

眼底血管分割实验报告

姓名：蒲尧 学号：2020E8017782032

一、摘要

研究表明，各类眼科疾病以及心脑血管疾病会对视网膜血管造成形变、出血等不同程度的影响。随着生活水平的提高，这类疾病的发病率呈现逐年增长的趋势。临床上，医疗人员能够从检眼镜采集的彩色眼底图像中提取视网膜血管，然后通过对血管形态状况的分析达到诊断这类疾病的目的。但是，由于受眼底图像采集技术的限制，图像中往往存在大量噪声，再加之视网膜血管自身结构复杂多变，使得视网膜血管的分割变得困难重重。最传统方法中，依靠人工手动分割视网膜血管，不仅工作量巨大极为耗时，而且受主观因素影响严重。传统方法中，利用提取设计的已知的图像特征，从而得到所需要的血管区域图像，效果较好，但还是存在手动调参，特征选取等问题。因此，利用计算机技术，找到一种能够快速、准确分割视网膜血管的算法，实现对眼底图像血管特征的实时提取，对辅助医疗人员诊断眼科疾病、心脑血管疾病等具有重要作用。在本文中，我们参考一些现有文献，采用传统方法与深度学习框架相结合的方法进行眼底血管分割实验。本人选择在医学领域较为广泛使用的 Unet 及其升级版 Unet++ 作为基本深度学习框架，其编码器和解码器都是 3×3 双层卷积，而且这两个网络都顾及了前层特征与后层特征的融合。本人在将原始图像输入网络训练测试之前，还进行了 4 步预处理：首先，RGB2Gray 将 3 通道彩色图变为 1 通道灰度图；其次，进行图像归一化处理，保证图像的仿射不变性，提高计算精度；再次，通过限制对比度自适应直方图均衡化（CLAHE）进行血管增强处理；最后，利用局部自适应伽马校正提升图像亮度信息并降低伪影的干扰。该算法在 CHASE 和 DRIVE 的测试集上的平均 Dice 达到 78.03% 和 77.45%，平均 Accuracy 达到 96.91% 和 95.83%。本文代码：<https://github.com/Allenem/MedicalImage/tree/main/DUNet-retinal-vessel-detection>。

关键词：眼底血管分割，限制对比度自适应直方图均衡化，Unet，Unet++。

二、背景

2.1 传统方法

虽然传统方法较为复杂，但其实现的效果之好是不可否认的。常见的眼底血管提取方法有如下几个方法步骤：1.去噪（常见的有高斯滤波），2.灰度图直方图均衡化大类（常见的有直方图均衡化(HE)，自适应直方图均衡化(AHE)，限制对比度自适应直方图均衡化(CLAHE)，自适应局部区域伸展(Local Region Stretch)直方图均衡化等)，3.对比度增强，4.图像归一化，5.形态学操作等。

1.高斯滤波去噪

根据眼底图像噪声分布不均匀、可叠加、服从正态分布等特性，可以采用高斯滤波去噪。高斯滤波是一种简单快速的线性平滑滤波去噪算法，对抑制服从正态分布的噪声很有效果。高斯滤波的原理是：离散化高斯函数，把离散点上的高斯函数值作为加权值，对采集到的眼底图像上每个像素点的灰度值做一定范围内的加权平均，用所得的函数值替换原来像素点的灰度值，即可消除噪声。公式如下：

$$h(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

$$f_i(x, y) = h(x, y) * f(x, y) \quad (2)$$

上式中， $f(x, y)$ 表示原图， $h(x, y)$ 表示高斯函数， $f_i(x, y)$ 表示高斯滤波后的图像。

2.直方图均衡化

直方图均衡化（HE）：就是将不同灰度数值分布不均匀的图像，经过映射变换，转为在 0~255 分布较为均匀的图像，从而实现不同数值的图像都能较为清晰展示，视觉观感明显。

自适应直方图均衡化（AHE）：在前面介绍的直方图均衡化中，是直接对全局图像进行均衡化，是 Global Histogram Equalization，而没有考虑到局部图像区域(Local Region)，自适应过程就是在均衡化的过程中只利用局部区域窗口内的直

方图分布来构建映射函数。

限制对比度自适应直方图均衡化(CLAHE): 为了避免由于 AHE 产生的图像不连续和过度增强的结果, 引入一种限制直方图分布的办法。CLAHE 来自文章 *Adaptive Histogram Equalization and Its Variations*[1], 是 1987 年 一篇论文, 相对于 AHE, 提出了两个改进的地方: 第一, 提出一种限制直方图分布的方法。考虑图像 A 的直方图, 设定一个阈值, 假定直方图某个灰度级超过了阈值, 就对之进行裁剪, 然后将超出阈值的部分平均分配到各个灰度级。通常阈值的设定可以直接设定灰度级出现频数, 也可以设定为占总像素比例, 后者更容易使用。改变后的图像所对应的 Cumulative Distribution Function (CDF) 图不会有太大的剧烈变化, 所以可以避免过度增强噪声点。第二, 提出了一种插值的方法, 加速直方图均衡化。首先, 将图像分块, 每块计算一个直方图 CDF, 这里简称每个小块为窗口。其次, 对于图像的每一个像素点, 找到其邻近的四个窗口(边界先不讨论), 分别计算左上、右上、左下、右下四个窗口直方图 CDF 对某像素点的映射值, 记, 然后进行双线性插值得到最终该像素点的映射值。CLAHE 涉及到窗口大小、影响区域大小、直方图阈值三个部分的参数, 可以通过调参更好地去控制对比度增强的结果。

自适应局部区域伸展(Local Region Stretch)直方图均衡化: 上面提到的 AHE 和 CLAHE 都是基于块状区域进行直方图均衡化的, 但是能不能根据灰度级区域近似的区域进行均衡化呢? 比如对图像中灰度级 $[\min, \max]$ 范围里面的所有像素点进行均衡化, 使得像素点的直方图尽量在 $[\min, \max]$ 上均匀分布。在论文 *Adaptive Contrast Enhancement Using Local Region Stretching*[3]中, 根据亮度(Brightness)对图像进行分割成几个区域, 然后分别做直方图均衡化。下面根据论文的主要思想, 引入下面的方法: 统计图像直方图, 按照灰度级划分为三个灰度区间, 使得三个区间的像素点数量近似相等, 这样就分别在 $[0, \text{level1})$, $[\text{level1}, \text{level2})$, $[\text{level2}, 255]$ 三个灰度区间做直方图均衡化, 最后合并。

