人工智能原理与技术

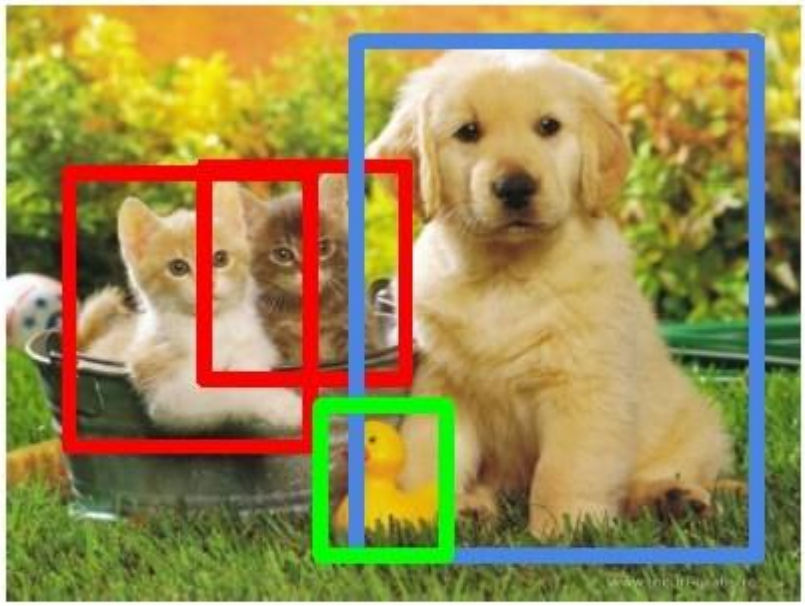
基于深度学习的目标检测

主要内容

- 背景介绍:概念、数据集和竞赛
- 战前准备：评价准则、滑动窗口、目标候选生成、难样本挖掘、非极大值抑制、检测框回归
- 两阶段方法：RCNN, SPPNet, Fast RCNN, Faster-RCNN,FPN,RFCN
- 单阶段方法：YOLO, SSD, RetinaNet
- 荟萃：目标检测方法对比
- 拓展：视频中目标检测
- 总结

背景介绍

什么是目标检测?



CAT, DOG, DUCK

对于一张图片，

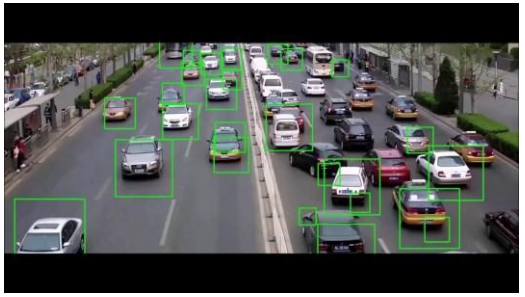
找到其中所有的物体，

并给出其类别和边界框

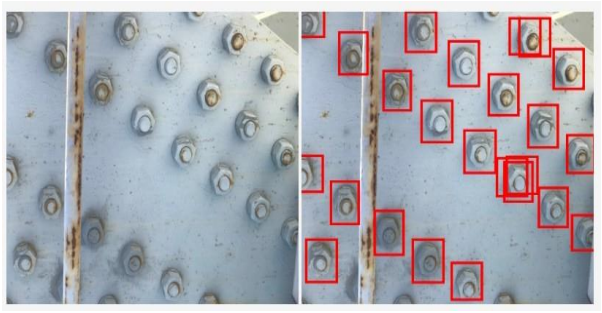
目标检测：应用

人脸检测

行人检测



车辆检测



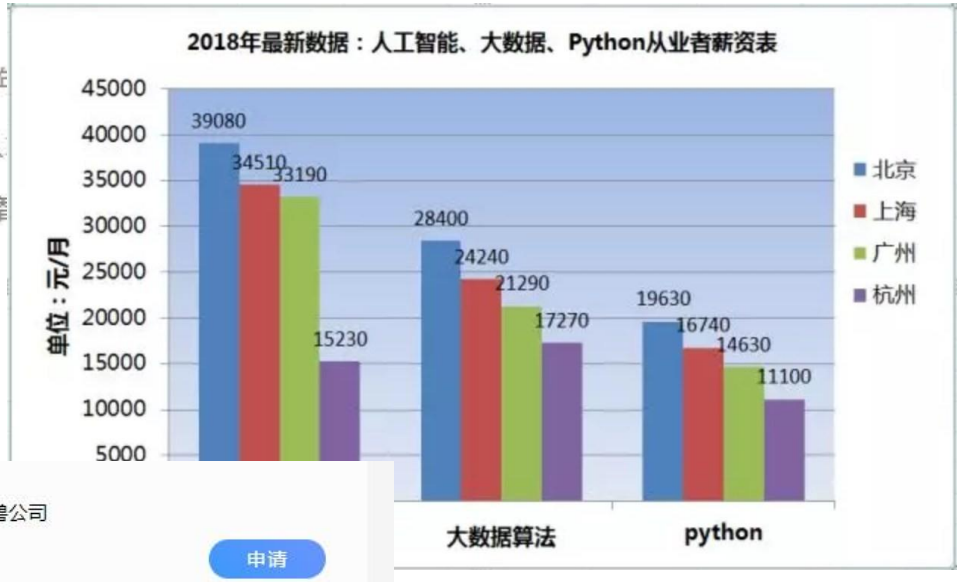
工业检测

网络图片敏感信息检测

公共安全

目标检测：人才巨大缺口&高薪资

据领英近日发布的《全球AI领域人才报告》显示，截至2017年
AI（人工智能）领域技术人才数量超过190万，其中美国相关人
国的相关人才总数为5万人，仅为美国的1/17，同时，国内人工智
比例仅为1:10，供需严重失衡。



计算机视觉算法开发工程师(某知名计算机...
25000以上 元/月 保障 直投
带薪年假
某知名计算机视觉独角兽公司
北京 | 本科 | 1-3年
12-10发布于猎聘网
申请

图像算法/深度学习
50000以上 元/月
国内某领先的计算机视觉感知交互...
北京 | 硕士 | 3-5年
10-29发布于智联卓聘
申请

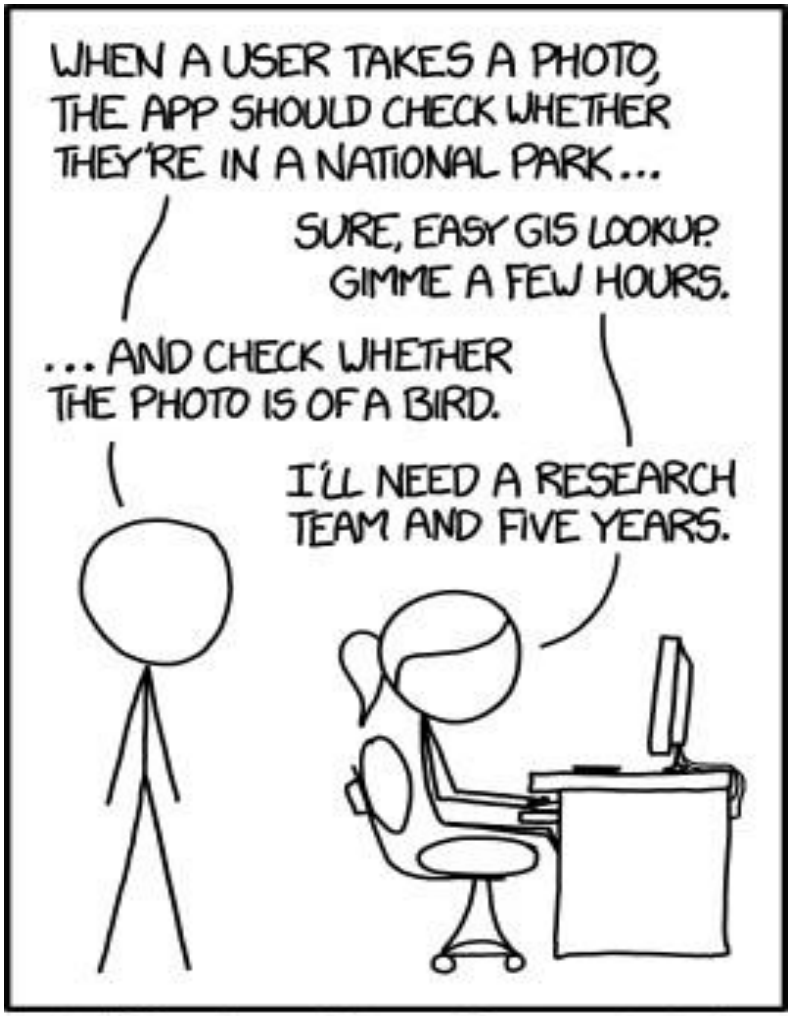
计算机视觉研究员
50000以上 元/月
国内某计算机视觉领域科技企业
北京 | 博士 | 1-3年
01-10发布于智联卓聘
申请

计算机视觉：先定一个小目标

Founder, MIT AI project

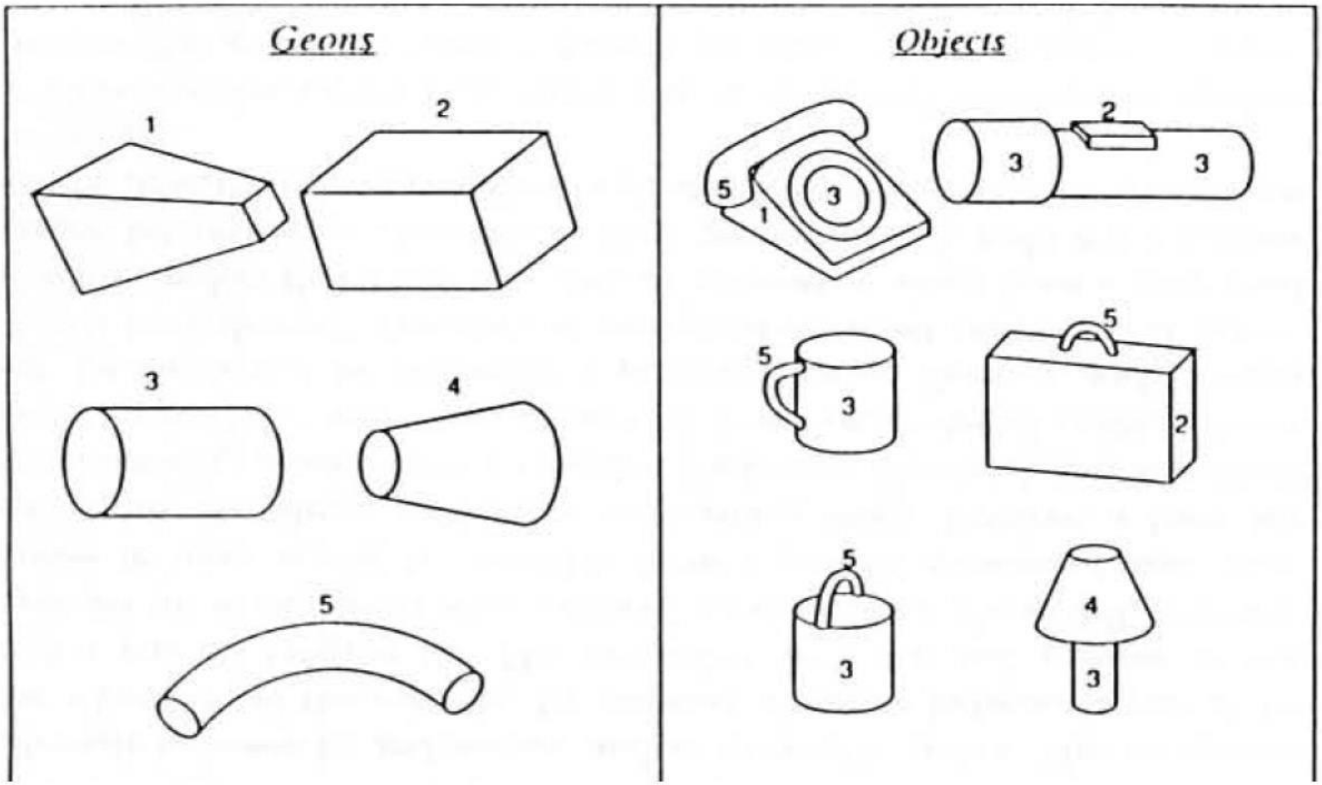
In 1966, Marvin Minsky at MIT asked his undergraduate student Gerald Jay Sussman to “spend the summer linking a camera to a computer and getting the computer to describe what it saw”. We now know that the problem is slightly more difficult than that. (Szeliski 2009, Computer Vision)

计算机视觉：并不容易！


WHEN A USER TAKES A PHOTO, THE APP SHOULD CHECK WHETHER THEY'RE IN A NATIONAL PARK...
SURE, EASY GIS LOOKUP. GIMME A FEW HOURS.
...AND CHECK WHETHER THE PHOTO IS OF A BIRD.
I'LL NEED A RESEARCH TEAM AND FIVE YEARS.
IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

目标检测的早期探索

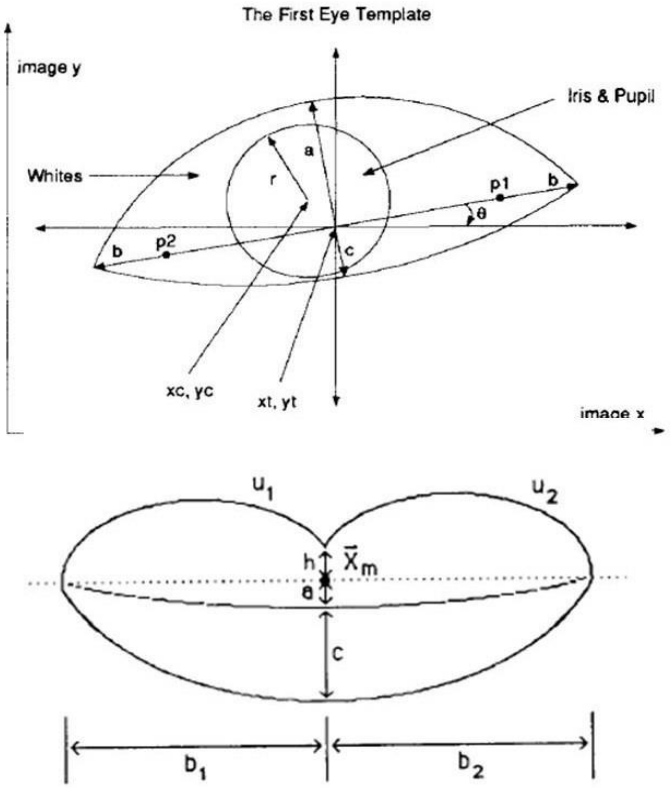
物理模型

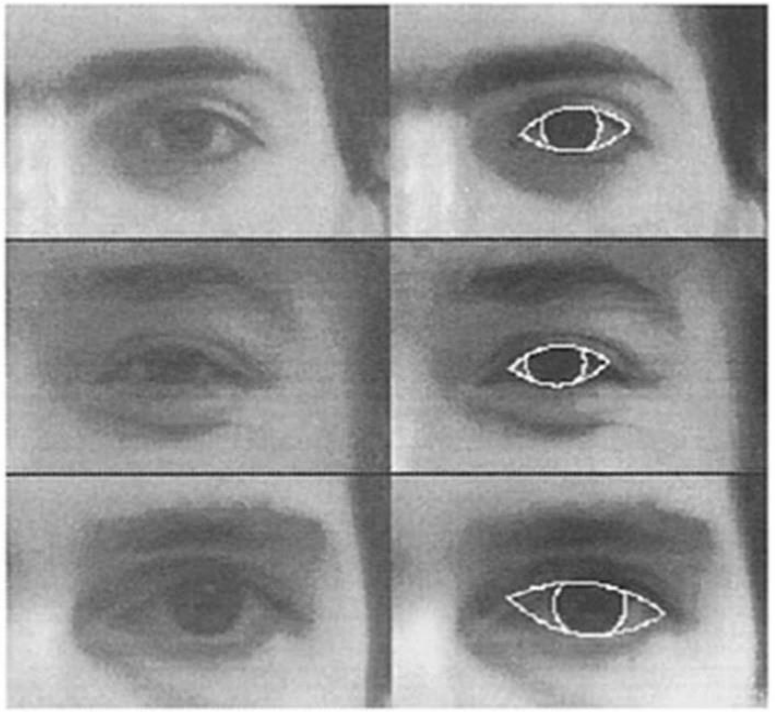


Irving Biederman, Recognition-by-Components: A Theory of Human Image Understanding. Psychological Review, 1987. Google citations: 5309

目标检测的早期探索

物理模型



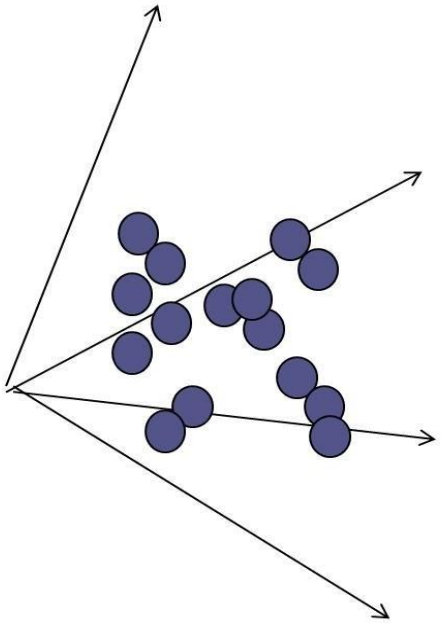


Alan L. Yuille, David S. Cohen, Peter W. Hallinan. Feature extraction from faces using deformable templates, CVPR, 1989. Google citations: 2303

目标检测的早期探索

表观统计模型

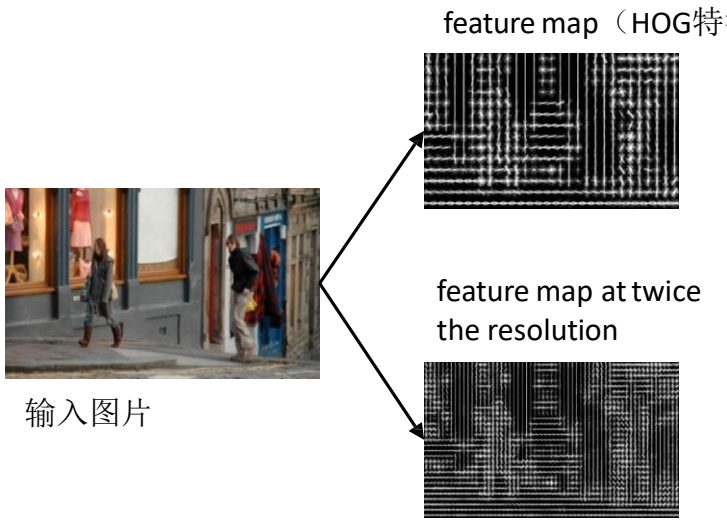
- Objects as appearance patches
  - E.g., a list of pixel intensities
- Learning patterns directly from image features



Eigenfaces (Turk & Pentland, CVPR, 1991) Google citations: 5573

目标检测的早期探索

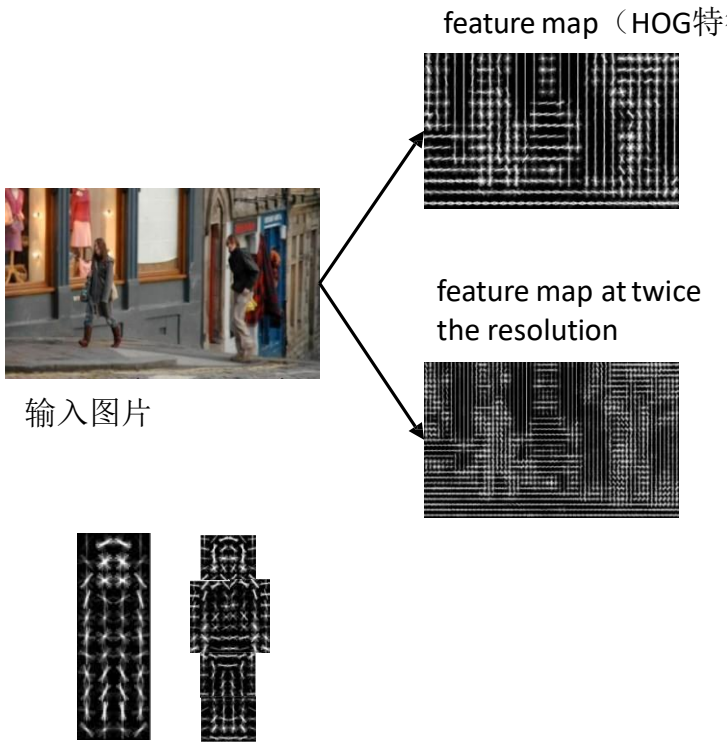
基于部件的模型（DPM）



Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, Deva Ramanan, “Object Detection with Discriminatively Trained Part-Based Models,” TPAMI 2010. **Google Citation: 7645**

目标检测的早期探索

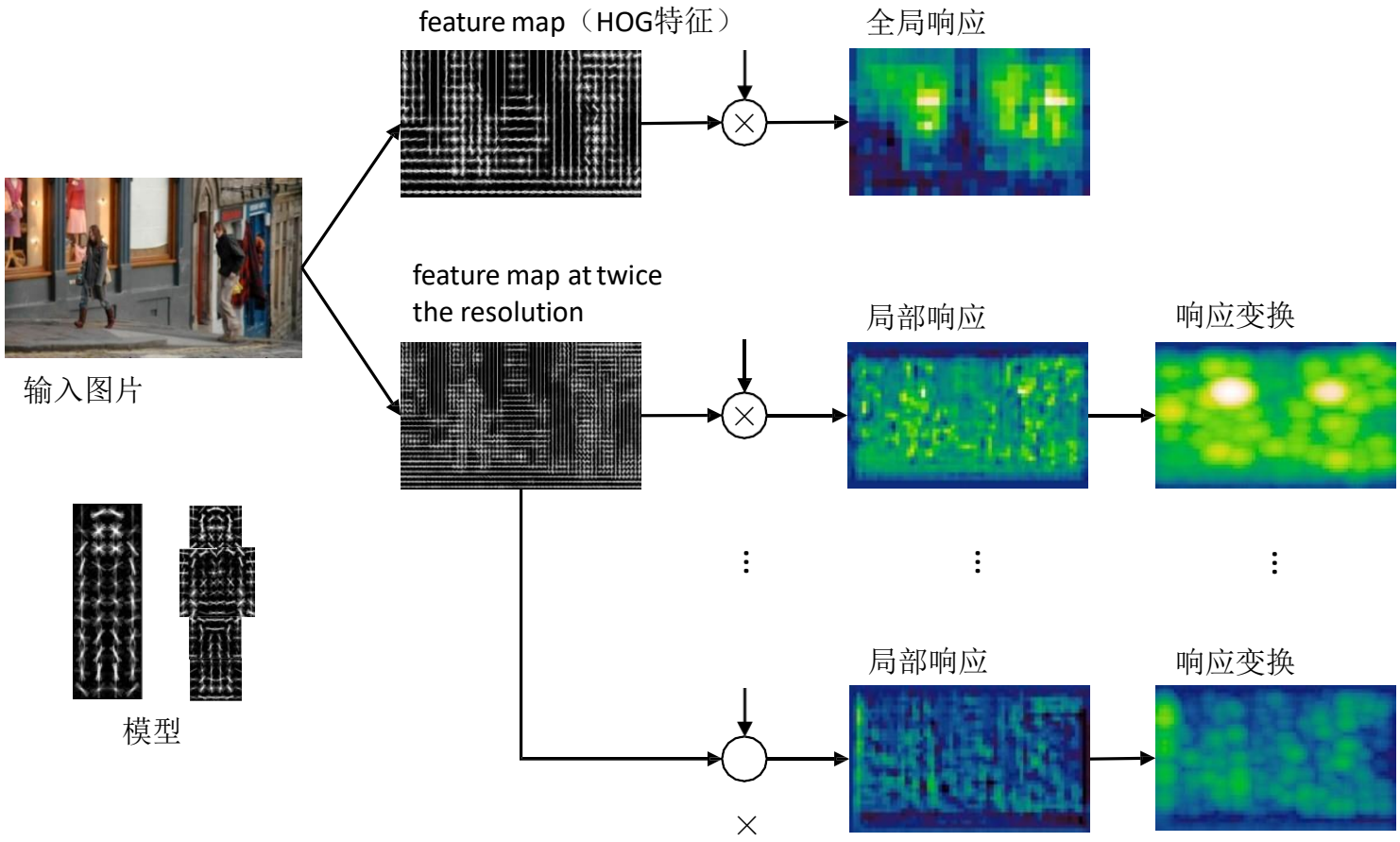
基于部件的模型（DPM）



Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, Deva Ramanan, “Object Detection with Discriminatively Trained Part-Based Models,” TPAMI 2010. **Google Citation: 7645**

目标检测的早期探索

基于部件的模型（DPM）

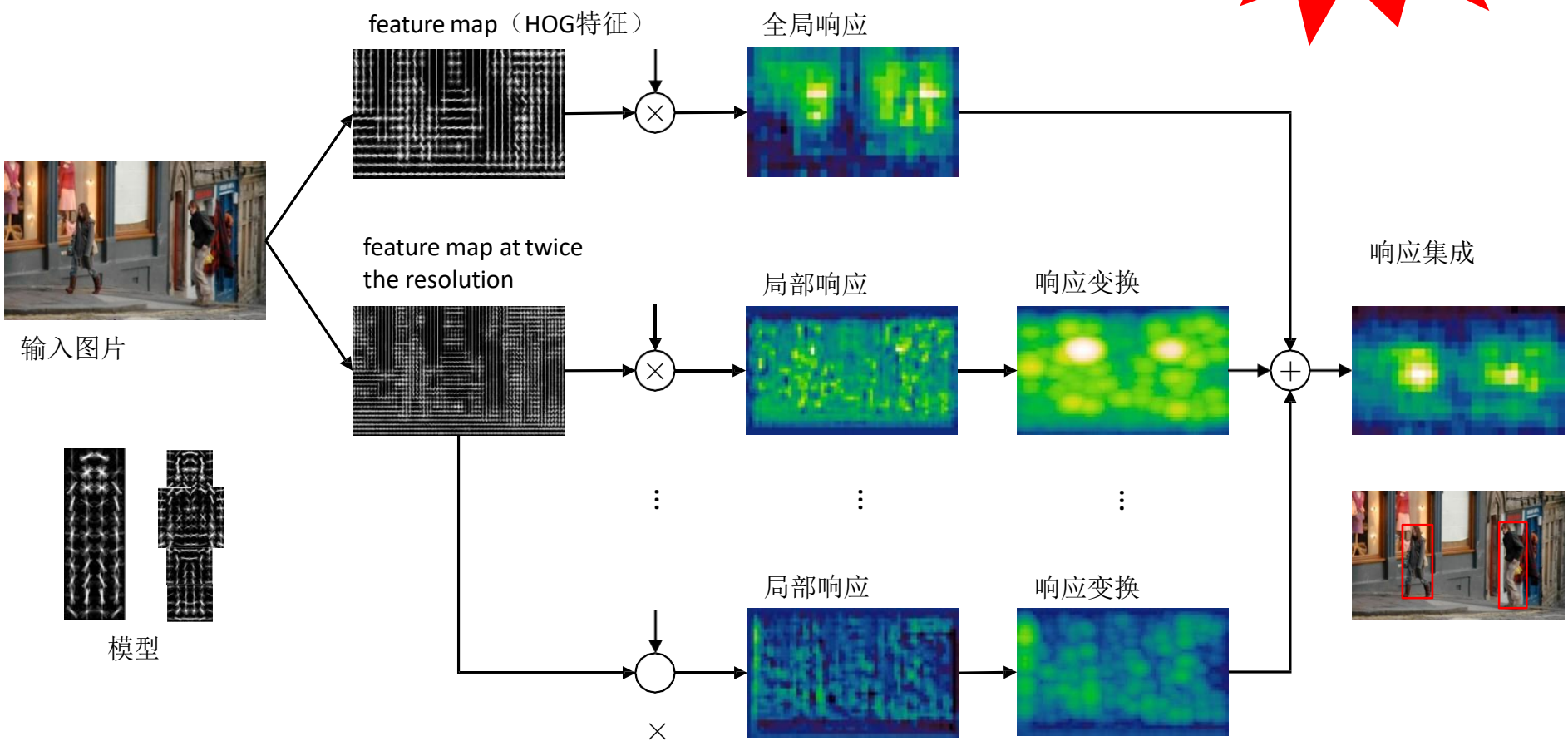


Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, Deva Ramanan, “Object Detection with Discriminatively Trained Part-Based Models,” TPAMI 2010. **Google Citation: 7645**

目标检测的早期探索

基于部件的模型（DPM）

精度有待提高！



Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, Deva Ramanan, “Object Detection with Discriminatively Trained Part-Based Models,” TPAMI 2010. **Google Citation: 7645**

目标检测近期工作：数据驱动

数据驱动

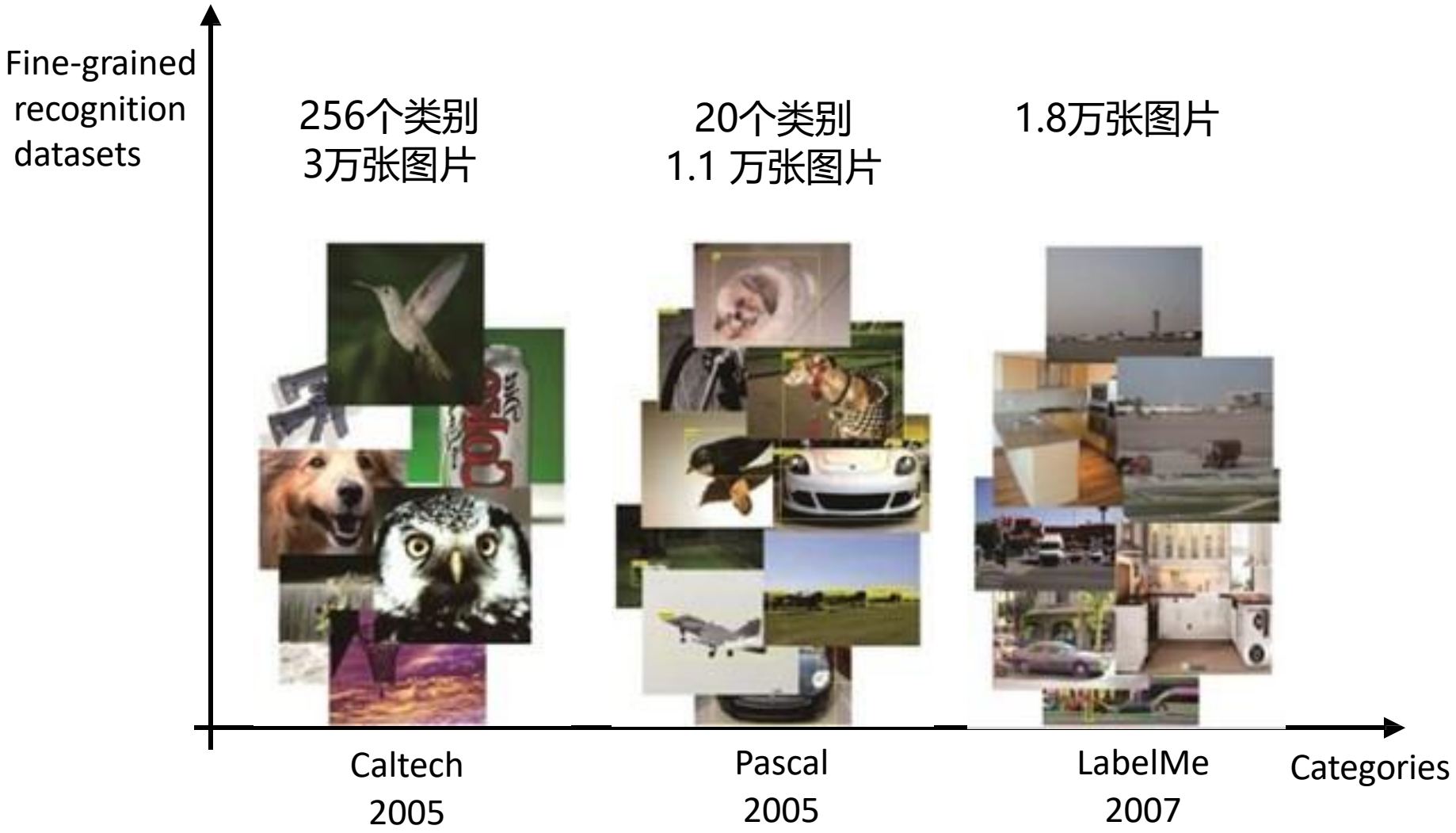
互联网（Internet）

物联网(IoT)

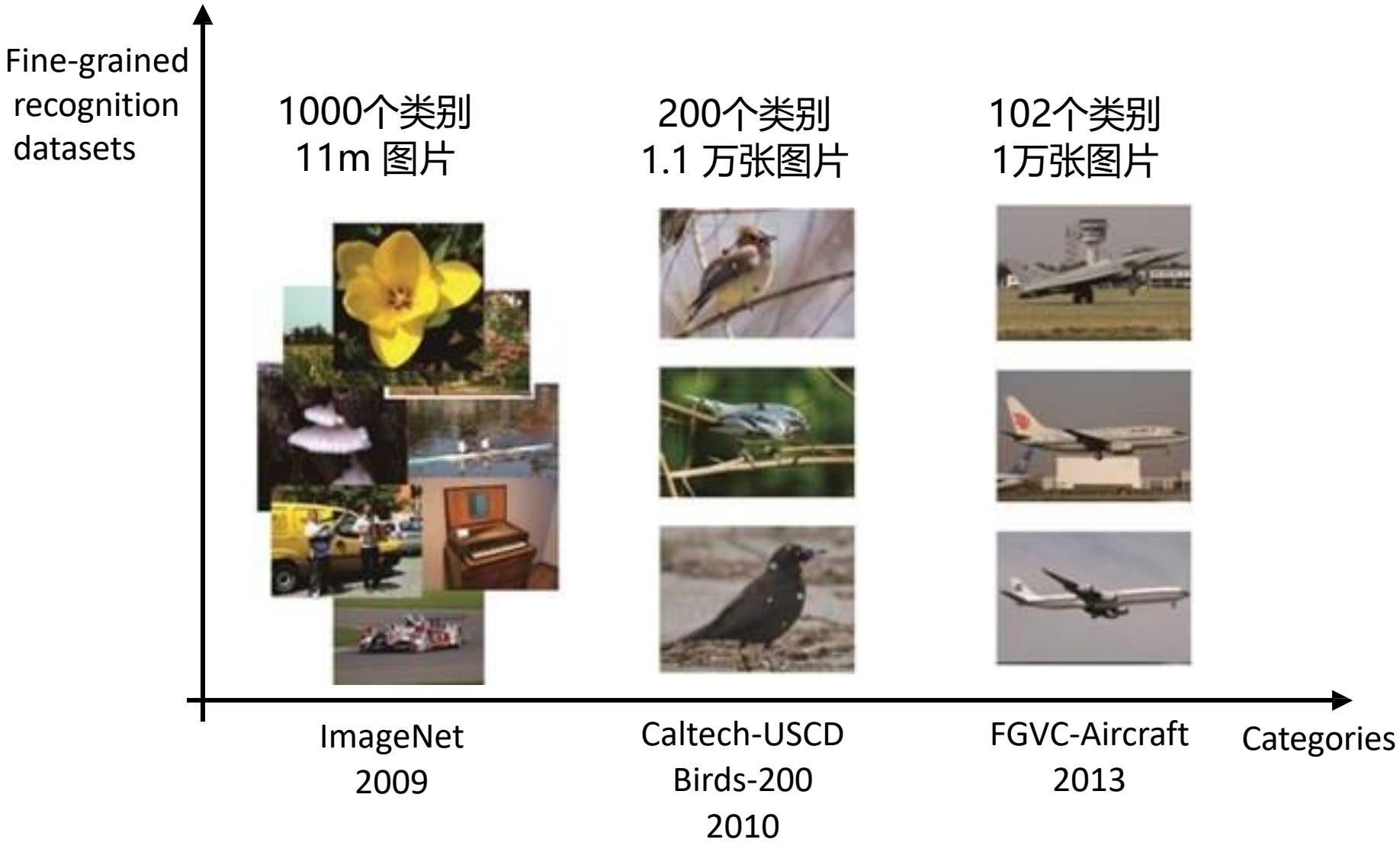
一台1080p的高清摄像头，一天24小时所产生数据量是64GB（Giga Byte），每年产生的数据量是23TB。以北京市50万台摄像头保存一年，数据量可达到11.5EB。

早期方法所能覆盖的数据只是视觉大数据中的一个很小子集！

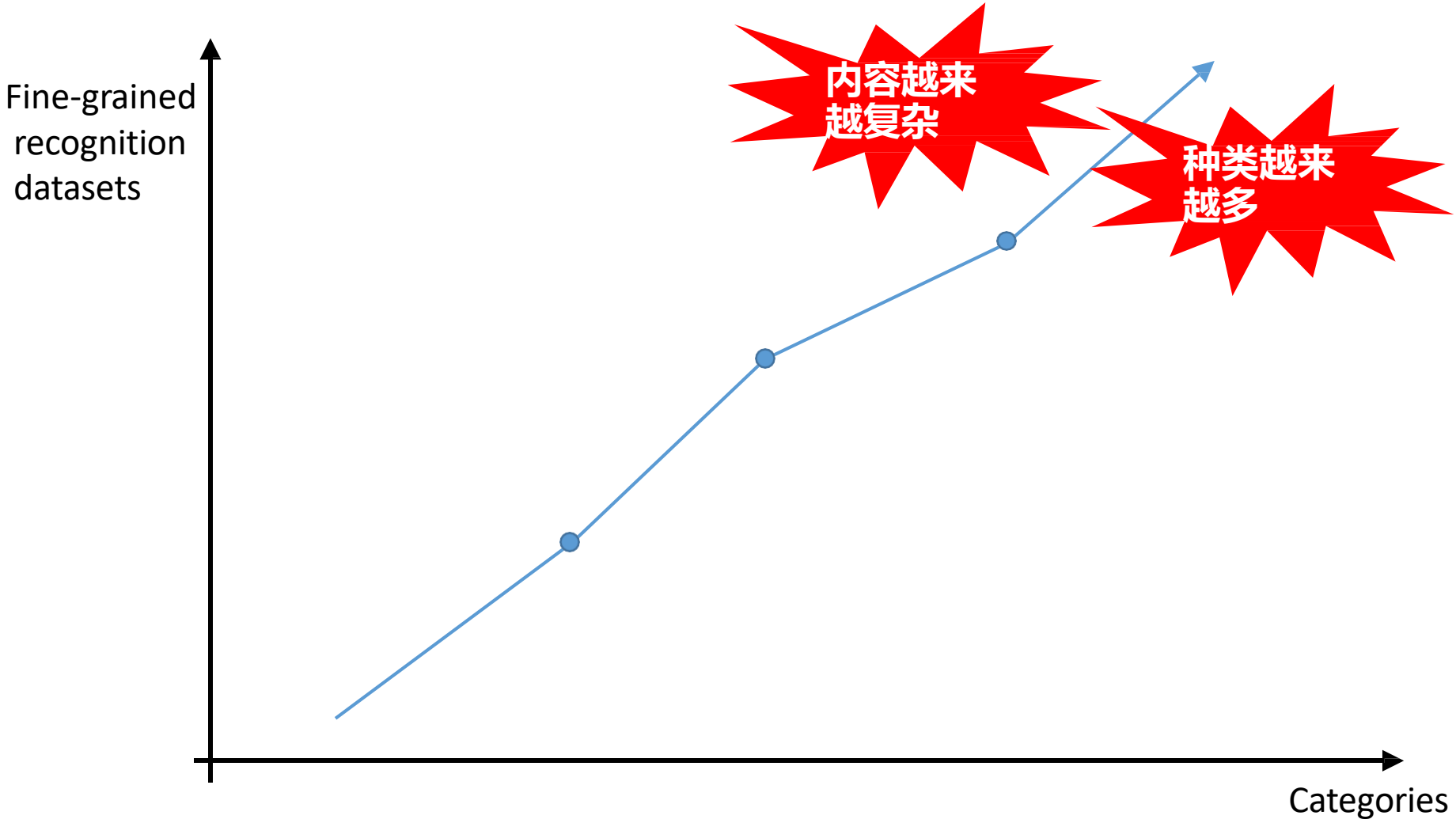
目标检测/识别：数据集


Fine-grained recognition datasets
256个类别
3万张图片
20个类别
1.1 万张图片
1.8万张图片
Caltech
2005
Pascal
2005
LabelMe
2007
Categories

目标检测/识别：数据集

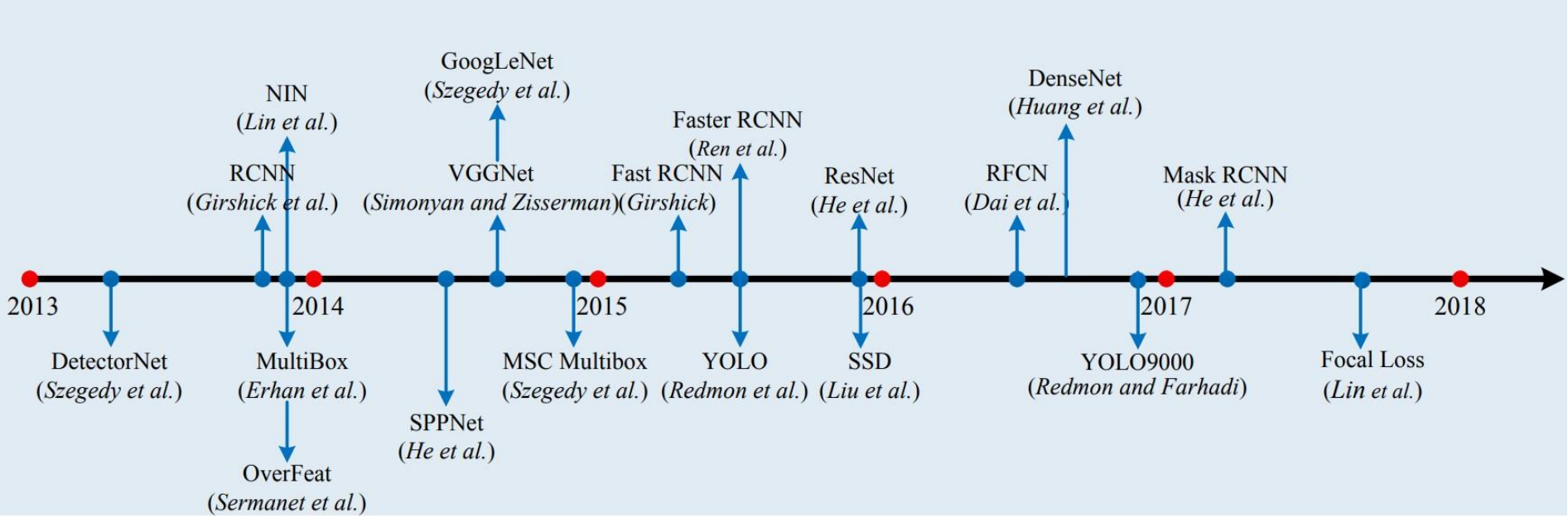

Fine-grained recognition datasets
1000个类别
11m 图片
200个类别
1.1 万张图片
102个类别
1万张图片
ImageNet
2009
Caltech-USCD Birds-200
2010
FGVC-Aircraft
2013
Categories

目标检测/识别：数据集



必须一个好的模型才能建模这些数据

目标检测/识别：经典方法



Liu, Li, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. "Deep learning for generic object detection: A survey." arXiv preprint arXiv:1809.02165 (2018).

