深度学习-目标检测篇

作者：神秘的wz

YOLOv3 SPP

Mosaic图像增强

SPP模块

CIOU Loss

Focal loss

Model	Size	COCO mAP @0.5:0.95	COCO mAP @0.5
YOLOv3	512	32.7	57.7
YOLOv3-SPP	512	35.6	59.5
YOLOv3-SPP-ultralytics	**512**	**42.6**	**62.4**

YOLOv3 SPP

Mosaic图像增强

增加数据的多样性
增加目标个数
BN能一次性统计 多张图片的参数

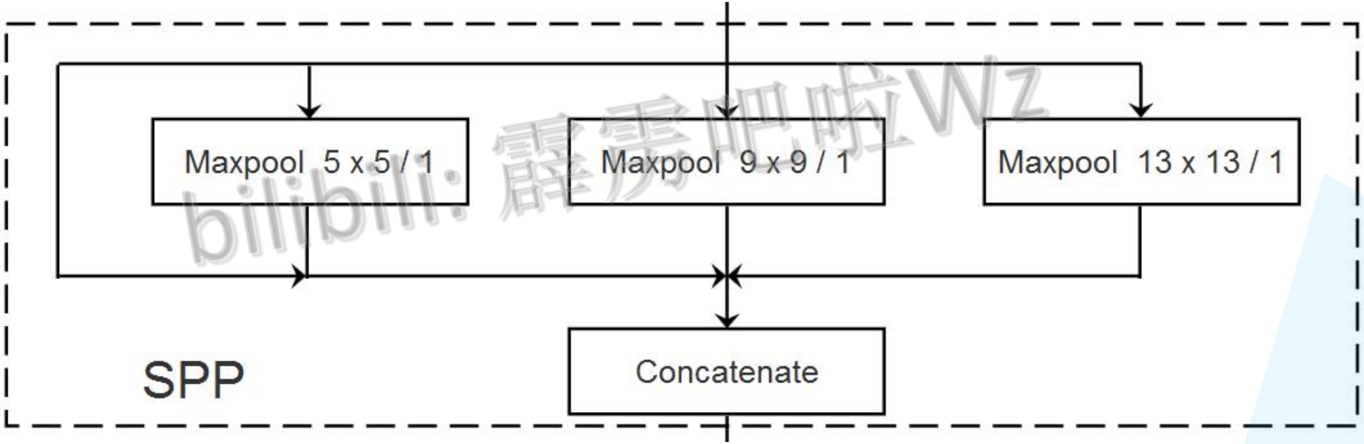
YOLOv3 SPP

SPP模块

实现了不同尺度的特征融合

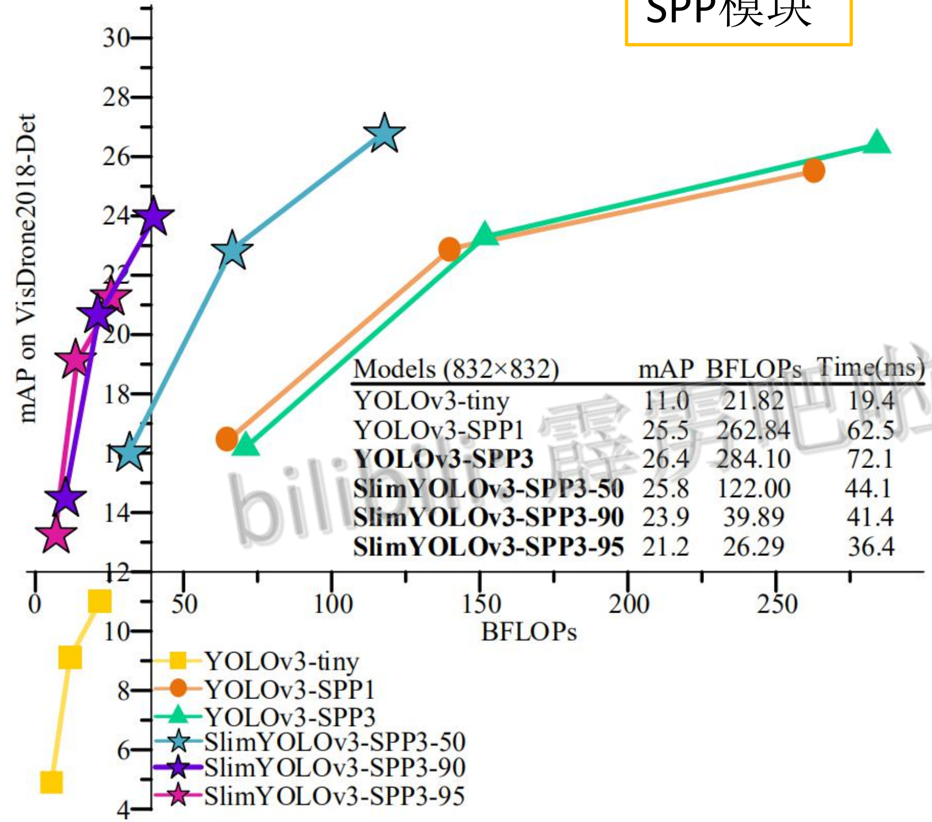
注意：这里的SPP和SPPnet中的SPP结构不一样

Spatial Pyramid Pooling



YOLOv3 SPP

SPP模块



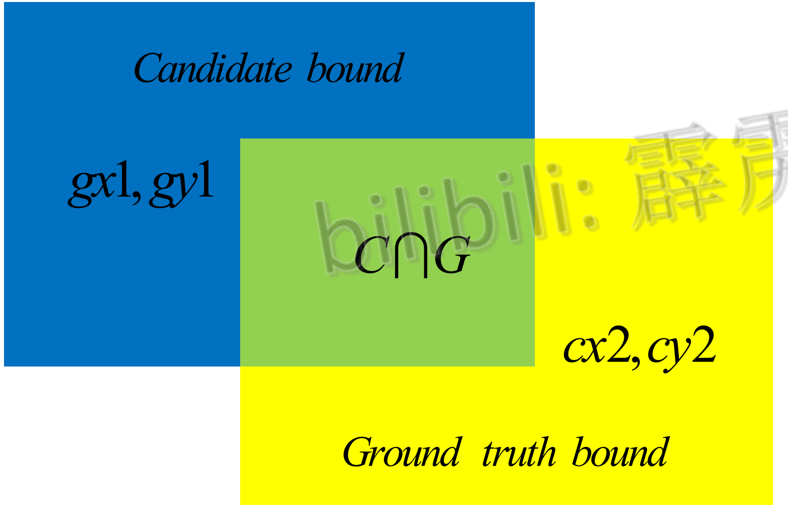
