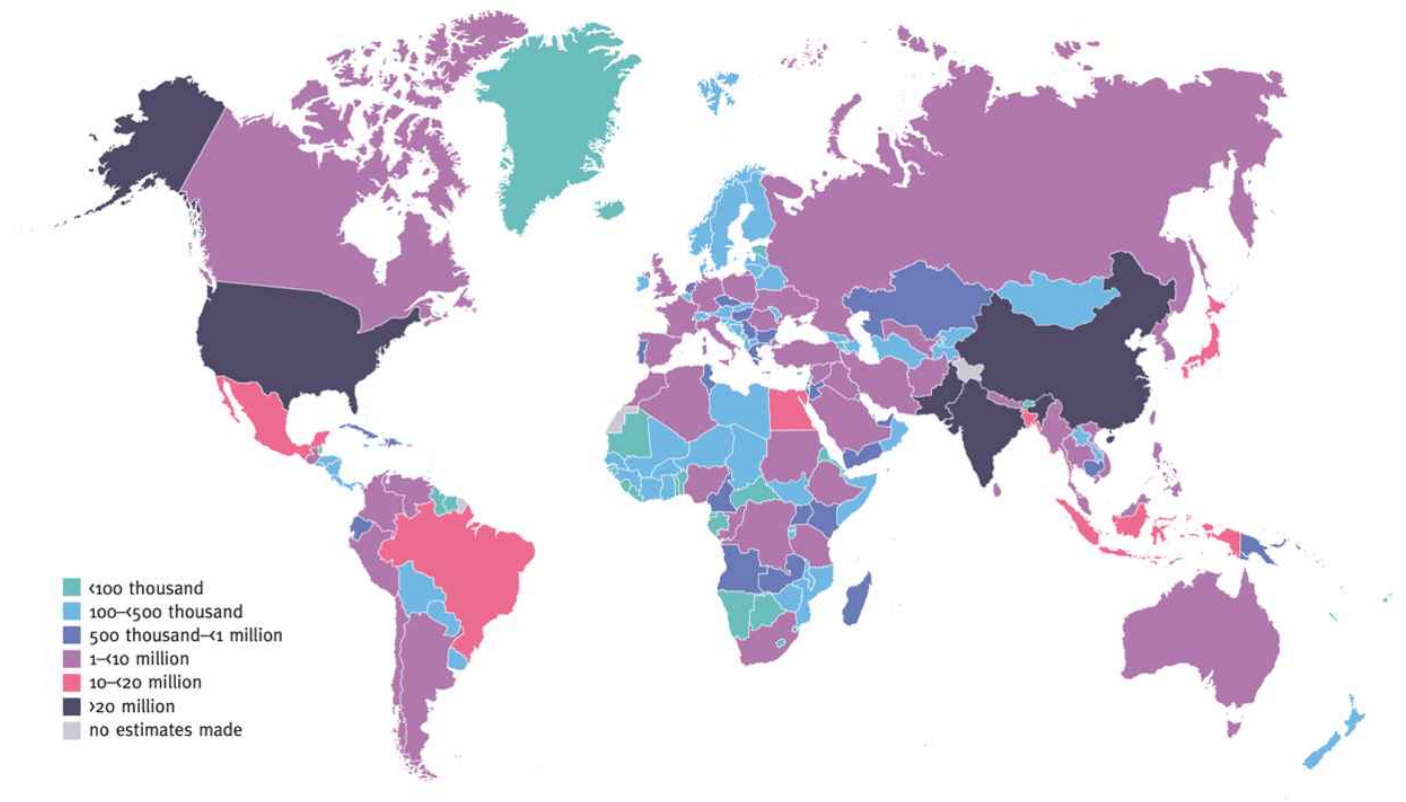


糖尿病性视网膜病变项目报告

一、项目背景

Map 3.1 Estimated total number of adults (20–79 years) with diabetes in 2021



近年来, 全球范围内糖尿病患者的数量逐年增加, 经查阅IDF糖尿病地图谱, 得知在全球范围已经有5亿人口被诊断出患有此病, 且预计到2045年, 这个数字将达到7亿之多。由此看来, 糖尿病已经成为一个全球性的健康问题。并且IDF糖尿病地图谱中还指出, 2040年约有1/3的糖尿病患者将会有患有糖尿病视网膜病变(diabetic retinopathy, DR), 同时DR是导致我国人口失明的主要原因。

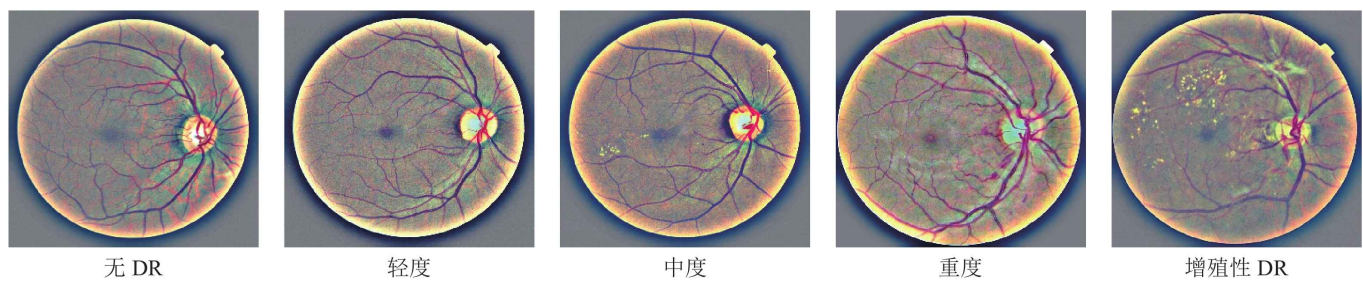
二、DR分级

1. 基本介绍

DR是一种视网膜血管受损导致的眼部病变, 视网膜血管会表现为出血、软硬渗出物、微动脉瘤等症状, 甚至可能导致视网膜血管的异常生长。依据DR病变严重程度可以将DR分为5个阶段, 分别为无病变(NO-DR); 轻度非增殖性病变(Mild NPDR); 中度非增殖性病变(Moderate NPDR); 重度非增殖性病变(Severe NPDR); 增殖性病变(PDR)。

这5个不同病变等级有着不同的临床表现（如下图所示），临床医生可以通过检查患者的眼底图像并结合DR分级标准（如下表所示）来评估DR的严重程度。然而这种传统的人工分级将耗费大量时间,

且就DR患者的数量而言, 目前临床经验十足的医生还是相对缺乏的。然而DR的分级对于防止DR患者视力失明尤为重要, 随着深度学习在临床诊断中的广泛应用, 深度学习为DR的分级提供了更为高效且可靠的诊断手段。



类别 标签	DR病变等级	临床表现
0	NO-DR	无病变特征
1	Mild NPDR	仅出现微动脉瘤
2	Moderate NPDR	除了微动脉瘤外, 出现血点絮状静脉串珠, 病变程度介于轻度和重度之间
3	Severe NPDR	大于两个象限出现静脉串珠, 四象限内, 每个象限出现20个以上的出血点, 至少一个象限出现微血管异常
4	PDR	玻璃体/视网膜出血, 增生新血管

2. 实验流程

2.1 数据集

DR分级常用的数据集有APTOS 2019、EyePACS、IDRiD、MESSIDOR-2、DDR等, 这里主要使用了IDRiD数据集。IDRiD来源于IEEE国际生物医学影像研讨会(ISBI-2018)举办的“糖尿病性视网膜病变: 分割和分级挑战”竞赛, 包括597张彩色眼底图像组成, 像素值为4288×2848, 存储格式为JPEG。此数据集由3部分组成: 第1部分是语义标注的81张眼底图像; 第2部分是516张通过CSV文件记录的病变等级; 第3部分通过CSV文件记录了每张图像中视盘和中间凹的位置信息。

2.2 数据增强与预处理

预处理: 由于训练能力有限, 将图像大小缩小到512*512; 然后通过计算亮度对图像进行缩放, 统一所有图像的尺度。

数据增强: 使用翻转、旋转、随机裁剪、颜色抖动等对图像进行随机的修改。

数据增强相关代码如下:

```
1 scale = transforms.Resize((512,512))
2 to_tensor = transforms.ToTensor()
3 horizontal_flip = transforms.RandomHorizontalFlip(p=0.5)
```

