YOLOv4 笔记

YOLOv4 笔记
- 摘要
- Introduction
- Related work
  - Ordinary object detector
  - Bag of freebies
    - data augmentation
    - Imbalance between different classes
    - BBox regression
  - Bag of specials
    - enhance receptive field
    - the attention module
    - feature integrations
    - good activation function
    - post-process
- Methodology
  - 架构选择
  - 用到的BoF/BoS
  - 针对单GPU训练的进一步改进
    - mosaic augmentation
    - SAT
    - CmBN
    - modify SAM/PAN
- 最终网络一览
- 一些值得注意的改进点
- Conclusions
- 复现

摘要

YOLOv4使用了一些新特性：

WRC/CSP/CmBN/SAT/Mish activation/Mosaic data augmentation/DropBlock regularization/CIou loss

Introduction

当前(2020)最精确的现代神经网络无法实时进行目标检测，并且需要多块GPU进行训练（对显存有较高要求）。

而YOLOv4可以在一个传统GPU上进行训练/推断，降低了使用门槛。具体来说，**GPU可以是1080 Ti/2080 Ti。**

除此之外，在YOLOv4上对比了**“Bag-of-freebies”** 和 **“Bag-of-Specials”**这两类方法对模型的影响。

最后，通过**改进CBN/PAN/SAM**，使YOLOv4更适合单GPU训练

Related work

Ordinary object detector

一般来说，一个现代detector应该由两部分组成——一个在ImageNet**预训练过的backbone**、一个**检测头用于类别和BBox的预测。**

骨干网络：

- GPU平台有VGG/ResNet/ResNeXt/DenseNet
- CPU平台有SqueezeNet/MobileNet/ShuffleNet

检测头：

- **two-stage:**R-CNN/faster R-CNN/R-FCN/Libra R-CNN **anchor-free:**RepPoints
- **one-stage:**YOLO/SSD/RetinaNet **anchor-free:**CenterNet/CornerNet/FCOS

Neck:

Neck介于backbone和head之间，主要功能是**融合不同阶段的特征图，混合浅层细粒度特征和深层的语义特征**

- 主要代表有：FPN/PAN/BiFPN/NAS-FPN

以下是通用检测器的结构图

- **Input**: Image, Patches, Image Pyramid
- **Backbones**: VGG16 [68], ResNet-50 [26], SpineNet [12], EfficientNet-B0/B7 [75], CSPResNeXt50 [81], CSPDarknet53 [81]
- **Neck**:
    - **Additional blocks**: SPP [25], ASPP [5], RFB [47], SAM [85]
    - **Path-aggregation blocks**: FPN [44], PAN [49], NAS-FPN [17], Fully-connected FPN, BiFPN [77], ASFF [48], SFAM [98]
- **Heads:**:
    - **Dense Prediction (one-stage)**:
        - RPN [64], SSD [50], YOLO [61], RetinaNet [45] (anchor based)
        - CornerNet [37], CenterNet [13], MatrixNet [60], FCOS [78] (anchor free)
    - **Sparse Prediction (two-stage)**:
        - Faster R-CNN [64], R-FCN [9], Mask R-CNN [23] (anchor based)
        - RepPoints [87] (anchor free)

Bag of freebies

> Definitions：We call these methods that only change the training strategy or only increase the training cost as “bag of freebies.“

