YOLOv3: An Incremental Improvement

Joseph Redmon, Ali Farhadi
University of Washington

Abstract

We present some updates to YOLO! We made a bunch of little design changes to make it better. We also trained this new network that's pretty swell. It's a little bigger than last time but more accurate. It's still fast though, don't worry. At 320×320 *YOLOv3 runs in 22 ms at 28.2 mAP, as accurate as SSD but three times faster. When we look at the old .5 IOU mAP detection metric YOLOv3 is quite good. It achieves* $57.9\ AP_{50}$ *in 51 ms on a Titan X, com-pared to* $57.5\ AP_{50}$ *in 198 ms by RetinaNet, similar perfor-mance but 3.8× faster. As always, all the code is online at* https://pjreddie.com/yolo/.

本文为YOLO提供了一系列更新！它包含一堆小设计，可以使系统的性能得到更新；也包含一个新训练的、非常棒的神经网络，虽然比上一版更大一些，但精度也提高了。不用担心，虽然体量大了点，它的速度还是有保障的。在输入320×320的图片后，YOLOv3能在22毫秒内完成处理，并取得28.2mAP的成绩。它的精度和SSD相当，但速度要快上3倍。和旧版数据相比，v3版进步明显。在Titan X环境下，YOLOv3的检测精度为57.9AP 50 57.9AP5057.9 AP_{50}，用时51ms；而RetinaNet的精度只有57.5AP 50 57.5AP5057.5 AP_{50}，但却需要198ms，相当于YOLOv3的3.8倍。与往常一样，所有代码均在https://pjreddie.com/yolo/。

1. Introduction

Sometimes you just kinda phone it in for a year, you know? I didn't do a whole lot of research this year. Spent a lot of time on Twitter. Played around with GANs a little. I had a little momentum left over from last year [12] [1]; I managed to make some improvements to YOLO. But, hon-estly, nothing like super interesting, just a bunch of small changes that make it better. I also helped out with other people's research a little.

Actually, that's what brings us here today. We have a camera-ready deadline [4] and we need to cite some of the random updates I made to YOLO but we don't have a source. So get ready for a TECH REPORT!

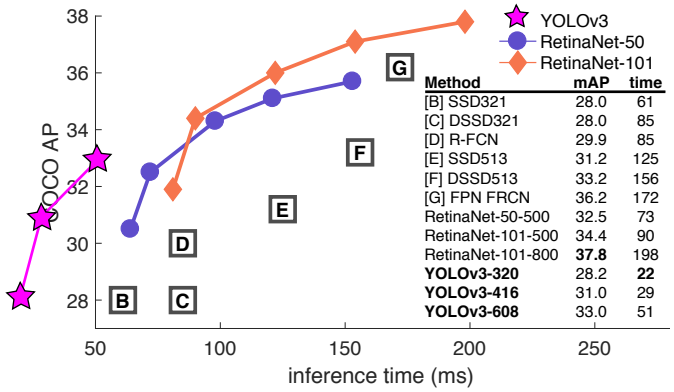
The great thing about tech reports is that they don't need intros, y'all know why we're here. So the end of this intro-duction will signpost for the rest of the paper. First we'll tell you what the deal is with YOLOv3. Then we'll tell you how we do. We'll also tell you about some things we tried that didn't work. Finally we'll contemplate what this all means.

有时候，一年你主要只是在打电话，你知道吗？今年我没有做很多研究。我在Twitter上花了很多时间。玩了一下GANs。去年我留下了一点点的动力[10] [1]；我设法对YOLO进行了一些改进。但是诚然，没有什么比这超级有趣的了，只是一小堆（bunch）改变使它变得更好。我也帮助了其他人的做一些研究。 其实，这就是今天带给我们的。所以为技术报告做准备！ 关于技术报告的好处是他们不需要介绍，你们都知道我们为什么来到这里。因此，这篇介绍性文章的结尾将为本文的其余部分提供signpost。首先我们会告诉你YOLOv3的详细内容。然后我们会告诉你我们是怎么做的。我们还会告诉你我们尝试过的一些没有奏效的事情。最后，我们将考虑这一切意味着什么。

2. The Deal

So here's the deal with YOLOv3: We mostly took good ideas from other people. We also trained a new classifier network that's better than the other ones. We'll just take you through the whole system from scratch so you can un-derstand it all.

