

赞同 14

分享

3.1 YOLO入门教程：YOLOv3(1)-解读YOLOv3



Kissrabbbit

业余写手/机器人/深度学习/计算机视觉

关注他

14 人赞同了该文章

在上一章的最后，我们提到不论是YOLOv1，还是YOLOv2，都有一个共同的致命缺陷：只使用了最后一个经过32倍降采样的特征图（简称**C5特征图**）。尽管YOLOv2使用了passthrough技术将16倍降采样的特征图（即**C4特征图**）融合到了C5特征图中，但最终的检测仍是在C5尺度的特征图上进行的，最终结果便是导致了模型的小目标的检测性能较差。

为了解决这一问题，YOLO作者做了第三次改进，不仅仅是使用了更好的主干网络：DarkNet-53，更重要的是使用了FPN技术与**多级检测方法**，相较于YOLO的前两代，YOLOv3的小目标的检测能力提升显著。

那么，在本章，就让我们一起来领略一下YOLOv3的强大风采吧。



	backbone	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
<i>Two-stage methods</i>							
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
<i>One-stage methods</i>							
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	43.2	50.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9

图1. YOLOv3的性能

一、解读YOLOv3

1.1 更好的backbone：DarkNet-53

YOLOv3的第一处改进便是换上了更好的backbone网络：DarkNet53。相较于YOLOv2中所使用的DarkNet19，新的网络使用了更多的卷积——53层卷积，同时，添加了残差网络中的残差连结结构，以提升网络的性能。DarkNet53的具体结构如图2所示，注意，DarkNet53网络中的降采样操作没有使用Maxpooling层，而是由stride=2的卷积来实现。卷积层仍旧是**线性卷积、BN层以及LeakyReLU激活函数**的串联组合。

	类型	卷积核数量	卷积核大小	输出大小
1×	卷积层	32	3 × 3	256 × 256
	卷积层	64	3 × 3 / 2	128 × 128
	卷积层	32	1 × 1	128 × 128
	卷积层	64	3 × 3	
	残差连接			
2×	卷积层	128	3 × 3 / 2	64 × 64
	卷积层	64	1 × 1	64 × 64
	卷积层	128	3 × 3	
	残差连接			
	卷积层	256	3 × 3 / 2	32 × 32
8×	卷积层	128	1 × 1	32 × 32
	卷积层	256	3 × 3	
	残差连接			
	卷积层	512	3 × 3 / 2	16 × 16
	卷积层	256	1 × 1	16 × 16
8×	卷积层	512	3 × 3	
	残差连接			
	卷积层	1024	3 × 3 / 2	8 × 8
	卷积层	512	1 × 1	8 × 8
	卷积层	1024	3 × 3	
4×	残差连接			
	平均池化		全局	1000
	全连接			
	预测层			

图2. DarkNet-53的网络结构

在ImageNet数据集上，DarkNet53的top1准确率和top5准确率几乎与ResNet101和ResNet152持平，但速度却显著高于后两者。因此，相较于所对比的两个残差网络，DarkNet53在速度和精度上具有更高的性价比。不过，由于DarkNet53是由较小众的DarkNet深度学习框架实现的，因此没有成为学术界的主流模型，其受欢迎程度仍不及ResNet系列。所以，除了YOLO系列的工作，我们几乎是很少能看到DarkNet的身影的，包括近来的CSPDarkNet系列，我们也几乎看不到别的工作。



