

# 项目说明

## 背景介绍

### 项目背景与研究内容

#### 9.2 面向环扫影像的多结构分割智能系统研发

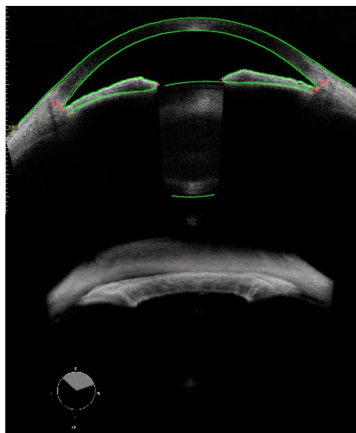
带头人：胡凌溪

##### ■ 研究背景

现有前房室分割模型大都基于二维影像且局限于单一结构分割，无法对疾病诊断提供精准且全面的辅助信息。环扫影像能弥补二维影像的三维信息缺失，因此，研发基于环扫影像的多结构分割算法，可以获取精准且全面的病理结构信息。

##### ■ 研究内容

- I. 基于环扫OCT重建前房室三维影像；
- II. 基于三维影像信息利用深度学习、马尔科夫链、临床先验研发可信性多结构分割算法；
- III. 开发一个多结构分割智能可视化系统。



### 小组成员与分工

组长：朱家润

组员：冯星洋、范书豪、刘乐平

朱家润：UNet与UNet++模型编写