谈到YOLOv3的更新情况，其实大多数时候我们就是直接把别人的好点子拿来用了。我们还训练了一个全新的、比其他网络更好的分类网络。为了方便你理解，让我们从头开始慢慢介绍。



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

Figure 1. We adapt this figure from the Focal Loss paper [9]. YOLOv3 runs significantly faster than other detection methods with comparable performance. Times from either an M40 or Titan X, they are basically the same GPU.

2.1. Bounding Box Prediction

Following YOLO9000 our system predicts bounding boxes using dimension clusters as anchor boxes [15]. The network predicts 4 coordinates for each bounding box, t_x, t_y, t_w, t_h. If the cell is offset from the top left corner of the image by (c_x, c_y) and the bounding box prior has width and height p_w, p_h, then the predictions correspond to:

在YOLO9000后，我们的系统开始用dimension clusters固定anchor box来选定边界框。神经网络会为每个边界框预测4个坐标：tx、ty、tw、th。如果目标cell距离图像左上角的边距是（cx, cy），且它对应边界框的宽和高为pw、ph，那么网络的预测值会是：

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

During training we use sum of squared error loss. If the ground truth for some coordinate prediction is $*$ our gra-dient is the ground truth value (computed from the ground Aruth box) minus Aur prediction: $* - t*$. This Around Aruth Aalue can be Aasily computed by Anverting Ahe Aquations above.

在训练期间，我们会计算方差。如果预测坐标的ground truth是t^，那相应的梯度就是ground truth值和预测值的差：t^-t*。利用上述公式，我们能轻松推出这个结论。通过对上面的公式变形，可以很容易地计算这个ground truth。

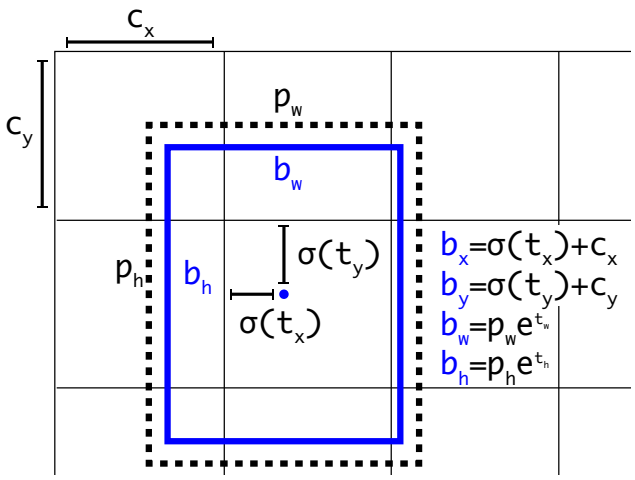


Figure 2. **Bounding boxes with dimension priors and location prediction.** We predict the width and height of the box as offsets from cluster centroids. We predict the center coordinates of the box relative to the location of filter application using a sigmoid function. This figure blatantly self-plagiarized from [15]

YOLOv3 predicts an objectness score for each bounding box using logistic regression. This should be 1 if the bound-ing box prior overlaps a ground truth object by more than any other bounding box prior. If the bounding box prior is not the best but does overlap a ground truth object by more than some threshold we ignore the prediction, follow-ing [17]. We use the threshold of .5. Unlike [17] our system only assigns one bounding box prior for each ground truth object. If a bounding box prior is not assigned to a ground truth object it incurs no loss for coordinate or class predic-tions, only objectness.

YOLOv3使用逻辑回归预测每个边界框（bounding box）的对象分数。如果先前的边界框比之前的任何其他边界框重叠ground truth对象，则该值应该为1。如果以前的边界框不是最好的，但是确实将ground truth对象重叠了一定的阈值以上，我们会忽略这个预测，按照[15]进行。我们使用阈值0.5。与[15]不同，我们的系统只为每个ground truth对象分配一个边界框。如果先前的边界框未分配给grounding box对象，则不会对坐标或类别预测造成损失。

2.2. Class Prediction

Each box predicts the classes the bounding box may con-tain using multilabel classification. We do not use a softmax as we have found it is unnecessary for good performance, instead we simply use independent logistic classifiers. Dur-ing training we use binary cross-entropy loss for the class predictions.