赞同 14



分享



一、解读YOLOv3
1.1 更好的backbone: Dark...
1.2 使用FPN与多级检测

多说几句，目前来看，目标检测领域的baseline几乎已经被RetinaNet工作统治了，很多增量式的改进也都是在RetinaNet的基础上做的，往往Mask R-CNN和Faster R-CNN也会用上，毕竟是双阶段检测器的经典之作。之所以会采用RetinaNet作为baseline，一个原因是RetinaNet的网络十分简洁，训练起来也没有太tricky的东西。也许有人会说，YOLO也很简洁呀，确实，YOLO正因为其网络十分简洁，因而有着较好的泛化性，没有设计过多的trick来在COCO上刷性能（有可能过拟合）。但另一个很重要的原因便是RetinaNet的训练时间很短，通常只需要在COCO上训练12个epoch，数据增强也只需要使用随机水平翻转即可。相反，YOLOv3往往需要在COCO上训练超过200个epoch，并且使用包括随机水平翻转、颜色扰动、随机剪裁和多尺度训练在内等大量的数据增强手段。因此，就训练时间而言，YOLOv3往往会需要多得多的时间，这对于没有太多显卡的研究员来说并不友好。尤其是当今又是一个“拼手速”的时代，我们往往急于求成，快点拿到涨点的结果然后写到实验里，发出论文来，因此，训练耗时更少的RetinaNet显然是个更好的选择。不过，在解决实际问题时，YOLO系列更加受欢迎，毕竟在实际任务里，“实时性”是个很重要的指标，这一点恰恰是RetinaNet的劣势。YOLO性能强、速度快、计算量也要远小于RetinaNet，因此更适合用在实际部署中，无非是训练成本大了些。所以孰优孰劣，不能一概而论。

言归正传。

笔者出于对这个工作的喜爱，尝试使用PyTorch深度学习框架对其进行了复现。复现此模型的最关键之处在于我们手上要有庞大的**ImageNet数据集**和算力足够的GPU设备。对此，我们不做要求，读者可以直接下载由笔者复现的DarkNet53网络的预训练权重文件，读者可在项目代码中的**README文件**中找到相关下载链接。

读者会得到两个文件： darknet53_75.42.pth 和 darknet53_hr_77.76.pth 。前者是使用224的输入图像尺寸进行训练得到的，而官方YOLOv3是使用256的图像尺寸进行训练的，因此性能上自然会有所差距，但这个并不影响我们的后续工作。而后者中的“hr”表示这个是在448的图像上微调过，这一技巧我们已经在讲解YOLOv2的章节中介绍过了。

DarkNet53的代码文件已放置在我们的YOLOv3项目中的 backbone/darknet.py 文件中，读者可以打开查看网络的具体实现细节。

1.2 使用FPN与多级检测

FPN的最早是在2017年的CVPR会议上提出的，其创新点在于提出了一种**自底向上（bottom-up）的结构**，融合多个不同尺度的特征图去进行目标预测。FPN工作认为网络浅层的特征图包含更多的细节信息，但语义信息较少，而深层的特征图则恰恰相反。原因之一便是卷积神经网络的降采样操作，降采样对小目标的损害显着大于大目标，直观的理解便是小目标的像素少于大目标，也就越难以经得住降采样操作的取舍，而大目标具有更多的像素，也就更容易引起网络的“关注”，在YOLOv1+和YOLOv2+的工作中我们也发现了，相较于小目标，大目标的检测结果要好很多。