Models (832×832)	mAP	BFLOPs	Time(ms)
YOLOv3-tiny	11.0	21.82	19.4
YOLOv3-SPP1	25.5	262.84	62.5
YOLOv3-SPP3	26.4	284.10	72.1
SlimYOLOv3-SPP3-50	25.8	122.00	44.1
SlimYOLOv3-SPP3-90	23.9	39.89	41.4
SlimYOLOv3-SPP3-95	21.2	26.29	36.4

在多个yolo layer前
加入了SPP模块

YOLOv3 SPP

CIoU Loss

IoU Loss → GIoU Loss → DIoU Loss → CIoU Loss

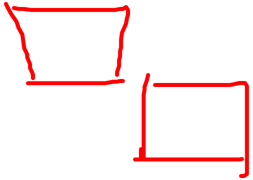


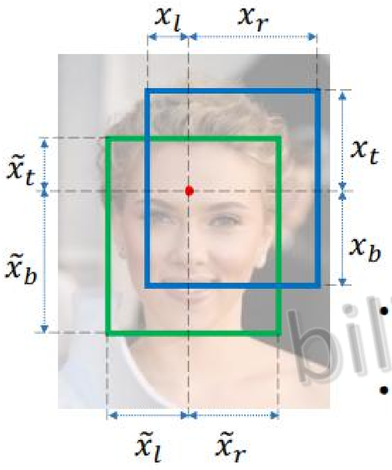
$$IoU = \frac{area(C) \cap area(G)}{area(C) \cup area(G)}$$

