

GOALS 挑战赛技术报告

一、任务一：青光眼检测

本项任务中，我们的实验基于 PyTorch 框架进行，使用单张 2080Ti GPU，pipeline 如下图所示。

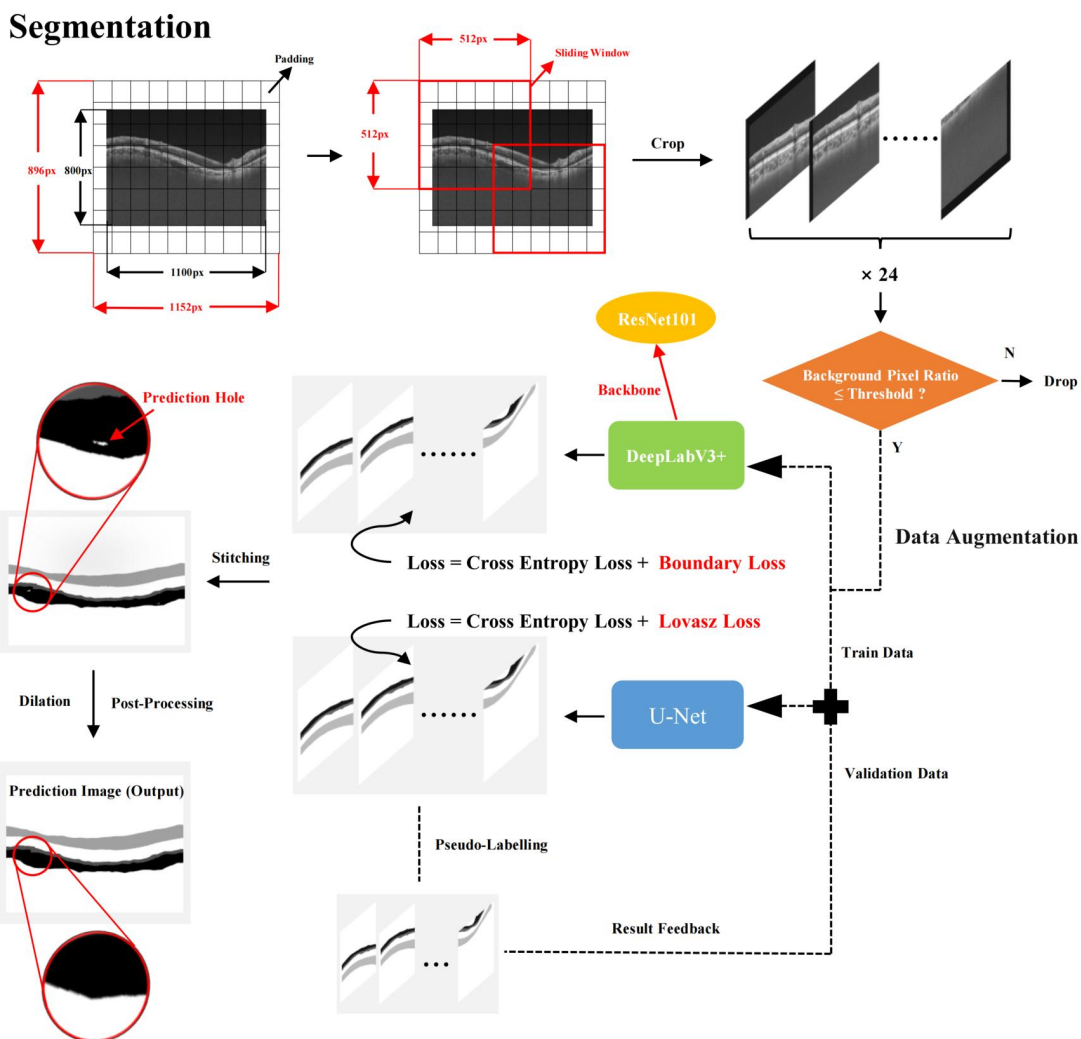


图 1 任务一整体流程示意图

1.1 数据集与预处理

我们使用的数据集为赛事举办方提供的数据集，并未使用任何外部数据集，训练集和验证集按照 4:1 随机划分。

数据集原始图像的分辨率为 1100×800，我们将训练集、验证集和测试集的每张图像切割为 512×512 的大小，训练集和验证集的 label 图像也同步处理，具体处理策略如下：