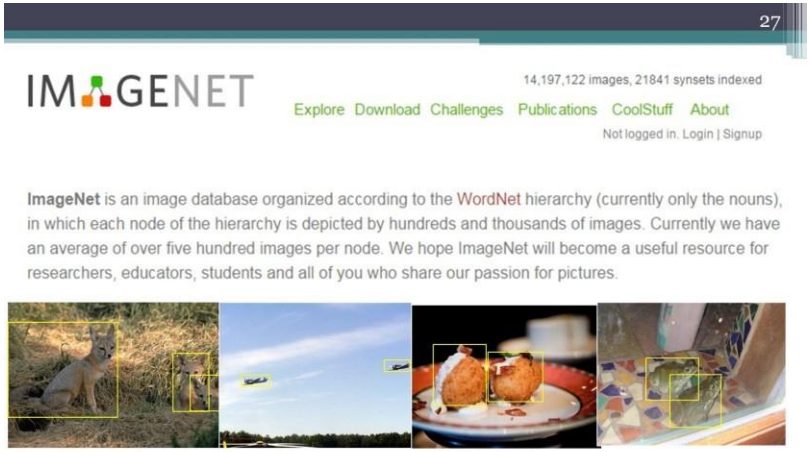
目标检测：竞赛

Pascal VOC

COCO

ImageNet ILSVRC

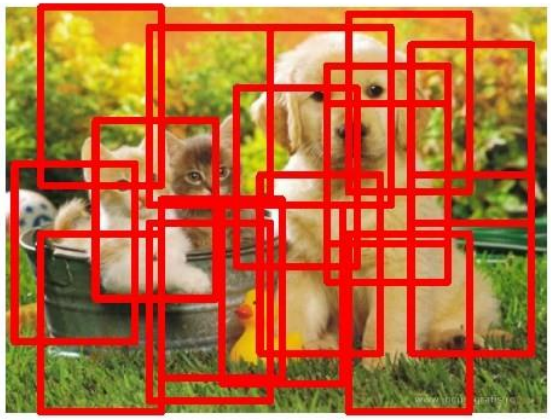
停办



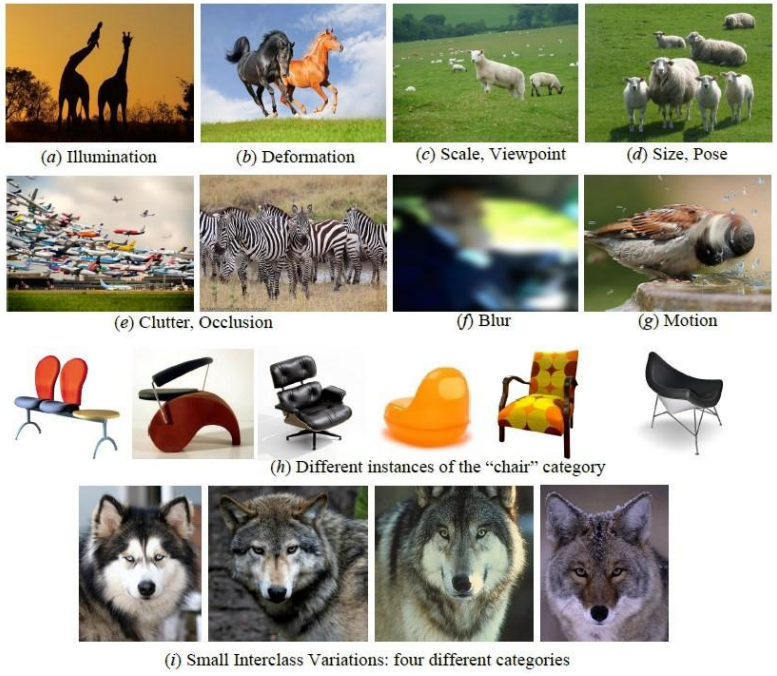
停办

2018年COCO竞赛中国团队包揽全部六项任务的冠军，其中旷视取得4项冠军

目标检测：挑战



太多尺度/位置的样本需要被测试！



目标可能存在的情况太多！

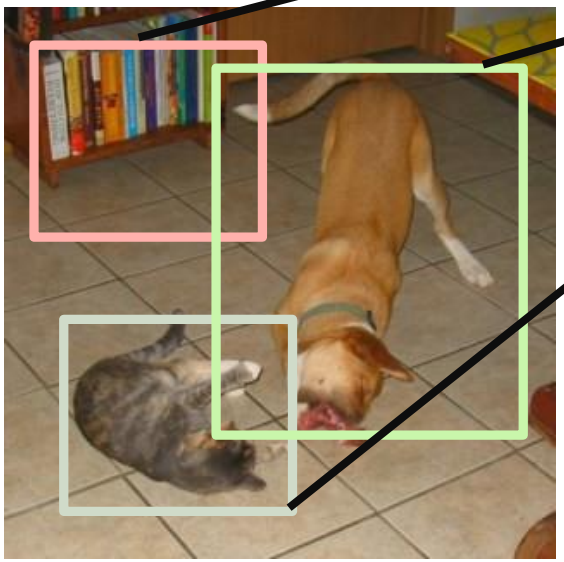
Liu, Li, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. "Deep learning for generic object detection: A survey." arXiv preprint arXiv:1809.02165 (2018).

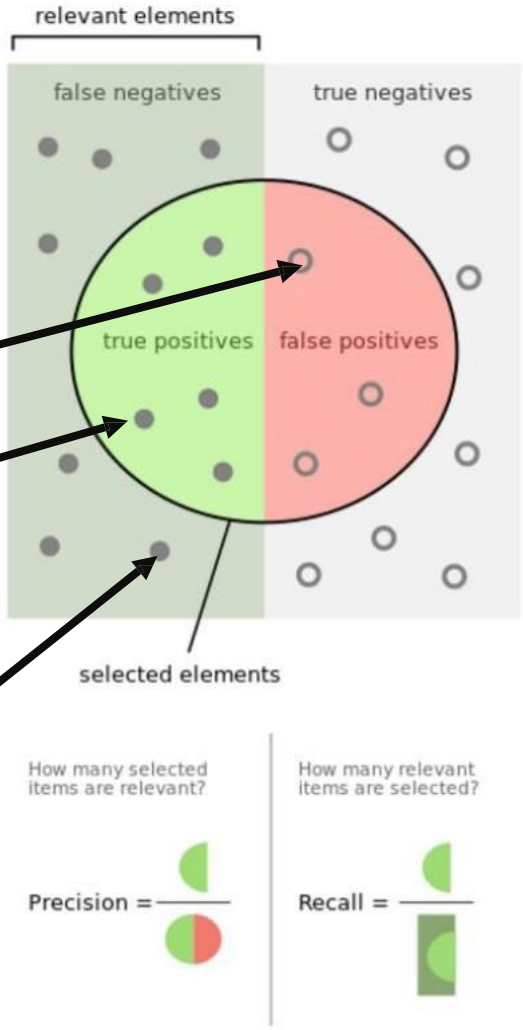
战前准备

战前准备：评价准则

准确率（Precision）

召回率（Recall）





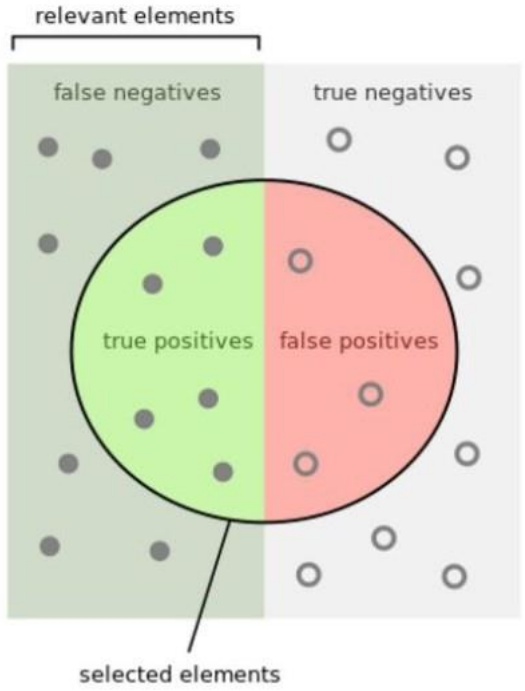
战前准备：评价准则

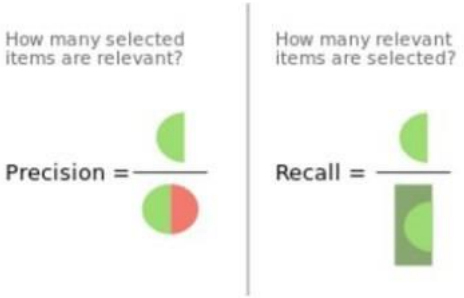
召回率（Recall）

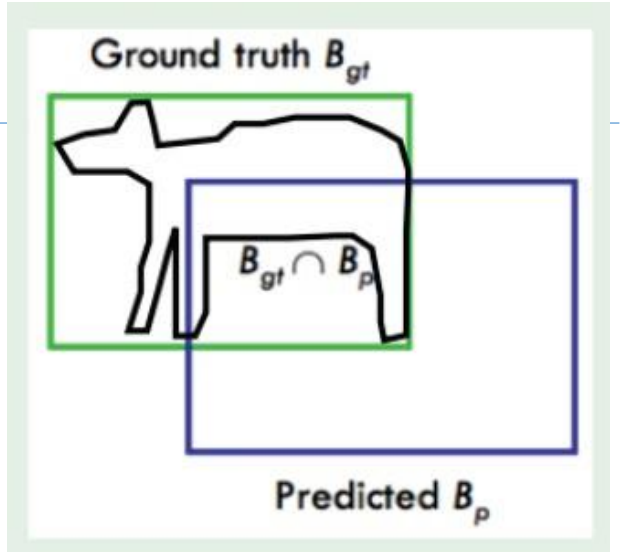
准确率（Precision）

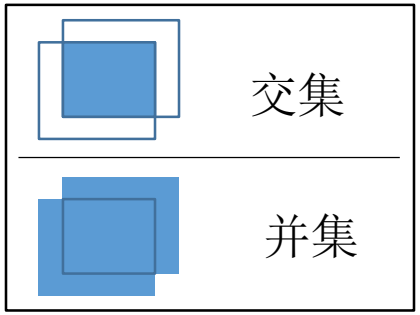
交 并 比 （IOU, Intersection-over-Union）

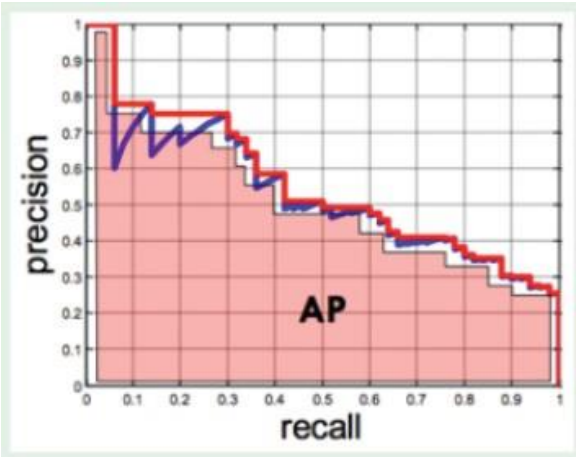
平均准确率（AP,Average Precision）









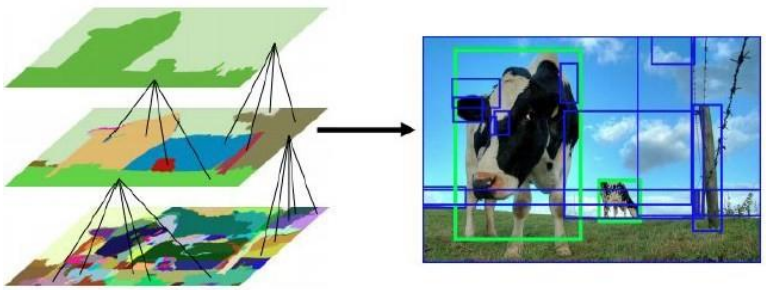


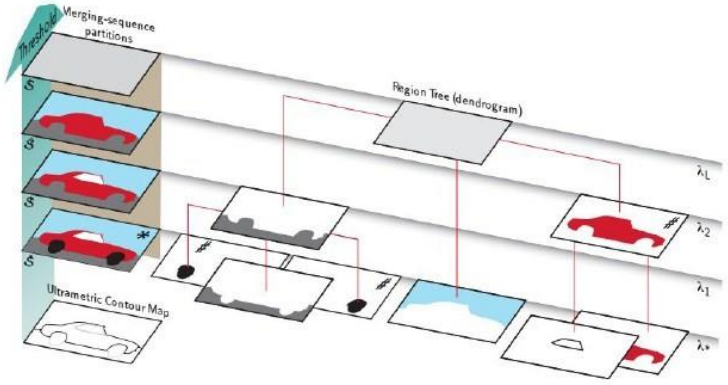
战前准备：滑动窗口（Sliding Window）

滑动窗口（从左向右，从上到下）

对于一张512*512的图像，用一个尺度为50*50的滑动窗口进行遍历，需要滑动几次?

战前准备：目标候选生成（Object Proposal Generation）





Selective Search (SS)

Multiscale Combinatorial Grouping (MCG)

Algorithm 1: Hierarchical Grouping Algorithm

Input: (colour) image
Output: Set of object location hypotheses L

Obtain initial regions $R = \{r_1, \cdots, r_n\}$ using [13]
Initialise similarity set $S = \emptyset$
foreach *Neighbouring region pair* (r_i, r_j) **do**
  Calculate similarity $s(r_i, r_j)$
  $S = S \cup s(r_i, r_j)$

while $S \neq \emptyset$ **do**
  Get highest similarity $s(r_i, r_j) = \max(S)$
  Merge corresponding regions $r_t = r_i \cup r_j$
  Remove similarities regarding r_i : $S = S \setminus s(r_i, r_*)$
  Remove similarities regarding r_j : $S = S \setminus s(r_*, r_j)$
  Calculate similarity set S_t between r_t and its neighbours
  $S = S \cup S_t$
  $R = R \cup r_t$

Extract object location boxes L from all regions in R

在目标检测中常用!

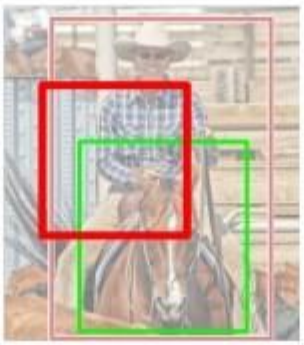
类似于一种层

次聚类算法

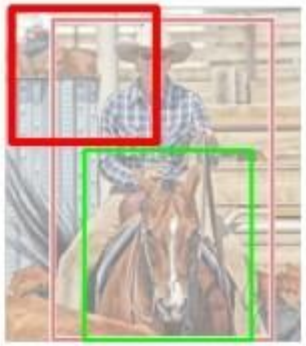
[SS] Uijlings et al. Selective search for object recognition. IJCV 2013
[MCG] Arbeláez, Pont-Tuset et al. Multiscale combinatorial grouping. CVPR 2014

战前准备：难样本挖掘（Hard Negative Mining）

Positive RoI
(IoU Overlap = 1.0)



Hard Negative
(IoU Overlap = 0.3)

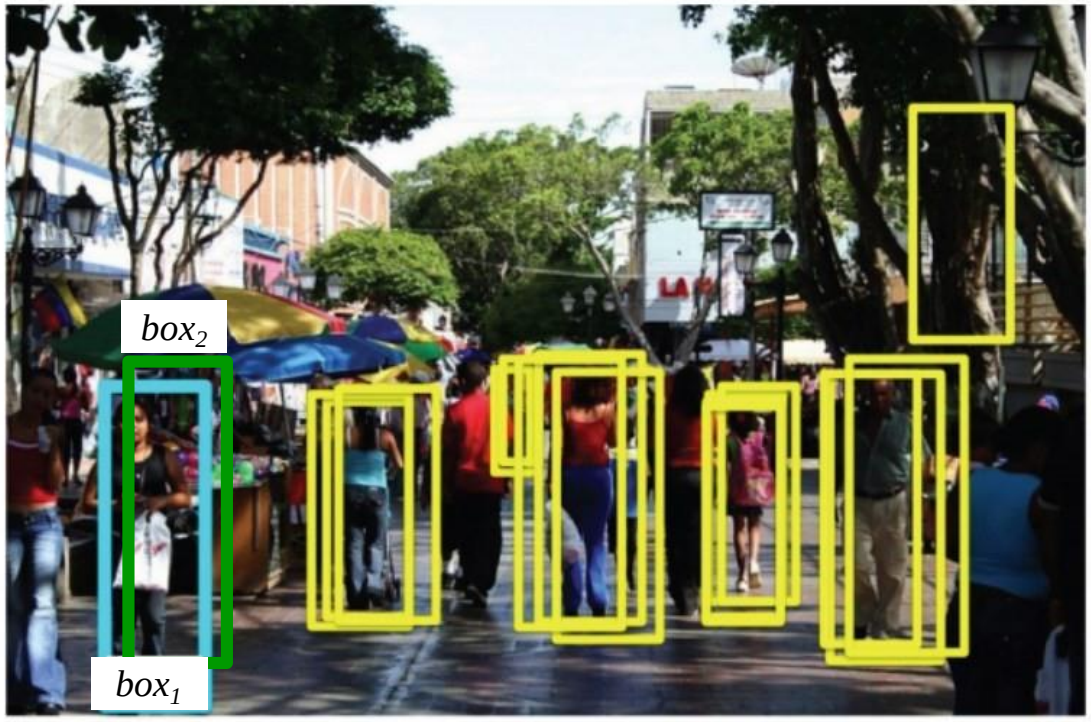


Not Used
(IoU Overlap = 0.0)

对于正负训练样本的重视程度不同；

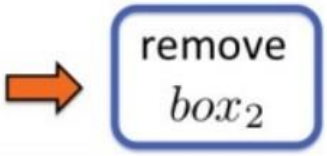
在训练过程中，给予难分的负样本更大的权重。

战前准备：非极大值抑制（Non-Maximum Suppression）



Non-maxima suppression (NMS)

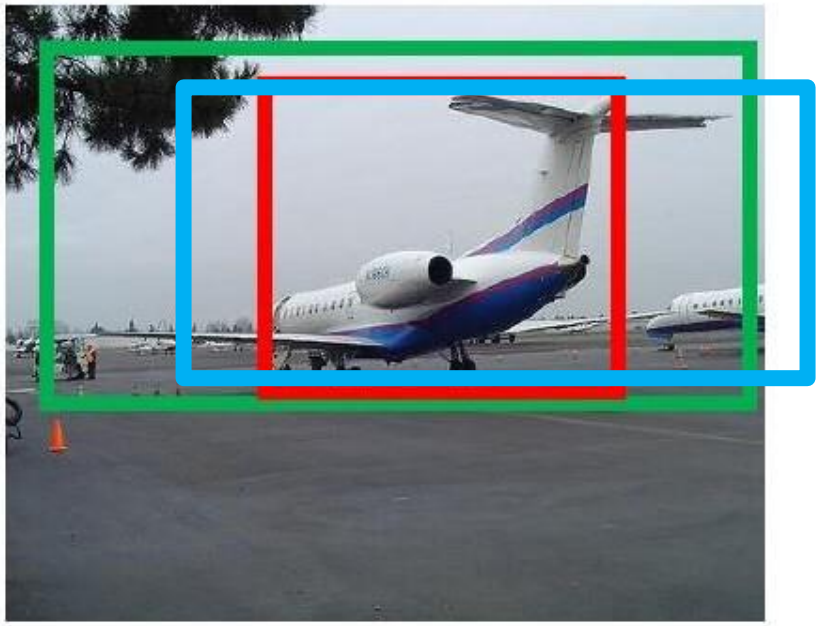
$$\text{overlap} = \frac{\text{area}(box_1 \cup box_2)}{\text{area}(box_1 \cap box_2)} > 0.5$$

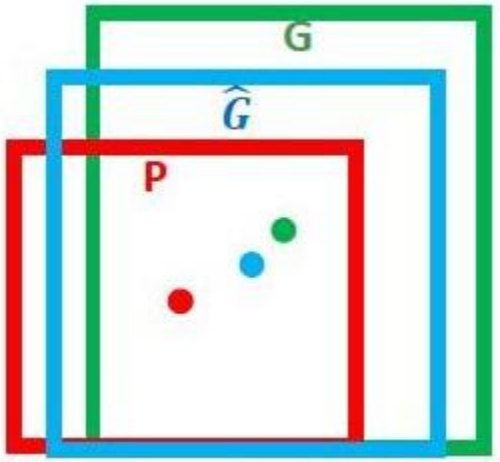


顾名思义就是抑制不是极大值的元素

把与指定框overlap大于一定阈值（通常为0.5）的其他框移除

战前准备：边界框回归（Bounding Box Regression）





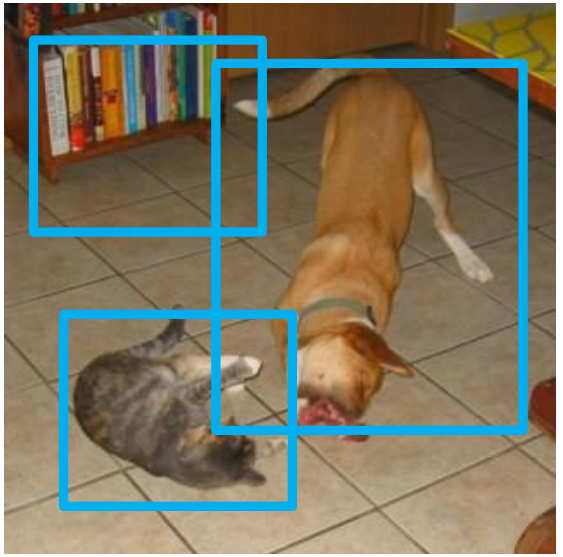
对目标候选的位置进行精化（Refine）

$$\sum_{i\ in\ I}[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2+(\sqrt{w_i}-\sqrt{\hat{w}_i})^2+(\sqrt{h_i}-\sqrt{\hat{h}_i})^2]$$

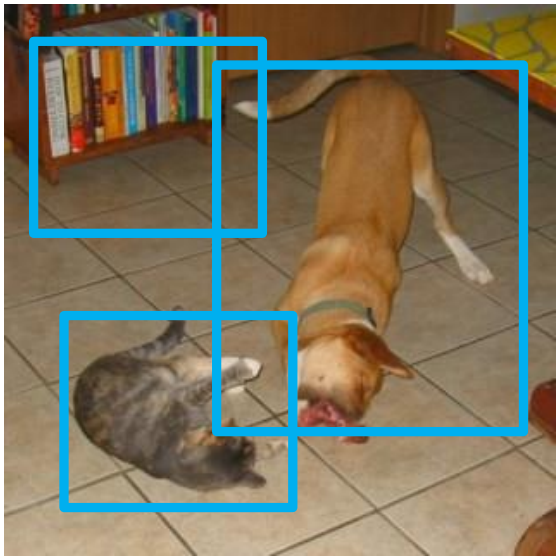
两阶段目标检测方法

哪两阶段？

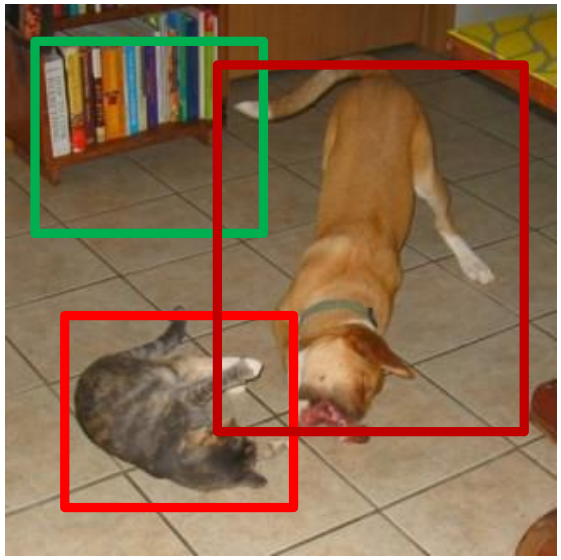
产生目
标候选



哪两阶段？



分类



cat, dog,
background

两阶段目标检测方法：R-CNN

Rich feature hierarchies for accurate object detection and semantic segmentation

Tech report (v5)

Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik
UC Berkeley
{rbg,jdonahue,trevor,malik}@eecs.berkeley.edu

Abstract

Object detection performance, as measured on the canonical PASCAL VOC dataset, has plateaued in the last few years. The best-performing methods are complex ensemble systems that typically combine multiple low-level image features with high-level context. In this paper, we propose a simple and scalable detection algorithm that improves mean average precision (mAP) by more than 30% relative to the previous best result on VOC 2012—achieving a mAP of 53.3%. Our approach combines two key insights: (1) one can apply high-capacity convolutional neural networks (CNNs) to bottom-up region proposals in order to localize and segment objects and (2) when labeled training data is scarce, supervised pre-training for an auxiliary task, followed by domain-specific fine-tuning, yields a significant performance boost. Since we combine region proposals with CNNs, we call our method R-CNN: *Regions with CNN features. We also compare R-CNN to OverFeat, a recently proposed sliding-window detector based on a similar CNN architecture. We find that R-CNN outperforms OverFeat by a large margin on the 200-class ILSVRC2013 detection dataset. Source code for the complete system is available at* http://www.cs.berkeley.edu/~rbg/rcnn.

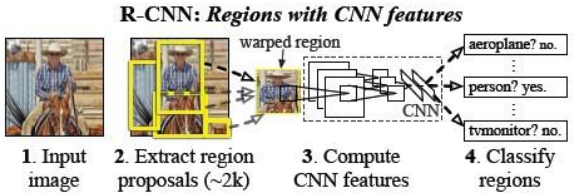


Figure 1: Object detection system overview. Our system (1) takes an input image, (2) extracts around 2000 bottom-up region proposals, (3) computes features for each proposal using a large convolutional neural network (CNN), and then (4) classifies each region using class-specific linear SVMs. R-CNN achieves a mean average precision (mAP) of **53.7% on PASCAL VOC 2010**. For comparison, [39] 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%. On the 200-class **ILSVRC2013 detection dataset, R-CNN's mAP is 31.4%**, a large improvement over OverFeat [34], which had the previous best result at 24.3%.

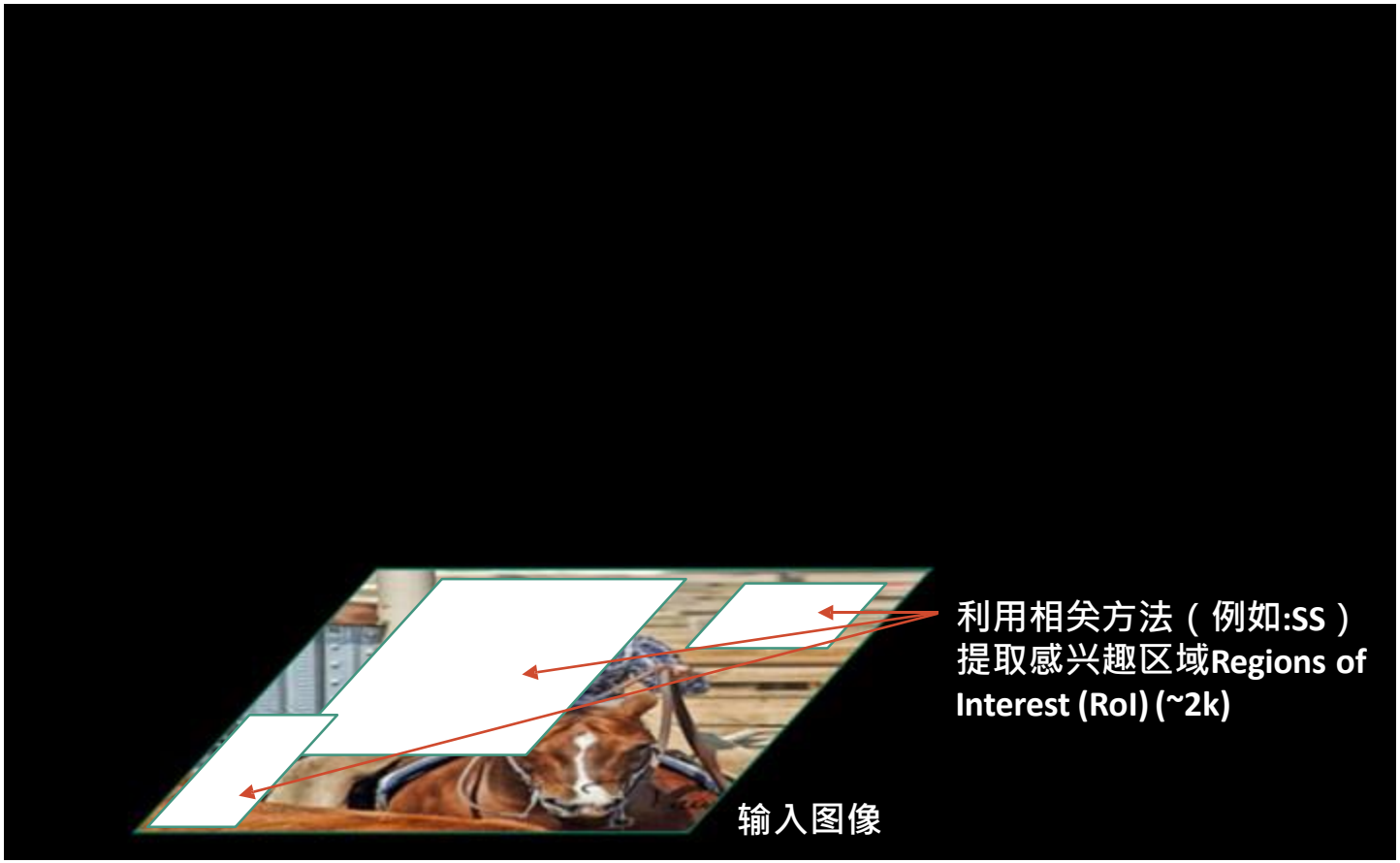
archical, multi-stage processes for computing features that are even more informative for visual recognition.

Fukushima's "neocognitron" [19], a biologically-inspired hierarchical and shift-invariant model for pattern recognition, was an early attempt at just such a process.