推荐博文：https://zhuanlan.zhihu.com/p/94799295

YOLOv3 SPP

IoU Loss



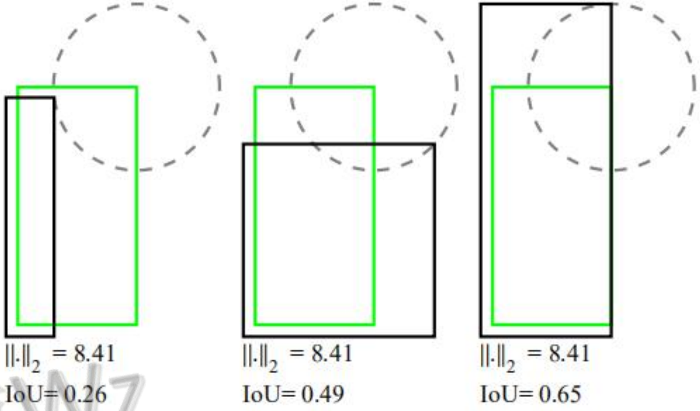


Ground truth: $\tilde{x} = (\tilde{x}_t, \tilde{x}_b, \tilde{x}_l, \tilde{x}_r)$

Prediction: $x = (x_t, x_b, x_l, x_r)$

- ℓ_2 $loss = ||\square - \square||_2^2$
- $IoU\ loss = -\ln \frac{Intersection(\square, \square)}{Union(\square, \square)}$

Figure 1: Illustration of IoU loss and ℓ_2 loss for pixel-wise bounding box prediction.



优点

1. 能够更好的反应重合程度
2. 具有尺度不变性

缺点

1. 当不相交时loss为0

YOLOv3 SPP

GIoU Loss

Generalized IoU

$$GIoU = IoU - \frac{A^c - u}{A^c}$$

$$-1 \le GIoU \le 1$$

$$L_{GIoU} = 1 - GIoU$$

$$0 \le L_{GIoU} \le 2$$

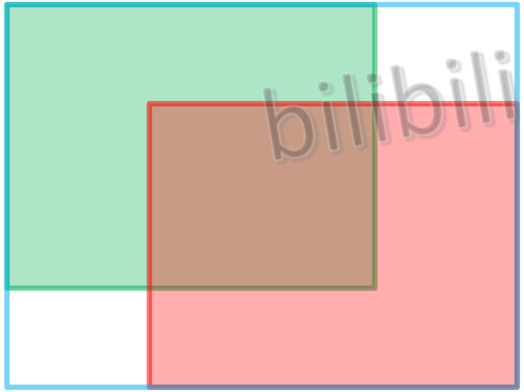


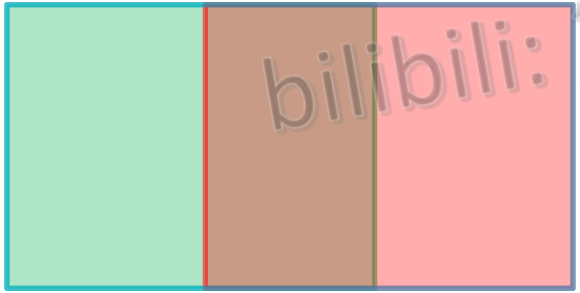
Table 1. Comparison between the performance of **YOLO v3** [21] trained using its own loss (MSE) as well as $\mathcal{L}_{IoU}$ and $\mathcal{L}_{GIoU}$ losses. The results are reported on the **test set of PASCAL VOC 2007.**

Loss / Evaluation	AP		AP75	
	IoU	**GIoU**	**IoU**	**GIoU**
MSE [21]	.461	.451	.486	.467
$\mathcal{L}_{IoU}$	.466	.460	.504	.498
Relative improv %	1.08%	2.02%	3.70%	6.64%
$\mathcal{L}_{GIoU}$	**.477**	**.469**	**.513**	**.499**
Relative improv %	**3.45%**	**4.08%**	**5.56%**	**6.85%**

