

### 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。
2. 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
3. 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 `numpy` 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。
4. 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
  5. 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\_loU\_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。
  6. 初始化 Unet
  7. 初始化 Tensorboard, 以便监控模型的损失变化。
  8. 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。
  9. 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