

算法技术方案

杨丽鹤
南京大学

liwe.yang.cs@gmail.com

1. 参赛队伍名称

Leo

2. 竞赛项目

变化检测

3. 算法描述

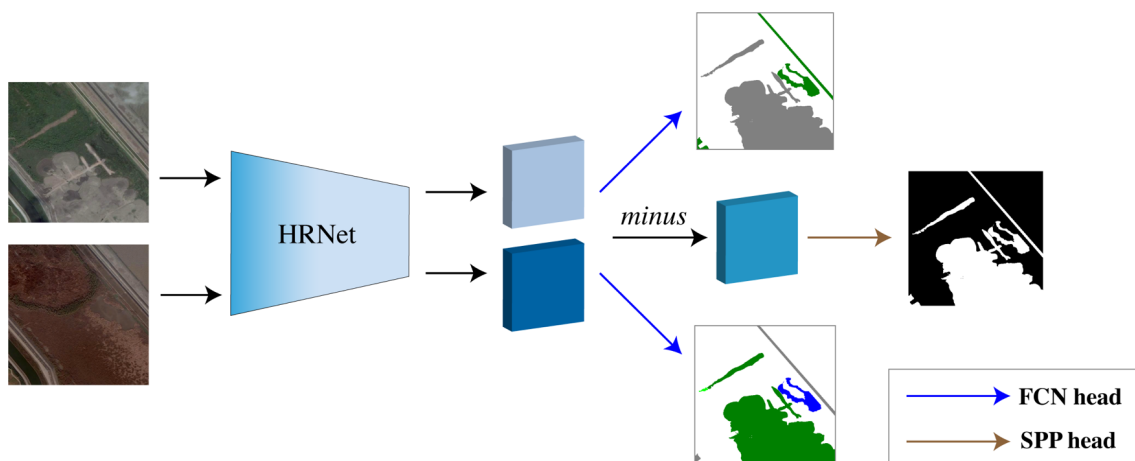
该问题可以看作是一个多任务的语义分割问题。有两个任务，分别是：

- 1) 是否变化的二分类分割
- 2) 变化区域的多类别语义分割

3.1 网络结构

在该任务中对比了ResNet[2]、ResNeXt[5]、ResNeSt[6]、HRNet[3]这一系列backbone的性能，最终发现在相近的参数量下，HRNet的表现优于ResNet、ResNeXt、ResNeSt，因此后面提到的所有网络结构所用的backbone都是HRNet系列。

根据上述的分析，为了解决两种不同的分割任务，采用的是共享backbone并分别用两个segmentation heads处理各自的分割任务的结构，大致示意如下：



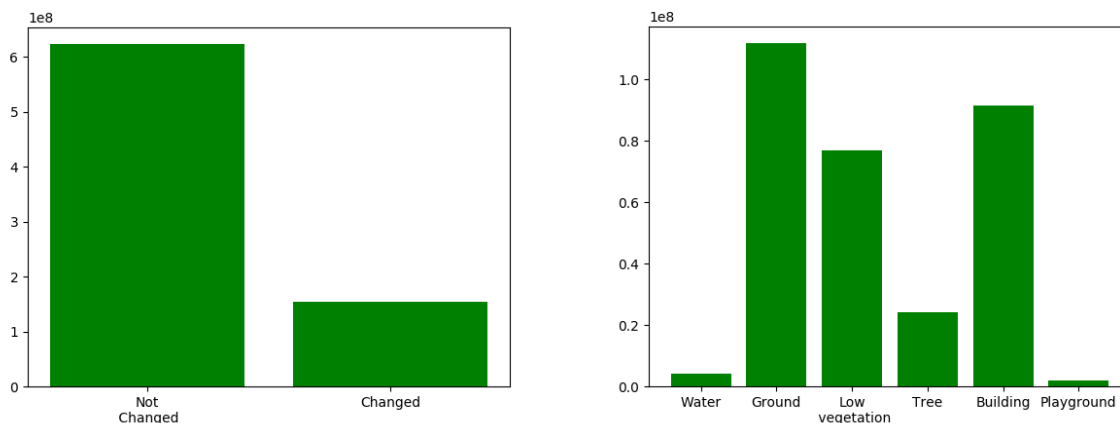
- 1) 对于两张相同区域但不同时间的图像，输入共享参数的backbone，随后用共享参数的segmentation head进行语义分割。具体来说，实验中使用的是经典的FCN head，也尝试了其他