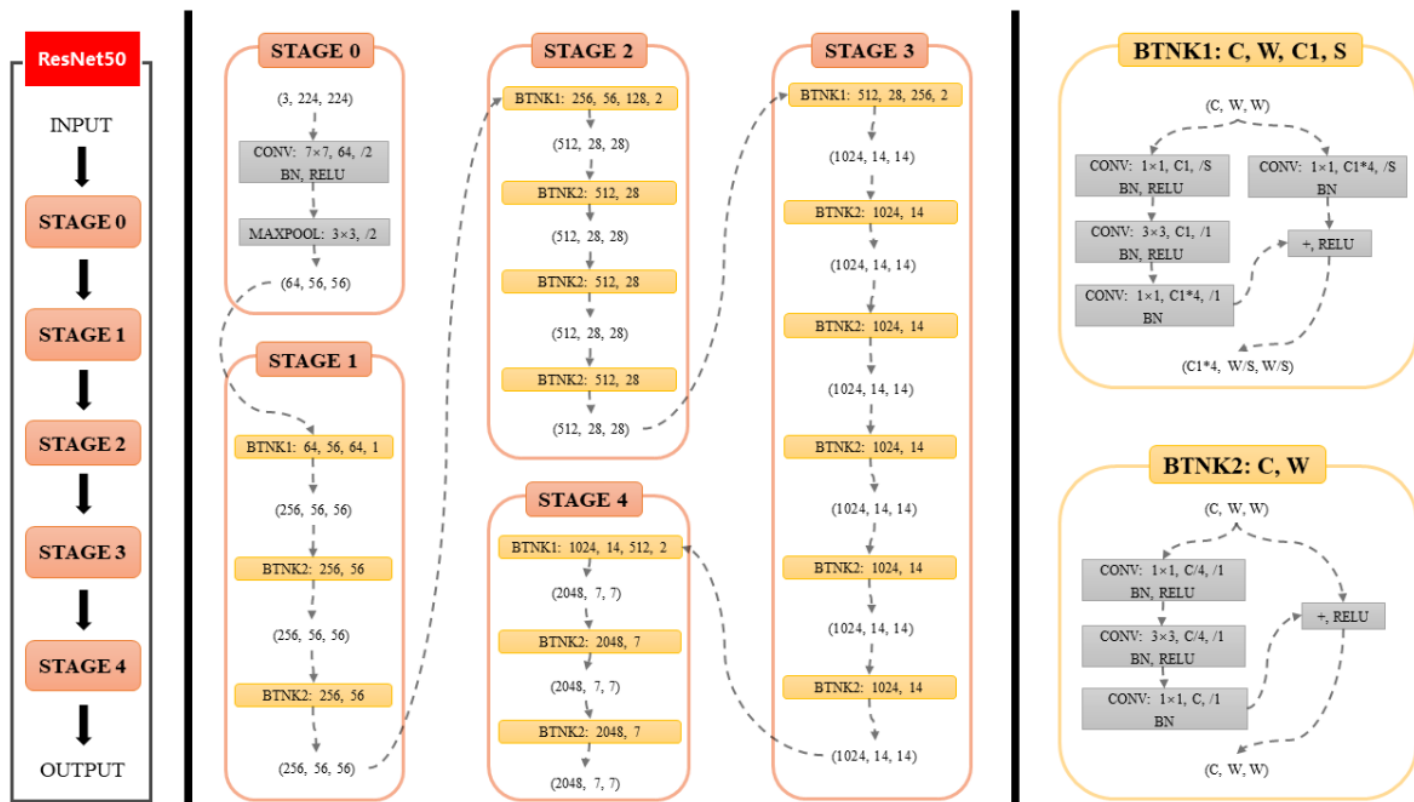
```

4 vertical_flip = transforms.RandomVerticalFlip(p=0.3)
5 color_jitter = transforms.ColorJitter(brightness=0.01 *
    torch.abs(torch.randn(1)).item(),
6                                     contrast=0.01 *
    torch.abs(torch.randn(1)).item(),
7                                     saturation=torch.abs(0.01 *
    torch.randn(1)).item(),
8                                     hue=torch.abs(0.01 *
    torch.randn(1)).item()
9                                     )
10 random_rotation = transforms.RandomRotation(30)
11 center_crop = transforms.CenterCrop(512)
12 composed = transforms.Compose([scale,
13                               horizontal_flip,
14                               random_rotation,
15                               vertical_flip,
16                               color_jitter,
17                               center_crop,
18                               to_tensor]
19                               )

```

2.3 模型选择

使用ResNet50作为基础网络架构，结合迁移学习的策略，在ImageNet数据集上的训练权重来初始化模型权重。ResNet提出“残差连接”的思想来解决退化现象，将浅层特征信息直接连接到深层，利用新的表达方式实现层间的恒等映射。ResNet的每层特征仅需要在前续层特征的基础上学习新的特征，残差连接中反向传播的导数需要加上恒等项，这样缓解了网络退化和梯度弥散问题，使深层网络具有更好的逐层学习能力和更强的模型表达能力。ResNet50网络的整体结构如下：

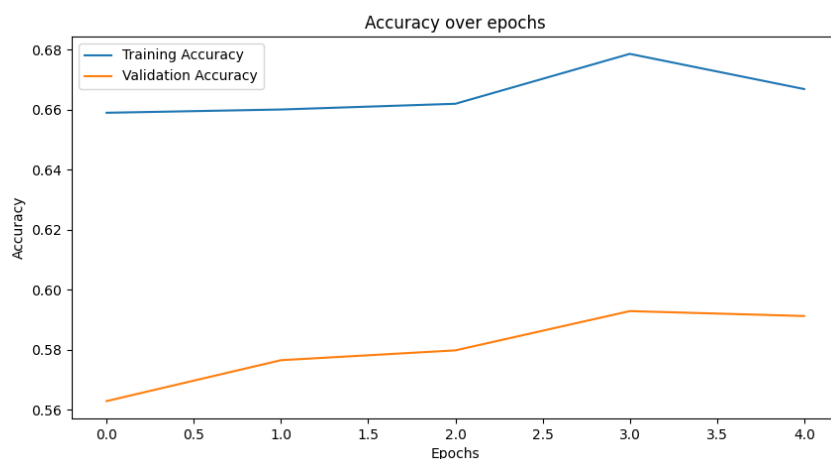


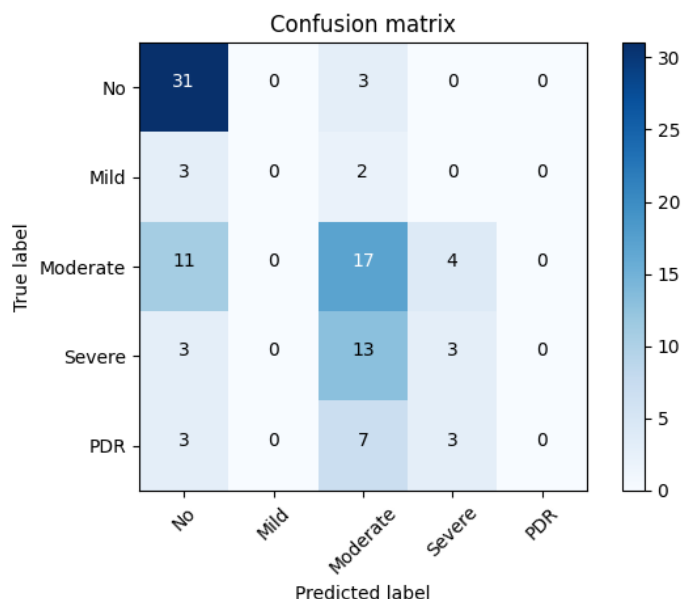
2.4 损失函数、评估指标、优化器

采用交叉熵损失函数，采用准确率、AUC-ROC、混淆矩阵三种评估指标，并采用Adam优化器，它结合了多种优化算法的优点，是较为常用的优化器。

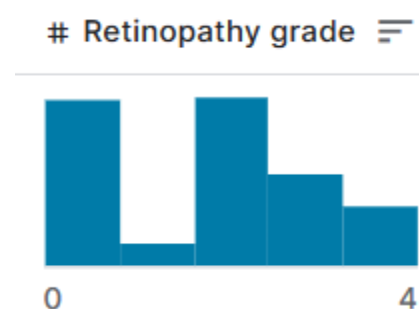
3. 实验结果与分析

对于准确率，在训练集上准确率最高达到了67.9%，在验证集上准确率最高达到了60.1%。对于最佳模型的AUC和混淆矩阵，AUC评分为0.689，混淆矩阵大多数错误分类错误在于容易把1类（轻度）和2类（中度）识别为0类（无病变），把3类（重度）和4类（增殖性病变）识别为2类（中度）。





