

# 物体检测概述



主讲人 **张士峰**

中国科学院自动化研究所  
模式识别国家重点实验室博士





# 目录

-  物体检测研究背景
-  物体检测发展脉络
-  物体检测常用数据集
-  物体检测评价指标



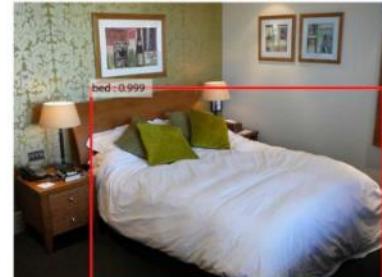
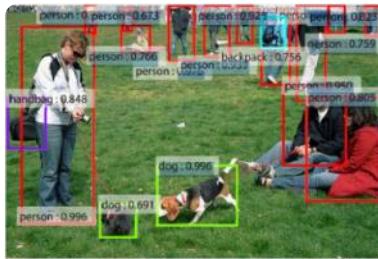
# 目录

-  物体检测研究背景
-  物体检测发展脉络
-  物体检测常用数据集
-  物体检测评价指标



# 物体检测的定义

- 物体检测：判断一副图像上是否存在感兴趣的物体，如果存在，就给出所有感兴趣物体的类别和位置  
(What and Where)





# 物体检测相关任务对比

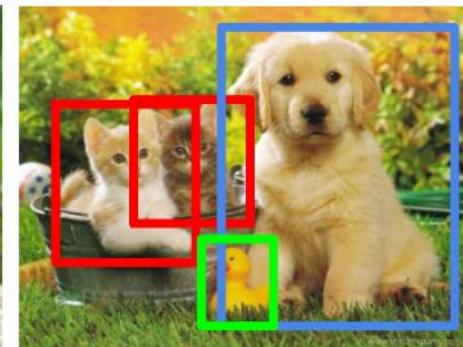
分类  
(Classification)



分类 + 定位  
(Classification+Localization)



物体检测  
(Object Detection)



实例分割  
(Instance Segmentation)



CAT

CAT

CAT, DOG, DUCK

CAT, DOG, DUCK

单个物体

多个物体



# 物体检测相关任务对比

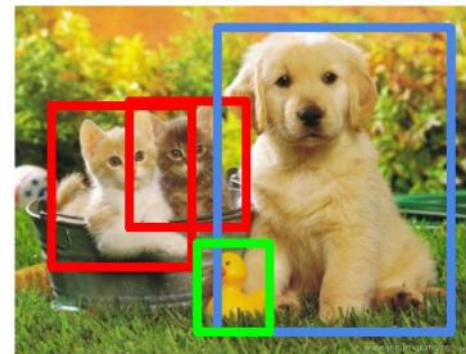
分类  
(Classification)



分类 + 定位  
(Classification+Localization)



物体检测  
(Object Detection)



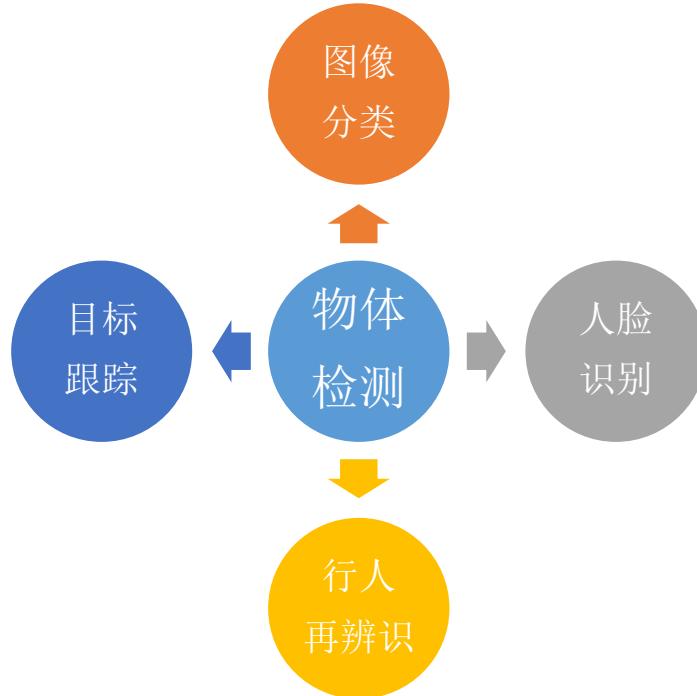
实例分割  
(Instance Segmentation)





# 物体检测的意义

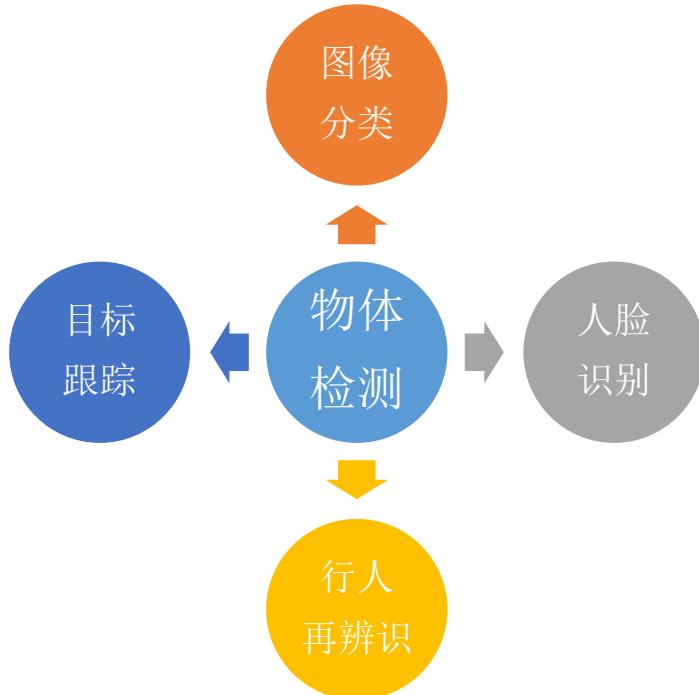
- 从研究角度来看，物体检测是计算机视觉的根本问题之一，是很多高层视觉任务的基础。





# 物体检测的意义

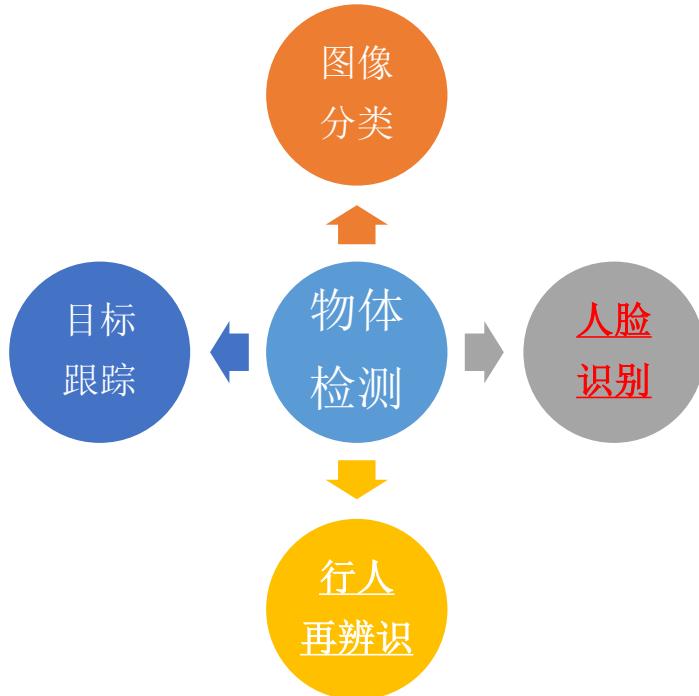
- 从研究角度来看，物体检测是计算机视觉的根本问题之一，是很多高层视觉任务的基础。





# 物体检测的意义

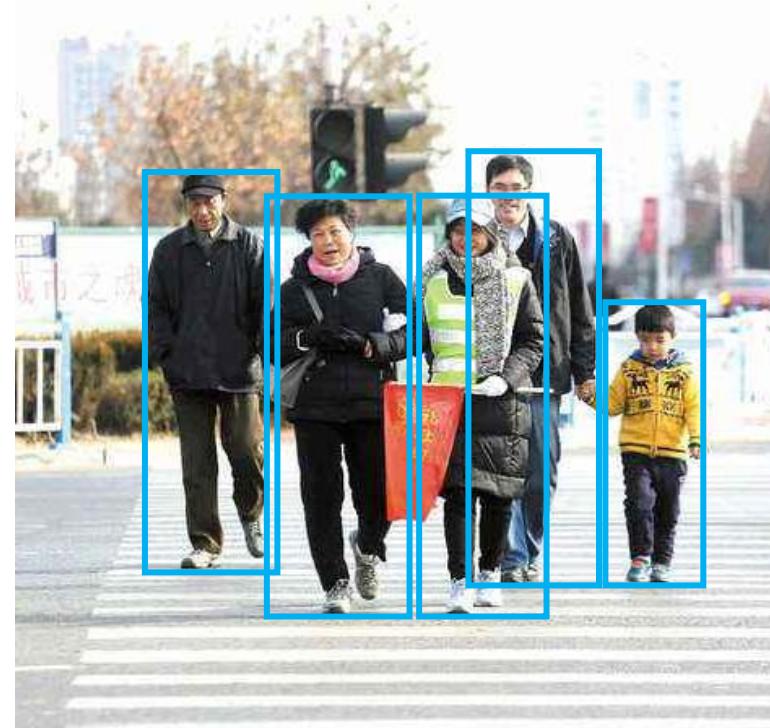
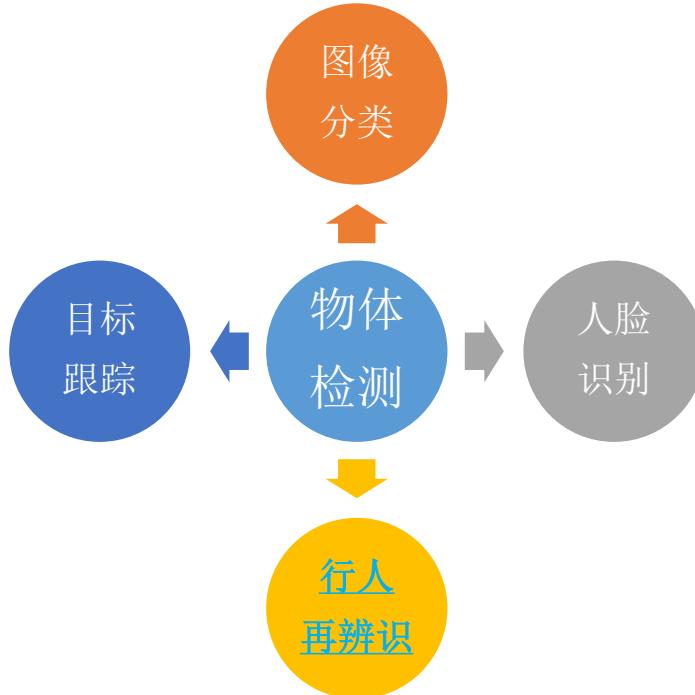
- 从研究角度来看，物体检测是计算机视觉的根本问题之一，是很多高层视觉任务的基础。





# 物体检测的意义

- 从研究角度来看，物体检测是计算机视觉的根本问题之一，是很多高层视觉任务的基础。





# 物体检测的意义

- 从研究角度来看，物体检测是计算机视觉的根本问题之一，是很多高层视觉任务的基础。

## 国外研究团队

- 麻省理工的计算机科学和人工智能实验室
- 卡内基梅隆大学的机器人系
- 加州大学伯克利分校的视觉和学习中心
- 加州理工大学的计算机视觉实验室
- 斯坦福大学的视觉实验室
- 加州大学洛杉矶分校的感知视觉和学习实验室
- 牛津大学的VGG研究组
- 苏黎世联邦理工大学计算机视觉研究组
- .....



# 物体检测的意义

- 从研究角度来看，物体检测是计算机视觉的根本问题之一，是很多高层视觉任务的基础。

## 国外研究团队

- 麻省理工的计算机科学和人工智能实验室
- 卡内基梅隆大学的机器人系
- 加州大学伯克利分校的视觉和学习中心
- 加州理工大学的计算机视觉实验室
- 斯坦福大学的视觉实验室
- 加州大学洛杉矶分校的感知视觉和学习实验室
- 牛津大学的VGG研究组
- 苏黎世联邦理工大学计算机视觉研究组
- .....

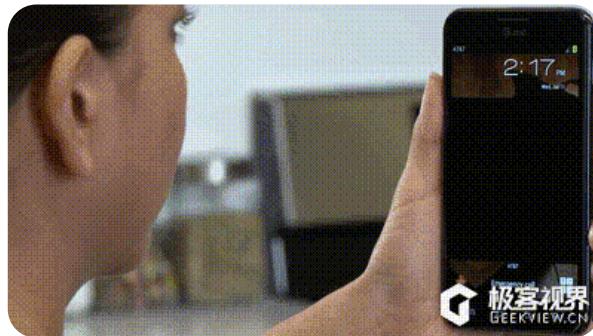
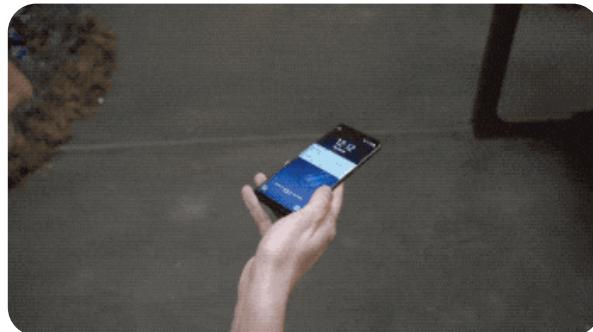
## 国内研究团队

- 中科院自动化所模式识别国家重点实验室
- 中科院计算所先进人机通信技术联合实验室
- 清华大学人机交互与媒体集成研究所
- 南京大学的LAMDA 研究组
- 香港中文大学的多媒体实验室
- 上海交通大学计算机视觉实验室
- 北京交通大学信息科学研究所
- 天津大学计算机视觉实验室
- .....



# 物体检测的意义

- 从应用角度来看，物体检测已经表现出广泛的应用需求。



## 人脸解锁

- 人脸检测技术



# 物体检测的意义

- 从应用角度来看，物体检测已经表现出广泛的应用需求。



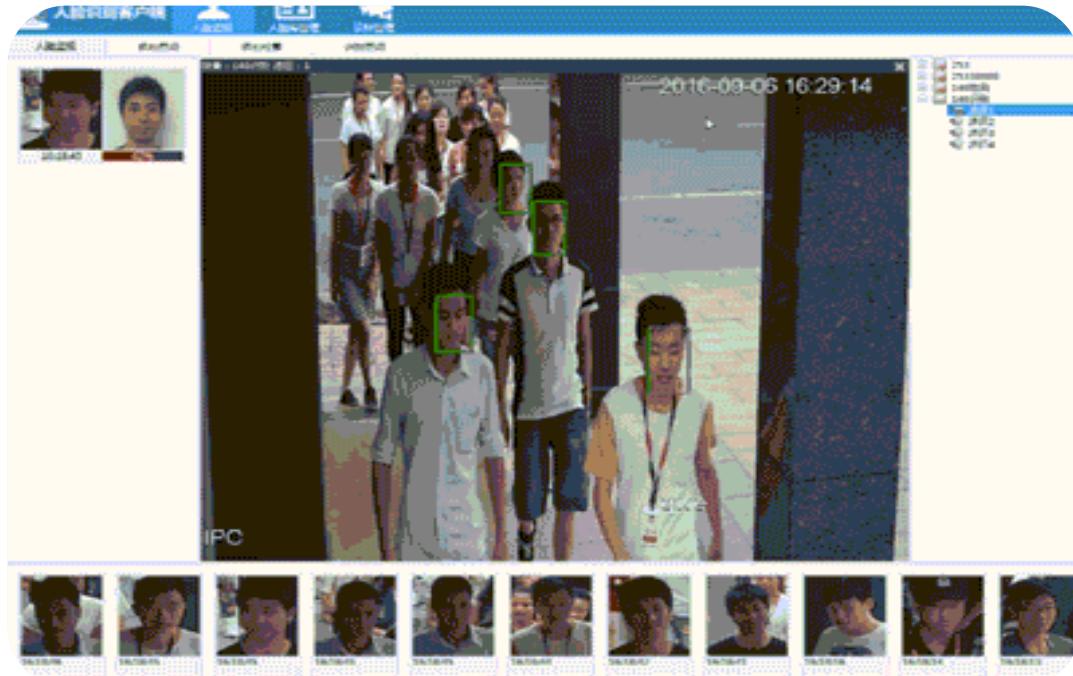
## 视频监控

- 人脸识别技术
- 行人检测技术



# 物体检测的意义

- 从应用角度来看，物体检测已经表现出广泛的应用需求。



## 出入口的人数统计

- 人脸检测技术
- 行人检测技术



# 物体检测的意义

- 从应用角度来看，物体检测已经表现出广泛的应用需求。



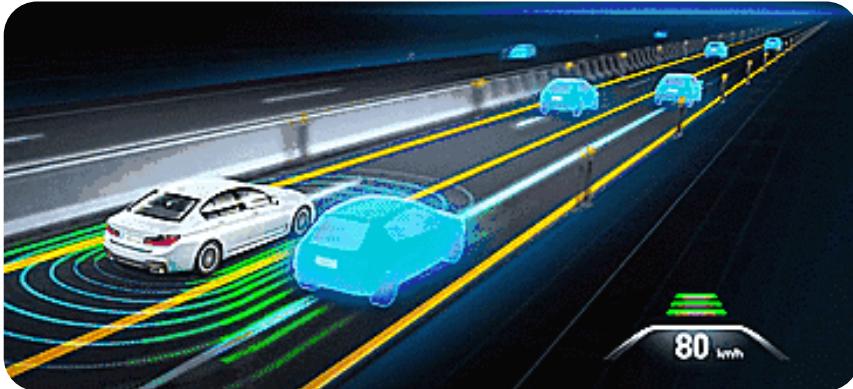
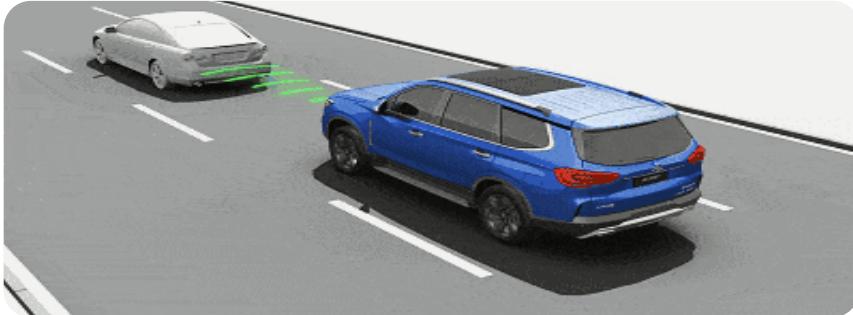
## 辅助驾驶

- 交通标志检测技术



# 物体检测的意义

- 从应用角度来看，物体检测已经表现出广泛的应用需求。



## 辅助驾驶

- 车辆检测技术
- 车道线检测技术
- .....



# 物体检测的意义

- 从应用角度来看，物体检测已经表现出广泛的应用需求。



## 车辆监控

- 车辆检测技术
- 车牌检测技术
- .....



# 物体检测的意义

- 从应用角度来看，物体检测已经表现出广泛的应用需求。



## 交通路口监控

- 车辆检测技术
- 行人检测技术
- .....



# 物体检测的意义

- 从应用角度来看，物体检测已经表现出广泛的应用需求。



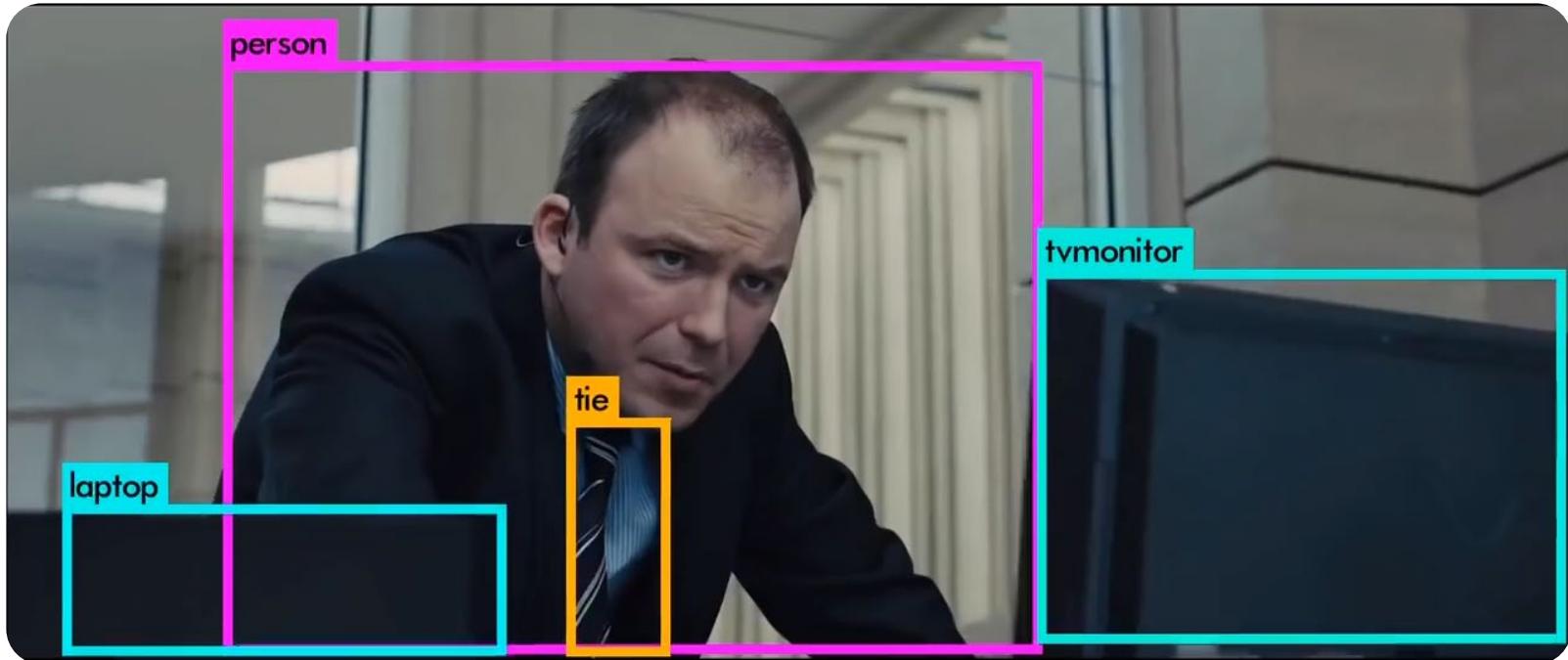
## 自动驾驶

- 车辆检测技术
- 行人检测技术
- 交通信号检测技术
- 行车道检测技术
- .....