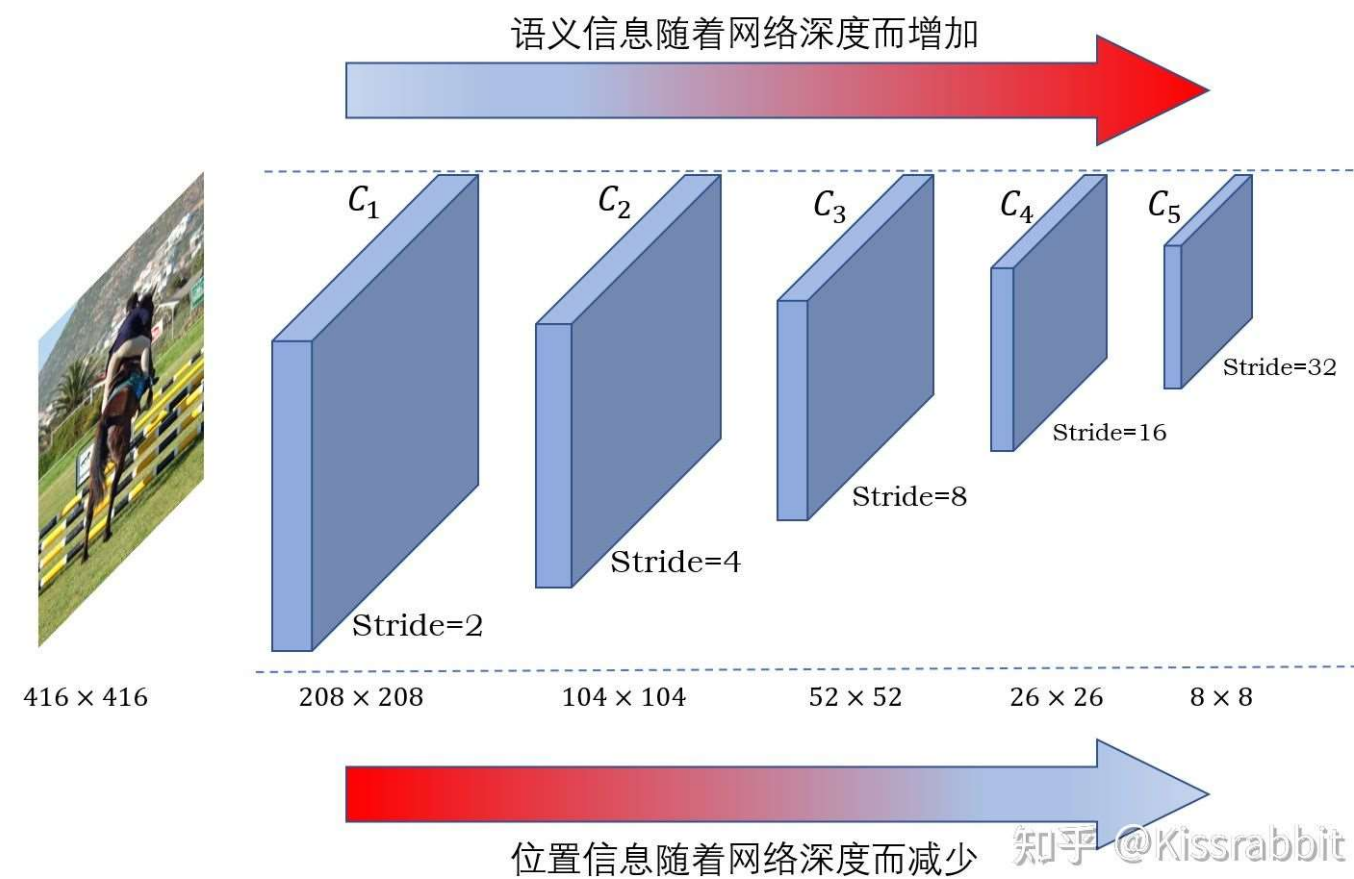


赞同 14



分享





赞同 14

分享

图3. 卷积神经网络中的语义信息和位置信息的变化趋势

随着网络深度的加深，降采样操作的增多，细节信息不断被破坏，致使小物体的检测效果逐渐变差，而大目标由于像素较多，仅靠网络的前几层还不足以使得网络能够认识到大物体（感受野不充分），但随着层数变多，网络的感受野逐渐增大，网络对大目标的认识越来越充分，检测效果自然会更好。于是，一个很简单的解决方案便应运而生：**浅层网络负责检测较小的目标，深层网络负责检测较大的目标**。考虑识别物体的类别依赖于语义信息，因此将深层网络的语义信息融合到浅层网络中去是个很自然的想法。

FPN工作的出发点便是如此，提出了一个行之有效的网络结构，如图4所示。其基本思想便是对深层网络输出的特征图使用上采样操作，然后与浅层网络进行融合，使得来自于不同尺度的细节信息和语义信息得到了有效的融合。

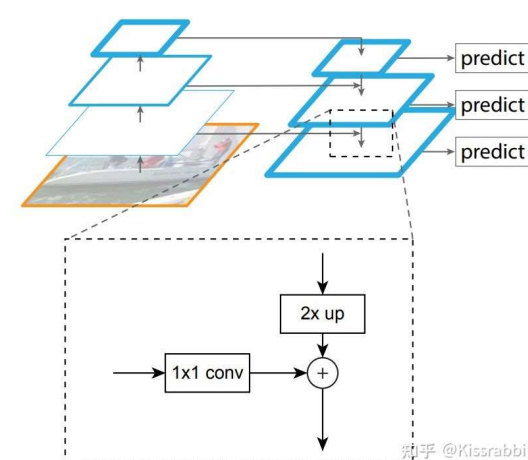


图4. FPN的核心结构

从网格的角度来看，越浅层的网格，划分出的网格也越精细，以416的输入尺寸为例，经过8倍降采样得到的特征图C3相当于是一个 52×52 的网格，这要比经过32倍降采样得到的特征图C5所划分的 8×8 的

