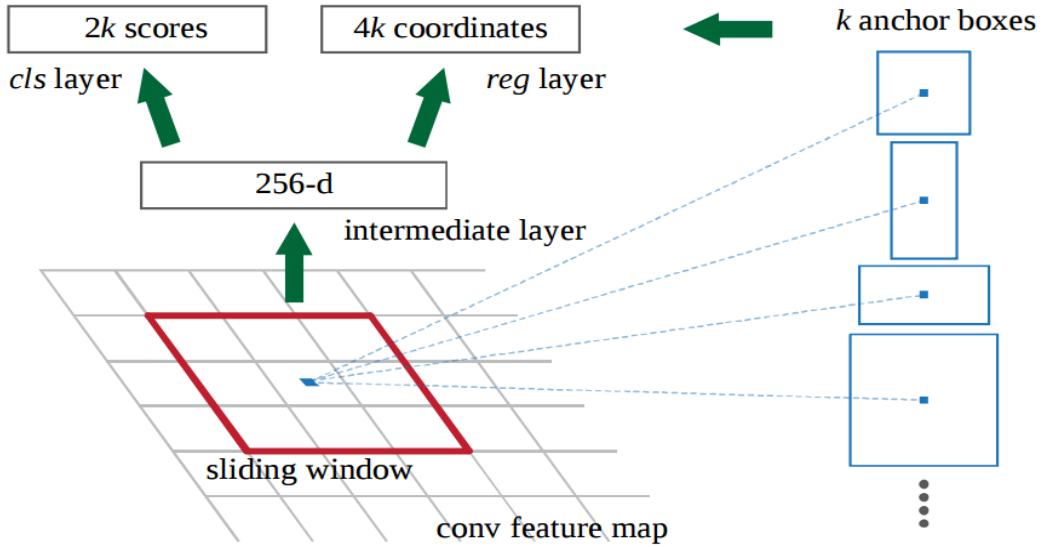
Faster R-CNN

- Faster R-CNN包含2个模块：
  - RPN(Region Proposal Network):在深度卷积层基础上给出一系列的矩形候选区域。
  - Fast R-CNN RoI 池化层：对每个proposal 区域进行分类，提取proposal定位。
- ***主要思想***是用最后一个卷积层去推断候选区域。



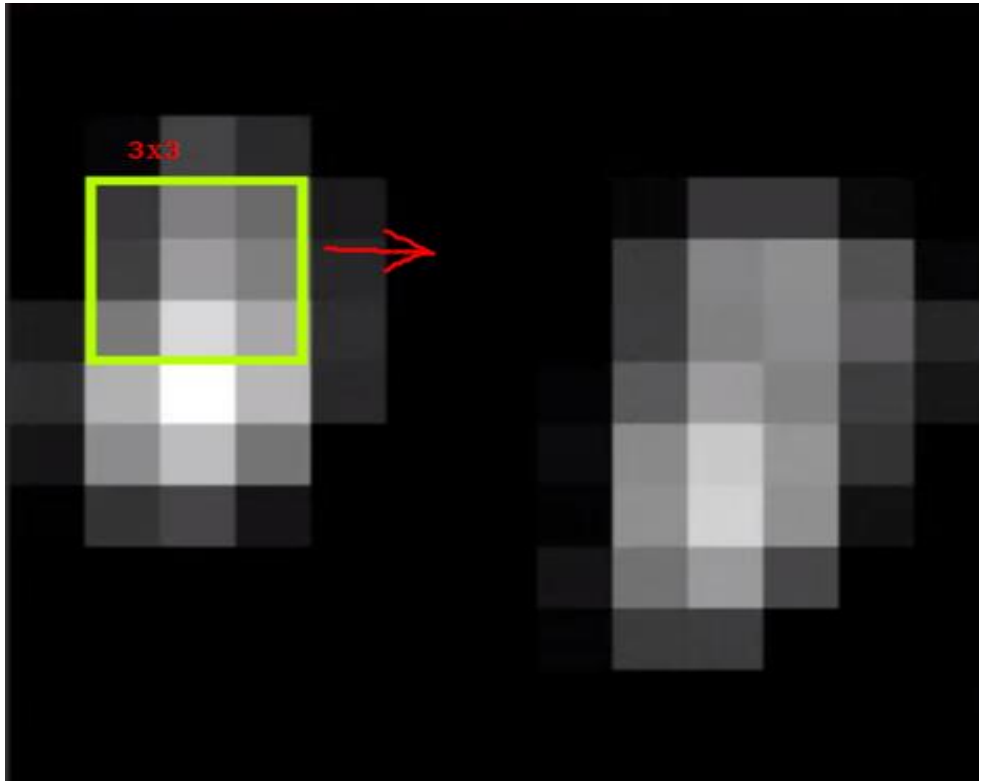
Region Proposal Network (RPN)

- a.找一个已经训练过的卷积网络;
- b.从最后一个卷积层中获取特征空间的映射;
- c.训练一个RPN来检测在图像中有没有对象，并且提出一个方框区域;
- d.把结果送到一个custom layer；
- e.把候选（proposals）送到一个ROI pooling layer（像Fast R-CNN）；
- f.在所有proposals被转换到一个特定固定大小后，将其送到一个全连接层来继续分类。



Region Proposal Network (RPN)