3.对比度增强

图像对比度指的是图像各部分之间灰度级反差的程度,图像对比度增强即增加图像各部分间反差。由于对眼底图像某些处理后, 目标血管与背景的对比度会

减弱，模糊血管边界，加大分离血管的难度，因此需要对眼底图像进行对比度增强，突出眼底图像中目标血管，同时去除或弱化不需要的干扰信息，使眼底图像上目标血管更加明显。

一般的图像对比度增强方法为全局映射法，针对单个像素，对图像上每个像素点使用相同的变换，不用考虑空间邻域关系，实现简单。但要找到对全局像素值都能进行合理映射的算法，计算量大，操作麻烦。

该文[4]采用一种基于匹配滤波的局部映射算法，可增强代表细节的高频部分，其原理为：设置一定大小的滤波器窗口(该文选取 15x15 的线性滤波窗口)，以某个像素点为中心，划分出对应滤波器窗口大小的局部区域，通过不同方向的核函数与该区域进行匹配增强，即滤波器窗口区域内图像乘以某个增益值，保留每个像素点的最大值，即为该点增强后的像素值，达到对比度增强的目的。

4.图像归一化

对图像进行归一化处理，保证眼底图像的仿射不变性，提高计算的精度。传统的归一化方法包括翻转归一化、尺度归一化和旋转归一化，属于简单的归一化操作，难以实现复杂的过程。

该文[4]采用一种基于不变矩的图像归一化方法，其原理为：利用图像的不变矩找到一组参数，使其能消除其他变换函数对图像的影响，通过该参数确定变换函数，将原始图像转换成唯一标准形式的图像。该处理方法包括四个步骤，即坐标中心化、*x-shearing* 归一化、缩放归一化和旋转归一化。

5.形态学操作

形态学操作是提取眼底图像目标血管的重要工具，包括腐蚀、膨胀和开闭运算。通过形态学操作可区分眼底图像中的目标血管与其它背景，将目标血管与背景分离开。

眼底图像分为血管、视盘区、黄斑区和眼底组织四大部分。血管是线性放射结构，视盘区和黄斑区均为圆形结构，眼底组织是变化均匀的大背景区域。该文[4]采用形态学背景去除法。通过选取直径大于血管、小于视盘区和黄斑区的圆盘结构，先对血管进行多次膨胀操作，接着对膨胀后图像进行多次腐蚀操作，得到

复原的背景图，最后再用原眼底图像减去复原的背景图像，即可得到去除背景区域的眼底图像，也就是血管图像。

2.2 深度学习方法

1. DenseNet-U-Net

基于改进的 *U-Net 眼底视网膜血管分割*[5]论文介绍如下。针对视网膜血管图像特征信息复杂程度高，现有算法存在微血管分割较低和病理信息误分割等问题，该文提出一种融合 DenseNet 和 U-Net 网络的血管分割模型。首先，通过限制对比度直方图均衡化和 filter 滤波对图像进行血管增强处理；其次，利用局部自适应 gamma 提升图像亮度信息并降低伪影的干扰；再次，由多尺度形态学滤波局部增强微血管特征信息；最后，利用 U 型密集链接模块进行分割。该算法在 DRIVE 数据集上实验，其平均准确率、灵敏度和特异性分别高达 96.74%、81.50% 和 98.20%。

针对现有卷积神经网络存在储存开销大、计算效率低等不足，该文提出将 U-Net 与 DenseNet 相结合，设计了一个 U 型密集链接模块(U-Net dense block, UDB)。该模型结合两者的优势，提高了网络的泛化性能，在减少训练时间的同时使得网络中每一层的血管特征都能得到充分利用，较好地解决了眼底视网膜图像存在病变与伪影的干扰，从而能更多地保留血管特征信息，获取更多微小血管，而且在训练集较少的视网膜图像中不易发生过拟合现象，提高了测试集的分割精度，具有较强的鲁棒性。该文章的核心部分如图 1：

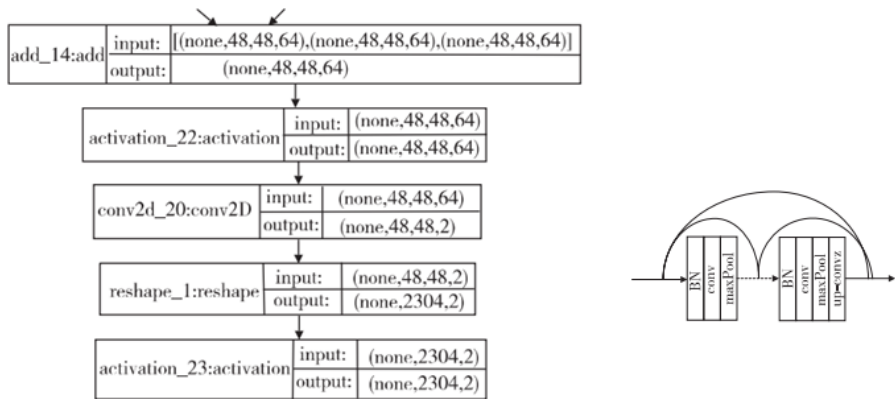


图 1 上采样合并过程，U 型密集模块

2. DUNet

在 *DUNet: A deformable network for retinal vessel segmentation* [6] 该文中, 作者提出了一种可变形的 U 型网络(DUNet), 它利用视网膜血管的局部特征, 以一种端到端方式进行视网膜血管分割。受最近引入的可变形卷积网络的启发, 作者将可变形卷积集成到所提出的网络中。DUNet 通过上采样算子提高输出分辨率, 通过结合低层次特征和高级特征提取上下文信息, 实现精确定位。此外, DUNet 通过根据血管的规模和形状自适应地调整感受野来捕获不同形状和规模的视网膜血管。公共数据集:DRIVE, STARE, CHASE_DB1 和 HRF 被用来测试其模型。DRIVE、STARE、CHASE_DB1 和 HRF 数据集的全局准确率为 0.9566/0.9641/0.9610/0.9651, 其 AUC 为 0.9802/0.9832/0.9804/0.9831。

该架构由 U-Net 框架中的卷积编码器(左侧)和解码器(右侧)组成。在每个编码和解码阶段, 通过学习局部的、密集的和自适应的接受野, 使用可变形的卷积块来模拟不同形状和规模的视网膜血管。该文提出可变形模块。每个可变形卷积块由卷积偏移层(可变形卷积的核心概念 ConvOffset)、卷积层(Conv)、批处理归一化层(BatchNormalization)和激活层(ReLU)这 4 个部分组成。在 DUNet 的底部, 使用了正常的卷积层而不是变形块, 因为会引入大量的参数而没有实质性的性能改进。通过这种结构, DUNet 能够学习可识别性特征, 生成较精确的视网膜血管分割结果。

该文章的基本架构如图 2:

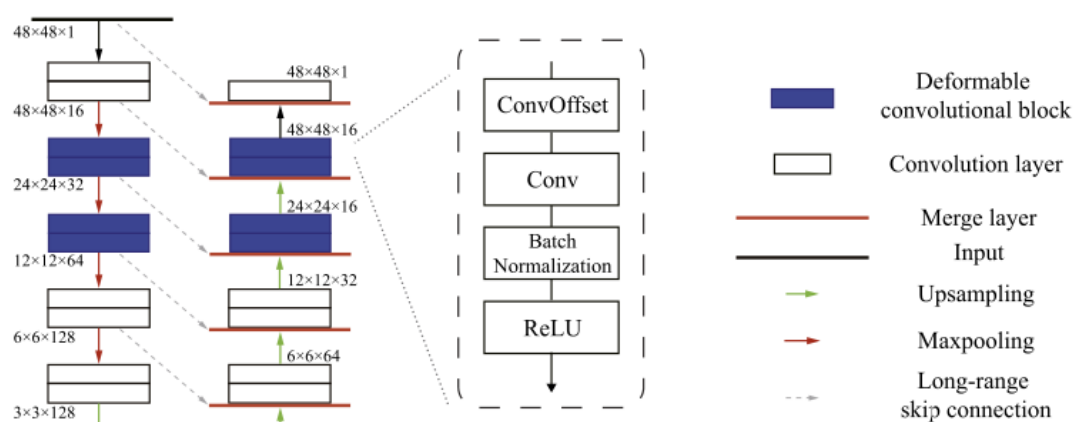


图 2 基于 U-Net 架构, 含有卷积编码器和解码器, 的 DUNet 架构。feature map 的输出大小在每两层旁边列出。

三、实验

本次研究我们针对 CHASE 和 DRIVE 数据集进行实验，CHASE 的 train 有 10 对左右眼共计 20 幅图像，test 有 4 对左右眼共计 8 幅图像；DRIVE 的 train 有 20 幅图像，test 有 20 幅图像。我们统一将大图分割为 256×256 的 patch，我们并没有 resize，而是裁剪。一是为了增加数据量（以上 4 个文件夹文件分别重整为 $20 \times 36 = 720$, $8 \times 36 = 288$, $20 \times 9 = 180$, $20 \times 9 = 180$ ，虽然我的实验数据量并不大，但可满足训练，以后将进行改进），二是也保留了原图的细节特征。

3.1 预处理

1.三通道变一通道

有论文[7]指出，单通道能够更加清晰地体现眼背景差异。因此，在本次实验中，本人将彩色图 3 通道按比例融合变为 1 通道灰度图。其他实验中也有采用 G 通道的做法，以后的实验中将进一步验证：只采用单个某个通道效果好，还是按照比例融合的单通道效果好。

2.归一化处理

进行图像归一化处理，能够保证图像的仿射不变性，提高计算精度。本人在实验只是进行了普通的减均值除以方差的归一化操作，以后实验争取将平移归一化（即坐标中性化、x-shearing 归一化）、尺度归一化（即缩放归一化）还有旋转归一化添加进去，以得到更好的实验效果。

3.自适应直方图均衡化

通过限制对比度自适应直方图均衡化（CLAHE）进行血管增强处理，在这步实验中，我们使用 OpenCV 的自带函数 `cv2.createCLAHE(clipLimit,tileGridSize)`，clipLimit: 限制对比度的阈值，默认为 40，直方图中像素值出现次数大于该阈值，多余的次数会被重新分配；tileGridSize: 图像会被划分的 size，进行像素均衡化

的网格大小，即在多少网格下进行直方图的均衡化操作，如 `tileGridSize=(8,8)`，默认为(8,8)。这步实验之后，图像明显对比度增强很多。效果图可见第四章结果部分。

4.自适应伽马矫正

利用局部自适应伽马矫正，提升图像亮度信息并降低伪影的干扰。

灰度增强中主要分为线性变换，对数变换和指数变换，其中指数变换又称为伽马变换或伽马校正。伽马变换中有补偿系数和伽马系数两个可调参数，伽马变换可以根据伽马(gamma)系数的不同取值选择性地增强低灰度区域的对比度或者高灰度区域的对比度。但是传统的 gamma 变换的 gamma 系数的调节是手动调节，而不同图像的明暗程度不一定，使用传统的 gamma 变换进行手动调参耗时且十分低效，工程应用性差。

本人经过几次随机实验设置了一个参数，效果还不错，但并不保证是最优参数，最优参数还有待进一步量化实验。

5.分割小 patch

```
max_x = int(((img.shape[0] - height) / stride) + 1)
max_y = int(((img.shape[1] - width) / stride) + 1)
max_crops = max_x * max_y
```

通过以上几步，得到每张图像在 x, y 方向可以分割的个数及总个数。CHASE: $x=\text{int}((999-256)/128+1)=6$, $y=\text{int}((960-256)/128+1)=6$ ，所以 CHASE 一张图像分为 36 张小图，总个数 $20*36=720$, $8*36=288$; DRIVE: $x=\text{int}((565-256)/128+1)=3$, $y=\text{int}((584-256)/128+1)=3$ ，所以 DRIVE 一张图像分为 9 张小图，总个数 $20*9=180$, $20*9=180$ 。

然而这里实验中我没有注意到 `int` 是向下取整，所以裁剪会把原图一部分丢失，之后打算采用 `math.ceil()` 函数进行向上取整计算，以充分保留原图信息。

3.2 网络设计

1.Unet[8]

Unet 网络结构如其名，呈现一个 U 字形，即由卷积和池化单元构成，左半边为编码器即如传统的分类网络是“下采样阶段”，右半边为解码器是“上采样阶段”，中间的灰色箭头为跳跃连接，将浅层的特征与深层的特征拼接，因为浅层通常可以抓取图像的一些简单的特征，比如边界、颜色。深层经过的卷积操作多抓取到图像的一些暂时无法言语解释的抽象特征，将浅深同时利用起来为上上策，同时允许解码器学习在编码器池化下采样中丢失的相关特征。纵观整个网络 Unet 其实有点带有残差(residual)结构的思想。