- (1) 首先填充原始图像至 896×1152，宽度方向左右各填充 26，高度方向上

下各填充 48，其中 image 用 RGB 为 (0, 0, 0) 的像素点填充，label 用 RGB 为 (255, 255, 255) 的像素点填充；

(2) 用 512×512 的窗口大小、步长为 128 的滑窗进行切割，并记录所在位置坐标，若窗口中 label 图片的背景区域像素比例大于 99%，视为干扰数据，跳过该窗口。这样处理过后，每张原始图片大约会产生 24 张小的切割图。

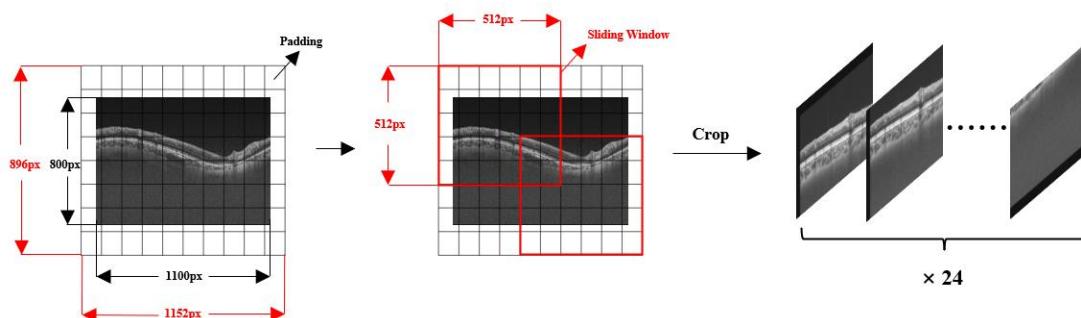


图 2 数据预处理流程

1.2 模型选择

我们选择 U-Net 和 DeepLabV3+模型进行训练。其中，U-Net 模型的输入改为 512×512 ，结构未作改动，DeepLabV3+模型的 backbone 替换为 ResNet101。

1.2.1 U-Net

U-Net 结构包括编码器下采样、解码器上采样和同层跳跃连接三个组成部分。编码器由 4 组卷积、ReLU 激活和最大池化构成，每一组均有两次 3×3 的卷积，每个卷积层后面都有一次 ReLU 激活函数，然后再进行一次步长为 2 的 2×2 最大池化进行下采样。对于原始模型来说，第一组操作输入图像大小为 572×572 ，两轮 3×3 的卷积之后的特征图大小为 568×568 ，再经过 22 最大池化后的输出尺寸为 284×284 。解码器由 4 组 2×2 转置卷积、 3×3 卷积构成和一个 ReLU 激活函数构成，在最后的输出层又补充了一个 1×1 卷积。最后是同层跳跃连接，这也是 U-Net 的特色操作之一，指的是将下采样时每一层的输出裁剪后连接到同层的上采样层做融合。每一次下采样都会有一个跳跃连接与对应的上采样进行融合，这种不同尺度的特征融合对上采样恢复像素大有帮助，具体来说就是高层（浅层）下采样倍数小，特征图具备更加细致的图特征，低层（深层）下采样倍数大，信息经过大量浓缩，空间损失大，但有助于目标区域判断，当高层和低层的特征进行融合时，分割效果往往会非常好。