- 工作原理:
- RPN网络在特征空间上滑动一个3*3的窗口
- 这个窗口是用来判断其覆盖的区域下有没有对象，并且给出bounding box的定位。



Faster R-CNN的训练

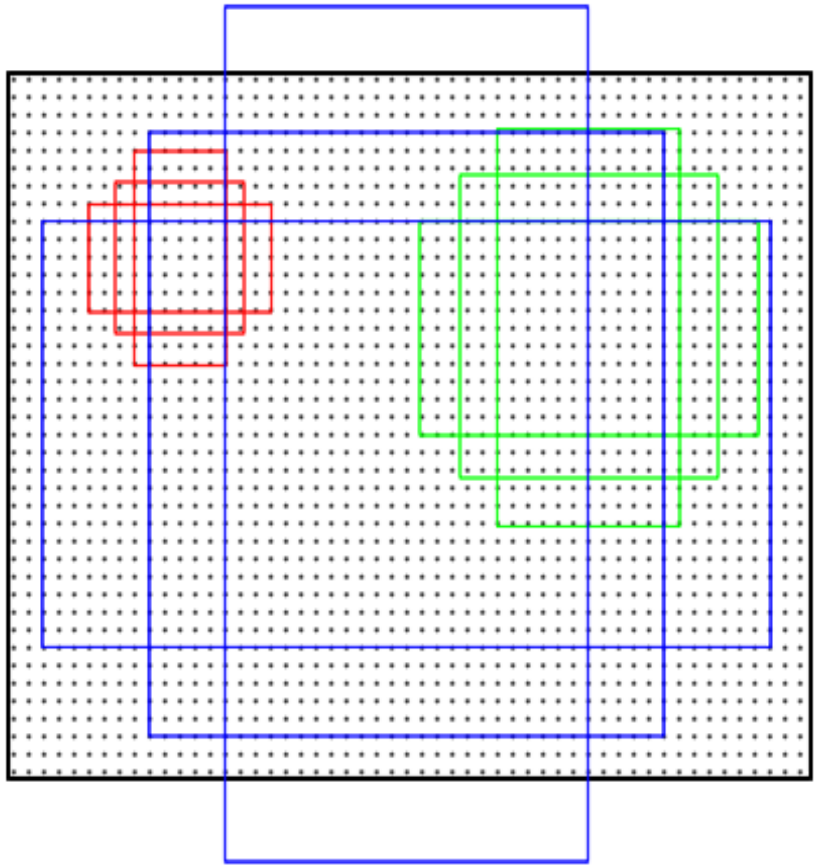
- 1. RPN Classification (Object or not object)
- 2. RPN Bounding box proposal
- 3. Fast R-CNN Classification (Normal object classification)
- 4. Fast R-CNN Bounding-box regression (Improve previous BB proposal)

Faster R-CNN效果

- Faster R-CNN用一个101层的resnet架构，称为ResNet101
- 对每幅图像（包括proposals）的处理速度是R-CNN的250倍，是Fast R-CNN的10倍。
- 精确度和Fast R-CNN一样，都比R-CNN高。

anchor候选区域

- 特征可以看做一个尺度51*39的256通道图像，
- 对于该图像的每一个位置，考虑9个可能的候选窗口：
- 三种面积{1282,2562,5122} × 三种比例{1:1,1:2,2:1}
- 这些候选窗口称为anchors。其大小是feature map上3*3滑动窗口对应的原图的大小，中心点对应关系也是一样。



训练时的样本：

- 对每个标定的真值候选区域，与其重叠比例最大的anchor记为前景样本；
- 对剩余的anchor，如果其与某个标定重叠比例大于0.7，记为前景样本；
- 如果其与任意一个标定的重叠比例都小于0.3，记为背景样本；
- 对再次剩余的anchor，弃去不用；
- 跨越图像边界的anchor弃去不用。

参数精调-fine tuning

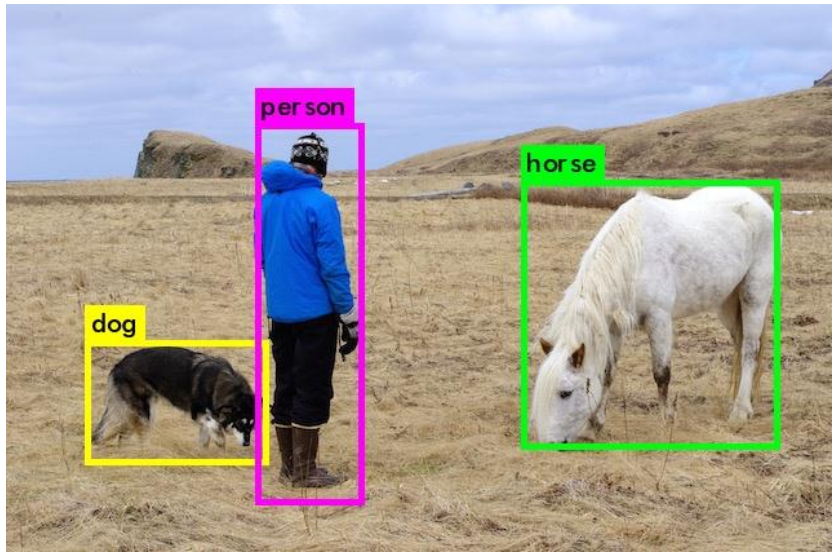
- 区域生成网络（RPN）和fast RCNN都需要一个原始特征提取网络，这个网络使用ImageNet的分类库得到初始参数W0，但要如何精调参数，使其同时满足两方的需求呢？
- 有三种方法。
  - A．轮流训练：从W0开始，训练RPN。用RPN提取训练集上的候选区域；从W0开始，用候选区域训练Fast RCNN，参数记为W1；从W1开始，训练RPN…
  - B．近似联合训练：在backward计算梯度时，把提取的ROI区域当做固定值看待；在backward更新参数时，来自RPN和来自Fast RCNN的增量合并输入原始特征提取层；此方法和前方法效果类似，但能将训练时间减少20%-25%。
  - C．联合训练：但在backward计算梯度时，要考虑ROI区域的变化的影响。