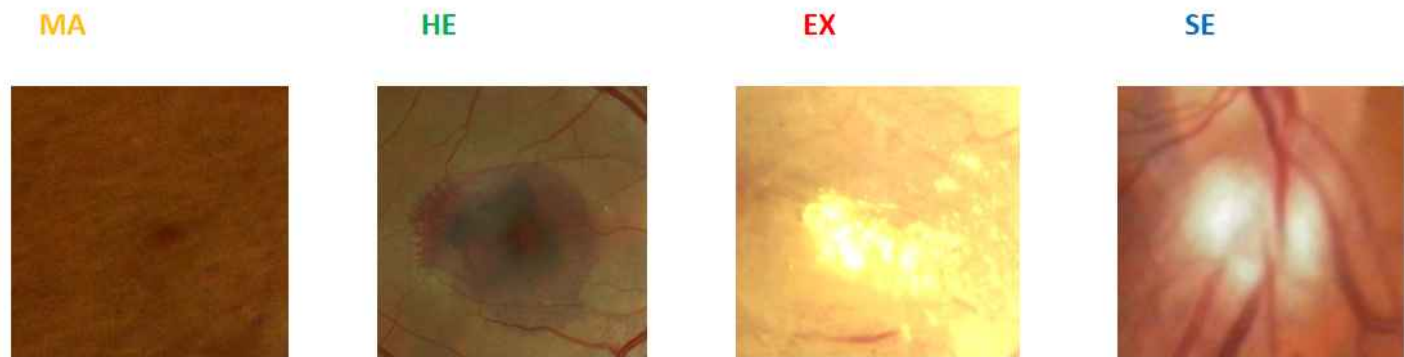
对实验结果进行分析，DR分级模型有一定分级效果但泛化能力还有很大提升空间。模型性能很大程度上受制于数据集, 如IDRiD数据集样本数较少且类别十分不均衡（0类和2类样本远远多于其它类）。而当前的DR数据集大多存在图像标签模糊、图像尺寸多规格、各类别间的不平衡等问题（由于来源不同、采集环境和采集设备不同、标记医生的专业能力不同），限制了DR分级模型的发展。同时因为深度学习的“黑箱”性质，导致DR分级模型的可解释性差，DR分级模型很难在临床应用中得到广泛推广。



三、DR分割

1. 基本介绍

近年来，以深度学习方法为基础的系统在DR疾病检测中取得显著成果，尤其是卷积神经网络及其相关算法的改进。传统的DR筛查系统通常首先识别感兴趣的区域进行特征提取，然后再使用提取的特征进行分类。但这种深度学习系统在DR筛查中缺乏可解释性，这导致了语义分割的研究，该研究将像素分为各种视网膜特征，微动脉瘤（MA）、出血（HE）、硬渗出物（EX）、软渗出物（SE）等。DR分割技术有望成为辅助医生进行快速、准确诊断的重要工具，从而提高DR的诊断效率和患者治疗的成功率。



我们观察发现，MA病变和HE病变比较相像，均为暗色；EX病变和SE病变比较相像，均为亮色。

2. 实验流程

2.1 数据集

DR分割常用的数据集有DDR、IDRiD等。DDR是一个广泛使用的糖尿病视网膜病变数据集，它包含了大量的视网膜眼底图像，这些图像涵盖了从无病变到严重病变的各种情况。DDR数据集的一个显著特点是它包含了详细的临床信息和专家的分级标注，这使得研究者可以在不同的DR分级上进行训练和测试。该数据集的图像通常具有高分辨率，图像质量优良，适合进行细致的图像分割任务。DDR数据集也常用于评估分割算法的鲁棒性和准确性，因为它包含了不同程度病变的图像，这对于测试算法在现实世界条件下的表现非常有用。IDRiD数据集在上面的DR分级部分已经介绍过。而本项目除了DDR、IDRiD两个数据集外还使用了同济大学私有的TJDR数据集，并将三个数据集用于语义分割的部分合并为一个更大的数据集。三个数据集用于语义分割的部分的图片数量如下：

	训练集	验证集	测试集	总计
TJDR	360	88	113	561
IDRiD	40	14	27	81
DDR	383	149	225	757
总计	1183	251	365	1399

2.2 数据增强与预处理

预处理：由于训练能力有限，将图像大小缩小到960*960；用预训练模型所用的Imagenet数据集的系数对图像进行归一化。

数据增强：使用alumentation库中的翻转、旋转、随机裁剪、高斯噪声、锐化、模糊、动态模糊等功能对图像进行随机的修改，还使用限制对比度自适应直方图均衡化（CLAHE）算法。

普通的自适应直方图均衡（AHE）算法往往会放大图像近恒定区域中的对比度，因为此类区域中的直方图高度集中。结果，AHE可能导致噪声在近恒定区域中被放大。CLAHE是AHE的一种变体，其中对比度放大受到限制，从而减少了这种噪声放大问题。

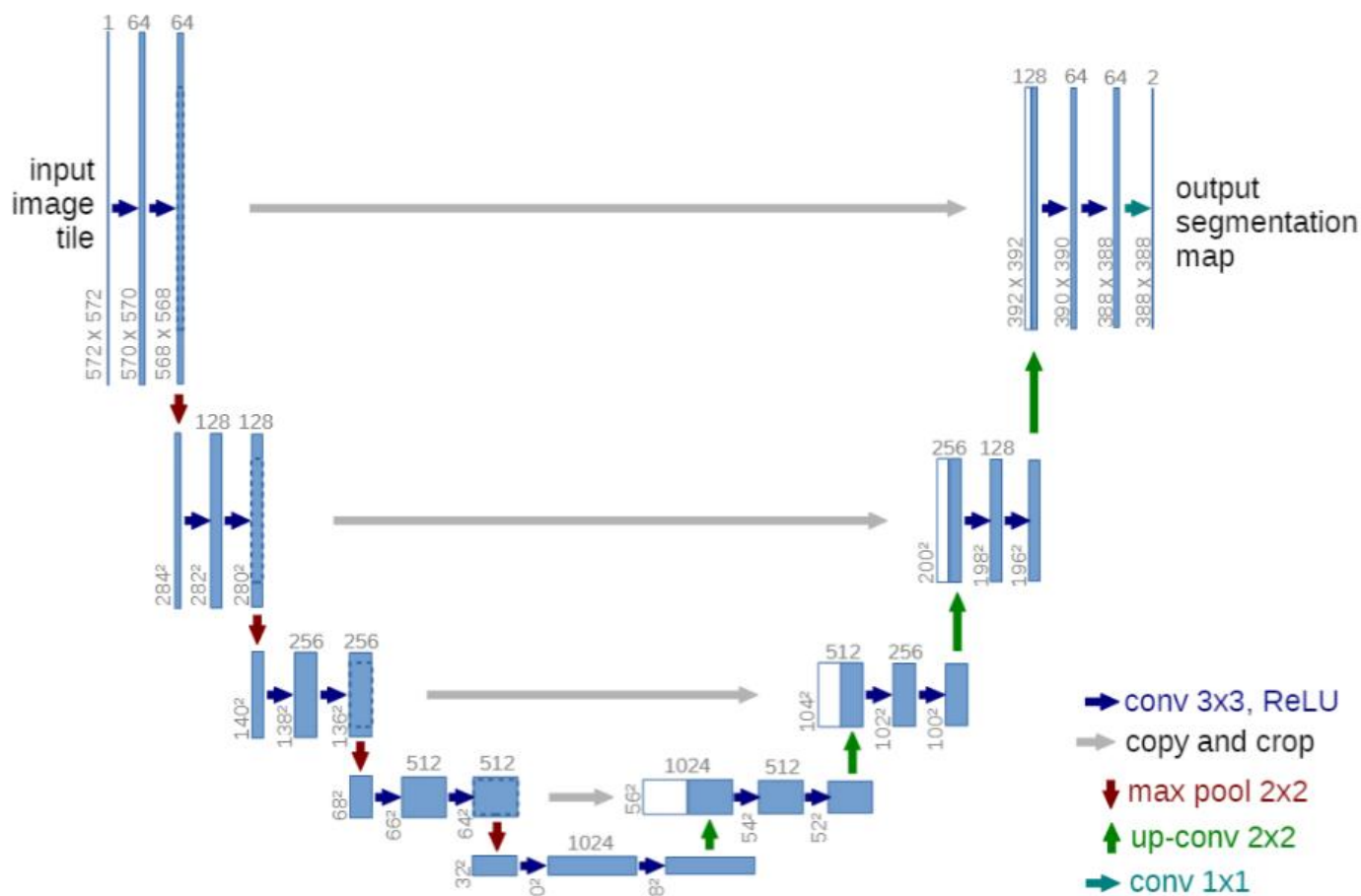
数据增强相关代码如下：

```
1 train_transform = [  
2     albu.Resize(height=480, width=480, always_apply=True),  
3     albu.HorizontalFlip(p=0.5),  
4     albu.ShiftScaleRotate(scale_limit=0.05, rotate_limit=45, shift_limit=0.05,  
5         p=1, border_mode=0),  
6     albu.PadIfNeeded(min_height=480, min_width=480, always_apply=True,  
7         border_mode=0),  
8     albu.RandomCrop(height=480, width=480, always_apply=True),  
9     albu.GaussNoise(p=0.2),  
10    albu.OneOf(  
11        [  
12            albu.CLAHE(p=1),  
13        ],  
14        p=1,  
15    ),  
16    albu.OneOf(  
17        [  
18            albu.Sharpen(p=1),  
19            albu.Blur(blur_limit=3, p=1),  
20            albu.MotionBlur(blur_limit=3, p=1),  
21        ],  
22        p=0.9,  
23    )]
```