## 物体检测的意义

- 从应用角度来看，物体检测已经表现出广泛的应用需求。





## 物体检测的意义

- 从应用角度来看，物体检测已经表现出广泛的应用需求。





## 物体检测的意义

- 从应用角度来看，物体检测已经表现出广泛的应用需求。



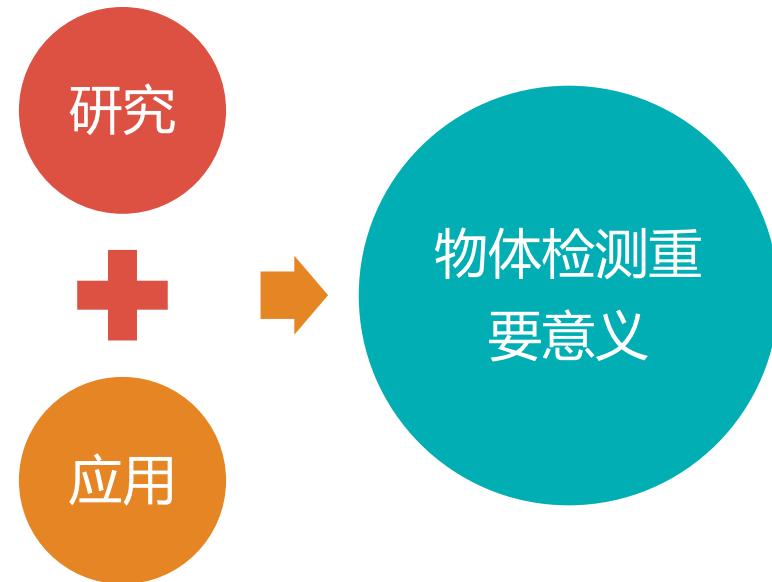
Tencent 腾讯





## 物体检测的意义

- 从研究角度来看，物体检测是计算机视觉的根本问题之一，是很多高层视觉任务的基础。
- 从应用角度来看，物体检测已经表现出广泛的应用需求。





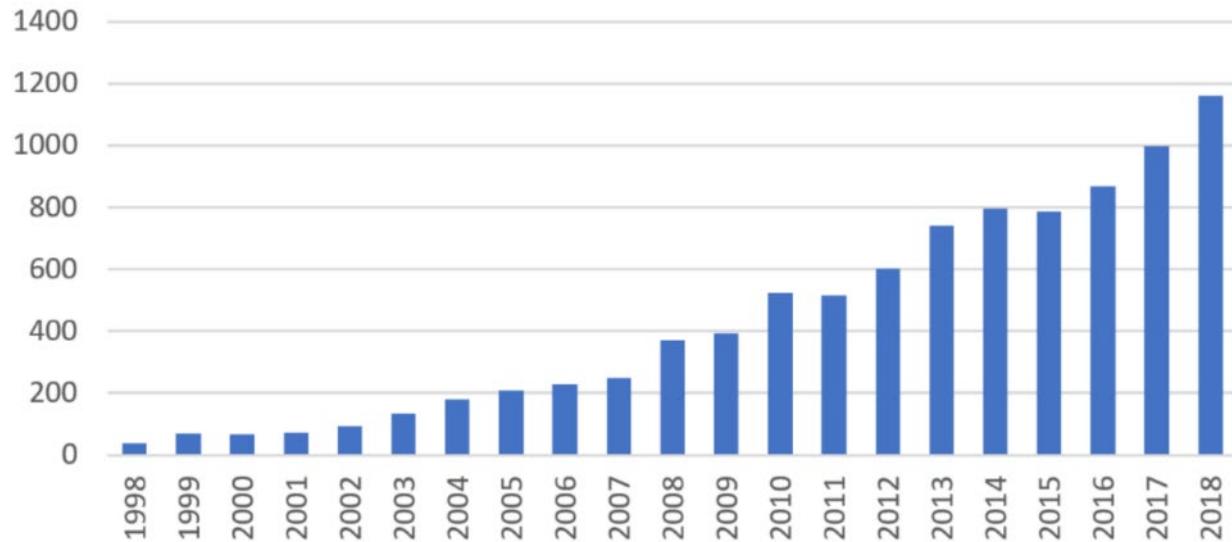
# 目录

-  物体检测研究背景
-  物体检测发展脉络
-  物体检测常用数据集
-  物体检测评价指标



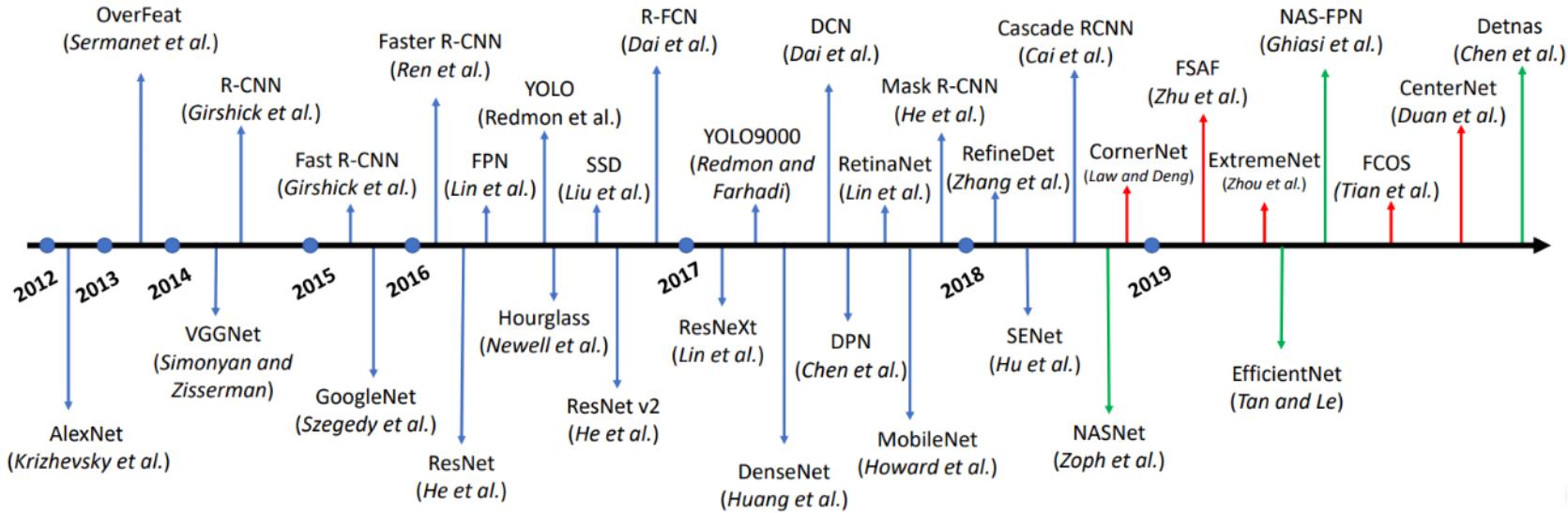
## 物体检测发展趋势

- 每年物体检测的论文数量的变化趋势



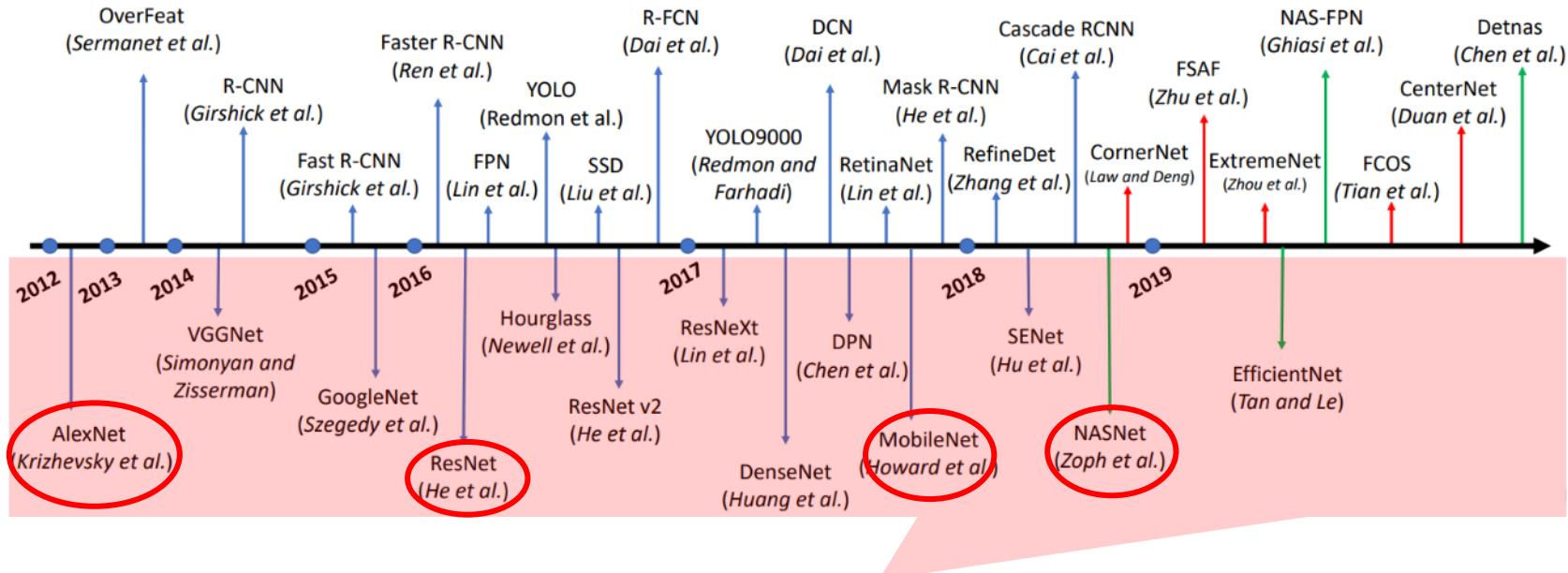


# 物体检测发展脉络





# 物体检测发展脉络

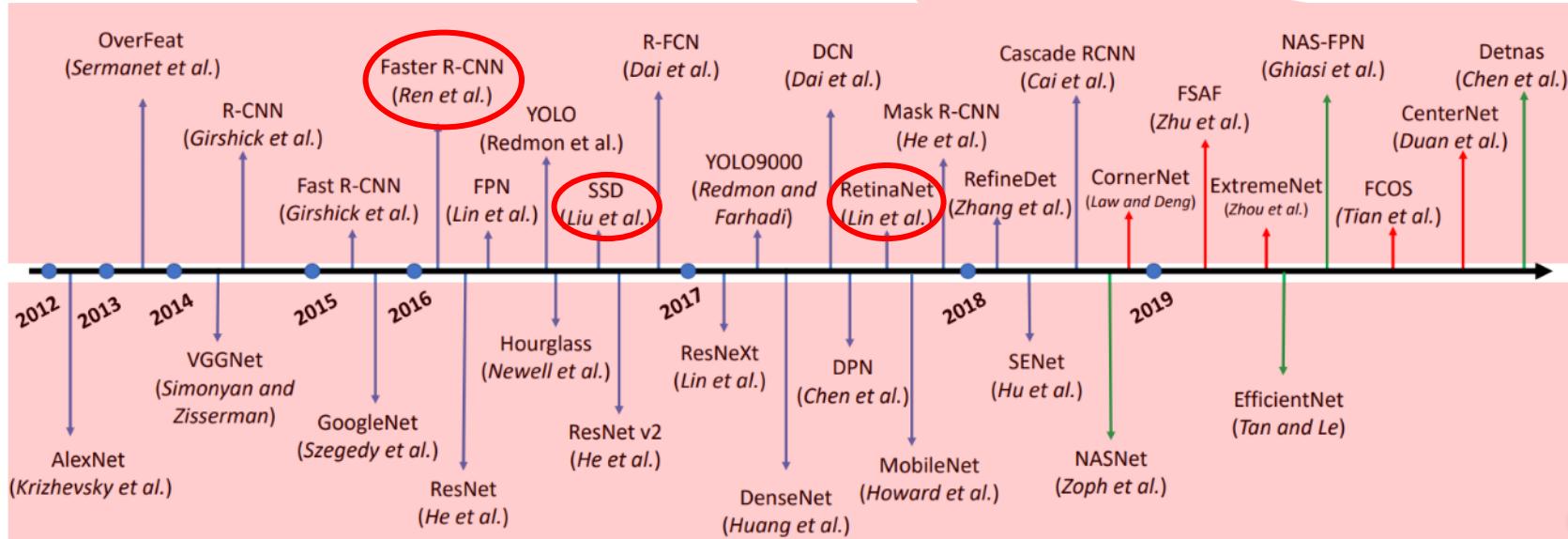


- 2012年后的深度学习时代，基础网络的发展脉络



# 物体检测发展脉络

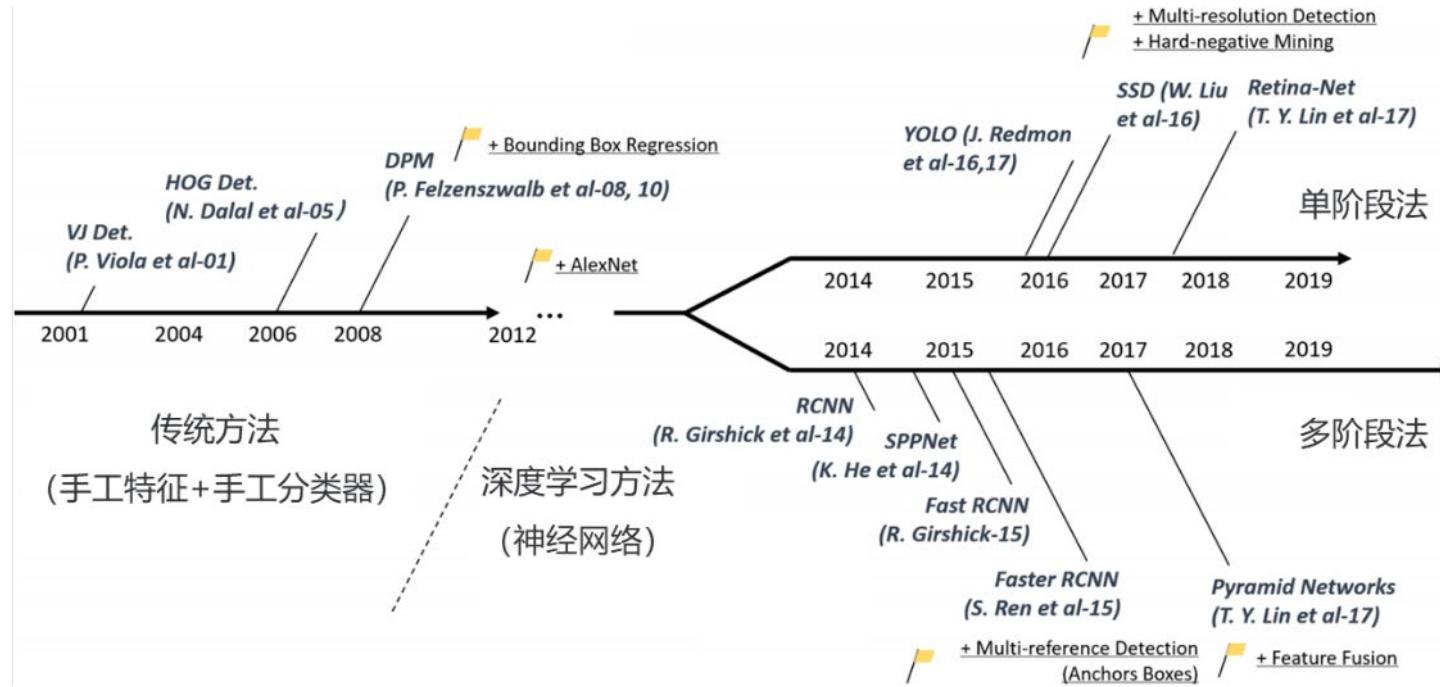
- 2012年后的深度学习时代，物体检测的发展脉络



- 2012年后的深度学习时代，基础网络的发展脉络

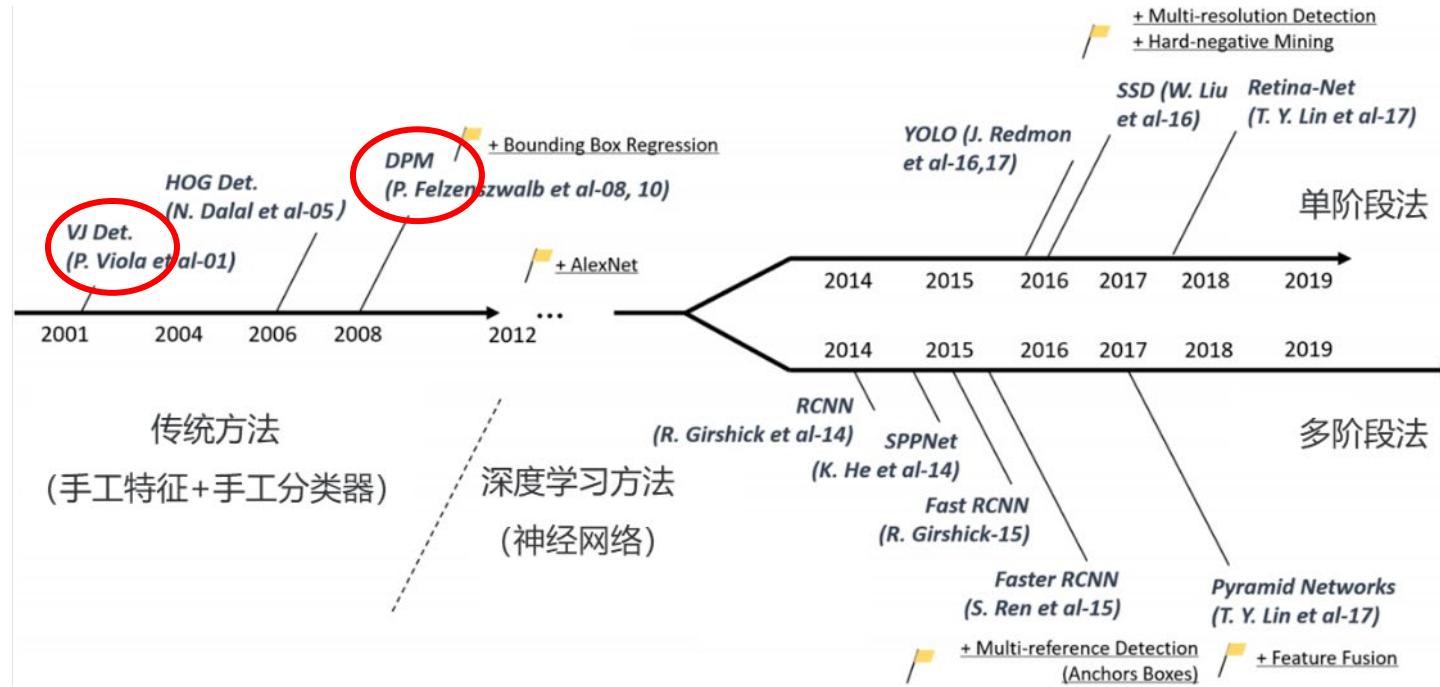


# 物体检测发展脉络



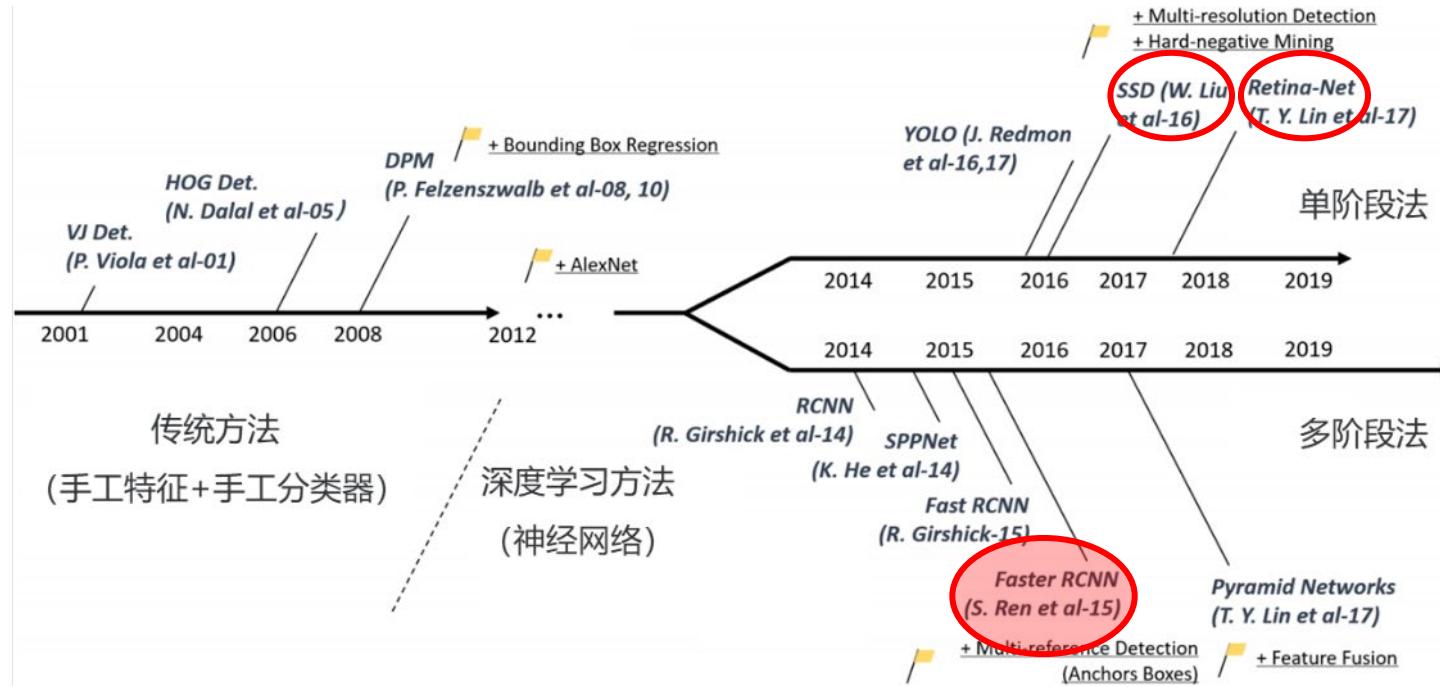


# 物体检测发展脉络





# 物体检测发展脉络



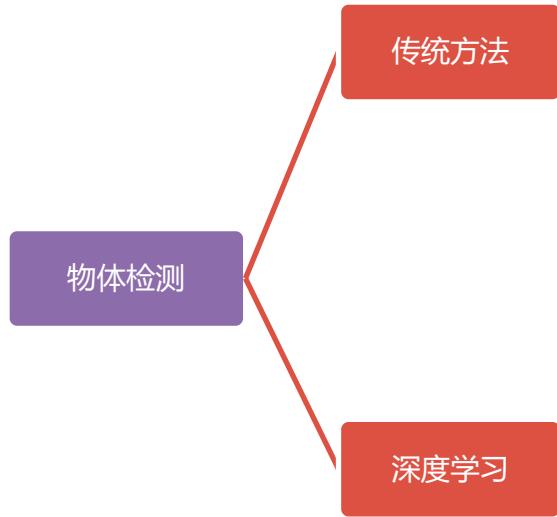


# 物体检测的派系

物体检测



# 物体检测的派系



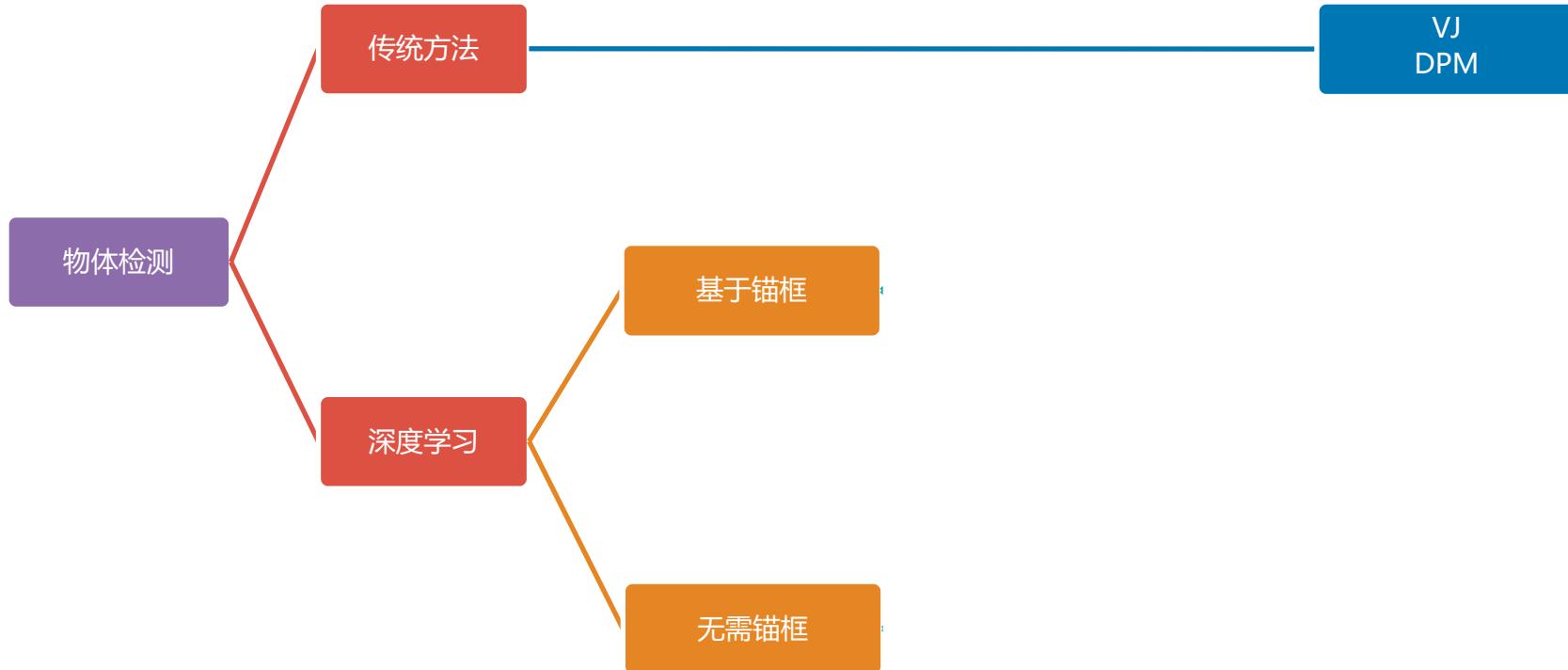