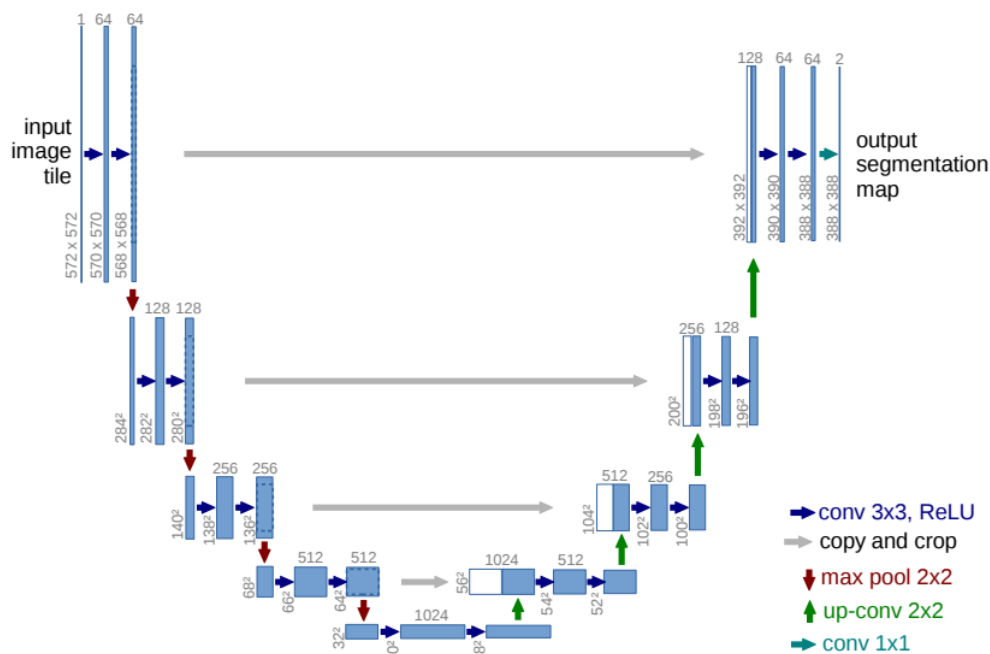


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

图 3 Unet 整体架构

2.Unet++[9]

该文作者在考虑，为何 Unet 偏偏就选 4 个下采样，于是做了不同数据集不

同深度的对比实验。实验表明，不同数据集，深度增加不一定效果更好。浅有浅的侧重，深有深的优势，因此如何将不同深度的 Unet 融合在一起就成了重点。

作者先将不同深度融合的子网络，结合到一块，如图 4 所示。但是该模型结构是存在一些问题的，中间的子模型并没有与输出连接，这样中间部分是无法通过反向传播更新权重的。

与此同时，图 5 这个结构由 UC Berkeley 的团队提出[10]，但二者相互独立。

Unet++与[10]结构（图 5）不同之处就在于：图 5 去掉了 Unet 的长连接全部使用短连接；而 UNet++的作者认为长连接也是有必要的，它联系了输入图像的很多信息，有助于还原降采样的信息损失，和残差的操作非常类似。如图 6 (a)。更有精妙之处，该结构可以进行网络拼接和裁剪，如图 6 (c)。