This formulation helps when we move to more complex domains like the Open Images Dataset [7]. In this dataset there are many overlapping labels (i.e. Woman and Person). Using a softmax imposes the assumption that each box has exactly one class which is often not the case. A multilabel approach better models the data.

每个框使用多标签分类来预测边界框可能包含的类。我们不使用softmax，因为我们发现它对于高性能没有必要，相反，我们只是使用独立的逻辑分类器。在训练过程中，我们使用二元交叉熵损失来进行类别预测。 这个公式有助于我们转向更复杂的领域，如Open Image Dataset[5]。在这个数据集中有许多重叠的标签（如女性和人物）。使用softmax会强加了一个假设，即每个框中只有一个类别，但通常情况并非如此。多标签方法更好地模拟数据。

2.3. Predictions Across Scales

YOLOv3 predicts boxes at A Aifferent Acales. Our Ays-tem Axtracts Aeatures from Ahose Acales using a Aimilar con-cept Ao Aeature pyramid Aetworks A8]. From Aur base Aea-ture Axtractor we add Aeveral convolutional Aayers. The Aast Af these predicts a A-d Aensor Ancoding bounding box, Ab-jectness, and class predictions. In Aur Axperiments with AOCO A10] we predict A boxes at Aach Acale so Ahe Aensor As $N\times N\times[3*(4+1+80)]$ Aor Ahe A bounding box Affsets, 1 objectness prediction, and 80 class predictions.

YOLOv3预测3种不同尺度的框（boxes）。我们的系统使用类似的概念来提取这些尺度的特征，以形成金字塔网络[6]。从我们的基本特征提取器中，我们添加了几个卷积层。其中最后一个预测了3-d张量编码边界框，对象和类别预测。在我们的COCO实验[8]中，我们预测每个尺度的3个框，所以对于4个边界框偏移量，1个目标性预测和80个类别预测，张量为N×N×[3*（4+1+80）]。

Next we take the feature map from 2 layers previous and upsample it by 2×. We also take a feature map from earlier in the network and merge it with our upsampled features using concatenation. This method allows us to get more meaningful semantic information from the upsampled fea-tures and finer-grained information from the earlier feature map. We then add a few more convolutional layers to pro-cess this combined feature map, and eventually predict a similar tensor, although now twice the size.

We perform the same design one more time to predict boxes for the final scale. Thus our predictions for the 3rd scale benefit from all the prior computation as well as fine-grained features from early on in the network.

We still use k-means clustering to determine our bound-ing box priors. We just sort of chose 9 clusters and 3 scales arbitrarily and then divide up the clusters evenly across scales. On the COCO dataset the 9 clusters were:(10×13), (16×30), (33×23), (30×61), (62×45), (59×119), (116×90), (156×198), (373×326).

接下来，我们从之前的两层中取得特征图（feature map），并将其上采样2倍。我们还从网络中的较早版本获取特征图，并使用element-wise addition将其与我们的上采样特征进行合并。这种方法使我们能够从早期特征映射中的上采样特征和更细粒度的信息中获得更有意义的语义信息。然后，我们再添加几个卷积层来处理这个组合的特征图，并最终预测出一个相似的张量，虽然现在是两倍的大小。我们再次执行相同的设计来预测最终尺度的方框。因此，我们对第三种尺度的预测将从所有先前的计算中获益，并从早期的网络中获得细粒度的特征。 我们仍然使用k-means聚类来确定我们的边界框的先验。我们只是选择了9个聚类（clusters）和3个尺度（scales），然后在整个尺度上均匀分割聚类。在COCO数据集上，9个聚类是：（10×13）；（16×30）；（33×23）；（30×61）；（62×45）；（59×119）；（116×90）；（156×198）；（373×326）。