YOLO对象检测的算法

YOLO: You Only Look Once

YOLO算法

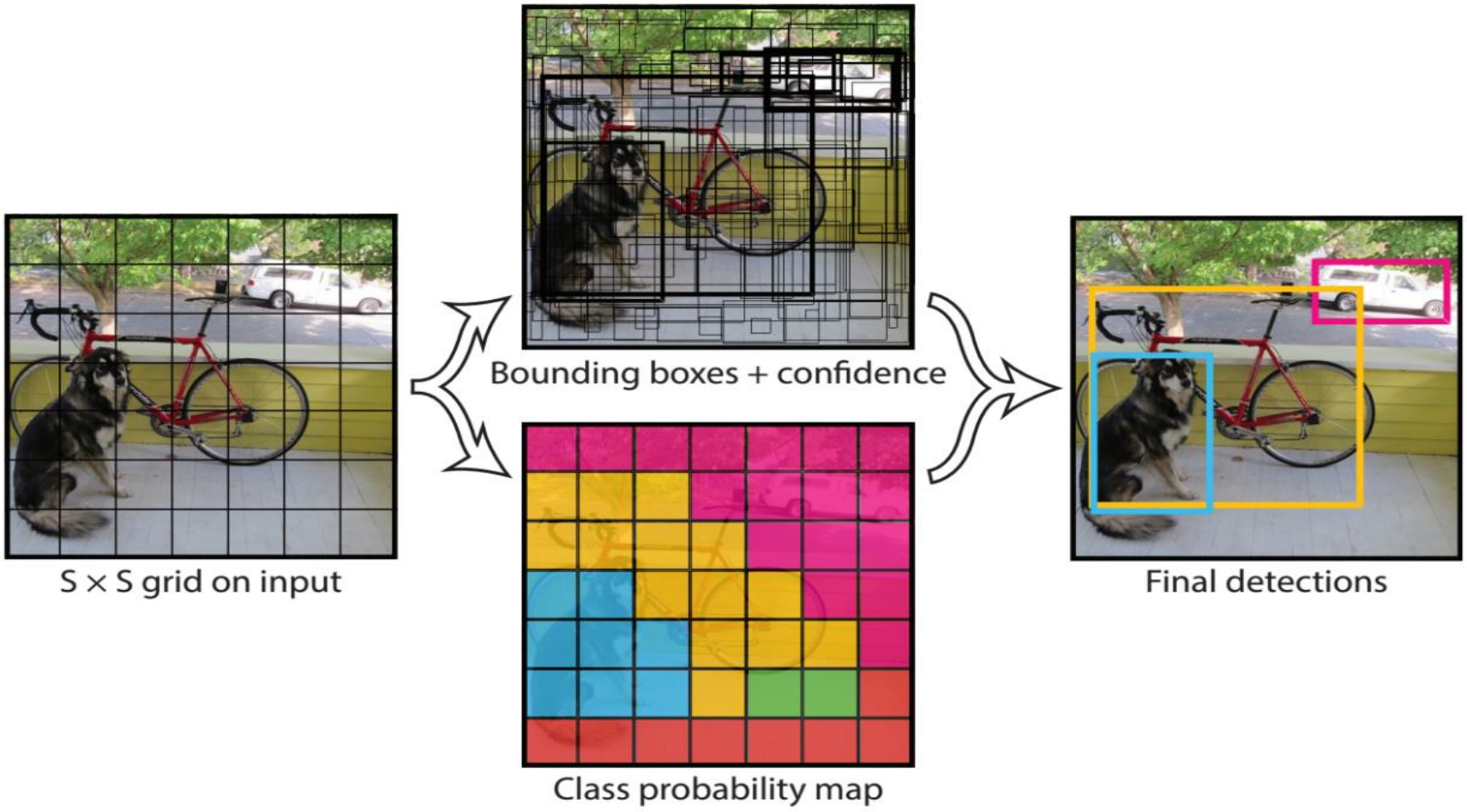
- YOLO算法将对目标检测任务的认识由分类问题（Classification）化简为回归问题（Regression）
- 在保证精度不过多损失的前提下，极大地提高了检测速度。
- 运算速度高，在Titan X GPU上的运行速度可以达到45 FPS（实时）



参考资料

- [3] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]. Computer Vision and Pattern Recognition, 2016:779-788.
- [4] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector[C]. European Conference on Computer Vision, 2016:21-37.

YOLO v1


Bounding boxes + confidence
S × S grid on input
Class probability map
Final detections

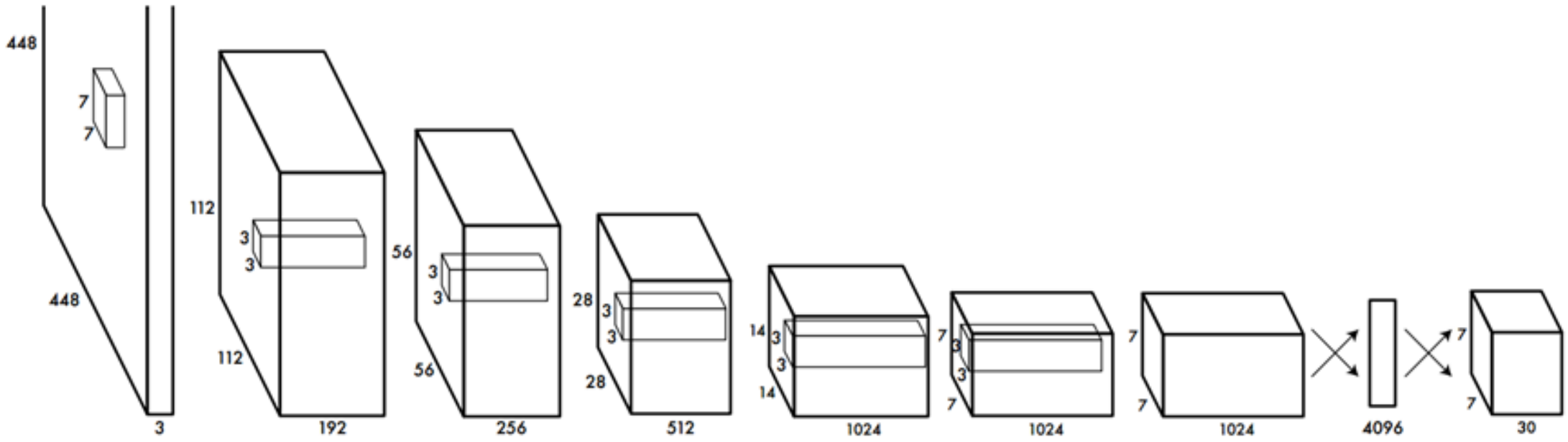
YOLO算法进行目标检测的基本流程

- （1）将整张图像输入到神经网络中，对图像进行预处理，分割为S × S个网格（Grid）：
  - 如果一个目标对象（Object）的中心落在某网格中，则该网格负责后续检测该目标对象的类别（Class）和位置（Location）
- （2）通过预先训练好的神经网络模型，每个网格负责预测B个边界框（Bounding Box），每个边界框对应5个预测参数：
  - 边界框的中心点坐标（x、y），边界框的宽度与高度（w、h）以及置信度（Confidence）。

