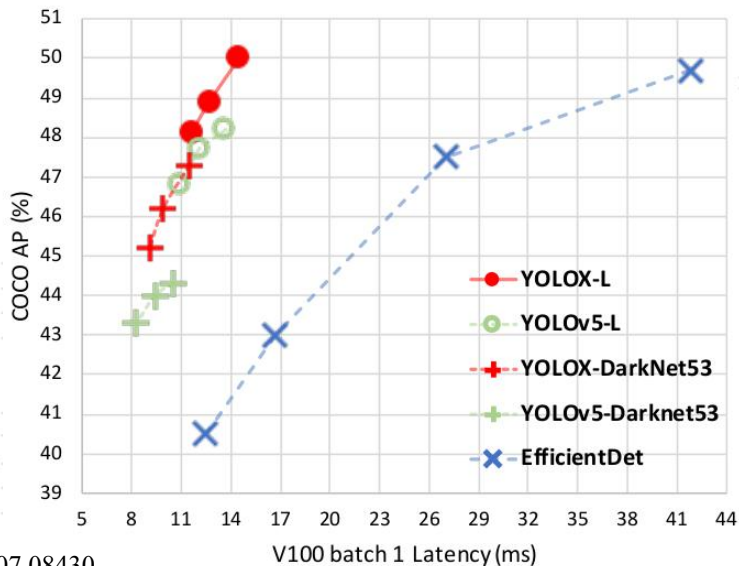


YOLOX

YOLOX: Exceeding YOLO Series in 2021

Zheng Ge* Songtao Liu*† Feng Wang Zeming Li Jian Sun
Megvii Technology

{gezhen, liusongtao, wangfeng02, lizeming, sunjian}@megvii.com



2021 CVPR

Anchor-Free

decoupled detection head

advanced label assigning
strategy(SimOTA)

1st Streaming Perception
Challenge

论文下载地址: <https://arxiv.org/abs/2107.08430>

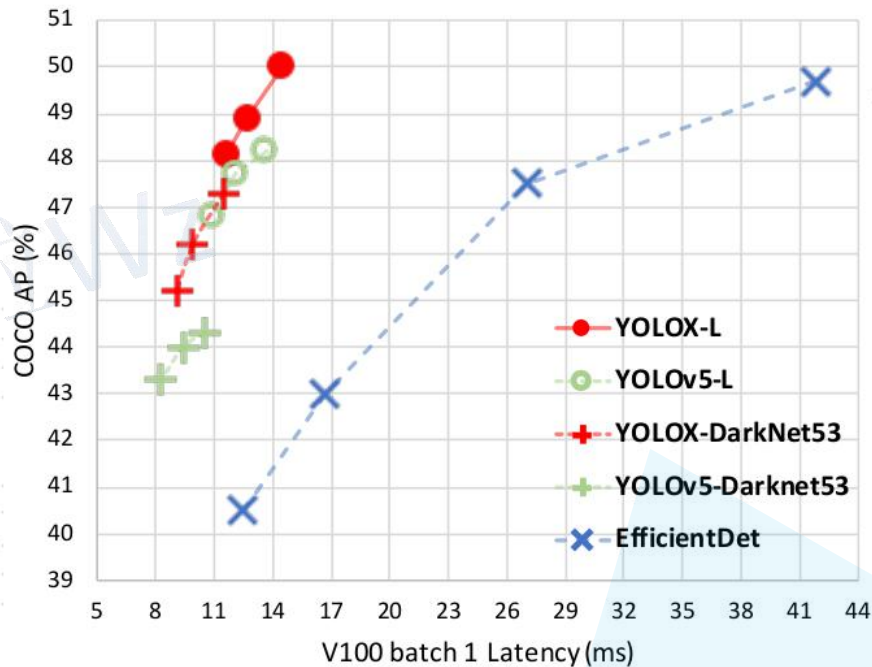
博文: https://blog.csdn.net/qq_37541097/article/details/125132817