# 物体检测的派系



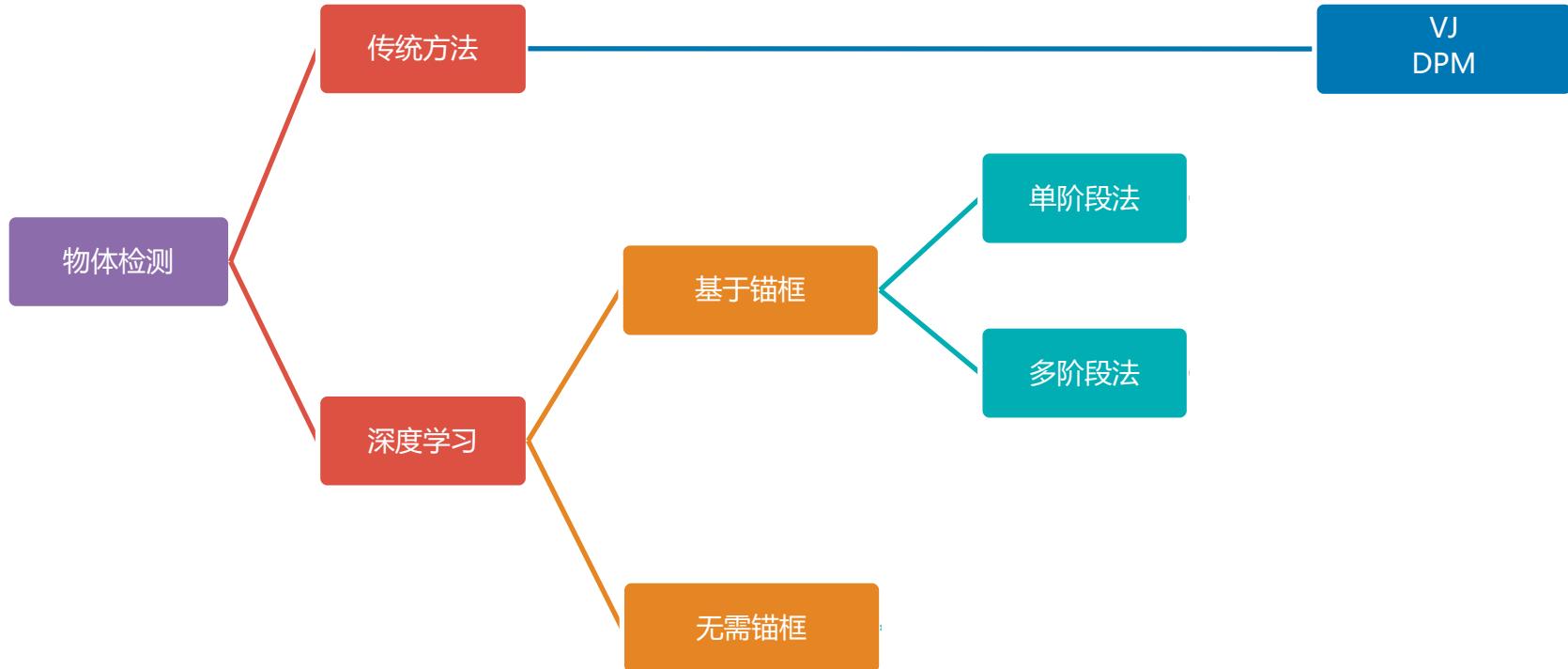


# 物体检测的派系



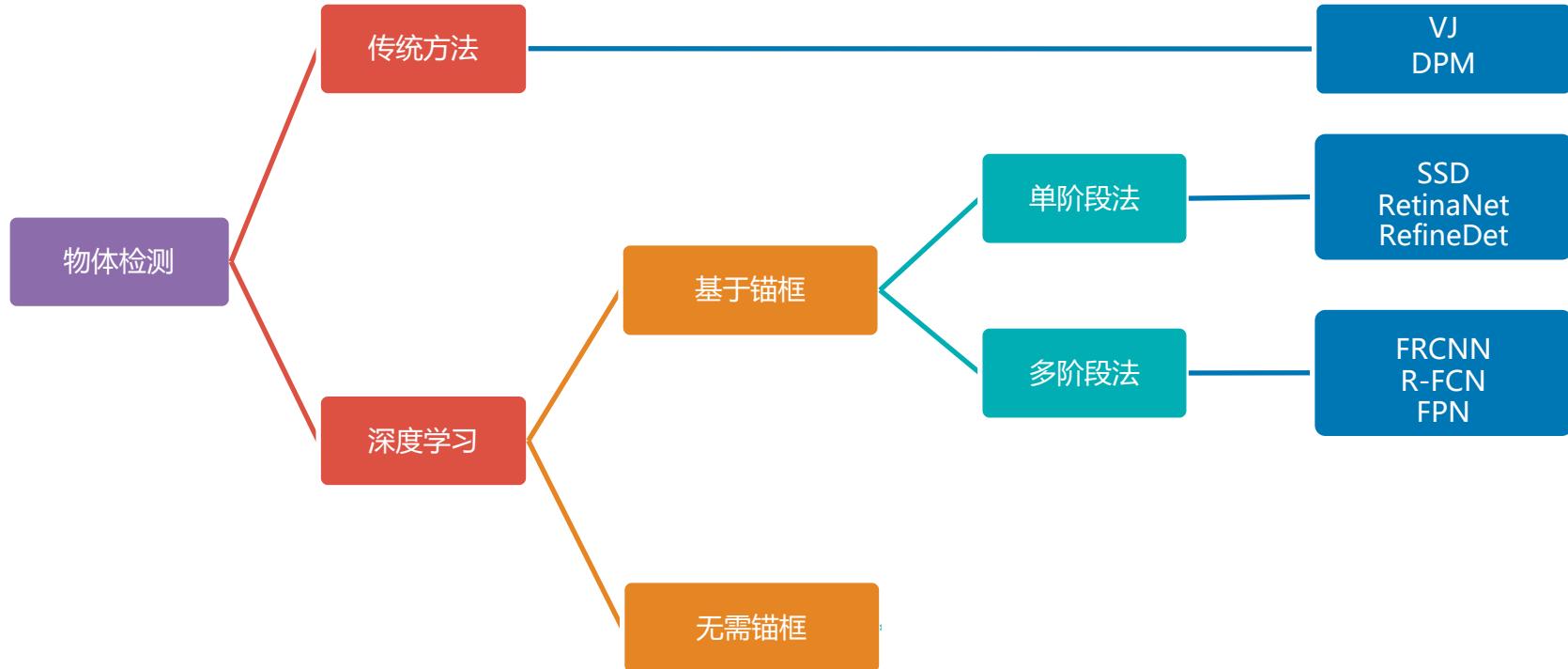


# 物体检测的派系



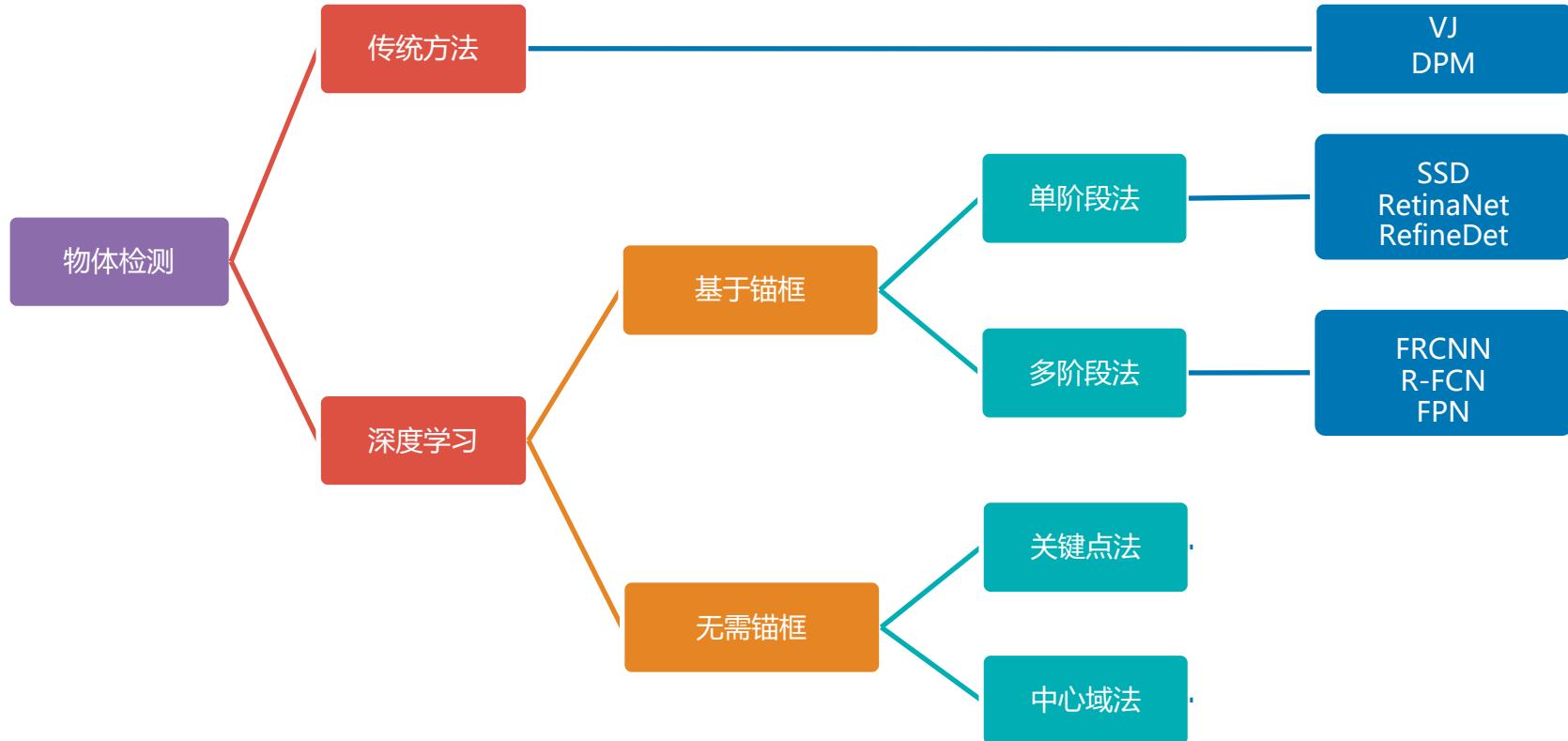


# 物体检测的派系



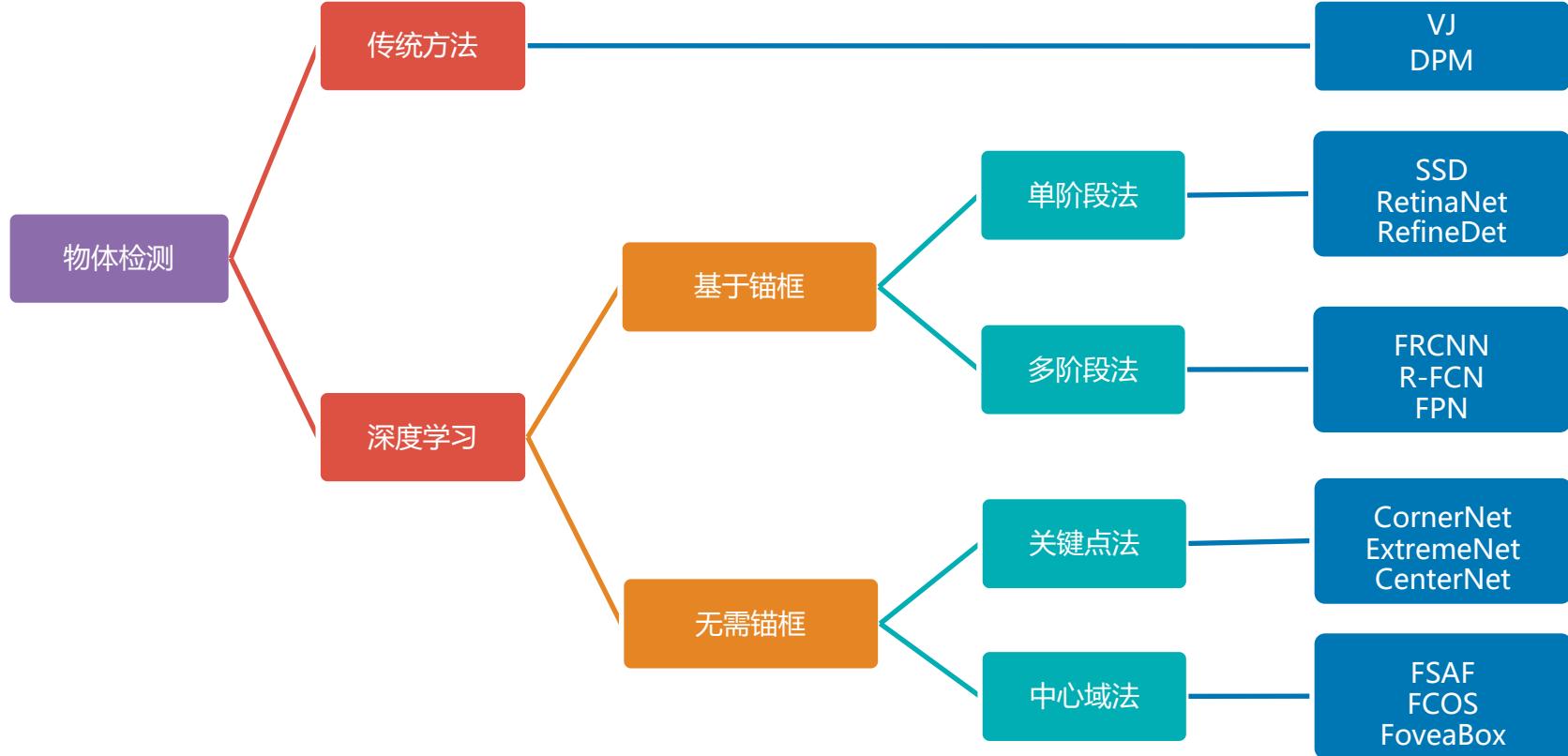


# 物体检测的派系



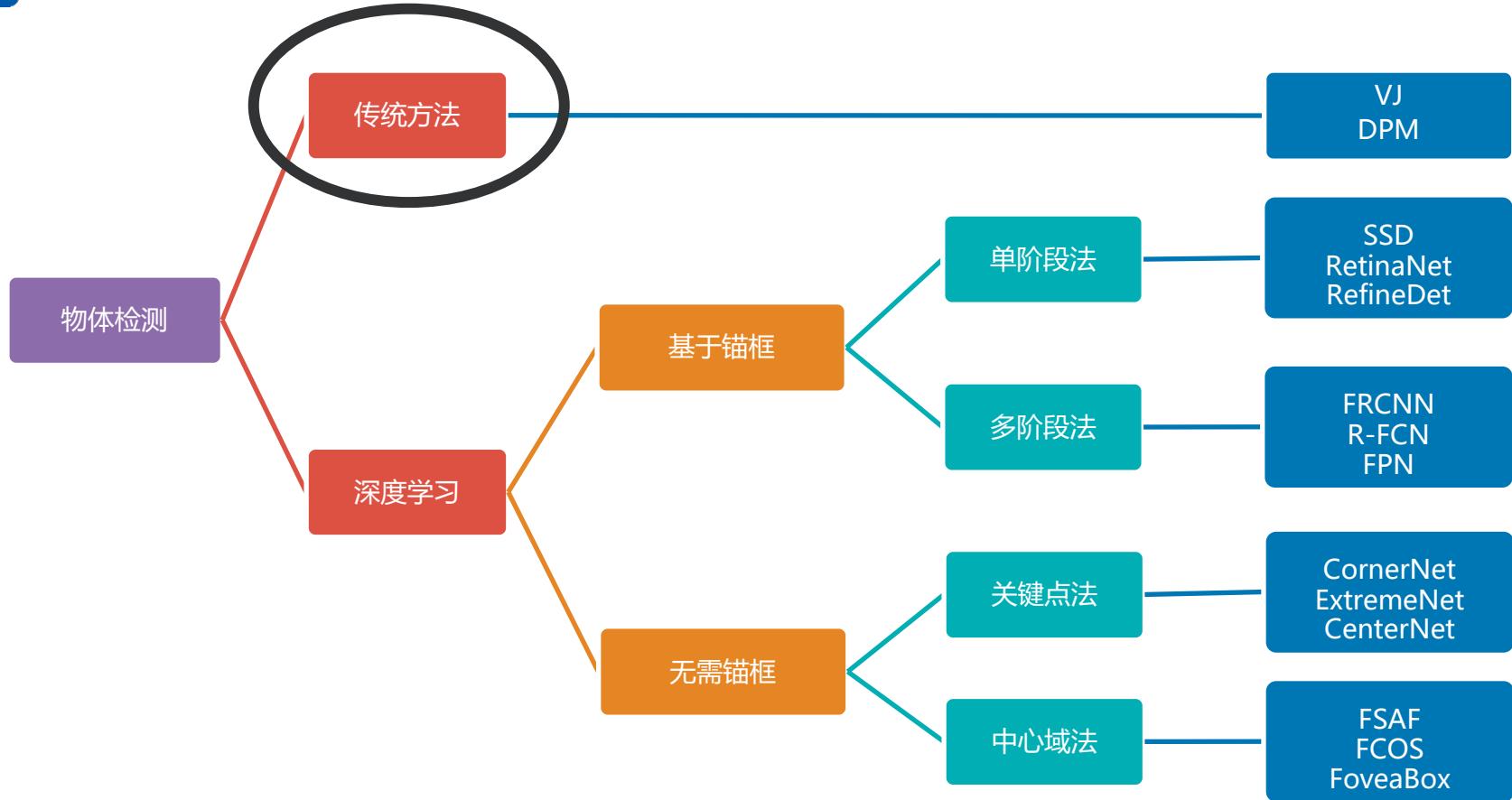


# 物体检测的派系





# 物体检测的派系

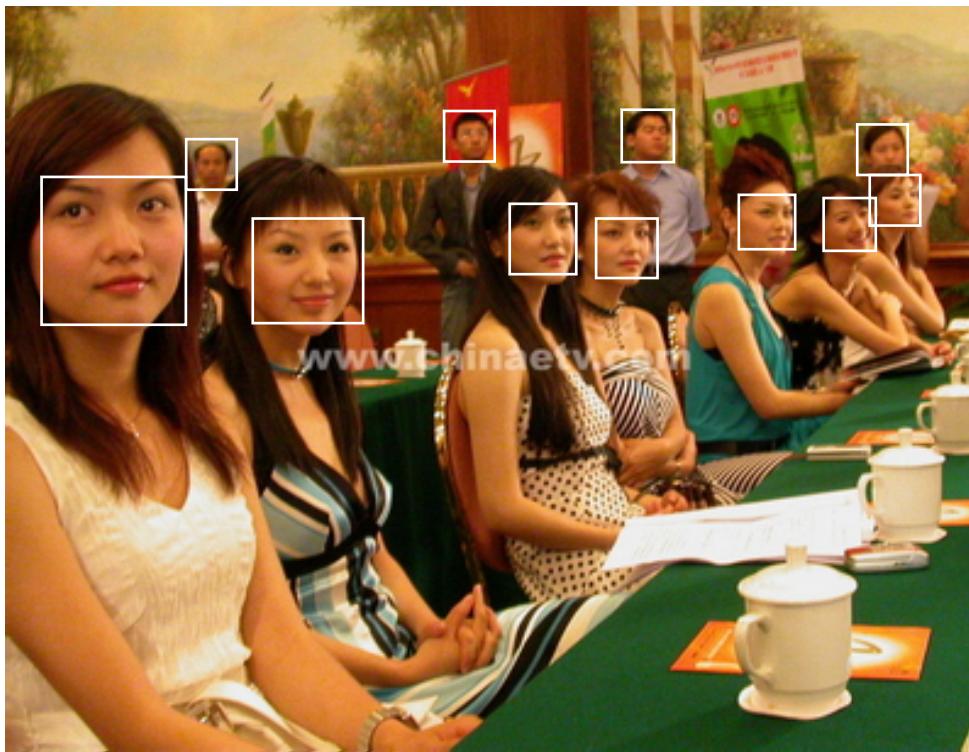




## 传统方法

- 利用手工特征+分类器，以滑窗方式在图像金字塔上遍历所有位置和大小，进行物体检测

滑窗方式  
遍历所有位置

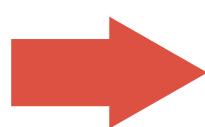


遍历不同大小  
图像金字塔



## 传统方法

- 利用**手工特征**+**分类器**，以滑窗方式在图像金字塔上遍历所有位置和大小，进行物体检测



手工特征

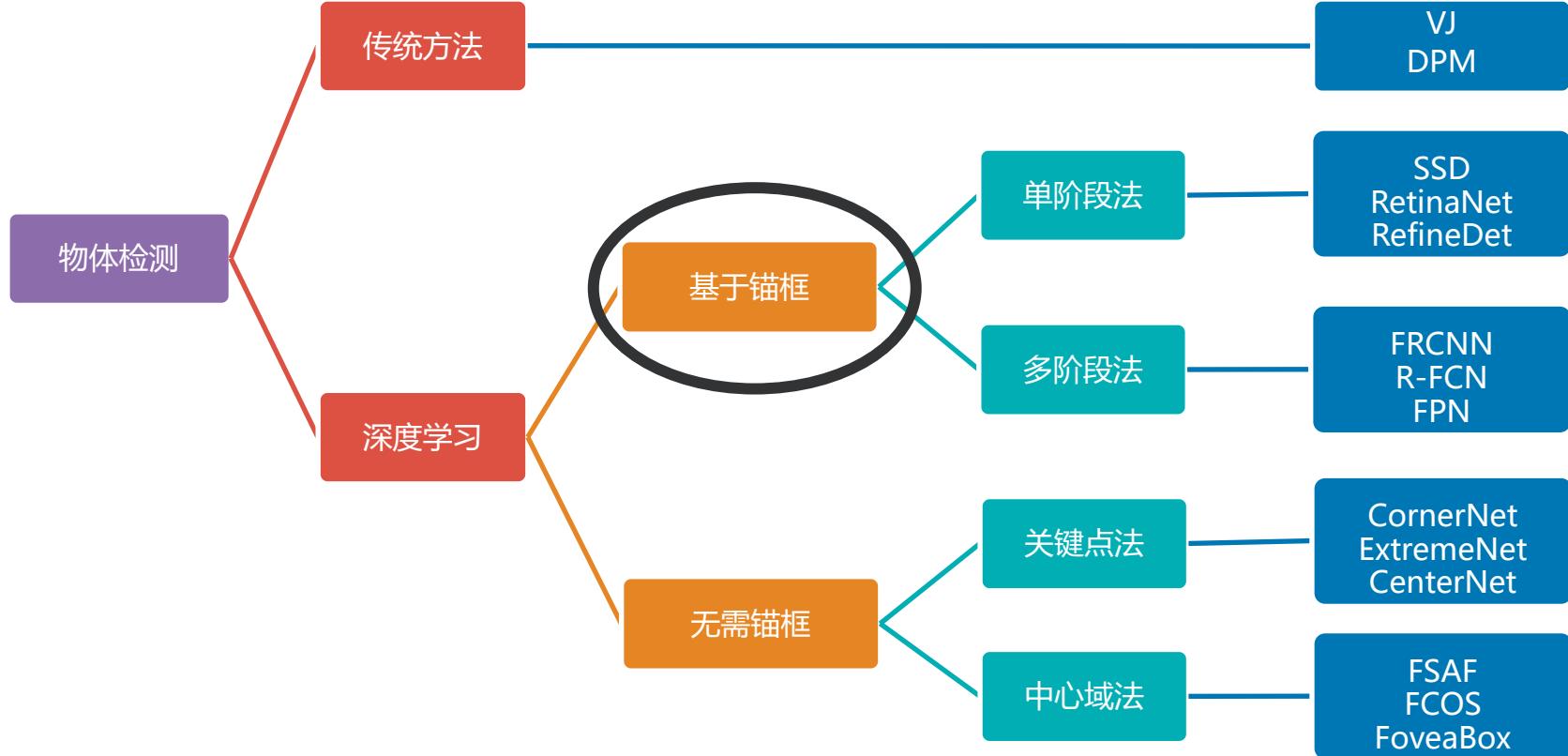


手工分类器

- |  |   |
|--|---|
| <ul style="list-style-type: none"><li>• Haar</li><li>• LBP (Local Binary Patterns)</li><li>• HOG (Histogram of Oriented Gradient)</li><li>• SIFT (Scale-Invariant Feature Transform)</li><li>• .....</li></ul> | <ul style="list-style-type: none"><li>• Adaboost</li><li>• SVM (Support Vector Machine)</li><li>• Decision Tree</li><li>• Random Forest</li><li>• .....</li></ul> |
|--|---|