Ross B. Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," CVPR 2014

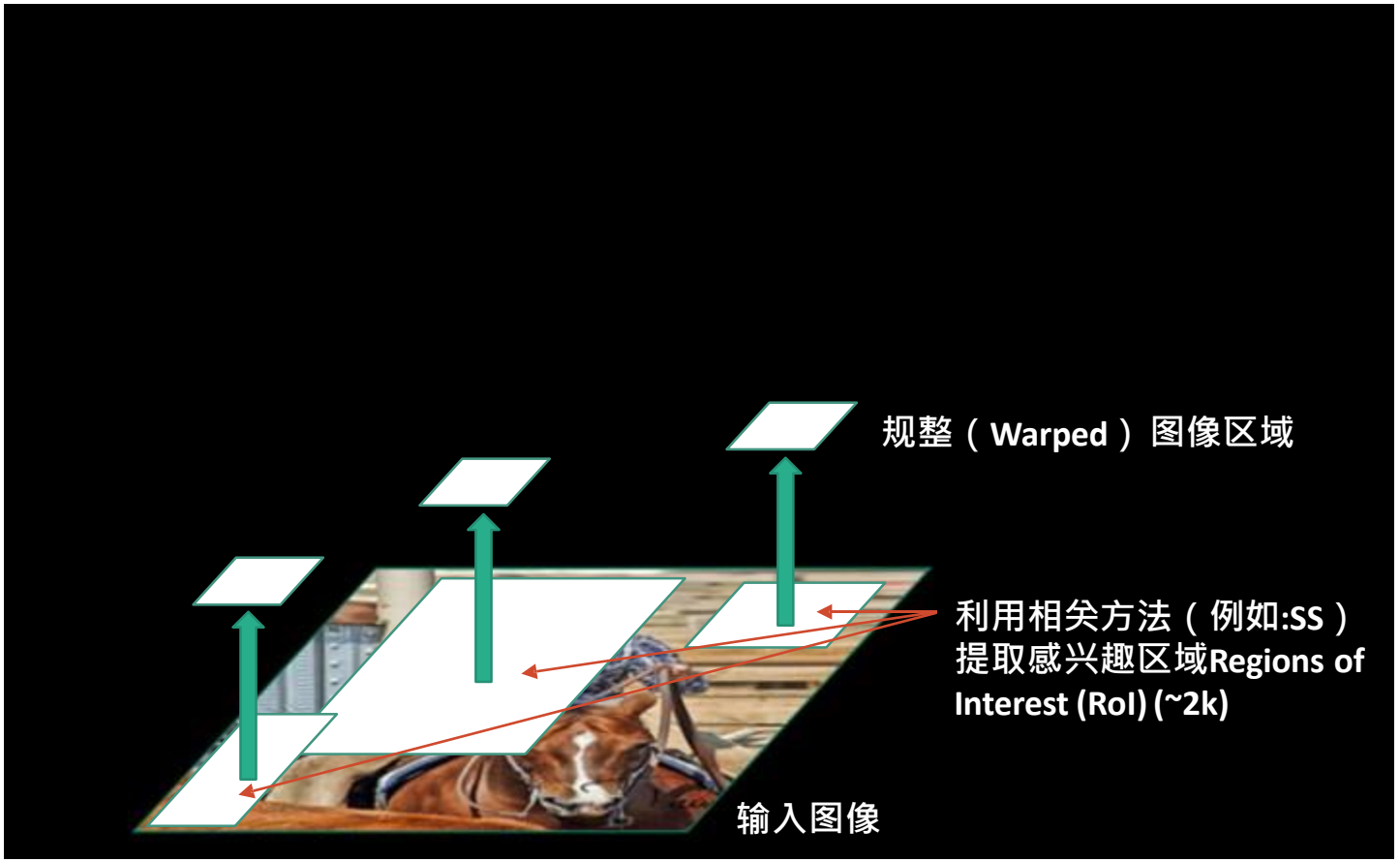
http://arxiv.org/abs/1311.2524

两阶段目标检测方法：R-CNN



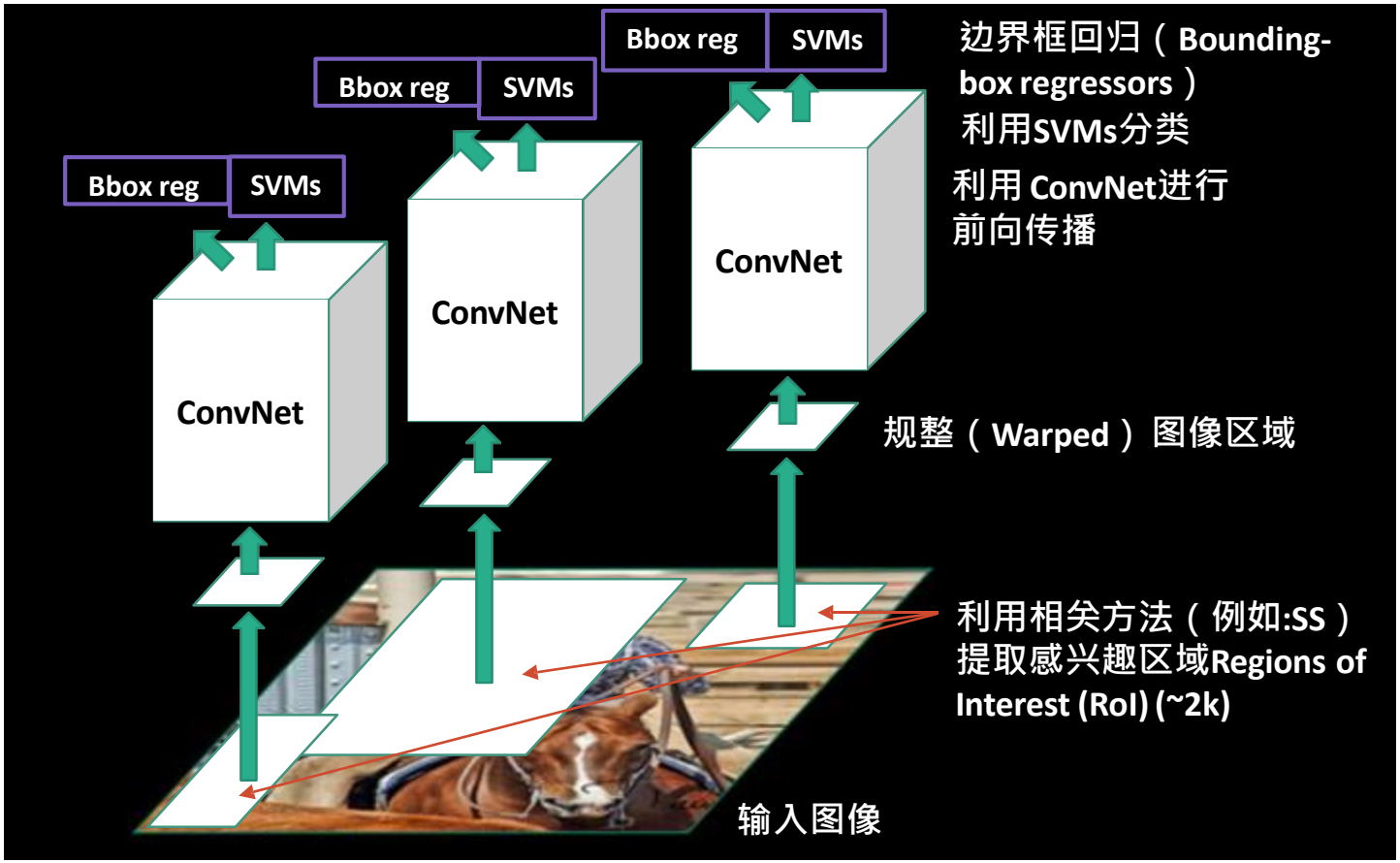
Ross B. Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” CVPR 2014

两阶段目标检测方法：R-CNN



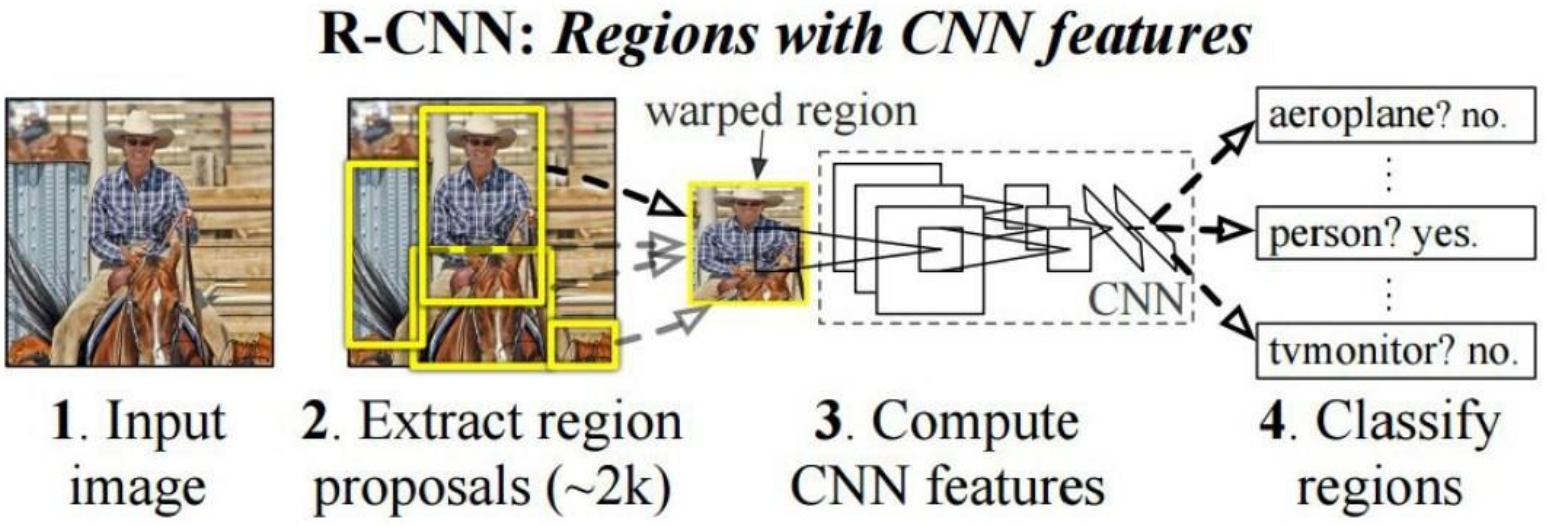
Ross B. Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” CVPR 2014

两阶段目标检测方法：R-CNN



Ross B. Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” CVPR 2014

R-CNN：框架再览



（1）用Selective Search方法产生目标候选；

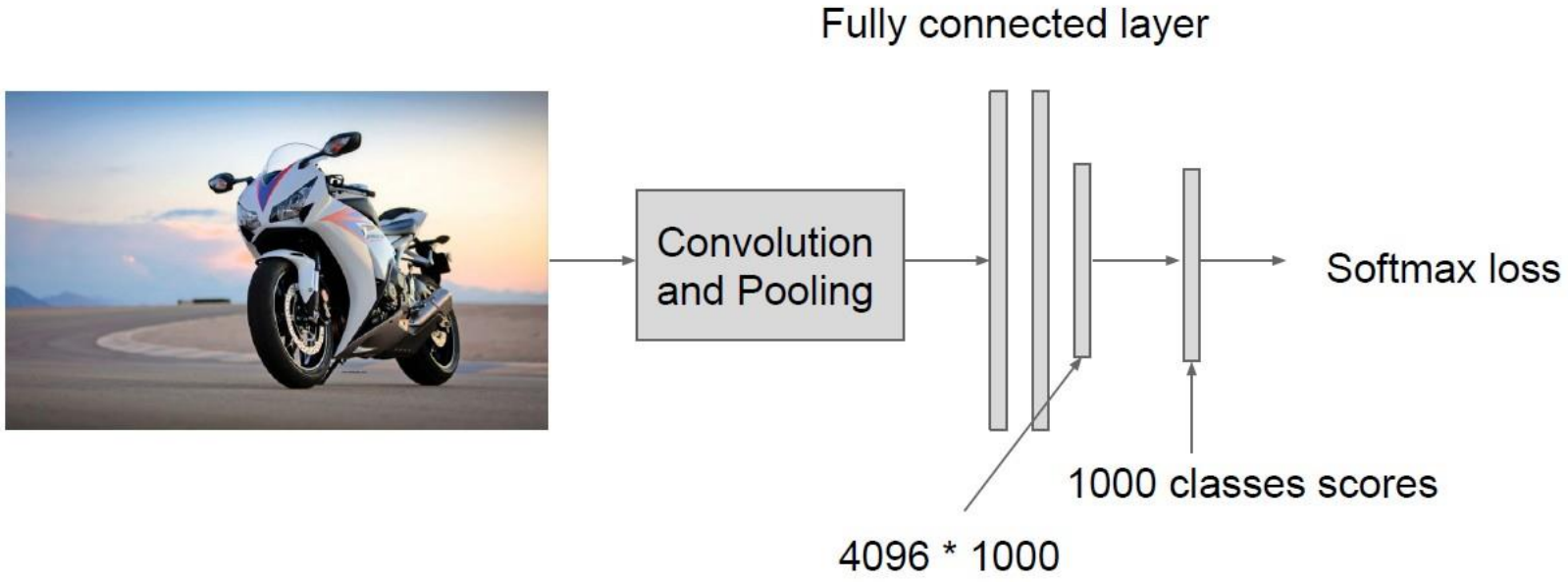
（2）利用CNN提特征；

（3）利用SVM分类。

第一个把CNN用到了目标检测中

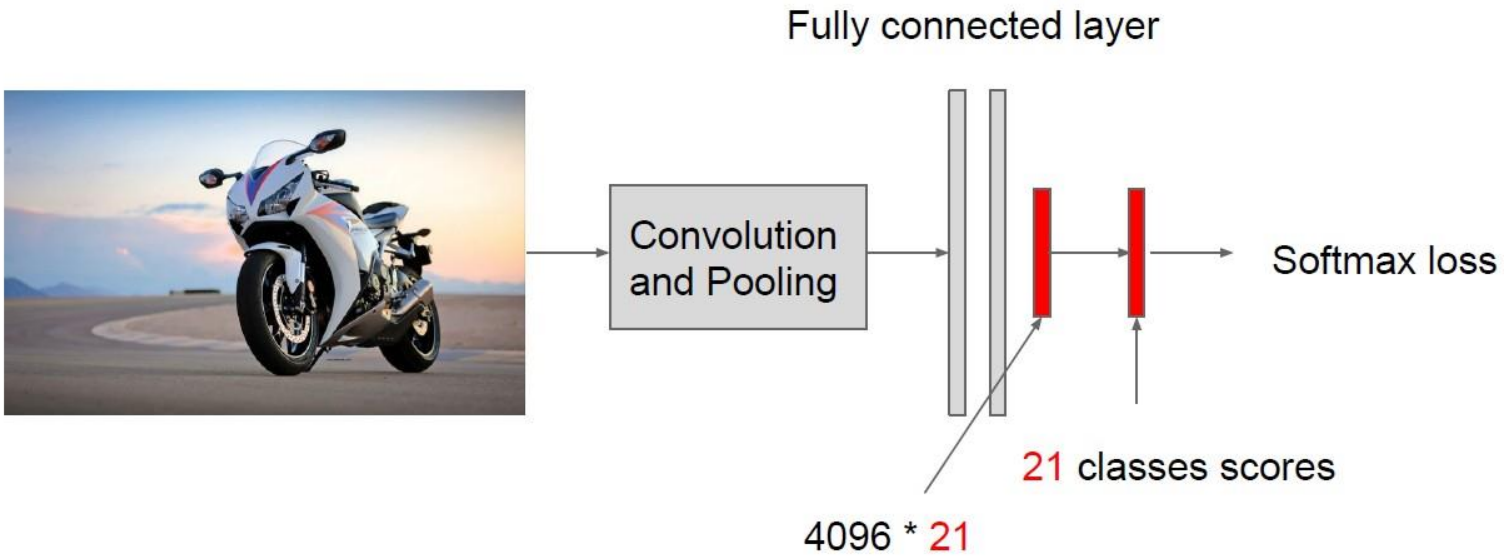
R-CNN：训练

1. 使用ImageNet等数据集，训练一个用于图片分类的CNN模型（或者使用现成的模型，如AlexNet等）



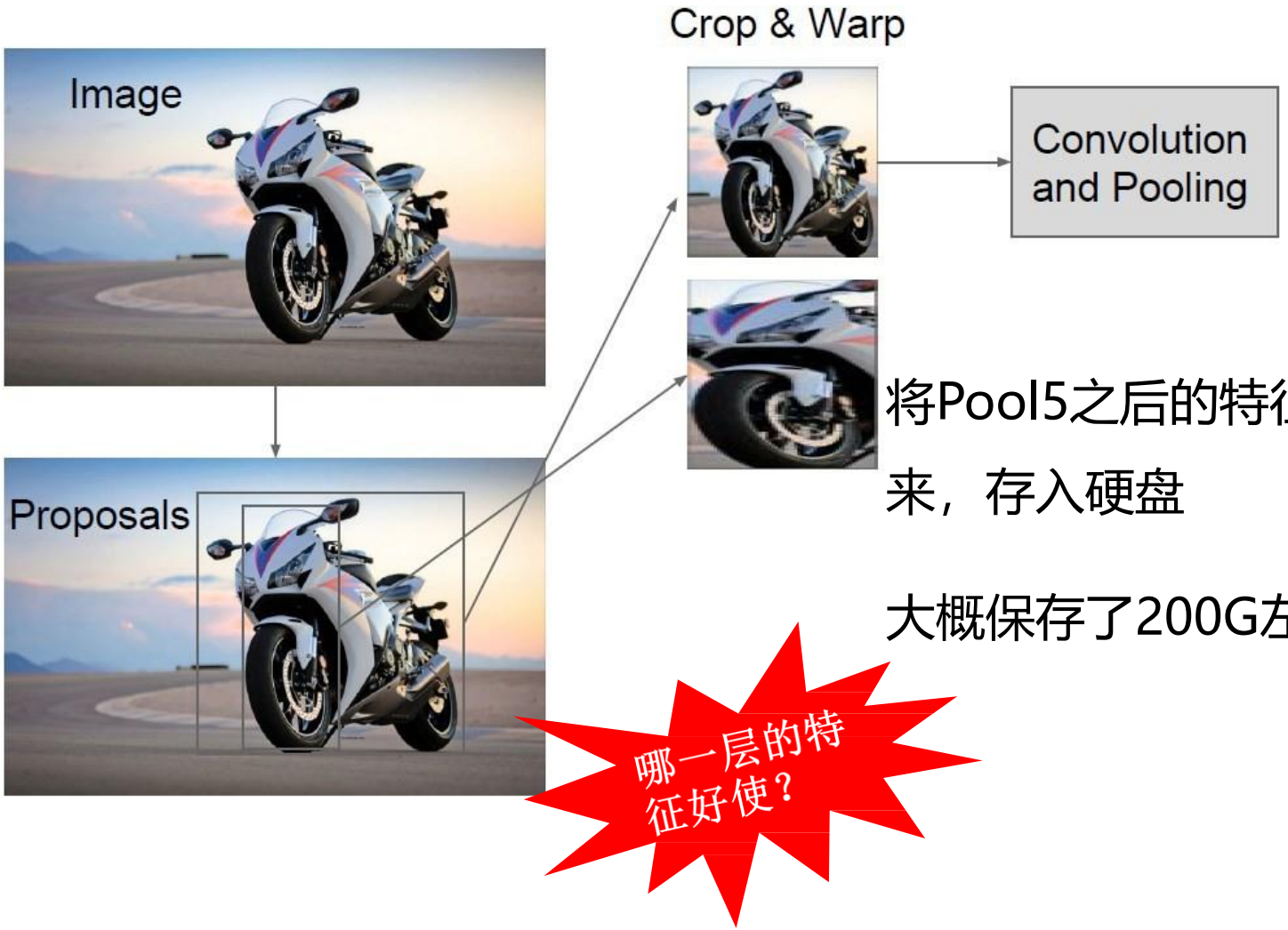
R-CNN：训练

2. 根据检测类别和数据集Fine-Tune网络。以PASCAL VOC数据集为例，我们关注于20个类别+1个背景类。因此，移除之前最后一个FC层（ImageNet分类层），替换为一个小的FC层并进行Fine-Tune。



R-CNN：训练

3. 对目标候选提CNN特征

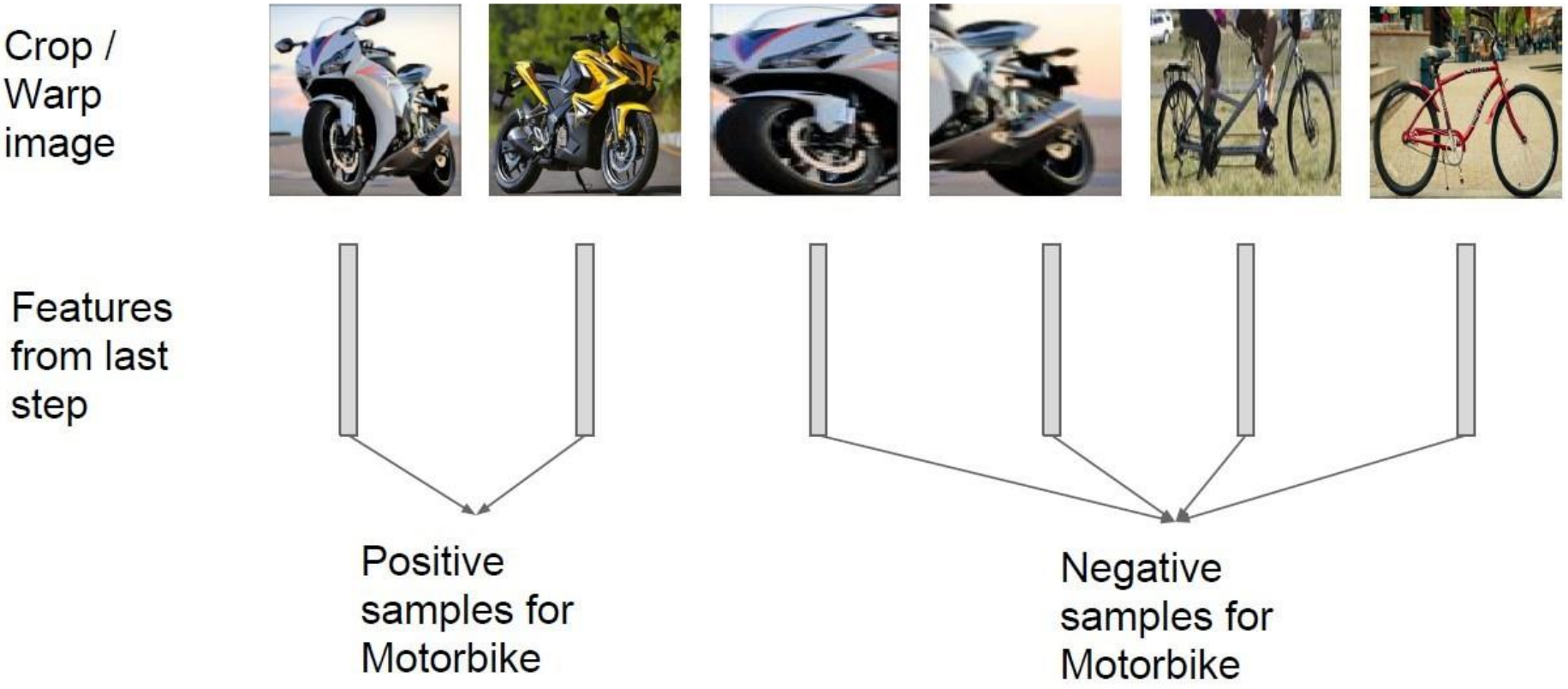


将Pool5之后的特征都保存起来，存入硬盘

大概保存了200G左右的特征

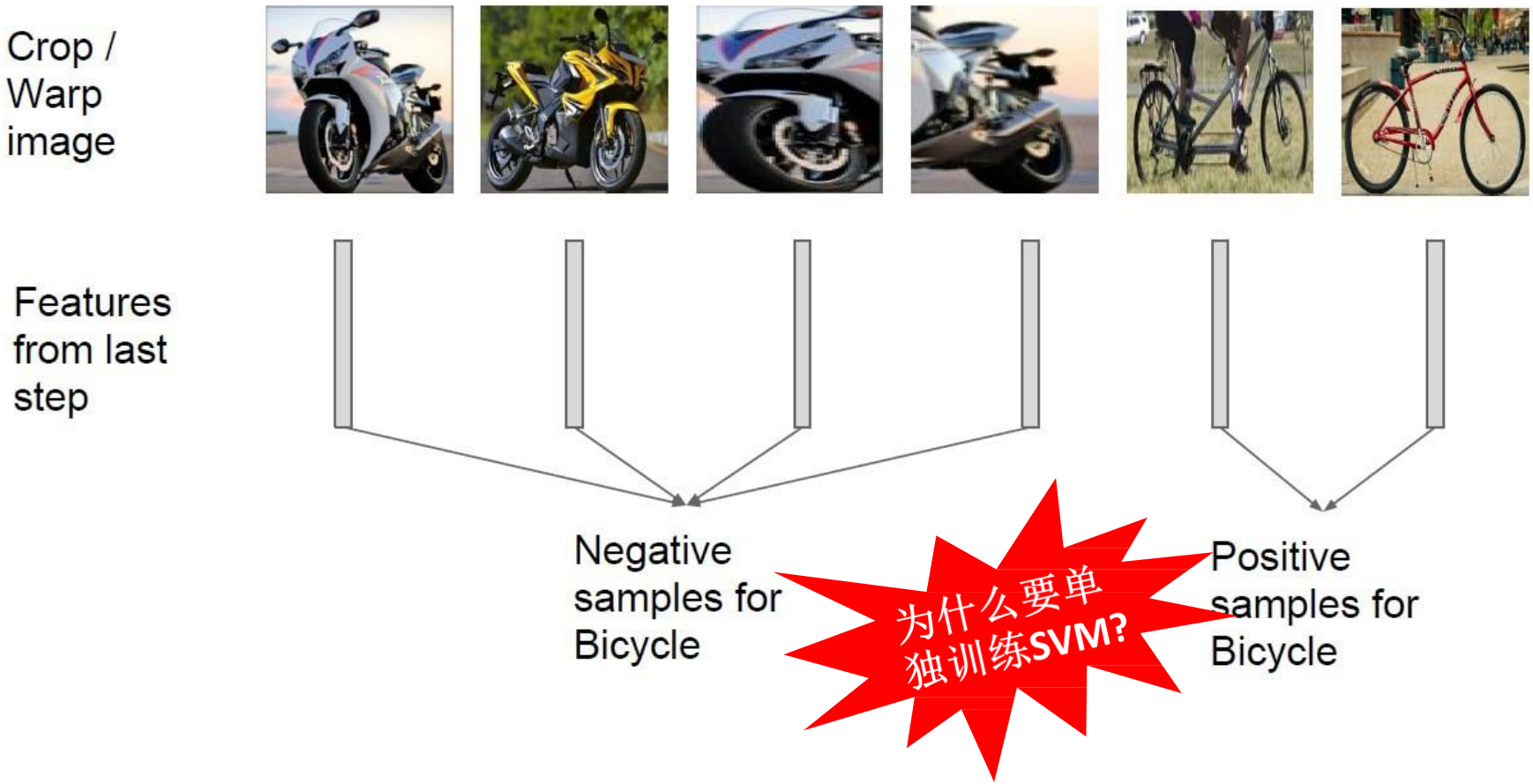
R-CNN：训练

4. 对于每一种检测类，训练SVM。



R-CNN：训练

4. 对于每一种检测类，训练SVM。



R-CNN：解惑

1. 哪一层特征好使?

VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN $pool_5$	51.8	60.2	36.4	27.8	23.2	52.8	60.6	49.2	18.3	47.8	44.3	40.8	56.6	58.7	42.4	23.4	46.1	36.7	51.3	55.7	44.2
R-CNN fc_6	59.3	61.8	43.1	34.0	25.1	53.1	60.6	52.8	21.7	47.8	42.7	47.8	52.5	58.5	44.6	25.6	48.3	34.0	53.1	58.0	46.2
R-CNN fc_7	57.6	57.9	38.5	31.8	23.7	51.2	58.9	51.4	20.0	50.5	40.9	46.0	51.6	55.9	43.3	23.3	48.1	35.3	51.0	57.4	44.7
R-CNN FT $pool_5$	58.2	63.3	37.9	27.6	26.1	54.1	66.9	51.4	26.7	55.5	43.4	43.1	57.7	59.0	45.8	28.1	50.8	40.6	53.1	56.4	47.3
R-CNN FT fc_6	63.5	66.0	47.9	37.7	29.9	62.5	70.2	60.2	32.0	57.9	47.0	53.5	60.1	64.2	52.2	31.3	55.0	50.0	57.7	63.0	53.1
R-CNN FT fc_7	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
R-CNN FT fc_7 BB	**68.1**	**72.8**	**56.8**	**43.0**	**36.8**	**66.3**	**74.2**	**67.6**	**34.4**	**63.5**	**54.5**	**61.2**	**69.1**	**68.6**	**58.7**	**33.4**	**62.9**	**51.1**	**62.5**	**64.8**	**58.5**

RCNN详解： https://blog.csdn.net/weixin_41923961/article/details/80246244

R-CNN：解惑

2.为什么要单独训练SVM?

（1）CNN需要大量的训练样本（对样本质量的要求松）

（2）SVM用少量的样本就可以训练（对样本质量的要求严）

RCNN详解：https://blog.csdn.net/weixin_41923961/article/details/80246244

R-CNN：讨论

精度提高

■ 实验结果

VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sh	sofa	rain	tv	mAP
DPM v5 [18]†	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34	20.7	43.8	38.3	33.4
UVA [34]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	31.8	47.0	44.8	35.1
Regionlets [36]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM [16]†	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	50.2
R-CNN BB	**71.8**	**65.8**	**53.0**	**36.8**	**35.9**	**59.7**	**60.0**	**69.9**	**27.9**	**50.6**	**41.4**	**70.0**	**62.0**	**69.0**	**58.1**	**29.5**	**59.4**	**39.3**	**61.2**	**52.4**	**53.7**

■ 存在不足

- 每个备选区域都进行特征提取，花费大量时间、空间且多有重叠部分
- 训练过程分步完成不是end-to-end
- Warp操作会使图片失真

Ross B. Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” CVPR 2014

两阶段目标检测方法：SPP-Net（一个过渡方法）

Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun

Abstract—Existing deep convolutional neural networks (CNNs) require a fixed-size (*e.g.*, 224×224) input image. This requirement is "artificial" and may reduce the recognition accuracy for the images or sub-images of an arbitrary size/scale. In this work, we equip the networks with another pooling strategy, "spatial pyramid pooling", to eliminate the above requirement. The new network structure, called SPP-net, can generate a fixed-length representation regardless of image size/scale. Pyramid pooling is also robust to object deformations. With these advantages, SPP-net should in general improve all CNN-based image classification methods. On the ImageNet 2012 dataset, we demonstrate that SPP-net boosts the accuracy of a variety of CNN architectures despite their different designs. On the Pascal VOC 2007 and Caltech101 datasets, SPP-net achieves state-of-the-art classification results using a single full-image representation and no fine-tuning.

The power of SPP-net is also significant in object detection. Using SPP-net, we compute the feature maps from the entire image only once, and then pool features in arbitrary regions (sub-images) to generate fixed-length representations for training the detectors. This method avoids repeatedly computing the convolutional features. In processing test images, our method is 24-102× faster than the R-CNN method, while achieving better or comparable accuracy on Pascal VOC 2007.

In ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014, our methods rank #2 in object detection and #3 in image classification among all 38 teams. This manuscript also introduces the improvement made for this competition.

空间金字塔Pooling

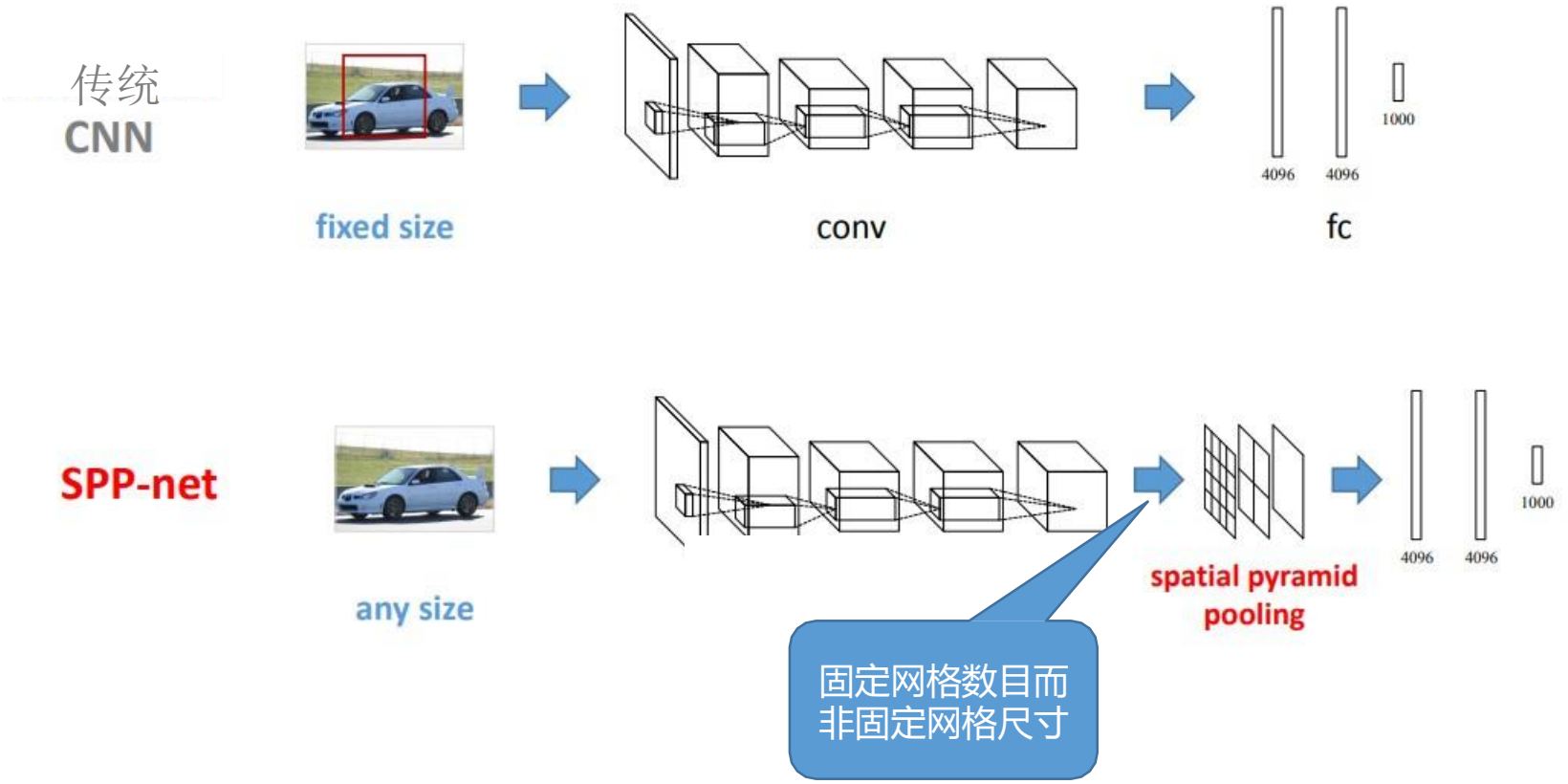
He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Spatial pyramid pooling in deep convolutional networks for visual recognition.", ECCV 2014, PAMI2015

http://arxiv.org/abs/1406.4729

@Kaiming He

两阶段目标检测方法：SPP-Net（一个过渡方法）

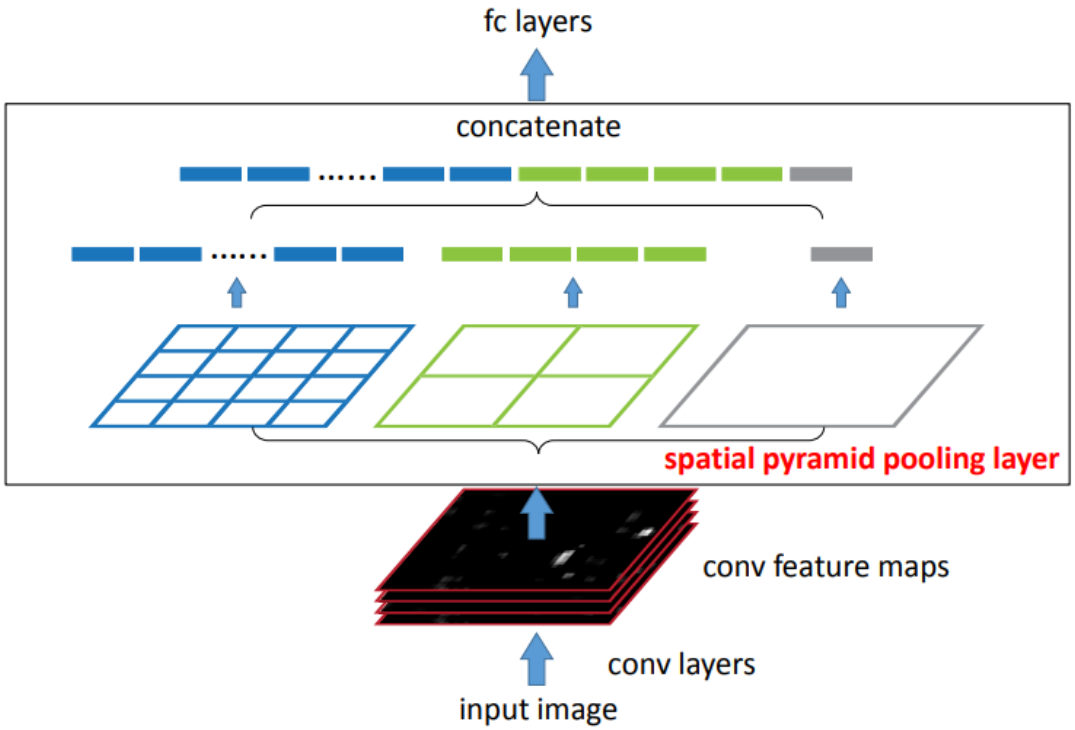
■ 动机



He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Spatial pyramid pooling in deep convolutional networks for visual recognition.“, ECCV 2014

@Kaiming He

SPP-Net：框架

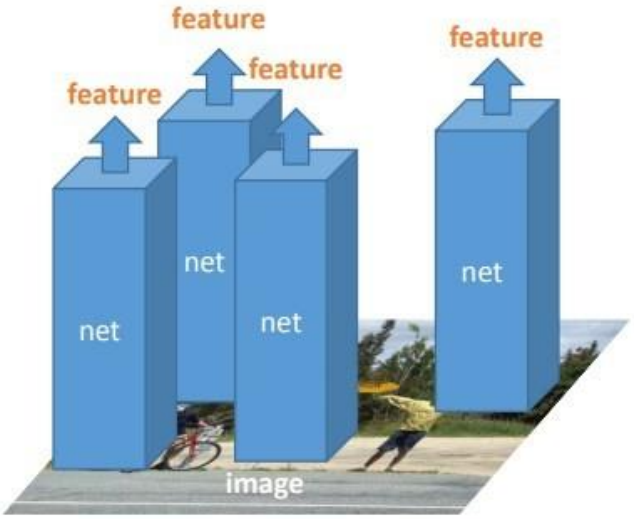


- 可变的尺度
  - 多尺度训练
  - 多尺度测试
  - 共享整张图片
- 多级Pooling
  - 对形变鲁棒
- 特征级操作
  - 对Region进行Pooling

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Spatial pyramid pooling in deep convolutional networks for visual recognition.“, ECCV 2014

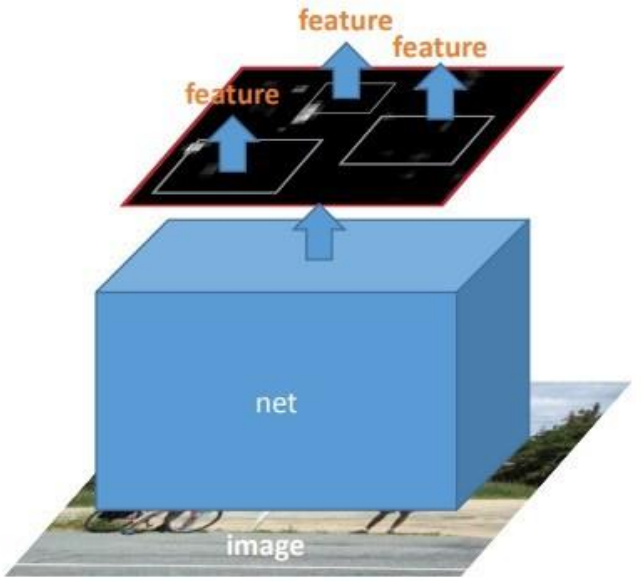
SPP-Net：RCNN vs. SPP

- 图像区域 vs. 特征区域



R-CNN

2000 nets on image regions



SPP-net

1 net on full image

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Spatial pyramid pooling in deep convolutional networks for visual recognition.", ECCV 2014