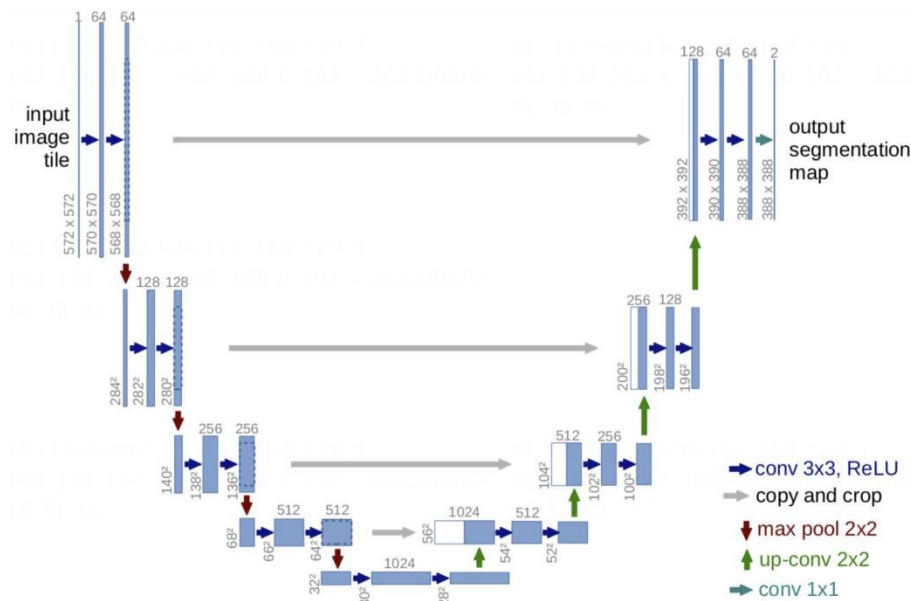


图 3 U-Net 网络结构

1.2.2 DeepLabV3+

DeepLabV3+的网络结构如下图所示，主要为 Encoder-Decoder 结构。其中，Encoder 为改进的 DeepLabV3，Decoder 为 3+版本新提出的。

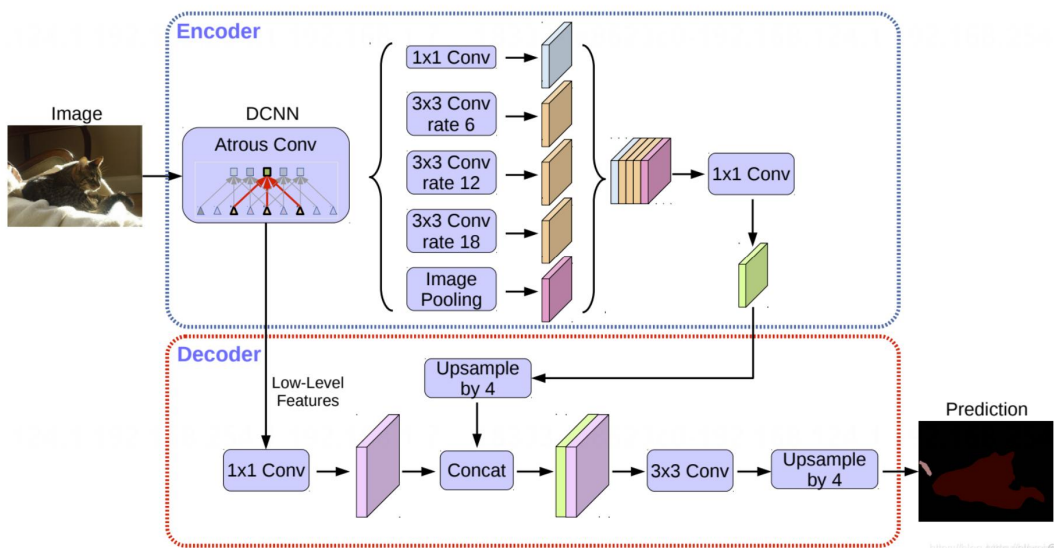


图 4 DeepLabV3+网络结构

在 Encoder 部分，主要包括了 backbone（即图中的 DCNN）、ASPP 两大部分。其中 backbone 有两种网络结构：将 layer4 改为空洞卷积的 Resnet 系列、改进的 Xception。从 backbone 出来的 feature map 分两部分：一部分是最后一层卷积输出的 feature maps，另一部分是中间的低级特征的 feature maps；backbone 输出的第一部分送入 ASPP 模块，第二部分则送入 Decoder 模块。ASPP 模块接受 backbone 的第一部分输出作为输入，使用了四种不同膨胀率的空间卷积块（包括卷积、BN、激活层）和一个全局平均池化块（包括池化、卷积、BN、激活层）

得到一共五组 feature maps，将其 concat 起来之后，经过一个 1×1 卷积块（包括卷积、BN、激活、dropout 层），最后送入 Decoder 模块。