2.4. Feature Extractor

We use a new network for performing feature extraction. Our new network is a hybrid approach between the network used in YOLOv2, Darknet-19, and that newfangled residual network stuff. Our network uses successive 3×3 and 1×1 convolutional layers but now has some shortcut connections as well and is significantly larger. It has 53 convolutional

我们使用新的网络来实现特征提取。我们的新网络是用于YOLOv2，Darknet-19中的网络和那些新颖的残差网络的混合方法。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shotcut连接，该网络明显更大。它有53个卷积层，所以我们称之为..... Darknet-53!

layers so we call it.... wait for it..... 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			

Table 1. **Darknet-53.**

This new network is much more powerful than Darknet-19 but still more efficient than ResNet-101 or ResNet-152. Here are some ImageNet results:

这个新网络比Darknet-19功能强大得多，而且比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

Table 2. **Comparison of backbones.** Accuracy, billions of oper-ations, billion floating point operations per second, and FPS for various networks.

Each network is trained with identical settings and tested at 256×256, single crop accuracy. Run times are measured on a Titan X at 256×256. Thus Darknet-53 performs on par with state-of-the-art classifiers but with fewer floating point operations and more speed. Darknet-53 is better than ResNet-101 and $1.5\times$ faster. Darknet-53 has similar perfor-mance to ResNet-152 and is $2\times$ faster.

Darknet-53 also achieves the highest measured floating point operations per second. This means the network struc-ture better utilizes the GPU, making it more efficient to eval-uate and thus faster. That's mostly because ResNets have just way too many layers and aren't very efficient.

每个网络都使用相同的设置进行训练，并以256×256的单精度测试进行测试。运行时间是在Titan X上以256×256进行测量的。因此，Darknet-53可与state-of-the-art的分类器相媲美，但浮点运算更少，速度更快。Darknet-53比ResNet-101更好，速度更快1：5倍。 Darknet-53与ResNet-152具有相似的性能，速度提高2倍。 Darknet-53也可以实现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，从而使其评估效率更高，速度更快。这主要是因为ResNets的层数太多，效率不高。

2.5. Training

We still train on full images with no hard negative mining or any of that stuff. We use multi-scale training, lots of data augmentation, batch normalization, all the standard stuff. We use the Darknet neural network framework for training and testing [14].

我们仍然训练完整的图像，没有hard negative mining or any of that stuff 。我们使用多尺度训练，大量的data augmentation，batch normalization，以及所有标准的东西。我们使用Darknet神经网络框架进行训练和测试[12]。

3. How We Do

YOLOv3 is pretty good! See table 3. In terms of COCOs weird average mean AP metric it is on par with the SSD variants but is $3\times$ faster. It is still quite a bit behind other models like RetinaNet in this metric though.

YOLOv3非常好！请参见表3。就COCO数据集的平均mAP成绩而言，它与SSD变体相当，但速度提高了3倍。尽管如此，它仍然比像RetinaNet这样的模型要差一点。

However, when we look at the "old" detection metric of mAP at IOU$=.5$ (or AP_{50} in the chart) YOLOv3 is very strong. It is almost on par with RetinaNet and far above the SSD variants. This indicates that YOLOv3 is a very strong detector that excels at producing decent boxes for ob-jects. However, performance drops significantly as the IOU threshold increases indicating YOLOv3 struggles to get the boxes perfectly aligned with the object.

In the past YOLO struggled with small objects. How-ever, now we see a reversal in that trend. With the new multi-scale predictions we see YOLOv3 has relatively high AP_S performance. However, it has comparatively worse performance on medium and larger size objects. More in-vestigation is needed to get to the bottom of this.

When we plot accuracy vs speed on the AP_{50} metric (see figure 5) we see YOLOv3 has significant benefits over other detection systems. Namely, it's faster and better.

