

# 调参侠队技术报告

## 任务一

### 任务介绍

光学相干断层扫描（OCT）因其无接触、非侵入性的特点，已成为眼部疾病诊疗中的常规检查，可为医生提供视网膜结构图像。与只能提供视网膜表面信息的彩色眼底图像相比，OCT 图像可以提供视网膜的横断面信息，因此可以更准确地分析视网膜结构。层的分割和厚度量化对许多视网膜和视神经疾病的诊断有帮助，例如青光眼、黄斑变性或糖尿病性视网膜病变。在青光眼的诊断中，使用 OCT 比使用眼底彩色图像更容易发现早期病例。本次任务为环扫 OCT 图像的层分割任务，以确定视网膜神经纤维层、神经节细胞丛层和脉络膜层，这有助于青光眼的诊断和区分

### 使用的编程语言及框架

基于 Python 的 Pytorch 框架

### 数据集

1. 中国广州中山大学中山眼科中心提供，数据集中包含 300 个视盘旁环扫 OCT 数据（官方数据）
2. 额外扩展辅助训练的数据集，[MGU-Net – Yuye Ling's Personal Website](#)

### 训练方法（NNUNET）

#### “数据指纹”

- ① 整个训练的第一步是将即将要训练的数据集的前景给抠出来（作者在这里说这个操作对于他们实验的数据来说没有影响，这可能是我在进行预处理时看到前后的图像尺寸没有变化的原因）；
- ② nnUNet 根据这个得到的 crop，来抓取它的相关参数和属性，来组成这个**数据指纹**，这些参数和属性包括：
  - I . image\_size: 每个空间维度上的体素的个数；
  - II . image\_spacing: 每个体素的物理体积大小；
  - III. 模态: 头文件获取到的信息；

- IV. 类别数：这次是四分类，剔除背景类别数为 3；
- V. 此外，还包括在所有训练案例上计算得到平均值、标准差以及属于任何标签的体素的灰度值的 99.5%和 0.5%

## “管道指纹”

- nnUNet 将所有应该进行设计的参数缩减到必要的那几个，并且使用 **启发性规则**对这些参数进行推理，该规则集成了很多行业内专业的知识，同时对上面提到的“数据指纹”和本地的硬件约束都能起到作用。
- 这些得到的**推理参数**被两种参数所完善：
  - 第一种：**蓝图参数**，具有数据独立性，不同数据的该参数不同；
  - 第二种：**经验参数**，训练期间被优化

## 预处理

我们针对视网膜扫描图像的模态为 CT，使用了以下策略：通过统计整个数据集中 mask 内像素的 HU 值范围，clip 出[0.05,99.5]百分比范围的 HU 值范围，然后使用 z-score 方法进行归一化；除此之外，我们的预处理还对图像进行了基础的图像增强方法，包括但不限于：

- ① 旋转
- ② 缩放
- ③ 高斯加噪
- ④ 高斯模糊
- ⑤ 亮度处理
- ⑥ 对比度处理
- ⑦ 低分辨率仿真
- ⑧ Gamma（灰度系数）

## 数据集划分标准

使用的划分标准遵从五折交叉验证的方法：把数据平均分成 5 等份，每次实验拿一份做测试，其余用做训练。实验 5 次求平均值。如上图，第一次实验拿第一份做测试集，其余作为训练集。第二次实验拿第二份做测试集，其余做训练集。

五折交叉验证时使用的划分比例为 2:8；ALL 模式下使用全部数据进行训练，没有验证集。

## 训练过程

从头训练，使用五折交叉验证，loss 函数：结合 dice loss 和交叉熵 loss：Ltotal=Ldice+LCE 对于在全训练集上训练的 3D-UNet（UNet Cascade 的第一阶段和非级联的 3D UNet，不包括 UNet Cascade 的第二阶段），对每一个样本单独计算 dice loss，然后在 batch 上去平均。对其他的网络（2D UNet 和 UNet Cascade 的第二阶段），将一个 batch 内的所有样本当做一个整体的样本计算整个 batch 上的 dice（防止当 crop 后出现局部区域内不存在某一类时单独计算该类 loss 导致分母为零的情况，这也要保证 batch-size 不能太小）。dice loss 形式如下：

$$\mathcal{L}_{dc} = -\frac{2}{|K|} \sum_{k \in K} \frac{\sum_{i \in I} u_i^k v_i^k}{\sum_{i \in I} u_i^k + \sum_{i \in I} v_i^k}$$