网格精细得多，也就更容易去检测小物体。同时，更加精细的网格，也就更能避免先前所提到的“语义歧义”的问题。

既然，FPN将不同尺度的特征图的信息进行了一次融合，那么一个很自然的方法也就应运而生：**多级检测**（multi-level detection）。最早，多级检测方法可以追溯到SSD网络，SSD正是使用不同大小的特征图来检测不同尺度的目标，这一方法的思想内核便是“**分而治之**”，即**不同尺度的物体由不同尺度的特征图去做检测**，而不是像YOLOv2那样，都堆在最后的C5特征图上去做检测。而FPN正是在这个基础上，让不同尺度的特征图先融合一遍，再去做检测。FPN的这一强大特性，使得它称为了“分而治之”检测方法的重要模块。也为后续许多的特征融合工作带去了启发，如PAN和BiFPN。

这里强调一下，“**分而治之**” **方法的内核不是FPN，而是多级检测**。FPN不过是锦上添花，即使我们不做特征融合，依旧可以做多级检测，如SSD。只是，使用特征融合手段，可以让检测的效果更好罢了。

多说一句，既然有“分而治之”，便也应有“合而治之”，所谓“合而治之”，是指**所有物体我们都在一个特征图上去检测**，换言之，就是“**单级检测**”（single-level detection），比如早期的YOLOv1和YOLOv2，便是最为经典的单级检测工作。只不过，主流普遍认为这种只在C5特征图上去单级检测的检测器，小目标检测效果是不行的，尽管这一点被ECCV2020的DeTR和CVPR2021的YOLOF工作否决了，却依旧难以扭转这一根深蒂固的观念，前者似乎只被关注了Transformer这一点上，而后者似乎被认为是“开历史倒车”。无数的历史已证明，根深蒂固的观念是很难被改变，而一旦被改变的那一天，便是一场旧事物的大毁灭与新事物的大喷发.....

不过，还有一类单级检测工作则另辟蹊径，借鉴人体关键点检测工作的思想，使用高分辨率的特征图如只经过4倍降采样得到的特征图C2来检测物体，典型的工作包括CornerNet和脍炙人口的CenterNet。以512的输入尺寸为例，只经过4倍降采样得到的特征图C2相当于是一个128×128的网格，要比C5的16×16精细的多，然后再将所有尺度的信息都融合到这一张特征图来，使得这样一张具有精细的网格的特征图既具备足够的细节信息，又具备足够的语义信息。不难想象，这样的网络只需要一张特征图便可以去检测所有的物体。这一类工作具有典型的encoder和decoder的结构，通常encoder由常用的ResNet组成，decoder由简单的FPN结构或者反卷积组成，当然，也可以使用Hourglass网络。这一类的单级检测很轻松的得到了研究学者们的认可，毕竟，相较于在粗糙的C5上做检测，直观上便很认同分辨率高得多的C2特征图检测方式。只不过，C2特征图的尺寸太大，会带来很大的计算量，但是，这类工作不需要诸如800×1333的输入尺寸，仅仅512×512的尺寸便可以达到与之相当的性能。

“分而治之”与“合而治之”各有千秋，这里我们不去下孰优孰劣的定论，由读者自己来判断吧。

再次言归正传。

YOLOv3的关键改进便是使用了FPN结构与多级检测方法。YOLOv3在3个尺度上去进行预测，分别是经过8倍降采样的特征图C3、经过16倍降采样的特征图C4和经过32倍降采样的特征图C5。完整的YOLOv3网络结构如图5所示，整体来看，其网络结构并不复杂。