然而，当我们在IOU = 0.5（或者图表中的AP50）看到mAP的"旧"检测度量时，YOLOv3非常强大。它几乎与RetinaNet相当，并且远高于SSD variants。这表明YOLOv3是一个非常强大的检测器，擅长为目标生成像样的框（boxes）。However, performance drops significantly as the IOU threshold increases indicating YOLOv3 struggles to get the boxes perfectly aligned with the object。 在过去，YOLO在小目标的检测上表现一直不好。然而，现在我们看到了这种趋势的逆转。 随着新的多尺度预测，我们看到YOLOv3具有相对较高的APS性能。但是，它在中等和更大尺寸的物体上的表现相对较差。需要更多的研究来达到这个目的。当我们在AP50指标上绘制精确度和速度时（见图3），我们看到YOLOv3与其他检测系统相比具有显著的优势。也就是说，速度越来越快。

4. Things We Tried That Didn't Work

We tried lots of stuff while we were working on YOLOv3. A lot of it didn't work. Here's the stuff we can remember.

Anchor box x, y offset predictions. We tried using the normal anchor box prediction mechanism where you pre-dict the x, y offset as a multiple of the box width or height using a linear activation. We found this formulation de-creased model stability and didn't work very well.

Linear x, y predictions instead of logistic. We tried using a linear activation to directly predict the x, y offset instead of the logistic activation. This led to a couple point drop in mAP.

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 sep-arate objectness predictions and conditional class predic-tions. Thus for most examples there is no loss from the class predictions? Or something? We aren't totally sure.

我们在研究YOLOv3时尝试了很多东西，以下是我们还记得的一些失败案例。

Anchor box坐标的偏移预测。我们尝试了常规的Anchor box预测方法，比如利用线性激活将坐标x、y的偏移程度预测为边界框宽度或高度的倍数。但我们发现这种做法降低了模型的稳定性，且效果不佳。

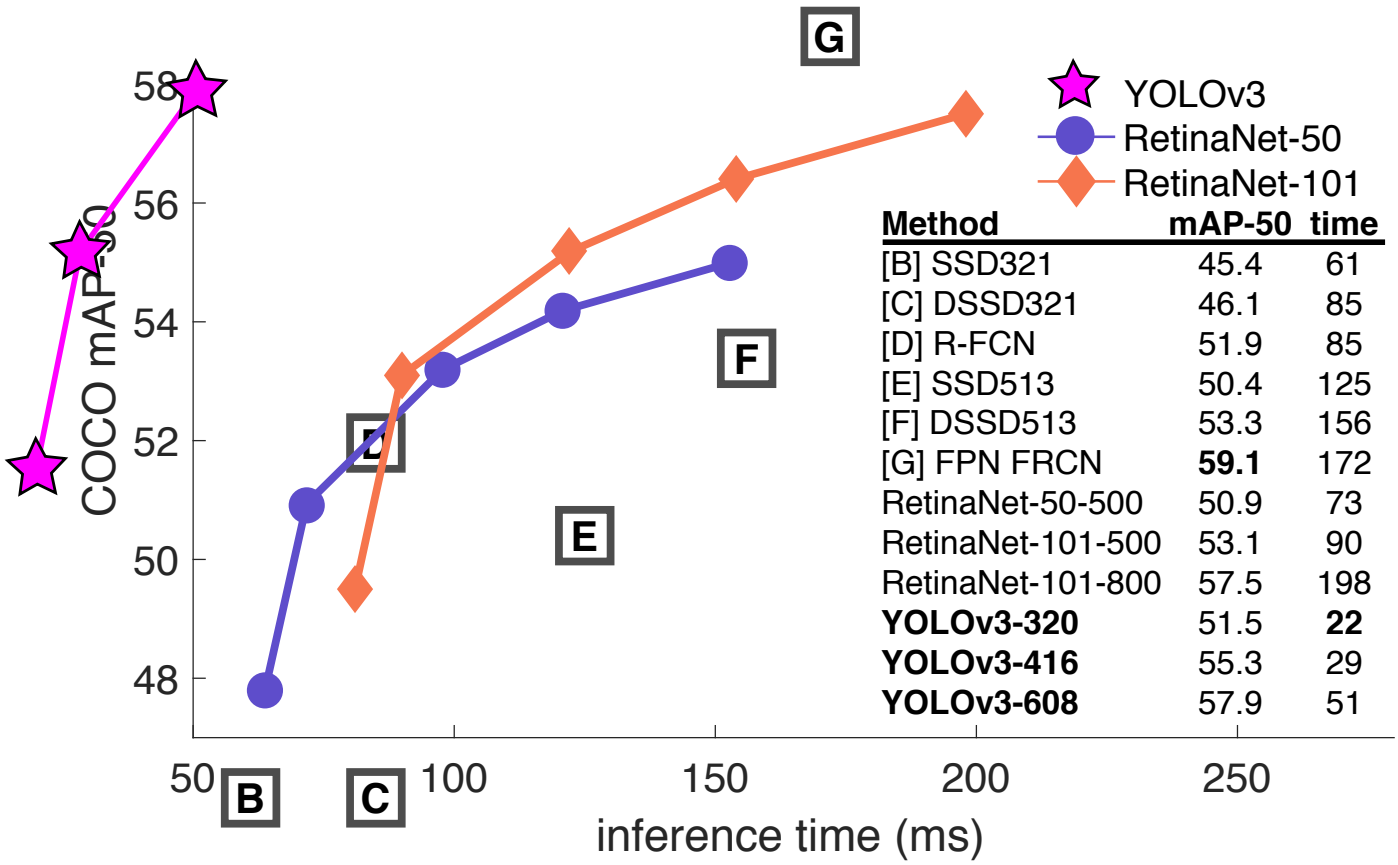
用线性方法预测x,y，而不是使用逻辑方法。我们尝试使用线性激活来直接预测x，y的offset，而不是逻辑激活。这降低了mAP成绩。