YOLOv3 SPP

GIoU Loss

$$GIoU = IoU - \frac{A^c - u}{A^c}$$



GIoU退化成IoU

YOLOv3 SPP

DIoU Loss

Distance-IoU

L_{IoU} **Slow Convergence**

L_{GIoU} **Inaccurate Regression**

如何更快的收敛？
如何达到更高的定位精度？

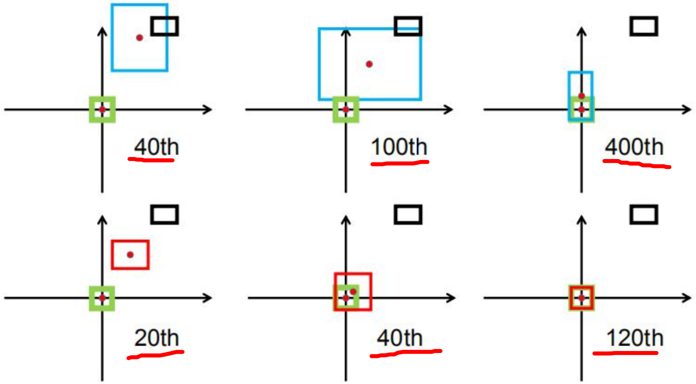
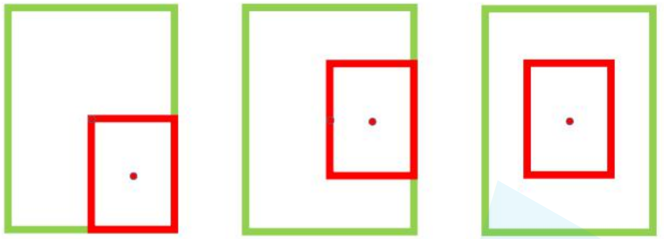


Figure 1: Bounding box regression steps by GIoU loss (first row) and DIoU loss (second row). Green and **black** denote target box and **anchor** box, respectively. Blue and red denote predicted boxes for GIoU loss and DIoU loss, respectively. GIoU loss generally increases the size of predicted box to overlap with target box, while DIoU loss directly minimizes normalized distance of central points.



$\mathcal{L}_{IoU} = 0.75$	$\mathcal{L}_{IoU} = 0.75$	$\mathcal{L}_{IoU} = 0.75$
$\mathcal{L}_{GIoU} = 0.75$	$\mathcal{L}_{GIoU} = 0.75$	$\mathcal{L}_{GIoU} = 0.75$
$\mathcal{L}_{DIoU} = 0.81$	$\mathcal{L}_{DIoU} = 0.77$	$\mathcal{L}_{DIoU} = 0.75$

Figure 2: GIoU loss degrades to IoU loss for these cases, while our DIoU loss is still distinguishable. Green and red denote target box and predicted box respectively.

YOLOv3 SPP

DIoU Loss

$$DIoU = IoU - \frac{\rho^2(b, b^{gt})}{c^2} = IoU - \frac{d^2}{c^2}$$

$$-1 \le DIoU \le 1$$

DIoU损失能够直接最小化两个boxes之间的距离，因此收敛速度更快。

$$L_{DIoU} = 1 - DIoU$$

$$0 \le L_{DIoU} \le 2$$

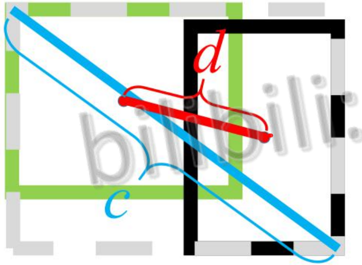


Figure 5: DIoU loss for bounding box regression, where the normalized distance between central points can be directly minimized. c is the diagonal length of the smallest enclosing box covering two boxes, and $d = \rho(\mathbf{b}, \mathbf{b}^{gt})$ is the distance of central points of two boxes.

Loss / Evaluation	AP	
	IoU	**GIoU**
$\mathcal{L}_{IoU}$	46.57	45.82
$\mathcal{L}_{GIoU}$	47.73	46.88
Relative improv. %	2.49%	2.31%
$\mathcal{L}_{DIoU}$	48.10	47.38
Relative improv. %	3.29%	3.40%