Bag of freebies指只改变训练策略或只增加训练开销的方法。其中一个典型代表是data augmentation(数据增强)。

data augmentation

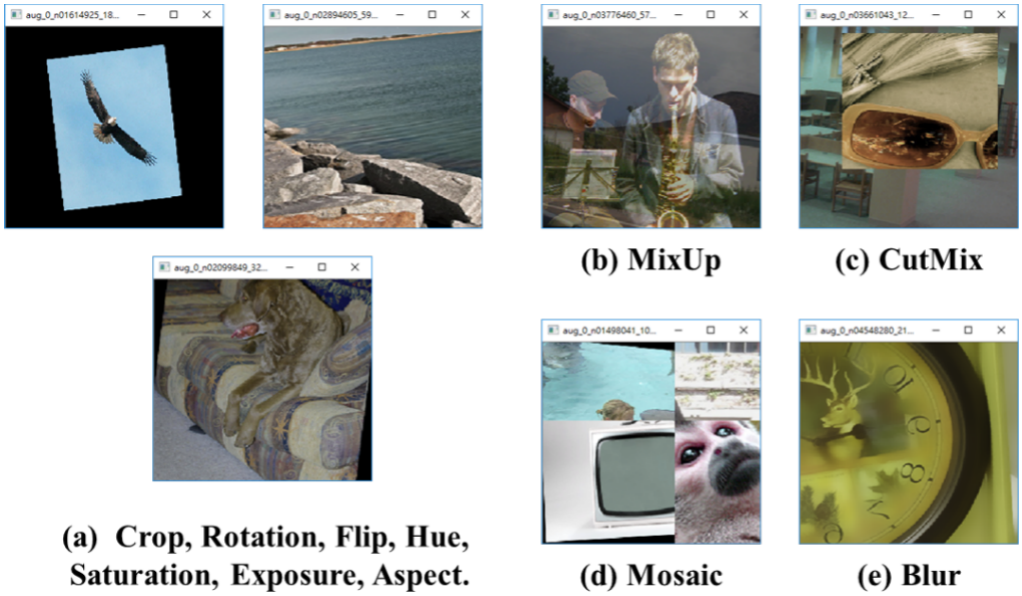
常见的数据增强有：

- 调整图像的**亮度、对比度、色调、饱和度，添加噪点**
- **随机缩放、裁剪、翻转、旋转**等

> 以上方法都是逐像素调整，保留了原始像素信息。

除此之外，还有数据遮挡方面的增强：

- CutOut：随机选择图像中的矩形区域，**填充随机值/0**
- hide-and-seek/grid mask：随机或均匀地选择图像中的多个矩形区域并替换为零
- 特征映射中也有和hide-and-seek类似的方法，DropOut/DropConnect/DropBlock
- MixUp：将**两个图像以不同系数叠加**，并按照系数调整标签
- CutMix：将**裁剪后的图像覆盖到其他图像的矩形区域**，并根据混合区域的大小调整标签
- GAN:**风格转移也可以做数据增强**，并且这种使用可以**减少CNN学习的纹理偏差**



Imbalance between different classes

在数据集中可能会存在语义分布偏差问题，也就是不同类之间存在较大的数据量差异。这个问题可以通过难例挖掘解决。不过**难例挖掘不适用于单阶段检测器，因为单阶段检测器属于密集预测架构。】**

因此**focal loss**被提出（这个方法在YOLOv3论文中有被提到。不过不怎么work），focal loss**用于解决正负类样本数量不均衡的问题**，对于YOLOv3而言，由于IOU阈值设得比较高(好像是0.5)，一些预测的比较好的框被判为负类，从而使负类里混了正样本。focal loss又会强调这些负类的loss，导致网络无法学到较好的效果。

BBox regression

之前BBox的损失计算，是直接对四个参数(可能是左上角右下角坐标，也可能是中心点宽高，再或者中线点以及相对于锚框大小的偏移)**直接计算MSE**，这有个问题就是**没有考虑到对象本身的完整性，四个参数都是视作独立变量处理。**

新改进就是**引入IoU损失**，因为IoU计算本身会用到BBox四个坐标点，反向传播的时候可以都更新到，并且这个时候**这四个点是作为一个整体考虑的。**其次IoU损失是一**种尺度不变的表示**，可以解决传统方法的问题。即无论是L1 loss还是L2 loss，尺度增加loss都会不可避免的增大。

近年来，推出了**GIoU损失，它不仅考虑了覆盖区域，还考虑了物体的形状和方向**，具体来说就是找到同时覆盖预测框和真实框的最小的BBox作为新的分母，代替原来的分母计算IoU.

DIoU损失则额外考虑了物体中心点的距离。CIoU则同时考虑了覆盖面积、中心点之间的距离和宽高比，具有更好的收敛速度和精度。

Bag of specials

> Definitions: For those plugin modules and post-processing methods that only increase the inference cost by a small amount but can significantly improve the accuracy of object detection, we call them “bag of specials”.