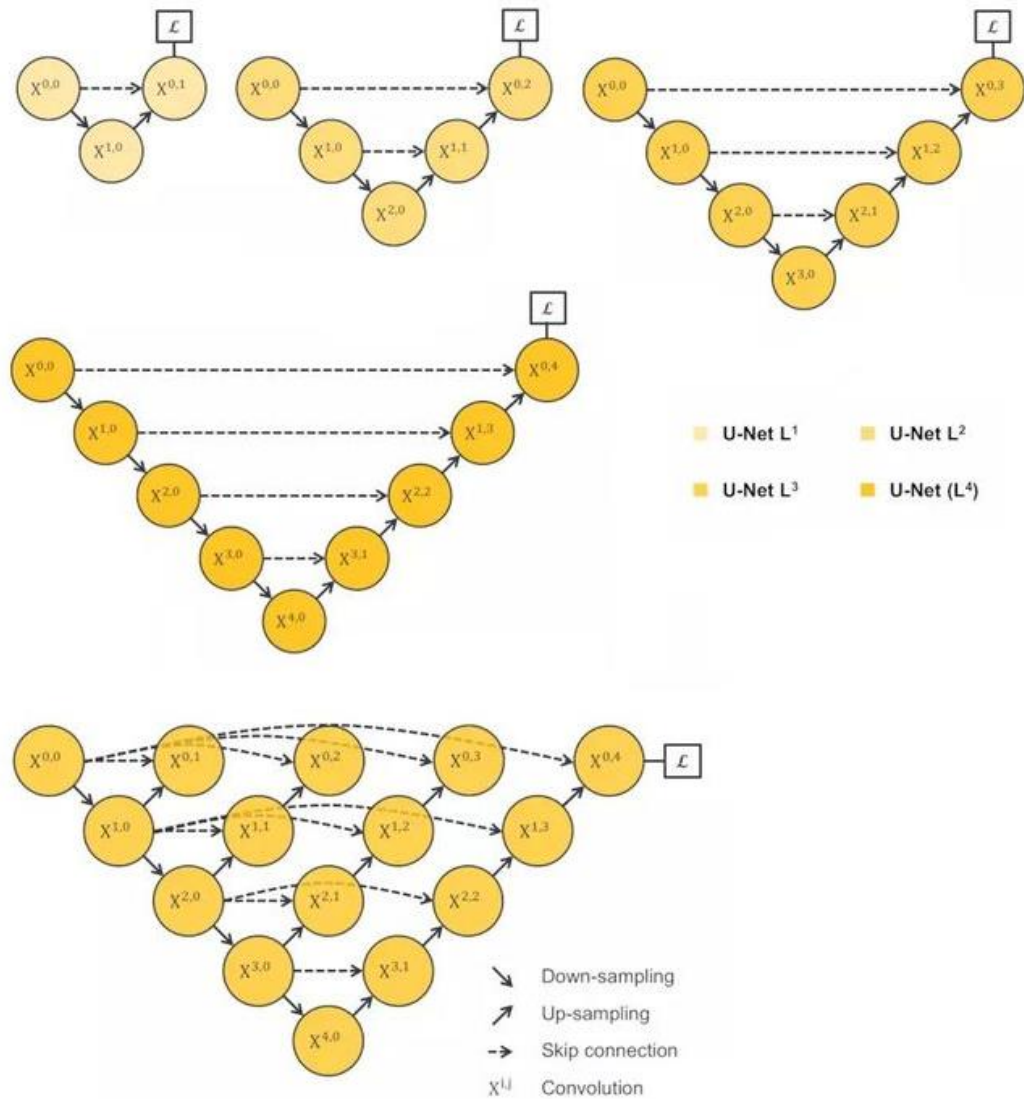


图 4 简单地结合，不同深度融合的子网络

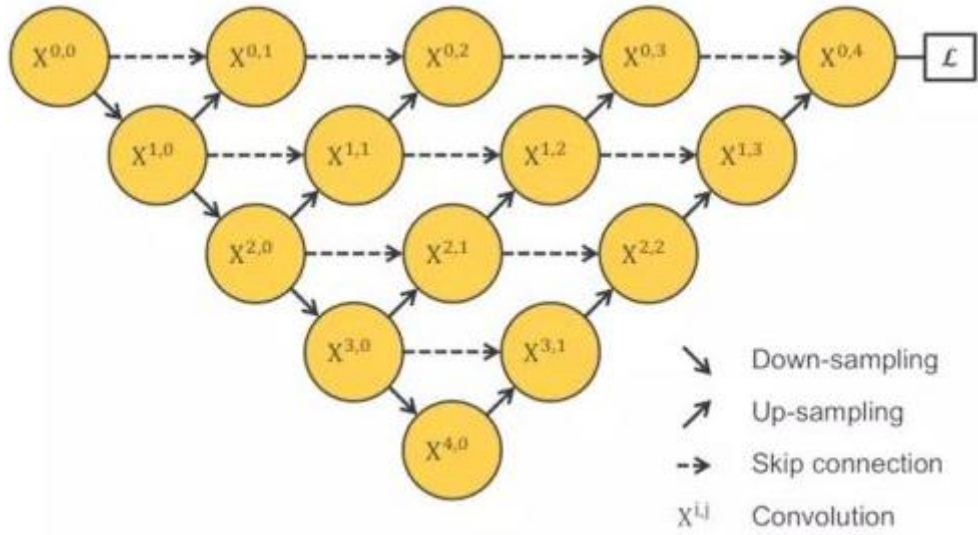


图 5 通过迭代深度聚合进行插值网络结构

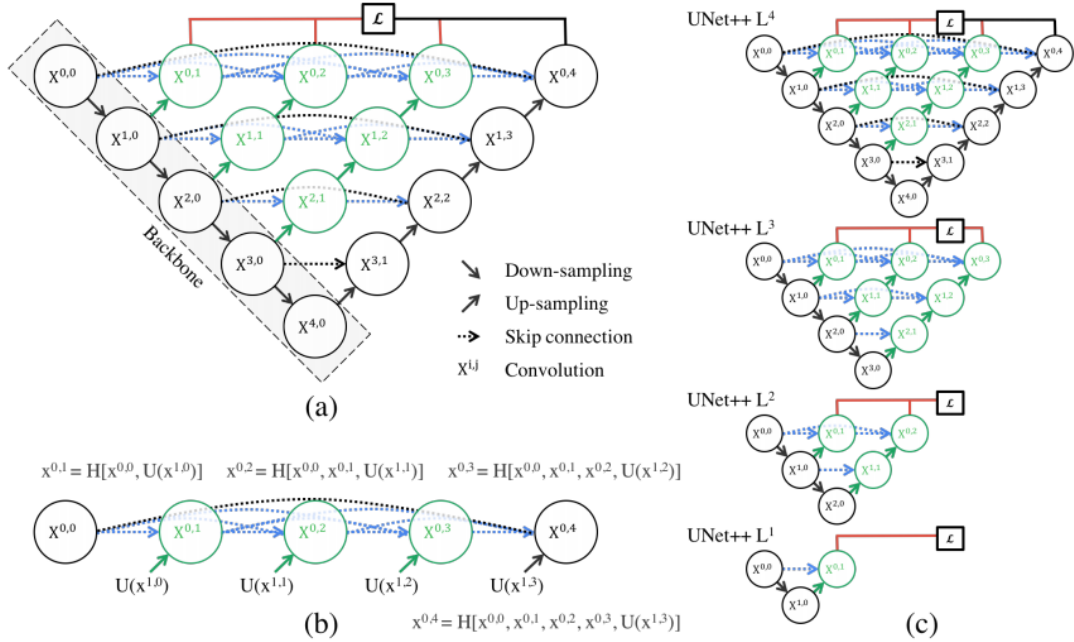


Fig. 1: (a) UNet++ consists of an encoder and decoder that are connected through a series of nested dense convolutional blocks. The main idea behind UNet++ is to bridge the semantic gap between the feature maps of the encoder and decoder prior to fusion. For example, the semantic gap between $(X^{0,0}, X^{1,3})$ is bridged using a dense convolution block with three convolution layers. In the graphical abstract, black indicates the original U-Net, green and blue show dense convolution blocks on the skip pathways, and red indicates deep supervision. Red, green, and blue components distinguish UNet++ from U-Net. (b) Detailed analysis of the first skip pathway of UNet++. (c) UNet++ can be pruned at inference time, if trained with deep supervision.

图 6 Unet++整体架构

3.3 训练测试与后处理

1.训练条件

本人采用阿里天池 NVIDIA Tesla V100 进行实验, torch.__version__=1.4.0, torchvision.__version__=0.5.0。训练集采用两个数据集的 train 集, 分别为 $20 \times 36 = 720$, $20 \times 9 = 180$ 张。训练只进行了 5 轮, 每轮计算平均损失, 损失函数为 $BCELoss = -1/n \sum (y_n \times \ln x_n + (1-y_n) \times \ln(1-x_n))$, BCEWithLogitsLoss 就是把 Sigmoid 和 BCELoss 合成一步在 torch.nn 中, 我们使用如下函数: `criterion = nn.BCEWithLogitsLoss()`。每轮训练损失数据见第四章结果部分。

2.测试过程

测试采用两个数据集的测试集分别为 $8 \times 36 = 288$, $20 \times 9 = 180$ 张。将预测的小块图像存为灰度图, 每个小块图计算预测图与标签图的 Dice 值, 最终计算平均 Dice 值。

3.后期合并与评价

后期将每个小块合并为整张图像, 由于前期分割存在问题, 所以我们不能至合并预测的小块, 还要合并分割之后的标签小块。将合并好的预测整张图与合并好的标签整张图进行 Dice 和 Accuracy 计算, 每张图计算 2 个指标值, 每个数据集计算两个指标的均值。

3.4 评价测度

1.Dice

计算两个数据 x, y 之间的重合程度:

$$Dice = 2 \times \frac{x \cap y}{x \cup y} \quad (3)$$

2.Accuracy

计算两个数据 x,y 之间的一致程度:

$$\text{Accuracy} = \frac{\sum(x[i] == y[i])}{\sum(x)} \text{ for } i \text{ in } \text{length}(x) \quad (4)$$

四、结果

经过彩图转灰度图, 图像归一化, 对比度限制自适应直方图均衡化, 伽马校正四步预处理的 CHASE、DRIVE 数据集图像如图 7 所示。由图可见, 对比度限制自适应直方图均衡化处理效果明显。

分割后的小块、分割后的标签、预测结果对比如图 8 所示。由图可见, 相比标签, 我们实验的预测结果在对细节部分, 如细小血管的检测还存在不足 (例如图 8 第 1 行第 3 张图)。但我们的实验有些图像显示出对人为没标出的类似血管的区域也有预测效果 (例如图 8 第 2 行第 6、9 张图), 可弥补人为的不足。