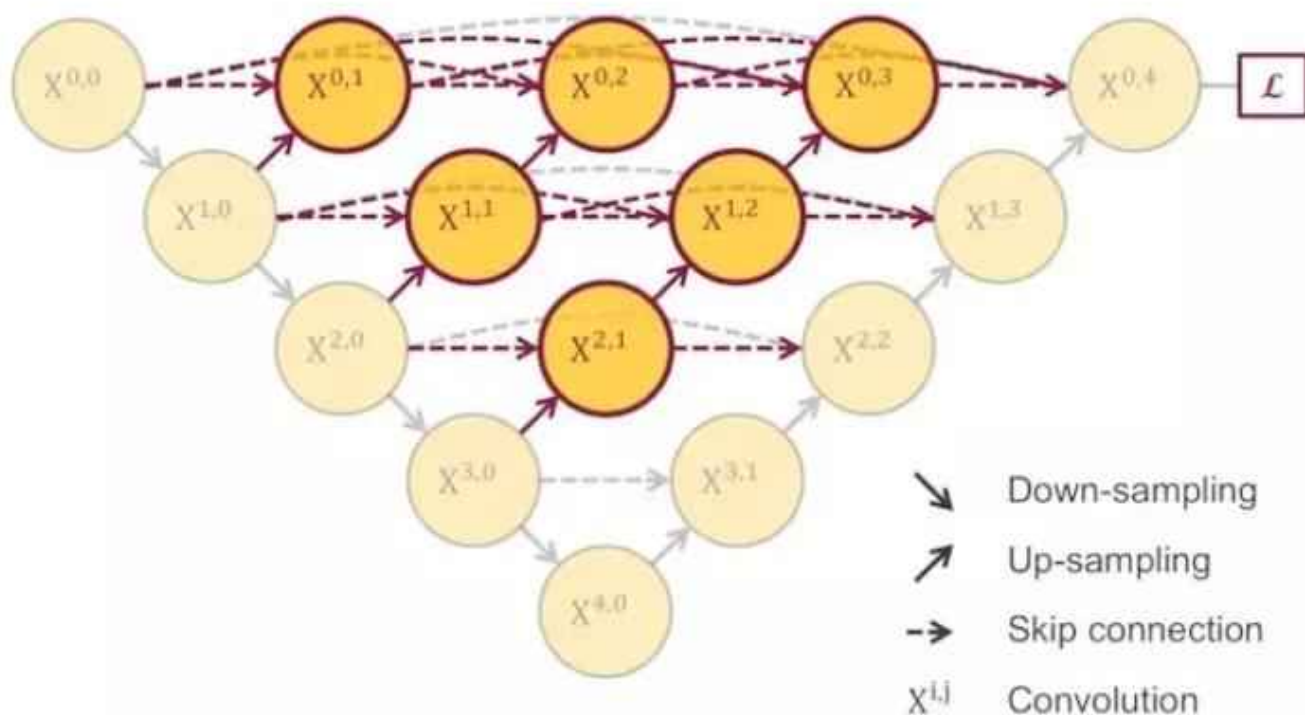
2.3 模型选择

使用Unet、Unet++、MAnet三种用于语义分割的网络，而编码器部分均选择在Imagenet数据集上预训练过的SE-ResNet50架构。

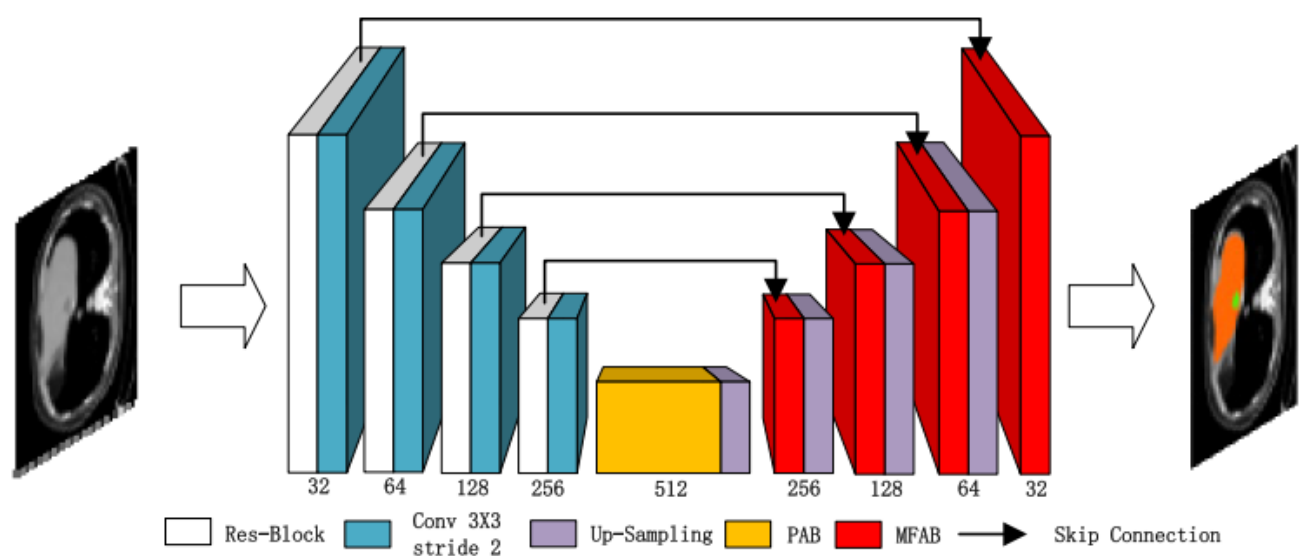
- **U-Net:** U-Net是一种流行的卷积神经网络（CNN），专为医学图像分割设计。它具有一个对称的U形结构，包括一个收缩（编码）路径和一个对称的扩展（解码）路径。这种结构使得网络能够捕获图像中的上下文信息并精确定位分割边界，特别适合于需要精细分割的医学图像。U-Net网络的整体结构如下：



- **U-Net++:** U-Net++是U-Net的扩展，它引入了更深的网络结构和更丰富的特征融合方式。U-Net++通过嵌入更多的跳跃连接和特征融合点，增强了模型对不同尺度特征的捕获能力，从而提高了分割精度和鲁棒性。U-Net++网络的整体结构如下：



- **MA-Net:** MA-Net（多尺度注意力网络）是一种结合了多尺度处理和注意力机制的网络。它通过在不同分辨率上分析图像特征，并使用注意力机制来增强重要特征的表示，从而提高了分割的准确性。MA-Net的设计允许它在处理具有复杂结构的医学图像时，更好地理解图像内容并进行精确分割。



- **SE-ResNet50:** SE-ResNet50通过引入SE模块，在几乎不增加计算成本的情况下，提高了模型对特征的表达能力，从而在多种视觉任务中取得了更好的性能。SE模块通过全局平均池化（Global Average Pooling, GAP）压缩特征图到一个单一的通道，然后使用一个全连接层来学习通道之间的相互关系。之后，通过ReLU激活函数和Sigmoid函数重新加权原始特征通道，增强了有用特征并抑制了不重要的特征。

2.4 损失函数、评估指标、优化器

采用Dice loss作为损失函数，相比于交叉熵损失函数，该损失函数更适用于前景像素远少于背景像素的情况。采用Dice, Iou两种评估指标（Dice等价于F1 score），相比于Accuracy，这两种指标更适用于前景像素远少于背景像素的情况。采用Adam优化器，它结合了多种优化算法的优点，是较为常用的优化器。

