

基于 Unet++ 网络的路面裂缝分割方法

张越

Saturday 22nd January, 2022

目录

1 前言	2
2 图像分割技术简介	2
3 对 Unet 的思考	3
4 Unet++ 网络简介	4
4.1 设计思路	4
4.2 深监督	5
4.3 剪枝	5
4.4 Unet++ 总结	6
5 项目介绍	6
5.1 项目结构	7
6 模型训练	8
6.1 训练参数设置	8
6.2 训练过程	8
6.3 关键训练方法及参数介绍	8
6.3.1 数据增强的方法	8
6.3.2 根据预训练确定迭代次数	9
6.3.3 优化器	9
6.3.4 激活函数和损失函数	9
6.3.5 评价指标	10
6.4 模型训练效果	10
7 展望	11
8 结语	11

1 前言

路面裂痕分割是图像多分类的特殊情况——二分类问题。图像分割主要思想：输入一张图片，经过一连串下采样编码后，得到比原来更小的特征，相当于图像压缩，再经过一系列解码，在理想状态下图像还原成原来的状态，并得到分割结果。

本文用 pytorch 框架上基于 Unet++ 神经网络很好地完成了路面裂痕检测任务，可能也是 Unet++ 首次运用于路面裂痕检测，实验表明，其在检测细节方面有着更高的精度。

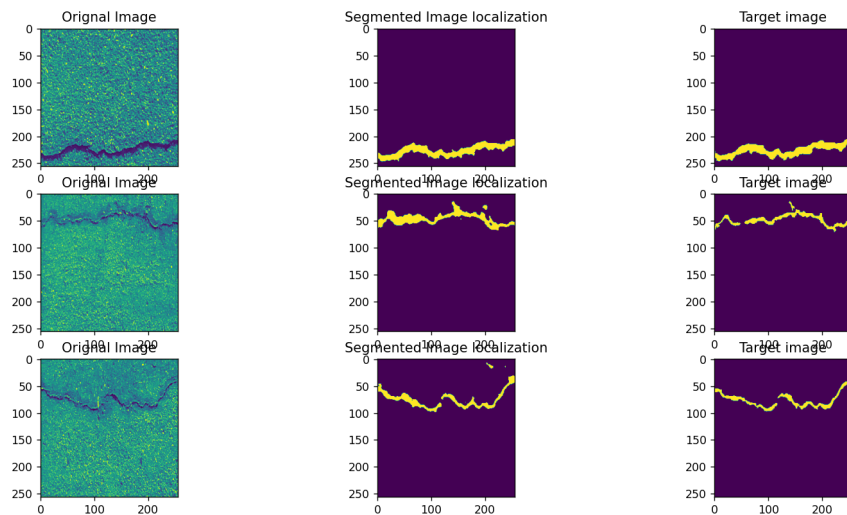


图 1: Unet++ 运用于路面裂痕检测的效果

2 图像分割技术简介

在计算机视觉领域，全卷积神经网络（FCN）、SegNet、Unet 是比较出名的几个图像分割网络，可以应用于医学影像分割等。

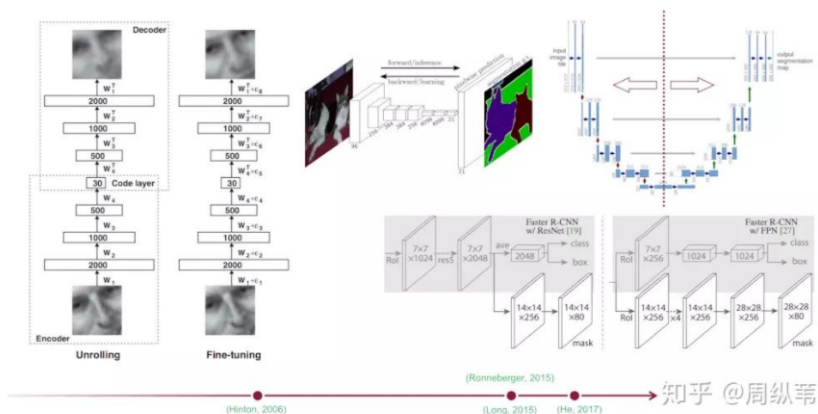


图 2: 图像分割技术的进程

图 2 显示了图像分割技术的进程，2006 年 Hinton 提出了图像编码、解码的过程；基于 encoder-decoder，2015 年 FCN、Unet 被发表。