YOLOv3 SPP

CIoU Loss

Complete-IoU

$$CIoU = IoU - (\frac{\rho^2(b, b^{gt})}{c^2} + \alpha\upsilon)$$

$$\upsilon = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2$$

$$\alpha = \frac{\upsilon}{(1 - IoU) + \upsilon}$$

一个优秀的回归定位损失应该考虑到3种几何参数：

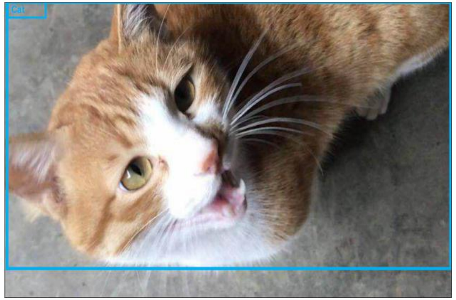
重叠面积 中心点距离 长宽比

$$L_{CIoU} = 1 - CIoU$$

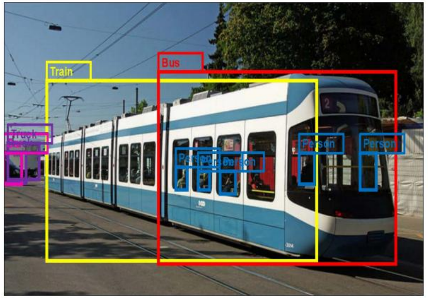
Loss / Evaluation	AP		AP75	
	IoU	GIoU	IoU	GIoU
$\mathcal{L}_{IoU}$	46.57	45.82	49.82	48.76
$\mathcal{L}_{GIoU}$	47.73	46.88	52.20	51.05
Relative improv. %	2.49%	2.31%	4.78%	4.70%
$\mathcal{L}_{DIoU}$	48.10	47.38	52.82	51.88
Relative improv. %	3.29%	3.40%	6.02%	6.40%
$\mathcal{L}_{CIoU}$	**49.21**	**48.42**	**54.28**	**52.87**
Relative improv. %	**5.67%**	**5.67%**	**8.95%**	**8.43%**
$\mathcal{L}_{CIoU}(D)$	**49.32**	**48.54**	**54.74**	**53.30**
Relative improv. %	**5.91%**	**5.94%**	**9.88%**	**9.31%**

YOLOv3 SPP

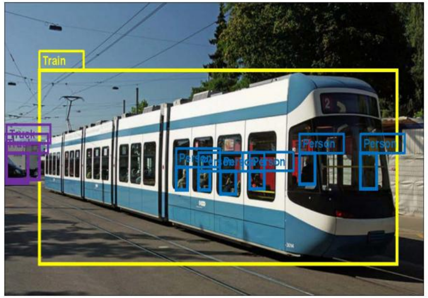
$\mathcal{L}_{GIoU}$



$\mathcal{L}_{CIoU}$

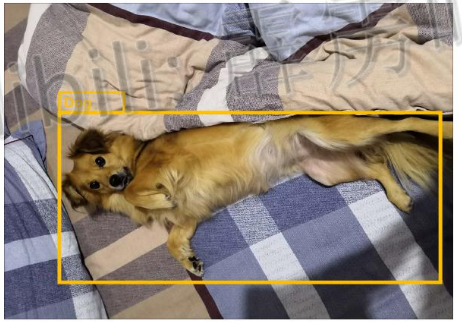


$\mathcal{L}_{GIoU}$



$\mathcal{L}_{CIoU}$

$\mathcal{L}_{GIoU}$



$\mathcal{L}_{CIoU}$

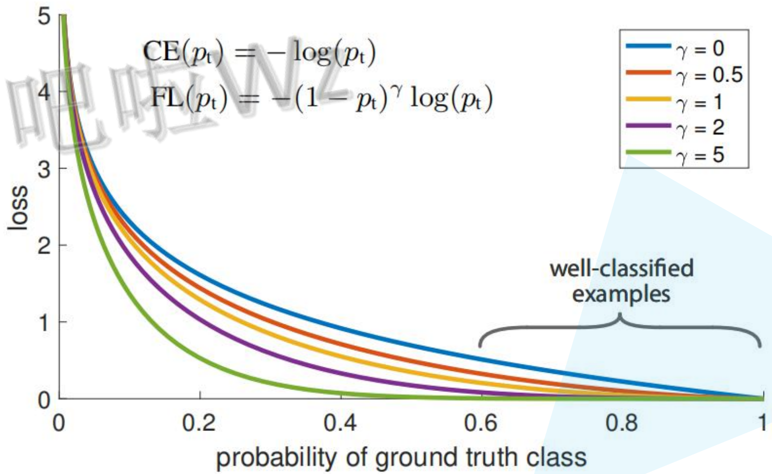
YOLOv3 SPP

Focal loss

Focal loss. We tried using focal loss. It dropped our mAP about 2 points. YOLOv3 may already be robust to the problem focal loss is trying to solve because it has separate objectness predictions and conditional class predictions. Thus for most examples there is no loss from the class predictions? Or something? We aren't totally sure.