置信度计算

- 置信度（Confidence）综合反映了当前边界框中存在目标的可能性 $\Pr(\text{Object})$ 以及目标位置预测的准确性 $\text{IOU}_{\text{pred}}^{\text{truth}}$
- $\text{Confidence} = \Pr(\text{Object}) \times \text{IOU}_{\text{pred}}^{\text{truth}}$
  - 如果边界框中不存在目标对象，则 $\Pr(\text{Object}) = 0$，不再进行后续操作，跳至判断下一边界框。
  - 如果存在目标，则在预测上述参数的同时，预测该网格中对象属于某一类别的条件概率 $\Pr(\text{Class}_i|\text{Object})$，取条件概率最大的类别作为我们的预测，如此就可得到每个边界框中的对象类别可能性和位置预测准确性：
- $\Pr(\text{Class}_i|\text{Object}) \times \Pr(\text{Object}) \times \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) \times \text{IOU}_{\text{pred}}^{\text{truth}}$
- 置信度超过设定阈值（Threshold）的所有边界框进行非极大值抑制（NMS）去重处理，即可得到最终的检测结果

YOLO v1网络架构


448
448
7
7
3
112
112
192
3
3
56
56
256
3
3
28
28
512
3
3
14
14
1024
3
3
7
7
1024
7
7
1024
4096
7
7
30
Conv. Layer
7x7x64-s-2
Maxpool Layer
2x2-s-2
Conv. Layer
3x3x192
Maxpool Layer
2x2-s-2
Conv. Layers
1x1x128
3x3x256
1x1x256
3x3x512
Maxpool Layer
2x2-s-2
Conv. Layers
1x1x256
3x3x512
×4
1x1x512
3x3x1024
Maxpool Layer
2x2-s-2
Conv. Layers
1x1x512
3x3x1024
×2
3x3x1024
3x3x1024-s-2
Conv. Layers
3x3x1024
3x3x1024
Conn. Layer
Conn. Layer

YOLO v1 网络结构

- 借鉴了GoogleNet和NIN网络（Network in Network）设计思想
- 网络结构：24个卷积层后接2个全连接层的
- 网络输出： $S \times S \times (B \times 5 + \#\text{Classes})$ 的三维张量（Tensor）

- 存在的问题漏检：
  - 对于一些长宽比较为特殊的对象，图像中出现成群相邻的小目标对象时，容易发生漏检的问题
  - 检测结果准确性方面，不如Faster R-CNN等

YOLO-损失函数

- 损失函数需要考虑如下的目标:
  - 1. 分类（20类）;
  - 2. 有无对象;
  - 3. Bounding box 的参数（x, y, width, height）的回归。
- 每一个子目标都有一个误差平方和（sum square error）和一个因子，用调节平衡边界框的位置大小参数和分类目标所占比重的。
- 一些注意事项:
  - 1.若一个grid cell没有对象，那么不会反向传播（BP）分类损失;
  - 2.只有与标准答案（ ground truth box ）的IOU（Intersect over union）最高的边界框（ bouding box ）的误差才会被反向传播（ BP ）用于训练。

