YOLOv3: 进一步地提升

Joseph Redmon Ali Farhadi
University of Washington

摘 要

我们向 YOLO 提供一些更新！ 我们进行了一些小的设计更改，以使其更好。我们还训练了这个非常膨胀的新网络。它比上次有点大，但检测更准确。 它仍然很快，不用担心。在 320 × 320， YOLOv3 以 22.2 毫秒的速度达到了 28.2mAP 的精度，与 SSD 一样准确，但速度提高了三倍。当我们查看旧的 .5 IOU mAP 检测指标时 YOLOv3 也非常好。 它在 Titan X 上在 51 毫秒内达到了 57.9 的 AP_{50}，相比于 RetinaNet 在 198 毫秒内达到了 57.5 的 AP_{50}，性能相似，但速度提高了 3.8。 与往常一样，所有代码都在 https://pjreddie.com/yolo/。

1. 介绍

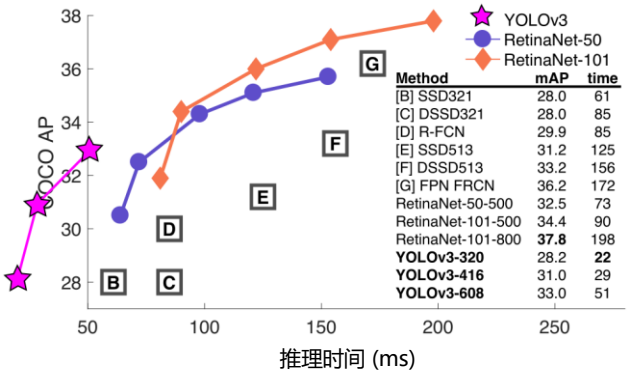
你知道，有时你只需要玩一年手机。我今年没有做过很多研究。在 Twitter 上花了很多时间。和 GAN 一起玩了一下。去年我留下了一点动力[12] [1];我设法对 YOLO 做了一些改进。但是，说实话，没有什么能更令人提起兴趣的了，只是一堆小改变让它变得更好。我也帮助了其他人的研究。

实际上，这就是我们今天来到这里的原因。我们有一个“相机就绪”截止日期[4]，我们需要引用一些我对 YOLO 进行的随机更新，但我们没有资源。所以准备好进行技术报告！

技术报告的好处是他们不需要介绍，你们都知道我们来这儿的理由。因此，此次介绍的结束将为本文的其余部分提供指示。首先，我们将告诉您与 YOLOv3 的协定。然后我们会告诉您我们是怎么做的。我们还会告诉你一些我们尝试过但不起作用的事情。最后，我们将考虑这一切意味着什么。

2. 协定

所以这是与 YOLOv3 的协定：我们主要是从其他人那里获得了很好的想法。 我们还训练了一个比其他分类器网络更好的新分类器网络。 我们将从头开始带您浏览整个系统，以便您可以全面了解它。



Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

图 1。我们从 Focal Loss 论文[9]中调整了这个数字。YOLOv3 的运行速度明显快于其他具有可比性能的检测方法。来自 M40 或 Titan X 的时间，它们基本上是相同的 GPU。

2.1. 边界框预测

在 YOLO9000 之后，我们的系统使用维度聚类作为锚框来预测边界框[15]。 网络预测每个边界框的 4 个坐标，t_x，t_y, t_w, t_h。 如果单元格从图像的左上角偏移（c_x, c_y）并且前面的边界框具有宽度和高度p_w, p_h，则预测对应于：

$$
\begin{aligned}
b_x &= \sigma(t_x) + c_x \\
b_y &= \sigma(t_y) + c_y \\
b_w &= p_w e^{t_w} \\
b_h &= p_h e^{t_h}
\end{aligned}
$$

在训练期间，我们使用平方误差损失的总和。 如果某些坐标预测的基本事实是$\hat{t}_*$，我们的梯度是真实标签值（从真实标签框计算）减去我们的预测：$\hat{t}_* - t_*$。 通过反转上面的等式可以容易地计算该真实值。

