

背景介绍

难点一：大中小目标分布不均

在遥感数据中

1. 小目标数量众多，排列密集或稀疏，中大目标的数量较少
2. 由于遥感图像需要进行切割，因此会加剧目标分布不均的情况。

在YOLOv5中，利用Mosaic数据增强方式可以有效解决这个问题。**Mosaic** 的拼接特性能过极大程度上改善 **DOTA** 数据集目标分布不均的情况，并且由于其拼接的随机性，随着训练时间的加长，改善效果会更明显。

难点二：目标尺度变化剧烈

大部分目标的尺度较小，提供的细节信息有限，且随着网络的加深，小目标的细节信息会逐渐丢失，因此我们需要网络结构对小目标有一定敏感性，即较强的多尺度特征提取能力。高空场景下，目标尺度时而密集，时而稀疏，模型需要满足单个图像中检测多个不同尺度多个对象的能力。

因此需要考虑选取合适的 featuremap 的结果（Backbone + Neck），既要保证丰富的浅层位置信息同时保证深层的语义信息。yolov5 的 SPP + PANet 对这个问题有很好的处理

难点三：目标长宽比大，排列密集

在遥感场景下，很多目标的长宽比较大，且排列密集，因此遥感目标更适合利用旋转框进行检测。

原始YOLOv5项目的应用场景为自然场景下的目标，目标检测边框为水平矩形框因此利用了改建后的YOLOv5，增加了一个预测角度的分支来进行旋转目标的检测。

损失计算：置信度损失、class分类损失、 θ 角度分类损失、bbox边框回归损失。

- class分类损失：无需更改，注意数据索引部分即可。
- θ 角度分类损失：由于我们添加的 θ 是分类任务，照葫芦画瓢，添加分类损失就可以了，
- bbox边框回归损失：边框回归损失部分依旧采用IOU/GIOU/CIU/DIOU损失函数。
- 置信度损失：置信度分支的权重系数依然选择水平边框的之间的IOU/GIOU/CIU/DIOU；

所做的工作

1. 对数据进行处理

1. 对标注格式的转换，将数据格式转换为YOLO的方式
 2. 对数据进行裁剪
 3. 对裁剪后的数据进行清洗，由于裁剪后的数据很多没有目标，因此遍历标签将不含有目标的数据进行剔除
2. 在进行训练的过程中，进行多尺度训练，按照两个不同的比例对图片进行裁剪，然后进行训练，这样的训练方式对最终的mAP提升效果不显著，只有零点几个点，且大幅增加了训练时间，因此在后期放弃了这个训练策略。
3. 进行多尺度训练检测，在进行检测的过程中，进行多尺度检测，将图片以两种不同的比率进行裁剪，进行检测，map 上升了 3 个百分点，其中小目标（车辆）的 AP 上升了 8 个百分点

最终的结果

达到了 68% 的 mAP，但是其中的大目标，如足球场，桥梁等较大的目标 AP 较低，只有 50% 左右，分析原因，一是在裁剪的过程中，将这些目标裁掉了，二是这些大目标在数据集中的数量占比较少，对 loss 的贡献较少。但是总的来说，这个结果还是较好的。