赞同 14



分享



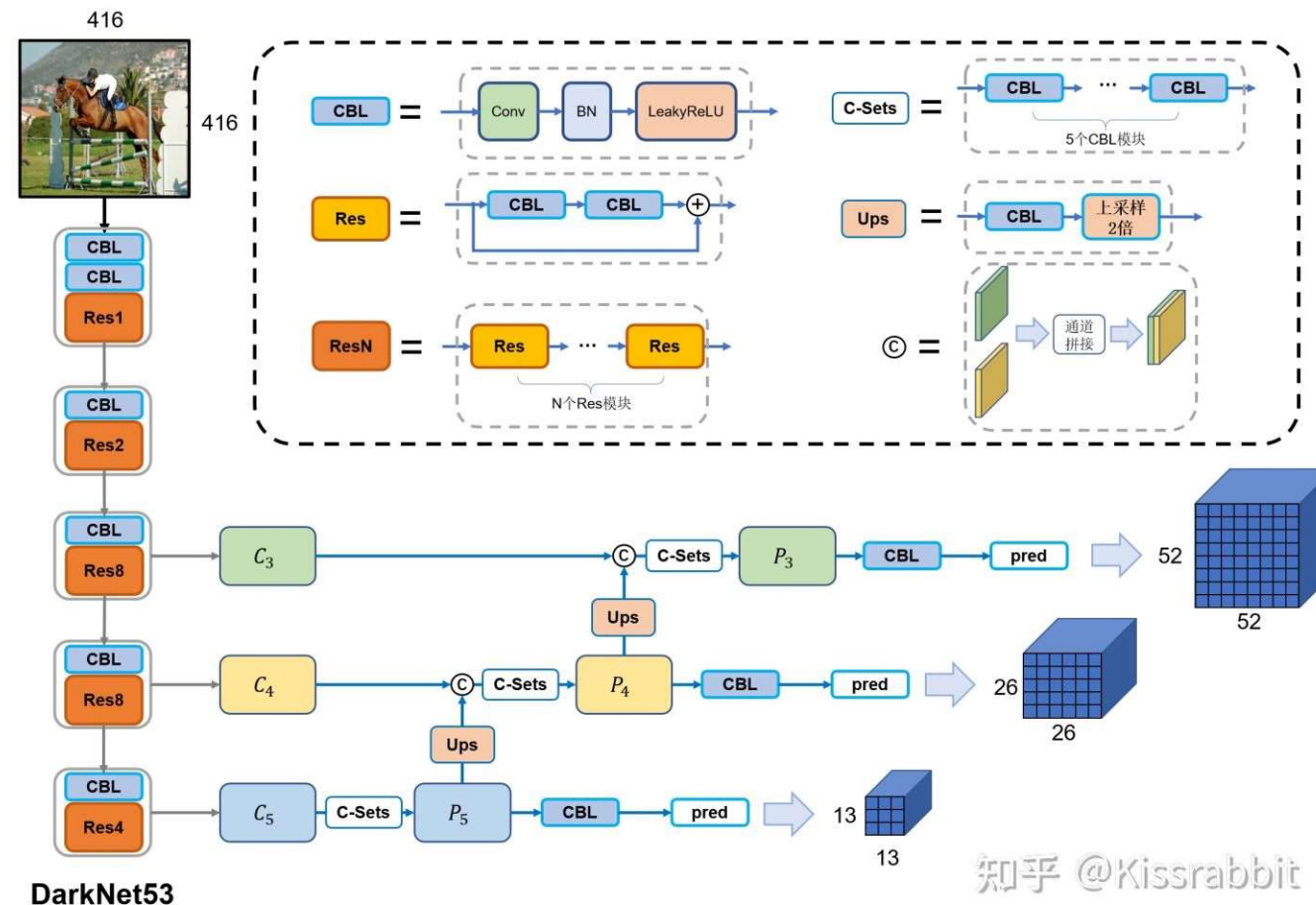


图5. YOLOv3的结构

在每个特征图上，YOLOv3在每个网格处放置3个先验框。由于YOLOv3一共使用3个尺度，因此，YOLOv3一共设定了9个先验框，这9个先验框仍旧是使用kmeans聚类的方法获得的。在COCO上，这9个先验框的宽高分别是(10, 13)、(16, 30)、(33, 23)、(30, 61)、(62, 45)、(59, 119)、(116, 90)、(156, 198)、(373, 326)。注意，YOLOv3的先验框尺寸不同于YOLOv2，后者是除以了32，而前者是在原图尺寸上获得的，没有除以32。