focal loss。我们尝试使用focal loss，但它使我们的mAP降低了2点。 对于focal loss函数试图解决的问题，YOLOv3从理论上来说已经很强大了，因为它具有单独的对象预测和条件类别预测。因此，对于大多数例子来说，类别预测没有损失？或者其他的东西？我们并不完全确定。

双IOU阈值和真值分配。在训练期间，Faster RCNN用了两个IOU阈值，如果预测的边框与.7的ground truth重合，那它是个正面的结果；如果在[.3—.7]之间，则忽略；如果和.3的ground truth重合，那它就是个负面的结果。我们尝试了这种思路，但效果并不好。我们对现在的更新状况很满意，它看起来已经是最佳状态。有些技术可能会产生更好的结果，但我们还需要对它们做一些调整来稳定训练。

	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

Table 3. I'm seriously just stealing all these tables from [9] they take soooo long to make from scratch. Ok, YOLOv3 is doing alright. Keep in mind that RetinaNet has like $3.8\times$ longer to process an image. YOLOv3 is much better than SSD variants and comparable to state-of-the-art models on the AP_{50} metric.



Method	mAP-50	time
[B] SSD321	45.4	61
[C] DSSD321	46.1	85
[D] R-FCN	51.9	85
[E] SSD513	50.4	125
[F] DSSD513	53.3	156
[G] FPN FRCN	**59.1**	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOv3-320	51.5	**22**
YOLOv3-416	55.3	29
YOLOv3-608	57.9	51

Figure 3. Again adapted from the [9], this time displaying speed/accuracy tradeoff on the mAP at .5 IOU metric. You can tell YOLOv3 is good because it's very high and far to the left. Can you cite your own paper? Guess who's going to try, this guy → [16]. Oh, I forgot, we also fix a data loading bug in YOLOv2, that helped by like 2 mAP. Just sneaking this in here to not throw off layout.

Dual IOU thresholds and truth assignment. Faster R-CNN uses two IOU thresholds during training. If a predic-tion overlaps the ground truth by .7 it is as a positive exam-ple, by $[.3 - .7]$ it is ignored, less than .3 for all ground truth objects it is a negative example. We tried a similar strategy but couldn't get good results.

We quite like our current formulation, it seems to be at a local optima at least. It is possible that some of these techniques could eventually produce good results, perhaps they just need some tuning to stabilize the training.

5. What This All Means

YOLOv3 is a good detector. It's fast, it's accurate. It's not as great on the COCO average AP between .5 and .95 IOU metric. But it's very good on the old detection metric of .5 IOU.

