深度学习实践-实验六

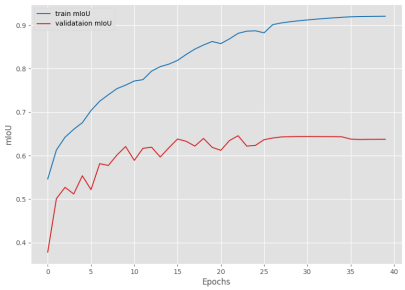
PB22151796-莫环欣

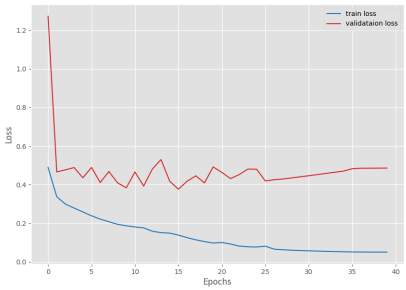
1.了解一下整段代码的结构(/work/code/src/)，将代码跑通 (跑通后，见 outputs 文件夹), 解释说明分割任务的评价标准。

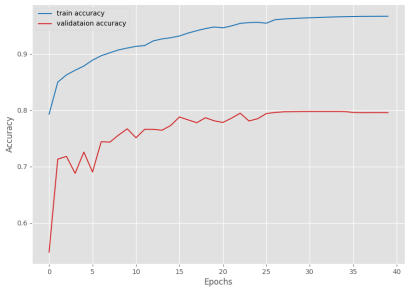
移除了`train.py`里调度器的`verbose`参数（这个已经废弃了，会报警告），还有`utils`里`imwrite`提示格式不匹配自动转为`unint8`

```
scheduler = MultiStepLR(optimizer, milestones=[25, 35], gamma=0.1)
```

使用默认参数运行，网络为`UNet5()`







评价标准有：

- `miou`
  - 平均交并比
  - `iou`：对每个类别：预测分割结果与真实分割结果的交集面积/并集面积
    - 对所有`iou`取平均即为`miou`
- `pa`
  - 像素准确率
  - 预测正确的像素占总像素的比例
  - 也可以对每个类分别计算然后取平均
- `loss`
  - 通用指标，损失
  - 预测结果与真实标签的差距，越小越好

2.用 torchsummary 或者 torchinfo 输出代码中提供的 UNet3() 和 UNet5()这两个模型的每一层的网络详细信息、参数数量和参数总量。