CHASE test, DRIVE test 数据集中原图、预处理后图、合并后的标签图、合并后的预测图对比如图 9,10 所示。由图可见预测图基本能与标签图重合, 但在一些细节处还存在不足。而且合并后的标签和预测图是比原图小的, 这是由于裁剪时向下取整没有包含完全原图, 在将来的工作中我们将进一步改进。

图 11 显示了运行训练测试代码时的打印日志内容。从中可见训练损失逐渐减小, Unet++5 轮损失更小, 但我们没有做控制变量对比实验, 这也将是下一步的工作。CHASE 的 test 的 288 个小块的 Average Dice Score: 73.4771%, DRIVE 的 test 的 180 个小块的 Average Dice Score: 75.1572%。图 12 显示了运行最终评估合成的图像的平均 Dice 和平均准确率。如图所示, CHASE avarage Dice:78.03%, avarage Accuracy:96.91%, DRIVE avarage Dice:77.45%, avarage Accuracy:95.83%。由此可见分割小块越多 (CHASE1 分 36, DRIVE1 分 9), 合成图后 Dice 增长会更加明显, 因此分割多图增加训练测试数据量也将是下一步的工作。

4.1 图像增强结果

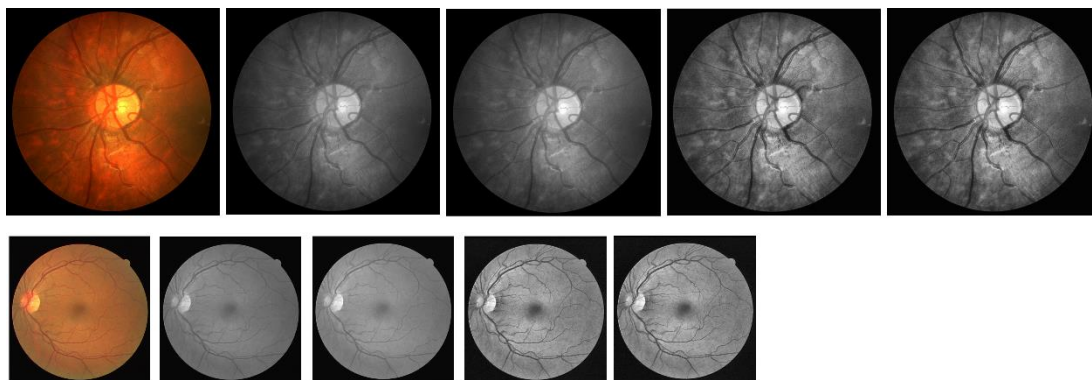


图 7 预处理每个步骤（彩图转灰度图，图像归一化，对比度限制自适应直方图均衡化，伽马校正）之后的效果图。第 1,2 行分别为 CHASE 和 DRIVE 数据集。

4.2 小块预测结果与标签对比

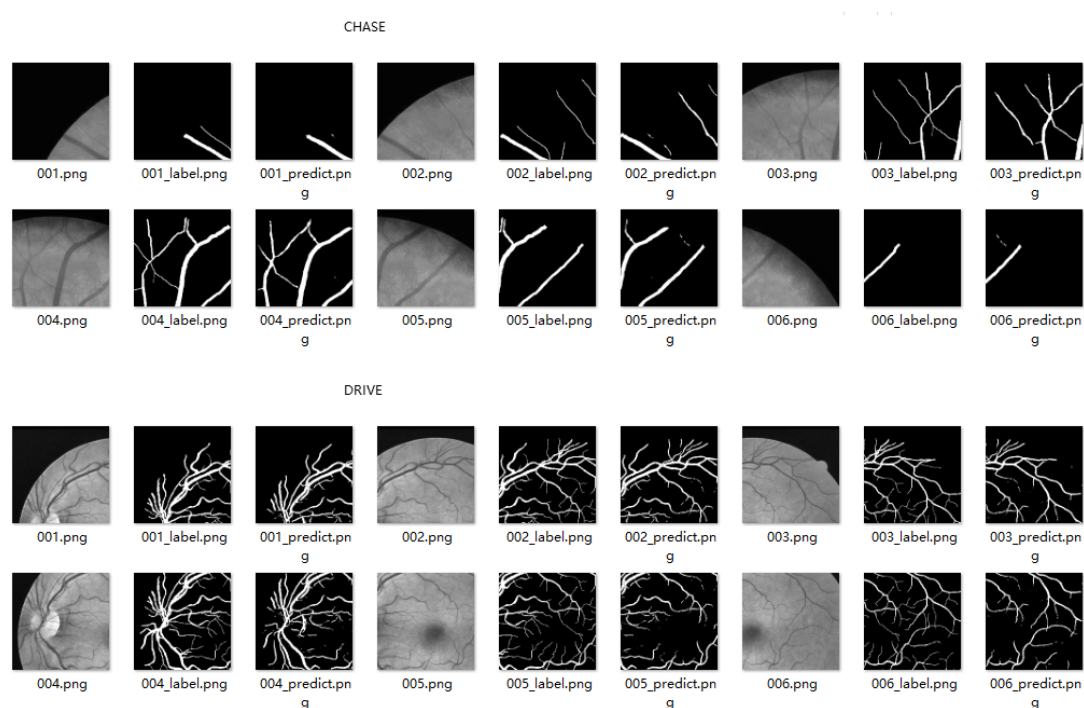


图 8 分割后的小块、标签图、预测图三者对比图。1,2 行为 CHASE 的 test 数据集；3,4 行为 DRIVE 的 test 数据集。每 3 张小图为 1 组对比。