3. 实验结果与分析

我们本学期的思路如下：

一开始只在TJDR数据集上对四个类别分别进行单分类训练，发现MA、SE病变效果很差，推测数据量过少

↓ 将TJDR与DDR、idrid数据集合并

在合并后的数据集上对四个类别分别进行单分类训练，MA、SE病变训练效果有所提升，但对一张图片进行分割时需要依次运行4个病变对应的模型，花费时间过长

↓ 改为多分类训练

改为多分类训练后，由于MA病变和HE病变比较相像，EX病变和SE病变比较相像，由于病变数据量不平衡，模型倾向于将MA病变识别为HE病变，将SE病变识别为EX病变

↓ 调节Dice Loss函数各通道权值

将MA病变和SE病变所对应的权值调大后，模型能够更好地区分特征相近的病变

1. 单分类语义分割的训练结果：

我们在网上找到了一篇基于IDRiD数据集的论文作为基线方法，一开始我们使用的是单分类语义分割方法，对四个类别进行单独的二分类训练。基线方法和我们项目所用的方法的效果对比如下：

指标	Dice			
种类	EX	SE	HE	MA
基线代码 (Unet)+TJDR	0.3503	0.2938	0.6027	0.2213
本项目代码 (Unet)+TJDR	0.5711	0.5898	0.5613	0.389

我们的方法运用了更多的数据增强手段，选用SE_ResNet50作为编码器部分，并使用了预训练参数，可以看到，在网络结构、数据集一致，均为二分类的前提下，我们的方法对HE病变的识别效果略差于基线方法，而对其余3种病变的识别效果均明显优于基线方法。

基线方法论文代码地址：https://github.com/khalilouardini/IDRiD-challenge/tree/master/Segmentation_task

在这之后，我们把训练方法从四次单分类改成了一次多分类，并基于多分类的特点对掩码格式、激活函数、Dice Loss函数进行了改进，具体的对比如下：

训练方式	四次单分类	一次多分类
掩码格式	1*960*960，对于所选类别，1为真，0为假	5*960*960，对于四种标签类别和无标签类别进行独热编码
激活函数	sigmoid函数，预测所选类别为真的概率	softmax2d函数，四种标签类别的概率和为1（而sigmoid函数会分别预测某一类别为真的概率，导致类别与类别之间不相关）
Dice Loss函数	普通的Dice Loss	对于每个标签类别加权的Dice Loss（具体是对不同病变的predicted result和ground truth加权）

在修改Dice Loss函数时，我们发现，将哪个类别对应的权值增大，模型就会偏向于预测出该类别，而不是那个与它特征相近的类别。同时，对该类别的预测精度也会提升。通过肉眼观察，我们得到的较为合适的一组权重是：

病变种类	EX	SE	HE	MA
权重	5/12	2/3	3/5	3/2

修改后的Dice Loss函数相关代码如下：

```
1 def f_score(pr, gt, beta=1, eps=1e-7, threshold=None, ignore_channels=None):
2     pr = _threshold(pr, threshold=threshold)
3     pr, gt = _take_channels(pr, gt, ignore_channels=ignore_channels)
4
5     # 添加权重项的地方
6     weights = torch.tensor([5/12, 2/3, 3/5, 3/2]).to(pr.device)
7     weighted_pr = pr * weights.view(1, -1, 1, 1)
8     weighted_gt = gt * weights.view(1, -1, 1, 1)
9
10    tp = torch.sum(weighted_gt * weighted_pr)
11    fp = torch.sum(weighted_pr) - tp
12    fn = torch.sum(weighted_gt) - tp
13
14    score = ((1 + beta**2) * tp + eps) / ((1 + beta**2) * tp + beta**2 * fn +
fp + eps)
15
16    return score
17
18 class DiceLoss(Loss):
19     def __init__(
20         self, eps=1.0, beta=1.0, activation=None, ignore_channels=None,
21         **kwargs #在此处设置要忽略的类??
22     ):
23         super().__init__(**kwargs)
```

```

23     self.eps = eps
24     self.beta = beta
25     self.activation = Activation(activation)
26     self.ignore_channels = ignore_channels
27
28     def forward(self, y_pr, y_gt):
29         y_pr = self.activation(y_pr)
30         return 1 - f_score(
31             y_pr,
32             y_gt,
33             beta=self.beta,
34             eps=self.eps,
35             threshold=None,
36             ignore_channels=self.ignore_channels,
37         )

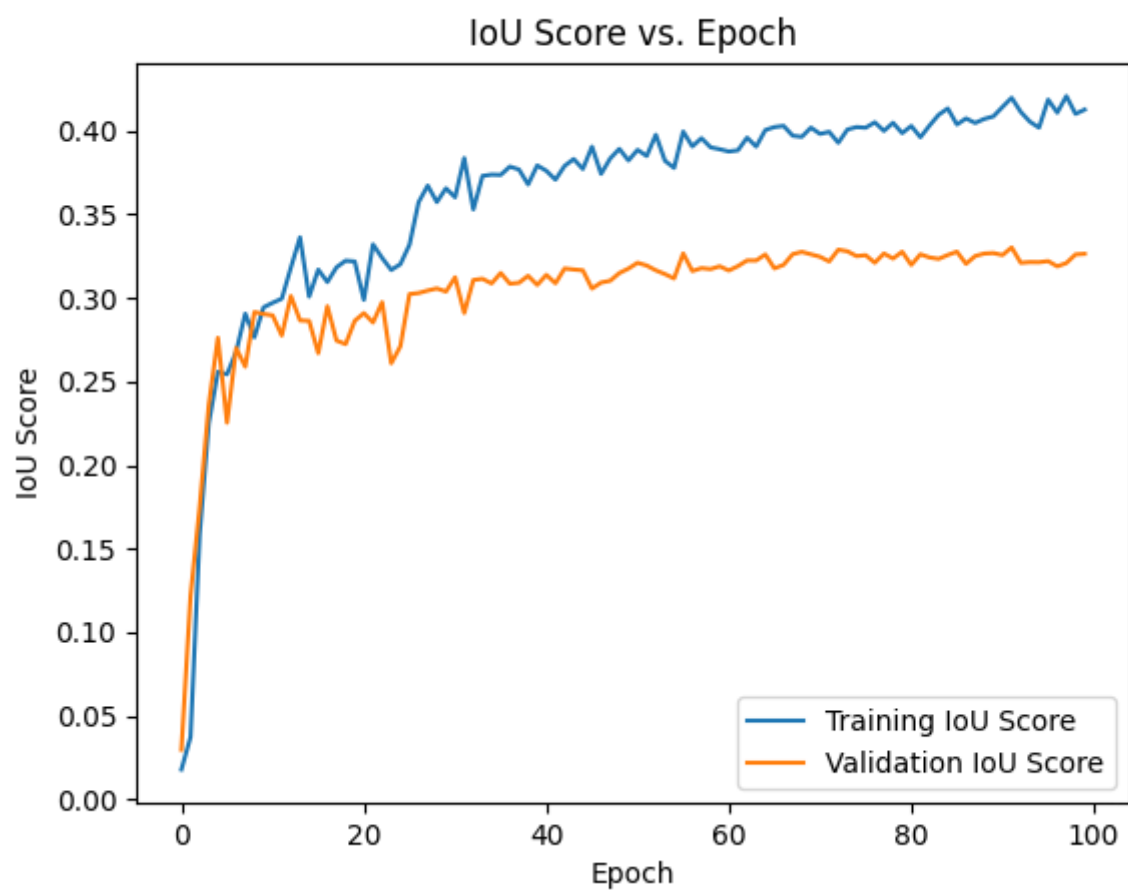
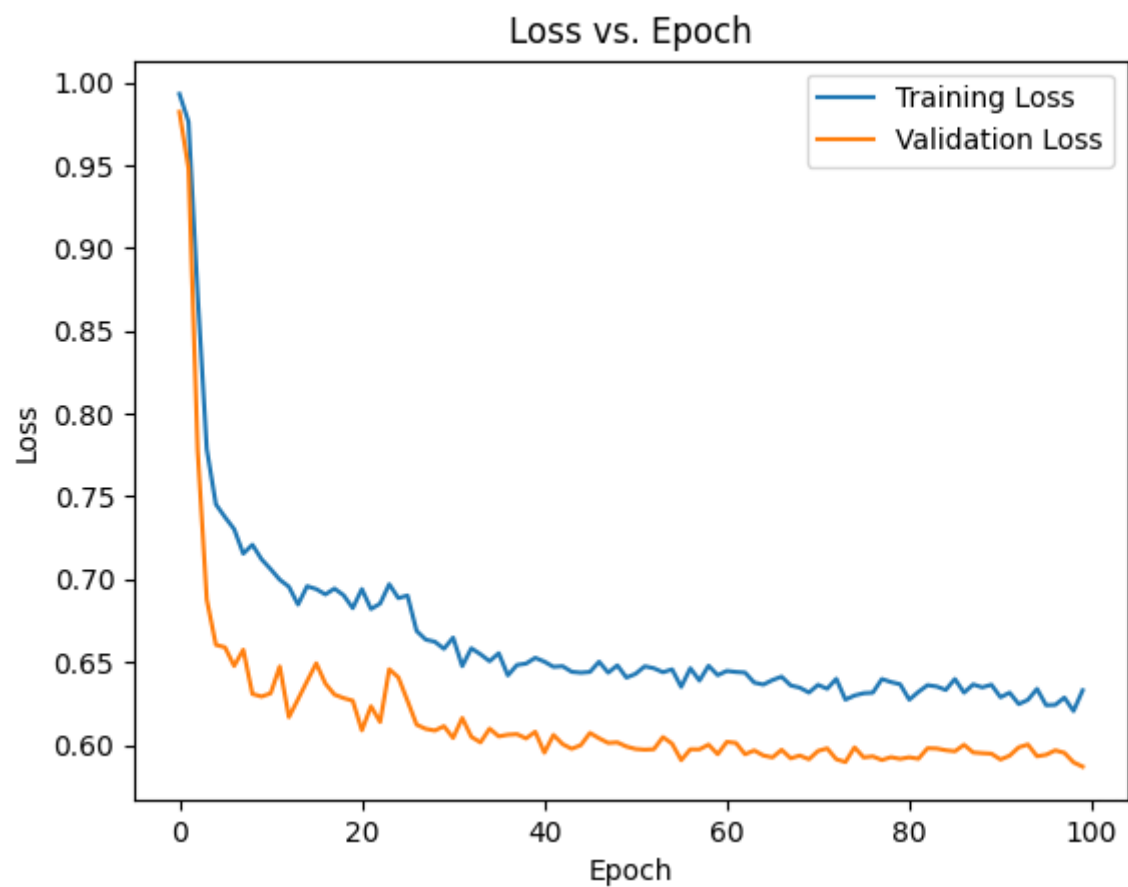
```

