火龙果 · 整理
uml.org.cn

You Only Look Once

Unified, Real-Time Object Detection

CVF 2016

目 录

火龙果 · 整理
uml.org.cn

1 研究背景

2 实现方法

3 性能比较

4 总结分析

研究背景

火龙果 · 整理
uml.org.cn

- **问题牵引**
  - 只需一眼，人类便可识别眼前物体及其关系，快速识别不同模式、根据早前知识进行归纳、以及适应不同的图像环境一直都是人类的专属技能，机器尚未获得
  - 计算机如何实现类似人类视觉系统快速、准确的检测？
  - 如何提升目标检测的速度？

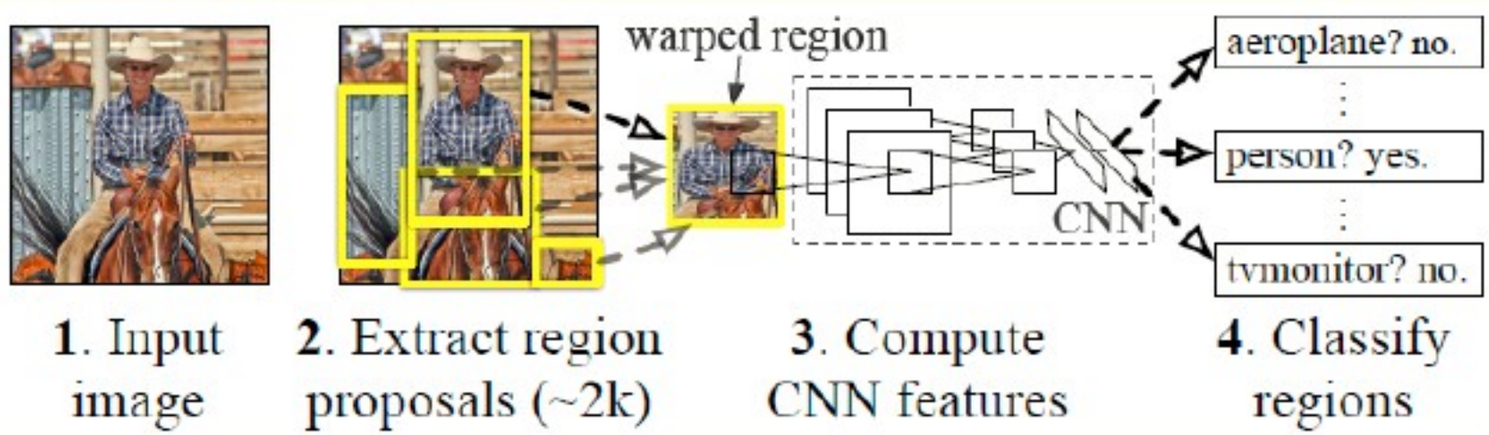
研究背景

火龙果 · 整理
uml.org.cn

- **研究现状**
  - 普遍使用分类器classifier执行检测
  - 过程复杂，速度较慢，训练耗时

Detection Systems

- **DPM**
  Deformable Parts Models（DPM），采用sliding window 检测
- **R-CNN、Fast R-CNN**
  采用 region proposals 的方法，先生成一些可能包含待检测物体的 potential bounding box，再通过一个 classifier (SVM) 去判断每个 bounding box 里是否包含有物体，以及物体所属类别的 probability 或者 confidence



研究背景

火龙果 · 整理
uml.org.cn

- **YOLO**

  将检测变为一个 regression problem，YOLO 从输入的图像，仅仅经过一个 neural network，直接得到 bounding boxes 以及每个 bounding box 所属类别的概率。正因为整个的检测过程仅仅有一个网络，所以它可以直接 end-to-end 的优化。

  YOLO 结构十分的快，标准版YOLO 每秒可以实时地处理 45 帧图像。一个较小版本Fast YOLO，可以每秒处理 155 帧图像，它的 mAP（mean Average Precision）依然可以达到其他实时检测算法的两倍。