YOLO损失函数

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

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

坐标预测

判断第i个网格中的第j个box是否负责这个object

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

含object的box的confidence预测

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

不含object的box的confidence预测

判断是否有object中心落在网格i中

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

类别预测

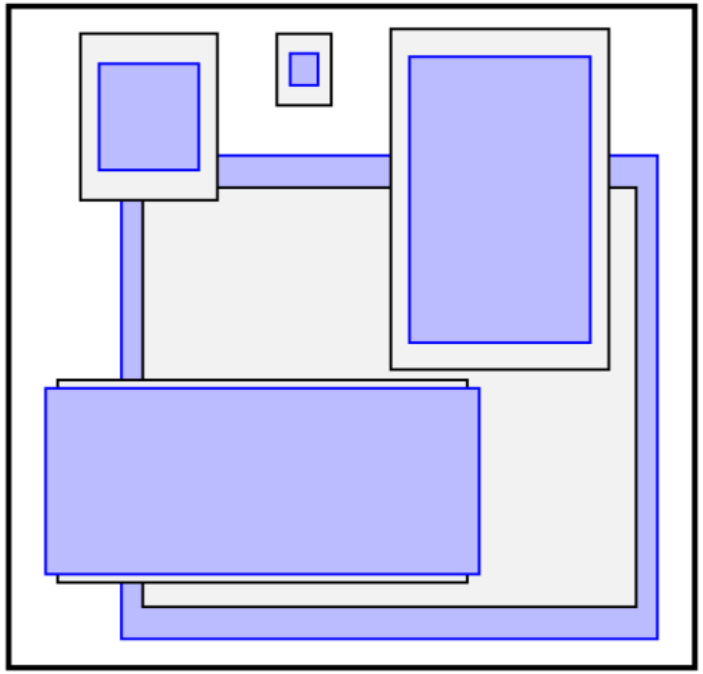
YOLO v2改进YOLO v1

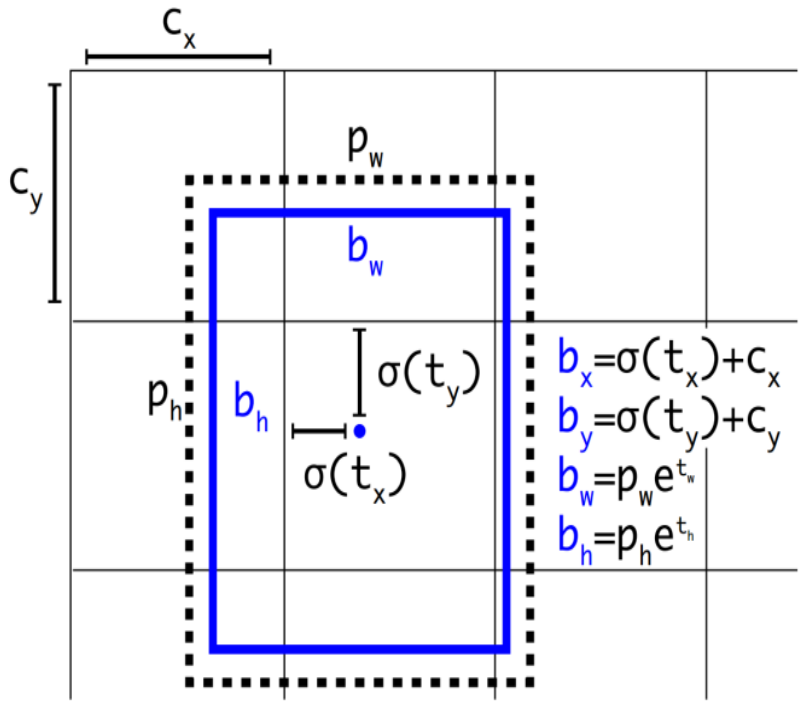
- YOLOv2 提升了检测结果的准确性，同时增快了运算速度（由原先的45 FPS提升至67 FPS）
- YOLOv2是当前最佳的实时高精度目标检测算法，在计算速度和检测准确度之间达到了一个较好的平衡
- 基本思想：采取了Anchor机制来处理不同长宽比例对象的检测问题。
- Anchor机制即神经网络通过学习训练，预先设定不同边界框的长宽比的可能值，从而使得边界框能够更容易探测到长宽比相近的对象。
- 使用了K均值聚类（K-means Cluster）的方法，得到了神经网络开始训练前的最佳起始Anchor长宽比和数量，提升了Anchor机制的最终效果。

YOLO v2

- 直接位置预测，优化神经网络训练时的收敛速度
- 批次规范化（Batch Normalization）防止过拟合现象（Over-fitting）的发生
- 在网络中加入转移层（Pass-through Layer），连接不同分辨率下的特征图谱（Feature Map）
- 增强网络检测小尺寸对象能力的同时，又不至于增加过多的计算量

Anchor机制与直接位置预测



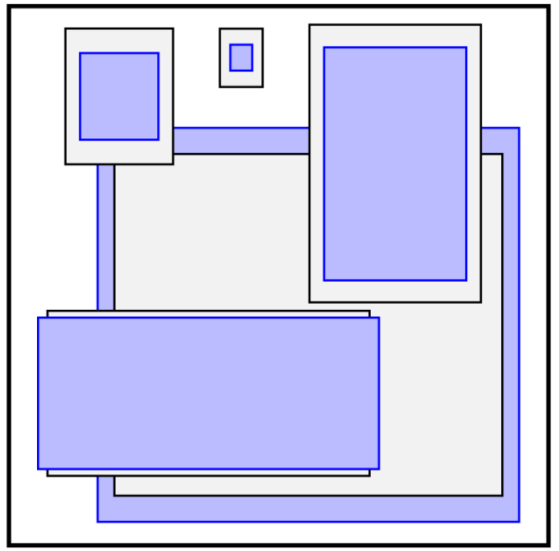


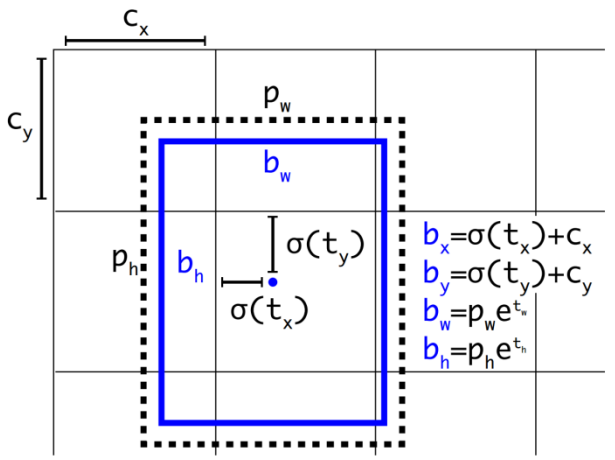
Anchor机制，称为参照机制。

YOLOv2算法

- YOLO的不足 —— 漏检
  - 成群相邻的小目标对象
  - 长宽比较为特殊的对象实例

- YOLOv2
- 改进办法：
  - Anchor机制
  - 不同分辨率下的特征图谱
  - 直接位置预测
  - ……





YOLO v3改进YOLO v2

- YOLO v3: An Incremental Improvement
- 多尺度预测 （类FPN）
- 基础分类网络（类ResNet）和分类器
- 不使用Softmax对每个框进行分类