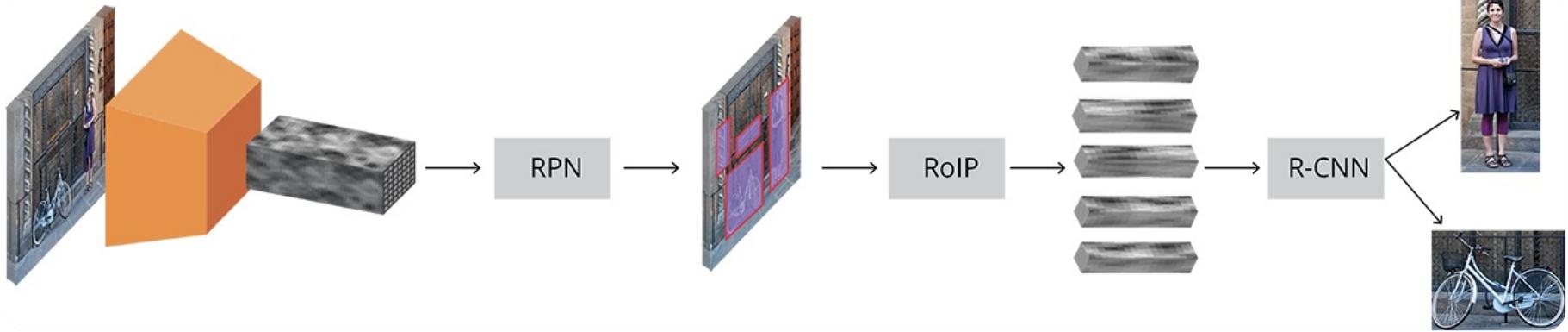


# 物体检测的派系



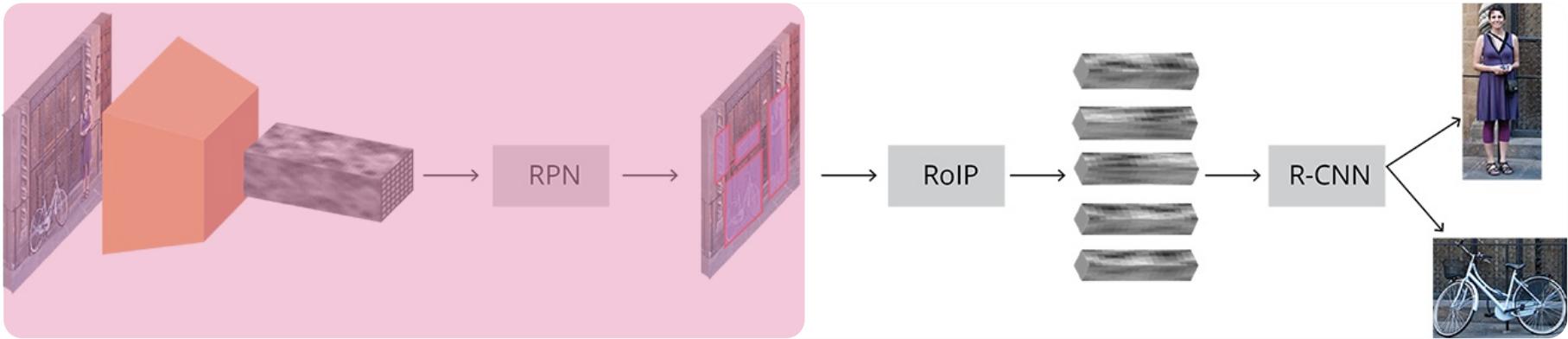


# 基于锚框的物体检测算法





# 基于锚框的物体检测算法

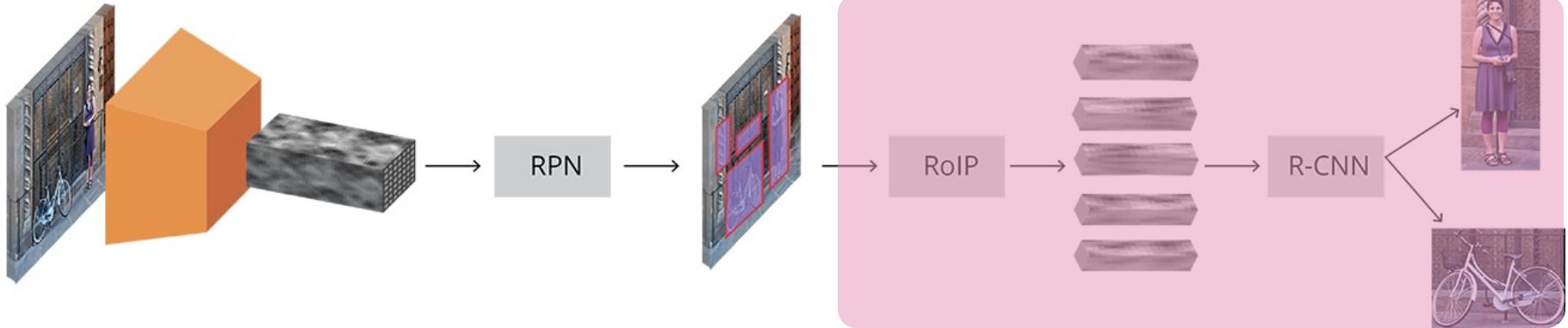


Faster R-CNN的第一阶段：

- ① 预设一系列不同大小和比例的anchors
- ② 整张图传入CNN提取特征
- ③ 区域生成网络 (Region Proposal Network, RPN) 对 anchors进行分类和回归，得到候选区域



# 基于锚框的物体检测算法



Faster R-CNN的第一阶段：

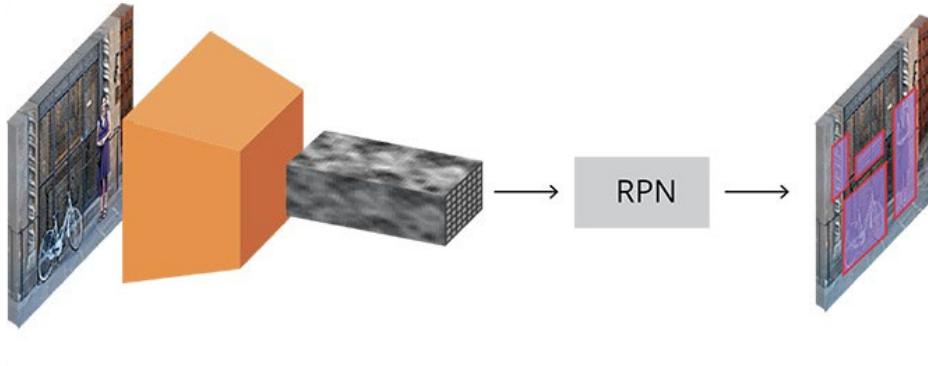
- ① 预设一系列不同大小和比例的anchors
- ② 整张图传入CNN提取特征
- ③ 区域生成网络 (Region Proposal Network, RPN) 对 anchors进行分类和回归，得到候选区域

Faster R-CNN的第二阶段：

- ① 利用RoIPooling (RoIPooling) 抠取每个候选区域的特征
- ② 把抠取特征的特征送入后续R-CNN网络
- ③ 对候选区域进一步分类和回归，得到最终的检测结果



# 基于锚框的物体检测算法

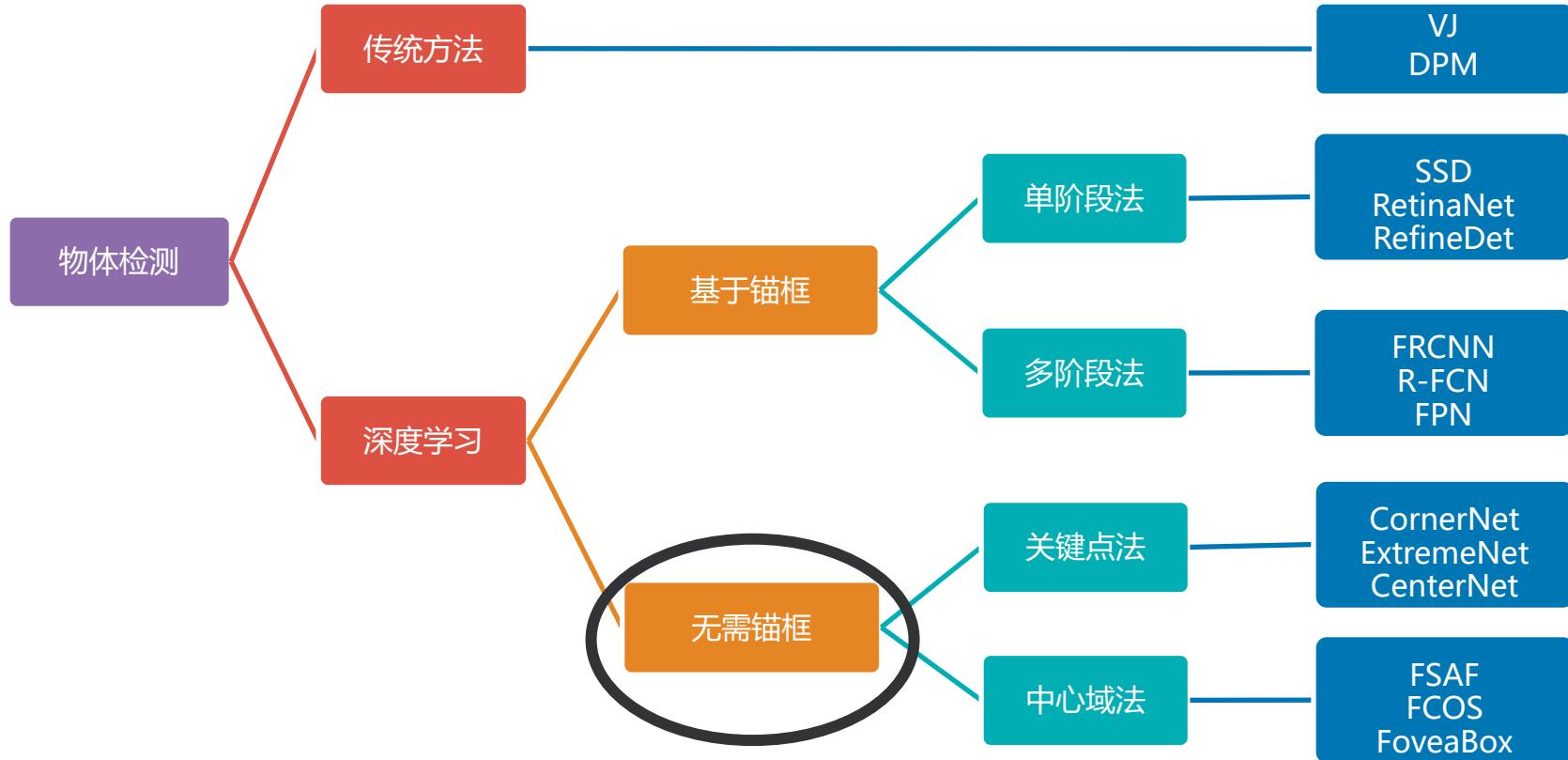


Faster R-CNN的第一阶段：

- ① 预设一系列不同大小和比例的anchors
- ② 整张图传入CNN提取特征
- ③ 区域生成网络 (Region Proposal Network, RPN) 对 anchors进行分类和回归，得到候选区域



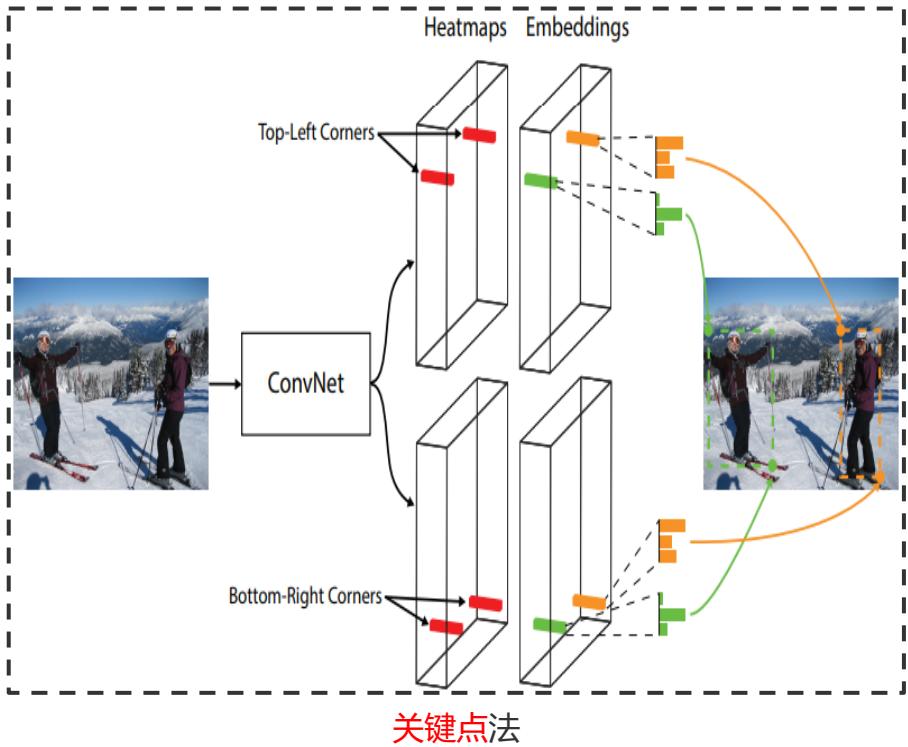
# 物体检测的派系



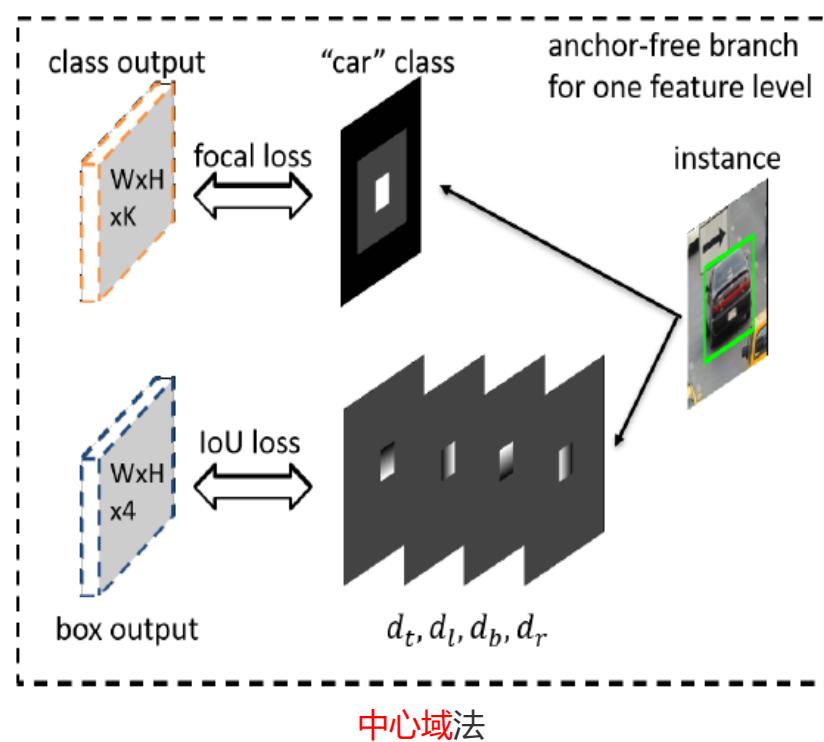


# 无需锚框的物体检测算法

CornerNet: Detecting Objects as Paired Keypoints

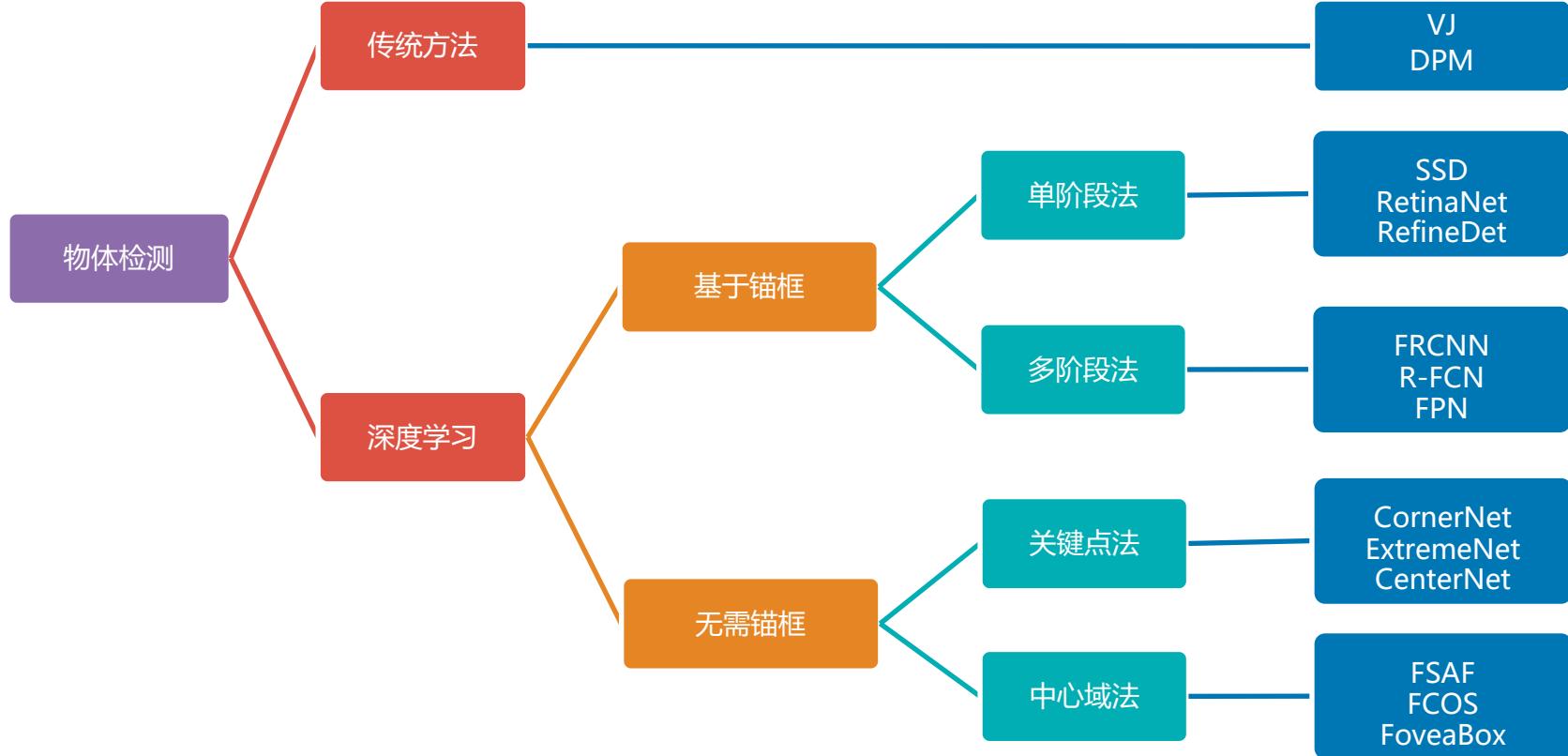


Feature Selective Anchor-Free Module for Object Detection





# 物体检测的派系





# 目录

-  物体检测研究背景
-  物体检测发展脉络
-  物体检测常用数据集
-  物体检测评价指标



# 物体检测常用数据集

## 通用物体检测数据集

- PASCAL VOC
- MS COCO
- OpenImages
- LVIS

## 人脸检测数据集

- AFW
- PASCAL FACE
- MALF
- MAFA
- FDDB
- WIDER FACE

## 行人检测数据集

- Caltech-USA
- CityPersons
- CrowdHuman
- WiderPerson
- EuroCityPersons



# 物体检测常用数据集

## 通用物体检测数据集

- PASCAL VOC
- MS COCO
- OpenImages
- LVIS

## 人脸检测数据集

- AFW
- PASCAL FACE
- MALF
- MAFA
- FDDB
- WIDER FACE

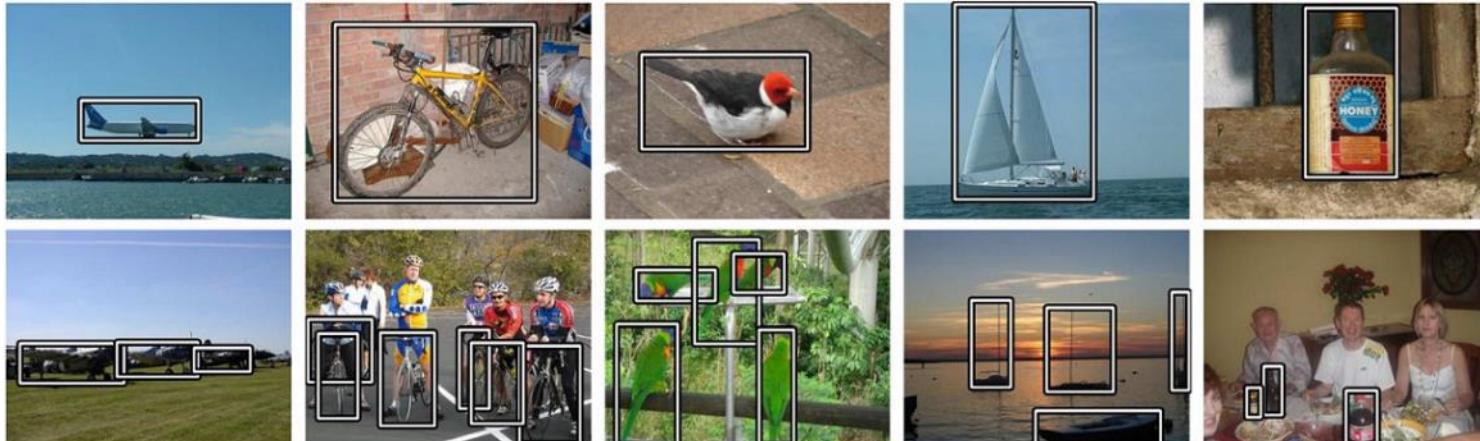
## 行人检测数据集

- Caltech-USA
- CityPersons
- CrowdHuman
- WiderPerson
- EuroCityPersons



## 通用物体检测数据集： PASCAL VOC

- PASCAL VOC 是一个从2005 年开始举办的著名竞赛，其中比赛内容就有物体检测
- 每年都会完善数据，图像数量越来越多，标注质量也越来越高，已于2012 年停止举办
- 常用数据集为PASCAL VOC 2007 和PASCAL VOC 2012，包含**20类**物体
- PASCAL VOC 2007： **9963** 张图像， **24640** 个标注
- PASCAL VOC 2012： **11530** 张图像， **27450** 个标注





# 通用物体检测数据集：PASCAL VOC

PASCAL VOC Challenge performance evaluation and download server

**PASCAL2**  
Pattern Analysis, Statistical Modelling and Computational Learning

Home | Leaderboard

**Detection Results: VOC2012 BETA**

Competition "comp4" (train on own data)

This leaderboard shows only those submissions that have been marked as public, and so the displayed rankings should not be considered as definitive.