2. 多分类语义分割的训练结果：

我们通过调整Dice Loss对于不同通道的权重，保证了各病变的训练效果尽量均衡。由下表可以看出，相比于单分类语义分割，多分类语义分割的平均效果并未明显变差，而识别速度却大大提高。效果最突出的是Unet++对SE病变的识别，Dice值达到了0.7317。

指标	网络	EX	SE	HE	MA
Dice	Unet	0.4958	0.6112	0.5104	0.3746
	Unet++	0.4968	0.7317	0.5068	0.3639
	MAnet	0.4724	0.622	0.4889	0.3621
Iou	Unet	0.4187	0.6373	0.4271	0.3172
	Unet++	0.4096	0.7342	0.4262	0.309
	MAnet	0.3883	0.6808	0.4116	0.3017

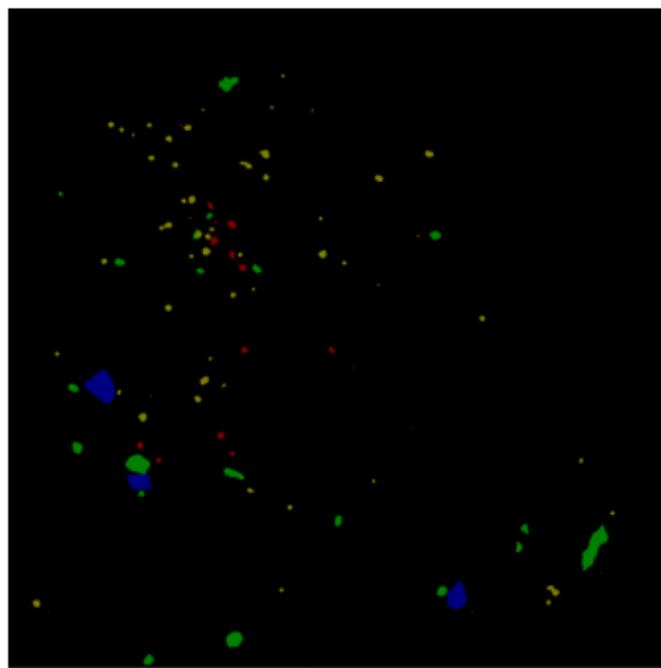
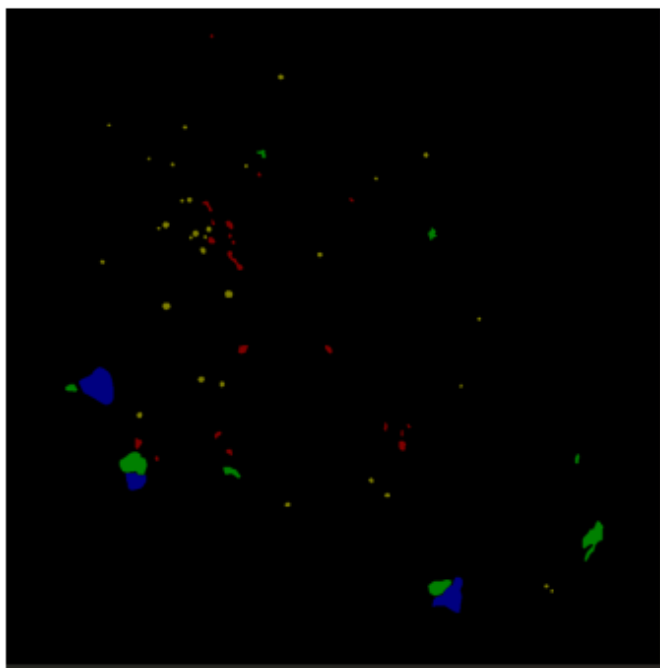
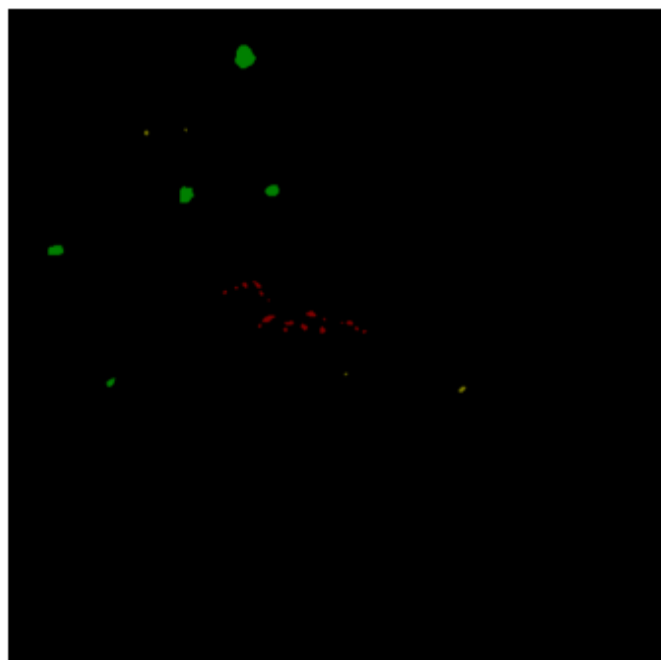
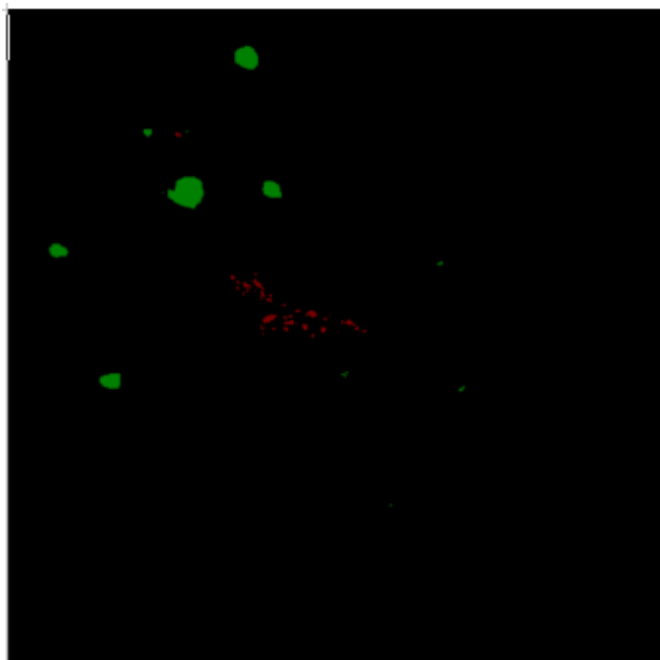
以下是Unet++的加权Dice Loss曲线和Iou曲线，可以看出，曲线基本达到收敛：