这里使用`torchinfo`，代码放置在`model`中

```
UNet3 Summary:
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
UNet3                                    [1, 10, 512, 512]         --
├─Bottleneck: 1-1                        [1, 64, 512, 512]         --
│    └─BaseConv: 2-1                     [1, 32, 512, 512]         --
│    │    └─Conv2d: 3-1                  [1, 32, 512, 512]         96
│    │    └─BatchNorm2d: 3-2             [1, 32, 512, 512]         64
│    │    └─SiLU: 3-3                    [1, 32, 512, 512]         --
│    └─BaseConv: 2-2                     [1, 64, 512, 512]         --
│    │    └─Conv2d: 3-4                  [1, 64, 512, 512]         18,432
│    │    └─BatchNorm2d: 3-5             [1, 64, 512, 512]         128
│    │    └─SiLU: 3-6                    [1, 64, 512, 512]         --
├─MaxPool2d: 1-2                         [1, 64, 256, 256]         --
├─Bottleneck: 1-3                        [1, 128, 256, 256]        --
│    └─BaseConv: 2-3                     [1, 64, 256, 256]         --
│    │    └─Conv2d: 3-7                  [1, 64, 256, 256]         4,096
│    │    └─BatchNorm2d: 3-8             [1, 64, 256, 256]         128
│    │    └─SiLU: 3-9                    [1, 64, 256, 256]         --
│    └─BaseConv: 2-4                     [1, 128, 256, 256]        --
│    │    └─Conv2d: 3-10                 [1, 128, 256, 256]        73,728
│    │    └─BatchNorm2d: 3-11            [1, 128, 256, 256]        256
│    │    └─SiLU: 3-12                   [1, 128, 256, 256]        --
├─MaxPool2d: 1-4                         [1, 128, 128, 128]        --
├─Bottleneck: 1-5                        [1, 256, 128, 128]        --
│    └─BaseConv: 2-5                     [1, 128, 128, 128]        --
│    │    └─Conv2d: 3-13                 [1, 128, 128, 128]        16,384
│    │    └─BatchNorm2d: 3-14            [1, 128, 128, 128]        256
│    │    └─SiLU: 3-15                   [1, 128, 128, 128]        --
│    └─BaseConv: 2-6                     [1, 256, 128, 128]        --
│    │    └─Conv2d: 3-16                 [1, 256, 128, 128]        294,912
│    │    └─BatchNorm2d: 3-17            [1, 256, 128, 128]        512
│    │    └─SiLU: 3-18                   [1, 256, 128, 128]        --
├─ConvTranspose2d: 1-6                   [1, 128, 256, 256]        131,200
├─Bottleneck: 1-7                        [1, 128, 256, 256]        --
│    └─BaseConv: 2-7                     [1, 64, 256, 256]         --
│    │    └─Conv2d: 3-19                 [1, 64, 256, 256]         16,384
│    │    └─BatchNorm2d: 3-20            [1, 64, 256, 256]         128
│    │    └─SiLU: 3-21                   [1, 64, 256, 256]         --
│    └─BaseConv: 2-8                     [1, 128, 256, 256]        --
│    │    └─Conv2d: 3-22                 [1, 128, 256, 256]        73,728
│    │    └─BatchNorm2d: 3-23            [1, 128, 256, 256]        256
│    │    └─SiLU: 3-24                   [1, 128, 256, 256]        --
├─ConvTranspose2d: 1-8                   [1, 64, 512, 512]         32,832
├─Bottleneck: 1-9                        [1, 64, 512, 512]         --
```

```
│    └─BaseConv: 2-9                     [1, 32, 512, 512]         --
│    │    └─Conv2d: 3-25                 [1, 32, 512, 512]         4,096
│    │    └─BatchNorm2d: 3-26            [1, 32, 512, 512]         64
│    │    └─SiLU: 3-27                   [1, 32, 512, 512]         --
│    └─BaseConv: 2-10                    [1, 64, 512, 512]         --
│    │    └─Conv2d: 3-28                 [1, 64, 512, 512]         18,432
│    │    └─BatchNorm2d: 3-29            [1, 64, 512, 512]         128
│    │    └─SiLU: 3-30                   [1, 64, 512, 512]         --
├─Conv2d: 1-10                          [1, 10, 512, 512]         650
==========================================================================================
Total params: 686,890
Trainable params: 686,890
Non-trainable params: 0
Total mult-adds (G): 44.24
==========================================================================================
Input size (MB): 3.15
Forward/backward pass size (MB): 1530.92
Params size (MB): 2.75
Estimated Total Size (MB): 1536.81
==========================================================================================

UNet5 Summary:
==========================================================================================
Layer (type:depth-idx)                  Output Shape              Param #
==========================================================================================
UNet5                                   [1, 10, 512, 512]         --
├─Bottleneck: 1-1                       [1, 64, 512, 512]         --