@Kaiming He

SPP-Net：实验结果

	mAP
NUS	**37.2**
ours, multi SPP-nets	**35.1**
UvA	32.0
ours, 1 SPP-net	31.8
Southeast-CASIA	30.4
1-HKUST	28.8
CASIA_CRIPAC_2	28.6

"provided data" track

	SPP-net	RCNN
GPU time / img	0.6s	32s
40k test imgs	8 hours	**15 days**

cost of a single model

ILSVRC 2014 DET Results

训练依旧分为多个阶段！步骤繁琐: 微调网络+训练SVM+训练回归器

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Spatial pyramid pooling in deep convolutional networks for visual recognition.", ECCV 2014

两阶段目标检测方法：Fast R-CNN

Fast R-CNN

Ross Girshick
Microsoft Research
rbg@microsoft.com

Abstract

This paper proposes a Fast Region-based Convolutional Network method (Fast R-CNN) *for object detection. Fast R-CNN builds on previous work to efficiently classify object proposals using deep convolutional networks. Compared to previous work, Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy. Fast R-CNN trains the very deep VGG16 network 9× faster than R-CNN, is 213× faster at test-time, and achieves a higher mAP on PASCAL VOC 2012. Compared to SPPnet, Fast R-CNN trains VGG16 3× faster, tests 10× faster, and is more accurate. Fast R-CNN is implemented in Python and C++ (using Caffe) and is available under the open-source MIT License at https://github.com/rbgirshick/fast-rcnn.*

while achieving top accuracy on PASCAL VOC 2012 [7] with a mAP of 66% (vs. 62% for R-CNN).[1]

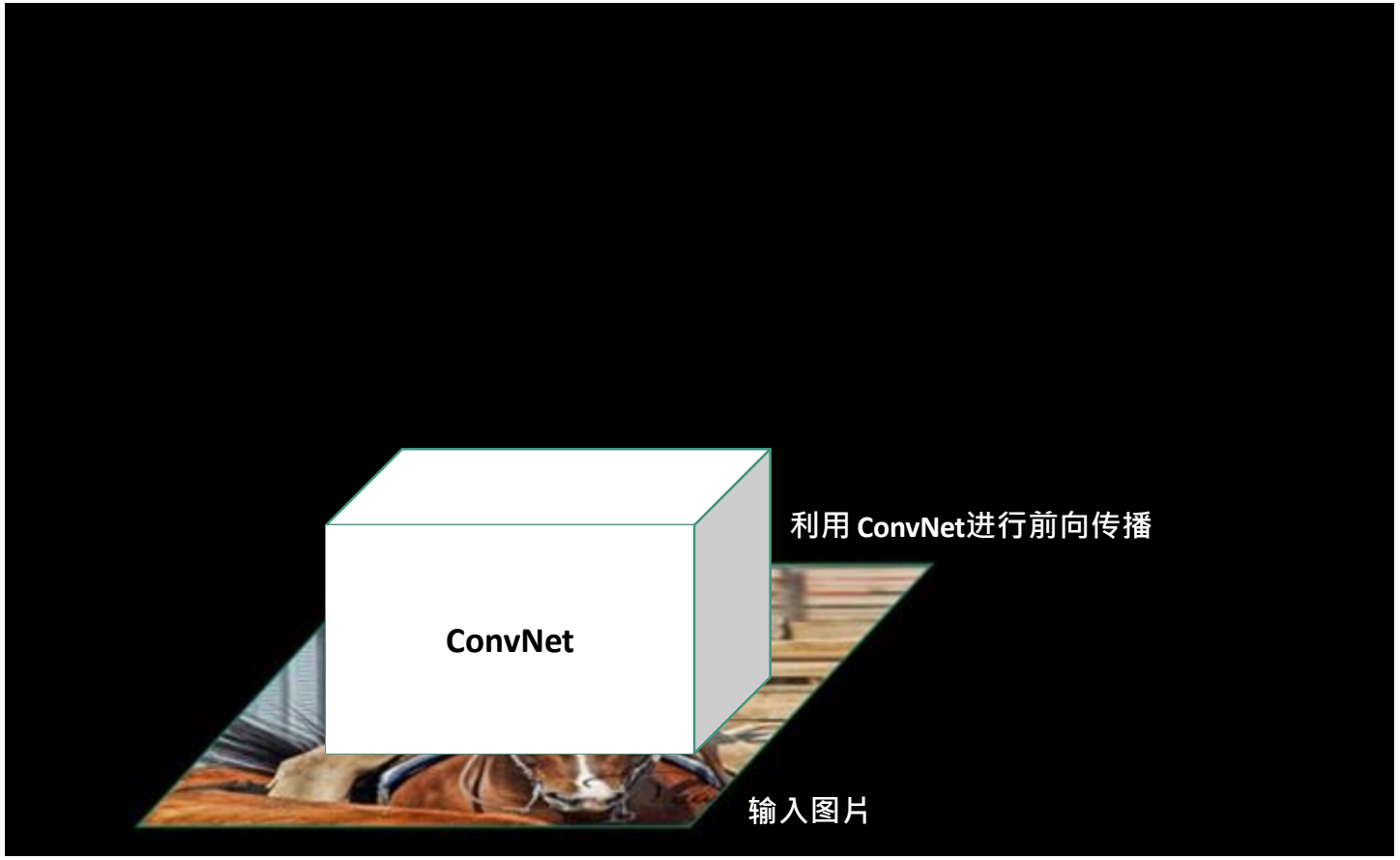
1.1. R-CNN and SPPnet

The Region-based Convolutional Network method (R-CNN) [9] achieves excellent object detection accuracy by using a deep ConvNet to classify object proposals. R-CNN, however, has notable drawbacks:

1. **Training is a multi-stage pipeline.** R-CNN first fine-tunes a ConvNet on object proposals using log loss. Then, it fits SVMs to ConvNet features. These SVMs act as object detectors, replacing the softmax classifier learnt by fine-tuning. In the third training stage, bounding-box regressors are learned.
2. **Training is expensive in space and time.** For SVM

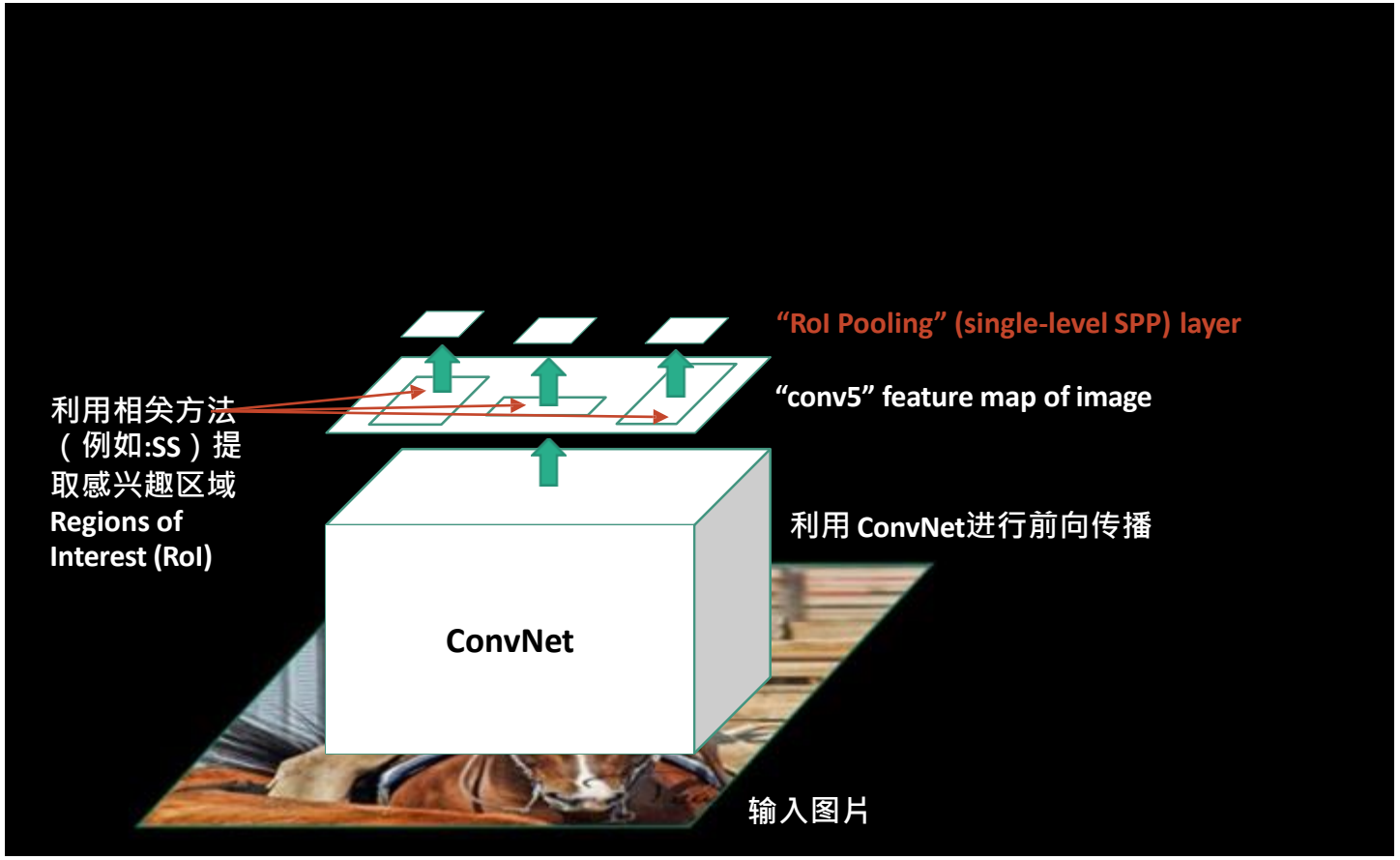
Ross B. Girshick, “Fast R-CNN,” ICCV 2015

两阶段目标检测方法：Fast R-CNN



Ross B. Girshick, “Fast R-CNN,” ICCV 2015

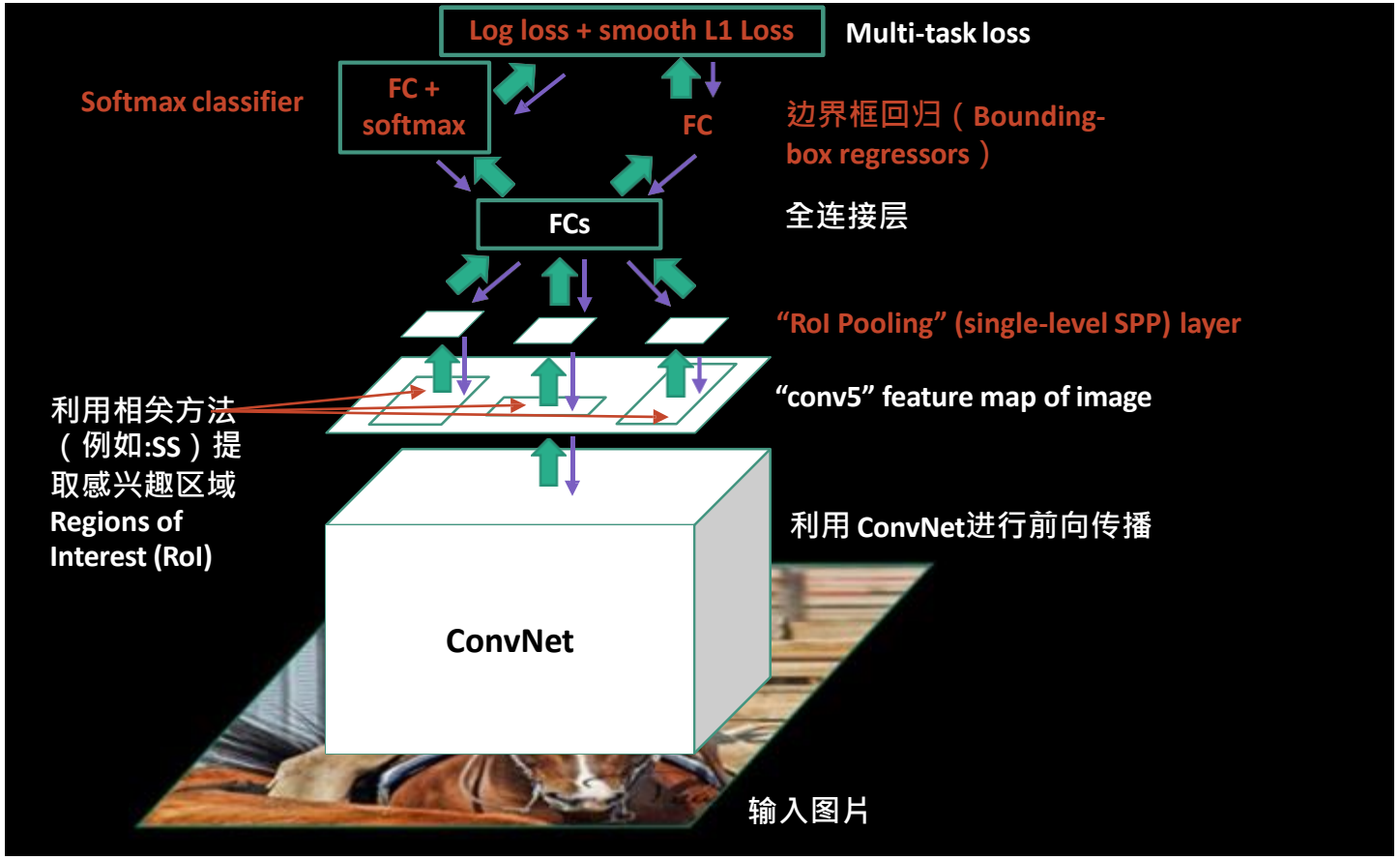
两阶段目标检测方法：Fast R-CNN



Ross B. Girshick, “Fast R-CNN,” ICCV 2015

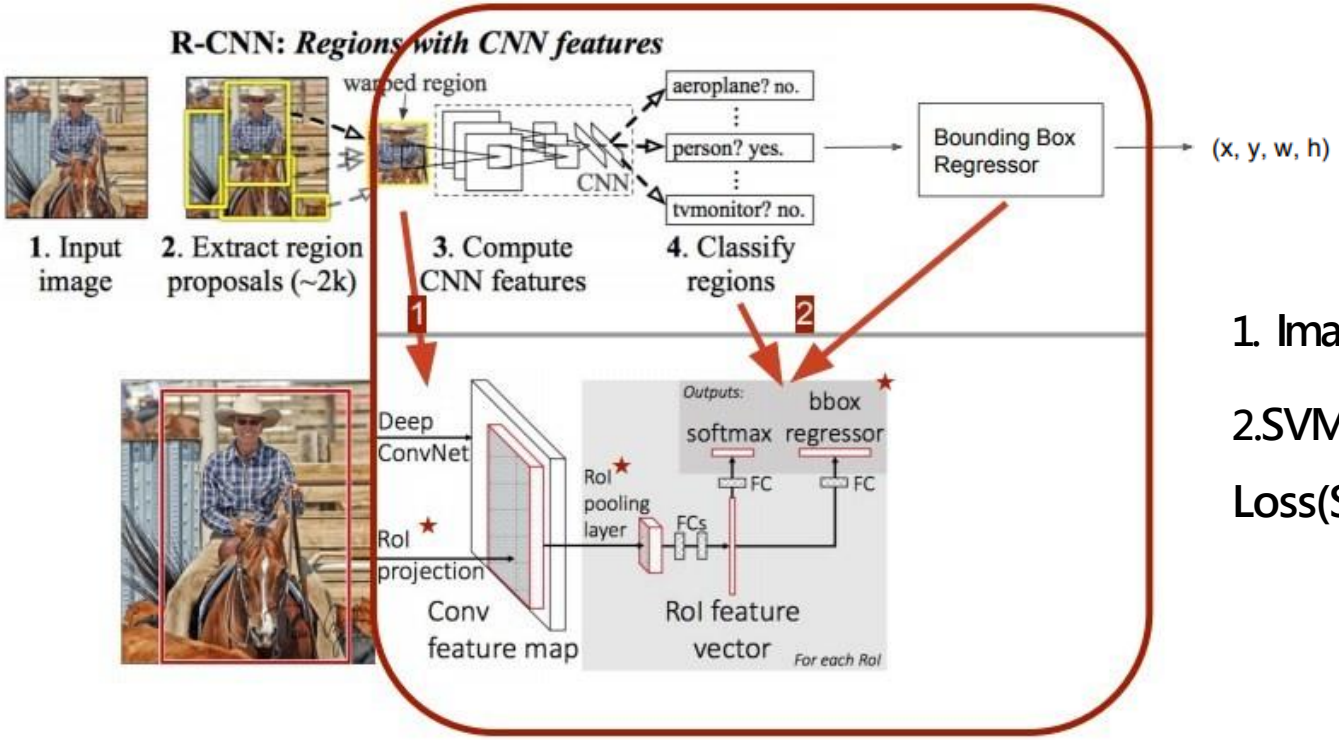
两阶段目标检测方法：Fast R-CNN

注意，这里没有使用SVM



Ross B. Girshick, “Fast R-CNN,” ICCV 2015

Fast R-CNN：框架再览

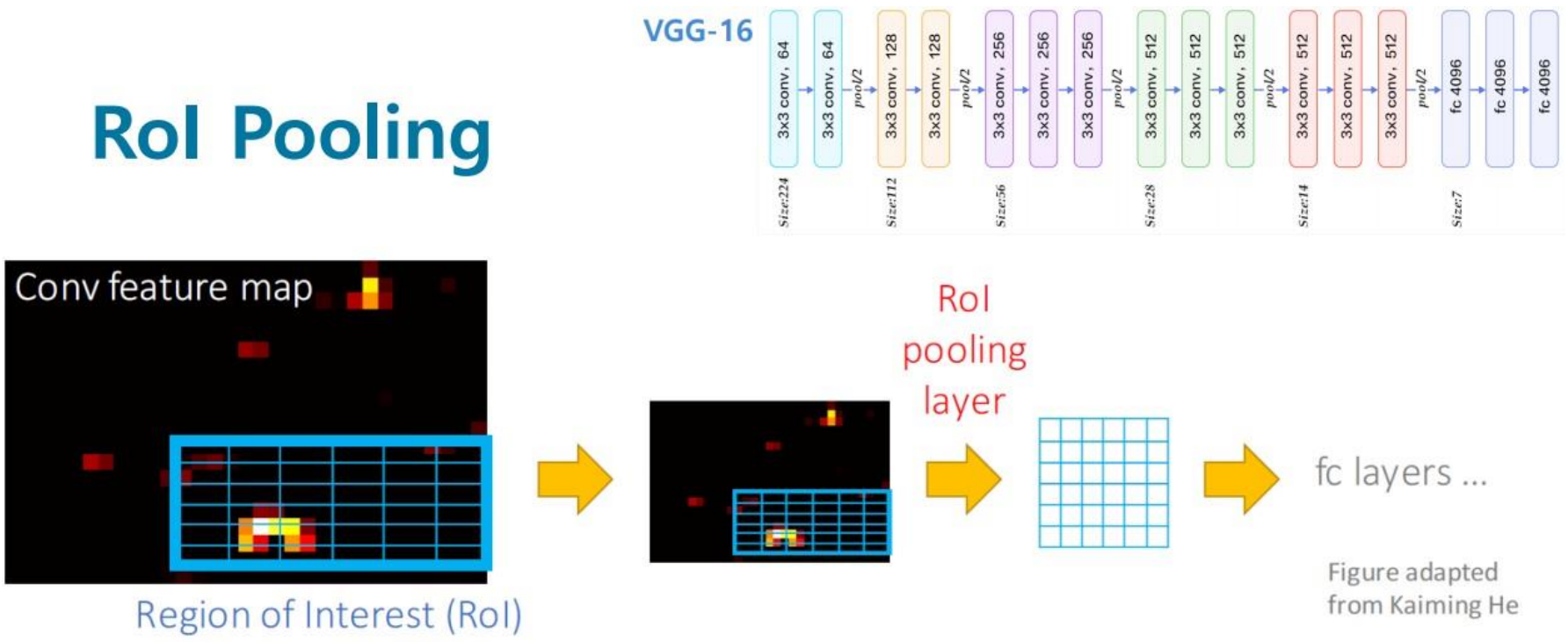


1. Image Warp->Feature Warp

2.SVM+Regressor->Multitask Loss(SoftMax+Regressor)

Ross B. Girshick, “Fast R-CNN,” ICCV 2015

Fast R-CNN：ROI Pooling



ROI Pooling就是一个简单SPP Pooling，只具有一个空间层级

RoI in Conv feature map : 21x14 → 3x2 max pooling with stride(3, 2) → output : 7x7
RoI in Conv feature map : 35x42 → 5x6 max pooling with stride(5, 6) → output : 7x7

Ross B. Girshick, “Fast R-CNN,” ICCV 2015

Fast R-CNN：训练&测试

1. 输入一张图片和一堆bounding boxes(由SS产生)；

2. 产生卷积特征图

3. 对于每一个box，通过ROI Pooling层得到一个定长特征向量

4. 输出：（1）K+1类别概率，（2）回归框位置

训练使用难样本挖掘以使网络获得高判别力，从而精准定位目标

- Loss function

True box coordinates

Predicted box coordinates

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

True class

Predicted class scores

Log loss

$L_{cls}(p,u)=-\log p_u$

Smooth L1 loss

$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{x,y,w,h\}} \text{smooth}_{L_1}(t_i^u - v_i),$$

in which

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

Ross B. Girshick, “Fast R-CNN,” ICCV 2015

Fast R-CNN：总结

■ 实验结果

精度提高

method	train set	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	[illegible]	tra[illegible]	tv	mAP
SPPnet BB [11]†	07 \ diff	73.9	72.3	62.5	51.5	44.4	74.4	73.0	74.4	42.3	73.6	57.7	70.3	74.6	74.3	54.2	34.0	56.4	76.4	67.9	73.5	63.1
R-CNN BB [10]	07	73.4	77.0	63.4	45.4	**44.6**	75.1	78.1	79.8	40.5	73.7	62.2	79.4	78.1	73.1	64.2	**35.6**	66.8	67.2	70.4	**71.1**	66.0
FRCN [ours]	07	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8	66.9
FRCN [ours]	07 \ diff	74.6	**79.0**	68.6	57.0	39.3	79.5	**78.6**	81.9	**48.0**	74.0	67.4	80.5	80.7	74.1	69.6	31.8	67.1	68.4	75.3	65.5	68.1
FRCN [ours]	07+12	**77.0**	78.1	**69.3**	**59.4**	38.3	**81.6**	**78.6**	**86.7**	42.8	**78.8**	**68.9**	**84.7**	**82.0**	**76.6**	**69.9**	31.8	**70.1**	**74.8**	**80.4**	70.4	**70.0**

	Fast R-CNN			R-CNN			SPPnet
	S	**M**	**L**	**S**	**M**	**L**	†**L**
train time (h)	**1.2**	2.0	9.5	22	28	84	25
train speedup	**18.3**×	14.0×	8.8×	1×	1×	1×	3.4×
test rate (s/im)	0.10	0.15	0.32	9.8	12.1	47.0	2.3
▷ with SVD	**0.06**	0.08	0.22	-	-	-	-
test speedup	98×	80×	146×	1×	1×	1×	20×
▷ with SVD	169×	150×	**213**×	-	-	-	-
VOC07 mAP	57.1	59.2	**66.9**	58.5	60.2	66.0	63.1
▷ with SVD	56.5	58.7	66.6	-	-	-	-

速度提高

■ 存在不足

- Region proposals是通过SS提取的，在一定程度上限制了速度

Ross B. Girshick, “Fast R-CNN,” ICCV 2015

两阶段目标检测方法：Faster R-CNN

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun

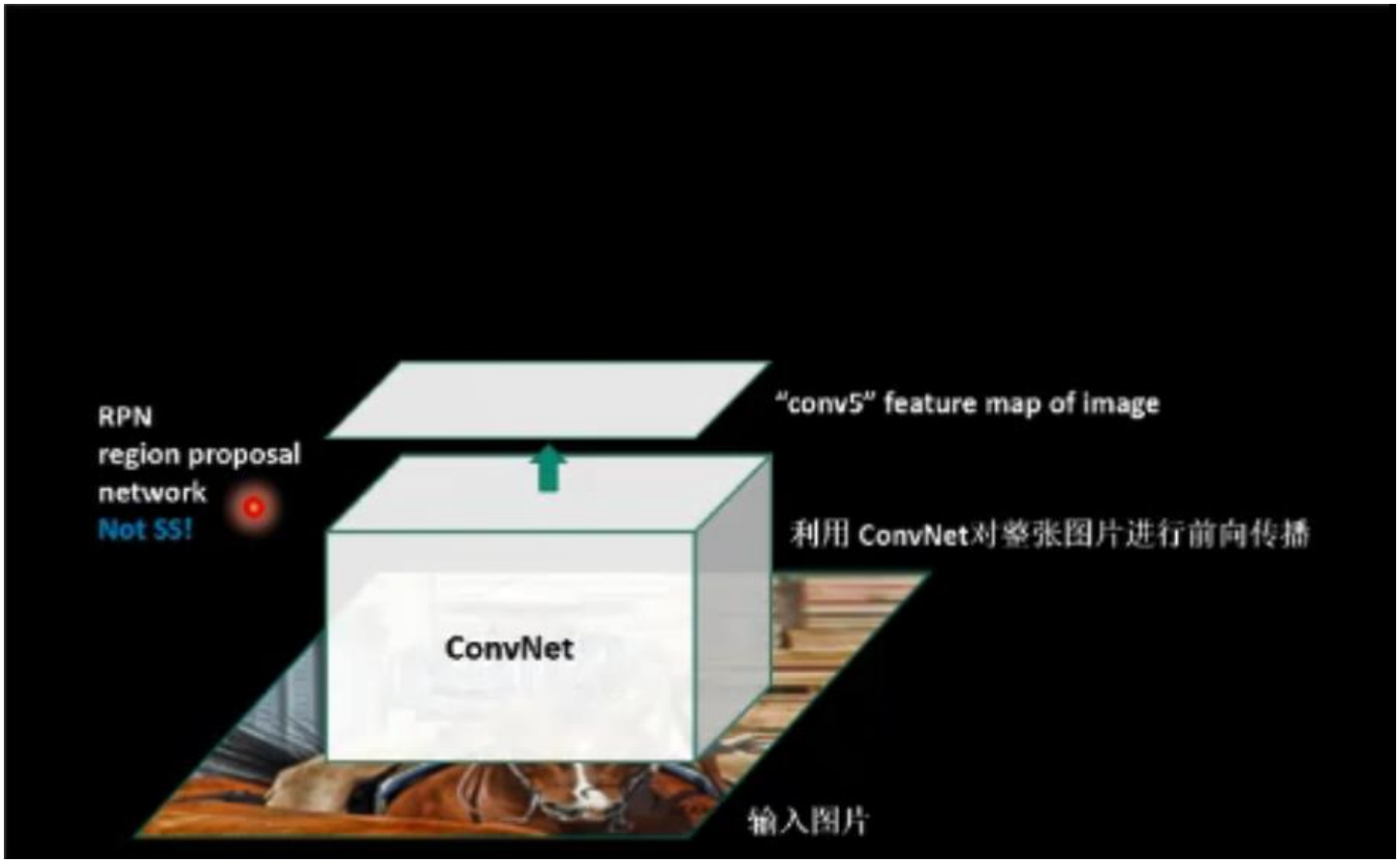
Abstract—State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet [1] and Fast R-CNN [2] have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. In this work, we introduce a *Region Proposal Network* (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. We further merge RPN and Fast R-CNN into a single network by sharing their convolutional features—using the recently popular terminology of neural networks with "attention" mechanisms, the RPN component tells the unified network where to look. For the very deep VGG-16 model [3], our detection system has a frame rate of 5fps (*including all steps*) on a GPU, while achieving state-of-the-art object detection accuracy on PASCAL VOC 2007, 2012, and MS COCO datasets with only 300 proposals per image. In ILSVRC and COCO 2015 competitions, Faster R-CNN and RPN are the foundations of the 1st-place winning entries in several tracks. Code has been made publicly available.

Index Terms—Object Detection, Region Proposal, Convolutional Neural Network.

Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," NIPS 2015, PAMI2016

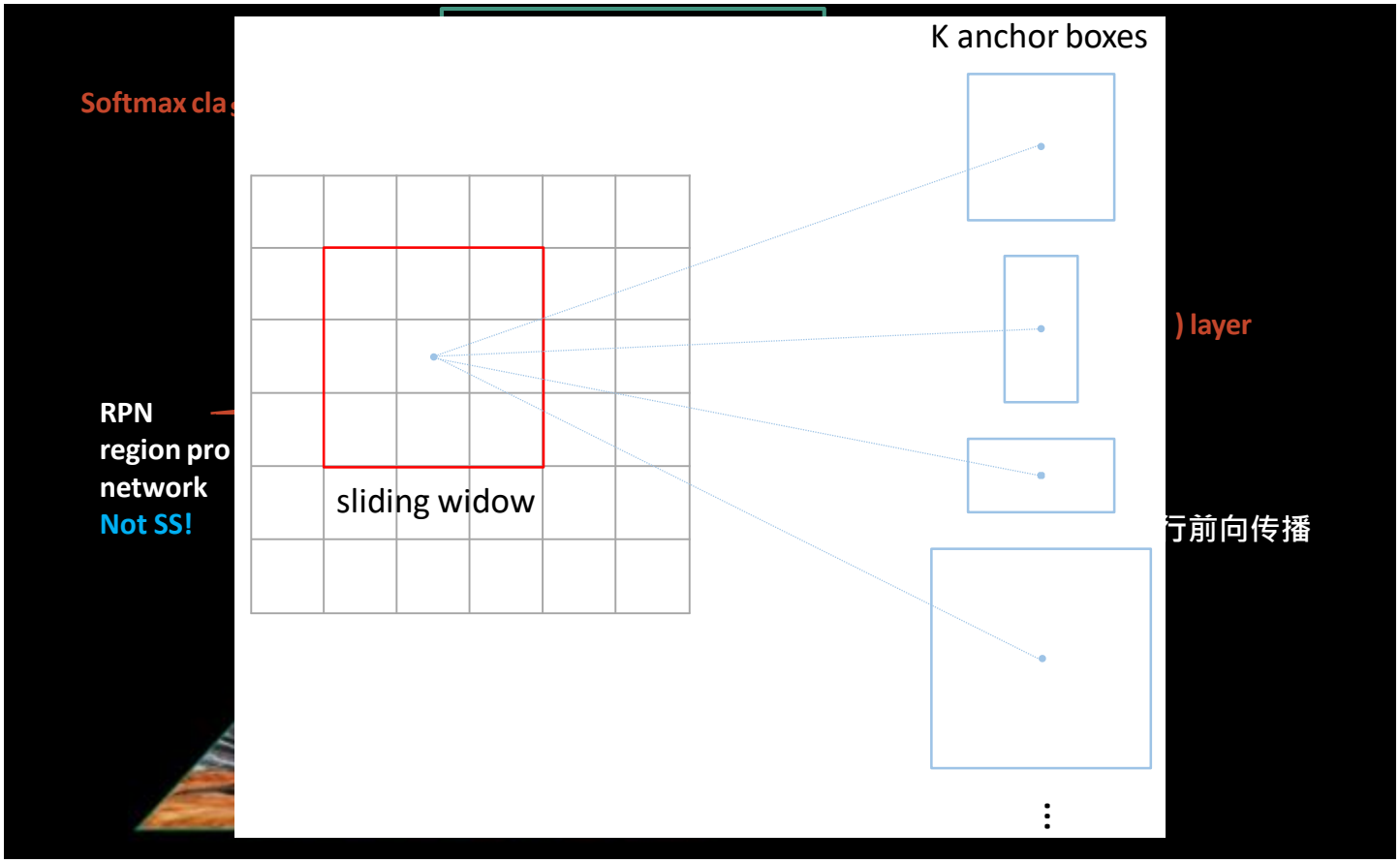
https://arxiv.org/pdf/1506.01497.pdf

两阶段目标检测方法：Faster R-CNN



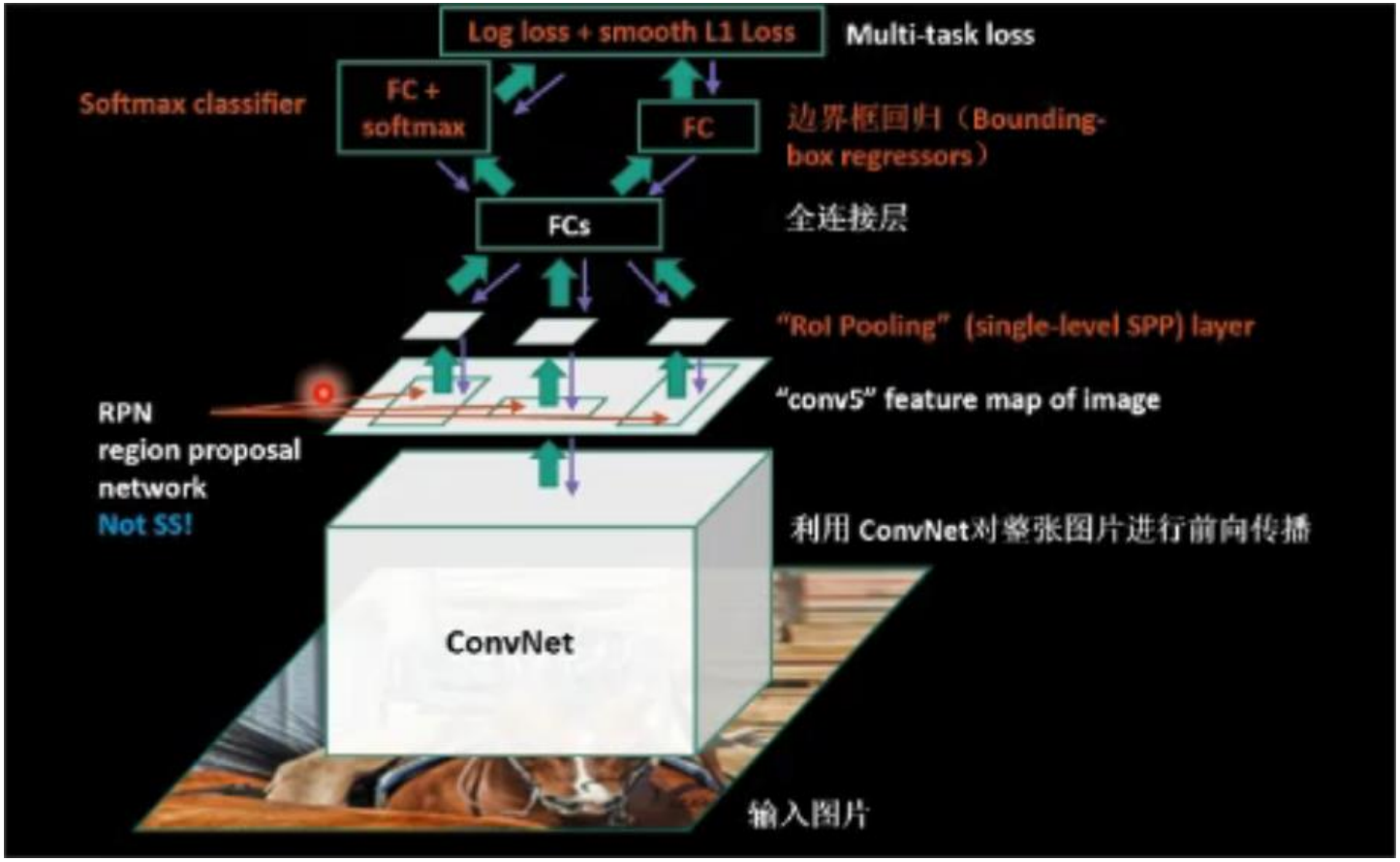
Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," NIPS 2015

两阶段目标检测方法：Faster R-CNN



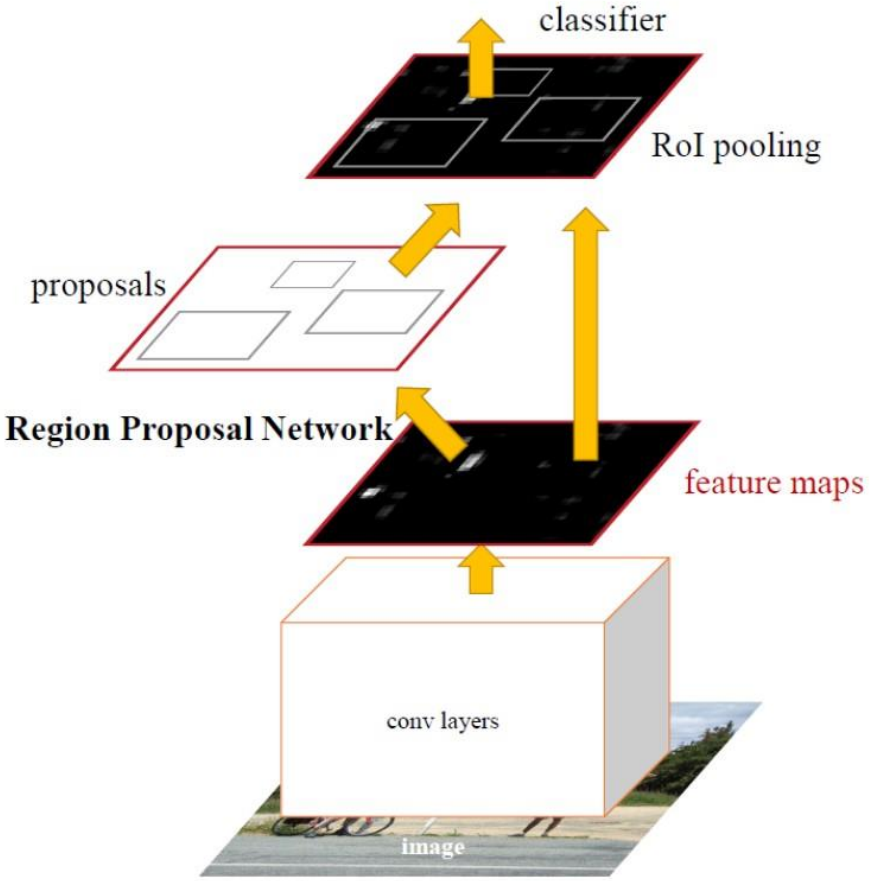
Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” NIPS 2015