SSD对象检测的算法

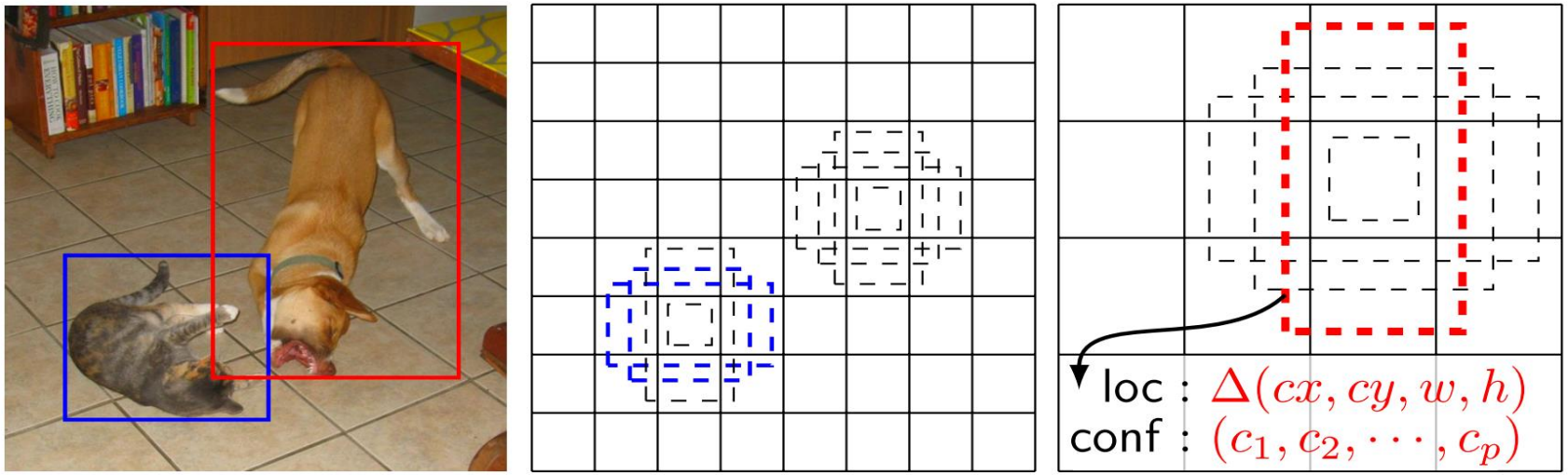
SSD: Single Shot MultiBox Detector

SSD

- SSD方法的核心：
  - 预测对象（ predict object ）及其归属类别的score（得分）
  - 在 feature map上使用小的卷积核去predict一系列bounding boxes的box offsets
- 为了得到高精度的检测结果：
  - 在不同层次的 feature maps（特征图谱）上去 predict object、box offsets,
  - 得到不同aspect ratio（纵横比）的predictions。
- 改进设计：
  - 能够在当输入分辨率较低的图像时，保证检测的精度。
  - 整体端到端（end-to-end）的设计，训练也变得简单。
  - 在检测速度、检测精度之间取得较好的折衷。
- SSD，比YOLOv1方法，还要快，还要精确。
- SSD，在保证速度的同时，mAP指标与使用region proposals 技术的方法（如 Faster R-CNN）相媲美

SSD术语

- 特征图格子（Feature Map Cell） 就是将 特征图（Feature Map）切分成 8×8 或者 4×4 之后的一个个格子
- 缺省框（Default Box） 就是每一个格子上，一系列固定大小的 box，即图中虚线所形成的一系列 boxes



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

SSD模型

- SSD是基于一个前向传播CNN网络，产生一系列固定大小（fixed-size）的框（bounding boxes），以及每一个 框 中包含对象实例的可能性，即分数（score）。

- 非极大值抑制（Non-maximum suppression）操作后，得到最终的预测（predictions）。

Multi-scale feature maps for detection

- 在最开始，是一个被称为base network的用于图像分类的标准架构。
- 在其之后，添加了一些auxiliary structure(辅助结构)来产生有下列特征的检测：
- 在基础网络结构后，添加了额外的卷积层，这些卷积层的大小是逐层递减的，可以在多尺度下进行 predictions。

Convolutional predictors for detection

- 每一个添加的特征层（或者在基础网络结构中的特征层），可以使用一系列 convolutional filters，去产生一系列固定大小的 predictions。
- 对于一个大小为 m×n，具有p通道的特征层，使用的 convolutional filters就是3×3×p的kernels。
- 产生的predictions，是归属类别的一个得分，相对于 default box coordinate的shape offsets。
- 在每一个 m×n 的特征图位置上，使用上面的 3×3 的 kernel，会产生一个输出值。bounding box offset值是输出的default box与此时feature map location之间的相对距离

Default boxes and aspect ratios

- 每一个box相对于与其对应的 feature map cell 的位置是固定的。在每一个 feature map cell 中，要 predict 得到的box与default box之间的offsets，以及每一个box中包含对象的score（每一个类别概率都要计算出）。
- 因此，对于一个位置上的k个boxes中的每一个box，我们需要计算出c个类，每一个类的score，还有这个box相对于它的默认box的 4 个偏移值（offsets）。
- 于是，在feature map中的每一个feature map cell上，就需要有(c+4)×k个filters。对于一张m×n大小的feature map，即会产生 (c+4)×k×m×n 个输出结果

SSD训练（training）

- 在训练时， 与基于 region proposals + pooling 方法的区别是，SSD训练图像中的标准答案（ground truth）需要赋予到那些固定输出的boxes上。
- SSD输出的是事先定义好的，一系列固定大小的边界框（bounding boxes） 。
- 当这种将训练图像中的标准答案（ ground truth ）与固定输出的boxes对应之后，就可以端对端（ end-to-end ）地进行损失函数（ loss function ）的计算以及反向传播（back-propagation）计算更新。

1. 将ground_truth box与 default box 配对

- A. 将ground_truth box与 default box 配对，组成label的方法：
  - 在开始的时候，用 MultiBox 中的 best jaccard overlap 来匹配每一个 ground truth box 与 default box，这样就能保证每一个 groundtruth box 与唯一的一个 default box 对应起来。
  - 但是不同于 MultiBox ，将 default box 与任何的 groundtruth box 配对，只要两者之间的jaccard overlap 大于一个阈值，这里阈值为 0.5。

2. SSD的训练目标

- 用 $x^p_{ij}=1$ 表示 第 *i* 个 default box 与 类别 p 的 第 *j* 个 ground truth box 相匹配，否则若不匹配的话，则 $x^p_{ij}=0$。根据上面的匹配策略，一定有 $\sum_i xpij \geqslant 1$，意味着对于第 j 个 ground truth box，有可能有多个 default box与其相匹配
- 总的目标损失函数（objective loss function）就由 localization loss（loc） 与 confidence loss（conf） 的加权求和：
- 其中，N 是与 ground truth box 相匹配的 default boxes 个数。

$$L(x,c,l,g)=\frac{1}{N}(L_{conf}(x,c)+\alpha L_{loc}(x,l,g))$$

A. Localization loss（loc）

- a.　Localization loss（loc）是 Fast R-CNN 中 Smooth L1 Loss，用在 predict box（l）与 ground truth box（g）参数（即中心坐标位置，width、height）中，回归 bounding boxes 的中心位置，以及 width、height。

$$L_{loc}(x,l,g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx,cy,w,h\}} x_{ij}^{k} \text{smooth}_{\text{L1}}(l_i^m - \hat{g}_j^m)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h$$

$$\hat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right) \qquad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right)$$