│    └─BaseConv: 2-1                    [1, 32, 512, 512]         --
│    │    └─Conv2d: 3-1                 [1, 32, 512, 512]         96
│    │    └─BatchNorm2d: 3-2            [1, 32, 512, 512]         64
│    │    └─SiLU: 3-3                   [1, 32, 512, 512]         --
│    └─BaseConv: 2-2                    [1, 64, 512, 512]         --
│    │    └─Conv2d: 3-4                 [1, 64, 512, 512]         18,432
│    │    └─BatchNorm2d: 3-5            [1, 64, 512, 512]         128
│    │    └─SiLU: 3-6                   [1, 64, 512, 512]         --
├─MaxPool2d: 1-2                        [1, 64, 256, 256]         --
├─Bottleneck: 1-3                       [1, 128, 256, 256]        --
│    └─BaseConv: 2-3                    [1, 64, 256, 256]         --
│    │    └─Conv2d: 3-7                 [1, 64, 256, 256]         4,096
│    │    └─BatchNorm2d: 3-8            [1, 64, 256, 256]         128
│    │    └─SiLU: 3-9                   [1, 64, 256, 256]         --
│    └─BaseConv: 2-4                    [1, 128, 256, 256]        --
│    │    └─Conv2d: 3-10                [1, 128, 256, 256]        73,728
│    │    └─BatchNorm2d: 3-11           [1, 128, 256, 256]        256
│    │    └─SiLU: 3-12                  [1, 128, 256, 256]        --
├─MaxPool2d: 1-4                        [1, 128, 128, 128]        --
```

```
├─Bottleneck: 1-5                        [1, 256, 128, 128]        --
│    └─BaseConv: 2-5                     [1, 128, 128, 128]        --
│    │    └─Conv2d: 3-13                 [1, 128, 128, 128]        16,384
│    │    └─BatchNorm2d: 3-14            [1, 128, 128, 128]        256
│    │    └─SiLU: 3-15                   [1, 128, 128, 128]        --
│    └─BaseConv: 2-6                     [1, 256, 128, 128]        --
│    │    └─Conv2d: 3-16                 [1, 256, 128, 128]        294,912
│    │    └─BatchNorm2d: 3-17            [1, 256, 128, 128]        512
│    │    └─SiLU: 3-18                   [1, 256, 128, 128]        --
├─MaxPool2d: 1-6                         [1, 256, 64, 64]          --
├─Bottleneck: 1-7                        [1, 512, 64, 64]          --
│    └─BaseConv: 2-7                     [1, 256, 64, 64]          --
│    │    └─Conv2d: 3-19                 [1, 256, 64, 64]          65,536
│    │    └─BatchNorm2d: 3-20            [1, 256, 64, 64]          512
│    │    └─SiLU: 3-21                   [1, 256, 64, 64]          --
│    └─BaseConv: 2-8                     [1, 512, 64, 64]          --
│    │    └─Conv2d: 3-22                 [1, 512, 64, 64]          1,179,648
│    │    └─BatchNorm2d: 3-23            [1, 512, 64, 64]          1,024
│    │    └─SiLU: 3-24                   [1, 512, 64, 64]          --
├─MaxPool2d: 1-8                         [1, 512, 32, 32]          --
├─Bottleneck: 1-9                        [1, 1024, 32, 32]         --
│    └─BaseConv: 2-9                     [1, 512, 32, 32]          --
│    │    └─Conv2d: 3-25                 [1, 512, 32, 32]          262,144
│    │    └─BatchNorm2d: 3-26            [1, 512, 32, 32]          1,024
│    │    └─SiLU: 3-27                   [1, 512, 32, 32]          --
│    └─BaseConv: 2-10                    [1, 1024, 32, 32]         --
│    │    └─Conv2d: 3-28                 [1, 1024, 32, 32]         4,718,592
│    │    └─BatchNorm2d: 3-29            [1, 1024, 32, 32]         2,048
│    │    └─SiLU: 3-30                   [1, 1024, 32, 32]         --
├─ConvTranspose2d: 1-10                  [1, 512, 64, 64]          2,097,664
├─Bottleneck: 1-11                       [1, 512, 64, 64]          --
│    └─BaseConv: 2-11                    [1, 256, 64, 64]          --
│    │    └─Conv2d: 3-31                 [1, 256, 64, 64]          262,144
│    │    └─BatchNorm2d: 3-32            [1, 256, 64, 64]          512
│    │    └─SiLU: 3-33                   [1, 256, 64, 64]          --
│    └─BaseConv: 2-12                    [1, 512, 64, 64]          --
│    │    └─Conv2d: 3-34                 [1, 512, 64, 64]          1,179,648
│    │    └─BatchNorm2d: 3-35            [1, 512, 64, 64]          1,024
│    │    └─SiLU: 3-36                   [1, 512, 64, 64]          --
├─ConvTranspose2d: 1-12                  [1, 256, 128, 128]        524,544
├─Bottleneck: 1-13                       [1, 256, 128, 128]        --
│    └─BaseConv: 2-13                    [1, 128, 128, 128]        --
│    │    └─Conv2d: 3-37                 [1, 128, 128, 128]        65,536
│    │    └─BatchNorm2d: 3-38            [1, 128, 128, 128]        256
│    │    └─SiLU: 3-39                   [1, 128, 128, 128]        --
│    └─BaseConv: 2-14                    [1, 256, 128, 128]        --
```