两阶段目标检测方法：Faster R-CNN



Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," NIPS 2015

Faster R-CNN：框架再览



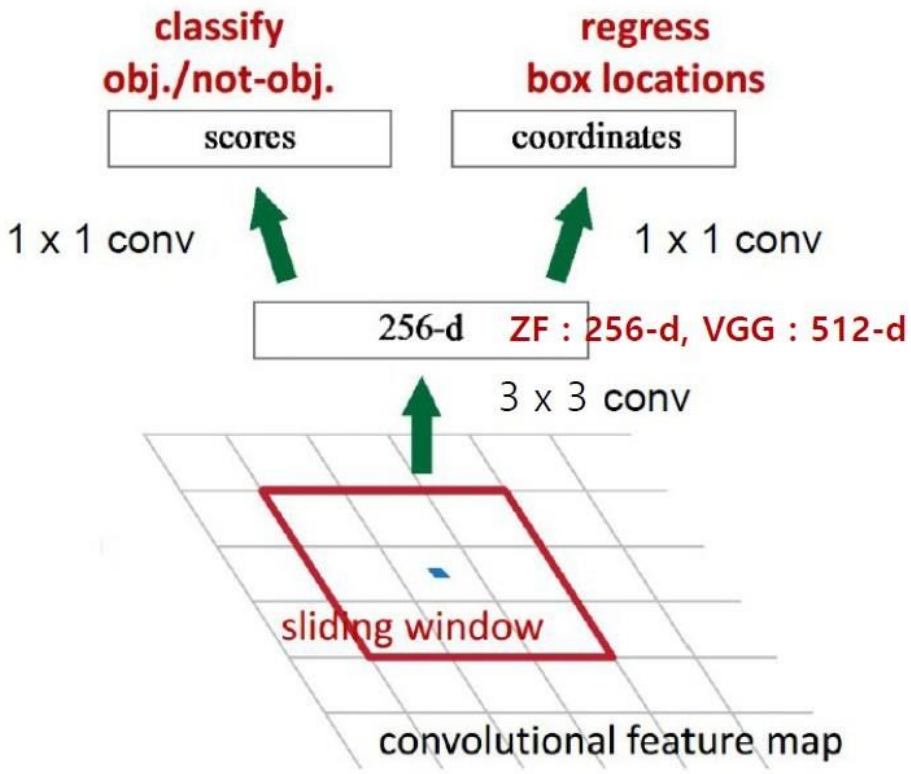
1.在最后一个卷积层后面插入一个Region Proposal Network（RPN）；

2.RPN用于直接产生区域候选，不需要额外算法产生区域候选；

3.在RPN之后，像Fast RCNN中一样，使用ROI Pooling和后续的分类器、回归器

Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” NIPS 2015

Faster R-CNN：RPN



1. 在特征图上使用一个小的滑动窗口；

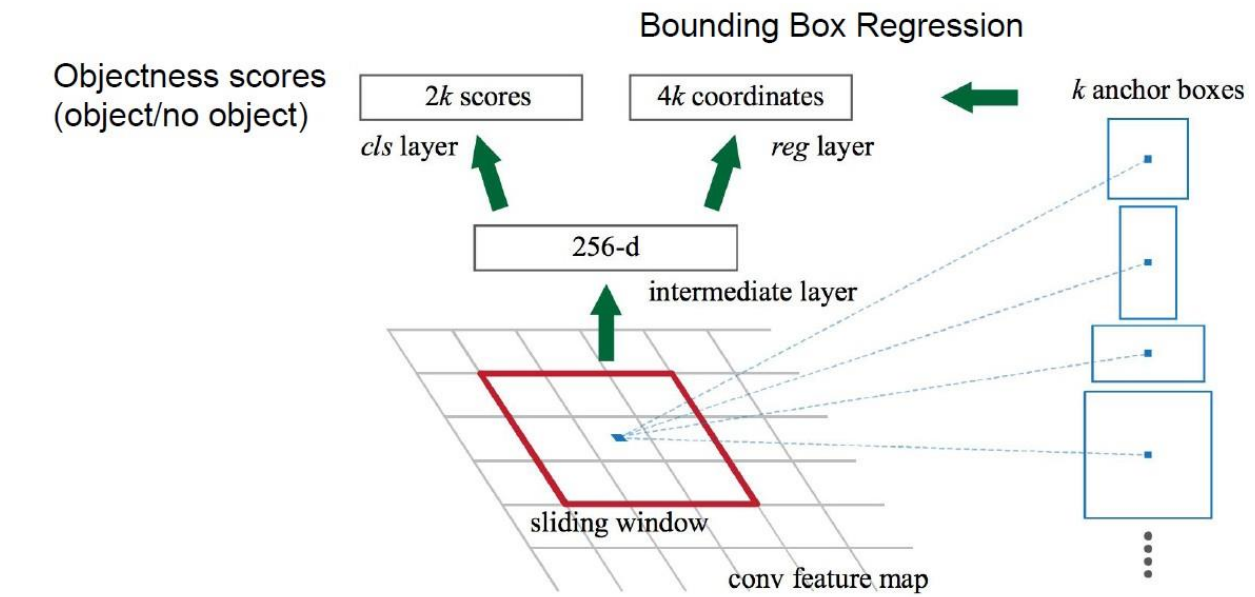
2. 构建一个小的网络以（1）分类目标/非目标；（2）回归边界框；

3. 滑动窗口的位置提供了在原图像上定位的参考信息；

4.边界框回归根据滑动窗口的参考信息提供了更精准的定位信息。

Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” NIPS 2015

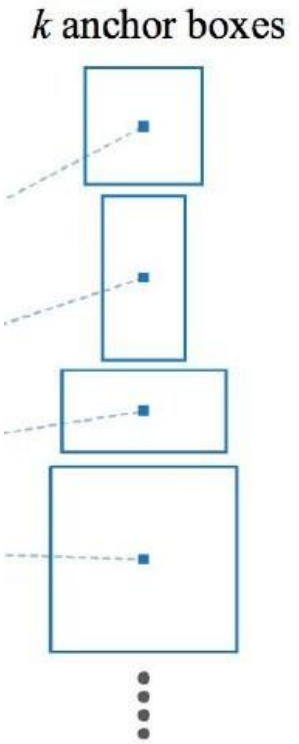
Faster R-CNN：RPN



In practice, k = 9 (3 different scales and 3 aspect ratios)

1. 使用k个锚点框（anchor boxes），这些锚点框是平移不变的（在每个位置处都相同）；
2. 回归器为每个锚点框计算偏移；
3. 分类器评测每个锚点框（回归后的）是物体的可能性；

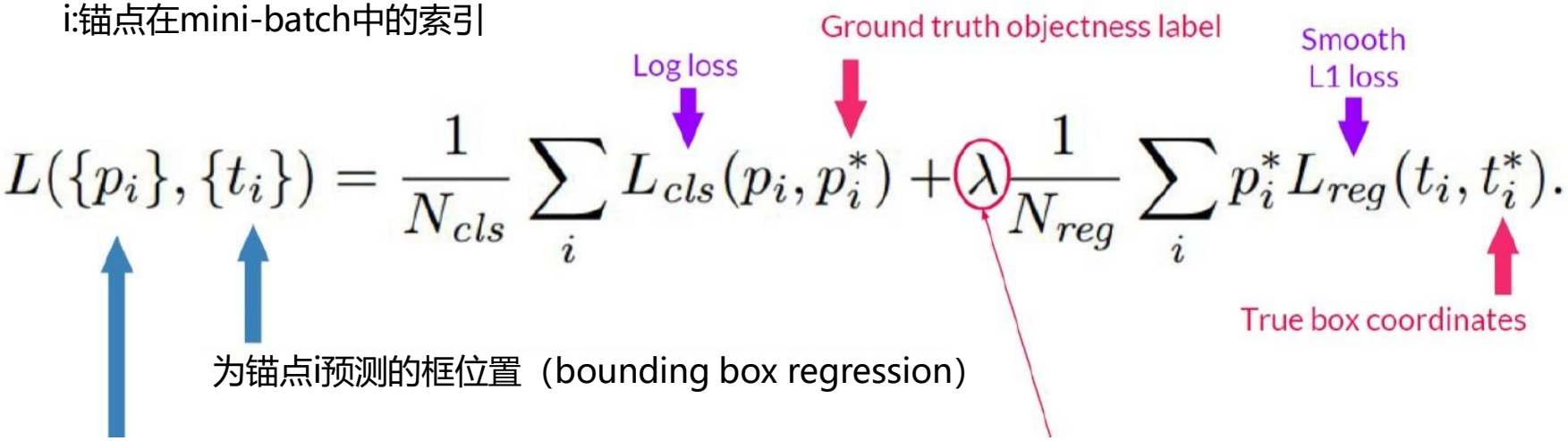
Faster R-CNN：Anchor Boxes



- 3种尺度（128*128,256*256，512*512），三种纵横比（2:1,1:2,1:1），产生9个anchors；
- 一个锚点框被标记为正样本的两种条件:
  - 这个锚点框是与ground-truth box具有最高重叠 率（IOU）的；
  - 这个锚点框与ground-truth box重叠率大于0.7
- 如果一个锚点框与所有的ground-truth boxes的重叠率不大 于0.3，那么这个锚点框被标记为负样本；
- 训练中，在一个batch里，正负锚点框的比例控制在50%

Faster R-CNN：RPN Loss Function

i:锚点在mini-batch中的索引



为锚点i预测的框位置（bounding box regression）

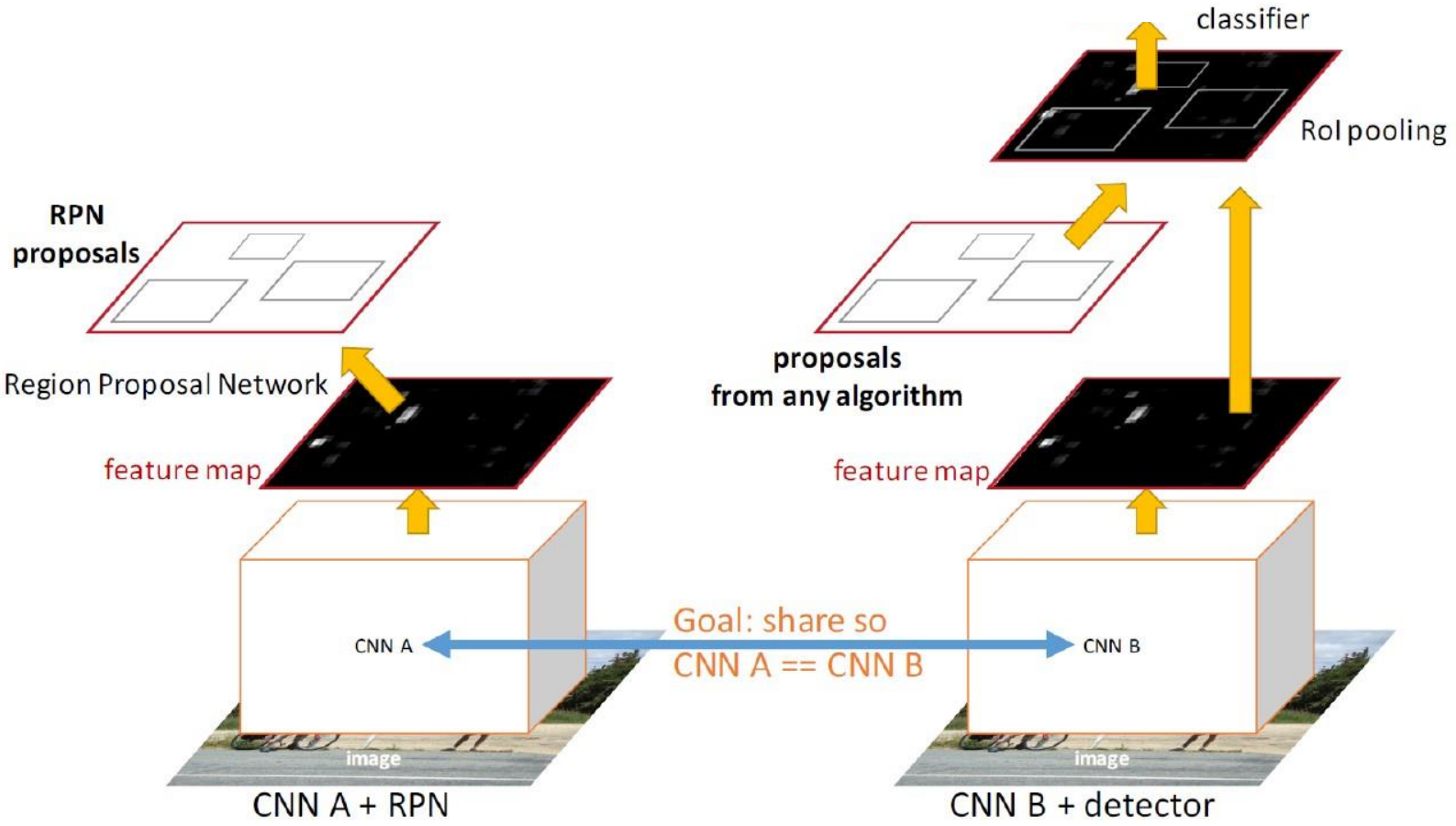
锚点i是物体的概率

训练过程中，λ设置为10，用以平衡分类和回归两个loss

N_{cls}：一个批次（mini-batch）中锚点数量（anchor），约为256个

N_{reg}：一个批次（mini-batch）中锚点（anchor）位置数量，约为2400个

Faster R-CNN：训练细节-共享特征



Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” NIPS 2015

Faster R-CNN：4步交替训练

```
# Let M0 be an ImageNet pre-trained network
1.  train_rpn(M0) → M1                # Train an RPN initialized from M0, get M1
2.  generate_proposals(M1) → P1       # Generate training proposals P1 using RPN M1
3.  train_fast_rcnn(M0, P1) → M2      # Train Fast R-CNN M2 on P1 initialized from M0
4.  train_rpn_frozen_conv(M2) → M3    # Train RPN M3 from M2 without changing conv layers
5.  generate_proposals(M3) → P2
6.  train_fast_rcnn_frozen_conv(M3, P2) → M4   # Conv layers are shared with RPN M3
7.  return add_rpn_layers(M4, M3.RPN)          # Add M3's RPN layers to Fast R-CNN M4
```

Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," NIPS 2015

Faster R-CNN：总结

■ 实验结果

速度提高

	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image (with proposals)	50 seconds	2 seconds	**0.2 seconds**
(Speedup)	1x	25x	**250x**
mAP (VOC 2007)	66.0	**66.9**	**66.9**

■ 突破

- 采用RPN来生成proposal，检测效果得以提高
- RPN与检测网络共享卷积部分，节省计算

但没能实现实时检测

到目前为止，我们似乎得到了一个很完善的检测框架了。

还有什么改进方式?

Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” NIPS 2015

两阶段目标检测方法：FPN

Feature Pyramid Networks for Object Detection

Tsung-Yi Lin[1,2], Piotr Dollár[1], Ross Girshick[1],
Kaiming He[1], Bharath Hariharan[1], and Serge Belongie[2]

[1]Facebook AI Research (FAIR)
[2]Cornell University and Cornell Tech

Abstract

Feature pyramids are a basic component in recognition systems for detecting objects at different scales. But recent deep learning object detectors have avoided pyramid representations, in part because they are compute and memory intensive. In this paper, we exploit the inherent multi-scale, pyramidal hierarchy of deep convolutional networks to construct feature pyramids with marginal extra cost. A top-down architecture with lateral connections is developed for building high-level semantic feature maps at all scales. This architecture, called a Feature Pyramid Network (FPN), shows significant improvement as a generic feature extractor in several applications. Using FPN in a basic Faster R-CNN system, our method achieves state-of-the-art single-model results on the COCO detection benchmark without bells and whistles, surpassing all existing single-model entries including those from the COCO 2016 challenge winners. In addition, our method can run at 6 FPS on a GPU and thus is a practical and accurate solution to multi-scale object detection. Code will be made publicly available.

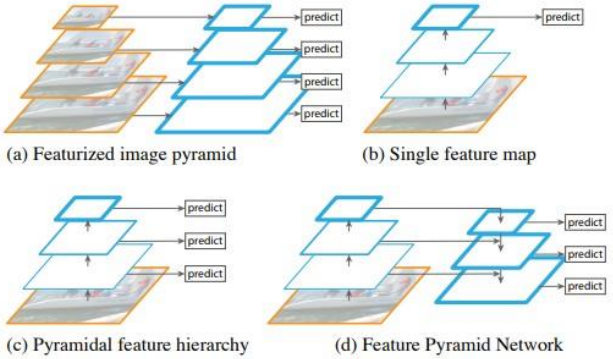


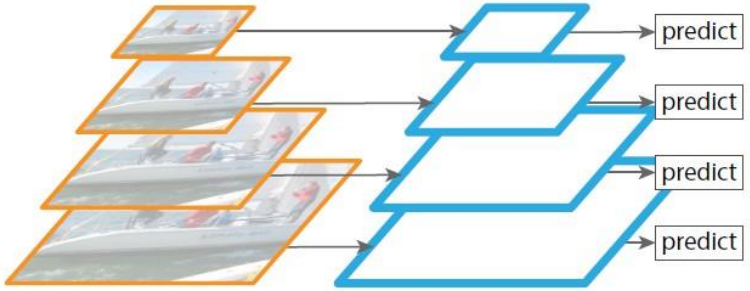
Figure 1. (a) Using an image pyramid to build a feature pyramid. Features are computed on each of the image scales independently, which is slow. (b) Recent detection systems have opted to use only single scale features for faster detection. (c) An alternative is to reuse the pyramidal feature hierarchy computed by a ConvNet as if it were a featurized image pyramid. (d) Our proposed Feature Pyramid Network (FPN) is fast like (b) and (c), but more accurate. In this figure, feature maps are indicate by blue outlines and thicker outlines denote semantically stronger features.

Lin, Tsung-Yi, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. "Feature Pyramid Networks for Object Detection." CVPR 2017.

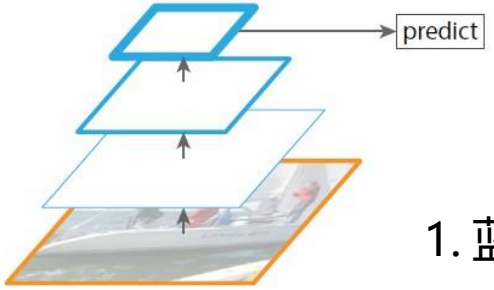
https://arxiv.org/abs/1612.03144

两阶段目标检测方法：FPN

- 从特征层面出发



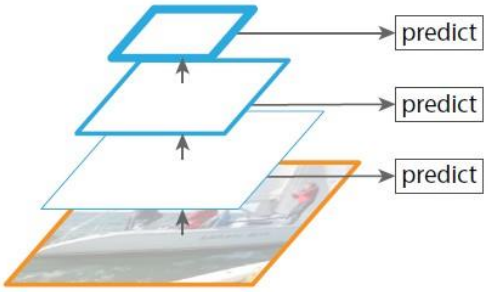
(a) Featurized image pyramid



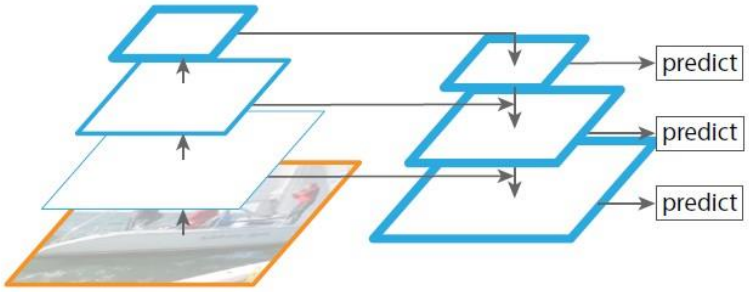
(b) Single feature map

1. 蓝色越深代表语义信息越强；

2. 框的大小代表分辨率



(c) Pyramidal feature hierarchy

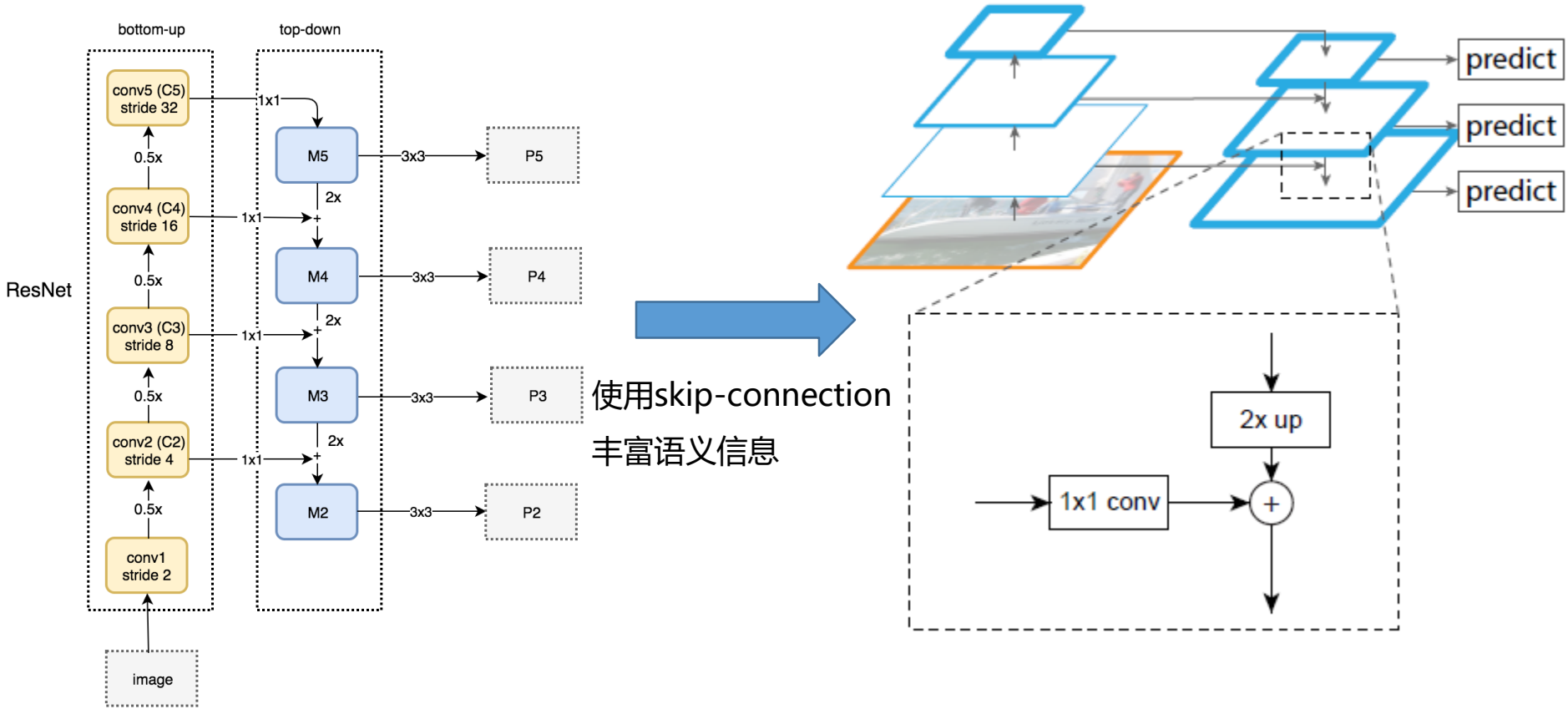


(d) Feature Pyramid Network

Lin, Tsung-Yi, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. "Feature Pyramid Networks for Object Detection." CVPR 2017.

FPN：模块

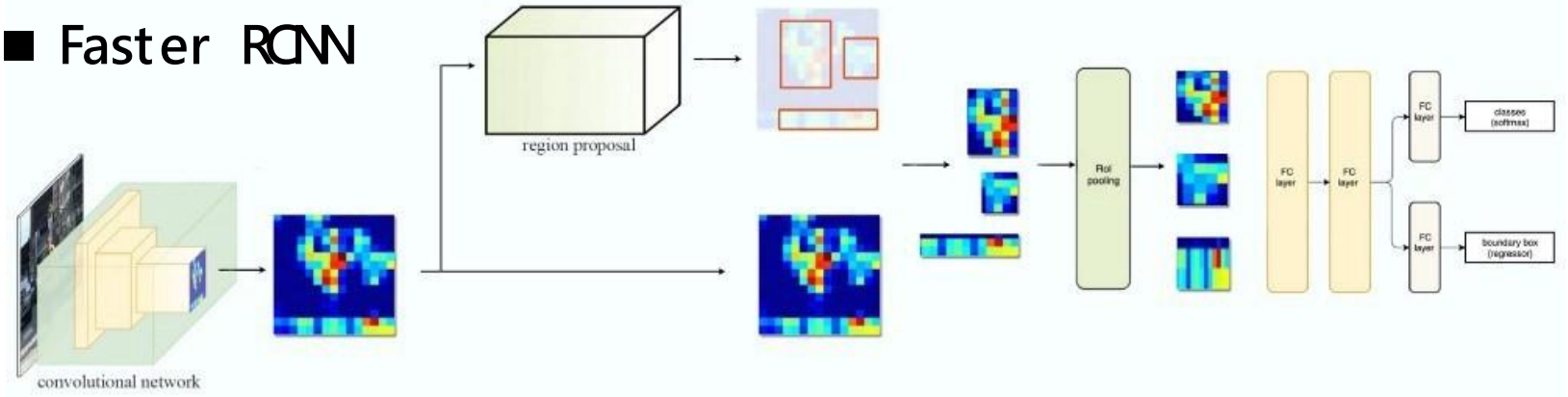
■ 回忆



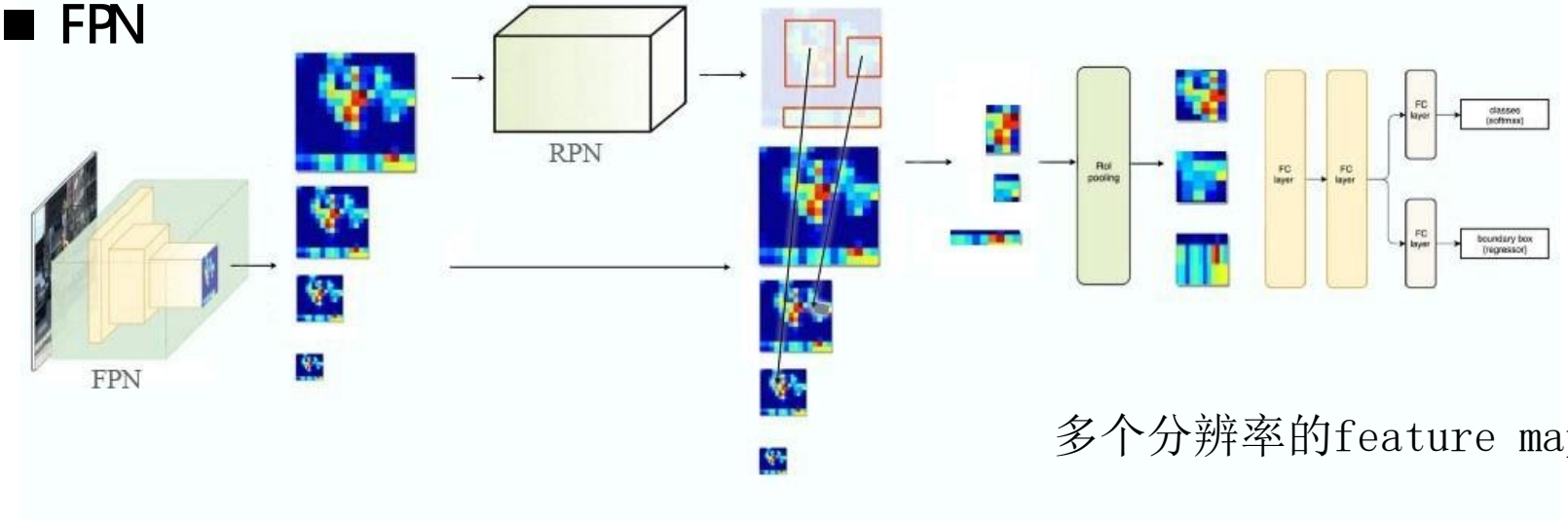
Lin, Tsung-Yi, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. "Feature Pyramid Networks for Object Detection." CVPR 2017.

FPN：嵌入至Fast RCNN & Faster RCNN

■ Faster RCNN



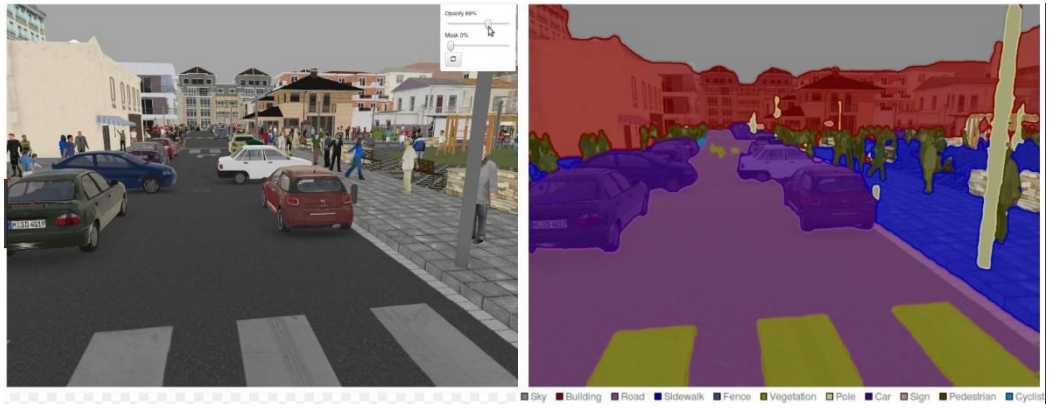
■ FPN



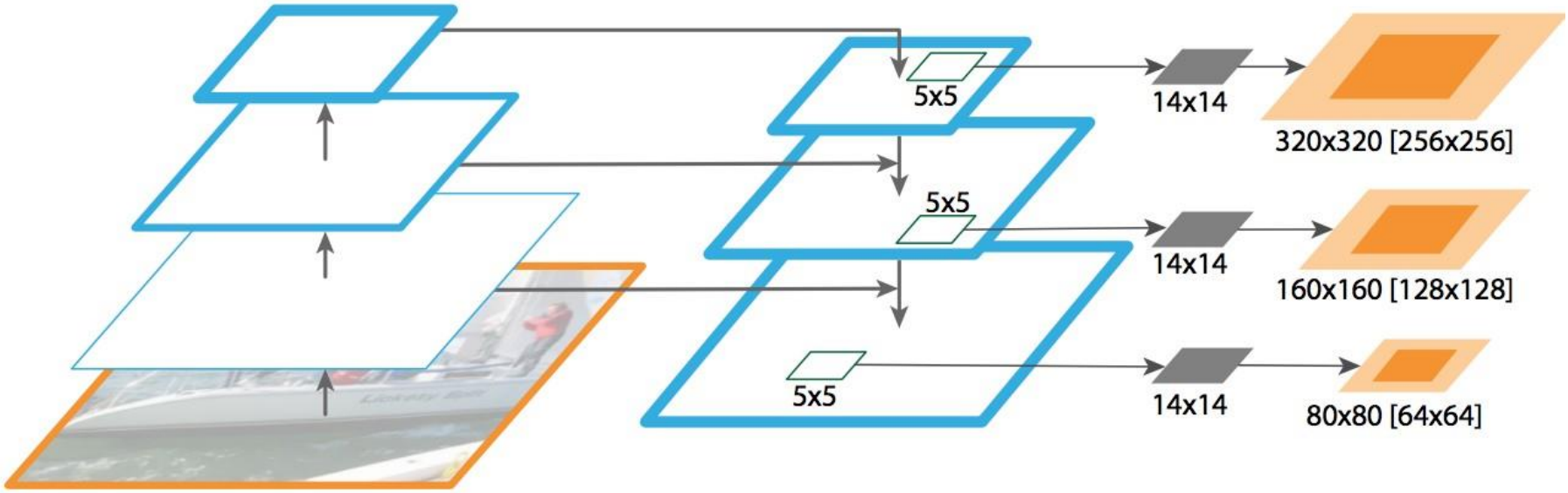
多个分辨率的feature map

Lin, Tsung-Yi, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. "Feature Pyramid Networks for Object Detection." CVPR 2017.

FPN：扩展至分割任务



图像分割



Lin, Tsung-Yi, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. "Feature Pyramid Networks for Object Detection." CVPR 2017.

FPN：结果

实验结果

Fast R-CNN	proposals	feature	head	lateral?	top-dow			$_s$	AP_m	AP_l
(a) baseline on conv4	RPN, $\{P_k\}$	C_4	conv5			54.7	31.9	15.7	36.5	45.5
(b) baseline on conv5	RPN, $\{P_k\}$	C_5	2*fc*			52.9	28.8	11.9	32.4	43.4
(c) FPN	RPN, $\{P_k\}$	$\{P_k\}$	2*fc*	✓	✓	**56.9**	**33.9**	**17.8**	**37.7**	**45.8**
Ablation experiments follow:										
(d) bottom-up pyramid	RPN, $\{P_k\}$	$\{P_k\}$	2*fc*	✓		44.9	24.9	10.9	24.4	38.5
(e) top-down pyramid, w/o lateral	RPN, $\{P_k\}$	$\{P_k\}$	2*fc*		✓	54.0	31.3	13.3	35.2	45.3
(f) only finest level	RPN, $\{P_k\}$	P_2	2*fc*	✓	✓	56.3	33.4	17.3	37.3	45.6

Table 2. Object detection results using **Fast R-CNN** [11] on *a fixed set of proposals* (RPN, $\{P_k\}$, Table 1(c)), evaluated on the COCO `minival` set. Models are trained on the `trainval35k` set. All results are based on ResNet-50 and share the same hyper-parameters.

Faster R-CNN	proposals	feature	head	lateral?	top-down?	AP@0.5	AP	AP_s	AP_m	AP_l
(*) baseline from He *et al.* [16]†	RPN, C_4	C_4	conv5			47.3	26.3	-	-	-
(a) baseline on conv4	RPN, C_4	C_4	conv5			53.1	31.6	13.2	35.6	**47.1**
(b) baseline on conv5	RPN, C_5	C_5	2*fc*			51.7	28.0	9.6	31.9	43.1
(c) FPN	RPN, $\{P_k\}$	$\{P_k\}$	2*fc*	✓	✓	**56.9**	**33.9**	**17.8**	**37.7**	45.8

Table 3. Object detection results using **Faster R-CNN** [29] evaluated on the COCO `minival` set. *The backbone network for RPN are consistent with Fast R-CNN.* Models are trained on the `trainval35k` set and use ResNet-50. †Provided by authors of [16].

突破

- 高层的low resolution，strong semantic info特征与浅层的high resolution，weak semantic info自然地结合
- 不引入过多的额外计算，只需要用single scale的原始输入
- 通用的框架，也可以在One-stage检测方法中使用

Lin, Tsung-Yi, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. "Feature Pyramid Networks for Object Detection." CVPR 2017.

两阶段目标检测方法： RFCN

R-FCN: Object Detection via Region-based Fully Convolutional Networks

Jifeng Dai
Microsoft Research

Yi Li*
Tsinghua University

Kaiming He
Microsoft Research

Jian Sun
Microsoft Research

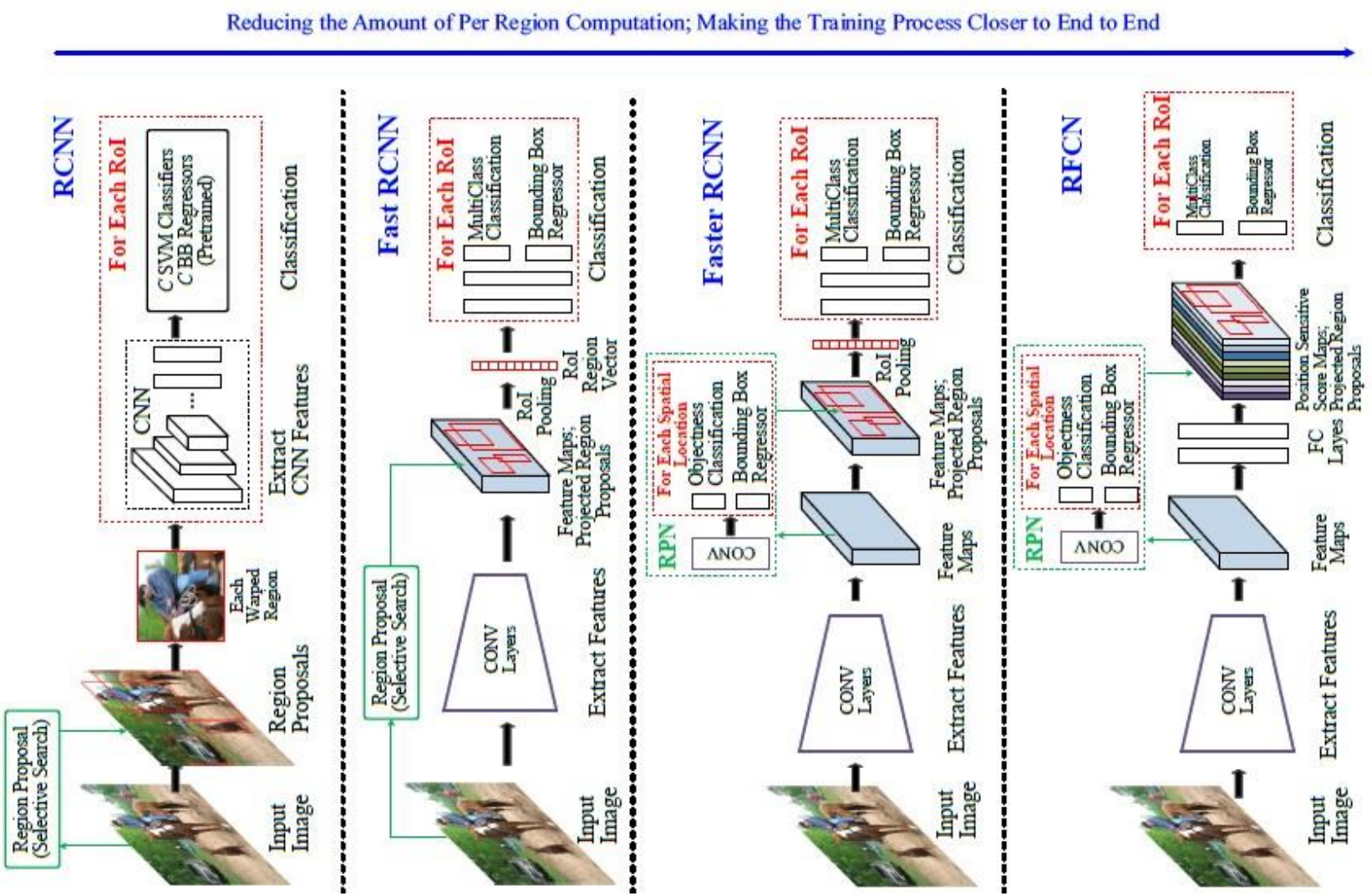
Abstract

We present region-based, fully convolutional networks for accurate and efficient object detection. In contrast to previous region-based detectors such as Fast/Faster R-CNN [7, 19] that apply a costly per-region subnetwork hundreds of times, our region-based detector is fully convolutional with almost all computation shared on the entire image. To achieve this goal, we propose position-sensitive score maps to address a dilemma between translation-invariance in image classification and translation-variance in object detection. Our method can thus naturally adopt fully convolutional image classifier backbones, such as the latest Residual Networks (ResNets) [10], for object detection. We show competitive results on the PASCAL VOC datasets (*e.g.*, 83.6% mAP on the 2007 set) with the 101-layer ResNet. Meanwhile, our result is achieved at a test-time speed of 170ms per image, 2.5-20× faster than the Faster R-CNN counterpart. Code is made publicly available at: `https://github.com/daijifeng001/r-fcn`.