Why did we switch metrics anyway? The original COCO paper just has this cryptic sentence: "A full discus-sion of evaluation metrics will be added once the evaluation server is complete". Russakovsky et al report that that hu-mans have a hard time distinguishing an IOU of .3 from .5! "Training humans to visually inspect a bounding box with IOU of 0.3 and distinguish it from one with IOU 0.5 is sur-prisingly difficult." [18] If humans have a hard time telling the difference, how much does it matter?

But maybe a better question is: "What are we going to do with these detectors now that we have them?" A lot of the people doing this research are at Google and Facebook. I guess at least we know the technology is in good hands and definitely won't be used to harvest your personal infor-

mation and sell it to.... wait, you're saying that's exactly what it will be used for?? Oh.

Well the other people heavily funding vision research are the military and they've never done anything horrible like killing lots of people with new technology oh wait.....[1]

I have a lot of hope that most of the people using com-puter vision are just doing happy, good stuff with it, like counting the number of zebras in a national park [13], or tracking their cat as it wanders around their house [19]. But computer vision is already being put to questionable use and as researchers we have a responsibility to at least consider the harm our work might be doing and think of ways to mit-igate it. We owe the world that much.

In closing, do not @ me. (Because I finally quit Twitter).

YOLOv3是一个很好的检测器。速度很快，很准确。COCO平均AP介于0.5和0.95 IOU指标之间的情况并不如此。但是，对于检测度量0.5 IOU来说非常好。 但是也许更好的问题是："现在我们有了这些检测器（detectors），我们要做什么？"很多做这项研究的人都在Google和Facebook上。我想至少我们知道这项技术是非常好的，绝对不会被用来收集您的个人信息，并将其出售给.....等等，您是说这就是它的用途？ 那么其他大量资助视觉研究的人都是军人，他们从来没有做过任何可怕的事情，例如用新技术杀死很多人哦等等..... 我有很多希望，大多数使用计算机视觉的人都是做的快乐，研究了很多好的应用，比如计算一个国家公园内的斑马数量[11]，或者追踪它们在它们周围徘徊时的猫[17]。但是计算机视觉已经被用于可疑的应用，作为研究人员，我们有责任至少考虑我们的工作可能会造成的伤害，并考虑如何减轻它的影响。

[1]The author is funded by the Office of Naval Research and Google.

References

[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) challenge. *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 learning for image recognition. In *Proceedings of the IEEE conference 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. Dollár. 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. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common 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 improvement. *arXiv*, 2018. 4

[17] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *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. Beyond skip connections: Top-down modulation for object detection. *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

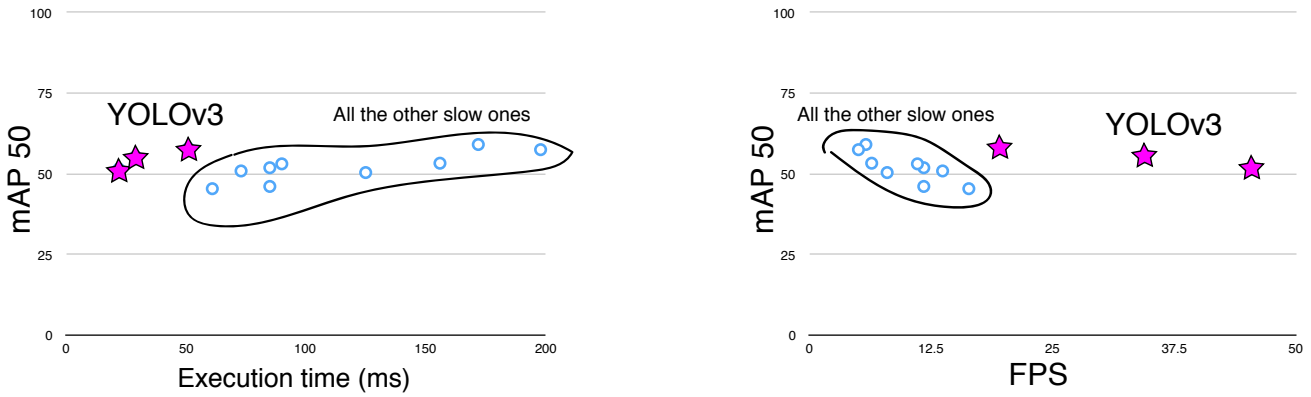


Figure 4. Zero-axis charts are probably more intellectually honest... and we can still screw with the variables to make ourselves look good!