FCN 和 Unet 的对比

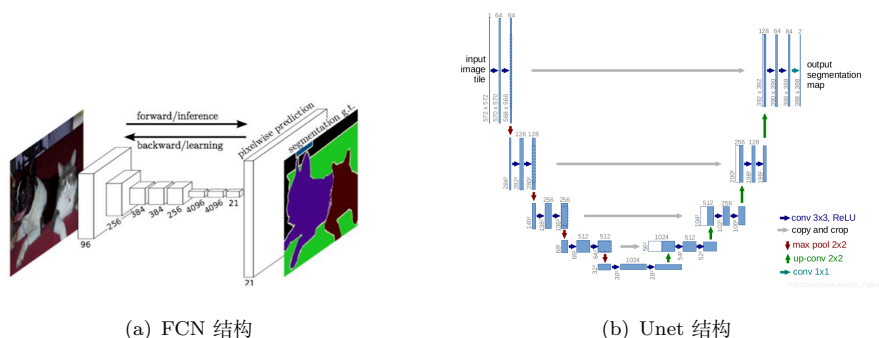


图 3: 两种常见的分割网络结构

1. Unet 网络由上采样、下采样、skip connection 构成，结构优美，完全对称，FCN 的结构简单，仅由卷积和反卷积组成。
2. 在 skip connection 上，FCN 用的是加操作 (add)，Unet 用的是叠操作 (concat)。

3 对 Unet 的思考

1. Unet 这个三年不动的拓扑结构真的一点儿毛病都没有吗？

这个结构最优秀的点就在于结合了深浅层特征，而且结合的方式优于 FCN，很多模型都是基于这个基础上做改进。

2. 2、Unet 要多深合适？

并不是所有的问题增加深度就能提高性能。有些问题简单，浅层网络就能解决，增加深度并不能增加效果。有些问题难，就需要深层网络。

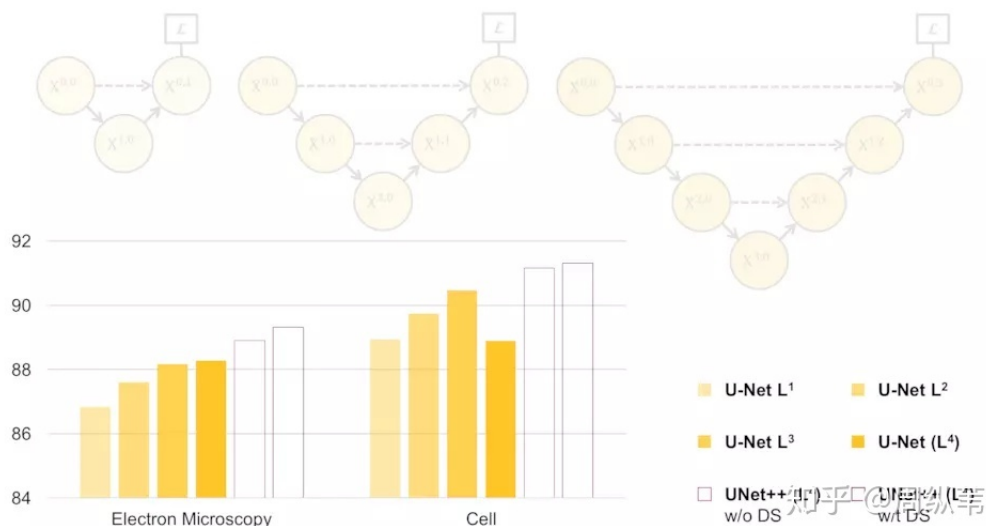


图 4: 不同深度 Unet 的表现

作者为了验证多深才好, 分别用了两个数据集: Electron Microscopy 和 Cell, 每加一个深度就训练一个网络, 然后测它们各自的分割表现。

先不要看图 4 后两个 UNet++, 就看这个不同深度的 UNet 的表现 (黄色条形图), 我们可以看出, 不是越深越好, 它背后的传达的信息就是, 不同层次特征的重要性对于不同的数据集是不一样的, 并不是说设计一个原论文给出的那个结构, 就一定对所有数据集的分割问题都最优。

