

目标检测

Sisyphes

2020 年 8 月 18 日

目录

第一节 计划	2
第二节 检测的基本认识	2
2.1 maskrcnn	2
2.2 RetinaFace	2
2.3 YOLOv4	2
2.4 CenterNet	5
2.5 OCR	5
第三节 实战	5
3.1 Wheat Detection	5

第一节 计划

- 详细讲解 mmdetection, 涉及模型 maskrcnn, retinaface, centernet... 和炼丹平台实现知识。
- 炼丹升级技术, 以 YOLOv4, 小麦检测为例, 涉及较多有用组件实现和模型优化技术。
- 实际落地项目, ncnn, onnx, docker 等。

第二节 检测的基本认识

基本认识可参考我写的[mmdet.pdf](#)。针对本小册来说: 以下这些算法都有其各自的说明点。maskrcnn 具有开创意义, retinaface 弥补了人脸的空白, yolov4 一个说明炼丹的极佳例子, Centernet 补了一些检测的历史信息, OCR 是另一个实用的综合方向。

2.1 maskrcnn

放到 mmdetection 中详细讲解。

2.2 RetinaFace

只讲从思考上, 如何对通用检测做改动, 并结合人脸的特征, 设计出人脸检测算法, RetinaFace。

2.3 YOLOv4

Alexey Bochkovskiy 等人认为, 提升网络性能的组件有很多, 组合实验这些技术, 实验分析这些结果, 是一件有必要的事 (给出了炼丹佳例)。

于是他们将已有组件进行分类: 特殊的 (具体网络, 特定问题, 小数据等) 和通用的 (batch-normalization, residual-connection, Cross-Stage-Partial-connections, Self-adversarial-training, Mish activation, Mosaic data

augmentation, DropBlock regularization, CIoU loss 等), 然后在这上面做组合实验, 给出了一些指导性结论, 也给出了 YOLOv4 模型。图解可参考 [pprp-yolov4](#)。

整篇论文的目的是寻找一个单 GPU1080Ti+ 训练, 推理高速, 性能佳的一个检测模型。具体实验细节, 见论文, 结合代码视频讲解。这里只是对一些思路, 做简要梳理。

首先综合当前已知检测模型, 给出了检测的基本组件¹。

- **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)

图 1: detect components in yolov4

然后对这些组件, 作者就改进点, 给出了两个划分: Bag of freebies, Bag of specials, 围绕这两个划分“空间”, 设计了一系列实验, 最终得出 YOLOv4: backbone: CSPDarknet53(Skip-connections), Neck: SPP(enlarging receptive field), PAN(path-aggregation block), Head: YOLOv3。

那么什么是 Bag of freebies? 什么是 Bag of specials?

freebies iCIBA 给出解释: freebie 的复数形式, 免费的赠品 (礼物, 膳食等), 有 free tickets(免票), free services(免费服务) 等组合。回到论文, 其含义是指改变训练策略或只增加训练成本却不影响推理速度的组件。

BoF1 有数据增强一系列方法, DropBlock, 对抗训练 SAT, (涉及模型鲁棒, 性能, 数据量等问题)

BoF2 有解决数据语义分布偏差 (不均衡) 问题的 focal loss, two-stage 的采样 (hard negative example mining) 等方法

BoF3 也有用 soft-label 来克服 one-hot 难以刻画不同类别之间的关联关系的问题

BoF4 还有改进 box 回归的一系列方法, MSE->IoU->GIoU->DIoU->CIoU 等。

specials: special 的复数, 表示特价, 特刊, 特约稿等含义。那么在论文中是指, 功能插件 (增加感受野, 引入注意力机制, 增强特征 integration(融合) 能力) 和后处理方法, 增加微弱推理时间的同时带来更大的性能提升。

BoS1 提升感受野: SPP, ASPP, RFB(实验表明: 增加 SPPP 模块, YOLOv3-608 upgrades AP50 by 2.7% on the MS COCO object detection task at the cost of 0.5% extra computation) ...

BoS2 注意力机制: 一般能提升 0.5-1% 的分类 ac

- channel-wise attention: Squeeze-and-Excitation(SE)
- point-wise attention: Spatial Attention Module(SAM)
- ...