其中  $u$  为概率输出（softmax output）， $v$  为硬编码（one hot encoding）的 ground truth。 $K$  为多分类类别数。

我们得到的最优训练结果是五折交叉验证中的 ALL（也就是使用全部数据进行训练）

## 训练流程

### 训练时间表：

I. 根据以往的一些经验，同时为了增加训练的时效性，我们用一轮大于 250 的 mini\_batch，来把网络训练 1000 轮。随机梯度下降的方法采用  $\mu = 0.99$  的 nesterov 的梯度下降，同时初始学习率为 0.01. 学习率的衰减遵循 poly 原则，即

$$(1 - \text{epoch}/\text{epoch}_{\max})^{0.9},$$

设计的损失函数是（交叉熵损失和 + dice\_loss），所以你得到的损失最优为-1，只要是在下降的就是正确的，损失的最小值没有意义。

为了得到每一个深度监督的输出，它对应的每一次分割后 mask 的 ground\_truth 都用来计算损失，训练的对象就是这每一层的损失的和：

$$L = w_1 * L_1 | w_2 * L_2 | w_3 * L_3 | \dots,$$

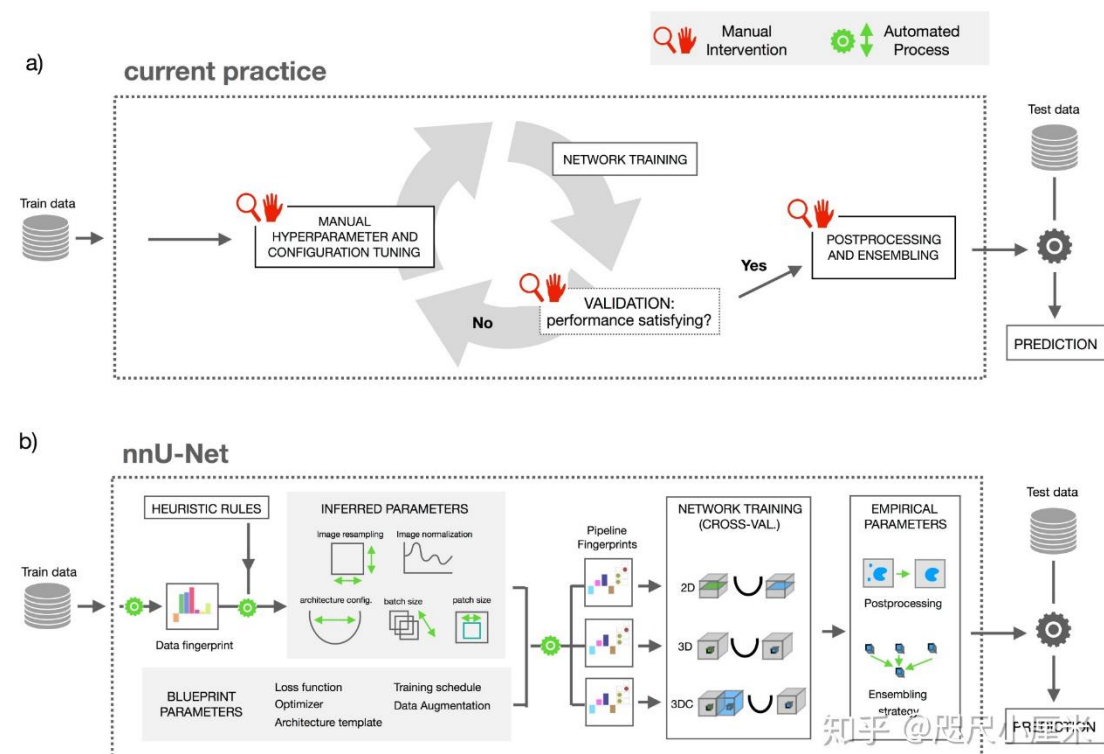
根据这种方式，在每一次分辨率降低（下采样）时权重减半，就会使得：

$$w_2 = (1/2) * w_1; w_3 = (1/4) * w_1, \dots$$

同时将这些权重归一化到和为 1。

II. mini\_batch 的样本都是从训练案例之中随机选择的，通过采用过采样的方式来控制样本不均衡的带来的稳定性问题：Oversampling is implemented to ensure robust handling of class imbalances: 66.7% of samples are from random locations within the selected training case while 33.3% of patches are guaranteed to contain one of the foreground classes that are present in the selected training sample (randomly selected ).The number of foreground patches is rounded with a forced minimum of 1 (resulting in 1 random and 1 foreground patch with batch size 2).