3. 降采样对于分割网络到底是不是必须的？

它可以增加对输入图像的一些小扰动的鲁棒性, 比如图像平移, 旋转等, 减少过拟合的风险, 降低运算量, 和增加感受野的大小。

4. 所抓取的特征都很重要, 为什么非要降 16 倍了才开始上采样回去呢？

并不是非要降低 16 倍, 而是根据自身问题的难度选择降低的倍数。更简明的说, 就是你选取多少层, 这个你需要尝试。

4 Unet++ 网络简介

4.1 设计思路

不同数据集的最优的深度是不一样的, 但是总不能把所有不同深度的 U-Net 都训练一遍, 太耗时间了, 作者于是提出 Unet++。

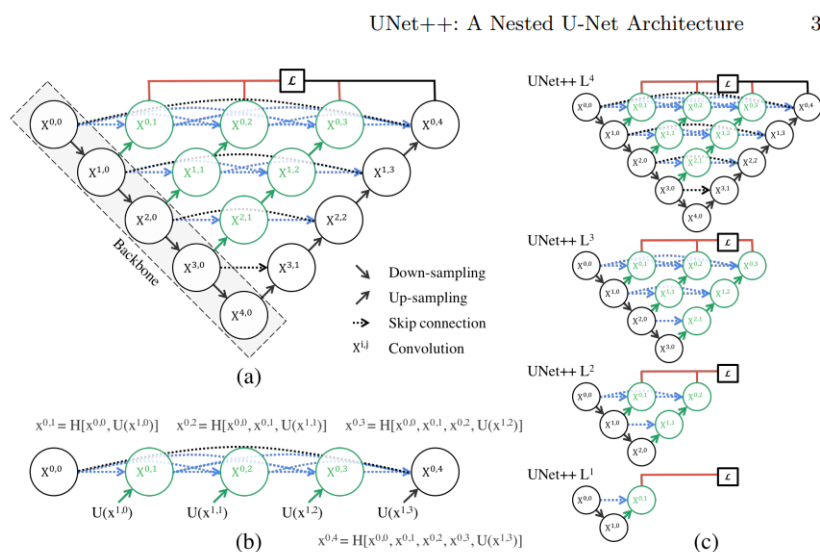


图 5: Unet++ 网络结构

这个网络里面可以找到 14 层的 UNet, 它们通过长短连接全都连在了一起, 短连接是为了使模型能够得到训练, 然后长连接是获取更多信息。这个结构的好处就是不管在哪个深度的特征有效, 都给用上, 让网络自己去学习不同深度的特征的重要性。第二个好处是它共享了一个特征提取器 (encoder), 也就是不需要训练一堆 UNet, 而是只训练一个 encoder, 它的不同层次的特征由不同的 decoder 路径来还原。这个 encoder 依旧可以灵活的用各种不同的 backbone 来代替。

4.2 深监督

在训练网络反向传播的时候，如果只用最右边的一个 loss 来做的话，中间部分会收不到过来的梯度，导致无法训练，解决办法除了用短连接的那个结构外，还有一个方法就是用深监督（deep supervision）。如图 6 所示，具体的实现操作就是在图中 $X^{0,1}$ 、 $X^{0,2}$ 、 $X^{0,3}$ 、 $X^{0,4}$ 后面加一个 1×1 的卷积核，相当于去监督每个 level，或者说监督每个 level，或者说监督每个分支的 UNet 的输出。这样可以解决那个结构无法训练的问题。

虽然通过加入短连接解决了无法训练的问题，但是作者仍然把 deep supervision 加入到了 UNet++ 中。因为这会带来一个非常棒的优势，就是剪枝。

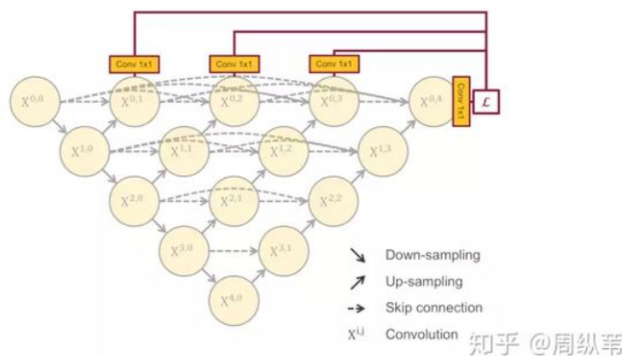


图 6: UNet++ 深监督示意图

4.3 剪枝

在测试阶段，由于输入的图片只会前向传播，扔掉这部分对前面的输出完全没有影响的，而在训练阶段，因为既有前向，又有后向传播，被剪掉的部分是会帮助其他部分做权重更新的。