Bag of specials 指那些增加少量推理开销却能显著提高目标检测准确率的模块和后处理方法。

一般来说，这些模块增强了模型中的某些属性，比如扩大感受野、引入注意力机制、增强特征整合等。

后处理则是对模型的预测结果进行筛选。

enhance receptive field

增强感受野的模块有SPP/ASPP/RFB。

SPP起源于SPM，SPM是分割特征图并提取词袋特征。SPP则是将SPM集成到CNN中，使用Maxpool代替词袋操作。改进后的SPP是具有核大小k × k的maxpool输出的级联，其中k = {1，5，9，13}，并且stride等于1。

ASPP和RFB是对SPP的进一步改进。

the attention module

物体检测中常用的注意力模型**主要分为通道注意力和点注意力**，这两种注意力模型的代表分别是**挤压和激励（SE）和空间注意力模型（SAM）。**

虽然SE模块可以在ImageNet图像分类任务中以仅增加2%的计算量为代价提高ResNet50 1%的top-1准确率，但在GPU上通常会增加约10%的推理时间，因此**更适合在移动设备上使用。**

而对于SAM，它只需要付出0.1%的额外计算，就可以将ResNet50-SE在ImageNet图像分类任务中的top-1准确率提高0.5%。最棒的是，**它完全不影响GPU上的推理速度。**

feature integrations

特征集成方面，早期的实践是使用跳跃连接(skip connection)或超列(hyper-column)**将低级物理特征集成到高级语义特征。**随着**FPN等多尺度预测方法**的流行，许多集成不同特征金字塔的轻量级模型被提出。这类模块包括SFAM、ASFF和BiFPN。SFAM的主要思想是利用SE模块对多尺度级联特征图进行通道级重加权。对于ASFF，它使用softmax进行逐点水平重加权，然后添加不同尺度的特征图。在BiFPN中，提出了多输入加权残差连接进行尺度级重加权，然后添加不同尺度的特征图。

good activation function

一个好的激活函数可以**使梯度更有效地传播，同时不会引起太多额外的计算代价。**2010年，Nair和Hinton提出ReLU，实质上**解决了tanh和sigmoid中经常遇到的梯度消失问题**.（但是relu也可能会导致梯度爆炸吧？）随后，LReLU、PReLU 、ReLU6 、缩放指数线性单元（SELU）、Swish 、硬Swish和Mish等，其也被用于解决梯度消失问题。**LReLU和PReLU的主要目的是解决当输出小于零时ReLU的梯度为零的问题。**(leaky relu也可以吧)至于ReLU6和硬Swish，它们是专门为量化网络设计的。为了实现神经网络的自归一化，提出了SELU激活函数来满足这一目标。值得注意的是Swish和Mish都是连续可微的激活函数。

post-process

基于深度学习的目标检测中**常用的后处理方法是NMS**，它可以**过滤掉对同一目标预测不好的BBox，只保留响应较高的候选BBox。**NMS试图改进的方法与优化目标函数的方法是一致的。NMS提出的原始方法没有考虑上下文信息，Girshick等在R-CNN中加入了分类置信度作为参考，并根据置信度的大小，**按照得分从高到低的顺序进行贪婪NMS。**对于**soft NMS，考虑了对象遮挡可能导致具有IoU值的贪婪NMS的置信度下降的问题。**DIoU NMS 开发者的思路是在soft NMS的基础上，在BBox筛选过程中加入中心点距离的信息。值得一提的是，由于上述后处理方法均未直接参考捕获的图像特征，因此**在后续开发无锚方法时不再需要进行后处理。**

Methodology

架构选择

- 骨干网络选择CSPDarknet53，有29个3 × 3的卷积层，725 × 725的感受野和27.6M的参数。
- 在CSPDarknet53上添加SPP块，因为它**显著增加了感受野，分离出最显著的上下文特征，并且几乎不引起网络运行速度的降低。**
- **使用PANet作为针对不同检测器水平的不同骨干水平的参数聚合方法**，而**不是YOLOv3中使用的FPN。**
- 选择CSPDarknet53 backbone、SPP附加模块、PANet Neck和YOLOv3 head作为YOLOv4的体系结构。