实现方法

火龙果 · 整理
uml.org.cn

- **核心思想**

输入：利用整张图作为网络的输入，

输出：直接在输出层回归bounding box的位置和bounding box所属的类别。

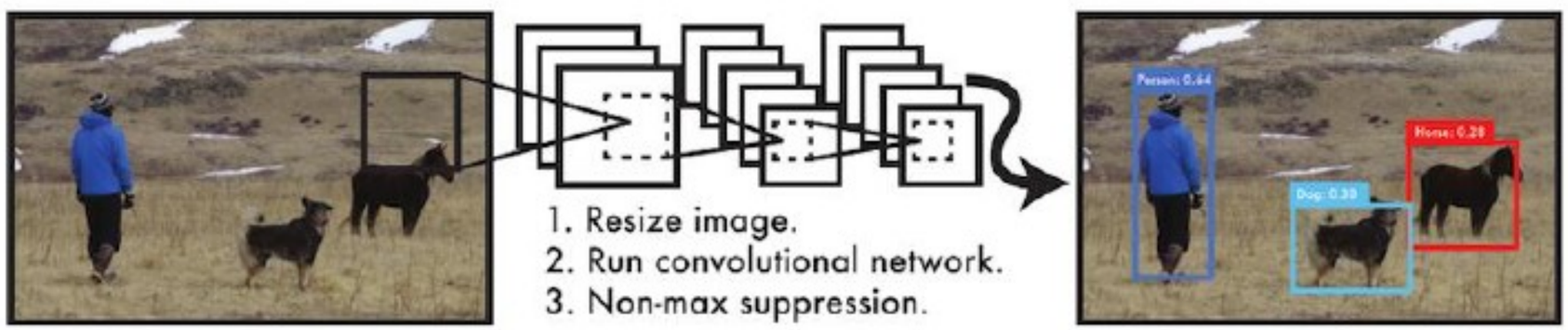


Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448×448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

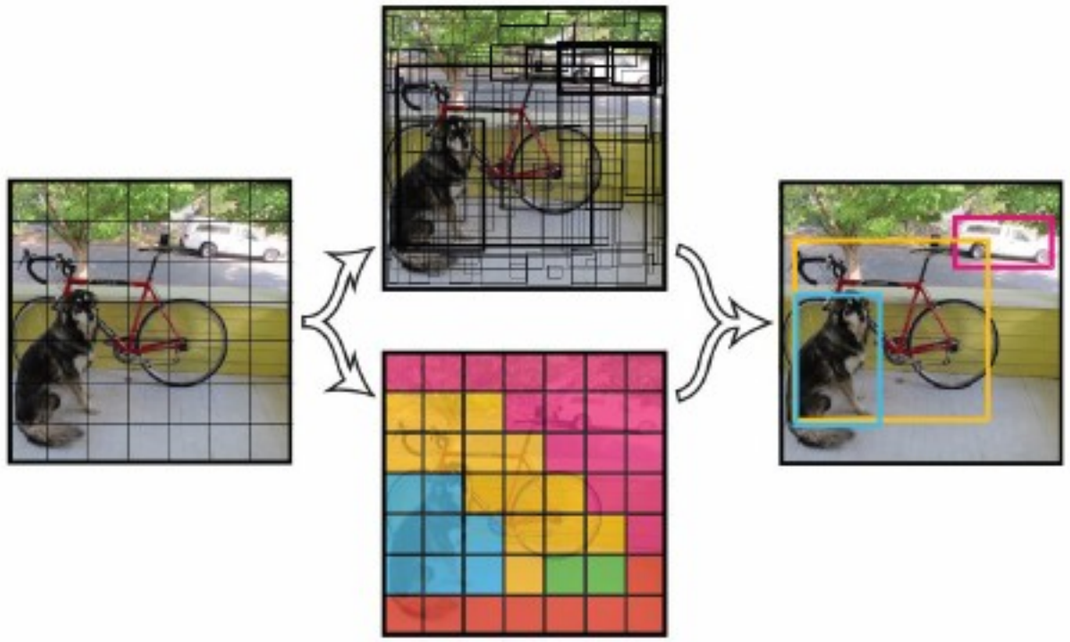
实现方法

火龙果 · 整理
uml.org.cn

- **核心思想**

1. 将一幅图像分成SxS个网格(grid cell)，如果某个object的中心落在这个网格中，则这个网格负责预测这个object。

2. 每个网格要预测B个bounding box，每个bounding box除了要回归自身的位置之外，还要附带预测一个confidence值，即每个box要预测(x, y, w, h)和confidence共5个值。



3. 这个confidence代表了所预测的box中含有object的置信度和这个box预测的有多准两重信息

$$\Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

其中如果有object落在一个grid cell里，第一项取1，否则取0。 第二项是预测的bounding box和实际的ground truth之间的IOU值。