在 Decoder 部分，接收来自 backbone 中间层的低级 feature maps 和来自 ASPP 模块的输出作为输入。首先，对低级 feature maps 使用 1×1 卷积进行通道降维，从 256 降到 48（之所以需要降采样到 48，是因为太多的通道会掩盖 ASPP 输出的 feature maps 的重要性，且实验验证 48 最佳）；然后，对来自 ASPP 的 feature maps 进行插值上采样，得到与低级 feature maps 尺寸相同的 feature maps；接着，将通道降维的低级 feature maps 和线性插值上采样得到的 feature maps 使用 concat 拼接起来，并送入一组 3×3 卷积块进行处理；最后，再次进行线性插值上采样，得到与原图分辨率大小一样的预测图。

1.3 数据增强

训练 U-Net 时，借助 imgaug 库中的弹性变形、水平翻转、水平翻转以及伽马增强对图像进行在线增强，并且它们作用在图像上的顺序随机。其中，弹性变形的参数设为 $\alpha=300$ 、 $\sigma=30$ ，水平翻转和水平翻转的概率都设为 0.5，伽马变换的参数设为 $\gamma=(0.5, 2.0)$ 。

训练 DeepLabV3+时，利用随机平翻转和垂直翻转对图像进行在线增强，两者概率均为 0.5。

1.4 训练细节

对于 U-Net，每个 batch 的大小为 4，进行 100 轮训练，使用交叉熵损失和 Lovasz Loss 之和作为损失函数。采用的优化器为 Adam，学习率为 1×10^{-3} ，L2 权重衰减为 1×10^{-5} ，并使用 AMSGrad 的变体。

对于 DeepLabV3+，每个 batch 的大小为 2，进行 100 轮训练，使用交叉熵 Loss 和 Boundary Loss 之和作为损失函数。采用的优化器为 Adam，学习率为 1×10^{-3} ，L2 权重衰减为 1×10^{-5} 。

Model	Batch_size	Epoch	Loss function	Learning rate	Optimizer
U-Net	4	100	CE&Lovasz loss	1×10^{-3}	AMSGrad
DeepLabV3+	2	100	CE&Lovasz loss	1×10^{-3}	Adam

表 1 任务一训练细节

1.5 测试增强

测试时，通过对图像水平翻转、垂直翻转、水平垂直翻转等操作进行多次预测，再对预测结果取平均以提高预测精度。

1.6 膨胀预测

由于模型对图片边缘像素预测不准，我们在测试集上进行预测时，只保留每张小图预测结果的中心区域（尺寸为 416×460），以舍弃预测不准的边缘，并根据预处理时记录的坐标将各小图的结果拼接起来，最终得到针对原始图像的完整预测结果。

1.7 后处理

由于观察到部分预测图像中存在空洞和明显的误判区域，我们在后处理中做了孔洞填充和小连通域去除的处理。

具体操作是将每个像素的预测结果转为 one-hot 向量，预测结果变为 4 通道图像，在每一通道上统计 0 连通域和 1 连通域的像素点个数，若某连通域的像素个数小于 3000，则用另一数字替换。

1.8 伪标签技术

对于 U-Net，在测试集预测效果最好的模型结果作为伪标签，将带有伪标签的测试集混入训练集中，使用有标记数据以及伪标签数据，按照上述步骤重新训练 U-Net 模型，进一步提升模型的分割质量。

1.9 人工筛选

通过对比 U-Net 的后处理结果和 DeepLabV3+ 的后处理结果，可以明显看出 U-Net 的分割结果普遍更好，但部分图像使用 DeepLabV3+ 模型的分割效果更好，所以我们将 U-Net 预测结果中的部分图片用 DeepLabV3+ 的结果进行了替换。