冯星洋：对中、去除离群点算法编写，3D重建代码改写与优化

刘乐平：UNet模型参数调优

范书豪：UNet++模型参数调优

### 主要任务

使用经人工分割的眼前节OCT环扫图像训练、测试、应用模型，并将分割后的图像应用于三维重建任务。

### 项目实现

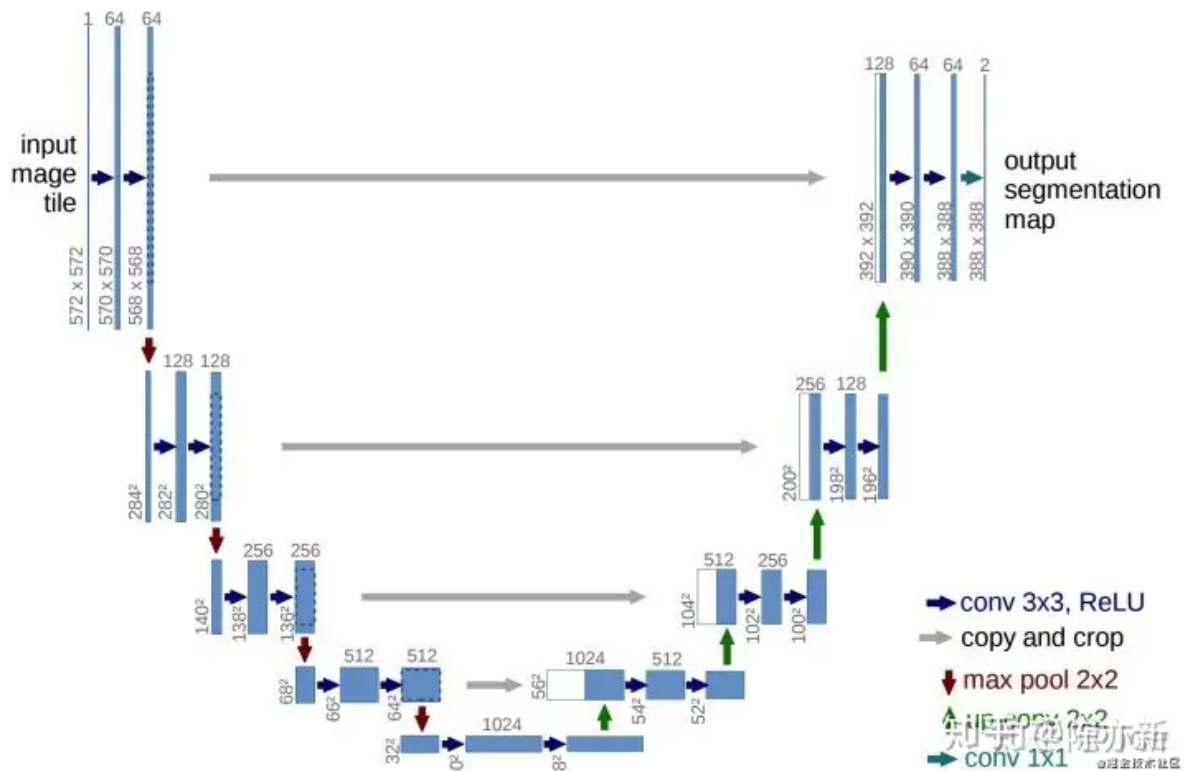
#### 技术路线

## 分割

将经人工分割的眼前节OCT环扫图分为train（训练集）和val（验证集），训练UNet/UNet++模型，并得到相应结果。

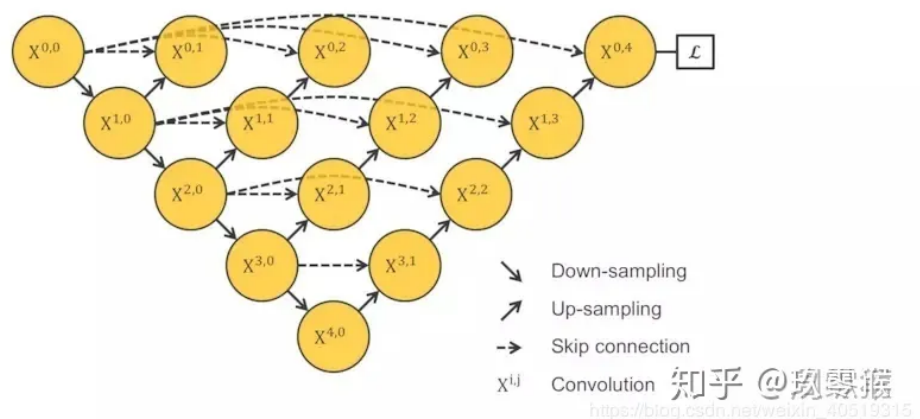
### UNet:

UNet模型由左侧实现下采样的解码器、右侧实现上采样的编码器，以及中间的跳跃连接层组成。



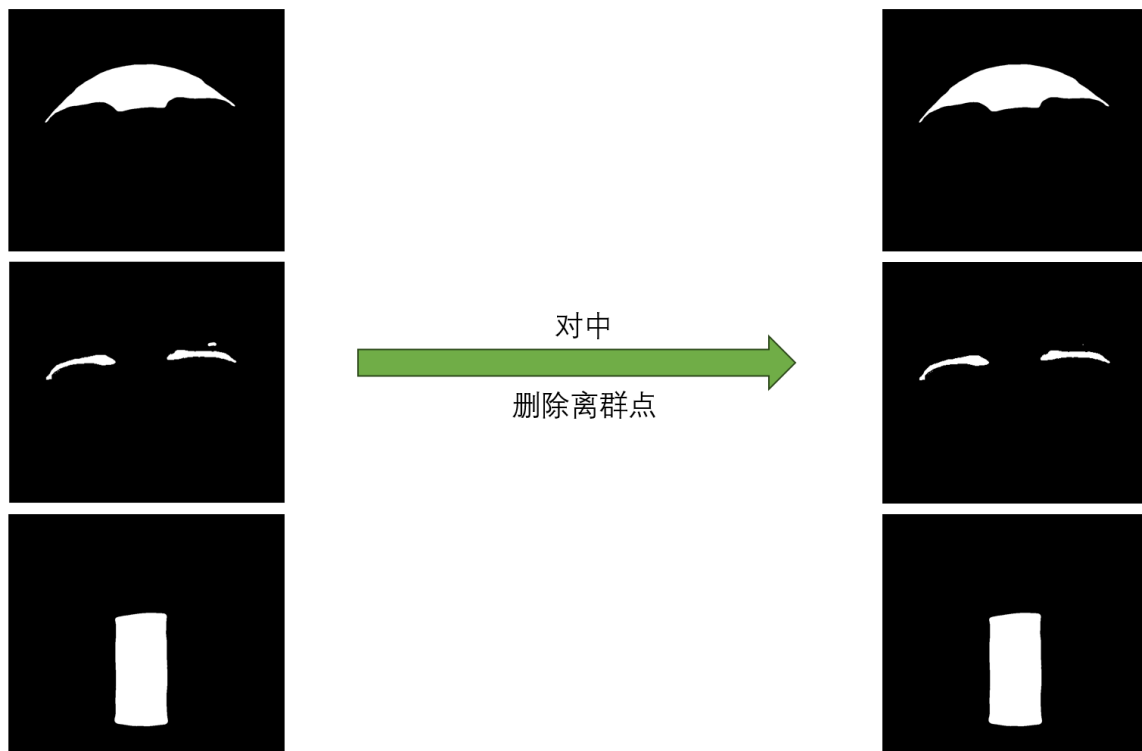
### UNet++:

UNet++模型由多个不同深度的UNet模型叠加而成。UNet++提供了一个可剪枝的网络结构，我们可以人为考察各输出层的效果后在“分割精度”和“运行性能”中做出取舍，选取特定深度的模型。



## 图像处理

图像处理部分由图像对中和删除离群点两个步骤组成。



### 图像对中

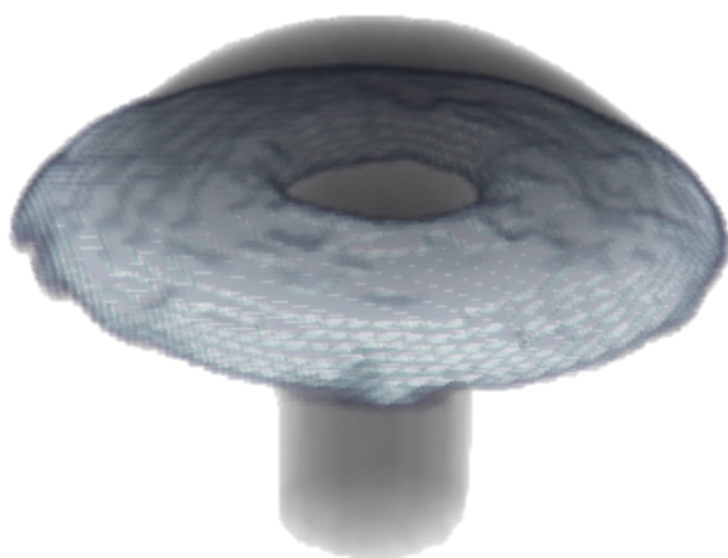
寻找图像左右特征点，求平均并以此作为图像实际中轴。通过图像平移等操作将实际中轴与图像中心对齐。

### 删除离群点

将遮罩图像的白色像素点建图，并为像素点相邻的点连边。对各个子连通图统计节点个数，获得区域面积。保留面积前二大的区域，并删除其他区域。

### 3维重建

将处理后图像按顺序打包为3维numpy数组，并通过笛卡尔坐标转化为柱坐标的方式将numpy数组各像素点还原至环扫前位置。利用线性插值等插值算法，填补变换后的空白位置。利用pyvista等python包，实现3维可视化展示。



## 项目结构

CS103_Project	# 根目录
├─ imgs	# 保存README.md所需的图片
├─ MS_dataset	
├─ train	# 训练数据集
└─ val	# 验证数据集
├─ Re3D	
├─ build	# 3维重建与展示
└─ transform	# 图像对中与删除离群点
├─ UNet	
├─ checkpoints	# 用于保存存档点
├─ imgs	# 用于保存分割后图像
├─ dataset.py	# 数据集
├─ hyper_parameters.py	# 超参数与常量
├─ model.py	# UNet模型
├─ test.py	# 测试模型
├─ train.py	# 训练模型
└─ utils.py	# 工具包
├─ UNetPP	
├─ checkpoints	
├─ imgs	
├─ dataset.py	
├─ hyper_parameters.py	
├─ model.py	# UNet++模型
├─ train.py	
└─ utils.py	
├─ README.md	# 项目说明
└─ requirements.txt	# 项目环境

## 运行方法

1. 通过requirements.txt配置项目环境。
2. 将数据放入MS\_dataset的train与val中（由于保密等相关原因，本小组不在此预置数据）。注：同一组OCT图像应当位于同一个二级文件夹下，为保证3维重建的稳定性，请务必确保环扫顺序与文件命名顺序相匹配。
3. 将hyper\_parameters.py中的LOAD\_MODEL项改为False。
4. 运行train.py，程序会对训练集运行10个循环，运行完毕后会在checkpoints目录下生成checkpoint.pth.tar文件（存档点），并运行验证集验证准确度，在imgs目录下生成运行结果（原始图像+遮罩图像+高亮图像）。
5. 如需继续训练，将LOAD\_MODEL项改为True（从存档点加载模型），然后重复运行第4步。
6. 通过修改MSDataset.getItem()中的mask\_target布尔数组为 <150 或 >=150 and <170 或 >=170 后从第3步重新运行，实现对眼前节不同部分的分类。
7. 将各部分图像分类取出并存入特定文件夹中。
8. 修改transform.py中的PATH和SAVE参数与build.py中相应参数以匹配第7步的文件夹路径。
9. 运行transform.py/build.py以查看结果。注：transform.py仅实现了对虹膜图像的对中与删除离群点，若需同步应用于其他部分，需要相应代码改进。