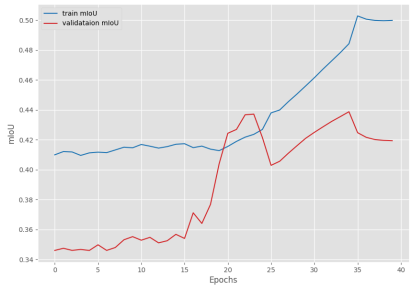
```
│    │    └─Conv2d: 3-40                      [1, 256, 128, 128]        294,912
│    │    └─BatchNorm2d: 3-41                 [1, 256, 128, 128]        512
│    │    └─SiLU: 3-42                        [1, 256, 128, 128]        --
├─ConvTranspose2d: 1-14                       [1, 128, 256, 256]        131,200
├─Bottleneck: 1-15                            [1, 128, 256, 256]        --
│    └─BaseConv: 2-15                         [1, 64, 256, 256]         --
│    │    └─Conv2d: 3-43                      [1, 64, 256, 256]         16,384
│    │    └─BatchNorm2d: 3-44                 [1, 64, 256, 256]         128
│    │    └─SiLU: 3-45                        [1, 64, 256, 256]         --
│    └─BaseConv: 2-16                         [1, 128, 256, 256]        --
│    │    └─Conv2d: 3-46                      [1, 128, 256, 256]        73,728
│    │    └─BatchNorm2d: 3-47                 [1, 128, 256, 256]        256
│    │    └─SiLU: 3-48                        [1, 128, 256, 256]        --
├─ConvTranspose2d: 1-16                       [1, 64, 512, 512]         32,832
├─Bottleneck: 1-17                            [1, 64, 512, 512]         --
│    └─BaseConv: 2-17                         [1, 32, 512, 512]         --
│    │    └─Conv2d: 3-49                      [1, 32, 512, 512]         4,096
│    │    └─BatchNorm2d: 3-50                 [1, 32, 512, 512]         64
│    │    └─SiLU: 3-51                        [1, 32, 512, 512]         --
│    └─BaseConv: 2-18                         [1, 64, 512, 512]         --
│    │    └─Conv2d: 3-52                      [1, 64, 512, 512]         18,432
│    │    └─BatchNorm2d: 3-53                 [1, 64, 512, 512]         128
│    │    └─SiLU: 3-54                        [1, 64, 512, 512]         --
├─Conv2d: 1-18                                [1, 10, 512, 512]         650
==========================================================================================
Total params: 11,344,170
Trainable params: 11,344,170
Non-trainable params: 0
Total mult-adds (G): 83.44
==========================================================================================
Input size (MB): 3.15
Forward/backward pass size (MB): 1807.75
Params size (MB): 45.38
Estimated Total Size (MB): 1856.27
==========================================================================================
```

