

## 技术报告

### WRMT (榜单上名字为: 奥特曼打怪兽)

任务 1: 使用大赛提供的训练集进行训练, 网络结构为 resnet-50, 修改最后输出层的通道数为 2, 使用了 Cross Entropy loss 作为损失函数进行训练。学习率为 0.001, 批量为 4, 优化器为 Adam, 训练 100 个 epoch, 数据增强方式为: resizecrop、rotation、horizontalflip、centercrop。我们把大赛提供的数据集 Train, 划分为 9: 1 的比例, 在 val 上我们发现准确率已经能达到 100%。因此我们使用初始模型对数据集 Validation 上的图片进行预测, 作为伪标签。然后我们把全部 200 个数据 (Train+Validation) 划分成 9: 1 的比例, 重新训练 resnet-50, 取第 100 个 epoch 的模型作为最后提交结果。

任务 2: 使用大赛提供的训练集进行训练, 虽然大赛提供的是四分类的结构, 但我们把 GCIPL 和 Choroid 中夹着的一层赋值为独立的一类再进行训练, 另外上下背景也区分为不同的两类, 所以最后转换成一个六分类的问题。网络结构我们采用了 UNet, 骨架为 resnet101e, 输入为三通道的图像, resize 成 800\*1120, 使用 ImageNet 的归一化参数进行归一化, 数据增强方式包括: Horizontalflip, ScaleRotate, IAAAdditiveGaussianNoise, IAAPerspective 和颜色增强方案, 学习率为 0.001, 批量为 4, 优化器为 Adam, 每 50 个 Epoch 调整一次, 训练 250 个 epoch。损失函数为 Dice loss。我们首先在数据集 Train 上进行训练, 划分为 9: 1 的比例, 选取在验证集上损失函数最小的 epoch。然后我们提交结果后, 发现效果不错, 我们用初始模型在数据集 Validation 上进行预测, 作为伪标签。因此在第二次训练中, 我们的训练集为 Train 中的 90 例+Validation 中的 100 例 (伪标签), 维持之前的 10 例验证集不变。同样的训练方法。此外, 我们还使用了测试时增强, 把图片左右翻转再预测一次, 把原图和翻转图片的概率输出进行平均, 再最后进行 argmax 的操作得到标签图, 然后我们把六分类的标签, 重新转换成大赛要求的 0, 80, 160, 255 的格式。

以上方法均基于 Python 及 Pytorch。