4.3 合并成完整图预测结果与 label 对比

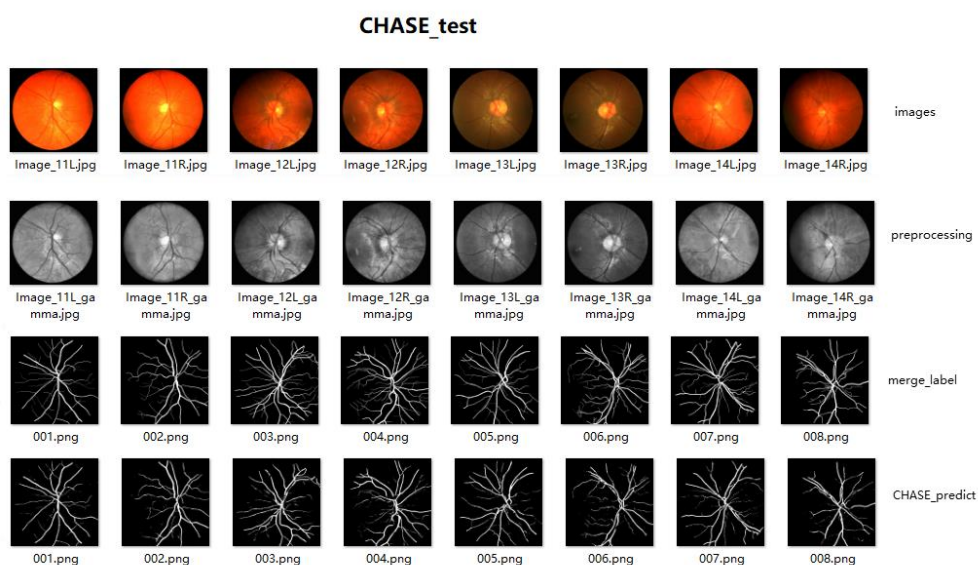


图 9 CHASEtest 数据集中原图、预处理后图、合并标签图、合并预测图对比。

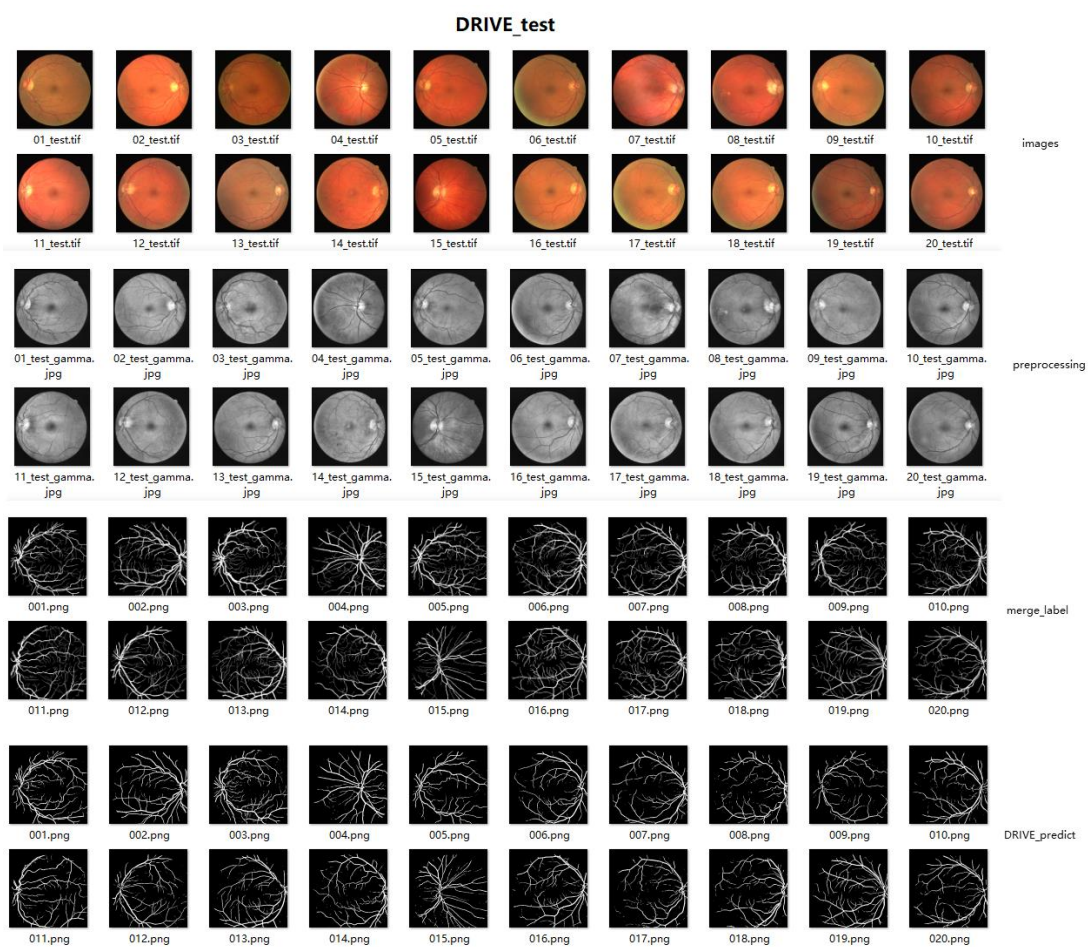


图 10 DRIVEtest 数据集中原图、预处理后图、合并标签图、合并预测图对比。

4.4 训练、测试及最后评估日志

<pre>DRIVE, Unet train_num=20*9=180, batch_size=8; test_num=20*9=180, batch_size=1 %run train_test_evaluate train --datasetname='DRIVE' Epoch 1/5 ----- 1/23,train_loss:0.736 ... 23/23,train_loss:0.319 epoch 1 loss:0.413 Epoch 2/5 ----- 1/23,train_loss:0.293 ... 23/23,train_loss:0.213 epoch 2 loss:0.253 Epoch 3/5 ----- 1/23,train_loss:0.217 ... 23/23,train_loss:0.174 epoch 3 loss:0.204 Epoch 4/5 ----- 1/23,train_loss:0.192 ... 23/23,train_loss:0.169 epoch 4 loss:0.174 Epoch 5/5 ----- 1/23,train_loss:0.156 ... 23/23,train_loss:0.134 epoch 5 loss:0.157 ----- %run train_test_evaluate test --datasetname='DRIVE' index:1 dice score:0.8297099055045265 ... index:180 dice score:0.8211850380668653 Average Dice Score: 75.1572%</pre>	<pre>CHASE, Unet++ train_num=20*36=720, batch_size=2; test_num=8*36=288, batch_size=1 %run train_test_evaluate train Epoch 1/5 ----- 1/360,train_loss:0.696 ... 360/360,train_loss:0.084 epoch 1 loss:0.188 Epoch 2/5 ----- 1/360,train_loss:0.224 ... 360/360,train_loss:0.200 epoch 2 loss:0.125 Epoch 3/5 ----- 1/360,train_loss:0.139 ... 360/360,train_loss:0.110 epoch 3 loss:0.110 Epoch 4/5 ----- 1/360,train_loss:0.098 ... 360/360,train_loss:0.090 epoch 4 loss:0.100 Epoch 5/5 ----- 1/360,train_loss:0.097 ... 360/360,train_loss:0.111 epoch 5 loss:0.093 ----- %run train_test_evaluate test index:1 dice score:0.820403825717322 ... index:288 dice score:0.7398648648648649 Average Dice Score: 73.4771%</pre>
---	--