任务二：青光眼检测

本项任务中，我们基于 PaddlePaddle 框架进行相关实验，使用单张 2080Ti GPU。

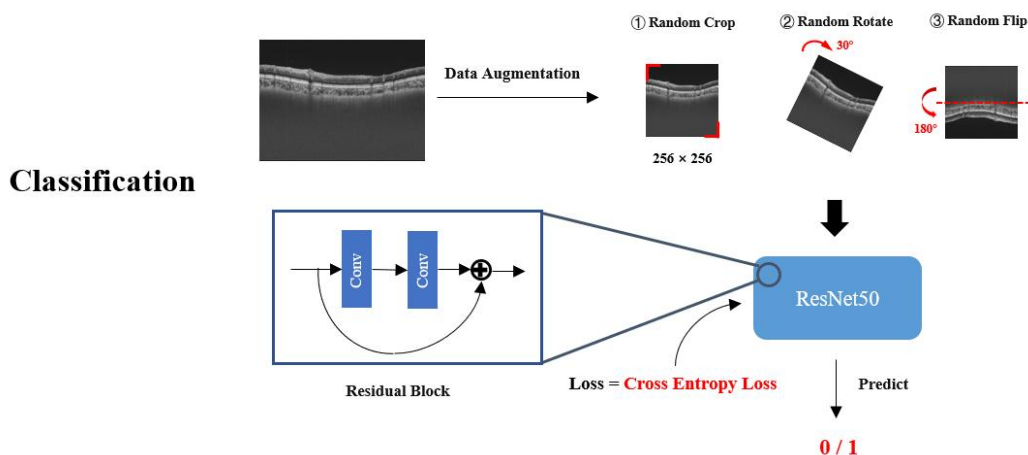


图 5 任务二整体流程示意图

2.1 数据集与数据增强

我们使用的数据集为赛事举办方提供的数据集，并未使用任何外部数据集，同时训练集和验证集按照 4:1 划分。

对于训练数据集，我们采用较为常用的数据增强方法，对于数据进行随机裁剪、随机上下水平翻转、随机旋转等。

对于验证数据集，我们也采用随机裁剪和固定值 `resize`。

2.2 模型选择

由于数据集数量较小，我们采用 `resnet50` 作为我们的基线模型。`resnet` 模型通过残差结构来解决模型梯度消失，深层网络难训练的问题。`resnet50` 主体上由 4 个大 `layer` 层构成，每个 `layer` 分别由 3、4、6、3 个小 `block` 组成，每个小 `block` 里面有三个卷积层，另外这个网络的最开始有一个单独的卷积层，因此是： $(3+4+6+3) \times 3 + 1 = 49$ 最后又一个全连接层，因而一共为 50 层。我们采用预训练好的 `resnet50`，去掉分类头，后再加两层全连接起到分类的效果。

2.3 训练细节

我们设置 `batch size` 为 8，对输入图片 `resize` 到 256×256 的大小，经过 1000 轮迭代训练，初始学习率为 1×10^{-3} ，采用 `Adam` 作为模型的优化器。

Model	Batch_size	Epoch	Img_size	Learning rate	Optimizer
ResNet50	8	1000	256×256	1×10^{-3}	Adam

表 2 任务二训练细节

结果

最终,对于任务一采用 DICE 系数和分割边缘平均欧氏距离来作为评价指标,通过表 2 我们可以看到,我们的模型对 RNFL 和 choroid 区域分割效果较好,但是对于 GCIPL 区域的效果较差,这也是我们未来在此项工作上的进一步研究方向。而对于任务二来说,我们对于数据拟合的效果非常好,在测试集的测试中,模型输出的 AUC、F1 分数、准确度、灵敏度、特异度都达到了满分。

Score	AUC	F1	ACC	SEN	SPE
8.7991	1	1	1	1	1

RNFL_DICE	RNFL_ED	GC IPL_DICE	GC IPL_ED	choroid_DICE	choroid_ED
0.9471	1.1041	0.8858	1.3197	0.9425	1.8394

表 3 任务一和任务二结果

参考文献

- [1] Chen L C, Zhu Y, Papandreou G, et al. Encoder-decoder with atrous separable convolution for semantic image segmentation[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 801-818.
- [2] Kervadec H, Bouchtiba J, Desrosiers C, et al. Boundary loss for highly unbalanced segmentation[C]//International conference on medical imaging with deep learning. PMLR, 2019: 285-296.
- [3] Jadon S. A survey of loss functions for semantic segmentation[C]//2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB). IEEE, 2020: 1-7.