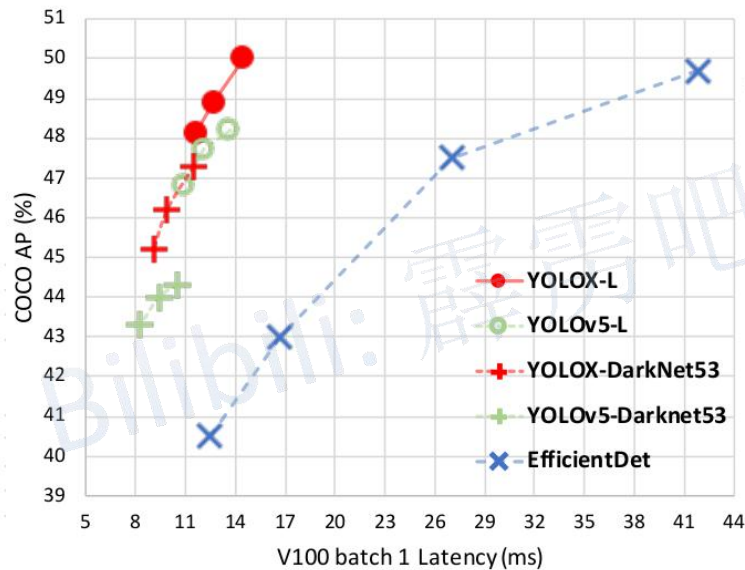
公众号“阿喆学习小记”输入YOLOX获取

YOLOX

目录

- 0 前言
- 1 YOLOX网络结构
- 2 Anchor-Free
- 3 损失计算
- 4 正负样本匹配SimOTA



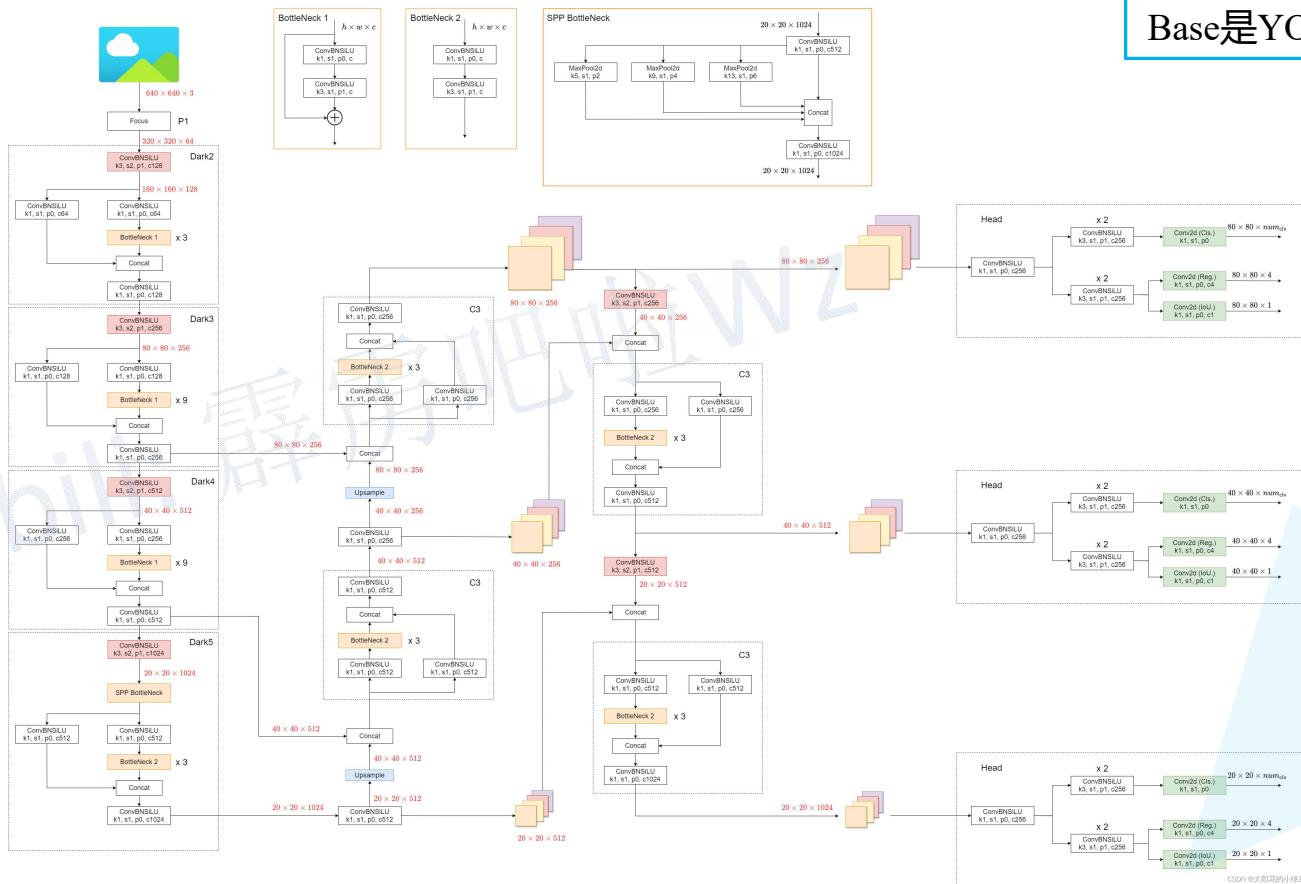


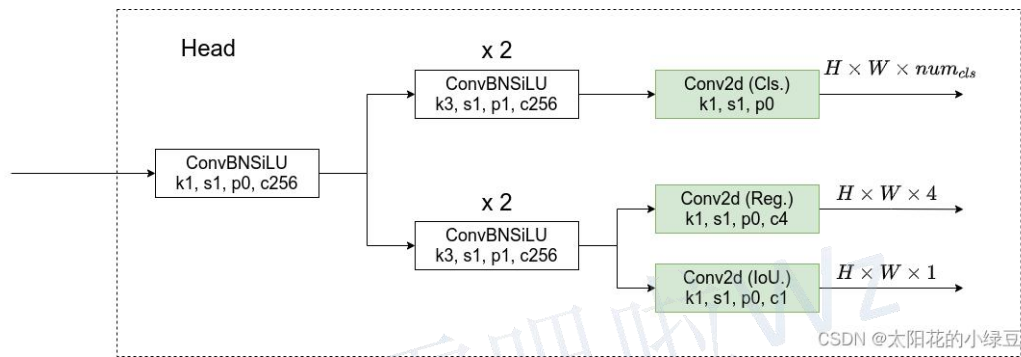
Models	AP (%)	Parameters	GFLOPs	Latency
YOLOv5-S	36.7	7.3 M	17.1	8.7 ms
YOLOX-S	39.6 (+2.9)	9.0 M	26.8	9.8 ms
YOLOv5-M	44.5	21.4 M	51.4	11.1 ms
YOLOX-M	46.4 (+1.9)	25.3 M	73.8	12.3 ms
YOLOv5-L	48.2	47.1 M	115.6	13.7 ms
YOLOX-L	50.0 (+1.8)	54.2 M	155.6	14.5 ms
YOLOv5-X	50.4	87.8 M	219.0	16.0 ms
YOLOX-X	51.2 (+0.8)	99.1 M	281.9	17.3 ms

YOLOX

网络结构 (YOLOX-L)

Base是YOLO v5 (tag:v5.0)