γ	α	AP	AP_{50}	AP_{75}
0	.75	31.1	49.4	33.0
0.1	.75	31.4	49.9	33.1
0.2	.75	31.9	50.7	33.4
0.5	.50	32.9	51.7	35.2
1.0	.25	33.7	52.0	36.2
2.0	.25	**34.0**	**52.5**	**36.5**
5.0	.25	32.2	49.6	34.8

(b) **Varying γ for FL** (w. optimal α)



YOLOv3 SPP

Focal loss

One-stage object detection model

Class Imbalance

一张图像中能够匹配到目标的候选框(正样本)个数一般只有十几个或几十个，而没匹配到的候选框(负样本)大概有10^4–10^5个。

在这10^4–10^5个未匹配到目标的候选框中大部分都是简单易分的负样本(对训练网络起不到什么作用，但由于数量太多会淹没掉少量但有助于训练的样本)。

50x3=150

100,000x0.1=10,000

degenerate models

hard negative mining

method	batch size	nms thr	AP	AP_{50}	AP_{75}
OHEM	128	.7	31.1	47.2	33.2
OHEM	256	.7	31.8	48.8	33.9
OHEM	512	.7	30.6	47.0	32.6
OHEM	128	.5	32.8	50.3	35.1
OHEM	256	.5	31.0	47.4	33.0
OHEM	512	.5	27.6	42.0	29.2
OHEM 1:3	128	.5	31.1	47.2	33.2
OHEM 1:3	256	.5	28.3	42.4	30.3
OHEM 1:3	512	.5	24.0	35.5	25.8
FL	n/a	n/a	**36.0**	**54.9**	**38.7**

(d) **FL** *vs.* **OHEM** baselines (with ResNet-101-FPN)

YOLOv3 SPP

3. Focal Loss

The *Focal Loss* is designed to address the one-stage object detection scenario in which there is an extreme imbalance between foreground and background classes during training (*e.g.*, 1:1000). We introduce the focal loss starting from the cross entropy (CE) loss for binary classification[1]:

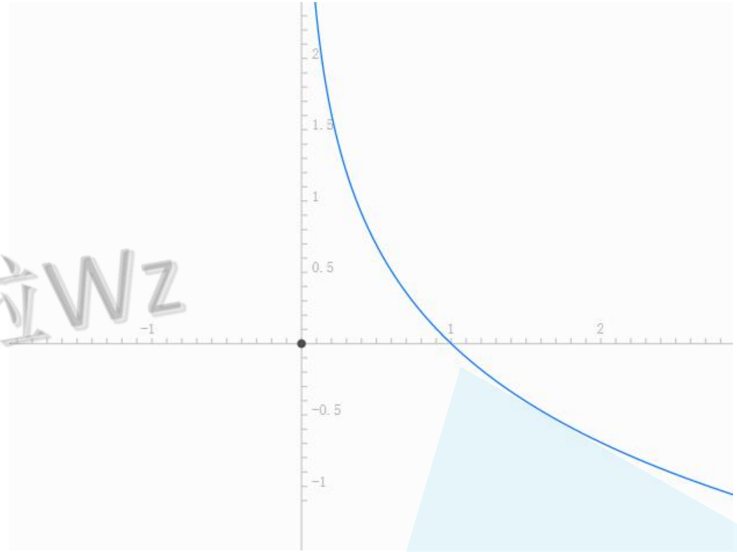
$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases} \tag{1}$$

In the above $y \in \{\pm 1\}$ specifies the ground-truth class and $p \in [0, 1]$ is the model's estimated probability for the class with label $y = 1$. For notational convenience, we define p_t:

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise,} \end{cases} \tag{2}$$

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

$$y = -\ln(x)$$



YOLOv3 SPP

3.1. Balanced Cross Entropy

A common method for addressing class imbalance is to introduce a weighting factor $\alpha \in [0, 1]$ for class 1 and $1-\alpha$ for class -1. In practice α may be set by inverse class frequency or treated as a hyperparameter to set by cross validation. For notational convenience, we define α_t analogously to how we defined p_t. We write the α-balanced CE loss as:

$$\mathrm{CE}(p_t) = -\alpha_t \log(p_t). \tag{3}$$

This loss is a simple extension to CE that we consider as an experimental baseline for our proposed focal loss.