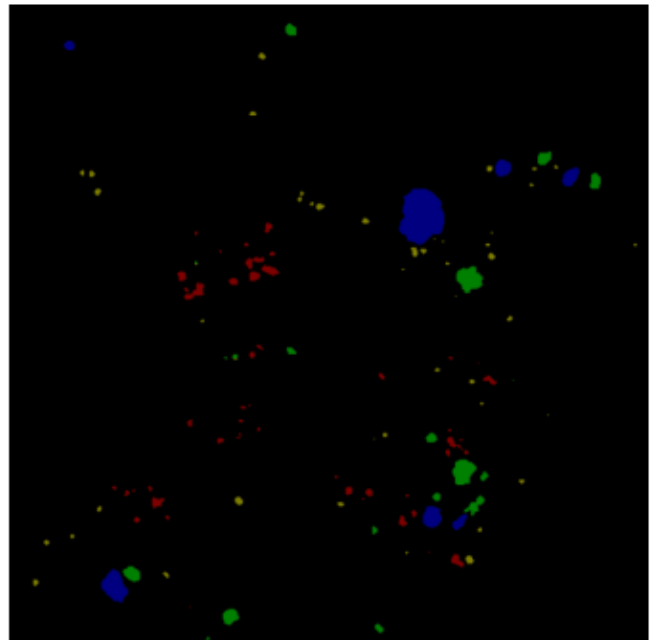
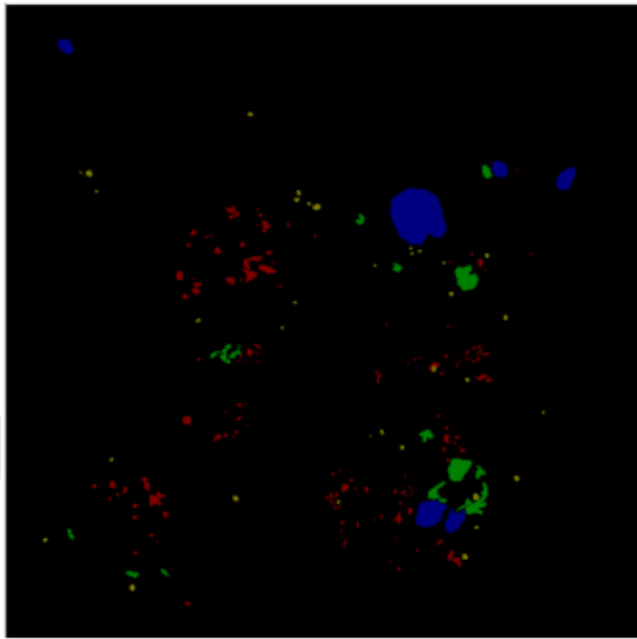
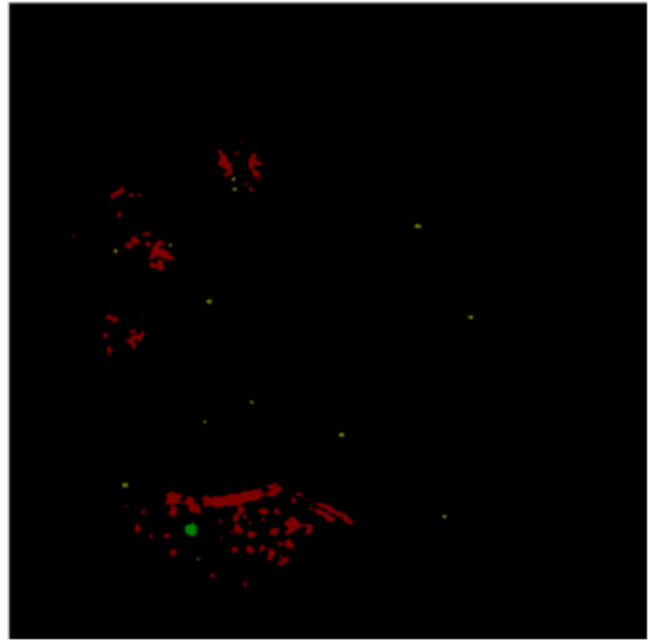
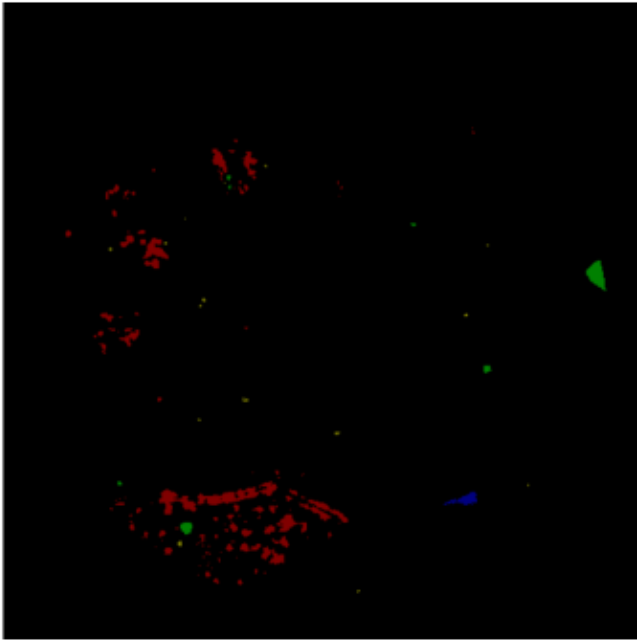


3. 可视化展示

以下是Unet++网络的多分类语义分割训练结果的可视化展示，左侧是ground truth，右侧是predicted result

由对比图可以看出，我们的方法能够较清楚地标记出不同病变的边缘，即使是很多论文公认的最难区分的HE（绿色）和MA（黄色）病变，也可以较好地地区别出来。





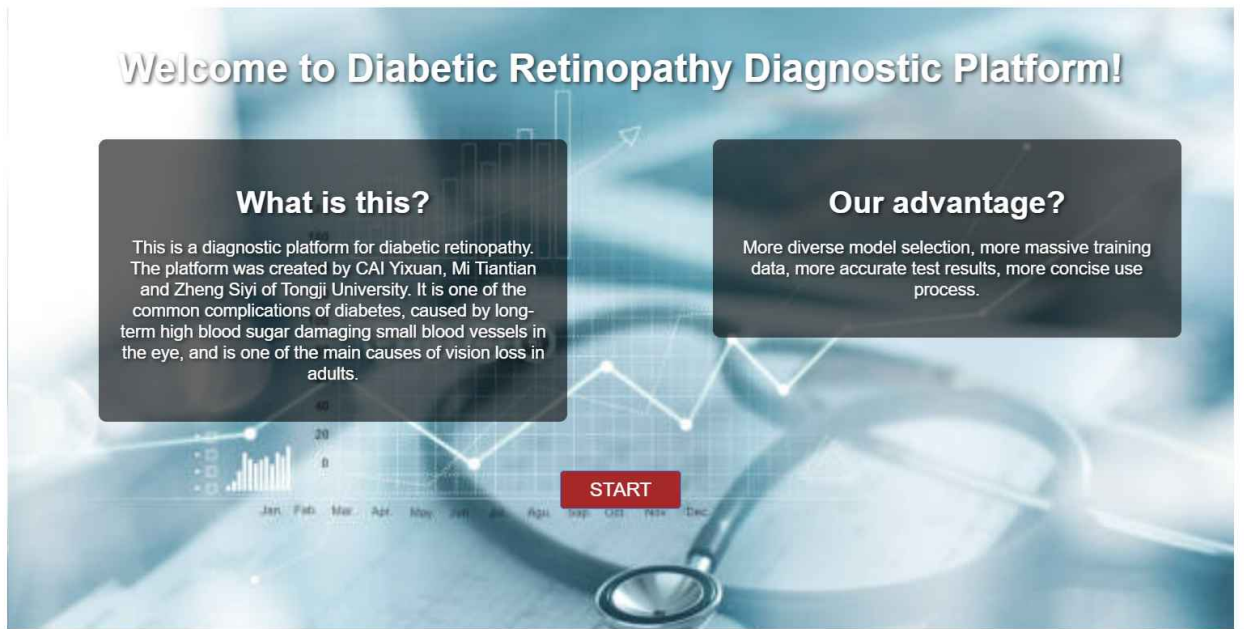
四、前后端

1. 概览

我们的项目采用了 Vue.js 作为前端框架，Python 和 Flask 作为后端框架。

系统分为三个页面：首页、在线诊断、成果介绍

首页简单介绍了我们的研发团队和我们的优势



在线诊断页面提供了两个基本功能：预测诊断和生成诊断报告

预测诊断部分：用户选择眼底病变照片进行上传，继而选择使用的模型（Unet/Unet++/MAnet），提交后，后端会进行推理，并返回四种病变的分割结果和等级预测，前端可以加以展示。此外，用户可以通过画图工具对分割结果进行进一步的修正。