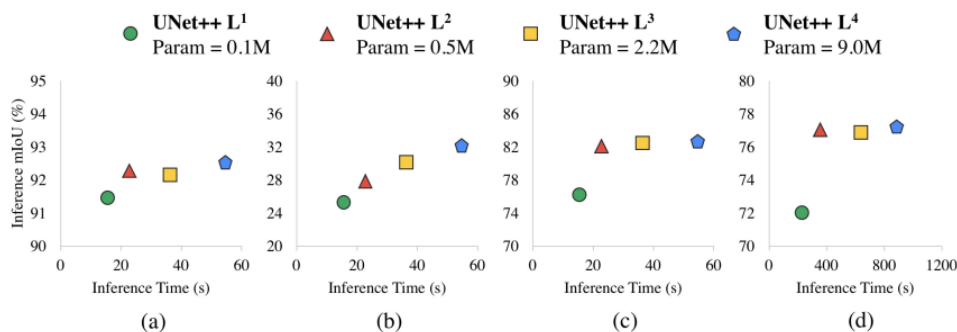


图 7: 监督各个子网络在验证集下的运行效果

由图 7 可知，其中有三个数据集显示 L2 的结果和 L4 已经非常接近了，也就是说对于这三个数据集，在测试阶段，我们不需要用 9M 的 L4 网络，用 0.5M 的 L2 网络就足够了。可以根据子网络在验证集的结果来决定剪多少，大大减少网络参数。

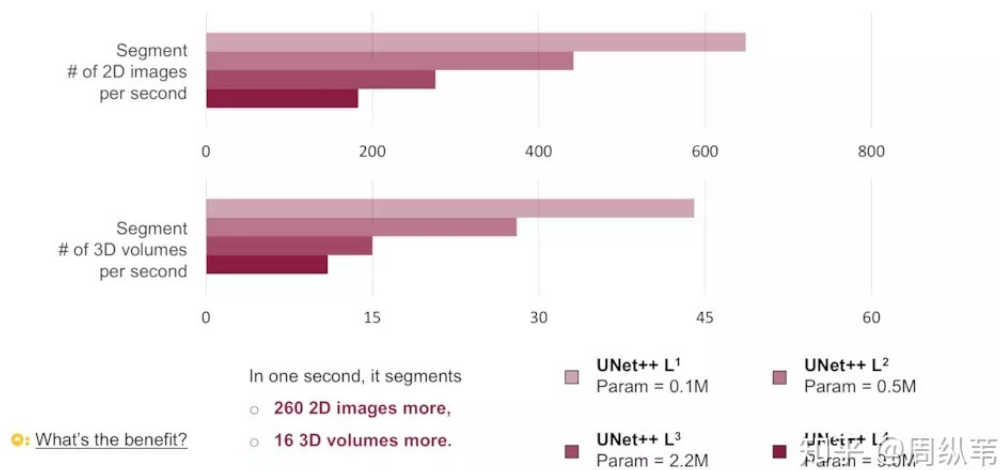


图 8: 各个子网络的测试速度

对于测试的速度，上图统计了用不同的模型，1 秒钟可以分割多少的图。如果用 L2 来代替 L4 的话，速度确实能提升三倍。

4.4 Unet++ 总结

- UNet++ 的第一个优势就是精度的提升，这是它整合了不同层次的特征所带来的。
- 是灵活的网络结构配合深监督，让参数量巨大的深度网络在可接受的精度范围内大幅度的缩减参数量。