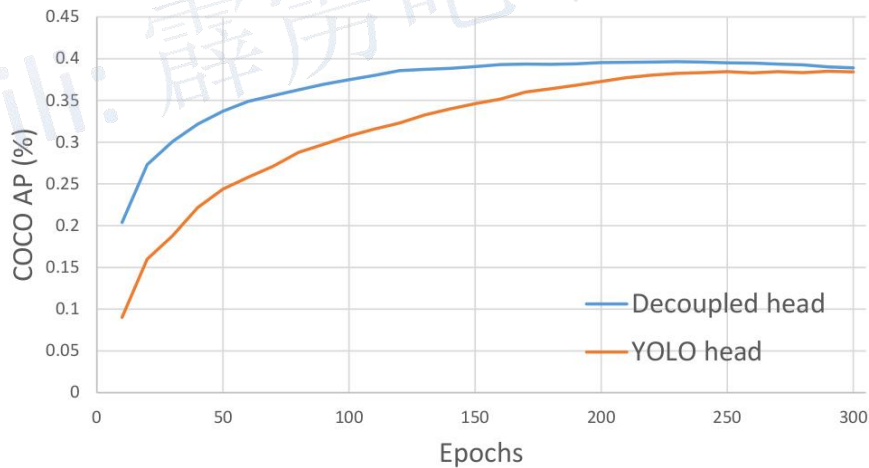




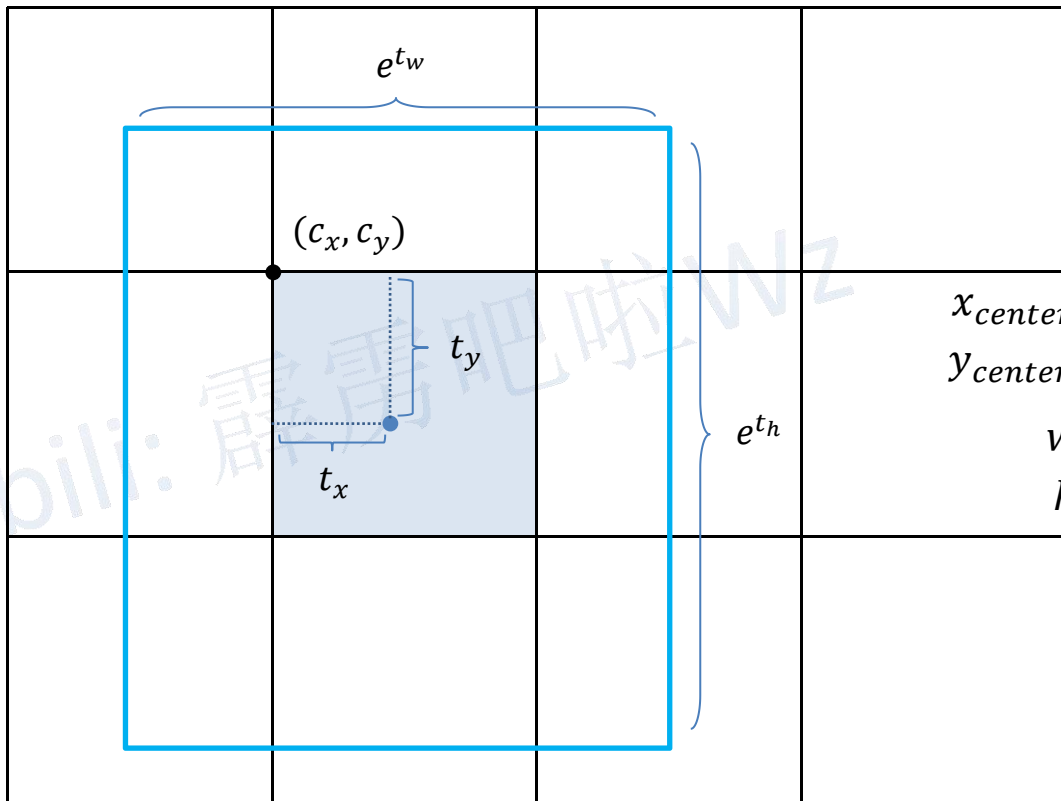
decoupled detection head

加速收敛, 提升AP

参数不共享



注意尺度



$$x_{center} = c_x + t_x$$

$$y_{center} = c_y + t_y$$

$$w = e^{t_w}$$

$$h = e^{t_h}$$

由于在网络的检测头中有**Cls.分支**、**Reg.分支**以及**IoU.分支**（其实是**Obj.分支**），所以损失由 L_{cls} 、 L_{reg} 以及 L_{obj} 这三部分组成。其中 L_{cls} 和 L_{obj} 采用的都是二值交叉熵损失（BCELoss）而 L_{reg} 采用的是IoULoss。还要注意的， L_{cls} 以及 L_{reg} 只计算正样本的损失，而 L_{obj} 既计算正样本也计算负样本的损失。

$$Loss = \frac{L_{cls} + \lambda L_{reg} + L_{obj}}{N_{pos}}$$

- L_{cls} 代表分类损失
- L_{reg} 代表定位损失
- L_{obj} 代表obj损失
- λ 代表定位损失的平衡系数，源码中设置是5.0
- N_{pos} 代表被分为正样的Anchor Point数

SimOTA是由OTA (Optimal Transport Assignment) 简化得到的, OTA也是旷视科技同年出的一篇文章, 论文名称叫做《Optimal transport assignment for object detection》目的是将匹配正负样本的过程看成一个最优传输问题。

Methods	AP (%)	Parameters	GFLOPs	Latency	FPS
YOLOv3-ultralytics ²	44.3	63.00 M	157.3	10.5 ms	95.2
YOLOv3 baseline	38.5	63.00 M	157.3	10.5 ms	95.2
+decoupled head	39.6 (+1.1)	63.86 M	186.0	11.6 ms	86.2
+strong augmentation	42.0 (+2.4)	63.86 M	186.0	11.6 ms	86.2
+anchor-free	42.9 (+0.9)	63.72 M	185.3	11.1 ms	90.1
+multi positives	45.0 (+2.1)	63.72 M	185.3	11.1 ms	90.1
+SimOTA	47.3 (+2.3)	63.72 M	185.3	11.1 ms	90.1
+NMS free (optional)	46.5 (-0.8)	67.27 M	205.1	13.5 ms	74.1