诊断报告生成：用户可输入患者的姓名年龄和诊断意见，点击生成报告，即可获得pdf, 报告中记录了眼底原始图片、模型分割结果、修正结果、患者信息以及诊断意见等

Choose File **Select Model Type** **Submit**

No Image Selected

EX Lesion HE Lesion MA Lesion SE Lesion

No Result Image

No Modified Image

Enter your name Enter your age

Enter diagnosis

Generate Report

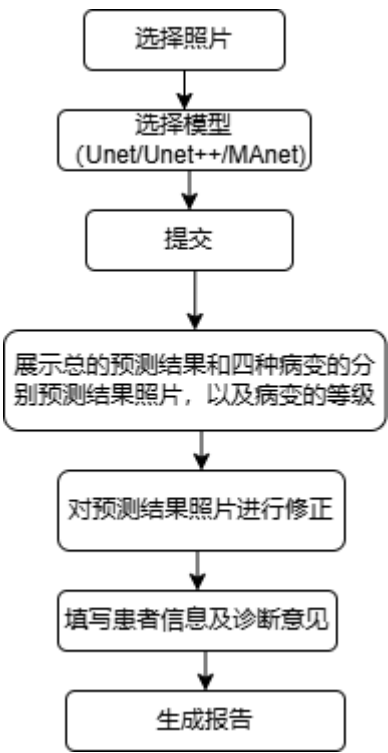
No Image No Image No Image No Image

成果介绍：这一部分主要是一个pdf显示器，其中介绍了我们团队针对糖尿病性视网膜病变的研究经过和研究结果等

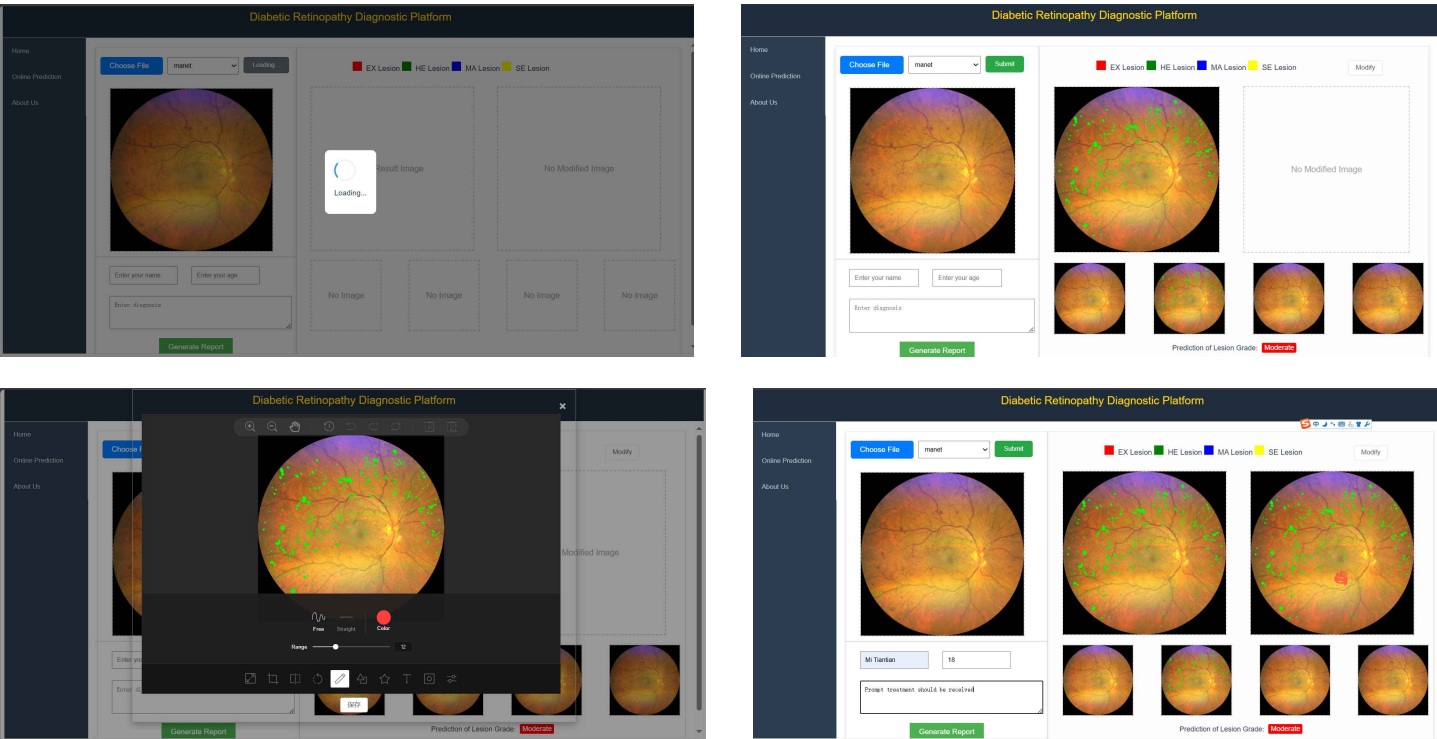
2 详细介绍

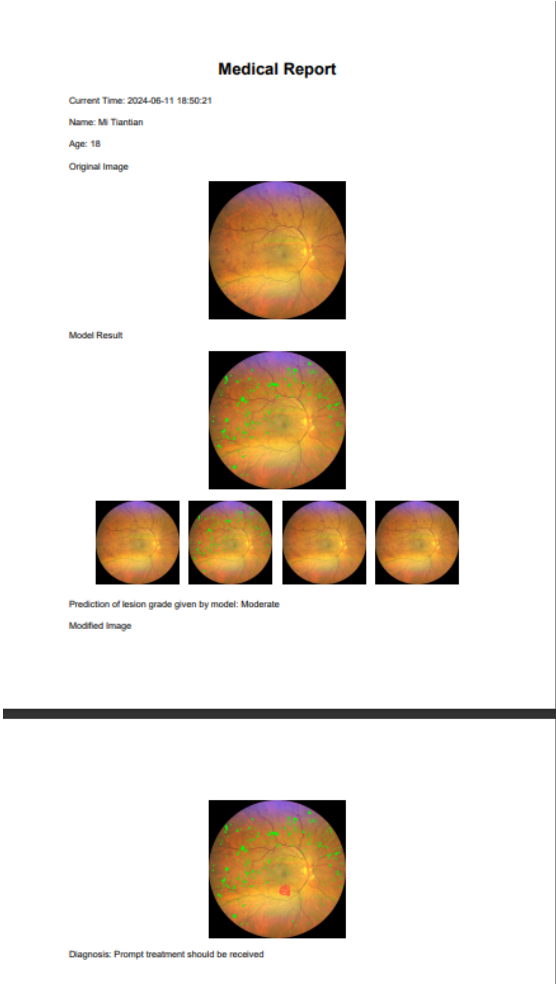
下面针对在线预测诊断部分进行详细的介绍

系统使用的流程图如下所示：



效果如下图所示：





接口介绍：

作用	上传图片
URL	api//upload_image
方法	POST
请求体	使用 <code>FormData</code> 对象传递图片文件，键名为 <code>image</code>
响应	成功上传图片后，服务器返回 JSON 格式的响应消息

作用	选择模型类型并提交
URL	api/choose_model
方法	POST
请求体	传递包含 <code>selectedModel</code> 的 JSON 对象
响应	成功提交后，服务器返回预测结果，包括病变等级和生成图片的 URL

--	--

作用	编辑图片
URL	api/edit_image
方法	POST
请求体	使用 <code>FormData</code> 对象传递图片文件，键名为 <code>image</code>
响应	成功上传图片后，服务器返回 JSON 格式的响应消息，并更新前端显示的图片 URL

作用	生成报告
URL	api/generate_report
方法	POST
请求体	传递包含用户姓名 (name)、年龄 (age)、诊断信息 (diagnosis)、病变等级 (grade) 的 JSON 对象
响应	成功生成报告后，服务器返回包含报告文件的 Blob 对象，前端创建并下载 PDF 报告