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

B. Classification loss (confidence loss)

- b. Classification loss (confidence loss)(conf) 是 Softmax Loss，输入为每一类的置信度 c。

$$L_{conf}(x,c) = -\sum_{i\in Pos}^{N} x_{ij}^{p} log(\hat{c}_i^p) - \sum_{i\in Neg} log(\hat{c}_i^0) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$

C. Choosing scales and aspect ratios for default boxes:

- 大部分 CNN 网络在越深的层，feature map 的尺寸（size）会越来越小。
- 这样做不仅仅是为了减少计算与内存的需求，还有个好处就是，最后提取的 feature map 就会有某种程度上的平移与尺度不变性。
- 为了处理不同尺度的对象，有些方法将图像转换成不同的尺度，将这些图像独立的通过 CNN 网络处理，再将这些不同尺度的图像结果进行综合。
- 如果使用同一个网络中的、不同层上的 feature maps，也可以达到相同的效果，同时在所有对象尺度中共享参数。
- 一般来说，一个 CNN 网络中不同的 layers 有着不同尺寸的 感受域（receptive fields）。
  - 这里的感受域，指的是输出的 feature map 上的一个节点，其对应输入图像上尺寸的大小。

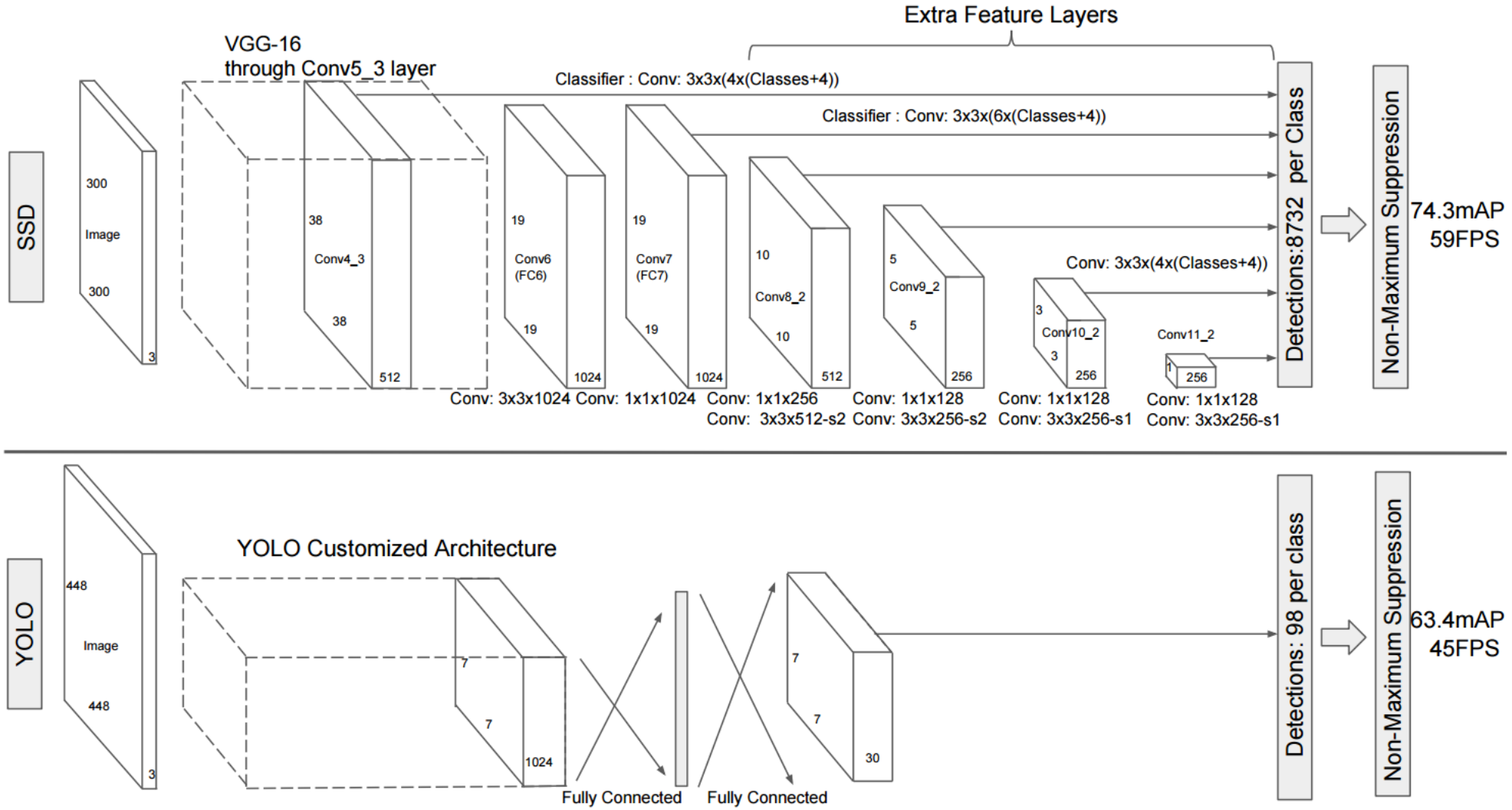
D. Hard negative mining

- 在生成一系列的 predictions 之后，会产生很多个符合 ground truth box 的 predictions boxes，但同时，不符合 ground truth boxes 也很多，而且这个 negative boxes，远多于 positive boxes。这会造成 negative boxes、positive boxes 之间的不均衡。训练时难以收敛。
- 因此先将每一个对象位置上对应 predictions（default boxes）是 negative 的 boxes 进行排序，按照 default boxes 的 confidence 的大小，选择最高的几个，保证最后 negatives、positives 的比例在 3:1。
- 通过实验发现，这样的比例可以更快的优化，训练也更稳定。