实现方法

火龙果 · 整理
uml.org.cn

- **核心思想**

4. 每个bounding box要预测(x, y, w, h)和confidence共5个值，每个网格还要预测一个类别信息，记为C类。则SxS个网格，每个网格要预测B个bounding box还要预测C个categories。输出S x S x (5*B+C)的一个tensor。
   注意：class信息是针对每个网格的，confidence信息是针对每个bounding box的。

5. 在test的时候，每个网格预测的class信息和bounding box预测confidence信息相乘，就得到每个bounding box的**class-specific confidence score**:

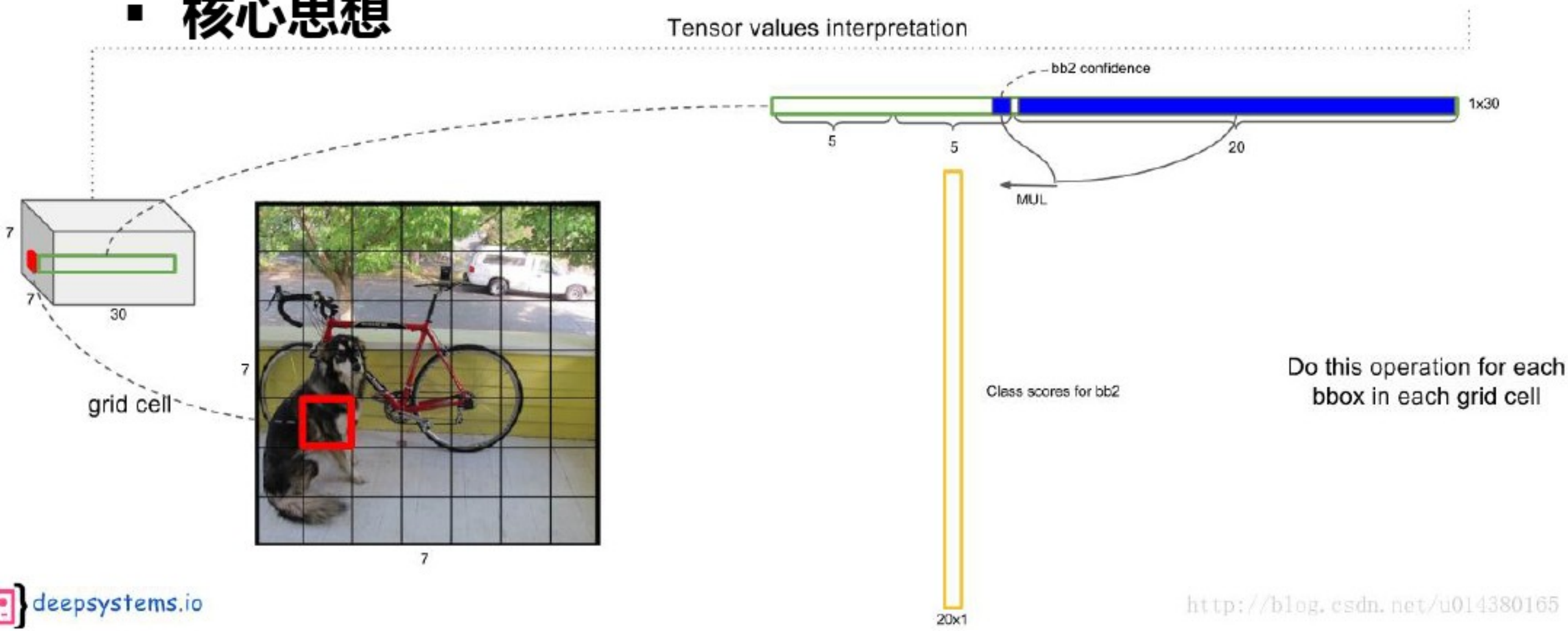
$$\Pr(\mathrm{Class_i}|\mathrm{Object}) * \Pr(\mathrm{Object}) * \mathrm{IOU}_{\mathrm{pred}}^{\mathrm{truth}} = \Pr(\mathrm{Class_i}) * \mathrm{IOU}_{\mathrm{pred}}^{\mathrm{truth}}$$

   等式左边第一项就是每个网格预测的类别信息，第二三项就是每个bounding box预测的confidence。这个乘积即表示了预测的b-box属于某一类的概率，也有该box准确度的信息。