YOLOv3 使用逻辑回归预测每个边界框的对象得分。 如果边界框先前与真实对象重叠超过任何其他边界框，则该值应为 1。 如果前面的边界框

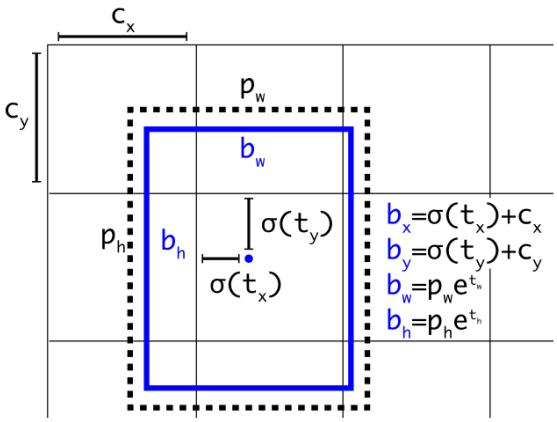


图 2.具有尺寸先验和位置预测的边界框。 我们预测盒子的宽度和高度作为聚类质心的偏移量。 我们使用 sigmoid 函数预测盒子相对于过滤器应用程序位置的中心坐标。 这个数字显然是自我抄袭的[15]。

不是最好但是与真实对象重叠超过某个阈值，我们会忽略预测，如[17]。 我们使用阈值 .5。 与[17]不同，我们的系统只为每个真实对象分配一个边界框。 如果先前的边界框未分配给真实对象，则它不会导致坐标或类预测的损失，只会导致对象损失。

2.2. 类预测

每个框使用多标记分类预测边界框可能包含的类。我们不使用 softmax，因为我们发现它不需要良好的性能，而只是使用独立的逻辑分类器。在训练期间，我们使用二元交叉熵损失进行类预测。

当我们转向更复杂的领域（如 Open Images Dataset [7]）时，这个公式会有所帮助。在此数据集中有许多重叠标签（即女人和人）。使用 softmax 假设每个盒子只有一个类，而通常不是这种情况。多标签方法可以更好地模拟数据。

2.3. 跨尺度的预测

YOLOv3 预测 3 种不同尺度的方框。我们的系统使用与金字塔网络相似的概念从这些尺度中提取特征[8]。从我们的基本特征提取器中，我们添加了几个卷积层。最后一个预测了 3-d 张量编码边界框，对象和类预测。在我们使用 COCO [10]的实验中，我们预测每个尺度有 3 个框，因此对于 4 个边界框偏移，1 个对象预测和 80 个类预测，张量为 $N \times N \times [3 * (4 + 1 + 80)]$。

接下来，我们从前面的 2 个图层中获取特征图，然后将其上采样至 2 倍。我们还从网络中较早的位置获取了一个特征图，并使用连接将其与我们的上采样功能合并。这种方法允许我们从上采样特征中获取更有意义的语义信息，并从先前的特征图中获得更细粒度的信息。然后我们再添加一些卷积层来处理这个组合特征图，并最终预测出类似的张量，尽管现在是两倍大小。

我们再次执行相同的设计来预测最终尺度的方框。因此，我们对第 3 级尺度的预测受益于所有先前的计算以及网络早期的细粒度特征。

我们仍然使用 k-means 聚类来确定我们的边界框先验。我们只是任意选择 9 个簇和 3 个尺度，然后在尺度上均匀地划分簇。在 COCO 数据集上，9 个集群是：（10 13），（16 30），（33 23），（30 61），（62 45），（59 119），（116 90），（156 198），（373 326）。

2.4. 特征提取器

我们使用新网络执行特征提取。我们的新网络 YOLOv2，Darknet-19 和那个新奇的残差网络之间的混合网络。 我们的网络使用连续的 3 × 3 和 1× 1 卷积层，但现在也有一些快捷连接，并且明显变得更大。 它有 53 个卷积层，所以我们称之为....噔噔噔..... Darknet-53！

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

表 1. Darknet-53.

这个新网络比 Darknet-19 强大得多，但仍然比 ResNet-101 或 ResNet-152 更有效。 以下是一些 ImageNet 结果：

Backbone	Top-1	Top-5	Bn Ops	BFLOP/s	FPS
Darknet-19 [15]	74.1	91.8	7.29	1246	**171**
ResNet-101[5]	77.1	93.7	19.7	1039	53
ResNet-152 [5]	**77.6**	**93.8**	29.4	1090	37
Darknet-53	77.2	**93.8**	18.7	**1457**	78