- The highest scoring entry in each column is shown in **bold**.
- Clicking on the blue arrow symbol ( at the top of a column will order the submissions from high to low wrt performance on that column.

Average Precision (AP %)

	mean	aero plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dining table	dog	horse	motor bike	person	potted plant	sheep	sofa	train	tv/ monitor	
<b>ATLDEtV2 [?]</b>	<b>92.9</b>	<b>97.4</b>	<b>96.3</b>	<b>94.2</b>	<b>89.0</b>	<b>89.0</b>	<b>95.5</b>	<b>95.7</b>	<b>98.0</b>	<b>84.7</b>	<b>96.4</b>	<b>82.1</b>	<b>97.4</b>	<b>97.6</b>	<b>96.6</b>	<b>96.1</b>	<b>79.4</b>	<b>96.2</b>	<b>87.0</b>	<b>96.2</b>	<b>92.5</b>	
AccurateDET (ensemble) [?]	92.1	97.0	95.2	92.6	88.7	88.4	92.9	95.2	96.9	94.4	85.5	94.4	83.4	96.4	96.5	96.0	<b>96.5</b>	<b>82.0</b>	95.2	86.6	95.1	91.3
AIInnoDetection [?]	92.3	96.6	95.3	<b>94.4</b>	87.3	87.5	94.4	94.1	<b>98.4</b>	82.6	<b>96.5</b>	82.9	<b>97.9</b>	96.9	96.2	95.0	79.8	95.7	86.9	<b>96.5</b>	91.0	
AccurateDET [?]	91.3	96.6	95.1	91.5	87.2	87.0	92.2	94.0	96.5	83.4	94.1	80.0	96.1	96.4	95.8	95.7	79.7	95.1	85.1	94.6	90.1	
tencent_retail_ft:DFI [?]	91.2	96.1	94.9	92.7	85.8	88.4	93.5	94.9	97.1	80.0	94.8	78.8	96.7	96.4	96.0	95.9	79.0	95.9	83.1	95.0	88.5	
Sogou_MM_GCFE_RCNN(ensemble model) [?]	91.1	95.9	94.6	93.3	86.2	87.1	93.2	95.1	97.1	81.1	94.4	77.1	96.5	96.6	95.8	95.4	77.9	95.4	84.1	95.0	89.5	
Sogou_MM_GCFE_RCNN(single model) [?]	91.0	95.9	94.1	93.3	86.2	87.0	93.1	95.1	97.1	81.1	94.4	77.1	96.5	96.6	95.8	95.4	77.9	95.4	83.4	94.9	89.5	
FXRCNN (single model) [?]	90.7	96.4	95.1	92.0	84.3	87.1	92.8	94.4	97.4	80.7	93.5	76.0	96.7	96.7	95.6	95.5	78.3	94.6	83.3	95.4	88.0	
ATLDET [?]	90.7	96.0	94.9	91.8	85.2	87.6	93.0	94.5	97.5	80.7	93.8	75.6	96.6	96.2	95.8	95.5	78.3	95.2	82.5	94.8	89.2	
PACITYAIDetection [?]	89.8	95.3	93.6	91.1	85.4	83.9	91.6	93.3	96.8	80.1	95.5	74.3	96.3	95.7	94.4	94.7	77.5	94.1	82.7	94.2	86.4	
Ali_DCN_SSD_ENSEMBLE [?]	89.2	95.4	93.7	91.8	82.8	81.7	92.4	93.4	97.6	75.7	94.1	74.2	96.4	95.1	94.2	93.3	72.5	94.1	82.8	94.6	87.7	
CM-CV&AR: DET [?]	89.1	95.7	94.4	92.0	81.1	82.9	93.8	90.0	97.1	74.6	95.4	70.4	96.7	96.2	95.3	93.4	73.7	94.8	81.1	96.0	88.2	
VIM_SSD(COCO+07++12, single model, one-stage) [?]	89.0	96.0	93.0	90.3	83.4	80.6	91.9	94.4	96.2	77.5	93.3	75.1	<b>95.2</b>	95.1	94.2	93.6	72.0	93.6	82.7	94.5	86.6	
FOCAI_DRFCN(VOC+COCO, single model) [?]	88.8	95.0	93.3	91.8	82.9	81.9	91.6	93.0	97.1	76.7	92.5	71.7	96.2	94.9	94.2	93.7	75.3	93.3	80.0	94.7	85.4	
R4D_faster_rcnn [?]	88.6	94.6	92.3	91.3	82.3	79.4	91.8	91.8	97.4	76.6	93.6	75.3	97.0	94.6	93.5	92.6	75.1	92.0	80.9	94.4	86.5	



## 通用物体检测数据集：MS COCO

- MS COCO数据集每年也会举办非常权威的竞赛，其中包含物体检测赛道
- 国内外各个研究机构以及各大公司都会投入大量的人力和资源参与其中，例如旷视
- 包含**80类**物体，**11.5万**多张训练集图像，**5千**张验证集图像，**2万**多张测试集图像
- 每张图像平均有3.5个类别和7.7个物体，仅有10%的图像包含一个物体。
- 测试集没有公布标注，用户需要提交检测结果到指定服务器去进行评测才能得到性能指标
- 通用物体检测算法，最常用学术界数据集





# 通用物体检测数据集：MS COCO

## Competition



### COCO Detection Challenge (Bounding Box)

Organized by richardaecn - Current server time: April 26, 2020, 6:14 p.m. UTC

Previous      ▶ Current      End

[test-challenge2019 \(bbox\)](#)      Aug. 11, 2019, midnight UTC      [test-dev2019 \(bbox\)](#)      Aug. 11, 2019, midnight UTC      [Competition Ends](#)      Oct. 1, 2017, 6:59 a.m. UTC

Learn the Details      Phases      Participate      Results      Forums       Team 

[Overview](#)      Evaluation      Terms and Conditions

#### COCO Detection Challenge (Bounding Box)



NOTE: Detection with bounding boxes outputs will NOT be featured at the COCO 2019 challenge. More details can be found [here](#). For detection with segmentation masks outputs, please refer to <https://competitions.codalab.org/competitions/20796>.



## 通用物体检测数据集：[LVIS](#)

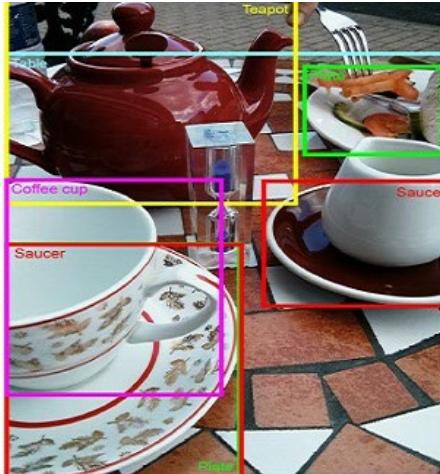
- 16.4万张图像，1000+类物体，约200万个标注
- 亮点是通过对超过1000+个类别进行标注，形成了贴近实际情况的长尾分布





# 通用物体检测数据集：OpenImages

- 整个数据集包含约900万张图像，横跨约6000个类别
- 物体检测：共包含**190万**张图像，**600**个类别，**1600**万个标注
- 目前OpenImages 已经公布到了第六版，还在完善中
- OpenImages 的完成，将会促进众多领域的进一步发展





# 通用物体检测数据集

数据库	图片数量	标注数量	类别数量	难度指数
PASCAL VOC 2007	9963	24640	20	★★
PASCAL VOC 2012	11540	27450	20	★★★
MS COCO	约14万	5171	80	★★★★
LVIS	16.4 万	200万	1000+	★★★★★
OpenImages	190万	1600万	600	★★★★★



# 物体检测常用数据集

## 通用物体检测数据集

- PASCAL VOC
- MS COCO
- OpenImages
- LVIS

## 人脸检测数据集

- AFW
- PASCAL FACE
- MALF
- MAFA
- FDDB
- WIDER FACE

## 行人检测数据集

- Caltech-USA
- CityPersons
- CrowdHuman
- WiderPerson
- EuroCityPersons



## 人脸检测数据集：AFW

- **205张图片、473张人脸**
- 仅用于测试的数据集





# 人脸检测数据集：PASCAL FACE

- **851**张图片、**1335**张人脸
- 仅用于测试的数据集





## 人脸检测数据集：MALE

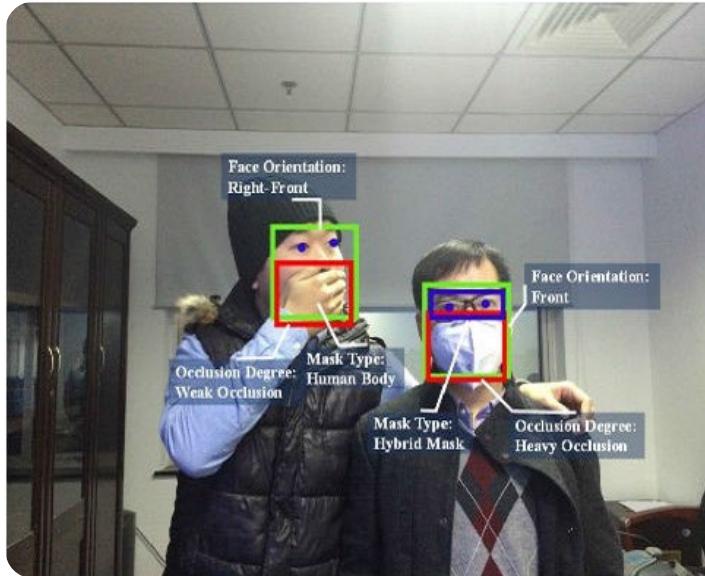
- MALF测试集：**5250**张图片、**11931**张人脸
- 5000张测试集，250张验证集





# 人脸检测数据集：MAFA

- 30881张图，35806个人脸，遮挡人脸检测
- 训练集25876，测试集4935





## 人脸检测数据集：FDDB

- **2845**张图片、**5171**张人脸
- 较流行的评测数据库，有标准的评测代码





# 人脸检测数据集：WIDER FACE

- **32203**张图片、**393703**张人脸
- 训练集12880、验证集3226、测试集16097
- 人脸检测最大最权威的数据集，难度非常大

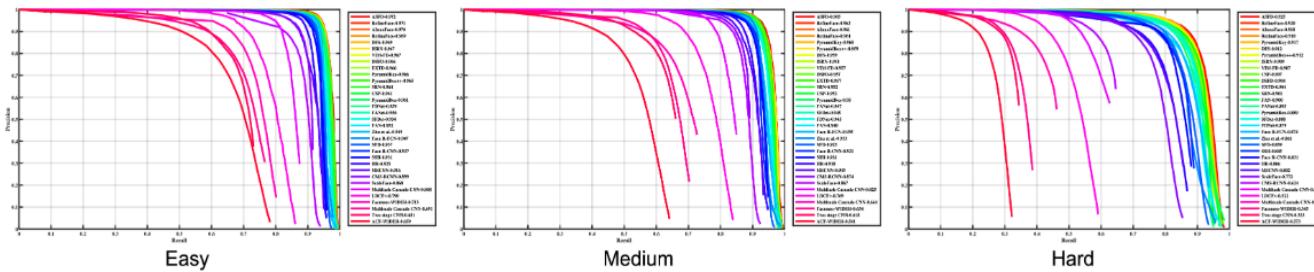




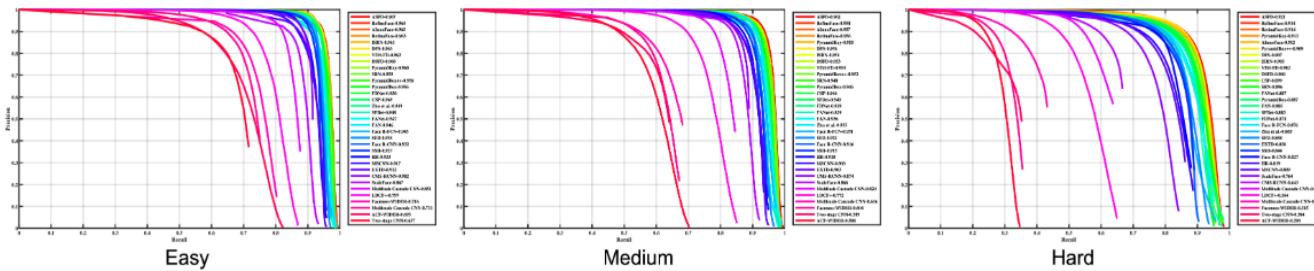
## 人脸检测数据集：[WIDER FACE](#)

- 官方排行榜：各大公司和科研机构都有参与刷榜

## Performance curves for the *validation set*



## Performance curves for the *test set*





# 人脸检测数据集

数据库	图片数量	人脸数量	测试集	验证集	训练集	难度指数
AFW	205	473	√	×	×	☆
PASCAL FACE	851	1335	√	×	×	☆☆
Fddb	2845	5171	√	×	×	☆☆
MALF	5250	11931	5000	250	×	☆☆☆
MAFA	30881	35806	4935	×	25876	☆☆☆
WIDER FACE	32203	393703	16097	3226	12880	☆☆☆☆☆



# 物体检测常用数据集

## 通用物体检测数据集

- PASCAL VOC
- MS COCO
- OpenImages
- LVIS

## 人脸检测数据集

- AFW
- PASCAL FACE
- MALF
- MAFA
- FDDB
- WIDER FACE

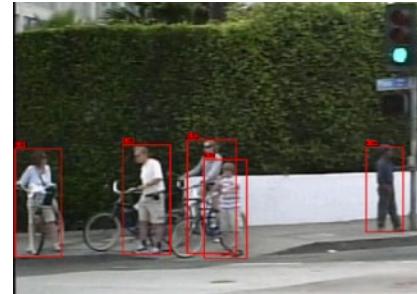
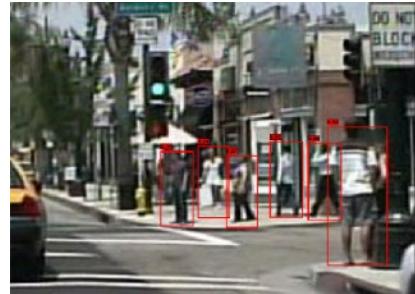
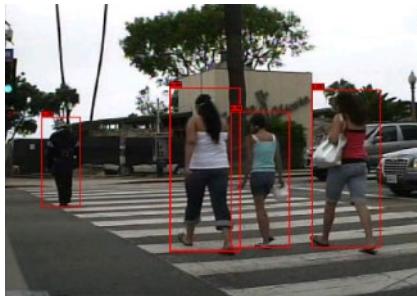
## 行人检测数据集

- Caltech-USA
- CityPersons
- CrowdHuman
- WiderPerson
- EuroCityPersons



## 行人检测数据集：Caltech-USA

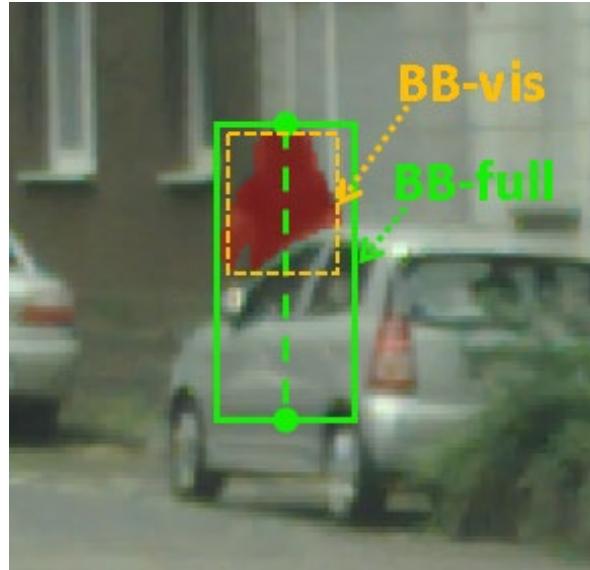
- 10小时30帧的VGA视频
- **42782**张训练集，**4024**张测试集





# 行人检测数据集: CityPersons

- **5000**张图: 2975张训练, 500张验证, 1525张测试
- **35000+**行人 (完整、可见区域), 13000+忽略区域

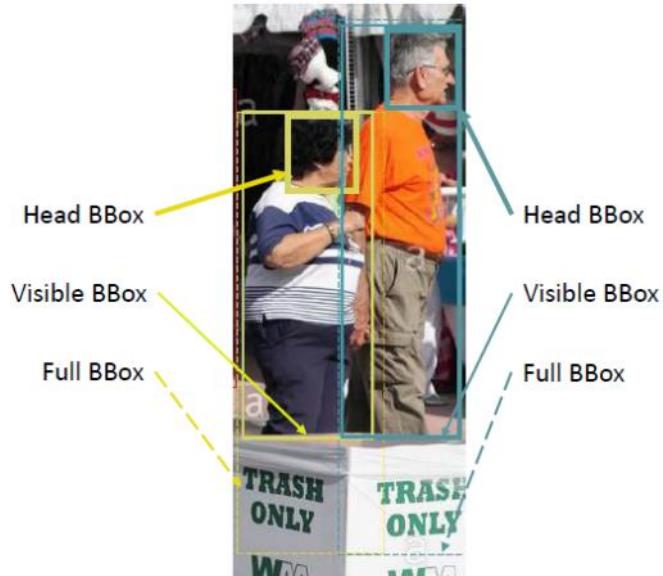


- Pedestrian (walking, running or standing up)
- rider (riding bicycles or motorbikes)
- sitting person
- other person (with unusual postures)
- Ignore region



# 行人检测数据集：CrowdHuman

- **24370**张图：训练集15000，验证集4370，测试集5000
- **47万**行人标注，平均每张图23个行人
- Head、human visible-region、human full-body





# 行人检测数据集：WiderPerson

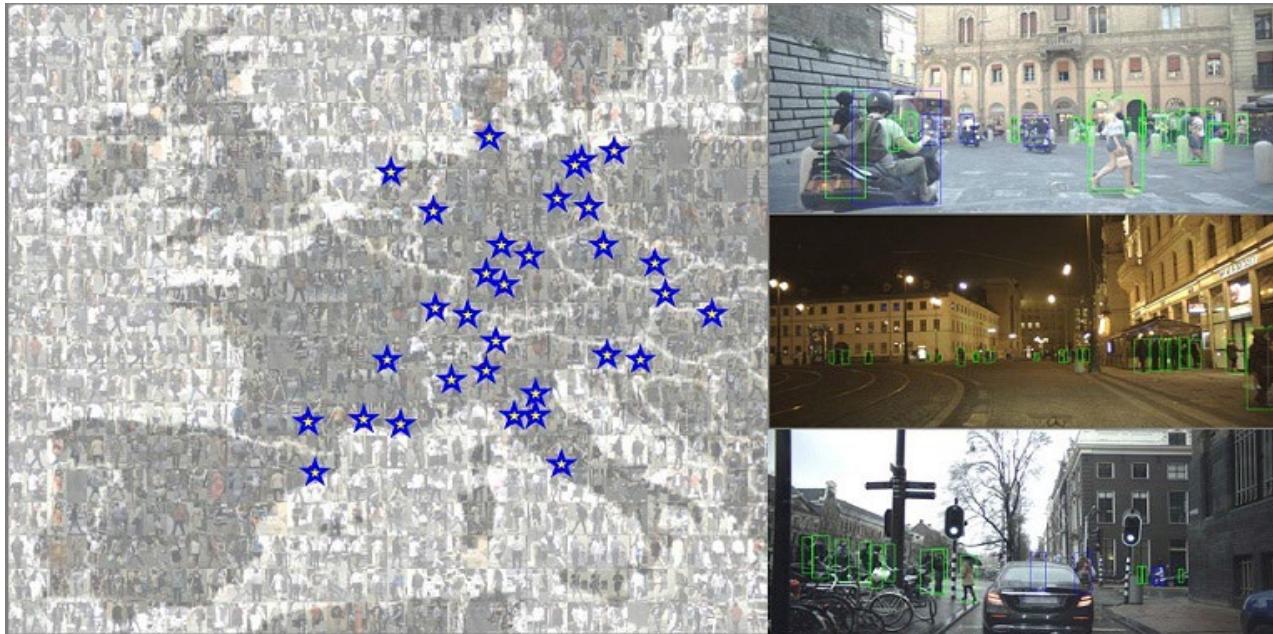
- 13382张图，约40万个标注，丰富的场景
- 训练集8000张、验证集1000张、测试集4382张





## 行人检测数据集：EuroCityPersons

- CityPersons 强化版：**47325** 张图像，**23.8万**+个标注（行人、骑行者以及忽略区域）
- 该数据不仅包含白天的图像，还有夜晚的图像





# 行人检测数据集

数据库	图片数量				标注类型			
	总共	训练集	测试集	验证集	可见行人	完整行人	人头	固定比例
Caltech-USA	46806	42782	4024	0	✗	✓	✗	✓
CityPersons	5000	2975	1525	500	✓	✓	✗	✓
CrowdHuman	24370	15000	5000	4370	✗	✗	✗	✗
WiderPerson	13382	8000	4382	1000	✗	✓	✗	✓
EuroCityPersons	47325	28114	14175	5036	✓	✓	✗	✗



# 目录

-  物体检测研究背景
-  物体检测发展脉络
-  物体检测常用数据集
-  物体检测评价指标



# 物体检测评价指标

检测精度

检测速度

召回率 (Recall Rate)

前传耗时 (ms)

平均精度均值 (mAP)

每秒帧数 (FPS)

平均对数漏检率 ( $MR^{-2}$ )

浮点运算量 (FLOPs)





# 物体检测评价指标

检测精度

召回率 (Recall Rate)

平均精度均值 (mAP)

平均对数漏检率 ( $MR^{-2}$ )

检测速度

前传耗时 (ms)

每秒帧数 (FPS)

浮点运算量 (FLOPs)