用到的BoF/BoS

对于Bag of freebies和Bag of specials见这里。

YOLOv4里用到了以下这些技术。

- **Activations**: ReLU, leaky-ReLU, parametric-ReLU, ReLU6, SELU, Swish, or Mish
- **Bounding box regression loss**: MSE, IoU, GIoU, CIoU, DIoU
- **Data augmentation**: CutOut, MixUp, CutMix
- **Regularization method**: DropOut, DropPath [36], Spatial DropOut [79], or DropBlock
- **Normalization of the network activations by their mean and variance**: Batch Normalization (BN) [32], Cross-GPU Batch Normalization (CGBN or SyncBN) [93], Filter Response Normalization (FRN) [70], or Cross-Iteration Batch Normalization (CBN) [89]
- **Skip-connections**: Residual connections, Weighted residual connections, Multi-input weighted residual connections, or Cross stage partial connections (CSP)

> 对于训练激活函数，由于PReLU和SELU较难训练，而ReLU6是专门为量化网络设计的，因此将上述激活函数从候选列表中移除。在正则化的方法上，发表DropBlock的人详细地将他们的方法与其他方法进行了比较，他们的正则化方法大获全胜。因此作者选择DropBlock作为正则化方法。对于归一化方法的选择，由于关注仅使用一个GPU的训练策略，因此不考虑syncBN。

针对单GPU训练的进一步改进

- 介绍了一种**新的数据增强马赛克**和**自对抗训练（SAT）方法。**
- 在应用遗传算法时选择**最佳超参数。**
- 修改了一些现有方法使其更适合于高效训练和检测：修改SAM、修改PAN和交叉小批量归一（CmBN）

mosaic augmentation

Mosaic是一种新的数据增广方法，它**混合了4个训练图像**，而CutMix仅混合2个输入图像。这允许**检测其正常上下文之外的对象。**(检测比较突兀的物体)此外，BN在每层上计算4个不同图像的activation statistics(这里不是很懂)。这显著降低了对大的小批量的需求。也就是一张图相当于四个图(batch size有x4的效果)。

Figure 3: Mosaic represents a new method of data augmentation.

SAT

自我对抗训练（SAT）也代表了一种新的数据增广技术，其在2个阶段中操作。在**第一阶段，神经网络改变原始图像**而不是网络权重。通过这种方式，神经网络对自身执行对抗性攻击，改变原始图像以创建图像上没有期望对象的欺骗。在**第二阶段，训练神经网络以正常方式检测该修改图像上的对象。**

> 这里论文也没细说如何修改原始图像的....说的比较笼统

CmBN

CmBN表示CBN的修改版本，如图4所示，定义为交叉小批量归一化（CmBN）。这只在单个batch内的mini-batch之间收集统计信息。(大概就是把batch norm再细化了一下，从更小的单位做归一化吧)