实现方法

火龙果 · 整理
uml.org.cn

- **核心思想**

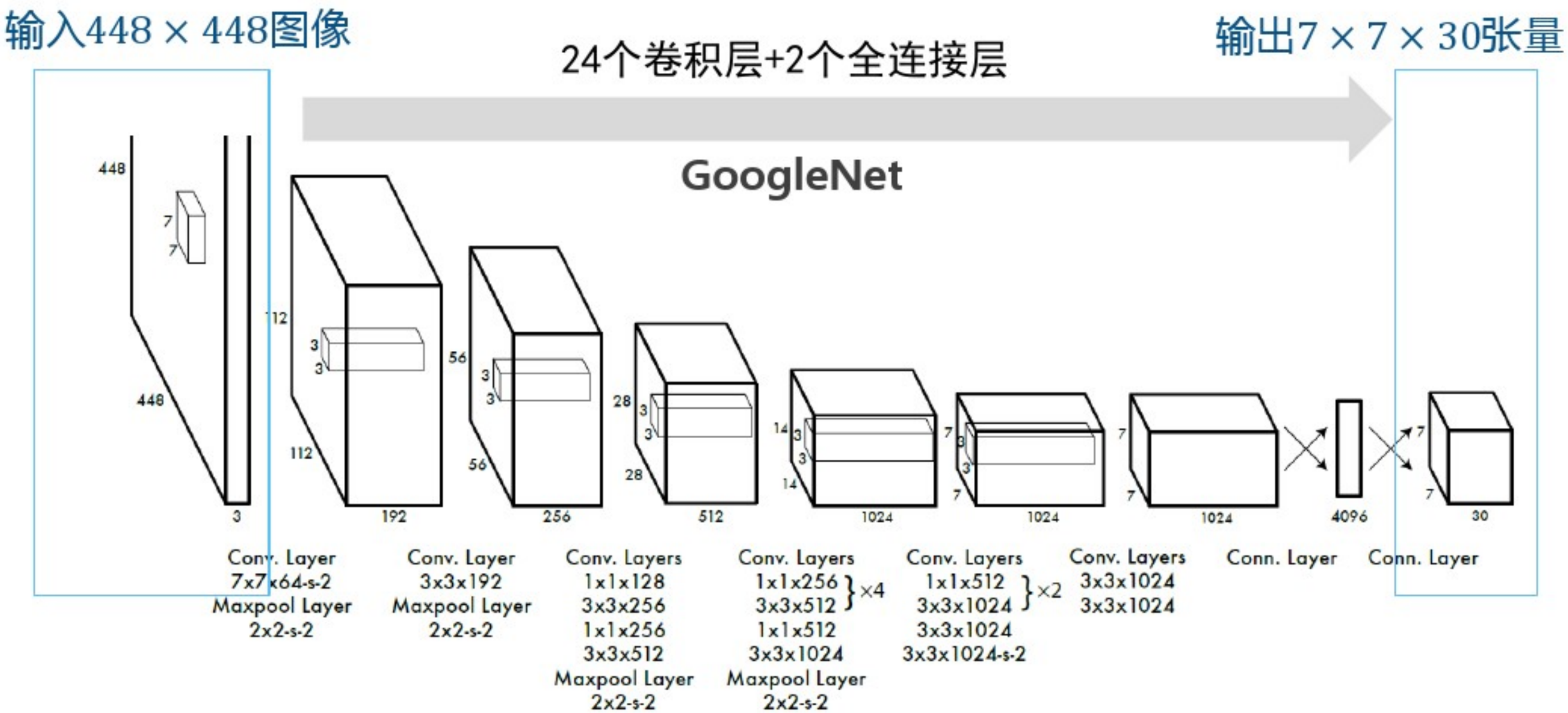


1. 图像分成 7×7 个网格(grid cell)，检测20个类别，每个网格要预测2个b-box
2. 一个b-box输出张量为 $1 \times (5 + 5 + 20)$
3. 得到 $20 \times (7 \times 7 \times 2)$ 的矩阵 → NMS方法去除重复率大的b-box