表 2.骨干（backbones）的比较。 准确性，数十亿次操作，每秒十亿次浮点运算，以及各种网络的 FPS。

每个网络都使用相同的集合进行训练，并以 256 × 256 单一裁剪精度进行测试。运行时间在 Titan X 上按 256 × 256 测量。因此，Darknet-53 的性能与最先进的分类器相当，但浮点运算更少，速度更快。Darknet-53 优于 ResNet-101，速度提高 1.5 倍。 Darknet-53 与 ResNet-152 具有相似的性能，速度提高 2 倍。

Darknet-53 还实现了每秒最高的测量浮点运算。这意味着网络结构可以更好地利用 GPU，从而提高计算效率，从而提高速度。这主要是因为 ResNets 的层数太多而且效率不高。

2.5. 训练

我们仍然训练完整的图像，没有困难负例挖掘或任何这些东西。我们使用多尺度训练，大量数据扩充，批量归一化，所有标准的东西。我们使用 Darknet 神经网络框架进行训练和测试[14]。

3. 我们如何做

YOLOv3 非常棒！参见表 3。就 COCO 而言，奇怪的平均 mAP 指标与 SSD 变体相当，但速度提高了 3 倍。尽管它仍然有点落后于其他的像 RetinaNet 这样的模型，在这个指标中。

但是，当我们在 IOU = .5（或图表中的AP_{50}）中查看 mAP 的“旧”检测度量时，YOLOv3 非常强大。它几乎与 RetinaNet 相当，远远超过 SSD 变体。这表明 YOLOv3 是一种非常强大的探测器，擅长为物体生产合适的边界框。然而，随着 IOU 阈值的增加，性能显著下降，表明 YOLOv3 努力使框与物体完美对齐。

在过去，YOLO 与小物件斗争。但是，现在我们看到了这种趋势的逆转。通过新的多尺度预测，我们看到 YOLOv3 具有相对较高的 AP_S 性能。但是，它在中型和大型物体上的性能相对较差。需要更多的研究来解决这个问题。

当我们在AP_{50}指标上绘制准确度与速度的关系时（见图 5），我们看到 YOLOv3 比其他检测系统具有显著的优势。也就是说，它更快更好。

4. 我们尝试过的但不起作用的事情

我们在 YOLOv3 上工作时尝试了很多东西。很多都行不通。下面是我们能记住的东西。

边界框 x，y 偏移预测。我们尝试使用普通的锚框预测机制，即使用线性激活偏移为框宽度或高度的倍数来预测 x，y。我们发现这种方法降低了模型的稳定性并且效果不佳。

线性 x，y 预测而不是逻辑激活。我们尝试使用线性激活来直接预测 x，y 偏移而不是逻辑激活。这导致了 mAP 的几个点下降。

焦点损失。我们尝试使用焦点丢失。它将我们的 mAP 降低了大约 2 个点。 YOLOv3 可能已经对焦点损失试图解决的问题具有鲁棒性，因为它具有独立的对象预测和条件类预测。因此，对于大多数例子，类预测没有损失？或者其他的东西？我们并不完全确定。

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [5]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [8]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [6]	Inception-ResNet-v2 [21]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [20]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [15]	DarkNet-19 [15]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [11, 3]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [3]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [9]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [9]	ResNeXt-101-FPN	**40.8**	**61.1**	**44.1**	**24.1**	**44.2**	51.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9

表 3。我真的只是从[9]中偷走了所有这些表，他们从头开始做太长时间。 好的，YOLOv3 做得还不错。 请记住，RetinaNet 处理图像要多用 3.8 倍的时间。 YOLOv3 比 SSD 变体要好得多，与AP_{50}指标上最先进的模型相当。

