**文件结构**：

* oriData

- train

- img

- mask

- test

- img

- mask

* augData

- train

- img

- mask

- test

- img

- mask

* trainUnet.ipynb
* dataPrepare.ipynb

**dataPrepare:**

dataset数量和丰富度不够，使用keras.preprocessing.image.ImageDataGenerator进行数据增强（旋转、移动、缩放）

1. 利用data\_gen\_args配置数据增强的参数，包括

* rotation\_range: 旋转角度范围，用于指定图像随机旋转的角度范围。
* width\_shift\_range: 宽度偏移范围，用于指定图像水平方向上的随机平移范围。
* height\_shift\_range: 高度偏移范围，用于指定图像垂直方向上的随机平移范围。
* shear\_range: 剪切范围，用于指定图像剪切变换的程度范围。
* zoom\_range: 缩放范围，用于指定图像随机缩放的程度范围。
* horizontal\_flip: 水平翻转，用于指定是否对图像进行水平翻转。
* fill\_mode: 填充模式，用于指定当进行平移或剪切操作时，超出边界的像素填充的方式。

1. 将data\_gen\_args作为参数传递给trainGenerator函数，在训练数据生成器中应用这些数据增强操作
2. 设置batch为3，以生成和保存三个批次的图像和对应的掩码

**trainUnet:**

使用tensorflow搭建深度为4的Unet模型。U-Net是一种常用的图像分割模型，由编码器和解码器组成。编码器用于提取图像的特征表示，而解码器用于将特征映射恢复到原始图像的尺寸，并生成分割掩码。

1. 将npy转成png
2. 加载Unet所需库函数
3. 定义Unet模型

* 输入层：根据指定的input\_size (512,512,1)创建输入层。
* 编码器部分：通过一系列的卷积层和池化层实现特征提取。其中，conv1到conv5是编码器的卷积层，分别具有不同的通道数和激活函数。每个卷积层后面跟着BatchNormalization层，用于加速收敛和稳定训练过程。
* dropout层：在第4个卷积层后面加入了一个dropout层，用于随机丢弃一部分特征图，以防止过拟合。
* 解码器部分：通过上采样和卷积层实现特征映射的恢复。每个卷积层后面跟着BatchNormalization层。其中，up6到up9是解码器的上采样层，merge6到merge9是解码器的特征融合部分。
* 最后的卷积层：conv10是最后的卷积层，输出二分类的分割掩码。
* 模型编译：使用Adam优化器，设置较小的学习率，使用二元交叉熵作为损失函数，计算准确率作为评估指标。

1. 进行数据预处理，将图像和掩码进行归一化，并掩码转换为01的二值图像
2. 定义用于模型评估的函数

* get\_confusion\_matrix\_elements函数用于计算混淆矩阵的元素，即真负例（TN）、假正例（FP）、假负例（FN）和真正例（TP）。
* get\_prec\_rec\_IoU\_accuracy函数用于计算精确度（precision）、召回率（recall）、交并比（IoU）和准确率（accuracy）等评估指标。读取混淆矩阵，根据公式计算相应的指标。
* get\_f1\_score函数用于计算F1分数。读取混淆矩阵，根据公式计算F1。

1. 初始化Unet
2. 初始化Tensorboard，以便监控模型的损失变化。
3. 用训练集训练Unet模型

* model\_checkpoint在每个训练周期结束时保存具有最低损失值的模型权重。save\_best\_only=True表示仅保存具有最低损失值的模型权重。
* model.fit用于训练模型。它接受数据生成器（data\_gen）作为输入，steps\_per\_epoch=10表示每个训练周期中的步数，epochs=80表示训练周期的数量。
* callbacks=[model\_checkpoint,tensorboard\_callback]将回调函数传递给model.fit函数，以便在训练过程中保存最佳模型权重和记录日志到Tensorboard。

1. 在测试图像上运行经过训练的模型并保存输出，并评估像素级分割性能