的一些heads，如Spatial Pyramid Pooling head(PSPNet[7])，ASPP head(DeepLabv3[1])，但是效果反而没有FCN head好。对于没有变化的区域，因为标注中没有具体的类别，所以在这些区域忽略loss的计算。

- 2) 为了进行变化检测，对上述两张图像提取出的特征作差并取绝对值，得到相同尺寸的特征图，在这个特征图上用Spatial Pyramid Pooling(SPP) head进行最终的二分类的分割。

3.2 不同类别的分割难易程度不一致

上述方案构建起了最基本的网络结构，但是实验中发现分割效果并不十分理想，不同的类别分割难易程度是相差较大的，且训练数据存在较严重的类别不平衡问题。有两方面的类别不平衡问题，首先是未变化区域和变化区域的面积相差较大，变化区域的面积较小；其次是各种地貌类型的面积相差较大，“水体”、“树木”、“运动场”这三个类别在标注中的面积较小，示意图如下：



为了解决上述的不同类别的分割难度差异较大的问题以及类别不平衡的问题，采用的是较为简单的对cross-entropy loss的各个类别加权的方式，对于分割精度较差的类别权重设为2（在语义分割分支中是“水体”，“低矮植被”，“树木”，在二分类的变化检测中是“变化的区域”），其他类别的权重都设为1。

通过这个改进，在公榜测试集上大概能有1%的提升。

3.3 伪标签缓解数据量少的问题

缓解了类别的分割难易程度不一致以及数量不平衡的问题后，实验发现分割的可视化结果仍然不理想，由于已经尝试了多种backbone（ResNet、ResNeXt、ResNeSt、HRNet）以及一些语义分割中常用的multi-scale[7]、attention[4]机制，且变化检测这方面的前沿工作并没有对网络结构有特别大的修改，于是希望转而从数据的角度继续提升模型的性能。

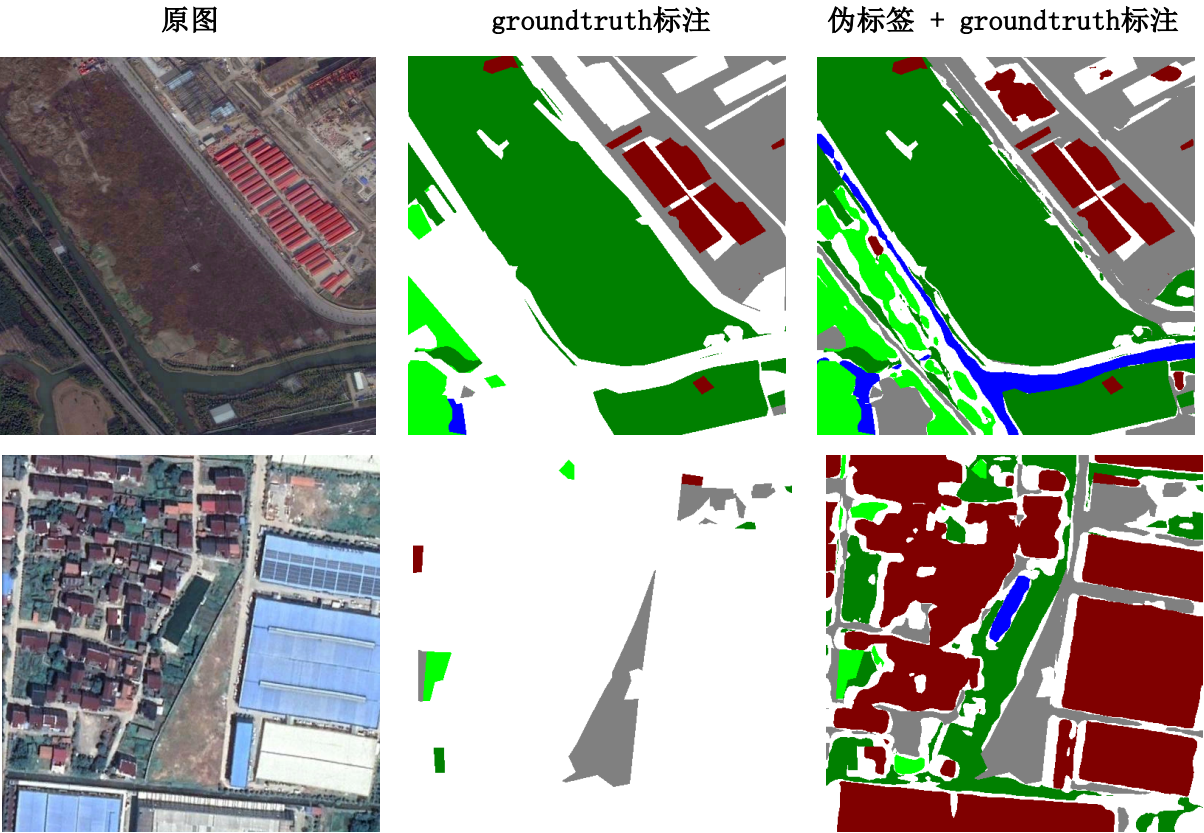
考虑到该数据集的数据量很小，并且更为严重的是，一组图像中的未变化区域没有具体的语义类别标注，有语义类别标注的像素只占每张图像的一小部分，因此尝试通过伪标签的方式来对未变化区域预测语义类别标签，从而增加标注像素的数量，缓解数据量小的问题。