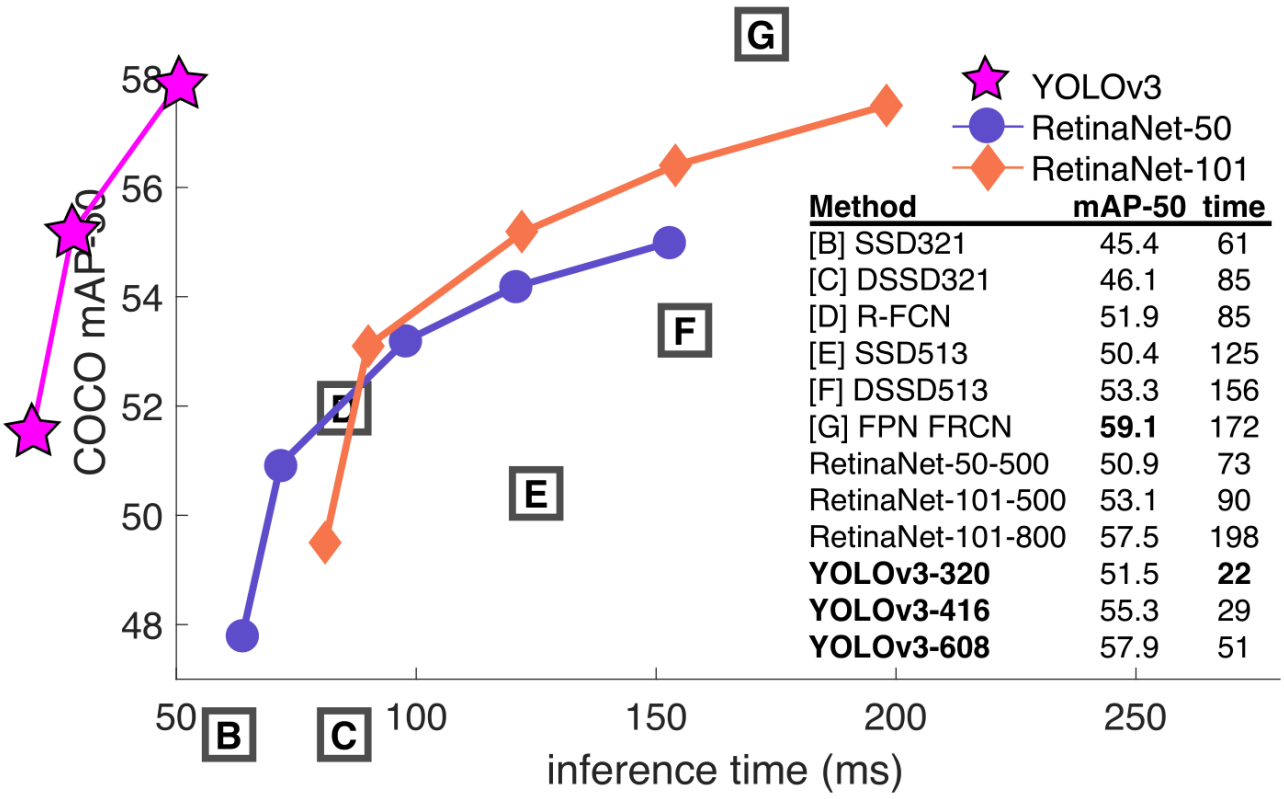


图 3。再次从[9]改编，这次显示 mAP 的速度/准确度权衡，为 .5 IOU 指标。 你可以说 YOLOv3 是好的，因为它非常高并且远在左边。 你能引用自己的论文吗？ 猜猜谁会去尝试，这家伙 → [16]。 哦，我忘记了，我们还修复了 YOLOv2 中的数据加载错误，这提升了大约 2 点 mAP。 只是这里偷偷摸摸一点不要甩掉原有布局。

双 IOU 阈值和真值分配。Faster R-CNN 在训练期间使用两个 IOU 阈值。如果一个预测与真实值重叠达到 .7 它是一个正面的例子，而处于[.3，.7]则会被忽略，小于 .3 对于所有真实物体就成为了一个反面的例子。我们尝试了类似的策略，但无法取得好成绩。

我们非常喜欢我们目前的方法，它似乎至少是当前的最佳状态。有可能这些技术中的一些最终会产生良好的结果，也许他们只需要一些调整来稳定训练。

5. 这一切意味着什么

YOLOv3 是一个很好的探测器。它很快，很准确。它在 COCO 的 mAP 处于.5 和 .95 IOU 指标之间上的表现不太好。但它对 .5 IOU 的旧检测指标非常好。

为什么我们还要切换指标？最初的 COCO 论文只有这句神秘的句子： “评估服务器完成后，将添加评估指标的完整讨论”。 Russakovsky 等人报告说，人类很难将.3 的 IOU 与.5 区分开来！ “训练人类用 IOU 为 0.3 目视检查边界框并将其与 IOU 0.5 区别开来是超乎寻常的困难。 “[18] 如果人类很难说出差异，那有多重要？

但也许一个更好的问题是： “现在我们拥有它们，我们将如何处理这些探测器？” 很多从事这项研究的人都在谷歌和 Facebook 上。我想至少我们知道技术掌握得很好，绝对不会用来收集你的个人信息并把它卖给.....等等，你说这正是它将用于什么？？哦。