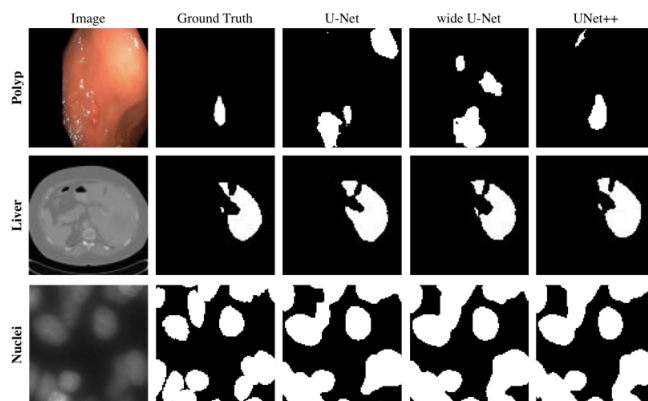


图 9: Unet、wide Unet、Unet++ 效果对比

5 项目介绍

在做本项目前，我对助教提供的网络进行了学习、理解和训练，该 Unet 网络在 epoch=5, 学习率为 0.001，迭代 300 次时模型训练效果还可以，loss 能下降到 0.01。

之所以选择用 Unet++ 来完成本任务是因为我无意间看到了 Unet++ 的作者在知乎上面的一篇文章，作者详细地剖析了 Unet++ 设计思路，相比 Unet 的优势和验证 Unet++ 优势的思路，让我对神经网络有了更深入的理解。他说的：“一个网络结构，我们真正值得关注的是它们设计给我们传达了什么信息，给我们怎样的思考，所以我们应该要关注论文所传递的大方向，不要被论文中的细节局限了创造力，而像这种细节参数的调整属于比较朴素的深度学习方法论，很容易花费你很多时间，而最终没有自身科研水平的提升。”令我颇有感触。