BoS3 特征融合: 单尺度 skip connection 等, 多尺度 FPN, SFAM, ASFF, BiFPN 等

BoS4 后处理: soft NMS, DIoU NMS 等。

总结: 在以生产系统和并行计算优化方面快速运算为前提, 对当前组件进行分类, 在分类的基础上, 结合检测任务的一些经验认识 (高分辨率输入能提升小物体性能, 更多层增加感受野, 更多参数提升单检测器对多尺度物体的表达能力), 设计基本网络, 功能模块, 和组合实验, 也做了一些创新 (增加了 SAT, Mosaic, 遗传算法超参数优化, 改进的 SAM, PAN, CmBN, 改进点主要考虑单 GPU 训练效率问题), 给出当前最佳组合, 取名 YOLOv4。同时实验也说明了, 分类网络的性能和最终检测的性能并不一定呈正相关。

问题：改变 one-hot 编码，使其具有类别关联刻画能力。初步想法就是增加一个关联矩阵。

2.4 CenterNet

以 `mmsdet` 为基础讲。

这里补充一些实际部署相关材料，或许更好。

2.5 OCR

研读 `PaddleOCR` 及其相关论文是一个不错的开始。

第三节 实战

3.1 Wheat Detection

小麦方案选，该方案 Public Leaderboard 15, Private Leaderboard 第一。利用 MMDetection 跑了系列模型，最后在 DetectoRS with the ResNet50 backbone 和 UniverseNet+GFL with the Res2Net101 backbone 做了融合。

具体优化方法：

- Jigsaw puzzles: 原始图片是从大图中裁剪出的，此团队对其做了类似还原，组成了 1300 张大图，参考了代码 `create-mosaic`，但对应的框有边界和重叠问题，因此做了离线数据合成，并使用了伪标签。
- 使用 `MultilabelStratifiedKfold` 将数据按框的个数，框的面积的中位数，以及数据来源 (不同国家) 分成 5 个层级，并且层级图没有交叠
- 不让 `usask_1`, `ethz_1` 数据参与 validation
- 以上在两大模型的性能上都有一定提升，但对官方评估 LB 指标没有提升，于是转到局部性能提升，超参数，增强，模型融合
- 使用第一 fold 作为训练数据

- 因数据量少,且测试数据和训练数据分布相差较大,于是用**albumentations**做了大量增强
 - HorizontalFlip, ShiftScaleRotate, RandomRotate90
 - RandomBrightnessContrast, HueSaturationValue, RGBShift
 - RandomGamma
 - CLAHE
 - Blur, MotionBlur
 - GaussNoise
 - ImageCompression
 - CoarseDropout
- RandomBBoxesSafeCrop: 随机选择 N 个框, 并以这 N 个框的外包裁剪
- 用**pix2pix**做图形上色
- 用**stylize-datasets**做风格变换
- Mosaic, Mixup 增强
- 多尺度训练: $[(768 + 32 * i, 768 + 32 * i) \text{ for } i \text{ in range}(25)]$
- 除了上色和风格转换为离线, 其他均是 online。
- 增加了额外训练数据: **SPIKE dataset**
- 模型方面 DetectoRS3:
 - LabelSmoothCrossEntropyLoss with parameter 0.1
 - **Empirical Attention**
- 训练流程采用了 mutil-stage traning pipeline²
- Multi-scale Testing with scales $[(1408, 1408), (1536, 1536)]$
- Pseudo labelling
 - 以 $\text{confidence} = \text{np.mean}(\text{scores} > 0.75)$ 作为测试图的置信度, 如果大与 0.6, 则接收此图, 并使用伪标签

- 来自 [usask_1, ethz_1] 的数据没有在伪标签上使用 mosaic, mixup, colorization, style transfer 增强技术
- 1 epoch, 1 round, 1 stage.
- Data: original data + 3 x pseudo test data
- Ensemble(模型融合)⁴:
 - 使用 **WBF** 融合技术
 - 两个模型分数分布不同, 使用了 **rankdata** 方法: $\text{scaled_scores} = 0.5 * (\text{rankdata}(\text{scores}) / \text{len}(\text{scores})) + 0.5$
 - 最后参数: $\text{weights}=[0.65, 0.35]$ respectively for models [DetectoRS, UniverseNet], $\text{iou_thr}=0.55$, $\text{score_thr}=0.45$ 。