具体来说，先按照所给的groundtruth标注训练得到一个变化检测模型，该模型能够进行两方面的任务，一是变化检测，二是语义分割。对于groundtruth中标注的未变化的区域，用语义分割分支预测具体的语义类别，在其他区域则用groundtruth已经给出的具体类别。结合未变化区域的伪标签以及变化区域的真实标签，可以重新训练一个分割模型，该模型最终用于测试。

为了提升伪标签的质量，采用了多个较大的模型（HRNet_w48, HRNet_w44, HRNet_w40等）投票的方式。需要说明的是，这些大模型只会在训练中打伪标签的过程中用到，测试时不需要这些模型。

通过这个改进，在公榜测试集上大概能有2%的提升。

部分伪标签的可视化结果如下：



3.4 Test-time augmentation (TTA) 和模型集成

TTA是融合了六种变换下的预测结果，分别为旋转0、90、180、270度，水平、垂直翻转。在模型集成方面，考虑到测试时间以及模型大小的限制，最终融合了两个backbone分别为HRNet_w40和HRNet_w18的模型。

4. 实验环境

1) 未使用外部数据集，只用了HRNet官方给出的在ImageNet上的预训练模型

- 2) 硬件资源: 4张Tesla V100 GPU(在4张2080Ti上也进行了实验, 其显存可以完成整个训练, 且对精度没有影响, 用V100是为了更快的训练速度), 整个训练过程约4~5小时, 训练框架是PyTorch
- 3) docker运行命令: `sudo docker run --gpus all -it --rm -v /local-val-path:/val -v /local-output-path:/output --shm-size 8G name-of-the-image`
- 4) 代码开源地址: <https://github.com/LiheYoung/SenseEarth2020-ChangeDetection>

References

- [1]. Chen L C, Papandreou G, Schroff F, et al. Rethinking atrous convolution for semantic image segmentation[J]. arXiv preprint arXiv:1706.05587, 2017.
- [2]. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
- [3]. Wang J, Sun K, Cheng T, et al. Deep high-resolution representation learning for visual recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2020.
- [4]. Wang X, Girshick R, Gupta A, et al. Non-local neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7794-7803.
- [5]. Xie S, Girshick R, Dollár P, et al. Aggregated residual transformations for deep neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1492-1500.
- [6]. Zhang H, Wu C, Zhang Z, et al. Resnest: Split-attention networks[J]. arXiv preprint arXiv:2004.08955, 2020.
- [7]. Zhao H, Shi J, Qi X, et al. Pyramid scene parsing network[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2881-2890.