α	AP	AP_{50}	AP_{75}
.10	0.0	0.0	0.0
.25	10.8	16.0	11.7
.50	30.2	46.7	32.8
.75	31.1	49.4	33.0
.90	30.8	49.7	32.3
.99	28.7	47.4	29.9
.999	25.1	41.7	26.1

(a) **Varying α for CE loss** ($\gamma = 0$)

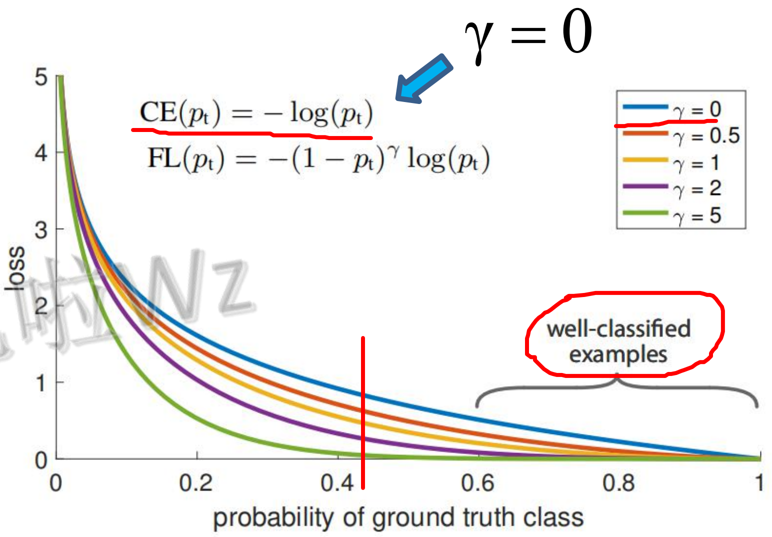
YOLOv3 SPP

3.2. Focal Loss Definition

As our experiments will show, the large class imbalance encountered during training of dense detectors overwhelms the cross entropy loss. Easily classified negatives comprise the majority of the loss and dominate the gradient. While α balances the importance of positive/negative examples, it does not differentiate between easy/hard examples. Instead, we propose to reshape the loss function to down-weight easy examples and thus focus training on hard negatives.

More formally, we propose to add a modulating factor $(1-p_t)^\gamma$ to the cross entropy loss, with tunable *focusing* parameter $\gamma \geq 0$. We define the focal loss as:

$$\mathrm{FL}(p_t) = -(1-p_t)^\gamma \log(p_t). \tag{4}$$



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

$(1-p_t)^\gamma$ 能够降低易分样本的损失贡献

YOLOv3 SPP

Focal loss

$$FL(p)\begin{cases}-\alpha(1-p)^{\gamma}\log(p) & \text{if } y=1\\ -(1-\alpha)p^{\gamma}\log(1-p) & \text{otherwise,}\end{cases}$$

In practice we use an α-balanced variant of the focal loss:

$$\mathrm{FL}(p_{\mathrm{t}}) = -\alpha_{\mathrm{t}}(1-p_{\mathrm{t}})^{\gamma}\log(p_{\mathrm{t}}). \tag{5}$$

We adopt this form in our experiments as it yields slightly improved accuracy over the non-α-balanced form. Finally, we note that the implementation of the loss layer combines the sigmoid operation for computing p with the loss computation, resulting in greater numerical stability.

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

γ	α	AP	AP_{50}	AP_{75}
0	.75	31.1	49.4	33.0
0.1	.75	31.4	49.9	33.1
0.2	.75	31.9	50.7	33.4
0.5	.50	32.9	51.7	35.2
1.0	.25	33.7	52.0	36.2
2.0	.25	**34.0**	**52.5**	**36.5**
5.0	.25	32.2	49.6	34.8

(b) **Varying γ for FL** (w. optimal α)

YOLOv3 SPP

Focal loss

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

$$FL(p)\begin{cases}-\alpha(1-p)^{\gamma}\log(p) & \text{if } y=1\\ -(1-\alpha)p^{\gamma}\log(1-p) & \text{otherwise,}\end{cases}$$

p	y	CE	FL	rate
0.9	1	0.105	0.00026	400
0.968	1	0.033	0.000008	3906
0.1	0	0.105	0.00079	133
0.032	0	0.033	0.000025	1302
0.1	1	2.3	0.466	4.9
0.9	0	2.3	1.4	1.6

易受噪音干扰

沟通方式

1.github

https://github.com/WZMIAOMIAO/deep-learning-for-image-processing

2.bilibili

https://space.bilibili.com/18161609/channel/index

3.CSDN

https://blog.csdn.net/qq_37541097/article/details/103482003

尽可能每周更新