

U-Net 训练与测试实验报告

吴清源 于 2023.11.10

一、文件夹及文件介绍

1. `.git`、`.gitattributes`: 保留脚本的历史修改记录。
2. `.idea`、`__pycache__`: pycharm 系统文件。
3. `32_32_图集`: 像素分辨率为 32×32 的训练、验证与测试集。使用此集合训练只需极少的训练时间, 可快速检查模型的正确性。
4. `512_512_图集`: 像素分辨率为 512×512 的训练、验证与测试集。使此集合训练 100 轮需要 25 小时。
5. `models`: 保存模型参数。
6. `models_manual_save`: 手动保存的模型参数。
7. `results`: 储存测试结果及训练过程中损失值等量化参数的 CVS 文件。
8. `test`, `test_GT`, `train`, `train_GT`, `valid`, `valid_GT`: 测试、训练和验证集, 其中 `_GT` 是 Ground Truth。
9. `data_loader.py`: 内有类 `ImageFolder` 和函数 `get_loader`。其中, `ImageFolder` 用来对训练集进行图像增强; `get_loader` 用来加载数据集。
10. `evaluation.py`: 内有函数 `get_accuracy`、`get_sensitivity`、`get_specificity`、`get_precision`、`get_F1`、`get_JS`、`get_DC`。这些函数用来评估模型的输出结果与 GT 之间的差别, 反应模型评估性能的好坏。(待优化)
11. `model.py`: 内有类 `conv_block`、`up_conv` 和 `U_net`。其中, `conv_block` 是一些卷积层的集合; `up_conv` 是上采样+卷积层的集合; `U_net` 是网络模型。
12. `RUN_ME.py`: 主函数, 在里面设置网络的各种参数; 控制训练和测试类型。
13. `solver.py`: 内有类 `Solver`。它用来加载数据集并训练网络。
14. `temp.py`、`temp2.py`、`test.py`: 一些测试用的脚本。

二、使用方法

1. 将训练、验证和测试集将其 GT 放在对应的文件夹中。注意：像素数越大，训练的时间越长。

2. 打开 RUN_ME.py, 在 if __name__ == '__main__': 中修改网络所需的训练参数。

`parser.add_argument('--image_channels', type=int, default=3)` 输入图像的通道数，若是 RGB 图像，则通道数为 3。

`parser.add_argument('--output_channels', type=int, default=1)` 模型的输出通道数，一般为 1，根据使用的模型以及任务修改。

`parser.add_argument('--num_epoch', type=int, default=100)` 模型训练的次数

`parser.add_argument('--num_epoch_decay', type=int, default=40)` 通常用于控制学习率在训练深度学习模型时的调整策略。它表示在经过多少个训练周期（epochs）后，要减小学习率的值。

`parser.add_argument('--batch_size', type=int, default=1)` batch size 指定同时有 default 个数据一起训练。注意：如果使用 data_loader.py 的图像增强方法，则 batch_size 必须设置为 1。因为该图像增强方法中存在 random.randint()，它会裁剪图像，使得各个 batch_size 的维度不同，进而报错。

`parser.add_argument('--num_workers', type=int, default=8)` 根据 CPU 的核心数修改。

`parser.add_argument('--lr', type=float, default=0.0002)` 学习率

`parser.add_argument('--beta1', type=float, default=0.5)` 优化函数需要的参数 1

`parser.add_argument('--beta2', type=float, default=0.999)` 优化函数需要的参数 2

`parser.add_argument('--augmentation_prob', type=float, default=0.4)` 一个参数，用来控制图像增强发生的概率。越接近 1，则越容易发生图像增强。

`parser.add_argument('--log_step', type=int, default=2)` 没有使用

`parser.add_argument('--val_step', type=int, default=2)` 没有使用

`parser.add_argument('--mode', type=str, default='test', help='train,test')` 指定是训练模式，还是测试模式。

`parser.add_argument('--model_path', type=str, default='D:\Repositories/U-Net/models')` 保存或读取模型的地址。

`parser.add_argument('--model_type',type=str, default='U_net')` 指定模型结构。

`parser.add_argument('--train_path', type=str, default='D:\Repositories/U-Net/train/')` 读取训练集地址。

`parser.add_argument('--valid_path', type=str, default='D:\Repositories/U-Net/valid/')` 读取验证集地址。

`parser.add_argument('--test_path', type=str, default='D:\Repositories/U-Net/test/')` 读取测试集地址。

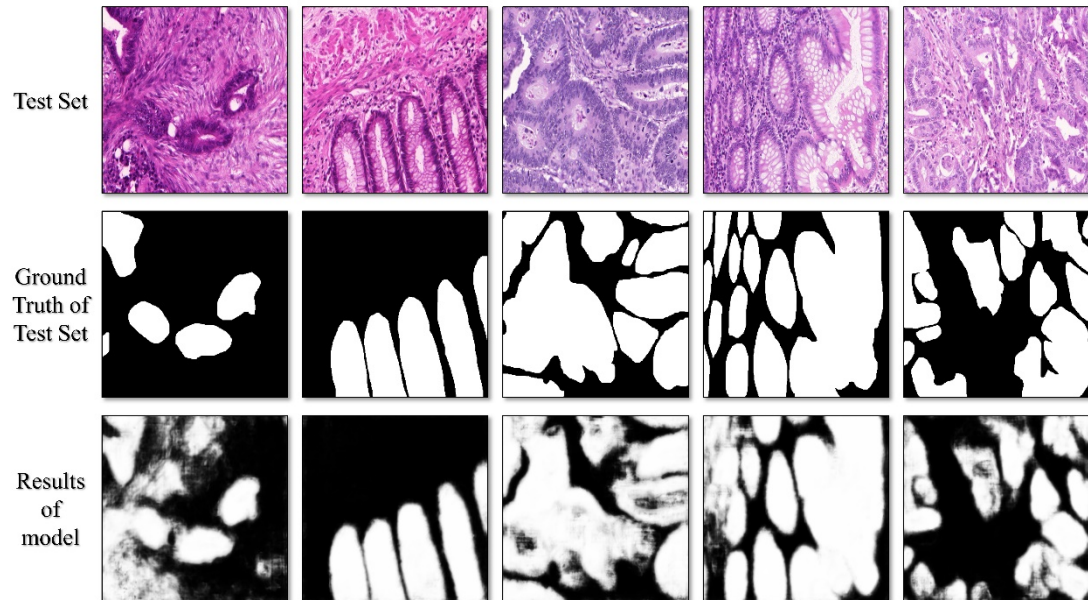
`parser.add_argument('--result_path', type=str, default='D:\Repositories/U-Net/results/')` 保存测试结果及 CSV 地址。

`parser.add_argument('--cuda_idx', type=int, default=1)` 指定是否用 GPU 加速计算。

三、两个模型的测试结果

1. 模型 1. 训练 100 轮, 学习率: 0.0002, num_epoch_decay: 40, augmentation_prob:

0.4



2. 模型 2. 训练 300 轮, 学习率: 0.0002, num_epoch_decay: 250, augmentation_prob:

0.4