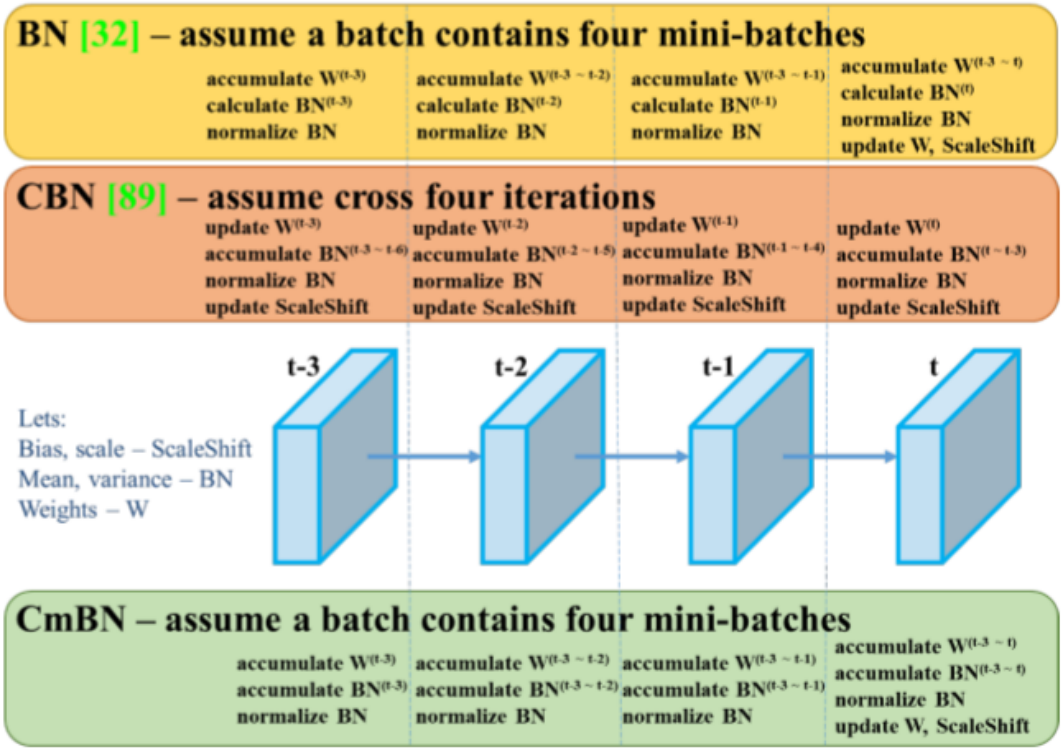


Figure 4: Cross mini-Batch Normalization.

modify SAM/PAN

modify SAM这玩意作者在最后代码里没用，估计是效果不好。

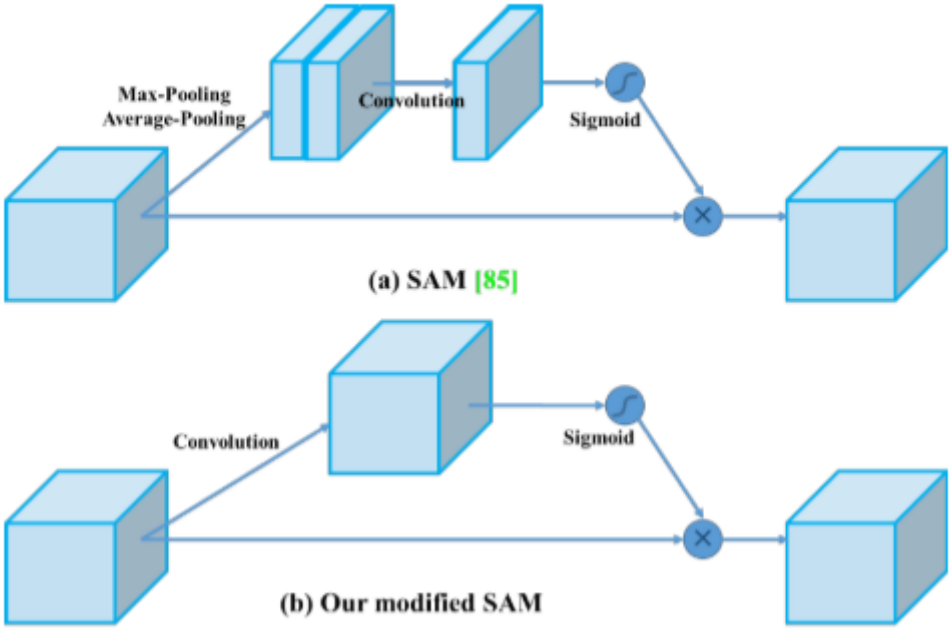


Figure 5: Modified SAM.