实现方法

火龙果 · 整理
uml.org.cn

- **核心思想**



实现方法

火龙果 · 整理
uml.org.cn

- **损失函数**

Output：S x S x (5*B+C)的一个tensor
B：(x, y, w, h, confidence)
C：Pi(C) of Category

误差 = 定位误差 + 分类误差 → 不能等值加权

误差 = 定位误差↑+ 置信度误差 + 分类误差 → 置信度有两类

误差 = 定位误差↑+ 包含目标的框置信度误差
+ 不包含目标的框置信度误差↓+ 分类误差

实现方法

火龙果 · 整理
uml.org.cn

- **损失函数**

采用平方根之差兼顾大框和小框的误差

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

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

坐标预测

判断第i个网格中的第j个box是否负责这个object

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

含object的box的confidence预测

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

不含object的box的confidence预测

判断是否有object中心落在网格i中

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

类别预测

实现方法

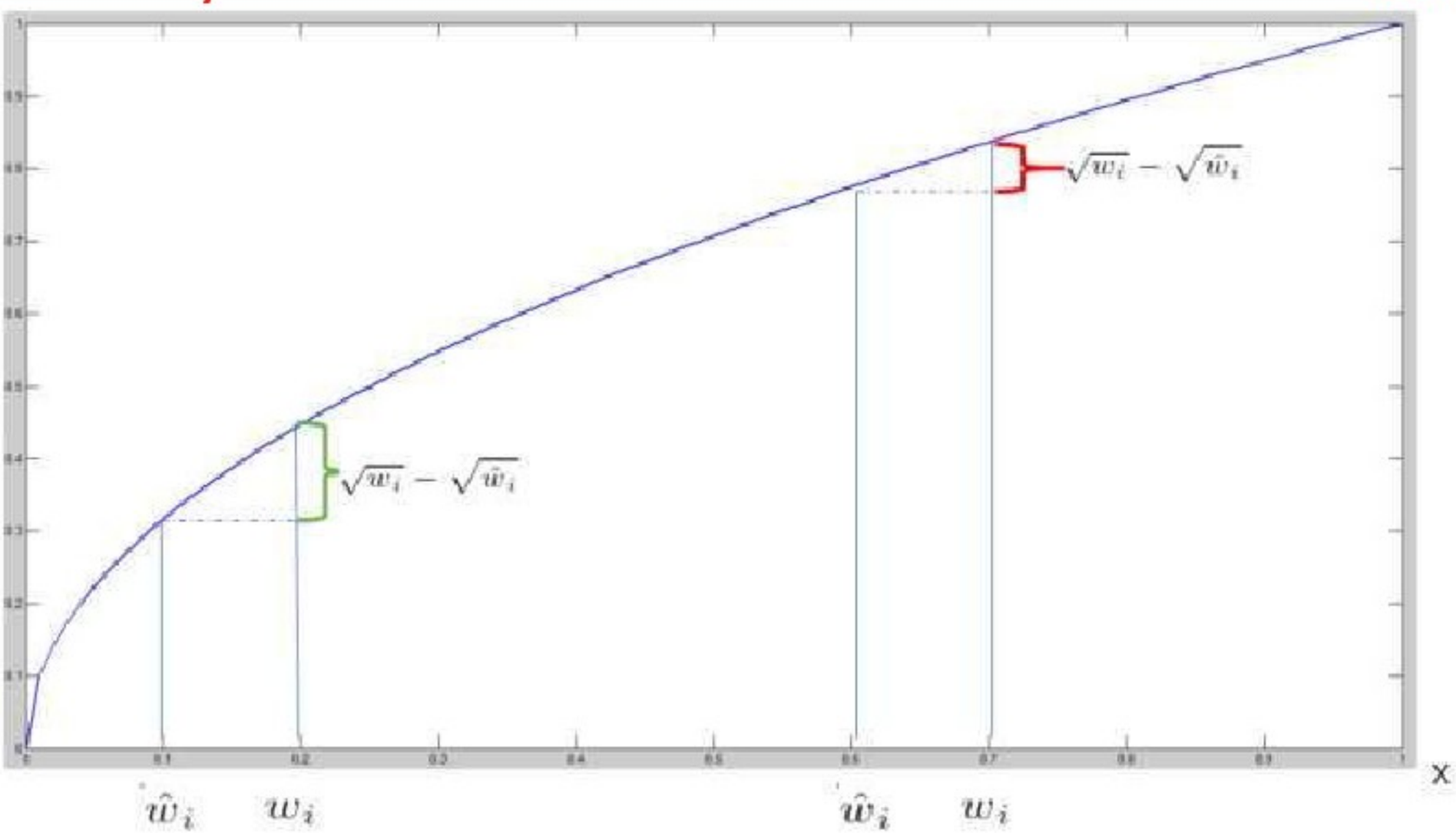
火龙果 · 整理
uml.org.cn

- **损失函数**

采用平方根之差兼顾大框和小框的误差

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

坐标预测



Small bbox

Big bbox

性能比较

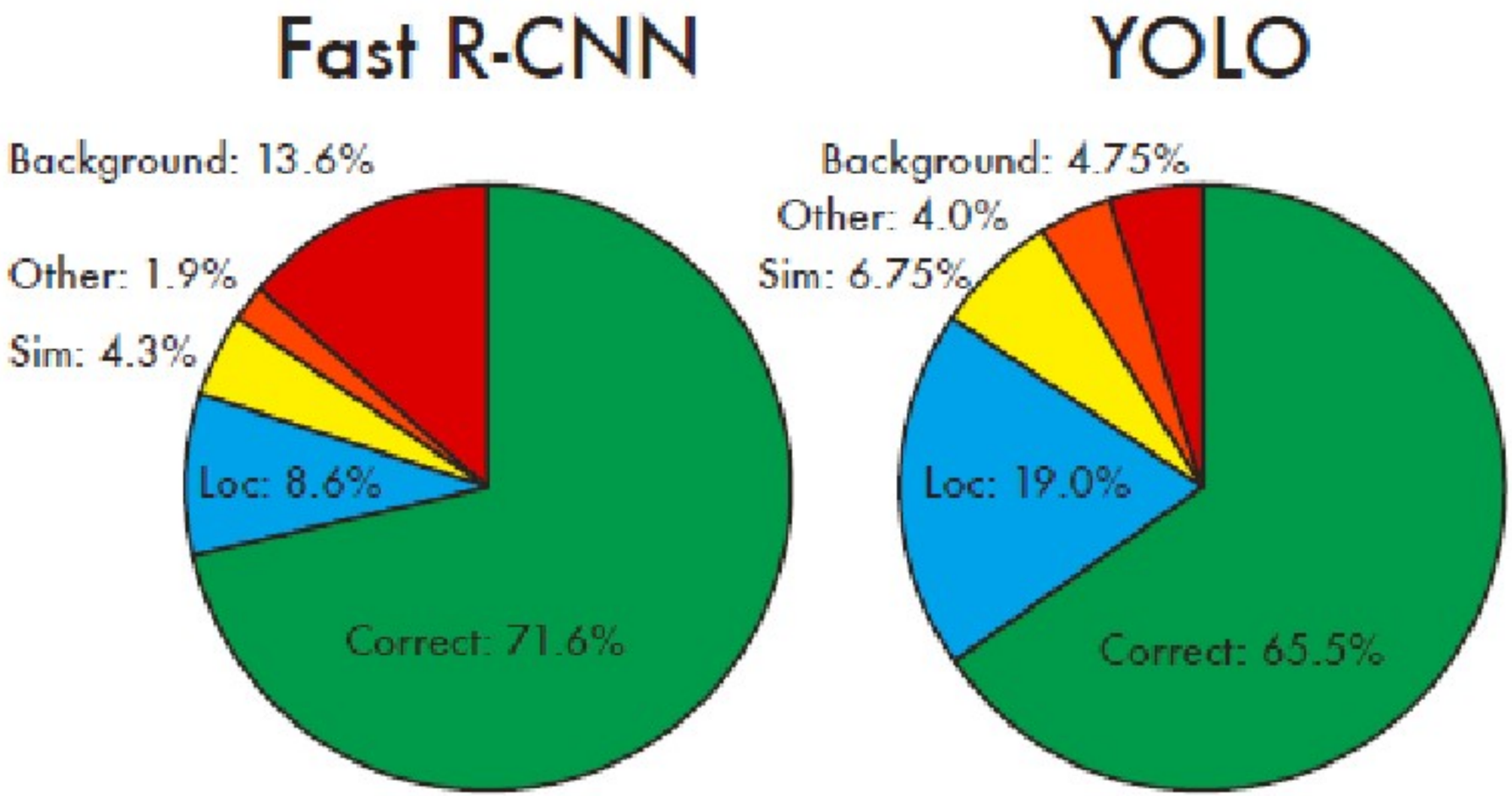
火龙果 · 整理
uml.org.cn

1. 与其它一些实时检测系统性能比较

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [30]	2007	16.0	100
30Hz DPM [30]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [37]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[27]	2007+2012	73.2	7
Faster R-CNN ZF [27]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