图 11 训练测试代码运行日志


```
python finally_evaluate.py

CHASE No.1, Dice:82.06%, Accuracy:97.71%
CHASE No.2, Dice:81.09%, Accuracy:97.62%
CHASE No.3, Dice:76.20%, Accuracy:96.22%
CHASE No.4, Dice:78.20%, Accuracy:96.52%
CHASE No.5, Dice:77.55%, Accuracy:96.84%
CHASE No.6, Dice:73.54%, Accuracy:96.46%
CHASE No.7, Dice:79.55%, Accuracy:96.93%
CHASE No.8, Dice:76.02%, Accuracy:97.00%
CHASE avarage Dice:78.03%, avarage Accuracy:96.91%
DRIVE No.1, Dice:80.14%, Accuracy:95.85%
DRIVE No.2, Dice:82.00%, Accuracy:96.07%
DRIVE No.3, Dice:75.77%, Accuracy:94.57%
DRIVE No.4, Dice:79.41%, Accuracy:95.84%
DRIVE No.5, Dice:76.71%, Accuracy:95.51%
DRIVE No.6, Dice:74.82%, Accuracy:95.07%
DRIVE No.7, Dice:76.64%, Accuracy:95.71%
DRIVE No.8, Dice:72.05%, Accuracy:95.28%
DRIVE No.9, Dice:70.27%, Accuracy:95.35%
DRIVE No.10, Dice:77.66%, Accuracy:95.96%
DRIVE No.11, Dice:77.46%, Accuracy:95.76%
DRIVE No.12, Dice:76.08%, Accuracy:95.71%
DRIVE No.13, Dice:76.95%, Accuracy:95.40%
DRIVE No.14, Dice:78.46%, Accuracy:96.37%
DRIVE No.15, Dice:79.59%, Accuracy:96.64%
DRIVE No.16, Dice:80.35%, Accuracy:96.23%
DRIVE No.17, Dice:72.79%, Accuracy:95.44%
DRIVE No.18, Dice:77.54%, Accuracy:96.23%
DRIVE No.19, Dice:84.87%, Accuracy:97.00%
DRIVE No.20, Dice:79.51%, Accuracy:96.64%
DRIVE avarage Dice:77.45%, avarage Accuracy:95.83%
```

图 12 最后合并图评估代码运行日志

五、总结

我们 14 小组分为 4 人：吴文霞，肖安琪，王国帅，蒲尧。在该协同实验中，我们小组合作部分是：讨论整体流程，选择网络架构，理解论文，探讨如何 Debug。而复现代码这部分相互独立。以下是本人对自己在该实验中工作的总结。

5.1 本实验优点

- 1.预处理思路清晰，采用了常见易行的几种预处理方法。
- 2.自己亲用 torch 写的 Unet 和 Unet++网络，其结构清晰明了易懂。
- 3.结合图像增强和深度学习网络的整体结构，训练测试结果良好。甚至在训练数据量很少、训练轮次很低的情况下，比部分论文的效果都好，如表 1 所示。

表 1 部分论文方法 accuracy 对比

Methods	DRIVE	CHASE
DenseNet-U-Net[5]	96.74%	
DUNet[6]	95.66%	96.10%
Preprocess+Unet/Unet++ (Ours)	95.83%	96.91%

5.2 不足与改进

由于时间、运算资源有限，许多细节工作没有仔细推敲。

1.预处理阶段，较为高级的一些方法没有采用，例如：①高斯滤波去噪，②对比度增强，③坐标中心化、 x -shearing 归一化、缩放归一化和旋转归一化等图像归一化方法，④形态学操作：腐蚀、膨胀和开闭运算。

2.裁剪工作没将图像裁剪完整，这是由于实验时没注意向下取整而非向上取整而导致的。这直接影响到：包含原图所有信息不完整，后期的合并工作无法按照原图尺寸复原。

3.同一数据集没有进行不同网络训练测试，从而进行性能对比。

4.同一网络没有进行网络输出 Sigmoid 之后二值化的最佳阈值测试，本实验直接设置 0~1 中值 0.5。调节该参数，可能会获取到最佳分割阈值。

5.进行训练的数据量和轮次较少。由实验结果可知，分割的小块越多，最终合并的大图的平均 Dice 将会越大，因此，增加数据量，即切分更小的块进行训练测试是必要的。同时，我们看到，train loss 随着训练轮次的增加而大幅减小，所以增加训练轮次也是必须的。

6. mask 感兴趣区域没有使用。这部分可以将眼球以外部分的噪声去除，有效使用，效果应该会更好。

7.Unet 和 Unet++的编码器还可以多采用网上较为先进的最近提出的编码器，进行网络结构优化。甚至也可以换掉 Unet 基本架构。我们可以选择如表 2 所示的不同基本结构与不同编码器进行结合，应用较先进的方法强强联合，相信其性能会有较大提升。详情可参考如下代码库：

https://github.com/qubvel/segmentation_models.pytorch

表 2 部分网络框架及编码器列表

Architectures	Encoders
Unet	ResNet
Unet++	ResNeXt
MAnet	ResNeSt
Linknet	Res2Ne(X)t
FPN	RegNet(x/y)
PSPNet	SE-Net
PAN	SK-ResNe(X)t
DeepLabV3	DenseNet
DeepLabV3+	Inception
	EfficientNet
	MobileNet
	DPN
	VGG

参考文献

- [1] Pizer S M, Amburn E P, Austin J D, et al. Adaptive histogram equalization and its variations[J]. Computer vision, graphics, and image processing, 1987, 39(3): 355-368.
- [2] Zuiderveld K . Contrast Limited Adaptive Histogram Equalization[J]. Graphics Gems, 1994:474-485.
- [3] Srinivasan S, Balram N. Adaptive contrast enhancement using local region stretching[C]//Proceedings of the 9th Asian symposium on information display. 2006: 152-155.
- [4] 马青柯. 眼底图像检测和分析系统的设计与开发[D]. 暨南大学.
- [5] 梁礼明, 盛校棋, 郭凯,等. 基于改进的 U-Net 眼底视网膜血管分割[J]. 计算机应用研究, 2020, v.37;No.342(04):293-297.
- [6] Jin Q , Meng Z , Pham T D , et al. DUNet: A deformable network for retinal vessel segmentation[J]. Knowledge-Based Systems, 2019, 178(AUG.15):149-162. <https://github.com/RanSuLab/DUNet-retinal-vessel-detection>
- [7] J.V. Soares, J.J. Leandro, R.M. Cesar, H.F. Jelinek, M.J. Cree, Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification, IEEE Trans. Med. Imaging 25 (9) (2006) 1214–1222.
- [8] Ronneberger O , Fischer P , Brox T . U-Net: Convolutional Networks for Biomedical Image Segmentation[M]. Springer, Cham, 2015.
- [9] Zhou Z , Siddiquee M , Tajbakhsh N , et al. UNet++: A Nested U-Net Architecture for Medical Image Segmentation[C]// 4th Deep Learning in Medical Image Analysis (DLMIA) Workshop. 2018.
- [10] Yu F , Wang D , Shelhamer E , et al. Deep Layer Aggregation[C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2018.