项目代码基于一个多分类的医学影像分割项目 [5] 改造而成，测试结果证明了该网络有较高的精度。

5.1 项目结构

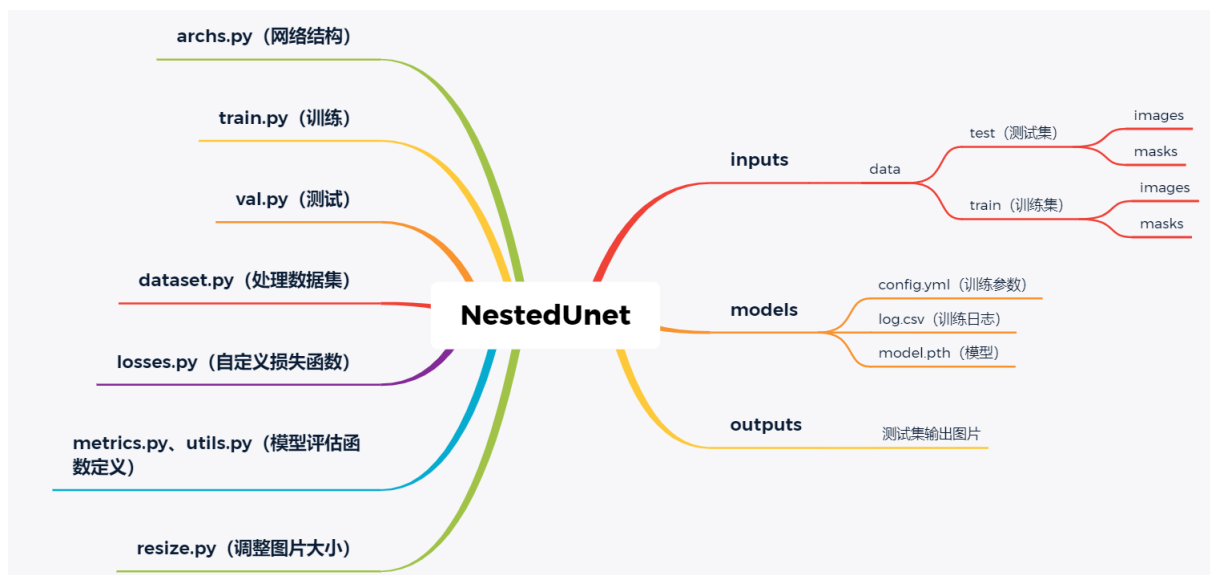


图 10: 项目结构

现在该项目已上传到我的 github 上：

<https://github.com/Kevin-Zhang-SYSU/NestedUnet>

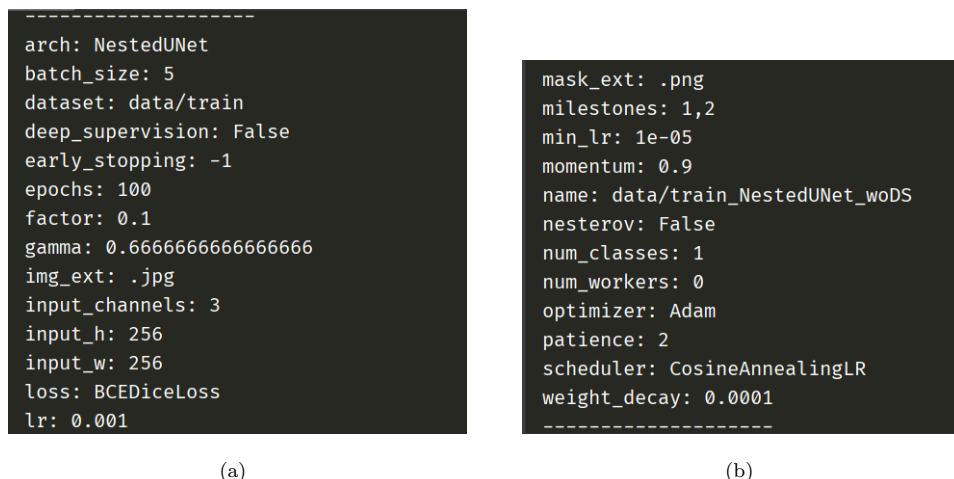


图 11: 训练参数

6 模型训练

6.1 训练参数设置

6.2 训练过程

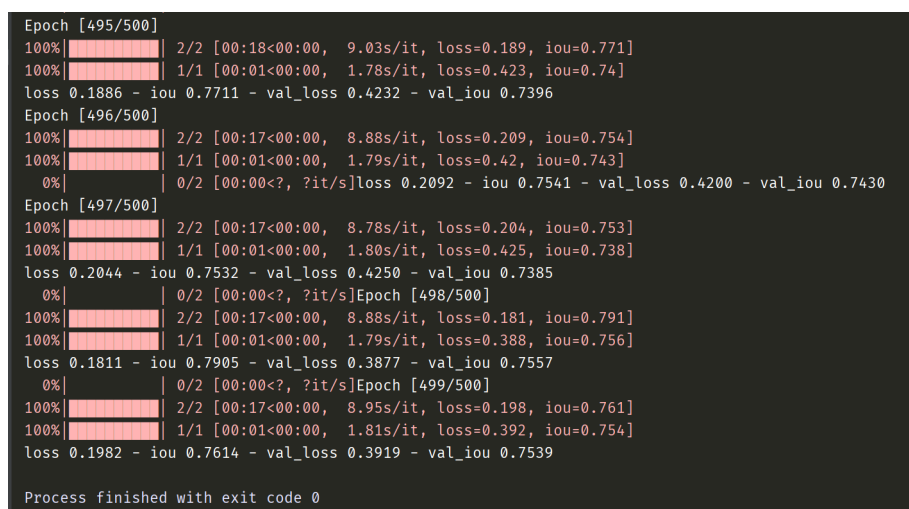


图 12: epochs=500 时的训练过程

6.3 关键训练方法及参数介绍

介绍本项目中几个对提高模型性能较为重要的方法。

6.3.1 数据增强的方法

只用 15 张图片进行训练显然是不够的，训练一般要求样本的数量要充足，样本数量越多，训练出来的模型效果越好，模型的泛化能力越强。对图像进行随机翻转，旋转和 HSV 上的调整可以增加训练的数据量，提高模型的泛化能力和鲁棒性。


```
# 数据增强:
train_transform = Compose([
    albu.RandomRotate90(),# 随机旋转
    transforms.Flip(),# 随机翻转
    OneOf([
        transforms.HueSaturationValue(),# 色调饱和和度值
        transforms.RandomBrightness(),# 随机亮度
        transforms.RandomContrast(),# 随机对比度
    ], p=1), # 按照归一化的概率选择执行哪一个
    albu.Resize(config['input_h'], config['input_w']),# 调整大小
    transforms.Normalize(),# 归一化
])
```