流程图:



## 神经网络

使用的仍是 **Unet**，在此基础上网络结构的一些改变：除了解码阶段的最底下两层，我们给解码器的每一层都加了额外的损失，这样使得梯度信息能够更深层的注入网络，同时促进网络中所有层的训练。所有的 Unet 网络在同一个像素层次上，都用一样的操作，无论是编码区还是在解码区（都是一个卷积 + 一个 instance\_normalization + Leaky\_Relu）。下采样是一个具有步幅长度的卷积，上采样是一次卷积的转置操作。为了平衡训练效果和显存消耗，最初的 feature\_map 的大小被设定为 32，如果要做一次下采样那么这个大小缩小一半，如果做上采样则会变成原来的两倍。为了限制最终生成的模型的大小，feature\_map 的数量也被做了一定的限制，比如，3D\_Unet 被限制在 320 而 2D\_Unet 被限制在 512。

## 优化器

初始优化器: Adam

详细参数:

- self.initial\_lr = 0.01
- self.lr\_scheduler\_eps = 0.001
- self.weight\_decay = 3e-5

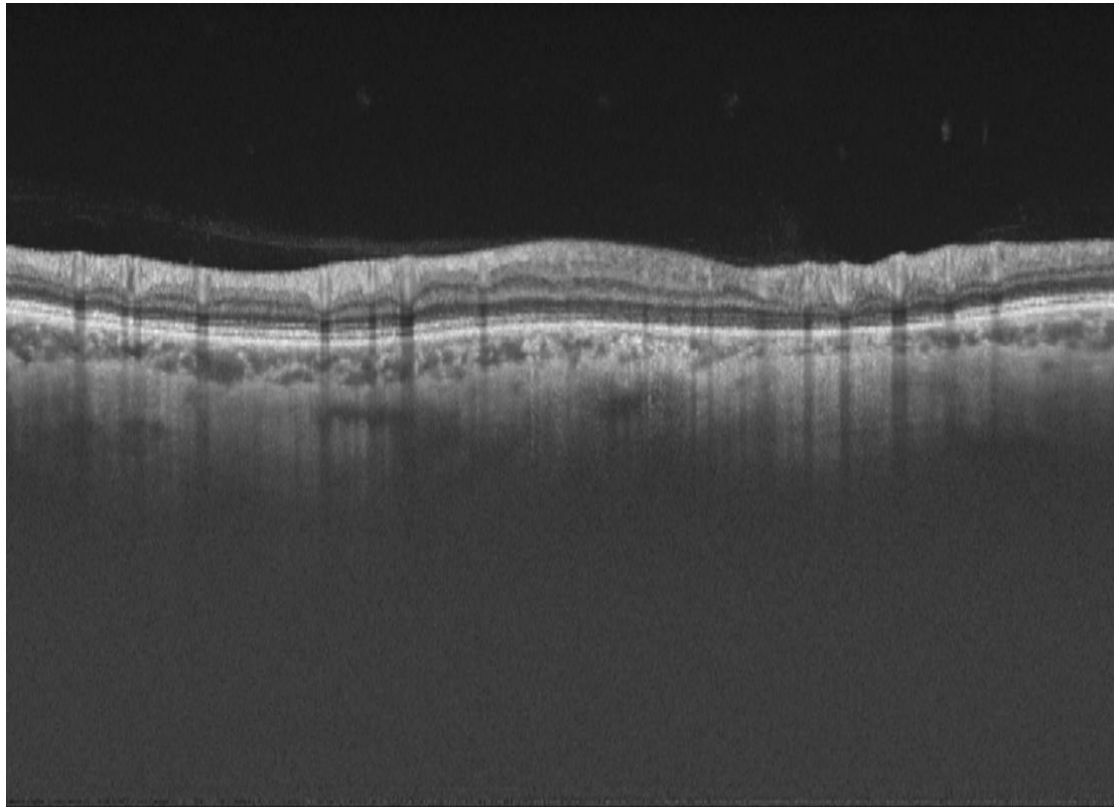
训练轮次: 1000

批量大小：32

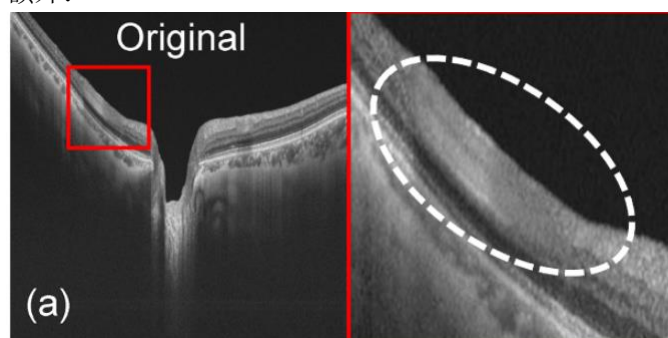
学习率调整策略：学习率的调整是用的损失函数的加权平均值来判断是否变动的方法。并且给每层的损失加一个权重，分辨率越高的权重越大，简单说就是针对中间隐藏层特征透明度不高以及深层网络中浅层以及中间网络难以训练的问题。

## 输入数据展示

官方：



额外：



## 推理

图像的推理预测是通过滑动窗口进行的，而窗口的大小等于训练时的 `patch_size`，相邻的 `patch` 尺寸的预测（即滑动一次的两个块）具有一半的重叠比例。分割的准确率随着窗口边

界的增大而降低。为了抑制拼接伪影，减少靠近边界位置的影响，采用高斯重要度加权，增加 softmax 聚合中中心体素的权重。（我现在的理解是这部分与推理一并进行，一个个的 patch 类似于一 yolo 的滑动窗口，但是这部分的时间真的太慢了！！！）通过沿所有轴进行镜像来增加测试时间，我将镜像去掉剪短了推理时间，一定程度上也损失了精度。

## 经验参数

- ① 推理模式：  
nnUNet 会在推理时根据训练集中的验证 dice 表现来自动选择配置——是选择单个推理方式还是选择一起推理方式。  