性能比较

2. 误差来源分析



- Corret: 正确检测
- Loc：类判别正确但定位错误
- Sim：类相似
- Other：类判别错误
- Background：判断为背景

火龙果 · 整理
uml.org.cn

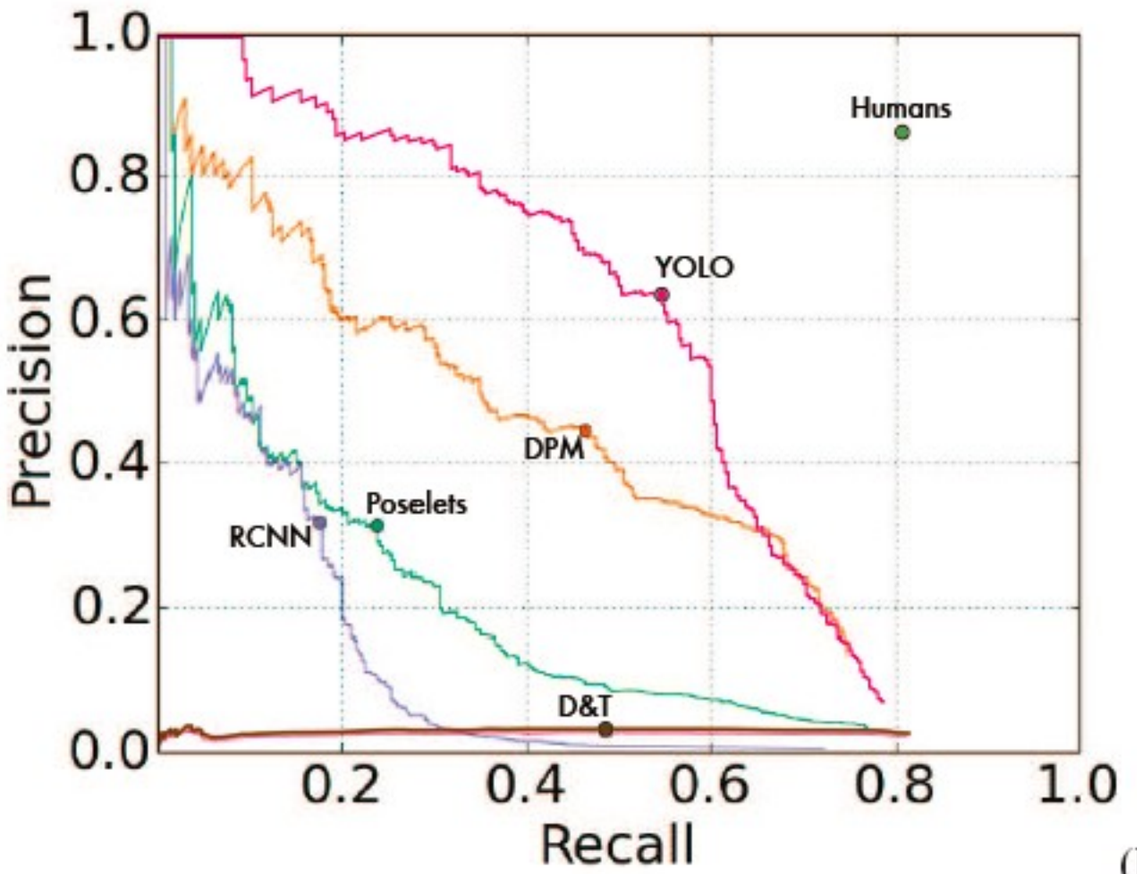
3. 结合fast RCNN的YOLO

- 思想：通过YOLO来消除fast RCNN中较大的背景误差
- 方法：对RCNN预测的每个边界框，检查YOLO是否预测类似的框。如果是，我们基于由YOLO预测的概率和两个框之间的重叠，使得该预测增强

	mAP	Combined	Gain
Fast R-CNN	71.8	-	-
Fast R-CNN (2007 data)	**66.9**	72.4	.6
Fast R-CNN (VGG-M)	59.2	72.4	.6
Fast R-CNN (CaffeNet)	57.1	72.1	.3
YOLO	63.4	**75.0**	**3.2**

火龙果 · 整理
uml.org.cn

4. 艺术品中的人检测



(a) Picasso Dataset precision-recall curves.