每个尺度的网格都放置3个先验框，且每个先验框的预测仍旧是包括置信度、类别和位置参数（换言之，输出共包括objectness+class+bbbox三部分输出），因此，每个尺度所预测的张量的通道数都是 $3 \times (1 + C + 4)$ 。以416的输入尺寸为例，YOLOv3最终会输出 $52 \times 52 \times 3(1 + C + 4)$ 、 $26 \times 26 \times 3(1 + C + 4)$ 和 $13 \times 13 \times 3(1 + C + 4)$ 三个预测张量，然后将这些预测结果汇总到一起，进行后处理，得到最终的检测结果。

	backbone	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
<i>Two-stage methods</i>							
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
<i>One-stage methods</i>							
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	43.2	51.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9

图6. YOLOv3在COCO test-dev上的测试结果

赞同 14

分享

其实，不难看出，相较于YOLOv2，YOLOv3主要就是额外多了两个尺度的预测。尽管YOLOv3的性能不及RetinaNet，但在AP50指标上，YOLOv3几乎和RetinaNet达到一个水准，但YOLOv3的速度是后者的3倍左右。在精度和速度的平衡上，YOLOv3做得十分出色，也因此，YOLOv3工作的问世使得工业界的模型又进行了一次迭代更新。

在下一节，我们将在YOLOv2+的工作基础上，来搭建一个我们自己的YOLOv3。同先前一样，我们不会百分之百地复现官方的YOLOv3，实现上会有些许差别，但没有实质性的差别。接下来，让我们开始准备实现一个更好的YOLOv3吧。

编辑于 2021-12-09 12:07

「真诚赞赏，手留余香」

赞赏

还没有人赞赏，快来当第一个赞赏的人吧！

[深度学习（Deep Learning）](#) [yolov3](#) [目标检测](#)

文章被以下专栏收录



第二卷-基于YOLO的目标检测入门教程
本专栏将从零开始完整实现YOLOv1至v3模型

推荐阅读

YOLOv3原理代码赏析

代码搬运工 发表于深度学习超...

目标检测论文阅读：YOLOv1-YOLOv3（二）

YOLOv2也已经更新，为了阅读方便，直接更新在上周的博客里了，有兴趣的不妨前去观摩，这里主要介绍下YOLOv3，也是目前YOLO最新的版本。YOLOv3Introduction和Conclusion有很多吐槽无力的...

扬之水 发表于从目标检测...

物体检测之YOLOv3

YOLOv3论文的干货并不多，用作者自己的话说是一篇“Tech Report”。这篇主要是在YOLOv2[2]的基础上的一些Trick尝试，有的Trick成功了，包括：考虑到检测物体的重叠情况，用多标签的方式...

大师兄



《目标检测》-第6章-YOLOv3！


在实现了YOLOv2的复现工作后，我接着又把YOLOv3也做了，网络结构和官方的是一样的，这一块的代码是和YOLOv2的项目放在一块了：
<https://github.com/yjh0410/yolo3>

Kissr... 发表于第一卷-目...





写下你的评论...




 LastMonody 05-11

我不知道在最后的結果中分別取objectness預測、類別class預測、bbox的txtytwith預測是不是有先後順序,假如k-means會選擇10個anchor,20個類別的話 這樣如果最終的預測是13, 13, 1024, $10 \times (1 + 4 + 20)$, 那當分離3個預測的時候, 假如將最後一維理解成一個線性的長度, 前面10個是anchor的預測, 中間是 10×20 的類別的預測,最後是 4×10 的坐標預測, 我現在如果改成前面10個是anchor的預測,中間改成 4×10 坐標預測,最後全部是 10×20 類別的預測請問這個可以嗎

 贊

 么么哒的小谦 05-02

FPN操作，下采样之后，又进行上采样操作融合，这样一来，不就信息丢失一部分嘛

 贊

 时间的影子 04-05


权重模型能在上传一下吗

 贊


 一条乐 03-28

老哥真的非常棒！有幸在目标检测中遇到老哥这么牛的人！

 贊

 蟹老板 02-22

博主，图3. 卷积神经网络中的语义信息和位置信息的变化趋势这个图最后一个 8×8 ，是不是应该改为 13×13 啊

 贊

 Kissrabbbit (作者) 回复 蟹老板 02-22

对~马虎了，感谢指正~

 贊

 未来可期 2021-12-03

开心开心，这个系列更新到yolov3了

 贊