那么大量资助视觉研究的人是军队，他们从来没有做过任何可怕的事情，就像用新技术杀死很多人一样哦等等..... 见页底 1。

我非常希望大多数使用计算机视觉的人只是用它来做快乐，好的事情，比如计算一个国家公园里的斑马数量[13]，或跟踪他们的猫在它们周围徘徊[19]。但是计算机视觉已经被用于质疑，作为研究人员，我们有责任至少考虑我们的工作可能造成的伤害并考虑如何减轻它。我们欠世界的那么多。

最后，不要@我。 （因为我最终退出 Twitter）。

[1] 作者由海军研究办公室和谷歌办公室资助。

参考文献

[1] Analogy. Wikipedia, Mar 2018. 1

[2] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) chal-lenge. International journal of computer vision, 88(2):303– 338, 2010. 6

[3] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659, 2017. 3

[4] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi. Iqa: Visual question answering in interactive environments. arXiv preprint arXiv:1712.03316, 2017. 1

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learn-ing for image recognition. In Proceedings of the IEEE con-ference on computer vision and pattern recognition, pages 770–778, 2016. 3

[6] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. 3

[7] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, and K. Murphy. Open-images: A public dataset for large-scale multi-label and multi-class image classification. Dataset available from https://github.com/openimages, 2017. 2

[8] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2117–2125, 2017. 2, 3

[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar´. Focal loss for dense object detection. arXiv preprint arXiv:1708.02002, 2017. 1, 3, 4

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ra-manan, P. Dollar,´ and C. L. Zitnick. Microsoft coco: Com-mon objects in context. In European conference on computer vision, pages 740–755. Springer, 2014. 2

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016. 3

[12] I. Newton. Philosophiae naturalis principia mathematica. William Dawson & Sons Ltd., London, 1687. 1

[13] J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. Rubenstein. Animal population censusing at scale with citizen science and photographic identification. 2017. 4

[14] J. Redmon. Darknet: Open source neural networks in c. http://pjreddie.com/darknet/, 2013–2016. 3

[15] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 6517–6525. IEEE, 2017. 1, 2, 3

[16] J. Redmon and A. Farhadi. Yolov3: An incremental improve-ment. arXiv, 2018. 4

[17] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: To-wards real-time object detection with region proposal net-works. arXiv preprint arXiv:1506.01497, 2015. 2

[18] O. Russakovsky, L.-J. Li, and L. Fei-Fei. Best of both worlds: human-machine collaboration for object annotation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2121–2131, 2015. 4

[19] M. Scott. Smart camera gimbal bot scanlime:027, Dec 2017. 4

[20] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta. Be-yond skip connections: Top-down modulation for object de-tection. arXiv preprint arXiv:1612.06851, 2016. 3