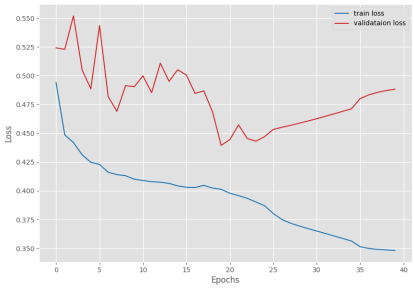
3.加入多个数据增广函数（旋转，翻转等），试一试效果。

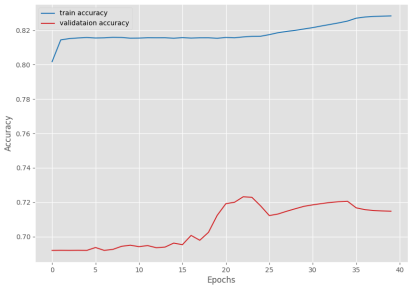
这里引入水平翻转、垂直翻转、旋转，并使用`UNet3()`进行训练

```
train_image_transform = transforms.Compose(
    [
        transforms.Resize([img_size, img_size]),
        transforms.RandomHorizontalFlip(p=0.5),
        transforms.RandomRotation(degrees=30),
        transforms.RandomVerticalFlip(p=0.5),
        transforms.ToTensor(),
    ]
)
```

这里发现引入数据增强后性能略有下降，且训练过程震荡较大

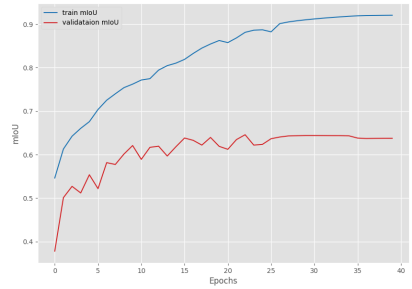


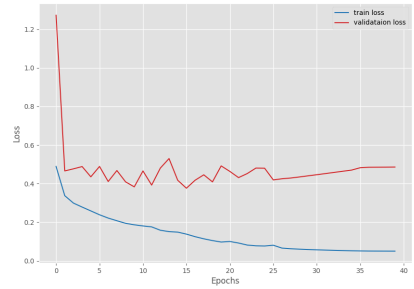


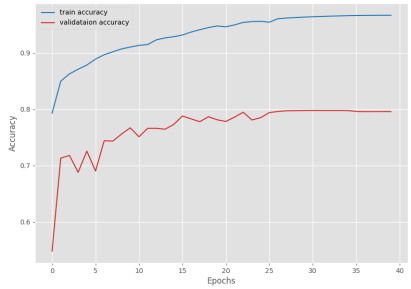


4.用 UNet3() 和 UNet5(), 分别试验一下分割效果，并分析结果。

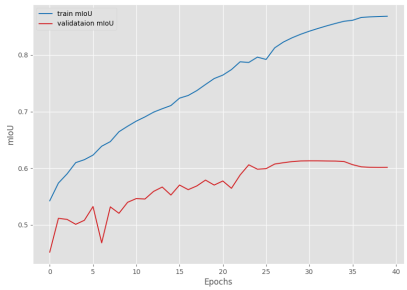
`UNet5()`

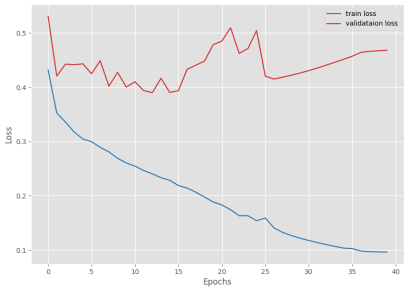


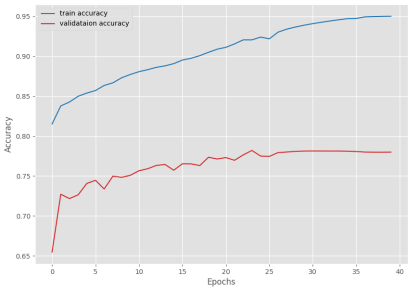




`UNet3()`








可以看到模型在经过训练后专注于目标的能力有了一些提升，不过由于本身性能较低，训练后人物的上半身完全没有被识别到，并且`UNet5()`的性能较`UNet3()`更好（网络深度影响）。

5.附加：将 UNet5 改造成 UNet++, 试一下效果

搓了一个“假冒伪劣”的小玩具，似乎不太符合`UNet++`的标准做法，在训练集上的效果似乎略有提升，但验证集依然过拟合，且性能没有太大提升