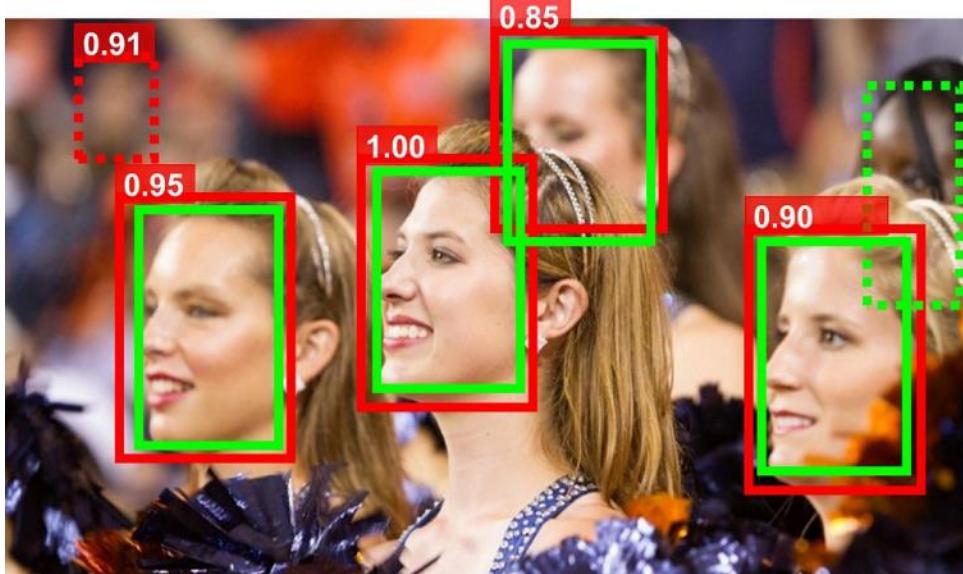
## 检测精度评价指标

- 通用物体检测（20类、80类、1000类），人脸检测或行人检测（1类）
- 检测精度评价指标：每类先单独计算评价指标，然后多类之间求平均
- 为了更清晰地描述检测精度的评价指标，在这里以单类物体检测为例来说明



## 检测精度评价指标

- 通用物体检测（20类、80类、1000类），人脸检测或行人检测（1类）
- 检测精度评价指标：每类先单独计算评价指标，然后多类之间求平均
- 为了更清晰地描述检测精度的评价指标，在这里以单类物体检测为例来说明

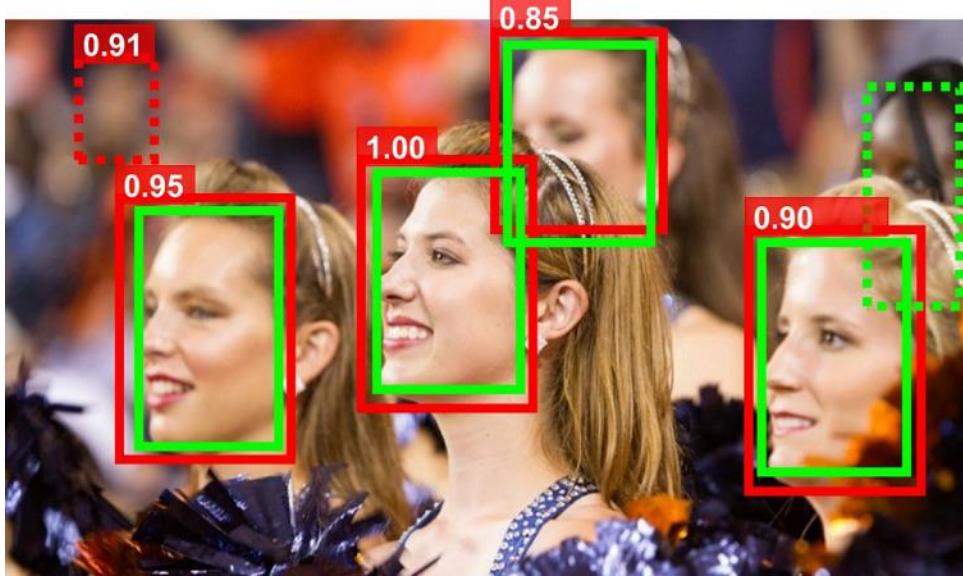


- 绿色实线或虚线框：人脸的真实标注
- 红色实线或虚线框：输出的检测结果
- 红色数字：检测结果为人脸的概率

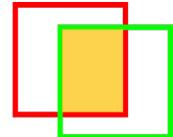


## 检测精度评价指标

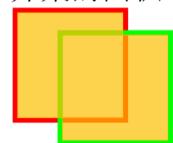
- 通用物体检测（20类、80类、1000类），人脸检测或行人检测（1类）
- 检测精度评价指标：每类先单独计算评价指标，然后多类之间求平均
- 为了更清晰地描述检测精度的评价指标，在这里以单类物体检测为例来说明



- 绿色实线或虚线框：人脸的真实标注
- 红色实线或虚线框：输出的检测结果
- 红色数字：检测结果为人脸的概率



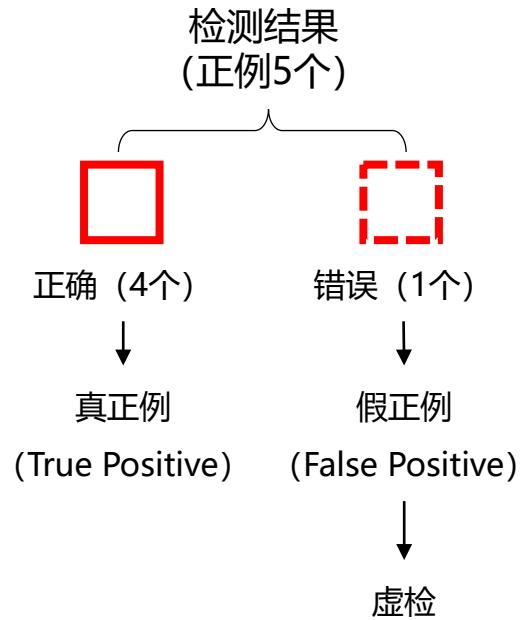
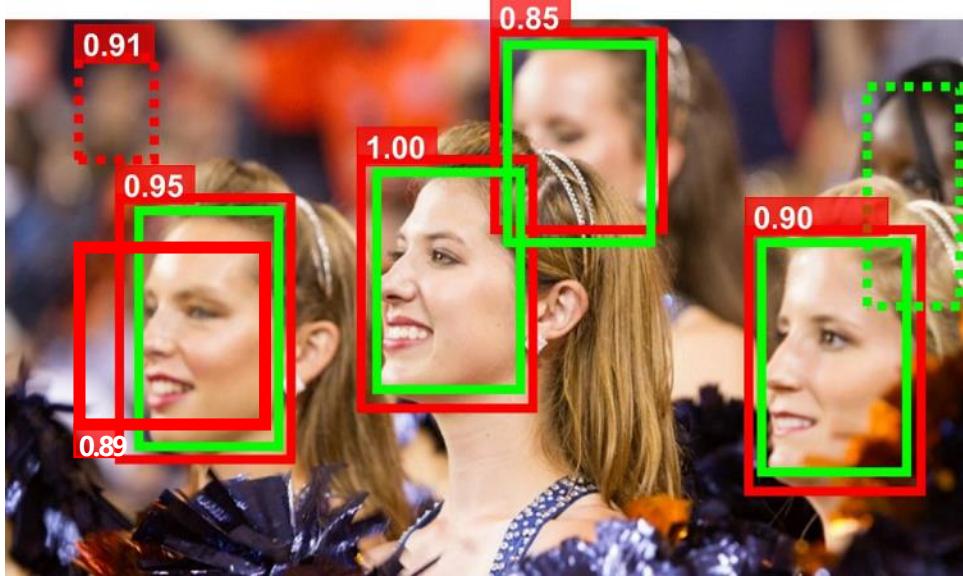
$$\text{交除并重叠比} = \frac{\text{交集的面积}}{\text{并集的面积}} \geq 0.5$$





## 检测精度评价指标

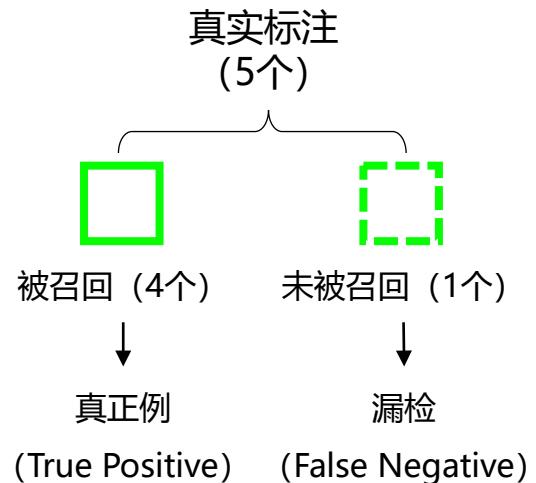
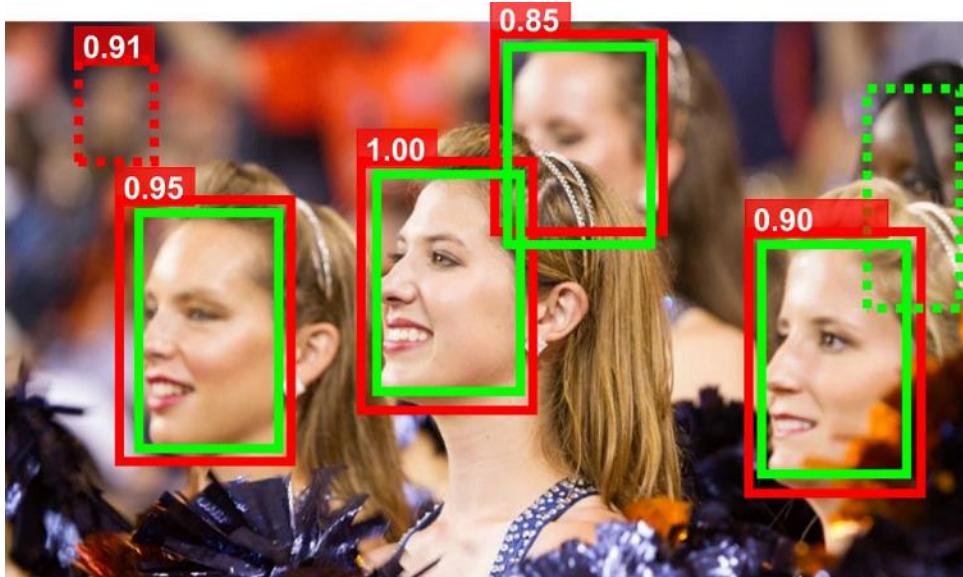
- 通用物体检测（20类、80类、1000类），人脸检测或行人检测（1类）
- 检测精度评价指标：每类先单独计算评价指标，然后多类之间求平均
- 为了更清晰地描述检测精度的评价指标，在这里以单类物体检测为例来说明





## 检测精度评价指标

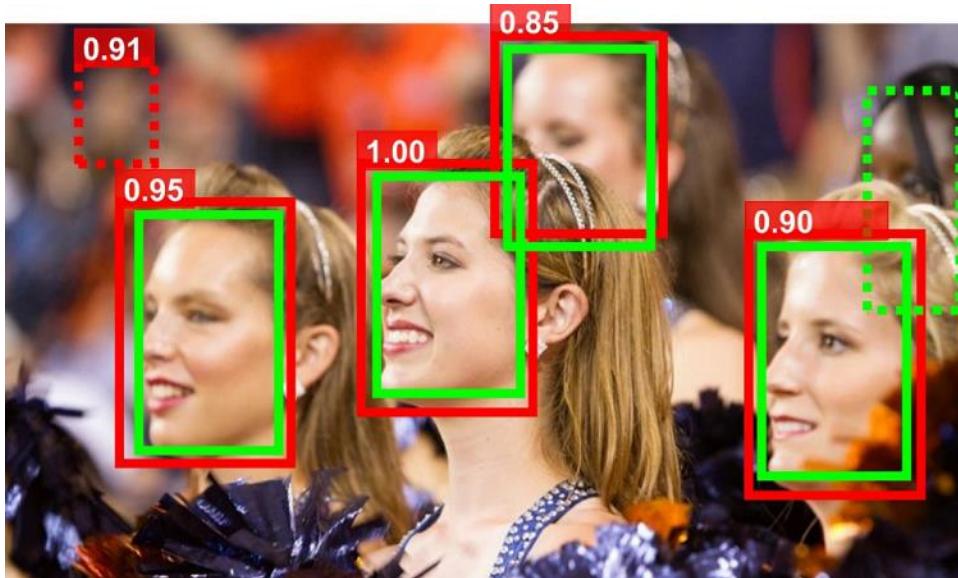
- 通用物体检测（20类、80类、1000类），人脸检测或行人检测（1类）
- 检测精度评价指标：每类先单独计算评价指标，然后多类之间求平均
- 为了更清晰地描述检测精度的评价指标，在这里以单类物体检测为例来说明





# 召回率 (Recall Rate)

- 通用物体检测（20类、80类、1000类），人脸检测或行人检测（1类）
- 检测精度评价指标：每类先单独计算评价指标，然后多类之间求平均
- 为了更清晰地描述检测精度的评价指标，在这里以单类物体检测为例来说明



检测结果(□□) = 5个 真实标注 (□□) = 5个

被召回的真实标注(□) = 真正例 (□) = 4个

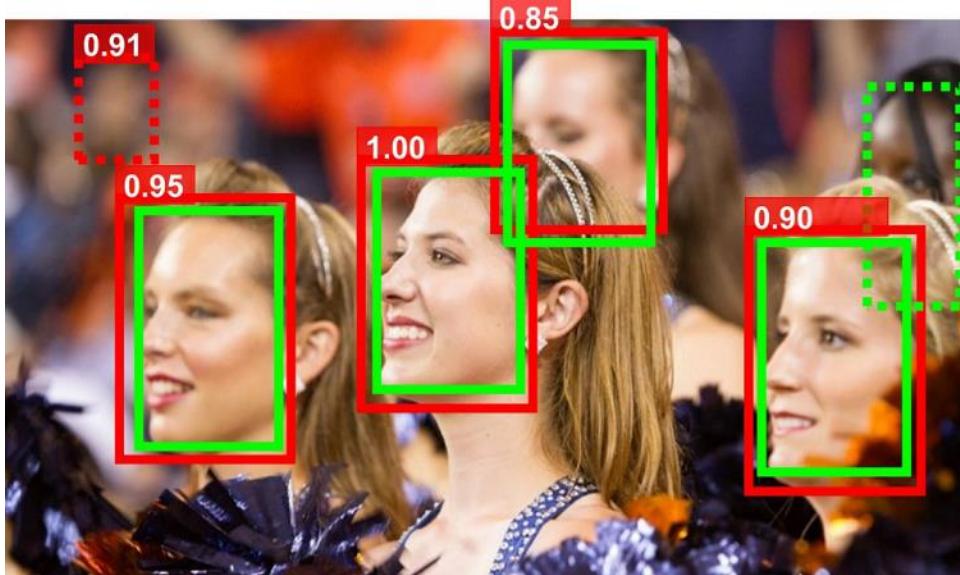
虚检(□) = 1个 漏检 (□) = 1个

$$\text{召回率} = \frac{\text{被召回的真实标注的数量}}{\text{真实标注的数量}} \\ = \frac{4}{5} = 80\%$$



## 精度均值 (Precision)

- 通用物体检测（20类、80类、1000类），人脸检测或行人检测（1类）
- 检测精度评价指标：每类先单独计算评价指标，然后多类之间求平均
- 为了更清晰地描述检测精度的评价指标，在这里以单类物体检测为例来说明



检测结果( $\boxed{\square}$ ) = 5个 真实标注 ( $\boxed{\square}_{\text{真实}}$ ) = 5个

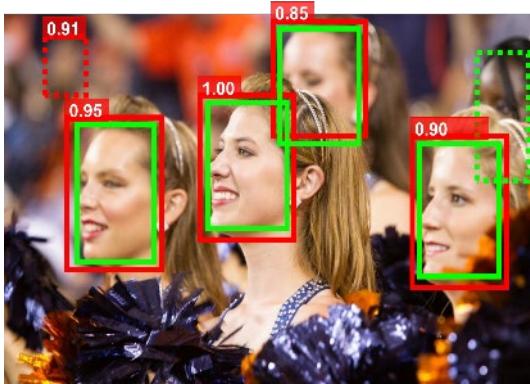
被召回的真实标注( $\boxed{\square}$ ) = 真正例 ( $\boxed{\square}$ ) = 4个

虚检( $\boxed{\square}$ ) = 1个 漏检 ( $\boxed{\square}_{\text{漏检}}$ ) = 1个

$$\text{精度} = \frac{\text{真正例的数量}}{\text{检测结果的数量}} = \frac{4}{5} = 80\%$$



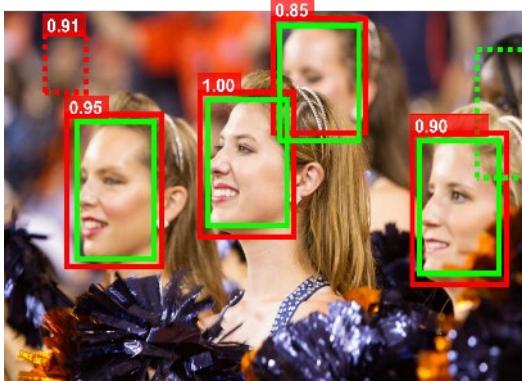
## 平均精度均值 (mAP)



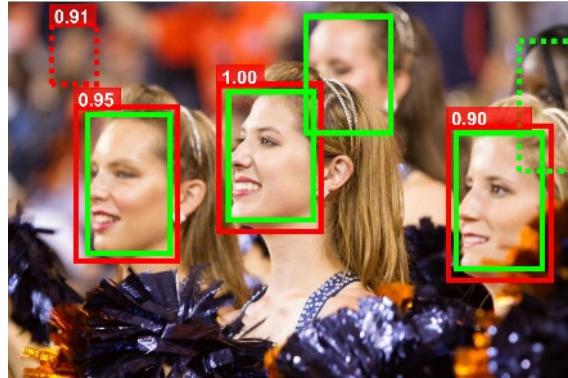
召回率(80%), 精度(80%)



## 平均精度均值 (mAP)



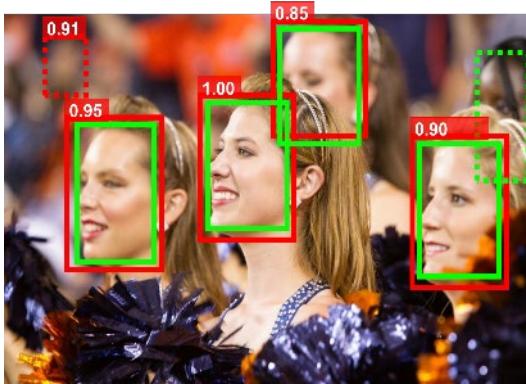
召回率(80%), 精度(80%)



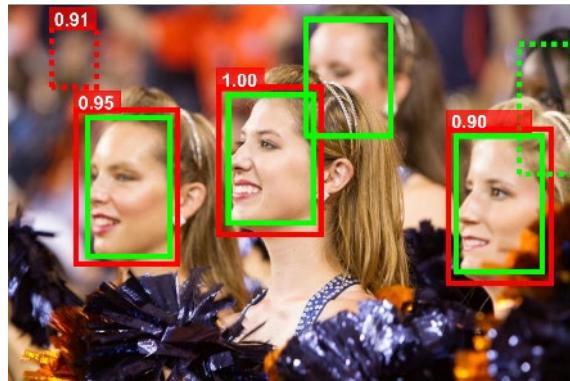
召回率(60%), 精度(75%)



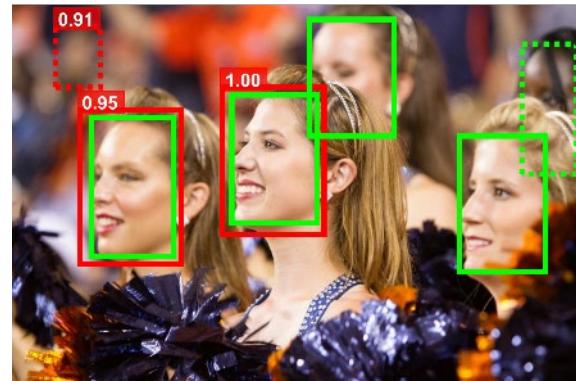
## 平均精度均值 (mAP)



召回率(80%), 精度(80%)



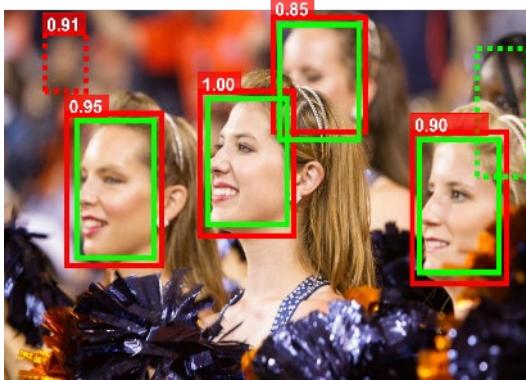
召回率(60%), 精度(75%)



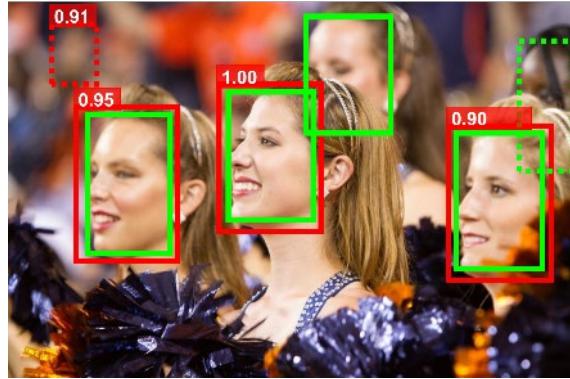
召回率(40%), 精度(67%)



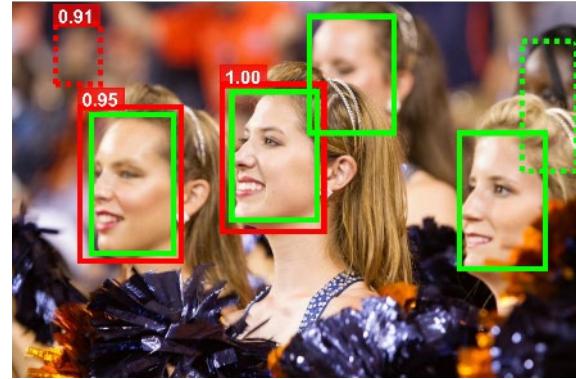
## 平均精度均值 (mAP)



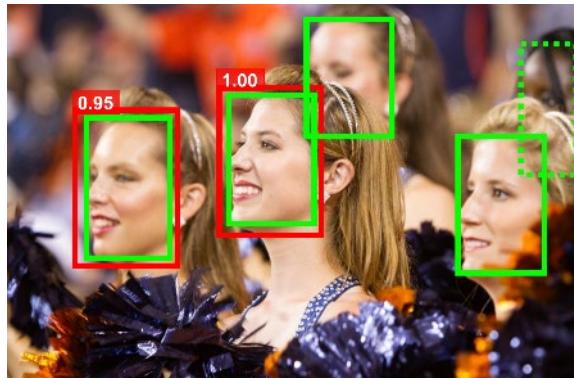
召回率(80%), 精度(80%)



召回率(60%), 精度(75%)



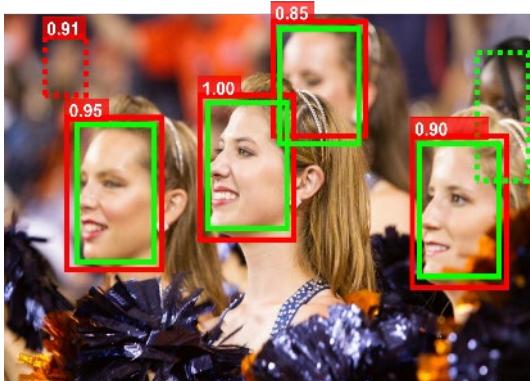
召回率(40%), 精度(67%)