[21] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. 2017. 3

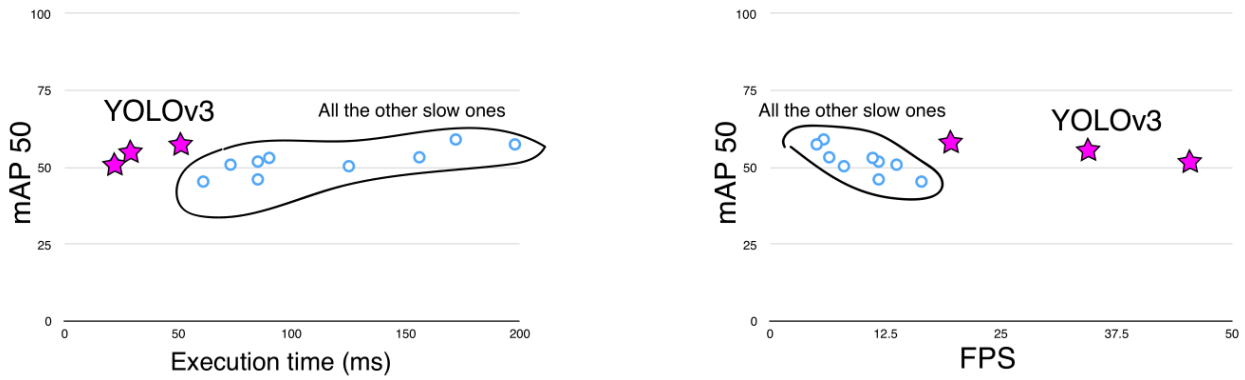


图 4.零轴图表可能更加智能诚实.....我们仍然可以调整变量来使结果看起来更好！

反驳

我们要感谢 Reddit 评论员，同学们，电子邮件发送者，以及在走廊里传递的呐喊，感谢他们可爱，心旷神怡的话语。如果你像我一样正在审查 ICCV，那么我们知道你可能还有其他 37 篇论文，你可能会一直推迟到上周，然后在该领域有一些传说，你会告诉你应该如何完成那些重新审视的观点除了它自己之外并不完全清楚它们在说什么，也许它们来自未来？无论如何，如果没有你过去自己在过去所做的所有工作，而且只是进一步前进，而不是像现在这样直到前进，那么这篇论文将不会成为现实。如果你发推文，我就不知道了。只是随便说说。

评论家＃2 AKA Dan Grossman（lol blinding 谁做的）坚持我在这里指出我们的图表只有两个非零来源。你是绝对正确的 Dan，那是因为它看起来比承认自己更好，我们都只是在这里争夺超过 2-3%的 mAP。但这是请求的图表。我也用 FPS 投了一个，因为当我们在 FPS 上进行投影时，我们看起来就像超级好。

评论家＃4 AKA JudasAdventus 在 Reddit 上写道“娱乐阅读，但反对 MS COCO 指标的论点似乎有点弱”。好吧，我一直都知道你会成为那个让我犹豫不决的人。你知道当你在一个项目上工作时它是如何出来的，所以你必须找出一些方法来证明你的实际操作是非常酷的吗？我基本上是在努力做到这一点，而且我对 COCO 指标有点猛烈抨击。但是现在我已经把这座小山盯上了，我也可能死在它上面。

看到这里的事情，mAP 已经有点破碎，所以它的更新应该可以解决它的一些问题，或者至少证明为什么更新的版本在某种程度上更好。而我认为最重要的是缺乏理由。对于 PASCAL VOC，IOU 阈值“故意设置为低，以计算真实数据中边界框中的不准确性”[2]。COCO 的标签比 VOC 好吗？这是绝对可能的，因为 COCO 有分段掩码概率标签更值得信赖，因此我们不担心不准确。但同样，我的问题是缺乏理由。

COCO 指标强调更好的边界框，但强调必须意味着它不再强调其他东西，在这种情况下分类准确性。是否有充分的理由去考虑更多精确的边界框比更好的分类更重要吗？ 错误分类的示例比稍微移位的边界框更明显。

mAP 已经被搞砸了，因为重要的是每级等级排序。 例如，如果你的测试集只有这两个图像，那么根据 mAP，产生这些结果的两个探测器就是好的：

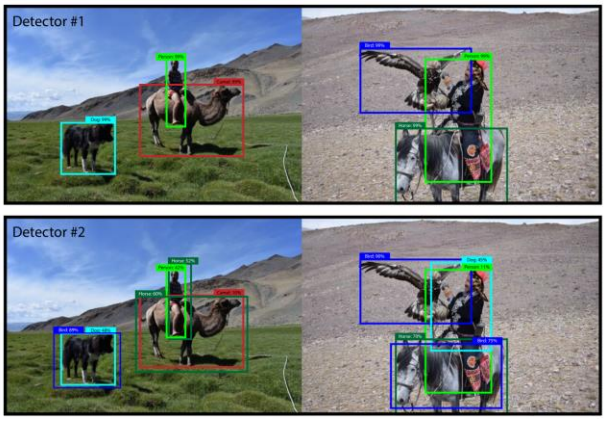


图 5.根据这两个图像的 mAP，这两个假想探测器是完美的。 它们都很完美。 完全相同。

现在这显然是对 mAP 问题的过度夸大，但我想我的新观点是，“现实世界”中人们关心的内容和我认为的当前指标之间存在明显的差异。 我们将提出新的指标，我们应该关注这些差异。 同样，它已经意味着平均精度，我们甚至称之为 COCO 指标，平均平均平均精度？

这是一个提议，人们真正关心的是图像和探测器，探测器在图像中找到并分类对象的能力如何。 如何摆脱每级 AP 并实现全局平均精度？ 或者对每个图像进行 AP 计算并对其进行平均？

无论如何，边界框都是愚蠢的，我可能是戴有面具的真正信徒，除非我不能让 YOLO 学习它们。