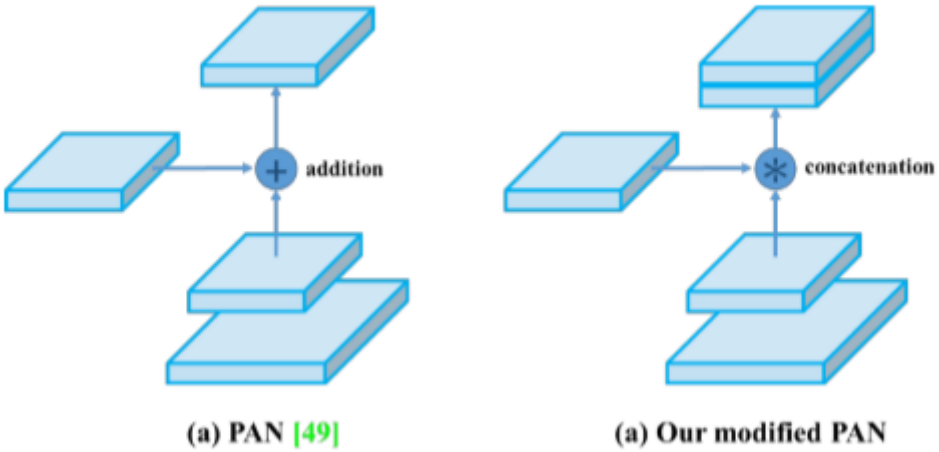


Figure 6: Modified PAN.

最终网络一览

3.4. YOLOv4

In this section, we shall elaborate the details of YOLOv4.

YOLOv4 consists of:

- Backbone: CSPDarknet53 [81]
- Neck: SPP [25], PAN [49]
- Head: YOLOv3 [63]

YOLO v4 uses:

- Bag of Freebies (BoF) for backbone: CutMix and Mosaic data augmentation, DropBlock regularization, Class label smoothing
- Bag of Specials (BoS) for backbone: Mish activation, Cross-stage partial connections (CSP), Multi-input weighted residual connections (MiWRC)
- Bag of Freebies (BoF) for detector: CIoU-loss, CmBN, DropBlock regularization, Mosaic data augmentation, Self-Adversarial Training, Eliminate grid sensitivity, Using multiple anchors for a single ground truth, Cosine annealing scheduler [52], Optimal hyper-parameters, Random training shapes
- Bag of Specials (BoS) for detector: Mish activation, SPP-block, SAM-block, PAN path-aggregation block, DIoU-NMS

一些值得注意的改进点

1.之前为了让预测的坐标中心点落在当前grid cell里面，采用sigmoid函数对txty进行约束，这导致了只有在tx/ty特别大的时候，这一项才会是1，即预测很难到达网格边缘，为了解决这个问题，在sigmiod前乘了一个大于1的因子，来稍微补偿一下这个问题。

- S: Eliminate grid sensitivity the equation $b_x = \sigma(t_x) + c_x, b_y = \sigma(t_y) + c_y$, where c_x and c_y are always whole numbers, is used in YOLOv3 for evaluating the object coordinates, therefore, extremely high t_x absolute values are required for the b_x value approaching the c_x or $c_x + 1$ values. We solve this problem through multiplying the sigmoid by a factor exceeding 1.0, so eliminating the effect of grid on which the object is undetectable.

2.实验对比表示，CSPResNeXt50+PAN+SPP+SAM效果最好

Table 5: Ablation Studies of Bag-of-Specials. (Size 512x512).

Model	AP	AP_{50}	AP_{75}
CSPResNeXt50-PANet-SPP	42.4%	64.4%	45.9%
CSPResNeXt50-PANet-SPP-RFB	41.8%	62.7%	45.1%
CSPResNeXt50-PANet-SPP-SAM	**42.7%**	**64.6%**	**46.3%**
CSPResNeXt50-PANet-SPP-SAM-G	41.6%	62.7%	45.0%
CSPResNeXt50-PANet-SPP-ASFF-RFB	41.1%	62.6%	44.4%

Conclusions

本篇文章算是对之前YOLOv3的延续，网络本身没有什么大的改动(无非换了一个backbone/把FPN换成SAM/加了SPP，算法核心没变)，而是使用了很多trick来提高性能(BOF/BOS)，并且对单GPU训练提供更好的支持。

6. Conclusions

We offer a state-of-the-art detector which is faster (FPS) and more accurate (MS COCO $AP_{50...95}$ and AP_{50}) than all available alternative detectors. The detector described can be trained and used on a conventional GPU with 8-16 GB-VRAM this makes its broad use possible. The original concept of one-stage anchor-based detectors has proven its viability. We have verified a large number of features, and selected for use such of them for improving the accuracy of both the classifier and the detector. These features can be used as best-practice for future studies and developments.

复现

感觉是在复现各种trick...暂时先咕了