6.3.2 根据预训练确定迭代次数

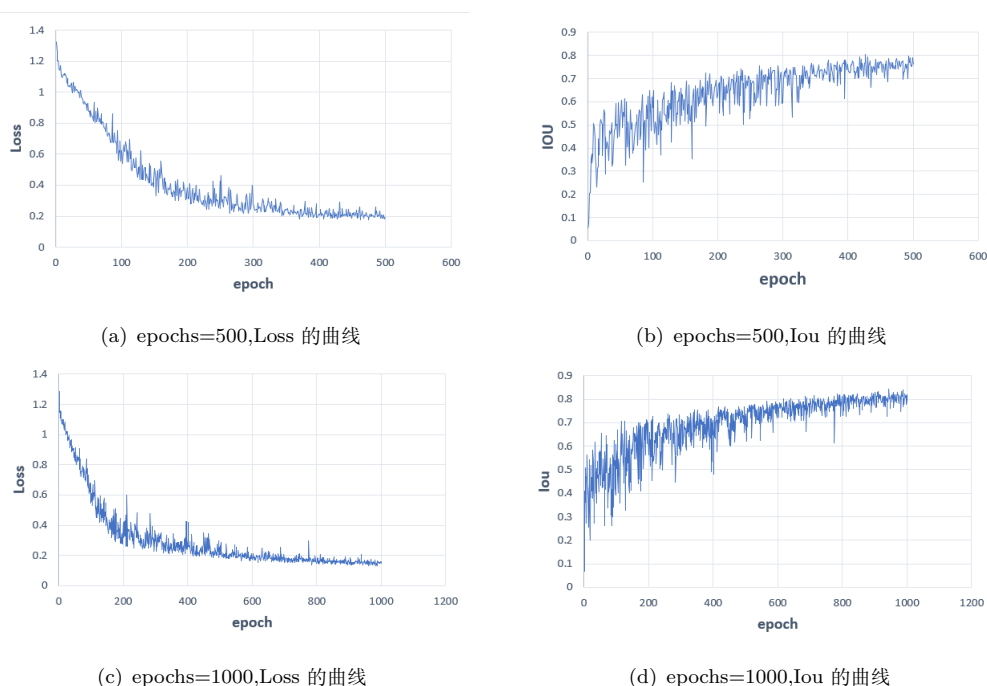


图 13: epochs=500/1000 时 Loss 和 Iou 的变化过程

分别对模型在 epochs=300, 500, 1000 时进行预训练。由上图可知, 在 epochs=300 时, 训练效果还未达到最好, 模型在 epochs=500 后时 Loss 和 Iou 下降得很缓慢, 趋于稳定。考虑到训练所需时间的消耗, 若在 cpu 上训练在正式训练时 epochs=600 左右较为合理。

6.3.3 优化器

项目一开始用的是 SGD, 结果发现收敛速度很慢, 换成 Adam 后能大大加快模型收敛速度。

6.3.4 激活函数和损失函数

网络模型中间激活层采用 Relu 激活函数, 同时对输入数据样本进行了多次正则化 (Batch Normalization, BN)。BN 是每层卷积网络的输出值调整为相同分布, 以避免网络加深导致特征向量的分布发生偏移

或者变动。

二分类交叉熵

$$L = -\frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (1)$$

其中， N 是总样本数， y_i 是第 i 个样本的所属类别， p_i 是第 i 个样本的预测值，一般来说，它是一个概率值。

基于二值交叉熵损失函数的自定义损失函数

```
class BCEDiceLoss(nn.Module):
```

```
    def __init__(self):
```

```
        super().__init__()
```

```
    def forward(self, input, target):
```

```
        bce = F.binary_cross_entropy_with_logits(input, target)
```

```
        smooth = 1e-5
```

```
        input = torch.sigmoid(input)
```

```
        num = target.size(0)
```

```
        input = input.view(num, -1)
```

```
        target = target.view(num, -1)
```

```
        intersection = (input * target)
```

```
        dice = (2. * intersection.sum(1) + smooth) / (input.sum(1) + target.sum(1) + smooth)
```

```
        dice = 1 - dice.sum() / num
```

```
        return 0.5 * bce + dice
```

6.3.5 评价指标

在训练过程中，会根据 Iou 来决定模型的保存与否，保存 Iou 尽可能大的模型。



图 14: Iou 计算原理

6.4 模型训练效果

每张图的检测时间约为 0.5s，模型输出图片大小为 256*256，经过 resize 后可以调整为原来的大小，下图中 *.jpeg 为原始图片，*.jpg 为预测结果，*.png 为掩膜。在细节方面，Unet++ 的效果很好，能感受到微小的细节变化。