★: 城市

●: 牛奶生产基地

在SimOTA正负样本匹配过程中，**城市**对应的是**每个样本**（对应论文中的**anchor point**，其实就是grid网格中的每个cell），**牛奶生产基地**对应的是标注好的**GT Bbox**，那现在的**目标是怎样以最低的成本（cost）将GT分配给对应的样本**。根据论文中的公式1，cost的计算公式如下，其中 λ 为平衡系数，代码中设置的是3.0：

$$c_{ij} = L_{ij}^{cls} + \lambda L_{ij}^{reg}$$

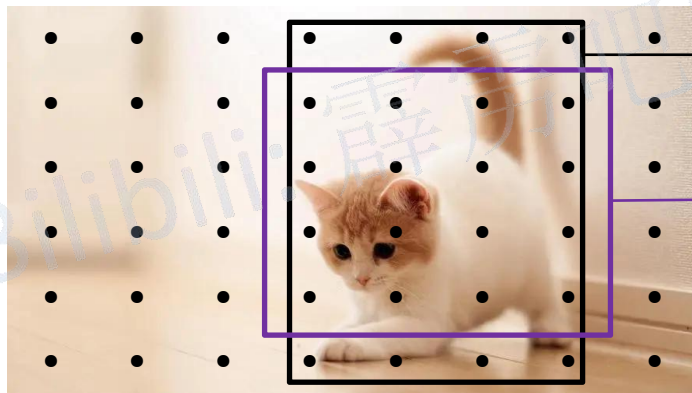
最小化cost可以理解为让网络以最小的学习成本学习到有用的知识

feature map
(grid)

			✓	✓	✓	✓	
			✓	✓	✓	✓	
			✓	✓	✓	✓	
			✓	✓	✓	✓	
			✓	✓	✓	✓	
			✓	✓	✓	✓	

$$c_{ij} = L_{ij}^{cls} + \lambda L_{ij}^{reg}$$

```
cost = (
    pair_wise_cls_loss
    + 3.0 * pair_wise_ious_loss
    + 100000.0 * (~is_in_boxes_and_center)
)
```



→ GT box

→ fixed center area

(center_radius = 2.5)

	A1	A2	A3	A4	A5	A6
GT1	0.1	0.2	1.2	2.9	0.3	0.5
GT2	1.5	7.7	0.2	0.1	0.2	8.6

Anchor Point与
GT的cost矩阵

	A1	A2	A3	A4	A5	A6
GT1	0.9	0.8	0.1	0	0.7	0.2
GT2	0.3	0.1	0.7	0.9	0.8	0.2

Anchor Point与
GT的IoU矩阵

根据IoU选取前 $n_candidate_k$ 个Anchor Point

```
n_candidate_k = min(10, ious_in_boxes_matrix.size(1))  
topk_ious, _ = torch.topk(ious_in_boxes_matrix, n_candidate_k, dim=1)
```

	A1	A2	A3	A4	A5	A6
GT1	0.1	0.2	1.2	2.9	0.3	0.5
GT2	1.5	7.7	0.2	0.1	0.2	8.6

Anchor Point与
GT的cost矩阵

	A1	A2	A3	A4	A5	A6
GT1	0.9	0.8	0.1	0	0.7	0.5
GT2	0.3	0.1	0.7	0.9	0.8	0.2

3

Anchor Point与
GT的IoU矩阵

3

计算每个GT对应Anchor Point个数 (动态)

```
dynamic_ks = torch.clamp(topk_ious.sum(1).int(), min=1)
```

	A1	A2	A3	A4	A5	A6
GT1	0.1	0.2	1.2	2.9	0.3	0.5
GT2	1.5	7.7	0.2	0.1	0.2	8.6

Anchor Point与
GT的cost矩阵

	A1	A2	A3	A4	A5	A6
GT1	1	1	0	0	1	0
GT2	0	0	1	1	1	0

3

3

Anchor Point
分配矩阵

根据dynamic_ks以及cost分配对应的Anchor Point

`dynamic_ks = torch.clamp(topk_ious.sum(1).int(), min=1)`

	A1	A2	A3	A4	A5	A6
GT1	0.1	0.2	1.2	2.9	0.3	0.5
GT2	1.5	7.7	0.2	0.1	0.2	8.6

Anchor Point与
GT的cost矩阵

	A1	A2	A3	A4	A5	A6
GT1	1	1	0	0	0	0
GT2	0	0	1	1	1	0

Anchor Point
分配矩阵

如果多个GT同时分配给一个Anchor Point, 那么只取cost最小的GT