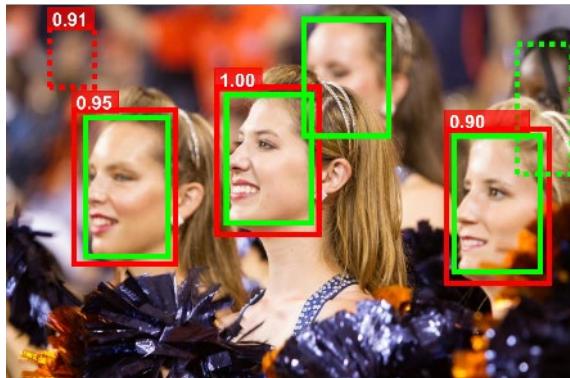
召回率(40%), 精度(100%)



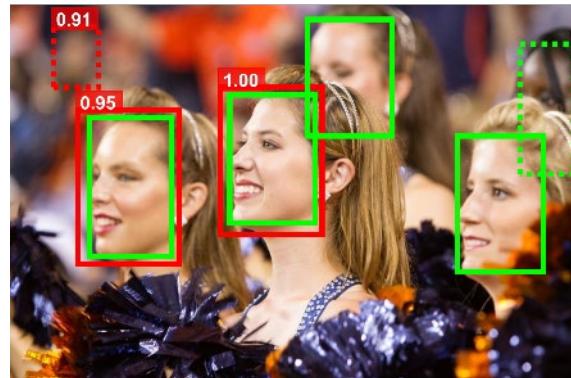
## 平均精度均值 (mAP)



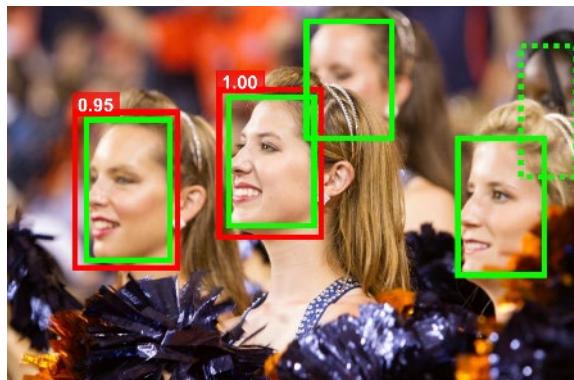
召回率(80%), 精度(80%)



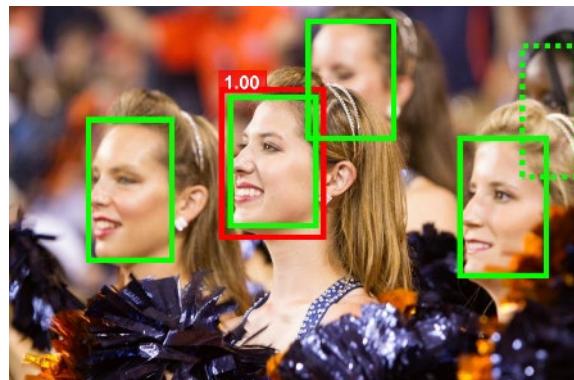
召回率(60%), 精度(75%)



召回率(40%), 精度(67%)



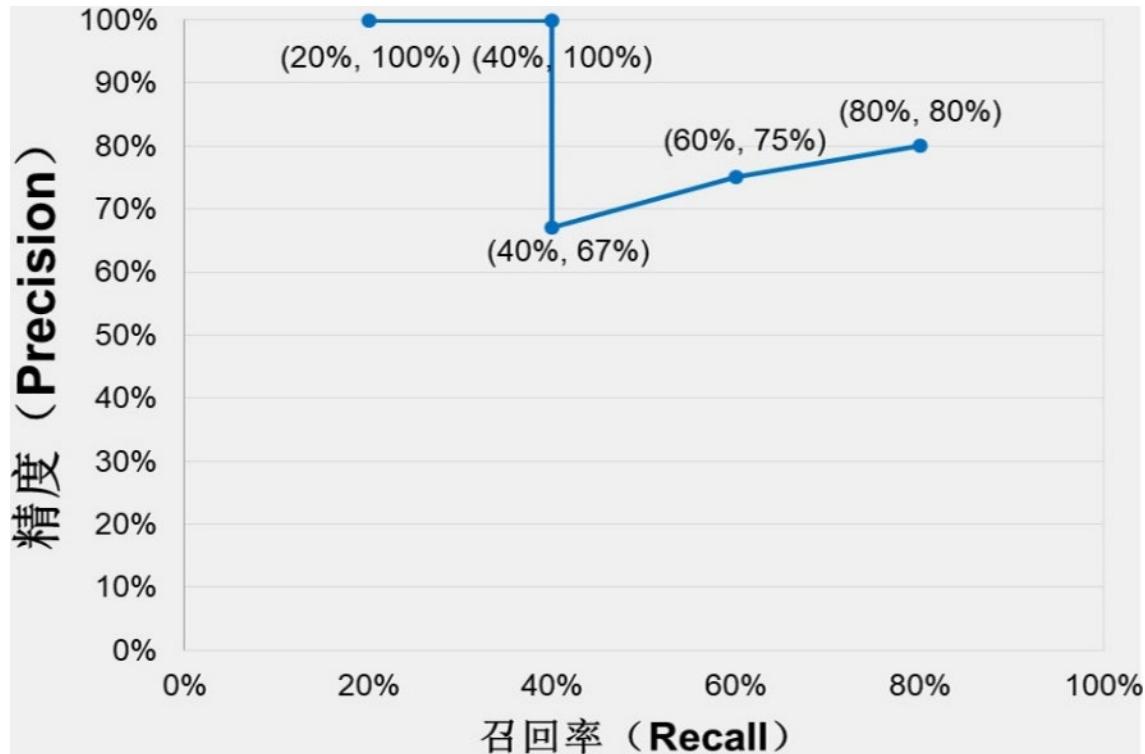
召回率(40%), 精度(100%)



召回率(20%), 精度(100%)

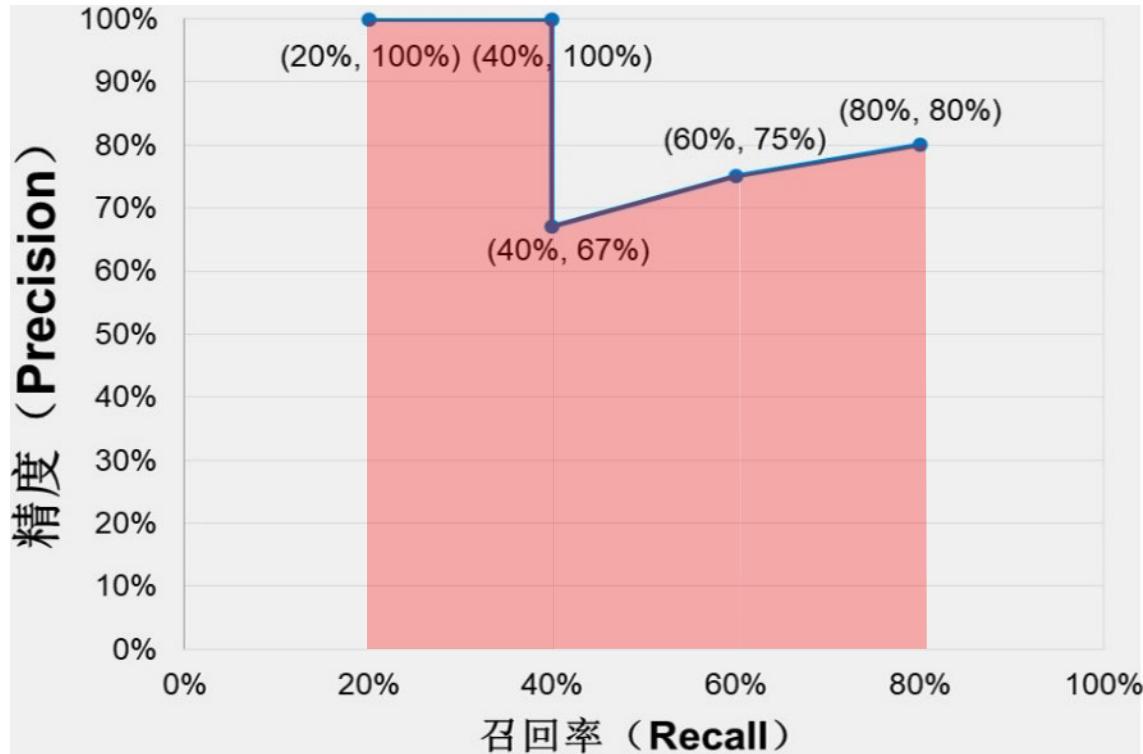


## 平均精度均值 (mAP)





# 平均精度均值 (mAP)

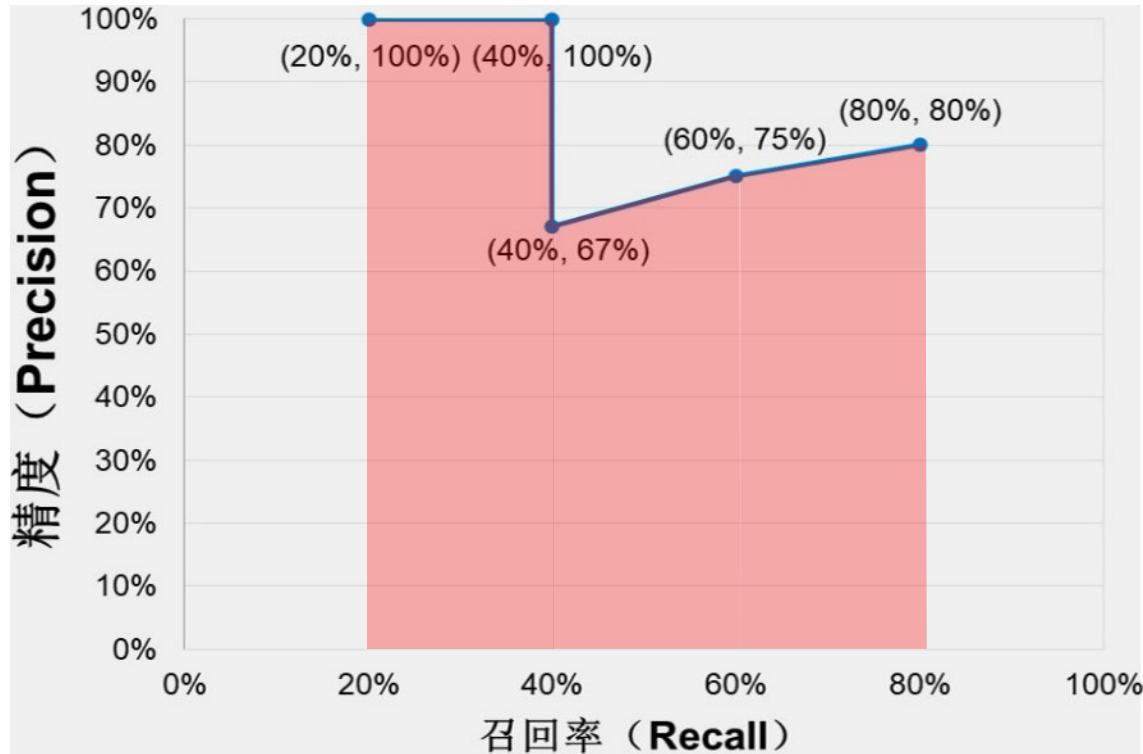


## 平均精度 (AP)

精度-召回率曲线下的面积



# 平均精度均值 (mAP)



## 平均精度 (AP)

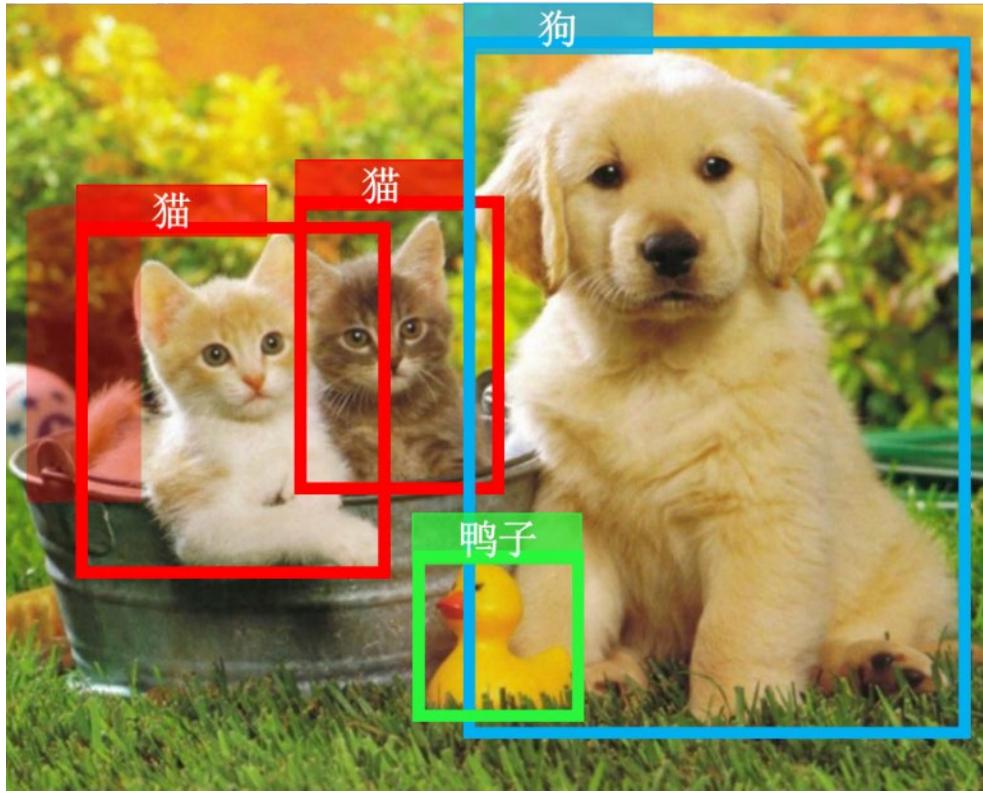
精度-召回率曲线下的面积

## 平均精度均值 (mAP)

不同类别的平均精度的均值



# 平均精度均值 (mAP)



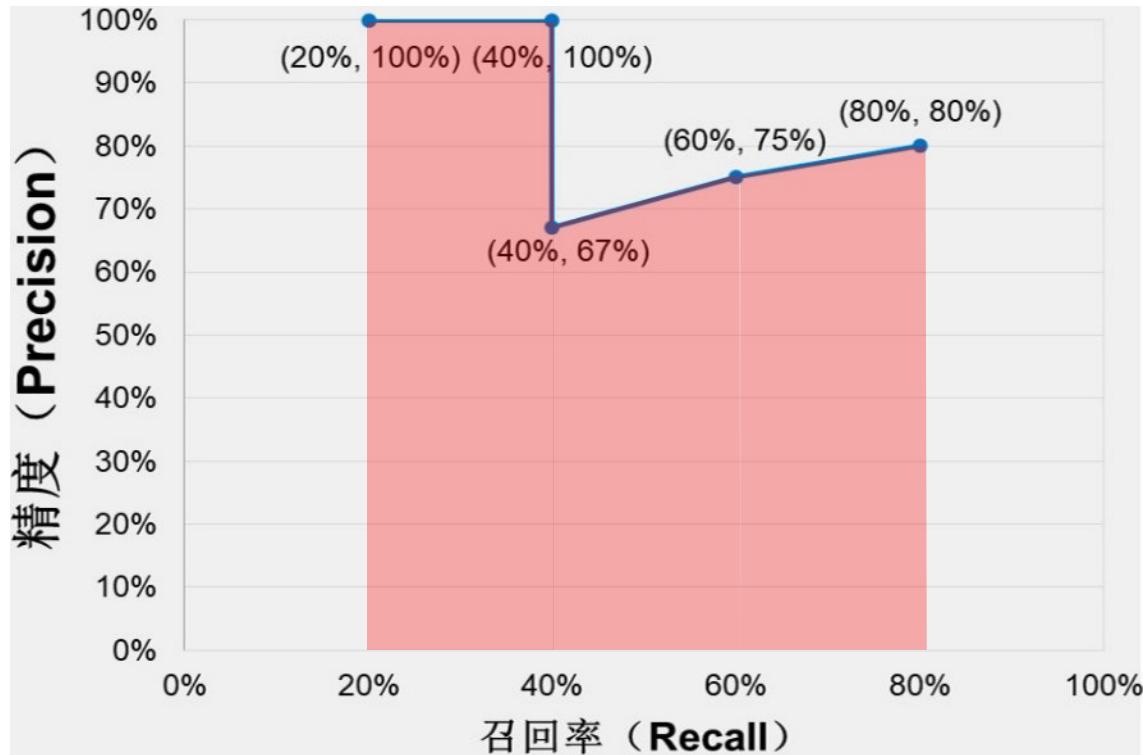
## 三类物体检测器

### (猫、狗、鸭子)

- 狗的平均精度: 97%
- 猫的平均精度: 99%
- 鸭子的平均精度: 98%
- 该检测器的平均精度均值:  
 $(97\% + 99\% + 98\%) / 3 = 98\%$



# 平均精度均值 (mAP)



$IoU \geq 0.5$ , 常用评价指标

## 平均精度 (AP)

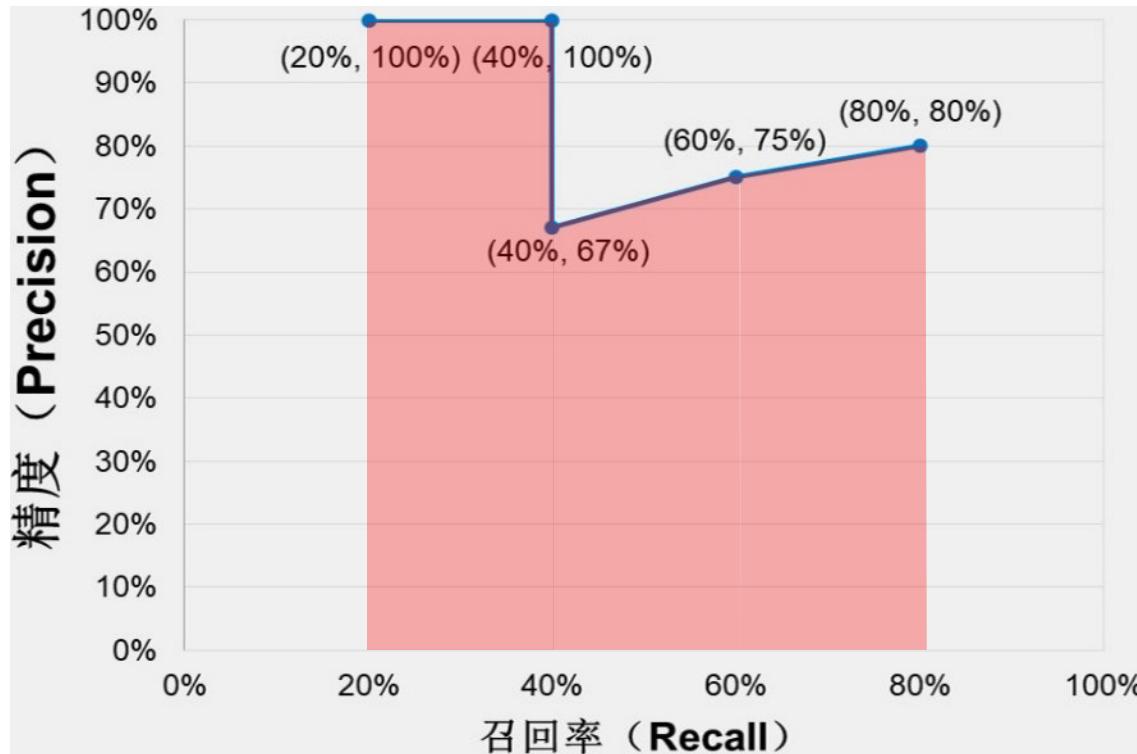
精度-召回率曲线下的面积

## 平均精度均值 (mAP)

不同类别的平均精度的均值



# 平均精度均值 (mAP)



$IoU \geq 0.5$ , 常用评价指标

## 平均精度 (AP)

精度-召回率曲线下的面积

## 平均精度均值 (mAP)

不同类别的平均精度的均值

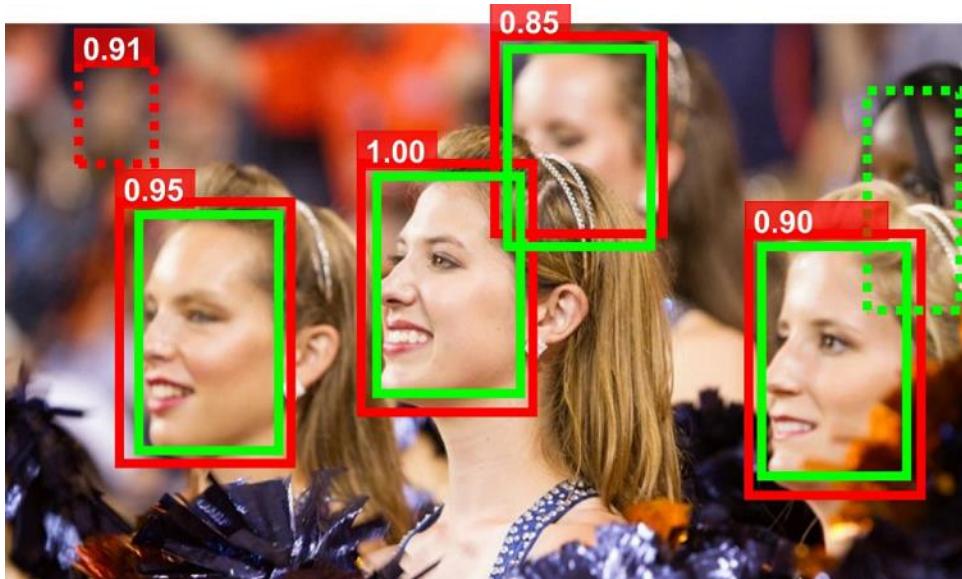
- $IoU \geq [0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95]$

- 10个不同IoU阈值下的mAP
- 再一次求平均得到mmAP
- MS COCO数据集用的评价指标



# 漏检率 (Miss Rate)

- 通用物体检测 (20类、80类、1000类) , 人脸检测或行人检测 (1类)
- 检测精度评价指标: 每类先单独计算评价指标, 然后多类之间求平均
- 为了更清晰地描述检测精度的评价指标, 在这里以单类物体检测为例来说明



检测结果(□□)= 5个 真实标注 (□□) = 5个

被召回的真实标注(□) = 真正例 (□) = 4个

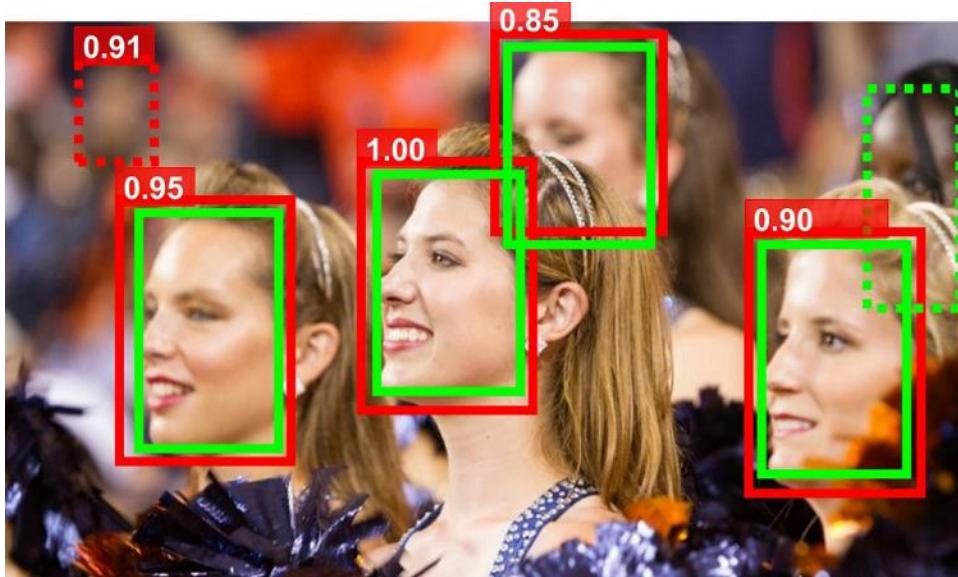
虚检(□)= 1个 漏检 (□) = 1个

$$\begin{aligned} \text{漏检率} &= \frac{\text{未被召回的真实标注的数量}}{\text{真实标注的数量}} \\ &= \frac{\text{漏检的数量}}{\text{真实标注的数量}} = \frac{1}{5} = 20\% \end{aligned}$$