E. Data augmentation

- 为了使模型对不同大小形状的输入对象更加鲁棒，每张训练图像被随机采样（sampled）,每一张训练图像，随机的进行如下几种选择:
- 使用原始的图像
- 采样一个 patch，与对象之间最小的 jaccard overlap 为：0.1，0.3，0.5，0.7 与 0.9
- 随机地采样一个 patch。
- 采用的patch占原始图片大小的[0.1, 1],纵横比在[1/2, 2] 中，在ground truth box的中心在采样patch中时，保留重叠部分，在这些采样步骤之后，每一个采样的 patch 被 resize 到固定的大小，并且以0.5的概率随机地水平翻转（horizontally flipped）。

YOLO与SSD比较


SSD
300
Image
300
3
VGG-16
through Conv5_3 layer
38
Conv4_3
38
512
19
Conv6
(FC6)
19
1024
19
Conv7
(FC7)
19
1024
10
Conv8_2
10
512
5
Conv9_2
5
256
3
Conv10_2
3
256
Conv11_2
1
256
Extra Feature Layers
Classifier : Conv: 3x3x(4x(Classes+4))
Classifier : Conv: 3x3x(6x(Classes+4))
Conv: 3x3x(4x(Classes+4))
Conv: 3x3x1024
Conv: 1x1x1024
Conv: 1x1x256
Conv: 3x3x512-s2
Conv: 1x1x128
Conv: 3x3x256-s2
Conv: 1x1x128
Conv: 3x3x256-s1
Conv: 1x1x128
Conv: 3x3x256-s1
Detections:8732 per Class
Non-Maximum Suppression
74.3mAP
59FPS
YOLO
448
Image
448
3
YOLO Customized Architecture
7
7
1024
Fully Connected
Fully Connected
7
7
30
Detections: 98 per class
Non-Maximum Suppression
63.4mAP
45FPS

图像语义分割方法

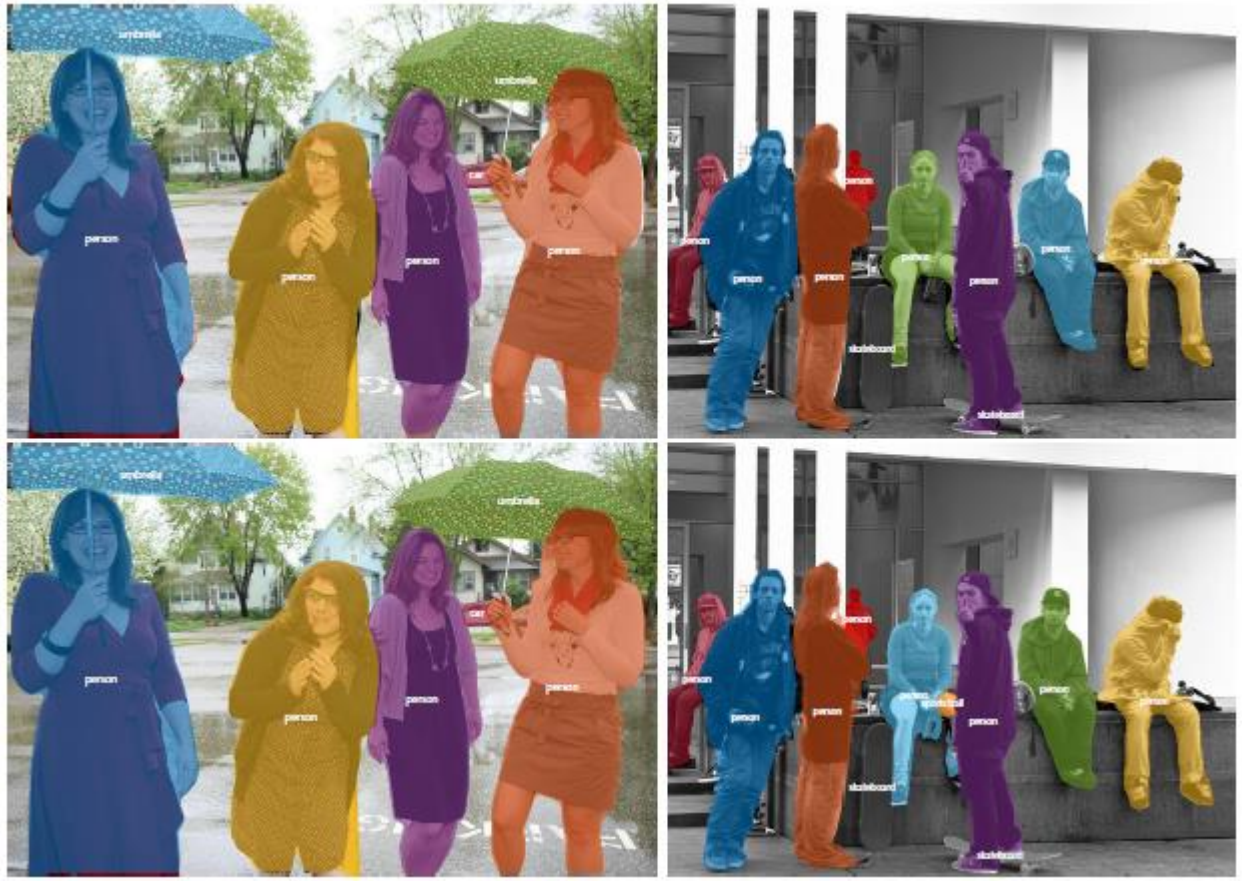
Semantic Segmentation

参考资料

- Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, Mask R-CNN, ICCV 2017.
- Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, Ross Girshick, Learning to Segment Every Thing, CVPR 2018.
- Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, Piotr Dollár, Panoptic Segmentation, CVPR 2019.
- Xinlei Chen, Ross Girshick, Kaiming He, Piotr Dollár, TensorMask: A Foundation for Dense Object Segmentation, 2019.

Semantic Segmentation

- TensorMask and Mask R-CNN



参考资料

- [4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016.
- [5] Huang et al., Speed/accuracy trade-offs for modern convolutional object detectors[C], CVPR 2017. (https://arxiv.org/abs/1611.10012)

谢谢指正！

zhenchen@tsinghua.edu.cn