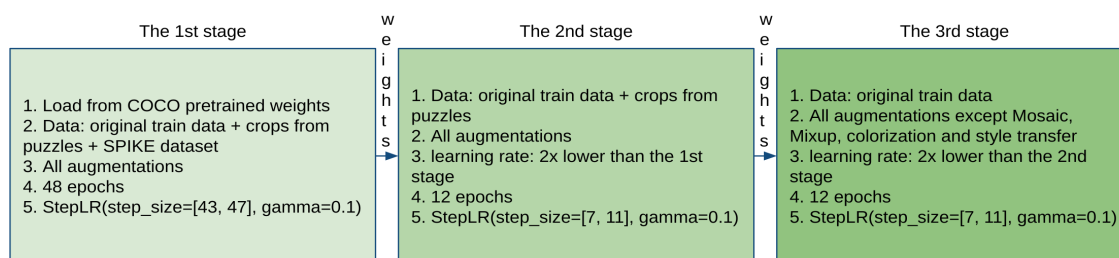


图 2: multi-stage training pipeline

题外话: 以上是一个大模型和一个小模型的融合, 因 YOLOV5 的 license 是 GPL 协议 (GPL 3.0), 而 Kaggle 不允许使用 GPL 协议的项目参赛, 所以被禁用了, 若 yolov5 和 mmdet 模型进行融合, 想必效果会稳居第一。

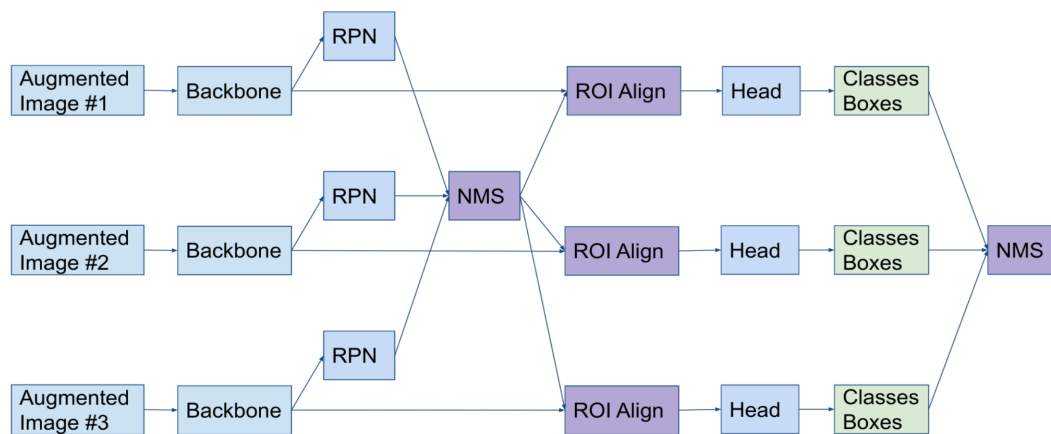


图 3: DetectoRS

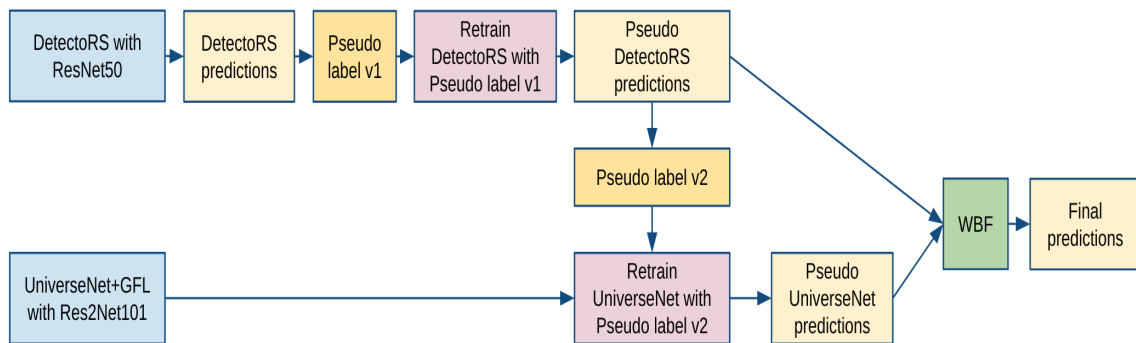


图 4: Ensemble