四种单个推理模式：2D、3D\_full resolution、3Dlangl\_lower resolution & the full resolution of the cascade。  
合作推理模式：四个模式中两两进行组合来实现合作推理。  
通过平均 softmax 概率对模型进行综合。
- ② 后处理：  
Connected Components 算法（连通分支算法）在医学图像领域非常常用，尤其是在器官分割中，经常用于去除最大的连接组件，从而来消除假阳性。nnU-Net 同样用到这个方法，并自动对交叉验证中抑制较小组件的效果进行测试：首先所有的前景会被当做一个组件（多类别的 1、2、3 都被当做 1），如果对除最大区域以外的所有分支的抑制提高了前景的平均 dice 而没有减小任何类别的 dice，这一步将会被选择作为第一步的后处理步骤。最终，nnUNet 会依靠这个步骤的表现来衡量要不要把相同的步骤用于不同的类别上。

## 任务二

任务介绍：任务二是一种基于 OCT 图像的青光眼诊断分类任务，定位十分尴尬，使用官方 baseline 轻松满分

语言：python 框架：PaddlePaddle

数据集：我们所使用的数据集由中国广州中山大学中山眼科中心提供，数据集中包含 300 个视盘旁环扫 OCT 数据

Train 和 val 的划分：我们测试了 7:3 和 8:2 后发现 8:2 更好，因此我们选择了 8:2

对于输入数据：我们对图片首先把图片 resize 成了 256X256，然后进行了随机垂直或水平翻转、随机角度的旋转和随机裁剪

优化器：adam 学习率 1e-6

神经网络：resnet50 移除最后一层全连接，让 numclass 变成 2。

## 其他内容

后续我们会通过网盘链接上传模型和代码文件以及预赛结果，目前在我们的服务器上，但是服务器关机了，因此无法及时上传。

## 参考文献

【1】Li, Jiaxuan, et al. "Multi-Scale Gcn-Assisted Two-Stage Network For Joint Segmentation Of Retinal Layers And Discs In Peripapillary Oct Images." 12 (2021): 2204-2220. [MGU-Net – Yuye Ling's Personal Website](#)

【2】[GitHub - MIC-DKFZ/nnUNet](#)