## 每张图的平均虚检个数 (FPPI)

- 通用物体检测 (20类、80类、1000类) , 人脸检测或行人检测 (1类)
- 检测精度评价指标：每类先单独计算评价指标，然后多类之间求平均
- 为了更清晰地描述检测精度的评价指标，在这里以单类物体检测为例来说明



检测结果(□□)= 5个 真实标注 (□□) = 5个

被召回的真实标注(□) = 真正例 (□) = 4个

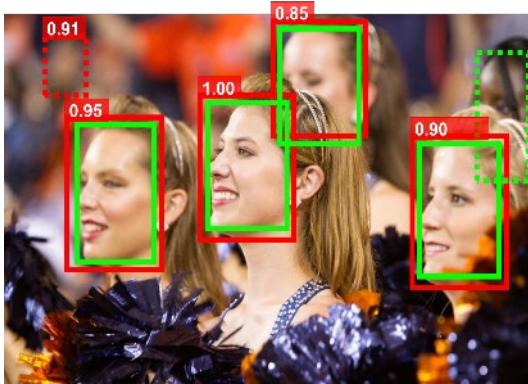
虚检(□)= 1个 漏检 (□) = 1个

每张图的平均虚检个数 =

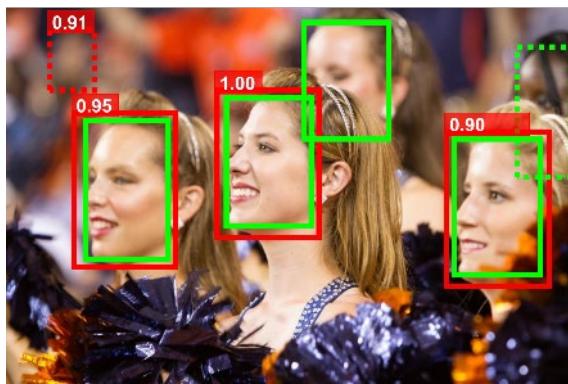
$$\frac{\text{虚检的数量}}{\text{图像的数量}} = \frac{1}{1} = 1$$



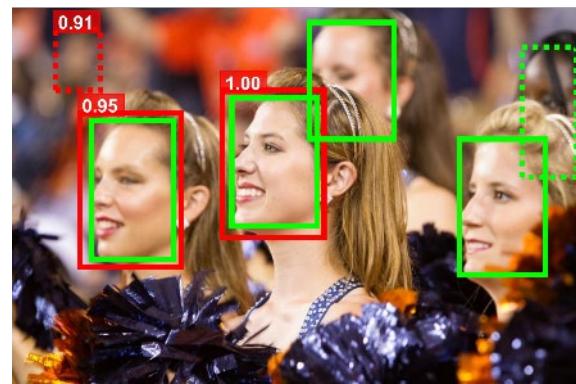
## 平均对数漏检率 (MR<sup>-2</sup>)



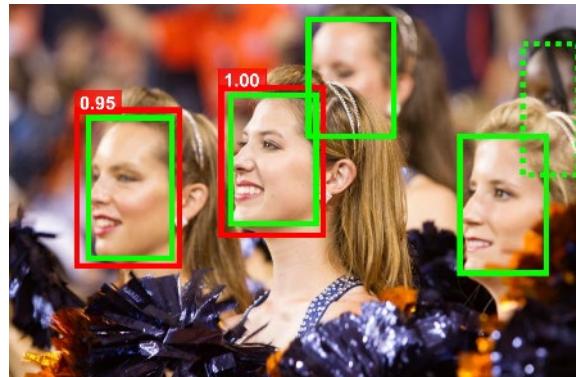
漏检率(20%), FPPI (1)



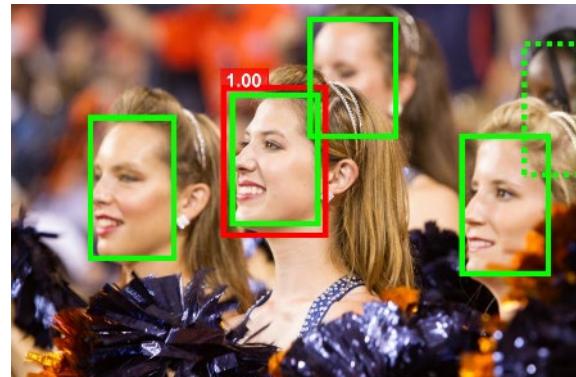
漏检率(40%), FPPI (1)



漏检率(60%), FPPI (1)



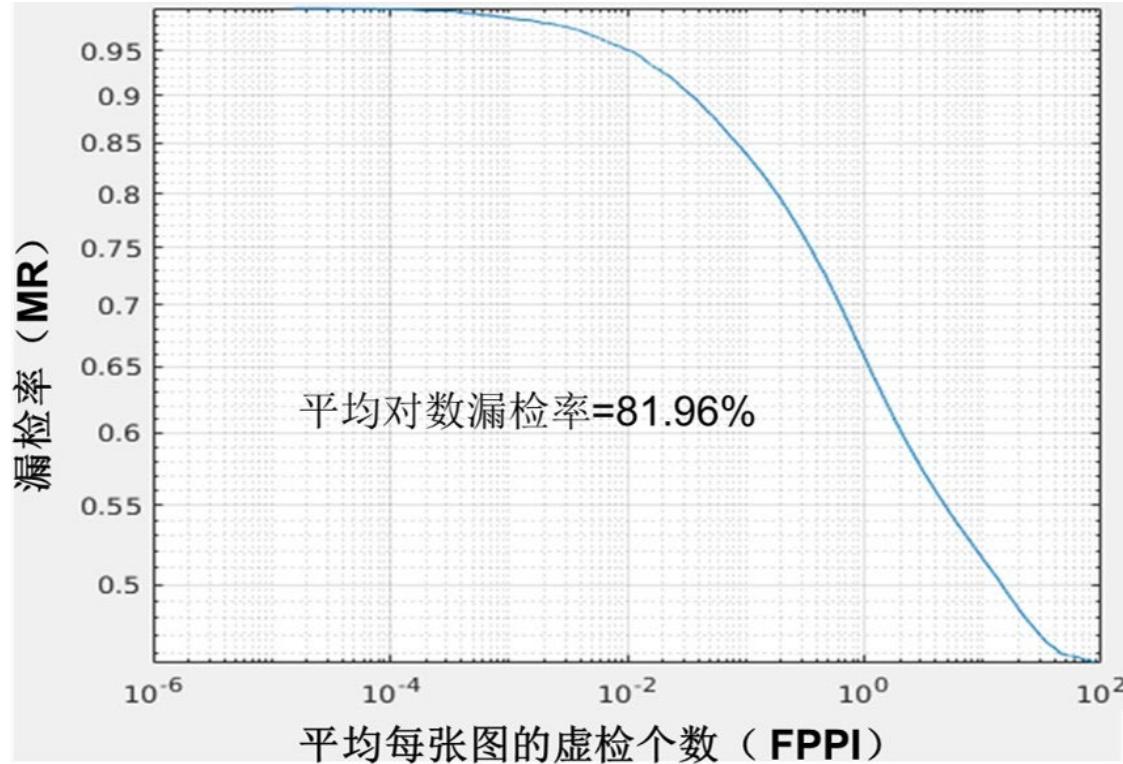
漏检率(60%), FPPI (0)



漏检率(80%), FPPI (0)

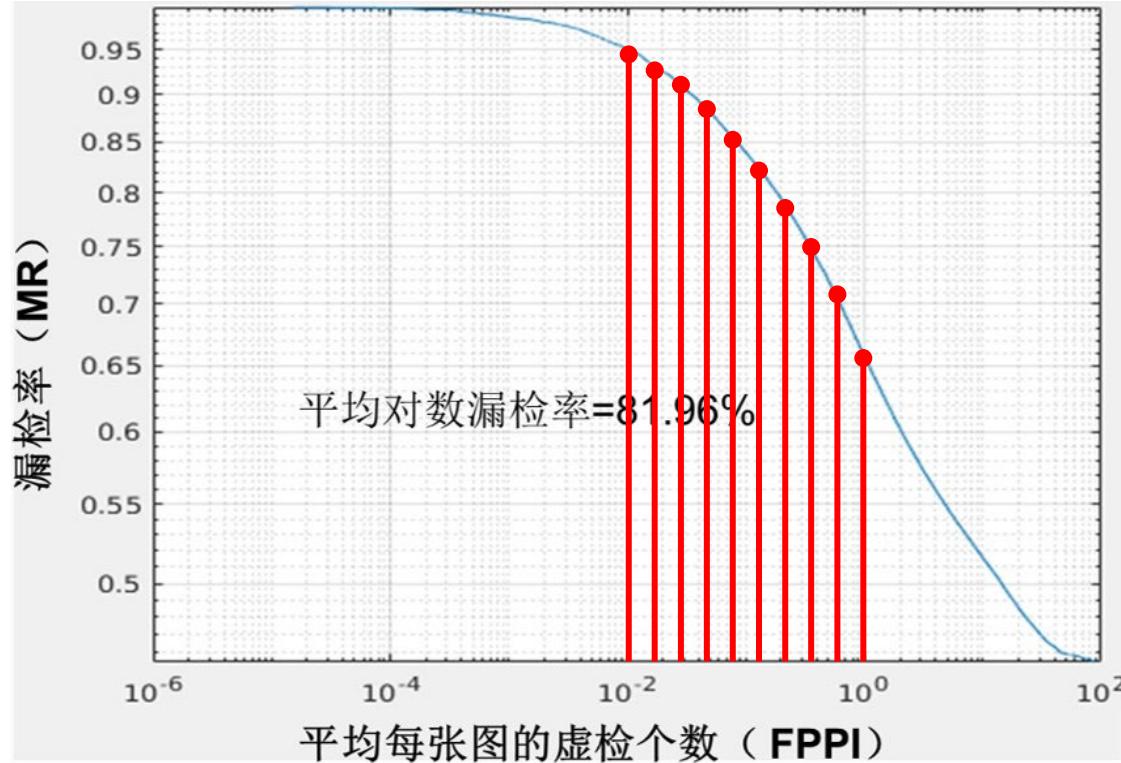


## 平均对数漏检率 ( $MR^{-2}$ )





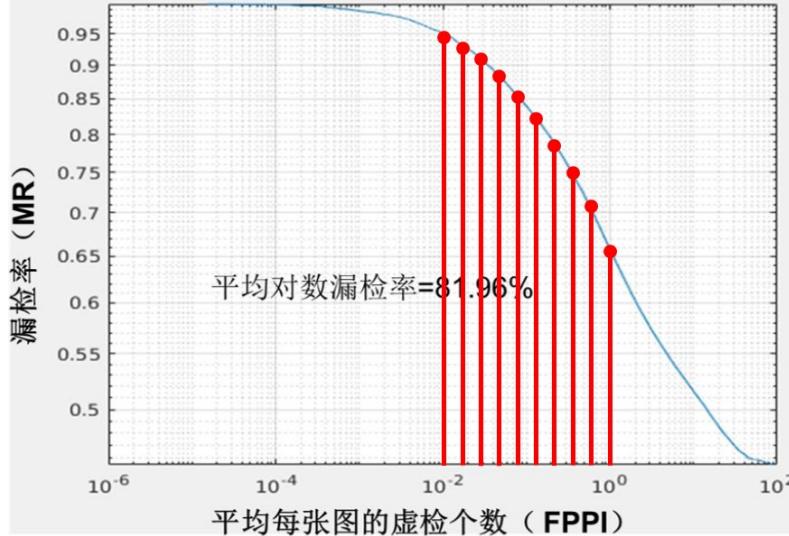
# 平均对数漏检率 ( $MR^{-2}$ )



平均对数漏检率  
 $= e^{\left(\frac{1}{9} \sum_{i=1}^9 \ln(MR_i)\right)}$



# 平均对数漏检率 ( $MR^{-2}$ )



平均对数漏检率

$$= e^{\left(\frac{1}{9} \sum_{i=1}^9 \ln(MR_i)\right)}$$



# 物体检测评价指标

检测精度

召回率 (Recall Rate)

平均精度均值 (mAP)

平均对数漏检率 ( $MR^{-2}$ )

检测速度

前传耗时 (ms)

每秒帧数 (FPS)

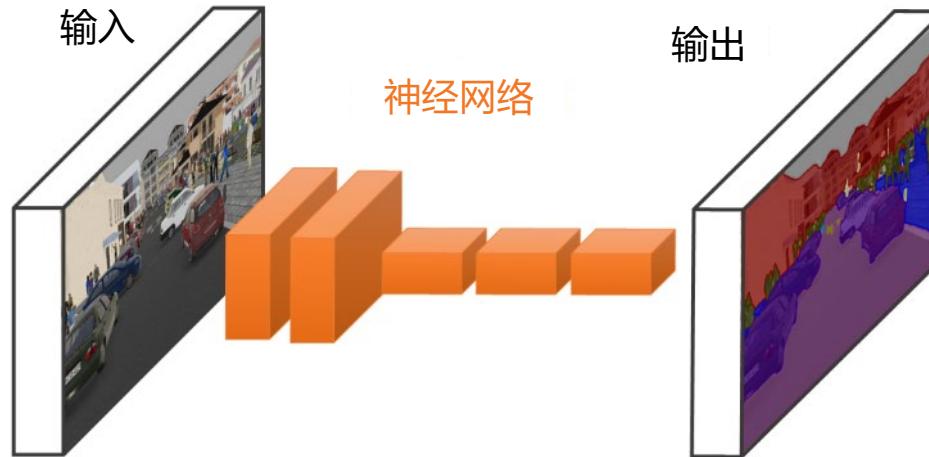
浮点运算量 (FLOPs)





## 检测速度评价指标

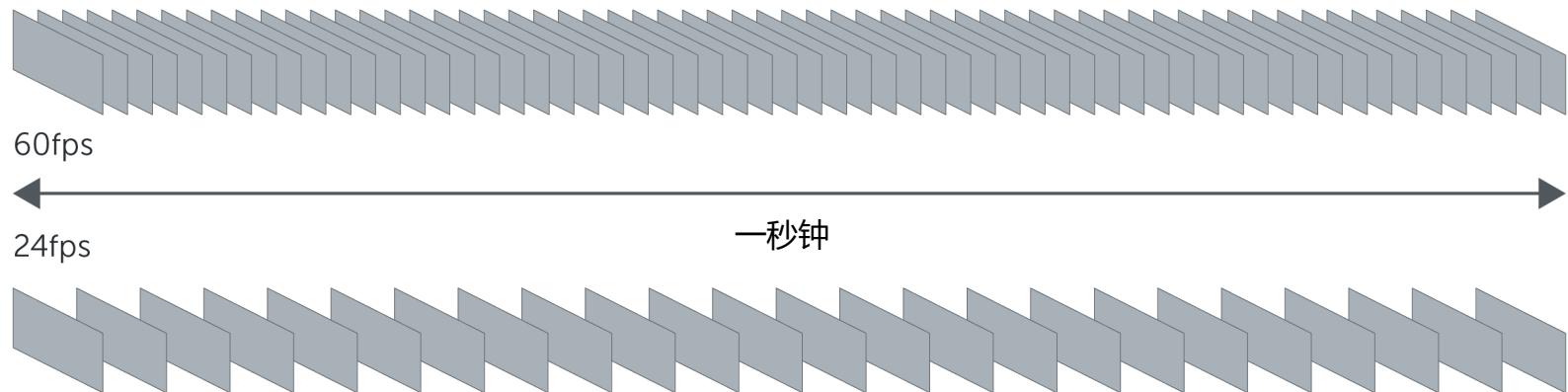
- 前传耗时 (ms): 从输入一张图像到输出最终结果所消耗的时间，这包括前处理耗时 (如图像归一化)、网络前传耗时、后处理耗时(如非极大值抑制)





## 检测速度评价指标

- **前传耗时 (ms)**: 从输入一张图像到输出最终结果所消耗的时间，这包括前处理耗时(如图像归一化)、网络前传耗时、后处理耗时(如非极大值抑制)
- **每秒帧数 (FPS)**: 每秒钟能处理的图像数量





## 检测速度评价指标

- **前传耗时 (ms)**: 从输入一张图像到输出最终结果所消耗的时间，这包括前处理耗时(如图像归一化)、网络前传耗时、后处理耗时(如非极大值抑制)
- **每秒帧数 (FPS)**: 每秒钟能处理的图像数量
- 前传耗时  $\times$  每秒帧数 = 1
- 两者高度依赖于软硬件情况，不同机器、不同环境下，差异非常大



低性能



中性能



高性能

方法库



# 检测速度评价指标

- **前传耗时 (ms)**: 从输入一张图像到输出最终结果所消耗的时间，这包括前处理耗时(如图像归一化)、网络前传耗时、后处理耗时(如非极大值抑制)
- **每秒帧数 (FPS)**: 每秒钟能处理的图像数量
- 前传耗时 × 每秒帧数 = 1
- 两者高度依赖于软硬件情况，不同机器、不同环境下，差异非常大
- **浮点运算量 (FLOPs)**: 处理一张图像所需要的浮点运算数量，它跟软硬件没有任何关系，可以公平地比较不同算法之间的检测速度

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

4		



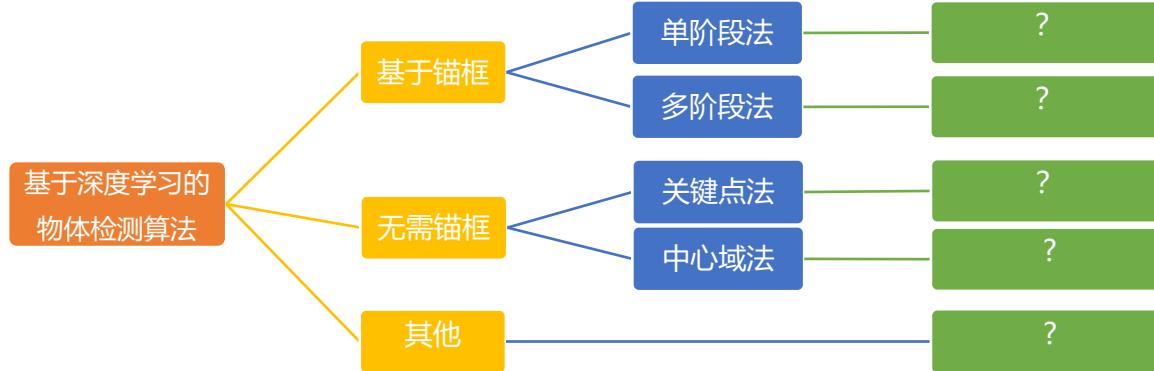
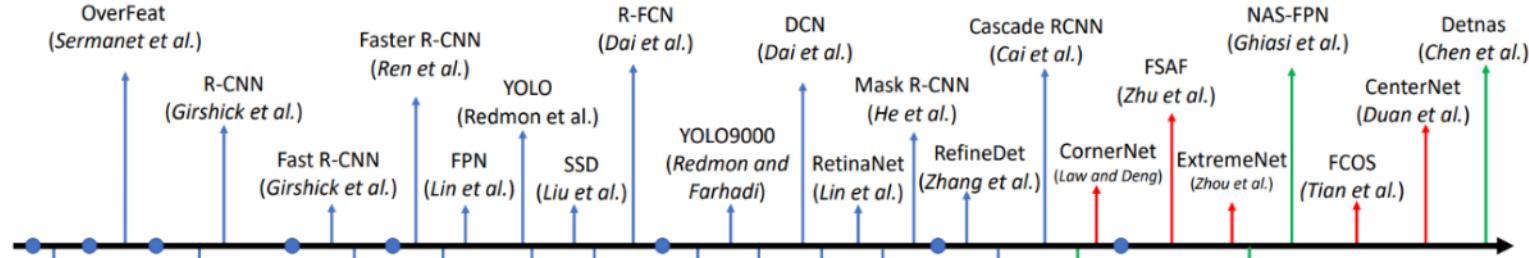
## 检测速度评价指标

- **前传耗时 (ms)**: 从输入一张图像到输出最终结果所消耗的时间，这包括前处理耗时(如图像归一化)、网络前传耗时、后处理耗时(如非极大值抑制)
- **每秒帧数 (FPS)**: 每秒钟能处理的图像数量
- 前传耗时  $\times$  每秒帧数 = 1
- 两者高度依赖于软硬件情况，不同机器、不同环境下，差异非常大
- **浮点运算量 (FLOPs)**: 处理一张图像所需要的浮点运算数量，它跟软硬件没有任何关系，可以公平地比较不同算法之间的检测速度
- 注意：在相同软硬件情况下，两个算法的浮点运算量相同，也并不代表它们的检测速度相同，因为检测速度还与数据的复用程度、占用的显存大小等有关
- **最好的办法，就是在相同的软硬件下，确定的应用场景下，比较不同算法的检测速度**



# 课程作业

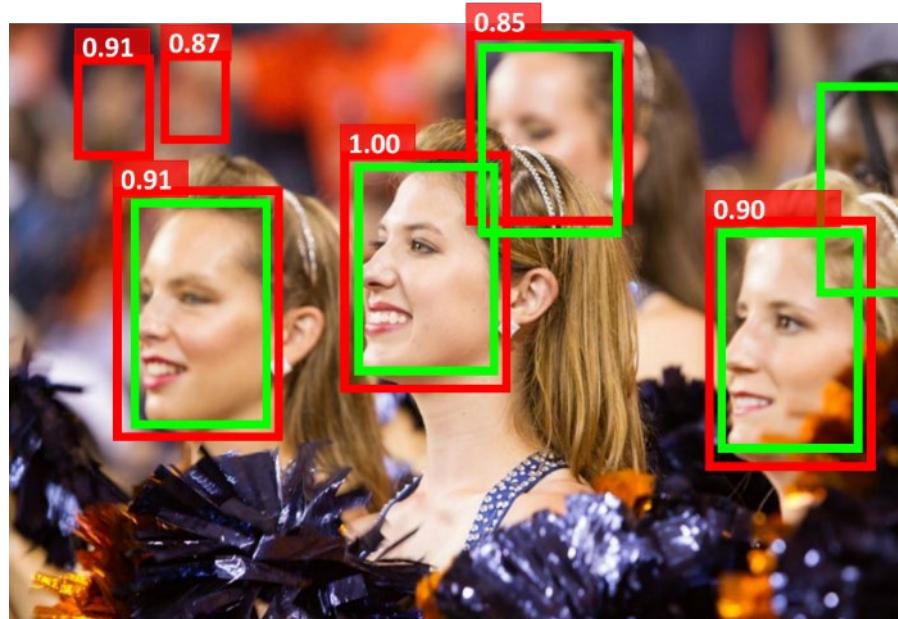
- 作业1：对下面的物体检测算法进行归纳，填入图中绿色框里





## 课程作业

- 作业2：手动计算下图的召回率、精度、漏检率，并画出召回率-精度曲线





感谢聆听 !

Thanks for Listening !