Dai, Jifeng, Yi Li, Kaiming He, and Jian Sun. "R-fcn: Object detection via region-based fully convolutional networks." NIPS 2016.

http://arxiv.org/abs/1605.06409

两阶段方法对比



Liu, Li, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. "Deep learning for generic object detection: A survey." arXiv preprint arXiv:1809.02165 (2018).

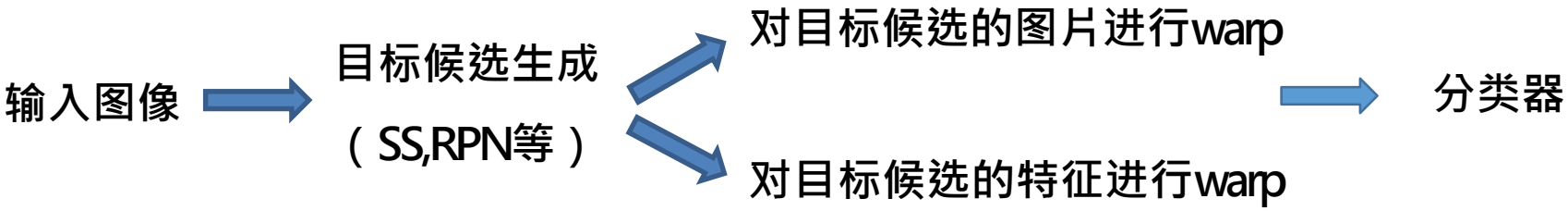
两阶段方法，END

两阶段方法，END

但是……

速度（实时性）？

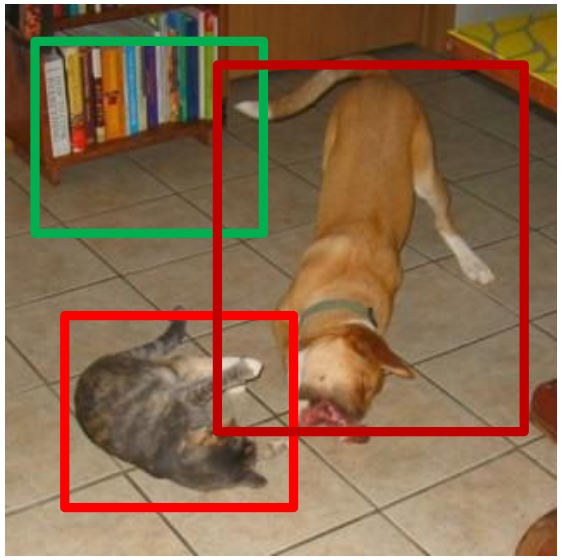
两阶段目标检测算法



计算复杂度高，无法达到实时性

Faster RCNN: 7FPS

单阶段目标检测方法



cat, dog,
background

YOLO

You Only Look Once: Unified, Real-Time Object Detection

Joseph Redmon*, Santosh Divvala*†, Ross Girshick¶, Ali Farhadi*†

University of Washington*, Allen Institute for AI†, Facebook AI Research¶

http://pjreddie.com/yolo/

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," CVPR 2016

YOLO：坊间趣事

Meta-Review

This paper proposes a new object detection algorithm with Deep CNNs.

Summary review -- [this is where you explain the final decision to the authors]

The paper got mixed reviews. Some reviewers liked the paper, some voted for rejection, which made this manuscript a borderline paper.

Since two of the long reviews raised some important concerns about the paper, and this year we got many very good submissions, therefore I cannot recommend this paper for publication at NIPS.

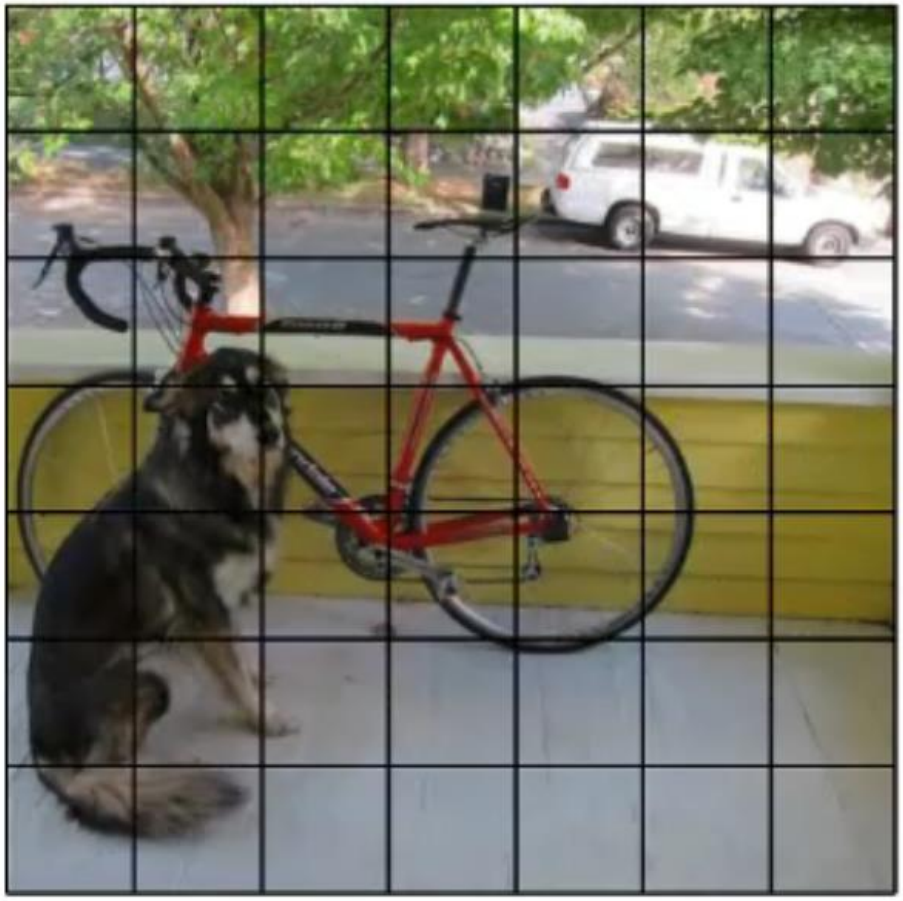
The reviewers, however, agreed that this is an interesting research direction, and I do encourage the authors to continue this work. The reviewers' suggestions might help improve the revised version of this paper.

--Rejected by NIPS2015

Finally this paper has been accepted by CVPR 2016

source code: http://pjreddie.com/darknet/yolo/

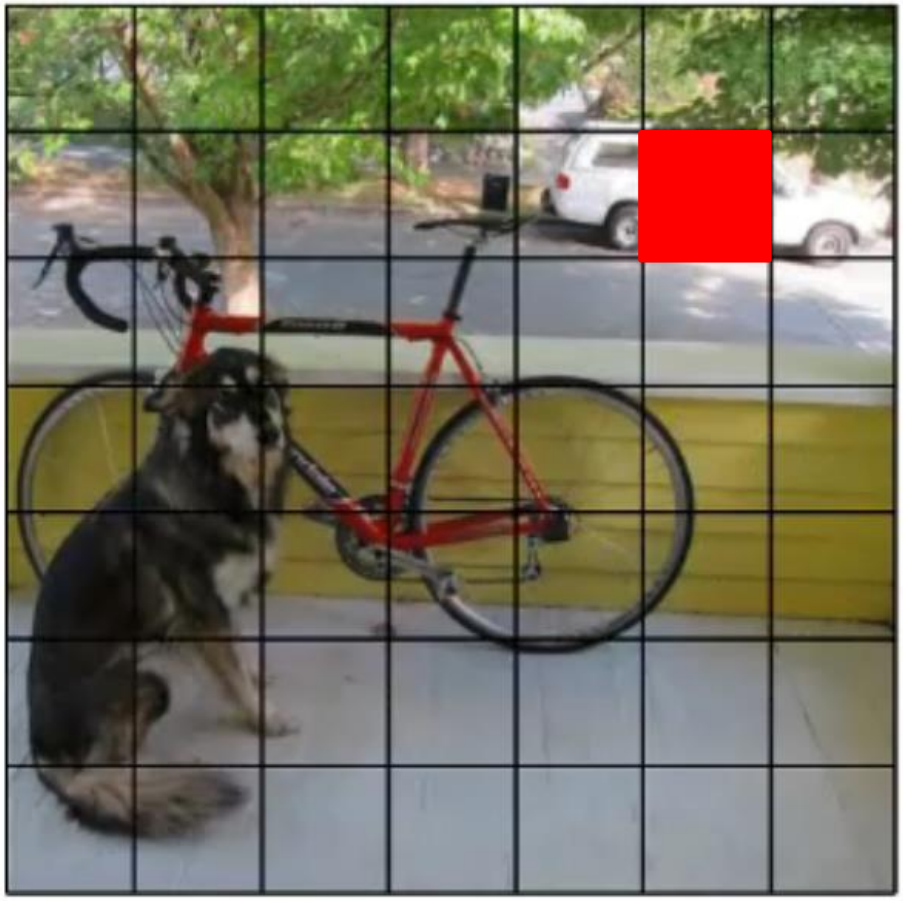
单阶段目标检测方法：YOLO



输入图像划分成7*7的格子，每个格子预测目标框，并给出这个框和目标的重叠率

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” CVPR 2016

单阶段目标检测方法：YOLO

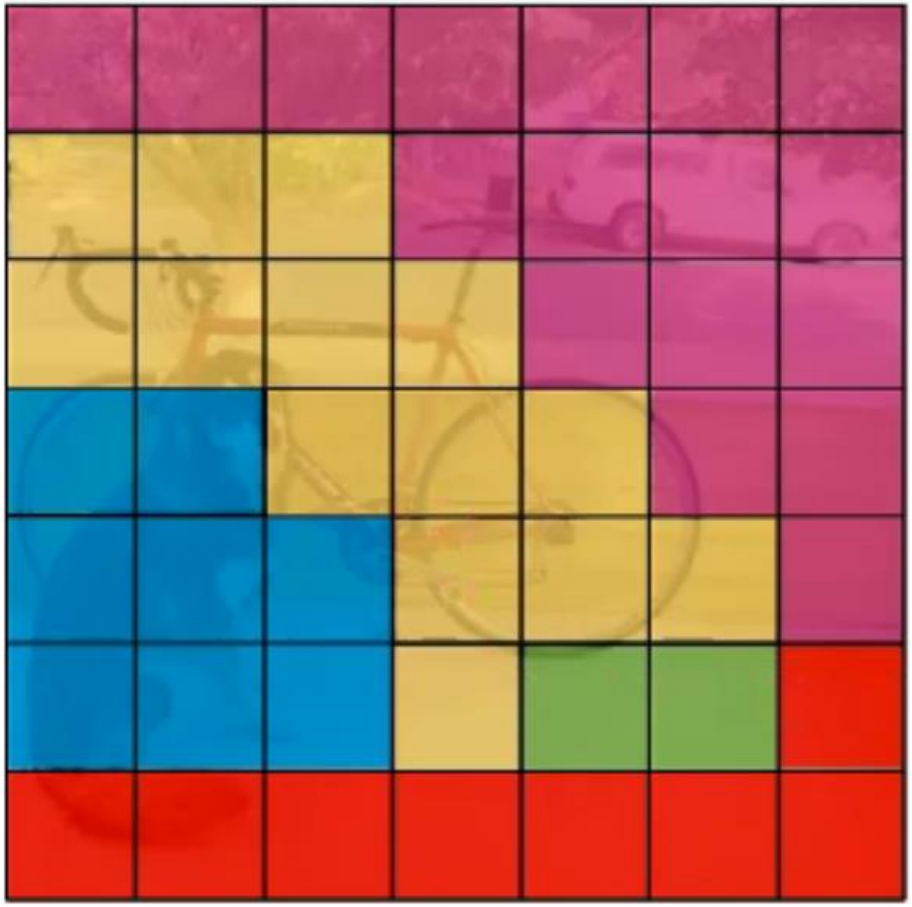


输入图像划分成7*7的格子，每个格子预测目标框，并给出这个框和目标的重叠率

$$(x, y, w, h, IOU)$$

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” CVPR 2016

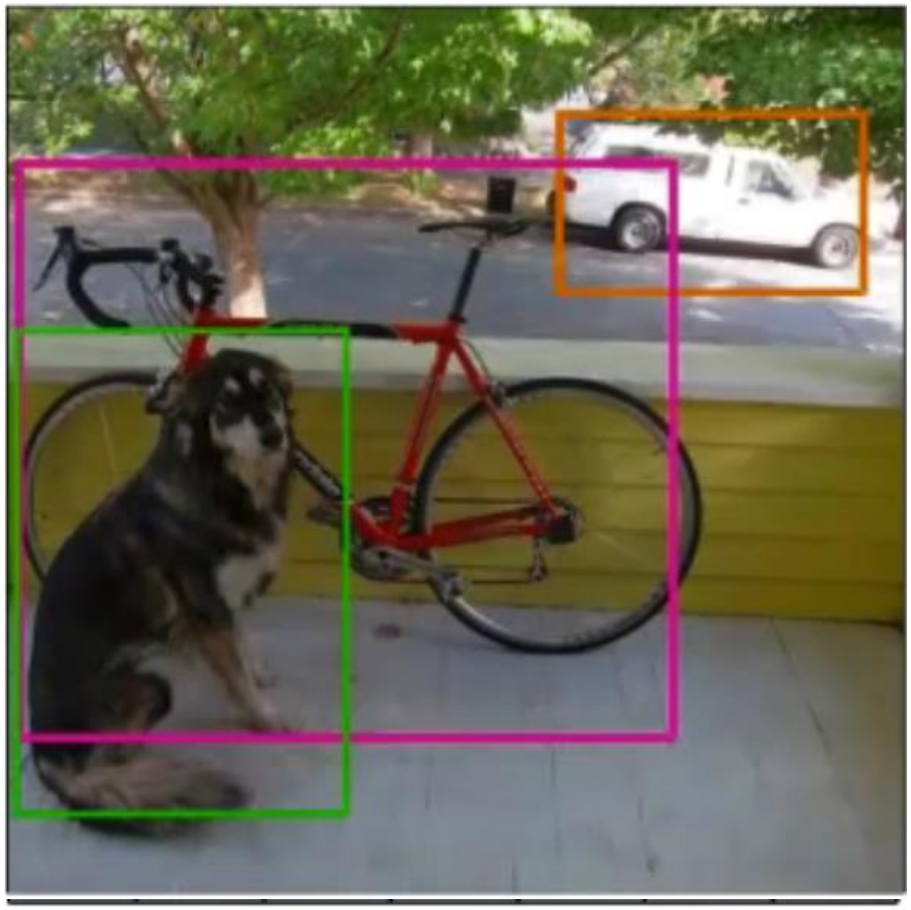
单阶段目标检测方法：YOLO



每个格子预测20个条件概率（20类物体）

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” CVPR 2016

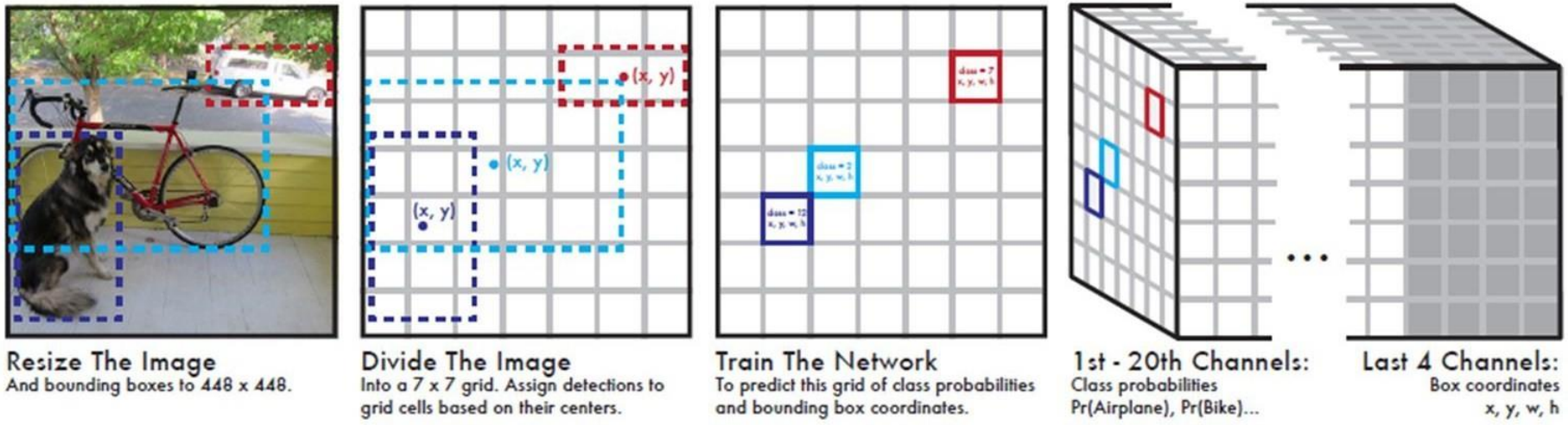
单阶段目标检测方法：YOLO



最后，将预测框和预测类结合

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” CVPR 2016

YOLO：框架再览

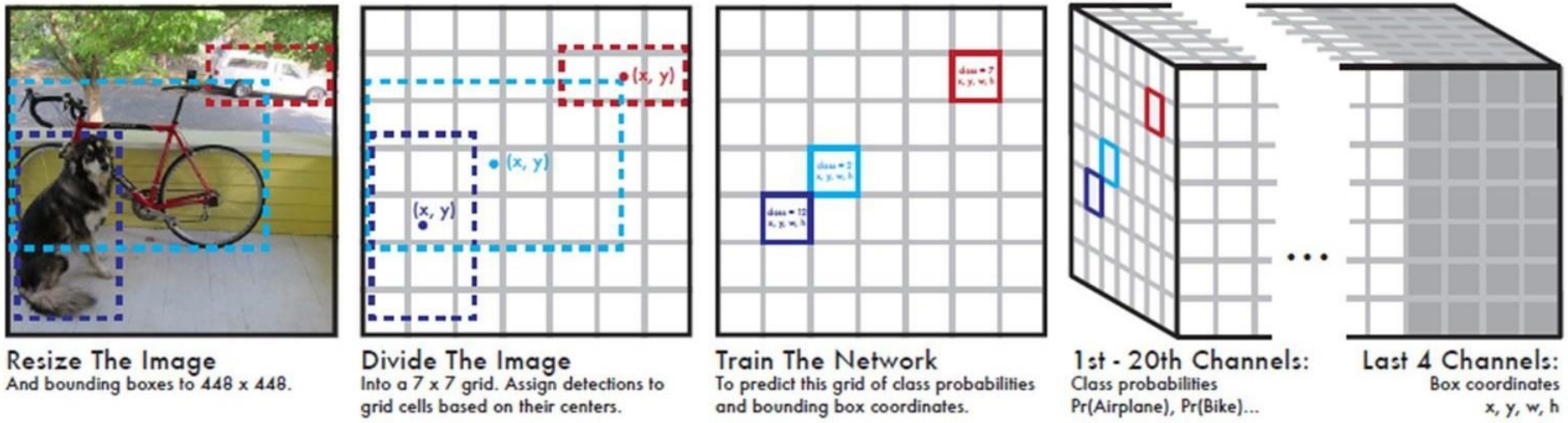


It divides the image into an even grid and simultaneously predicts bounding boxes, confidence in those boxes, and class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.(7x7x30 in this paper)

1 把输入图像划分为S*S个网格（S=7）。位于目标中心的网格负责检测该目标

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” CVPR 2016

YOLO：框架再览



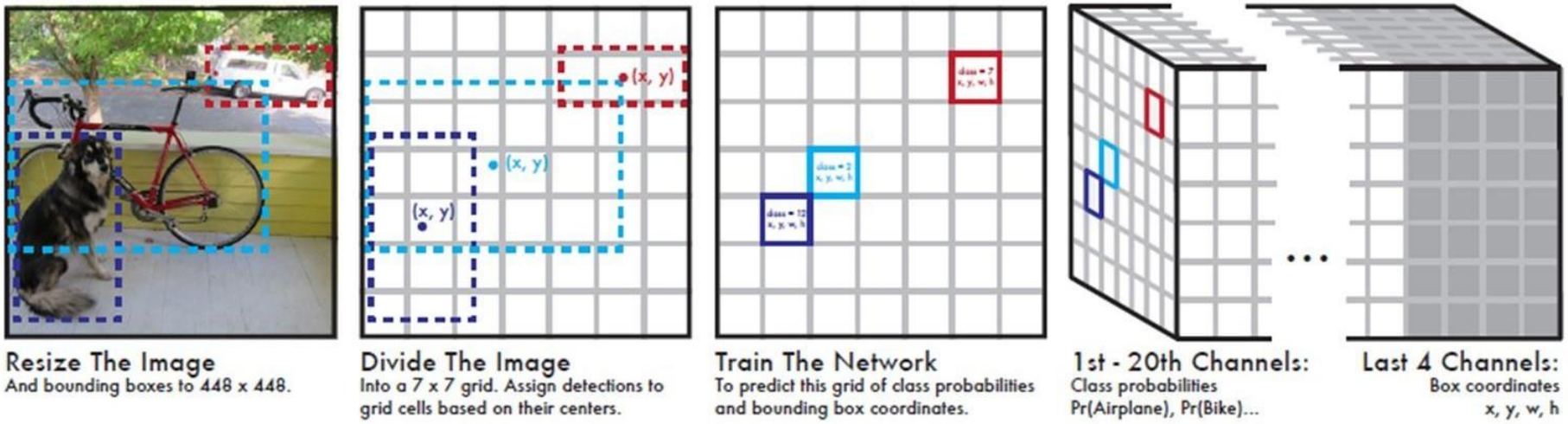
It divides the image into an even grid and simultaneously predicts bounding boxes, confidence in those boxes, and class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.(7x7x30 in this paper)

1.每个网格预测B个边界框和这个边界框是物体的概率（Objectness）；具体的，每个边界框会预测出5个值：x,y,w,h和置信度Pr(Object)*IOU(truth&pred)

2. 每个网格预测C个条件概率Pr（$Class_i$|Object）

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” CVPR 2016

YOLO：框架再览



It divides the image into an even grid and simultaneously predicts bounding boxes, confidence in those boxes, and class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.(7x7x30 in this paper)

3 在测试阶段，预测每个检测框的分数：

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}} \quad (1)$$

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” CVPR 2016

YOLO：网络结构

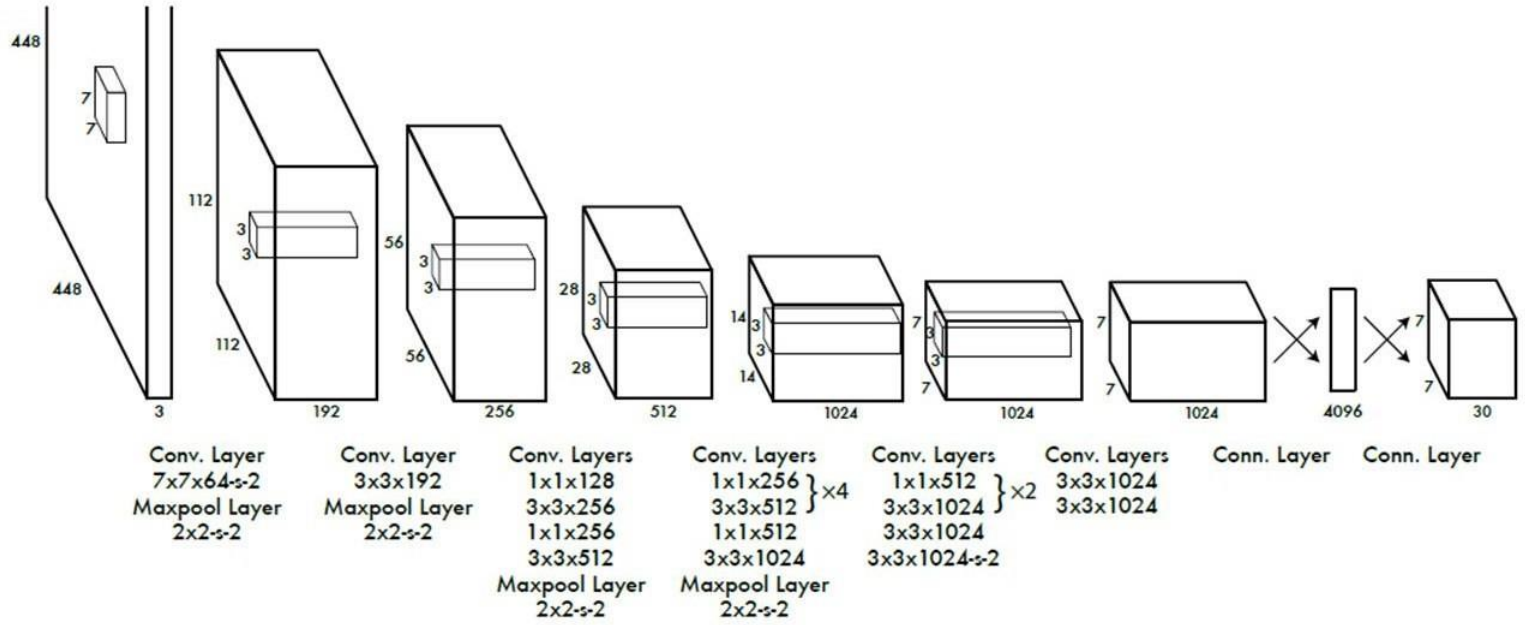


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224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

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” CVPR 2016

YOLO：损失函数

$$
\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(x_i - \hat{x}_i\right)^2 + \left(y_i - \hat{y}_i\right)^2 \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i\right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i\right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2 \qquad (3)
\end{aligned}
$$

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” CVPR 2016

YOLO：损失函数

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

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

回归检测框

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

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

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2 \quad (3)$$

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” CVPR 2016

YOLO：损失函数

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

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

回归检测框

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

预测置信度（物体性）

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

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2 \quad (3)$$

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” CVPR 2016

YOLO：损失函数

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

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

回归检测框

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

预测置信度（物体性）

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

预测是背景的分数

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2 \quad (3)$$

预测物体类别

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” CVPR 2016

YOLO：损失函数

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

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

回归检测框

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

预测置信度（物体性）

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

预测是背景的分数

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2 \quad (3)$$

预测物体类别

$\lambda_{\text{coord}} = 5$ and $\lambda_{\text{noobj}} = .5$.

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” CVPR 2016

YOLO：损失函数

表示由第i个网格中第j个边界框预测器负责预测

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

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

回归检测框

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

预测置信度（物体性）

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

预测是背景的分数

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2 \quad (3)$$

预测物体类别

$\lambda_{\text{coord}} = 5$ and $\lambda_{\text{noobj}} = .5$.

表示物体是否出现在网格i中

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” CVPR 2016

YOLO：总结

■ 实验结果

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [38]	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[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

■ 特点

- 速度特别快
- 泛化能力强

- 后续：YOLOv2,v3,YOLO9000

但小物体、重叠物体无法检测

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” CVPR 2016

单阶段目标检测方法：SSD

SSD: Single Shot MultiBox Detector

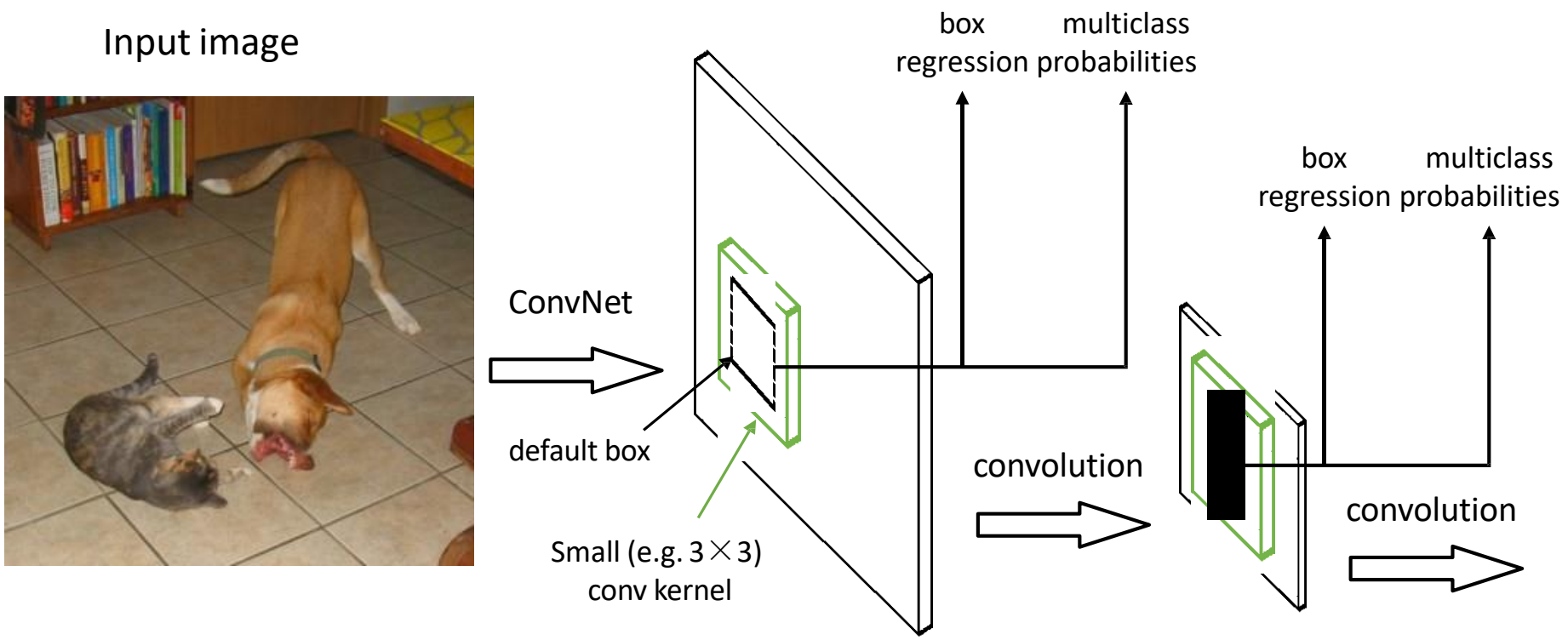
Wei Liu[1], Dragomir Anguelov[2], Dumitru Erhan[3], Christian Szegedy[3],
Scott Reed[4], Cheng-Yang Fu[1], Alexander C. Berg[1]

[1]UNC Chapel Hill [2]Zoox Inc. [3]Google Inc. [4]University of Michigan, Ann-Arbor
[1]wliu@cs.unc.edu, [2]drago@zoox.com, [3]{dumitru,szegedy}@google.com,
[4]reedscot@umich.edu, [1]{cyfu,aberg}@cs.unc.edu

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, Alexander C. Berg, “SSD: Single Shot MultiBox Detector,” ECCV 2016

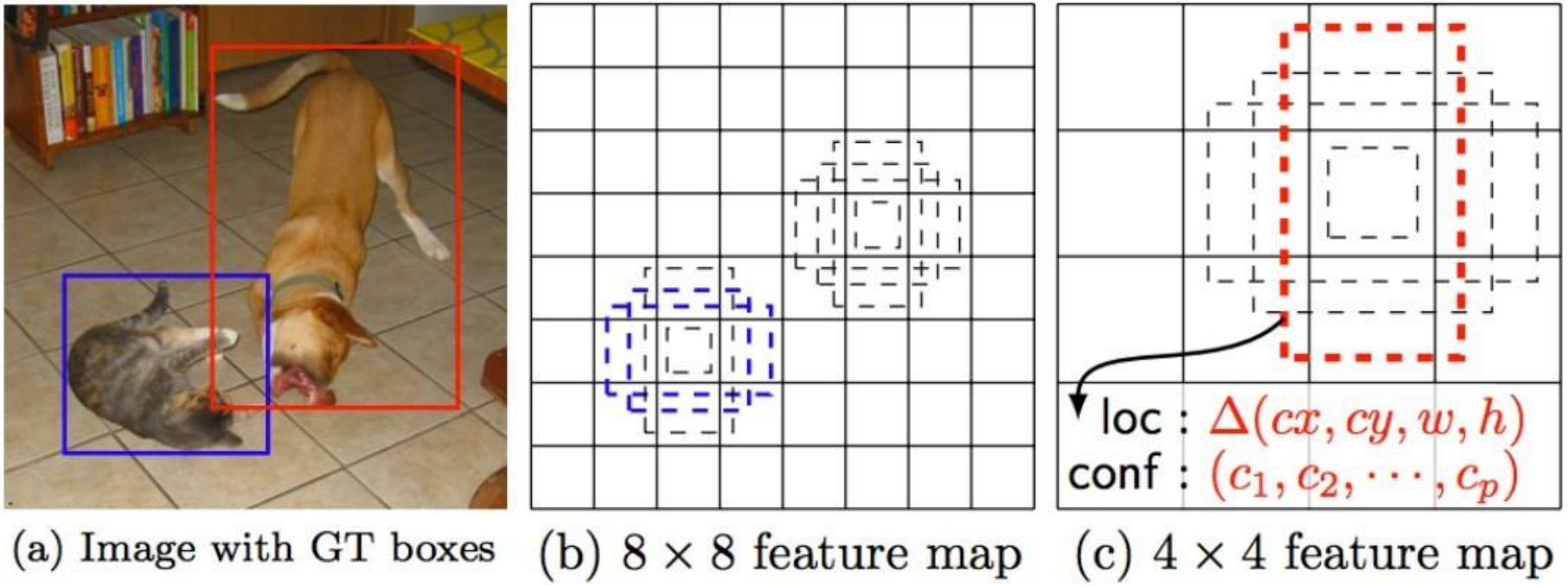
http://arxiv.org/abs/1512.02325

单阶段目标检测方法：SSD



Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, Alexander C. Berg, "SSD: Single Shot MultiBox Detector," ECCV 2016

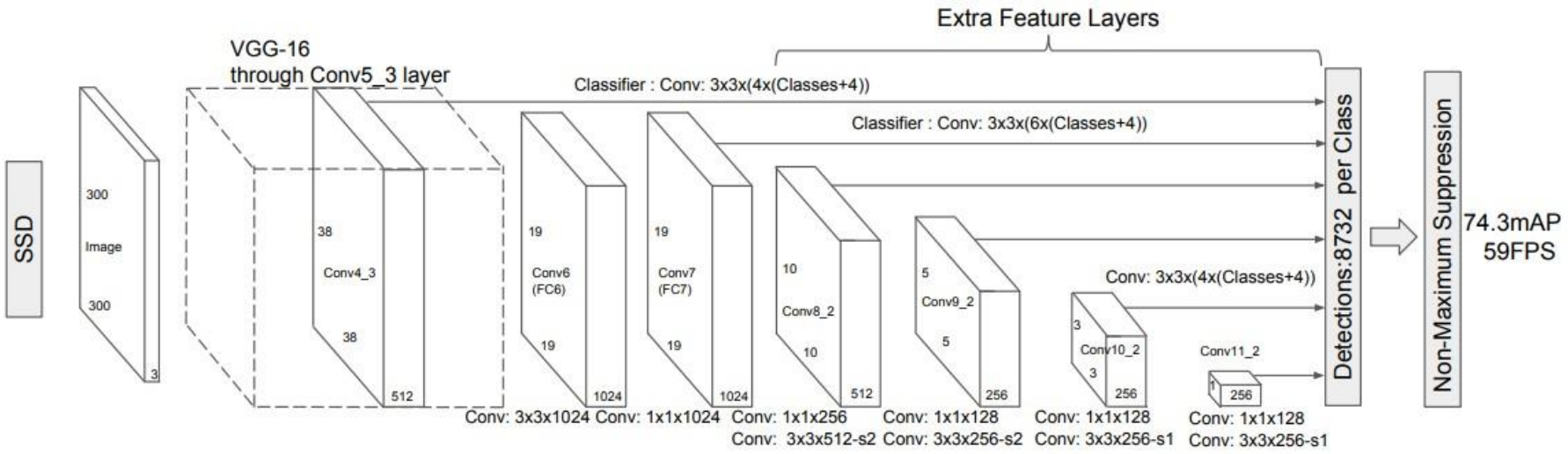
SSD=YOLO + Defult box shape + Multi-Scale



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

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, Alexander C. Berg, "SSD: Single Shot MultiBox Detector," ECCV 2016