Rebuttal

We would like to thank the Reddit commenters, labmates, emailers, and passing shouts in the hallway for their lovely, heartfelt words. If you, like me, are reviewing for ICCV then we know you probably have 37 other papers you could be reading that you'll invariably put off until the last week and then have some legend in the field email you about how you really should finish those reviews execept it won't entirely be clear what they're saying and maybe they're from the future? Anyway, this paper won't have become what it will in time be without all the work your past selves will have done also in the past but only a little bit further forward, not like all the way until now forward. And if you tweeted about it I wouldn't know. Just sayin.

Reviewer #2 AKA Dan Grossman (lol blinding who does that) insists that I point out here that our graphs have not one but two non-zero origins. You're absolutely right Dan, that's because it looks way better than admitting to ourselves that we're all just here battling over 2-3% mAP. But here are the requested graphs. I threw in one with FPS too because we look just like super good when we plot on FPS.

Reviewer #4 AKA JudasAdventus on Reddit writes "Entertaining read but the arguments against the MSCOCO metrics seem a bit weak". Well, I always knew you would be the one to turn on me Judas. You know how when you work on a project and it only comes out alright so you have to figure out some way to justify how what you did actually was pretty cool? I was basically trying to do that and I lashed out at the COCO metrics a little bit. But now that I've staked out this hill I may as well die on it.

See here's the thing, mAP is already sort of broken so an update to it should maybe address some of the issues with it or at least justify why the updated version is better in some way. And that's the big thing I took issue with was the lack of justification. For PASCAL VOC, the IOU threshold was "set deliberately low to account for inaccuracies in bounding boxes in the ground truth data" [2]. Does COCO have better labelling than VOC? This is definitely possible since COCO has segmentation masks maybe the labels are more trustworthy and thus we aren't as worried about inaccuracy. But again, my problem was the lack of justification.

The COCO metric emphasizes better bounding boxes but that emphasis must mean it de-emphasizes something else, in this case classification accuracy. Is there a good reason to think that more precise bounding boxes are more important than better classification? A miss-classified example is much more obvious than a bounding box that is slightly shifted.

mAP is already screwed up because all that matters is per-class rank ordering. For example, if your test set only has these two images then according to mAP two detectors that produce these results are JUST AS GOOD:

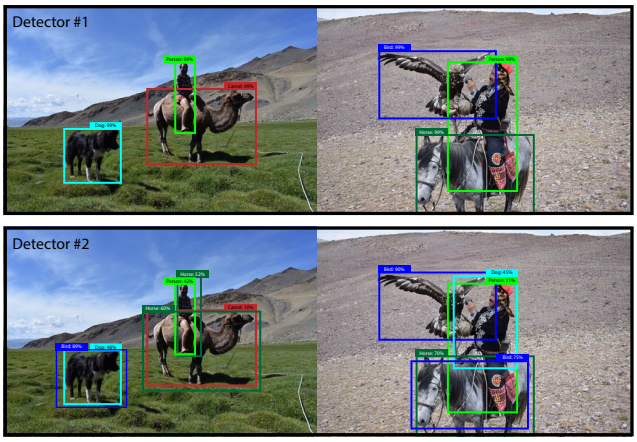


Figure 5. These two hypothetical detectors are perfect according to mAP over these two images. They are both perfect. Totally equal.

Now this is OBVIOUSLY an over-exaggeration of the problems with mAP but I guess my newly retconned point is that there are such obvious discrepancies between what people in the "real world" would care about and our current metrics that I think if we're going to come up with new metrics we should focus on these discrepancies. Also, like, it's already mean average precision, what do we even call the COCO metric, average mean average precision?

Here's a proposal, what people actually care about is given an image and a detector, how well will the detector find and classify objects in the image. What about getting rid of the per-class AP and just doing a global average precision? Or doing an AP calculation per-image and averaging over that?

Boxes are stupid anyway though, I'm probably a true believer in masks except I can't get YOLO to learn them.