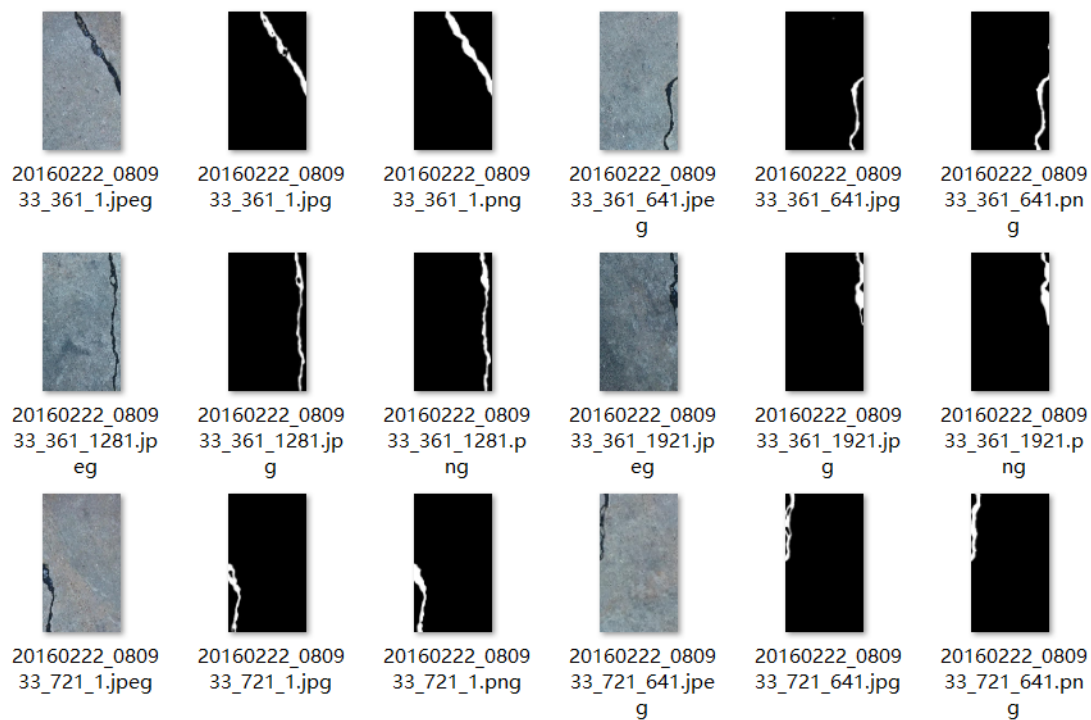


图 15: 模型训练效果

7 展望

由于实验硬件条件及时间的限制，所以目前的学习工作还有很多欠缺，仍存在一些需要改进的地方。比如，训练样本少，只取了 Crack500 路面裂缝数据集里切片后的 15 张作为训练集和验证集；超参数设置没有经过大量的实验验证，模型收敛速度和识别精度仍有提升的空间。

8 结语

本文训练 Unet++ 神经网络完成了路面裂缝分割的任务。相比于 Unet 网络，Unet++ 通过网络不同层次的特征融合，提高了裂缝区域分割的准确性。同时，本文通过数据增强，提高了基于小样本数据集神经网络训练的泛化能力和鲁棒性。通过对比试验和结果分析，证明了 Unet++ 网络裂缝分割方法的有效性。

参考文献

- [1] Zhou, Zongwei, et al. "Unet++: A nested u-net architecture for medical image segmentation." Deep learning in medical image analysis and multimodal learning for clinical decision support. Springer, Cham, 2018. 3-11.
- [2] 孙朝云, 马志丹, 李伟, 郝雪丽, 申浩. 基于深度卷积神经网络融合模型的路面裂缝识别方法 [J]. 长安大学学报 (自然科学版), 2020, 40(04): 1-13. DOI: 10.19721/j.cnki.1671-8879.2020.04.001.
- [3] 王丽苹. 基于深度学习的混凝土路面裂缝检测 [D]. 河北工程大学, 2018.

- [4] <https://zhuanlan.zhihu.com/p/44958351>
- [5] <https://gitee.com/xiaoshaKK/unet-pp-Medical-cell-segmentation>
- [6] https://blog.csdn.net/weixin_40519315/article/details/104457459
- [7] https://blog.csdn.net/qq_39056987/article/details/105103365