SSD:框架再览



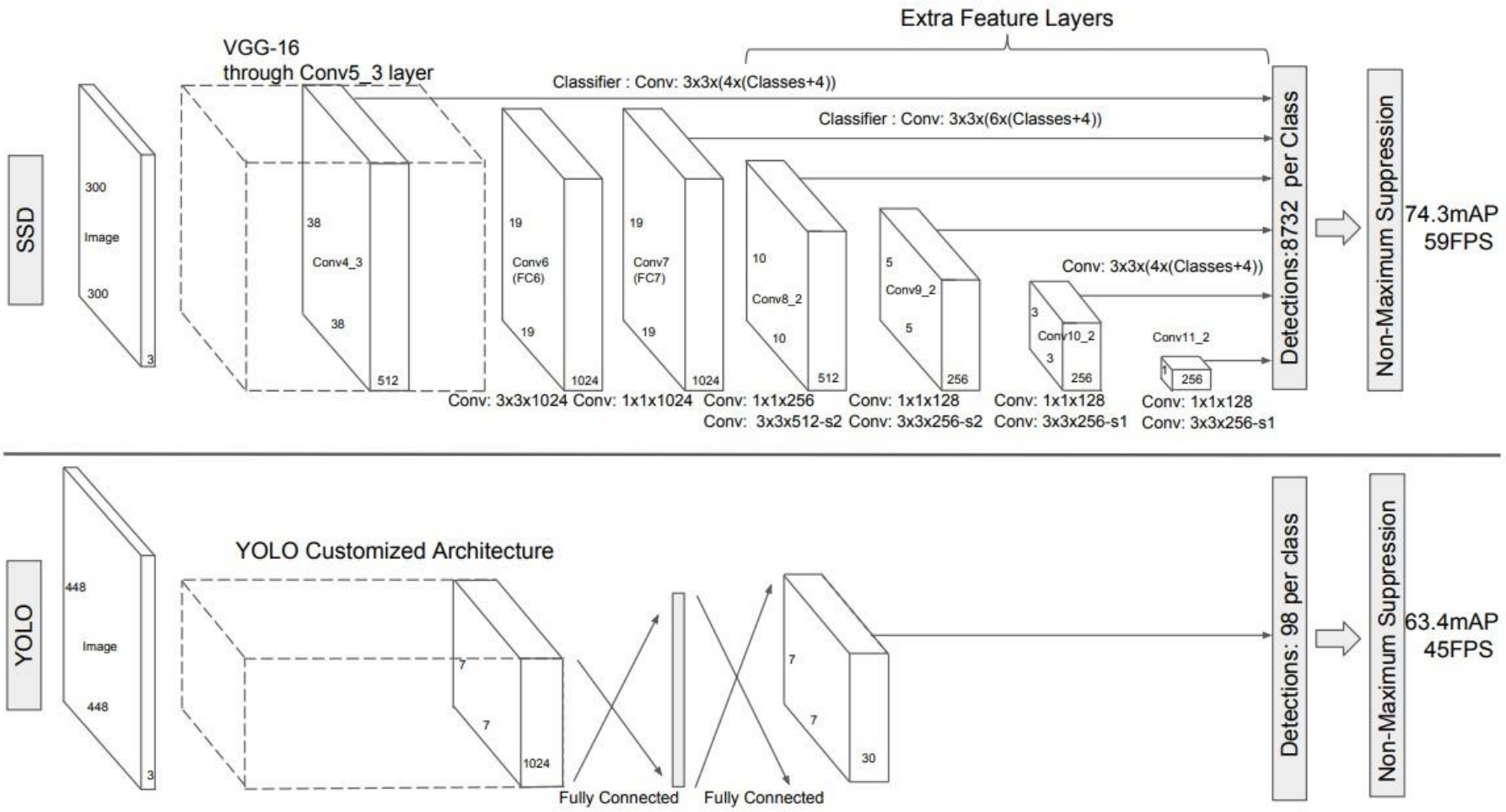
对于每个尺寸为m*n，通道数为p的特征层，SSD使用3*3*p的卷积核去产生对应于多个default boxes的物体概率分布（分类）和相对偏移（回归）

So, for each conv layer considered, there are

(*classes* + 4) x *default boxes* x *m* x *n*
outputs

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, Alexander C. Berg, “SSD: Single Shot MultiBox Detector,” ECCV 2016

SSD vs. YOLO



Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, Alexander C. Berg, "SSD: Single Shot MultiBox Detector," ECCV 2016

SSD：损失函数

Training objective:

Similar to MultiBox but handles multiple categories.

$$L(x, c, l, g) = \frac{1}{N}(\underbrace{L_{conf}(x, c)}_{\text{confidence loss}} + \alpha \underbrace{L_{loc}(x, l, g)}_{\text{localization loss}})$$

confidence loss
softmax loss

localization loss
Smooth L1 loss

is 1 by
cross-validation

N: number of default matched BBs
x: is 1 if the default box is matched to a determined ground truth box, and 0 otherwise
l: predicted bb parameters
g: ground truth bb parameters
c: class

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, Alexander C. Berg, “SSD: Single Shot MultiBox Detector,” ECCV 2016

SSD：损失函数

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

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

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

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

d: default bounding box
i: index of default box
j: index of ground truth box

N: number of default matched BBs
x: is 1 if the default box is matched to a determined ground truth box, and 0 otherwise
l: predicted bb parameters
g: ground truth bb parameters
c: class

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, Alexander C. Berg, “SSD: Single Shot MultiBox Detector,” ECCV 2016

SSD：总结

■ 实验结果

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	~ 1000 × 600
Fast YOLO	52.7	155	1	98	448 × 448
YOLO (VGG16)	66.4	21	1	98	448 × 448
SSD300	74.3	46	1	8732	300 × 300
SSD512	76.8	19	1	24564	512 × 512
SSD300	74.3	59	8	8732	300 × 300
SSD512	76.8	22	8	24564	512 × 512

■ 特点

- 对不同尺度feature map中目标进行预测，提高检测精度
- 运用底层特征可以检测出小物体
- 同时提高速度和精度，速度比YOLO快，精 度和早期版本的Faster R-CNN差不多

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, Alexander C. Berg, “SSD: Single Shot MultiBox Detector,” ECCV 2016

单阶段目标检测方法：RetinaNet

Focal Loss for Dense Object Detection

Tsung-Yi Lin Priya Goyal Ross Girshick Kaiming He Piotr Dollár
Facebook AI Research (FAIR)

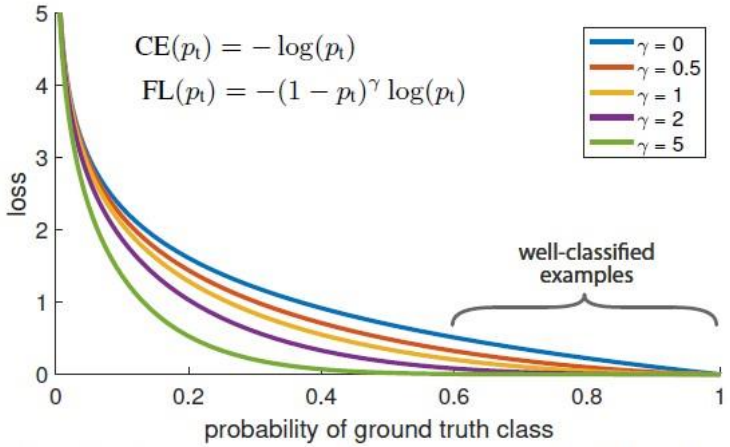
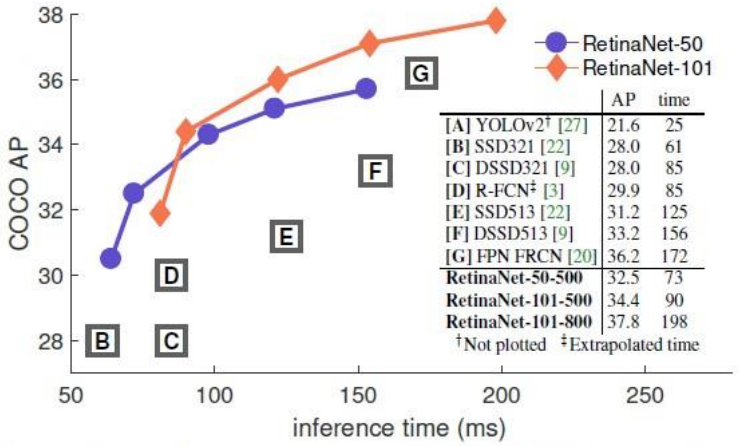


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1 - p_t)^\gamma$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments will demonstrate, the proposed focal loss enables training highly accurate dense object detectors in the presence of vast numbers of easy background examples.



	AP	time
[A] YOLOv2† [27]	21.6	25
[B] SSD321 [22]	28.0	61
[C] DSSD321 [9]	28.0	85
[D] R-FCN‡ [3]	29.9	85
[E] SSD513 [22]	31.2	125
[F] DSSD513 [9]	33.2	156
[G] FPN FRCN [20]	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	37.8	198

†Not plotted ‡Extrapolated time

Figure 2. Speed (ms) versus accuracy (AP) on COCO `test-dev`. Enabled by the focal loss, our simple one-stage *RetinaNet* detector outperforms all previous one-stage and two-stage detectors, including the best reported Faster R-CNN [28] system from [20]. We show variants of RetinaNet with ResNet-50-FPN (blue circles) and ResNet-101-FPN (orange diamonds) at five scales (400-800 pixels). Ignoring the low-accuracy regime (AP<25), RetinaNet forms an upper envelope of all current detectors, and an improved variant (not shown) achieves 40.8 AP. Details are given in §5.

Abstract

Lin, Tsung-Yi, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal loss for dense object detection." ICCV2017, Best Student Paper.

https://arxiv.org/abs/1708.02002

RetinaNet：出发点

-	one stage系	two stage系
代表性算法	YOLOv1、SSD、YOLOv2、YOLOv3	R-CNN、SPPNet、Fast R-CNN、Faster R-CNN
检测精度	低	高
检测速度	快	慢

- One Stage方法为什么效果差？ 类别不平衡！

1. #object vs. #background 背景太多了！
2. 类别不平衡导致准确率降低 模型聚焦于大量的背景样本中！
3. 为什么two-stage方法效果好？ RPN预筛了大量背景样本！
4. 为什么one-stage方法效果差？ 直接对大量样本进行分类！

思路：降低背景样本在训练中的权重

https://blog.csdn.net/JNingWei/article/details/80038594

RetinaNet：思想

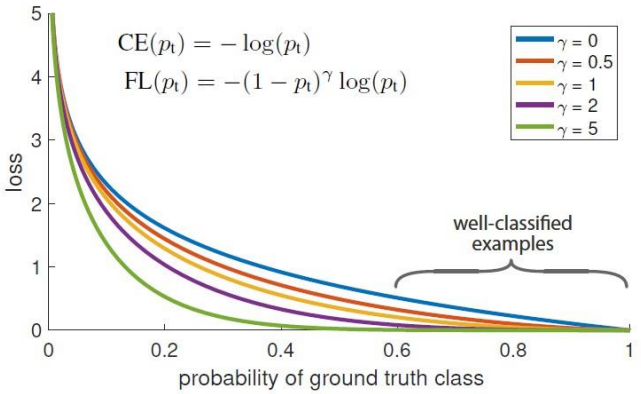


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1-p_t)^{\gamma}$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments will demonstrate, the proposed focal loss enables training highly accurate dense object detectors in the presence of vast numbers of easy background examples.

常规的交叉熵损失公式如下：

$$\mathrm{CE}(p,y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases}$$

当为正样本（y=1），则置信分数p越大loss越小，当为负样本（y=-1），则置信分数p越大则loss越小。
若将p和（1-p）统一表示为如下：

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise,} \end{cases}$$

则交叉熵可以简化为：

$$\mathrm{CE}(p,y) = \mathrm{CE}(p_t) = -\log(p_t).$$

这种损失有一个显著的特征，就是即使是很容易分类的例子（pt》0.5）也会造成损失，理论上这种easy的样本本不应对检测器产生影响，但是当有大量简单的样本存在时，即使他们各自产生的loss很小，他们loss的总和可能会对检测器产生巨大影响。
为此，作者提出Focal Loss：

$$\mathrm{FL}(p_t) = -(1-p_t)^{\gamma}\log(p_t).$$

使得越easy（pt越接近于1）的样本权值越小。比如，当gamma=2，pt=0.9时简单样本的loss会减小100倍。从而使得loss主要来源于更challenge的样本。

https://blog.csdn.net/qinhuai1994/article/details/80643447

RetinaNet：框架

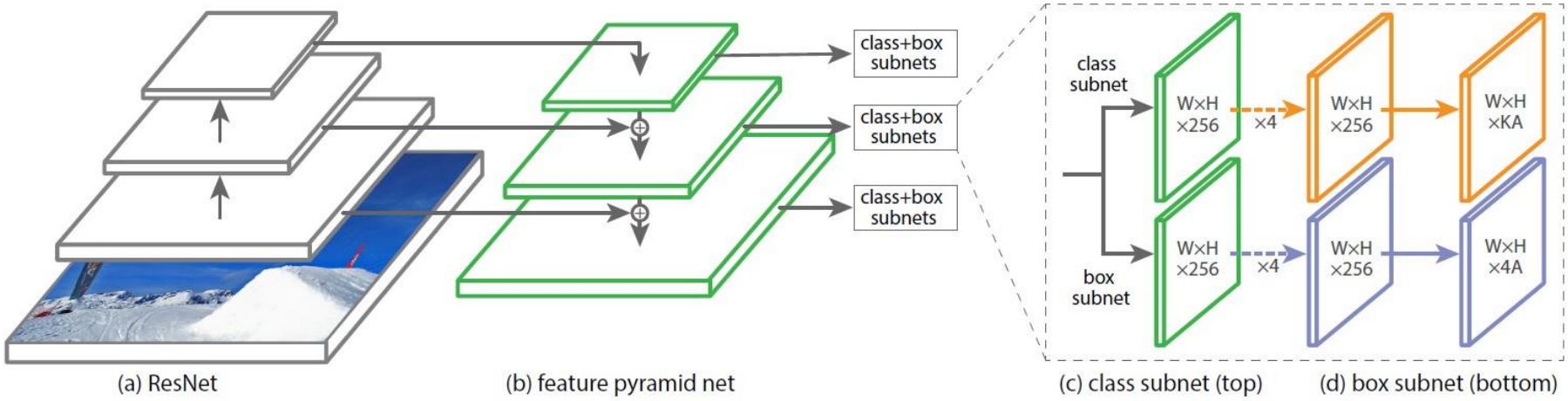


Figure 3. The one-stage **RetinaNet** network architecture uses a Feature Pyramid Network (FPN) [20] backbone on top of a feedforward ResNet architecture [16] (a) to generate a rich, multi-scale convolutional feature pyramid (b). To this backbone RetinaNet attaches two subnetworks, one for classifying anchor boxes (c) and one for regressing from anchor boxes to ground-truth object boxes (d). The network design is intentionally simple, which enables this work to focus on a novel focal loss function that eliminates the accuracy gap between our one-stage detector and state-of-the-art two-stage detectors like Faster R-CNN with FPN [20] while running at faster speeds.

4个卷积层、A个Anchor Boxes、K个类别

Lin, Tsung-Yi, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal loss for dense object detection." ICCV2017, Best Student Paper.

RetinaNet：总结

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [16]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [20]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [17]	Inception-ResNet-v2 [34]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [32]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [27]	DarkNet-19 [27]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [22, 9]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [9]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet (ours)	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet (ours)	ResNeXt-101-FPN	**40.8**	**61.1**	**44.1**	**24.1**	**44.2**	51.2

Table 2. **Object detection** *single-model* results (bounding box AP), *vs.* state-of-the-art on COCO `test-dev`. We show results for our RetinaNet-101-800 model, trained with scale jitter and for 1.5× longer than the same model from Table 1e. Our model achieves top results, outperforming both one-stage and two-stage models. For a detailed breakdown of speed versus accuracy see Table 1e and Figure 2.

- 简单，实用！
- 在多个其他视觉任务上也取得了很好的效果

Lin, Tsung-Yi, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal loss for dense object detection." ICCV2017, Best Student Paper.

荟萃：目标检测方法对比

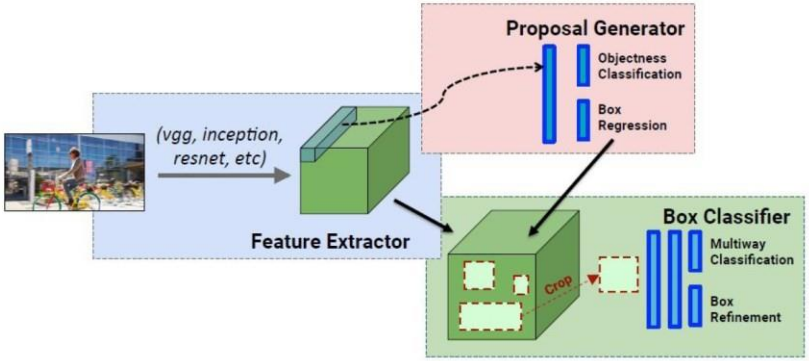
荟萃：Speed / Accuracy Trade-off

Unified tensorflow architecture

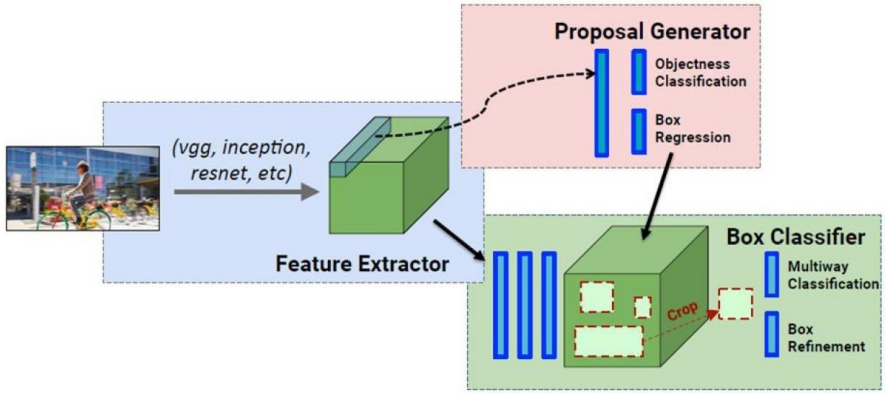
Compare speed, accuracy and memory usage

Speed/Accuracy trade-offs for modern convolutional object detectors
Jonathan Huang, Kevin Murphy et al.
Nov 2016

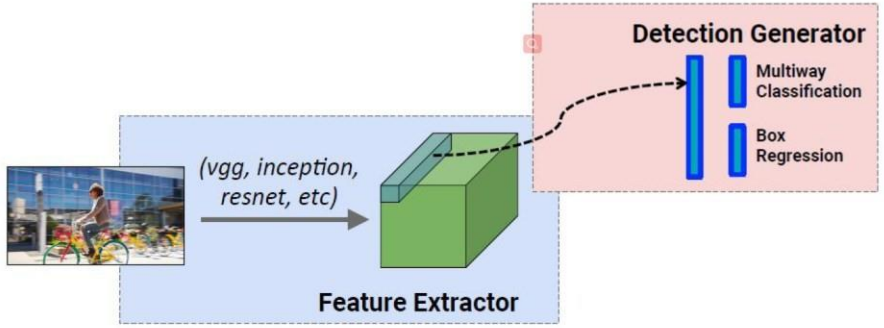
荟萃：比较方法



Faster RCNN

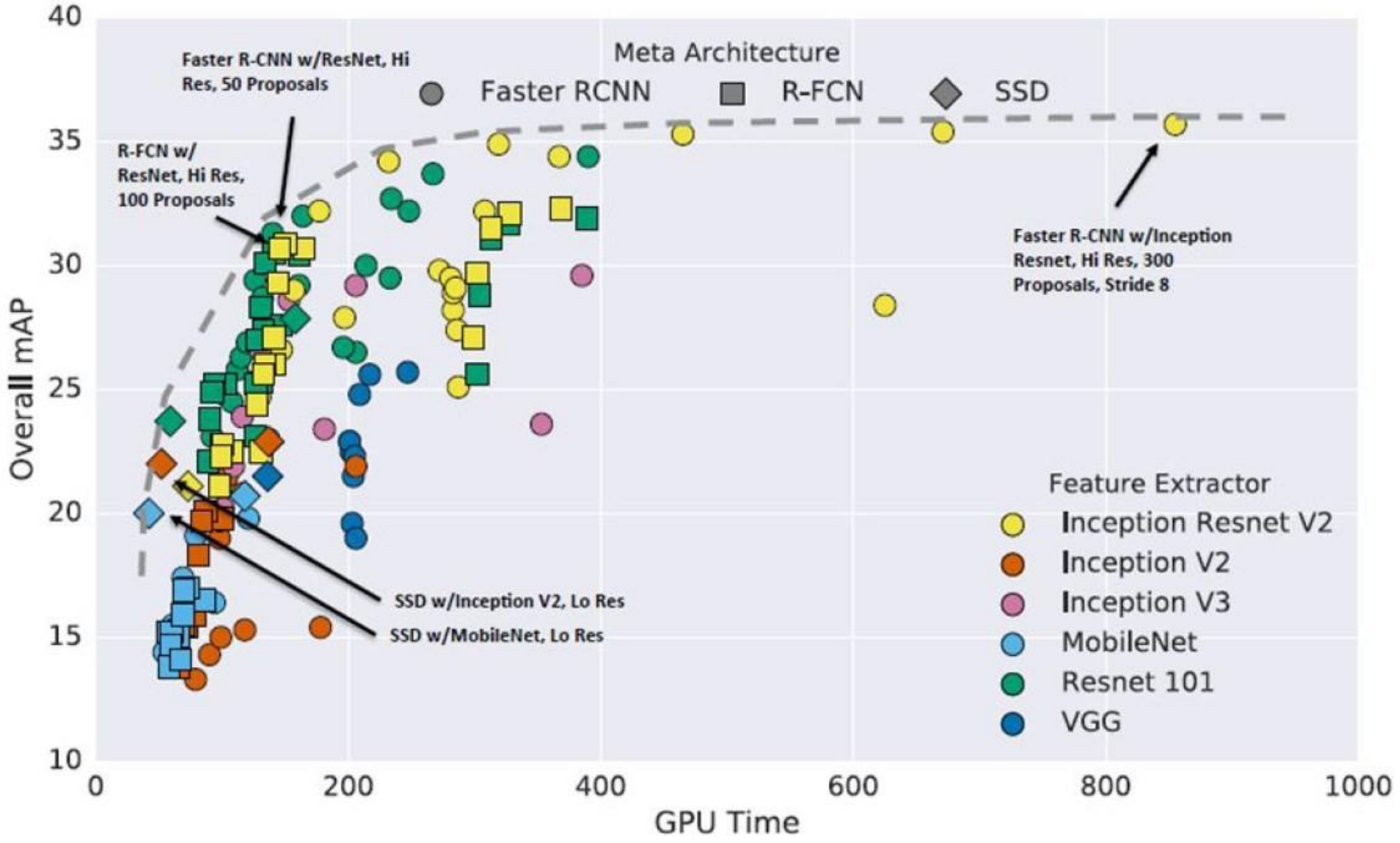


RFCN

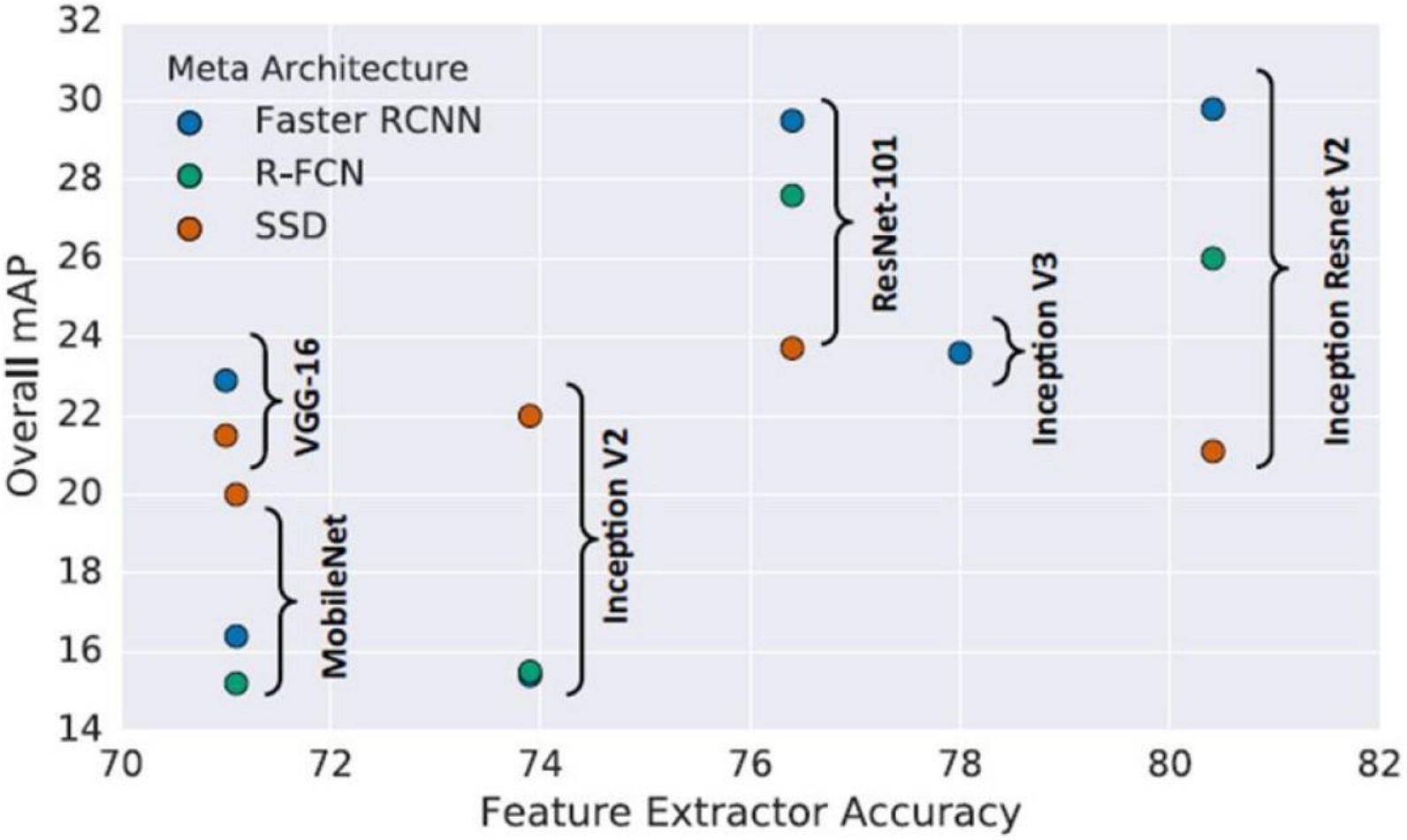


SSD

荟萃：Accuracy VS. Speed



荟萃：不同的特征提取器


Meta Architecture
Faster RCNN
R-FCN
SSD
VGG-16
MobileNet
Inception V2
ResNet-101
Inception V3
Inception Resnet V2
Overall mAP
Feature Extractor Accuracy
32
30
28
26
24
22
20
18
16
14
70
72
74
76
78
80
82

荟萃：不同的目标尺寸

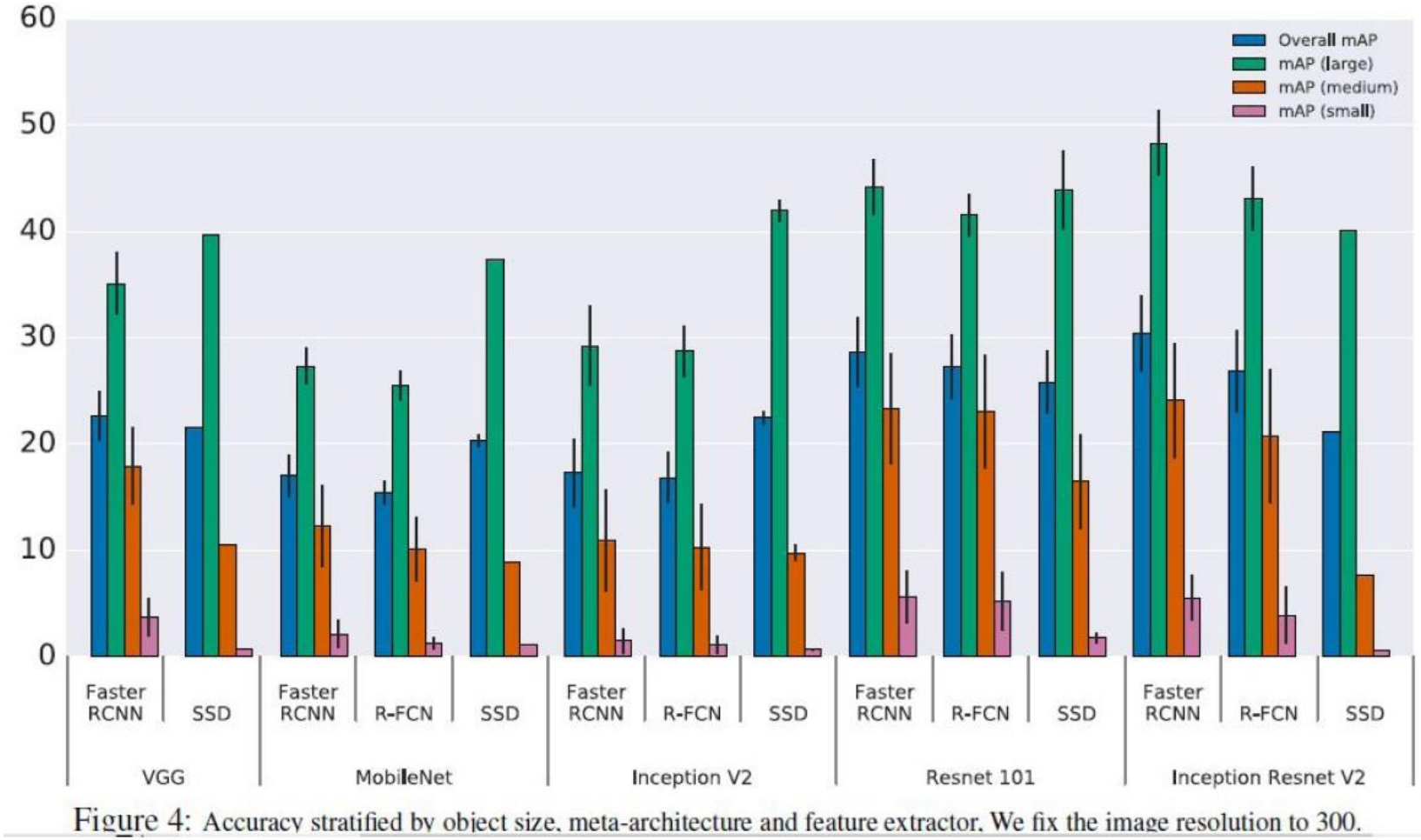
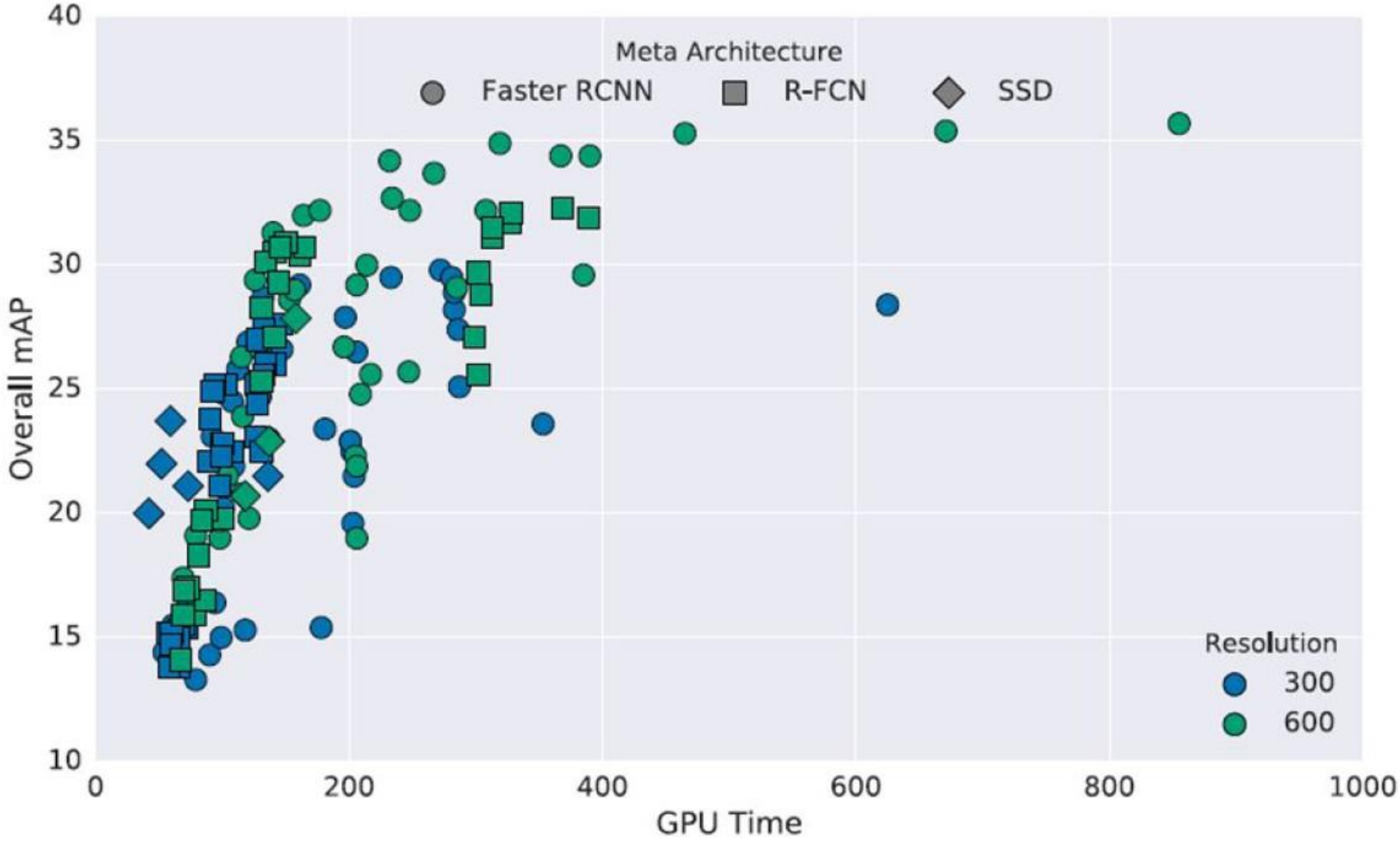
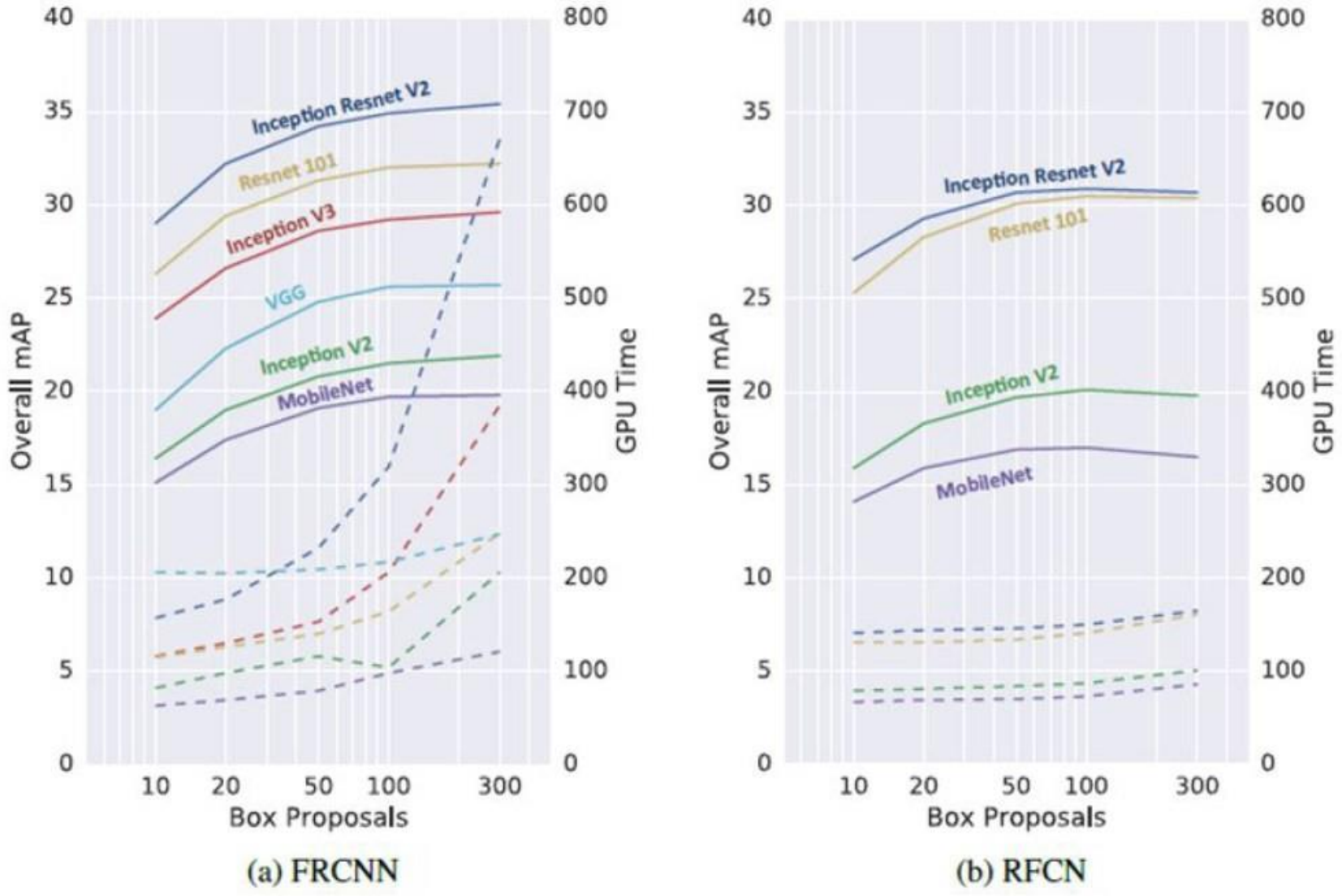


Figure 4: Accuracy stratified by object size, meta-architecture and feature extractor. We fix the image resolution to 300.