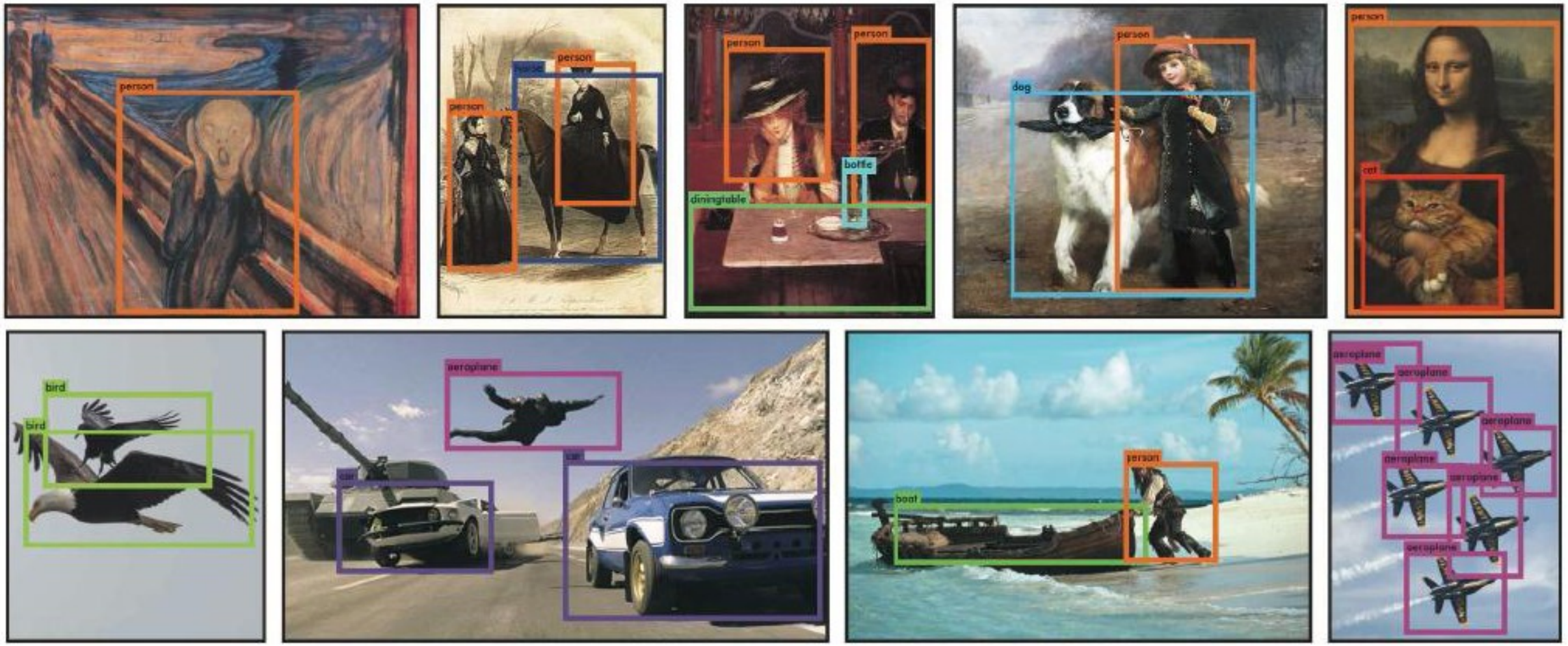
	VOC 2007	Picasso		People-Art
	AP	AP	Best F_1	AP
YOLO	**59.2**	**53.3**	**0.590**	**45**
R-CNN	54.2	10.4	0.226	26
DPM	43.2	37.8	0.458	32
Poselets [2]	36.5	17.8	0.271	
D&T [4]	-	1.9	0.051	

(b) Quantitative results on the VOC 2007, Picasso, and People-Art Datasets. The Picasso Dataset evaluates on both AP and best F_1 score.

在Picasso和People-Art数据集上的综合结果

火龙果 · 整理
uml.org.cn

4. 艺术品中的人检测



总结分析

火龙果 · 整理
uml.org.cn

- **YOLO优点**

1. 网络简单，容易构建
2. 速度快，实时性最好
3. 检测性能好，背景误差小

- **YOLO缺点**

1. 对图像上的小目标检测性能不佳
2. 对临近物体及很小的群体，检测效果不好
3. 定位误差较大。尤其是对小物体的处理上，还有待加强

火龙果 · 整理
uml.org.cn

Thank you !

CvF

You Only Look Once

2016 Unified, Real-Time Object Detection