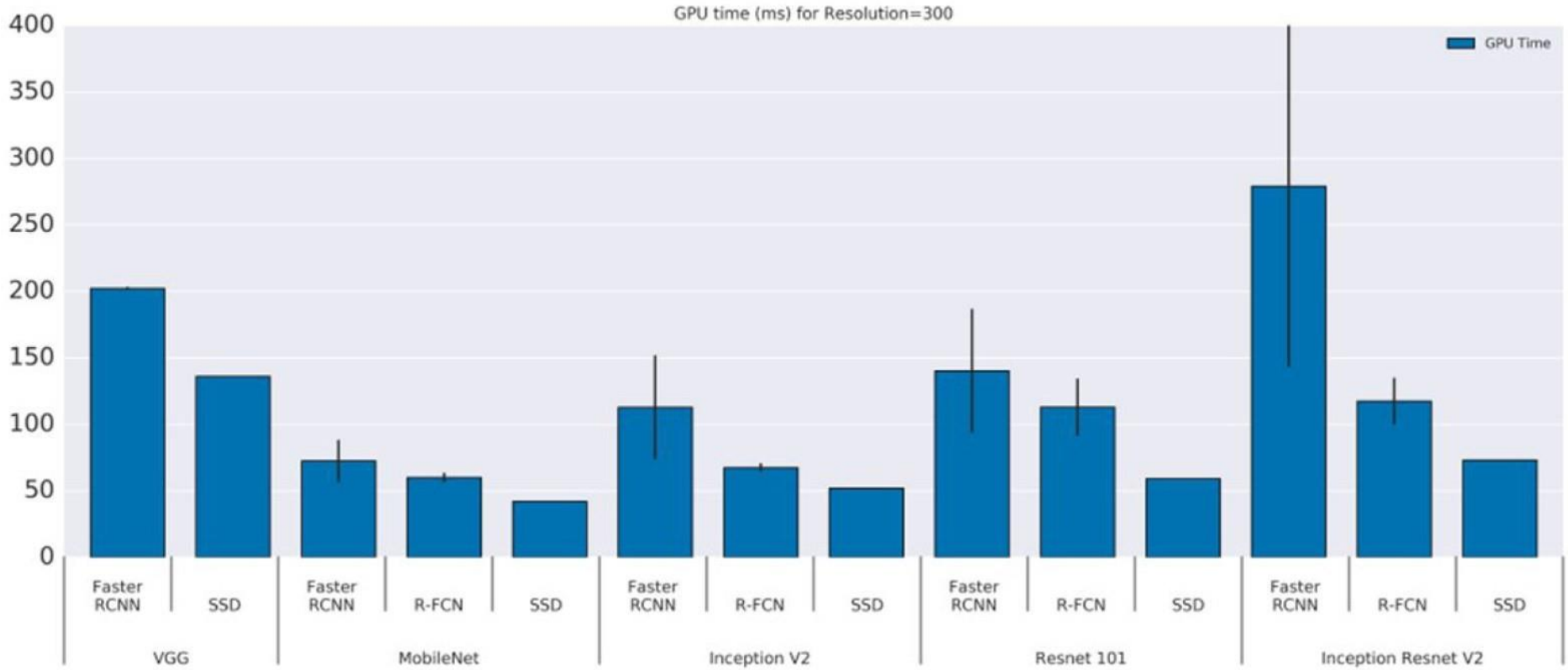
荟萃：不同的图像分辨率


Meta Architecture
Faster RCNN
R-FCN
SSD
Resolution
300
600
Overall mAP
GPU Time
40
35
30
25
20
15
10
0
200
400
600
800
1000

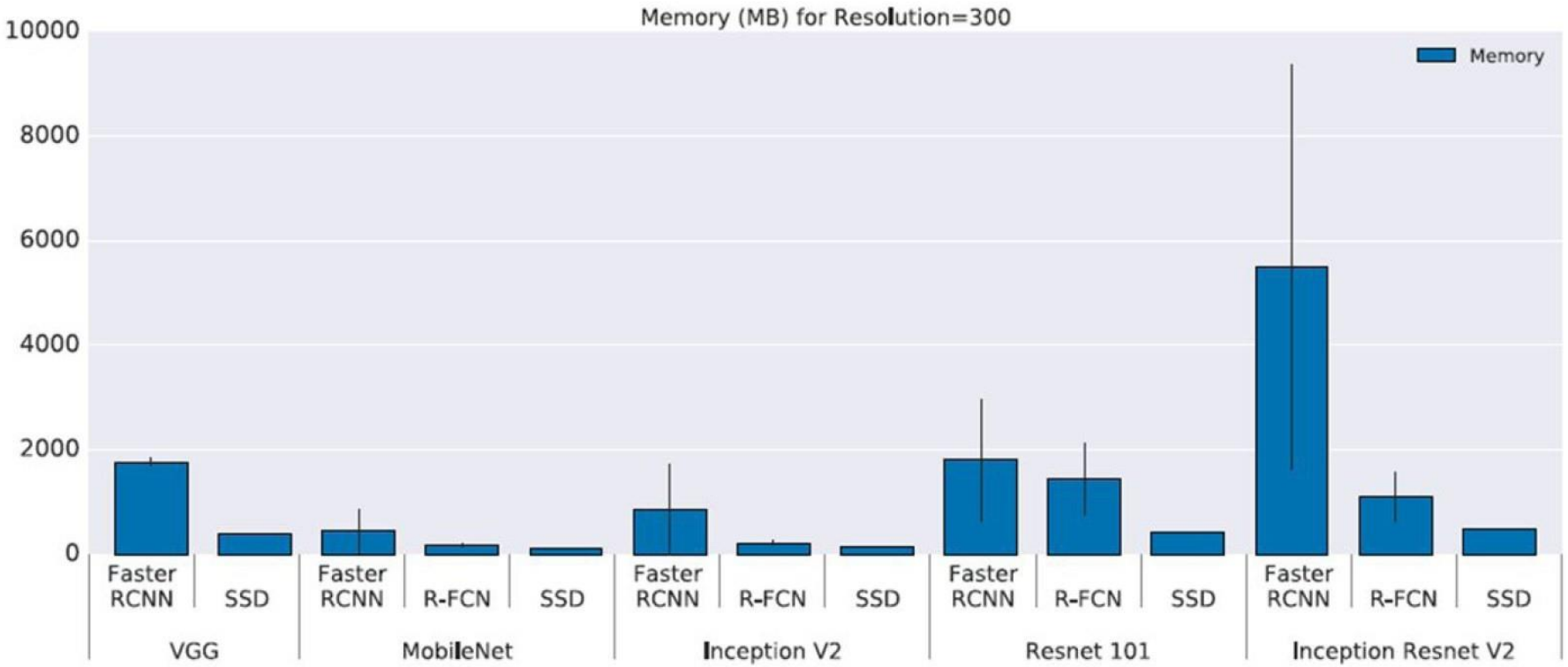
荟萃：不同的proposal数量



荟萃：速度（GPU）


GPU time (ms) for Resolution=300
GPU Time
400
350
300
250
200
150
100
50
0
Faster RCNN
SSD
Faster RCNN
R-FCN
SSD
Faster RCNN
R-FCN
SSD
Faster RCNN
R-FCN
SSD
Faster RCNN
R-FCN
SSD
VGG
MobileNet
Inception V2
Resnet 101
Inception Resnet V2

荟萃：内存


Memory (MB) for Resolution=300
Memory
10000
8000
6000
4000
2000
0
Faster RCNN
SSD
Faster RCNN
R-FCN
SSD
Faster RCNN
R-FCN
SSD
Faster RCNN
R-FCN
SSD
Faster RCNN
R-FCN
SSD
VGG
MobileNet
Inception V2
Resnet 101
Inception Resnet V2

荟萃：总结

1. 速度：SSD

2.效果：Faster RCNN

3.折中：RFCN

RetinaNet和一些其他的最新目标检测方法也是值得一试的对象！

10行代码实现目标检测

OlafenwaMoses/ImageAI

ImageAI - A python library built to empower developers to build applications and systems with self-contained Computer...

github.com

https://towardsdatascience.com/object-detection-with-10-lines-of-code-d6cb4d86f606

代码

```python
from imageai.Detection import ObjectDetection
import os

execution_path = os.getcwd()

detector = ObjectDetection()
detector.setModelTypeAsRetinaNet()
detector.setModelPath( os.path.join(execution_path , "resnet50_coco_best_v2.0.1.h5"))
detector.loadModel()
detections = detector.detectObjectsFromImage(input_image=os.path.join(execution_path , "image.jpg"), output_image_path=os.path.join(

for eachObject in detections:
    print(eachObject["name"] , " : " , eachObject["percentage_probability"] )
```

MOSES
OLAFENWA
OlafenwaMoses

Co-Founder & CEO at DeepQuest AI.
A self-Taught computer
programmer, Deep Learning,
Computer Vision Researcher and
Developer.

1. 在计算机上安装python
2. 安装ImageAI及其依赖库
3. 下载目标检测模型文件
4. 运行示例代码（只有十行）

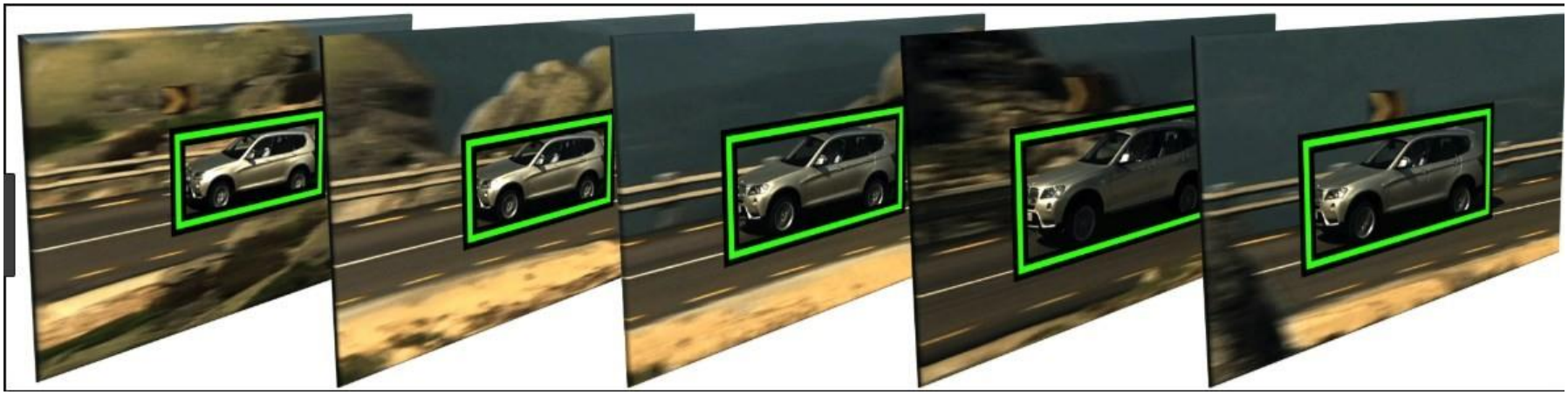
https://towardsdatascience.com/object-detection-with-10-lines-of-code-d6cb4d86f606

视频中目标检测

视频中目标检测：定义

把视频中每一帧出现的物体检测出来

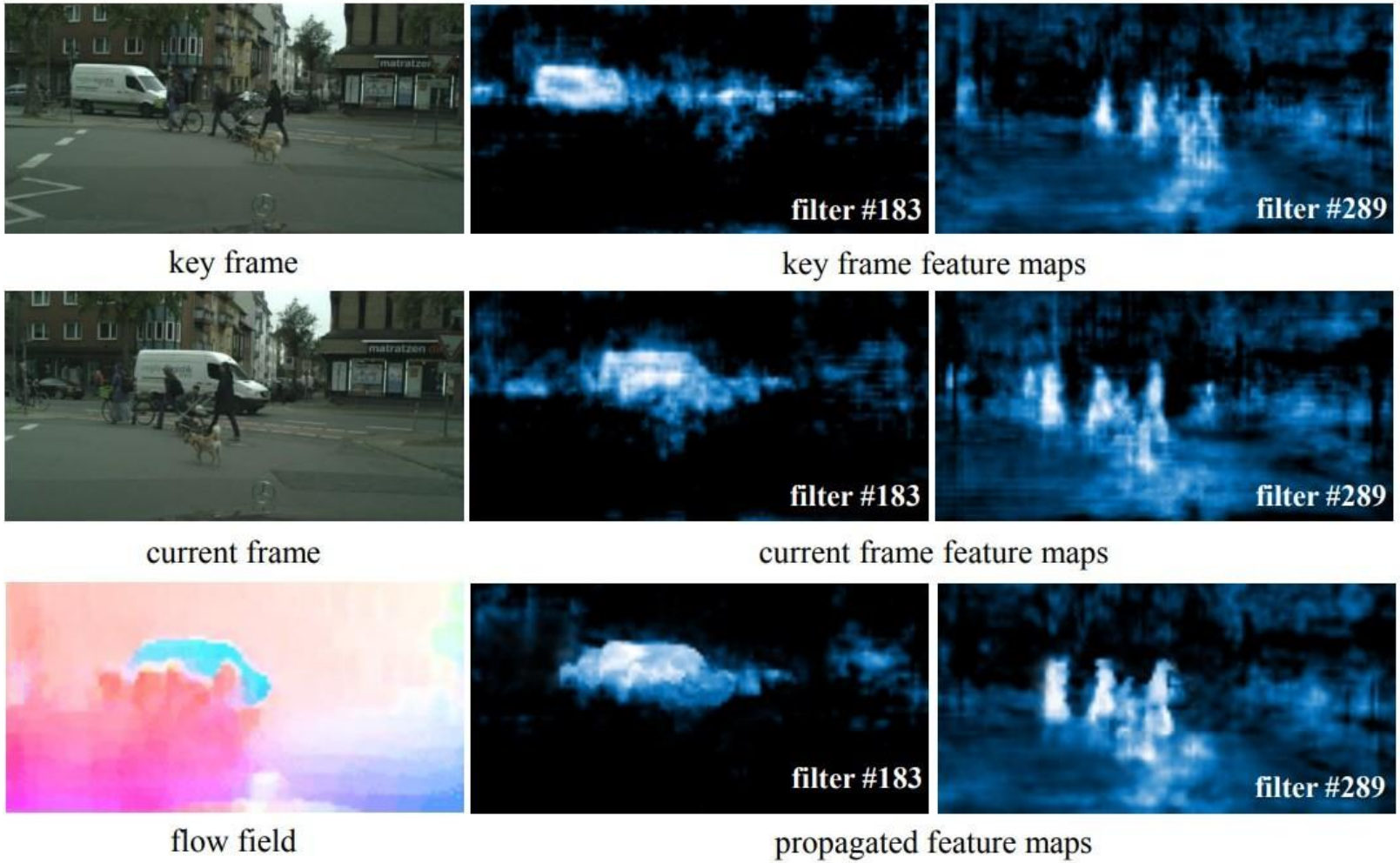
视频中目标检测： naïve solution



目标检测器，如
Faster RCNN等

耗时！

视频中目标检测：一种可能的思路

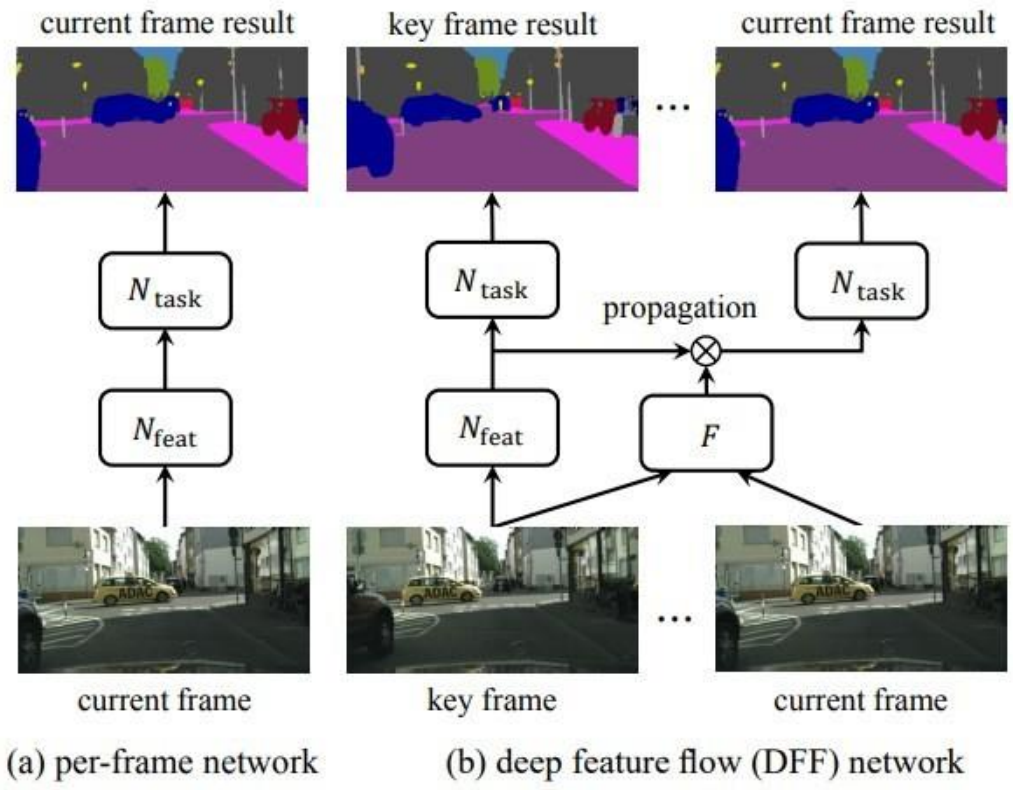


特征传递！

Zhu, Xizhou, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. "Deep feature flow for video recognition." CVPR2017.

https://arxiv.org/abs/1611.07715

视频中目标检测：Deep Feature Flow



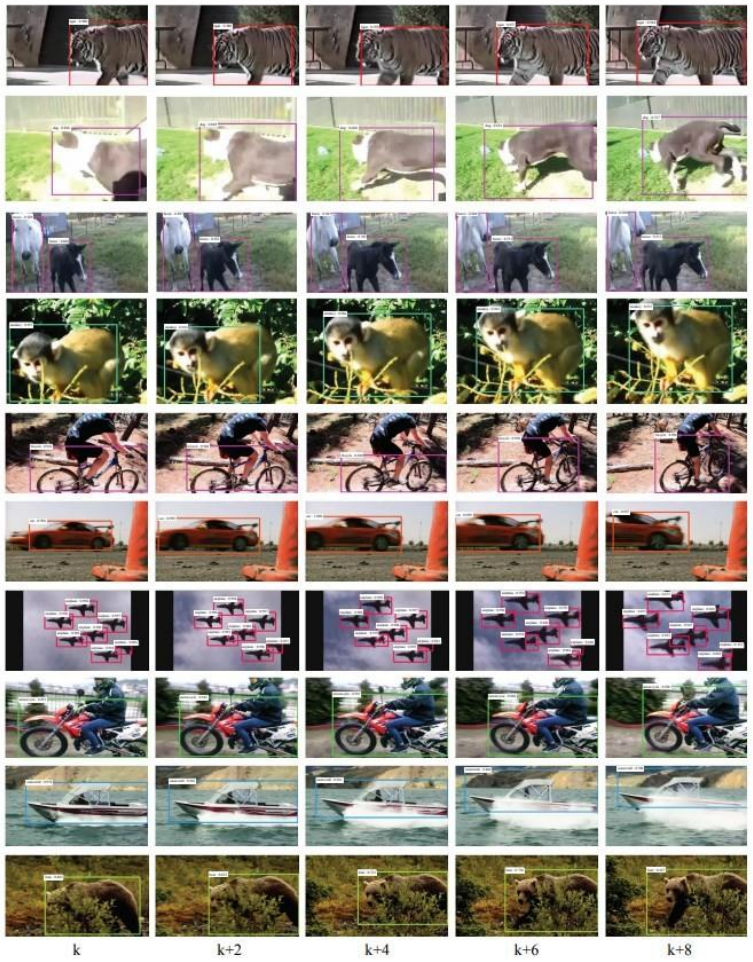
Algorithm 1 Deep feature flow inference algorithm for video recognition.

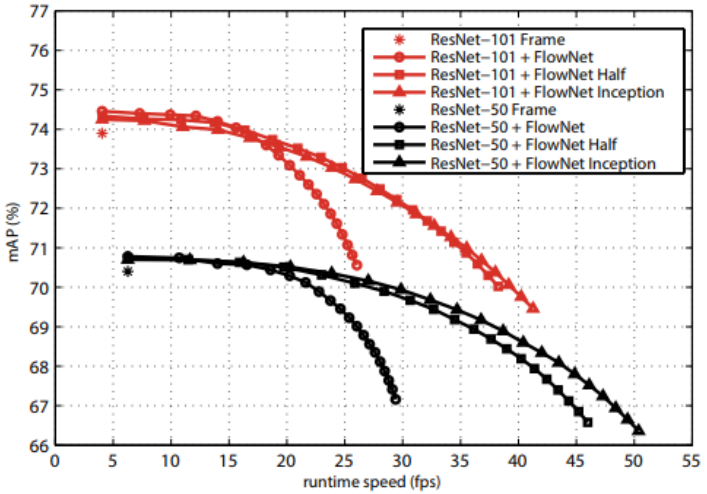
input: video frames $\{\mathbf{I}_i\}$
$k = 0$; ▷ initialize key frame
$\mathbf{f}_0 = \mathcal{N}_{feat}(\mathbf{I}_0)$
$\mathbf{y}_0 = \mathcal{N}_{task}(\mathbf{f}_0)$
for $i = 1$ **to** ∞ **do**
  if $is\_key\_frame(i)$ **then** ▷ key frame scheduler
    $k = i$ ▷ update the key frame
    $\mathbf{f}_k = \mathcal{N}_{feat}(\mathbf{I}_k)$
    $\mathbf{y}_k = \mathcal{N}_{task}(\mathbf{f}_k)$
  else ▷ use feature flow
    $\mathbf{f}_i = \mathcal{W}(\mathbf{f}_k, \mathcal{F}(\mathbf{I}_k, \mathbf{I}_i), \mathcal{S}(\mathbf{I}_k, \mathbf{I}_i))$ ▷ propagation
    $\mathbf{y}_i = \mathcal{N}_{task}(\mathbf{f}_i)$
  end if
end for
output: recognition results $\{\mathbf{y}_i\}$

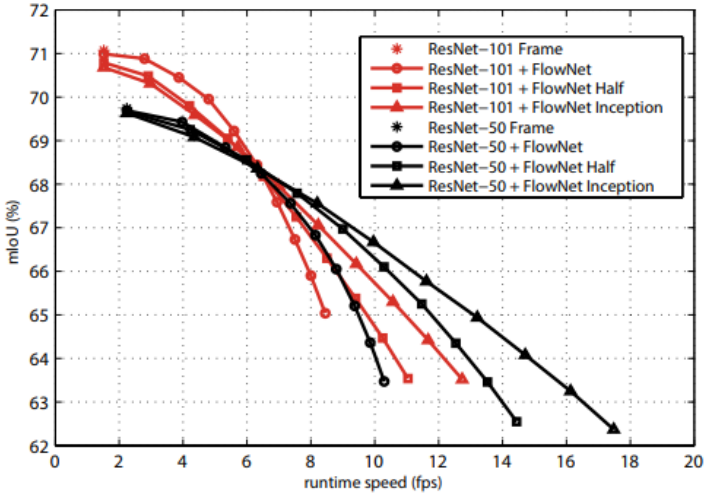
利用光流信息指导特征传递！

Zhu, Xizhou, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. "Deep feature flow for video recognition." CVPR2017.

Deep Feature Flow：效果



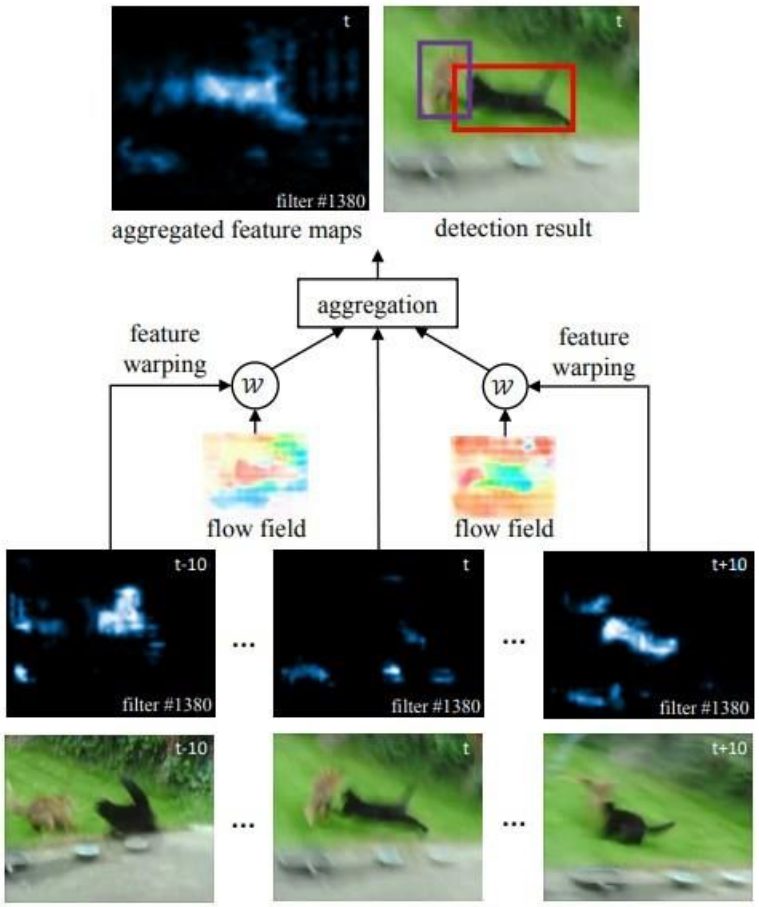




如何进一步改进？

Zhu, Xizhou, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. "Deep feature flow for video recognition." CVPR2017.

视频中目标检测：Flow-guided Feature Aggregation



Algorithm 1 Inference algorithm of flow guided feature aggregation for video object detection.

input: video frames $\{I_i\}$, aggregation range K
for $k = 1$ **to** $K + 1$ **do** ▷ initialize feature buffer
  $f_k = \mathcal{N}_{\text{feat}}(I_k)$
end for
for $i = 1$ **to** ∞ **do** ▷ reference frame
  for $j = \max(1, i - K)$ **to** $i + K$ **do** ▷ nearby frames
    $f_{j\to i} = \mathcal{W}(f_j, \mathcal{F}(I_i, I_j))$ ▷ flow-guided warp
    $f^e_{j\to i}, f^e_i = \mathcal{E}(f_{j\to i}, f_i)$ ▷ compute embedding features
    $w_{j\to i} = \exp(\frac{f^e_{j\to i} \cdot f^e_i}{|f^e_{j\to i}||f^e_i|})$ ▷ compute aggregation weight
  end for
  $\bar{f}_i = \sum_{j=i-K}^{i+K} w_{j\to i} f_{j\to i}$ ▷ aggregate features
  $y_i = \mathcal{N}_{\text{det}}(\bar{f}_i)$ ▷ detect on the reference frame
  $f_{i+K+1} = \mathcal{N}_{\text{feat}}(I_{i+K+1})$ ▷ update feature buffer
end for
output: detection results $\{y_i\}$

融合相邻帧特征，避免了目标检测中逐帧提特征的过程

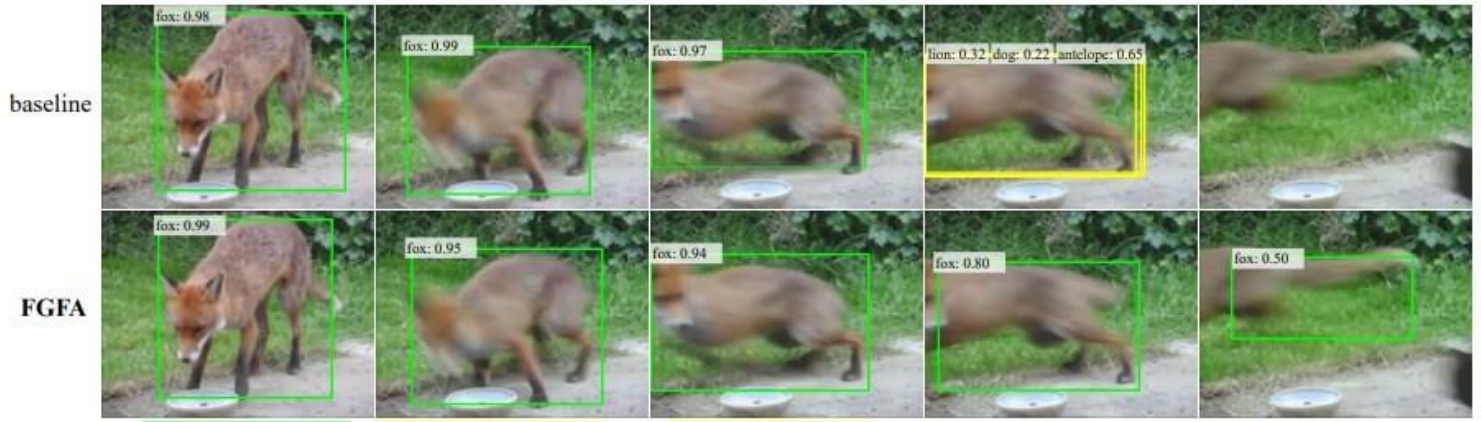
Zhu, Xizhou, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. "Flow-guided feature aggregation for video object detection." ICCV 2017.

https://arxiv.org/abs/1703.10025

Flow-guided Feature Aggregation：效果

$\mathcal{N}_{\text{feat}}$	ResNet-50					ResNet-101				
methods	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(e)
multi-frame feature aggregation?		✓	✓	✓	✓		✓	✓	✓	✓
adaptive weights?			✓	✓	✓			✓	✓	✓
flow-guided?				✓	✓				✓	✓
end-to-end training?		✓	✓	✓			✓	✓	✓	
mAP (%)	70.6	$69.6_{\downarrow 1.0}$	$71.8_{\uparrow 1.2}$	$\mathbf{74.0}_{\uparrow \mathbf{3.4}}$	$72.1_{\uparrow 1.5}$	73.4	$72.0_{\downarrow 1.4}$	$74.3_{\uparrow 0.9}$	$\mathbf{76.3}_{\uparrow \mathbf{2.9}}$	$74.5_{\uparrow 1.1}$
mAP (%) (slow)	79.3	$81.4_{\uparrow 2.1}$	$81.5_{\uparrow 2.2}$	$\mathbf{82.4}_{\uparrow \mathbf{3.1}}$	$81.3_{\uparrow 2.0}$	82.4	$82.3_{\downarrow 0.1}$	$82.2_{\downarrow 0.2}$	$\mathbf{83.5}_{\uparrow \mathbf{1.2}}$	$82.5_{\uparrow 0.1}$
mAP (%) (medium)	68.6	$71.4_{\uparrow 2.8}$	$71.4_{\uparrow 2.8}$	$\mathbf{72.6}_{\uparrow \mathbf{4.0}}$	$71.5_{\uparrow 2.9}$	71.6	$74.5_{\uparrow 2.9}$	$74.6_{\uparrow 3.0}$	$\mathbf{75.8}_{\uparrow \mathbf{4.2}}$	$74.6_{\uparrow 3.0}$
mAP (%) (fast)	50.1	$42.5_{\downarrow 7.6}$	$50.4_{\uparrow 0.3}$	$\mathbf{55.0}_{\uparrow \mathbf{4.9}}$	$51.2_{\uparrow 1.1}$	51.4	$44.6_{\downarrow 6.8}$	$52.3_{\uparrow 0.9}$	$\mathbf{57.6}_{\uparrow \mathbf{6.2}}$	$53.2_{\uparrow 1.8}$
runtime (ms)	203	204	220	647	647	288	288	305	733	733

Table 1. Accuracy and runtime of different methods on ImageNet VID validation, using ResNet-50 and ResNet-101 feature extraction networks. The relative gains compared to the single-frame baseline (a) are listed in the subscript.



Zhu, Xizhou, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. "Flow-guided feature aggregation for video object detection." ICCV 2017.

总结

内容回顾

1. 目标检测

   两阶段方法

   单阶段方法

2. 视频中目标检测

One stage

- Yolov1 You Only Look Once: Unified, Real-Time Object Detection darknet caffe
- Yolov2 YOLO9000:Better, Faster, Stronger tensorflow pytorch caffe
- Yolov3 YOLOv3: An Incremental Improvement pytorch keras
- SSD SSD: Single Shot MultiBox Detector caffe tensorflow
- DSSD DSSD : Deconvolutional Single Shot Detector caffe
- RFB-SSD Receptive Field Block Net for Accurate and Fast Object Detection pytorch
- RetinaNet Focal Loss for Dense Object Detection caffe tensorflow
- RefineDet Single-Shot Refinement Neural Network for Object Detection caffe

Two stage

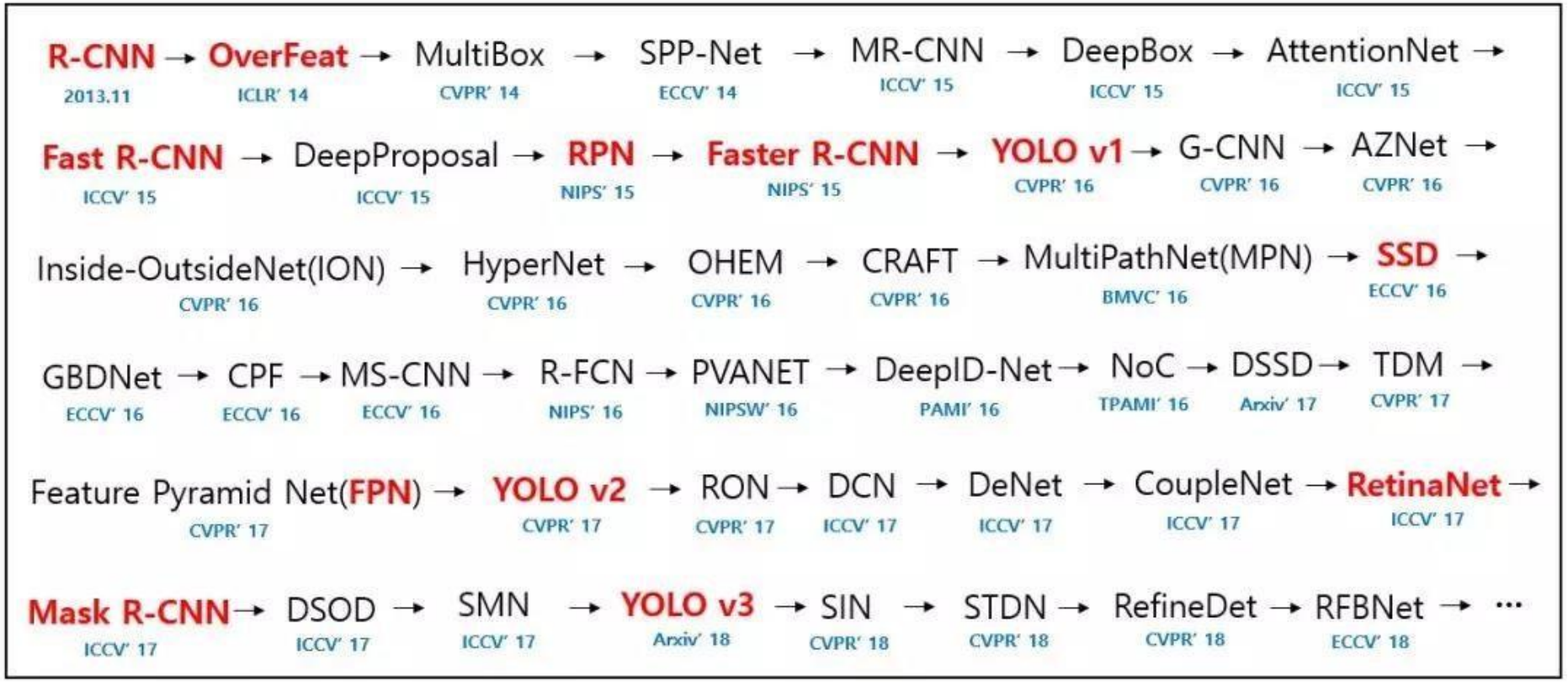
- RCNN Rich feature hierarchies for accurate object detection and semantic segmentation caffe
- Fast RCNN Fast RCNN caffe
- Faster RCNN Towards Real-Time Object Detection with Region Proposal Networks caffe tensorflow pytorch
- Mask RCNN Mask RCNN keras caffe2 tensorflow pytorch
- R-FCN R-FCN: Object Detection via Region-based Fully Convolutional Networks caffe tensorflow pytorch
- Light Head RCNN Light-Head R-CNN: In Defense of Two-Stage Object Detector tensorflow pytorch
- Cascade RCNN Cascade R-CNN: Delving into High Quality Object Detection caffe

到此为止，目标检测方法有很多……

R-CNN
Fast R-CNN
Faster R-CNN
Mask R-CNN
Light-Head R-CNN
Cascade R-CNN
SPP-Net
YOLO
YOLOv2
YOLOv3
YOLT
SSD
DSSD
FSSD
ESSD
MDSSD
Pelee
Fire SSD

R-FCN
FPN
DSOD
RetinaNet
MegDet
RefineNet
DetNet
SSOD
CornerNet
M2Det
3D Object Detection
ZSD（Zero-Shot Object Detection）
OSD（One-Shot object Detection）
Weakly Supervised Object Detection
Softer-NMS
2018
Other

到此为止，目标检测方法有很多……



Take Home Message：后续的研究方向

1. 网络结构？细化、再细化!

2. 知识信息？更高级的理解方式

3.其他的目标检测应用？细粒度、无监督、半监督、零样本、融合检测和分割、 融合检测和跟踪……

目标检测领域还有什么可以做的?
https://www.zhihu.com/question/280703314/answer/564235579

THANKS