User's guide of FID-STORM

Zhiwei Zhou,[1] Junnan Wu,[3] Zhengxia Wang,[2,4] and Zhen-Li Huang[3,5]

[1]*Britton Chance Center and MoE Key Laboratory for Biomedical Photonics, School of Engineering Sciences, Wuhan National Laboratory for Optoelectronics-Huazhong University of Science and Technology, Wuhan 430074, China*
[2]*School of Computer Science and Technology, Hainan University, Haikou 570228, China*
[3]*Key Laboratory of Biomedical Engineering of Hainan Province, School of Biomedical Engineering, Hainan University, Haikou 570228, China*
[4]*zxiawang@hainanu.edu.cn*
[5]*Huang2020@hainan.edu.cn*

1. Introduction

FID-STORM 是一个基于深度学习用于在线处理原始图像的方法，现阶段，该方法能对256x256 pixels @ 10 ms 曝光时间下的原始图像进行实时处理。

为了使用好该方法，我们分三部分对该方法的使用步骤进行描述：第一，怎么去安装使用环境；第二，怎么去训练模型；第三，基于训练好的模型，怎么使用推理代码去作推理。

2. How to make an inference based on ImageJ plugin

2.1 Environment preparation for ImageJ plugin

1) Down load necessary package

Table 1. List of running environment

Environments	Download link
Windows 10 x64	https://www.microsoft.com/en-us/windows?wa=wsignin1.0
visual studio 2017	https://visualstudio.microsoft.com/zh-hans/
cuda_11.6	https://developer.nvidia.com/cuda-toolkit-archive
cudnn8.4	https://developer.nvidia.com/rdp/cudnn-archive
tensorRT-8.4.2.4	https://developer.nvidia.com/nvidia-tensorrt-download

2) Install viusal studio 2017 on 64-bit windows 10

3) Install cuda11.6 on 64-bit windows 10, and add the bin library in the cuda11.6 installation directory to the PATH variable of the system environment variable.

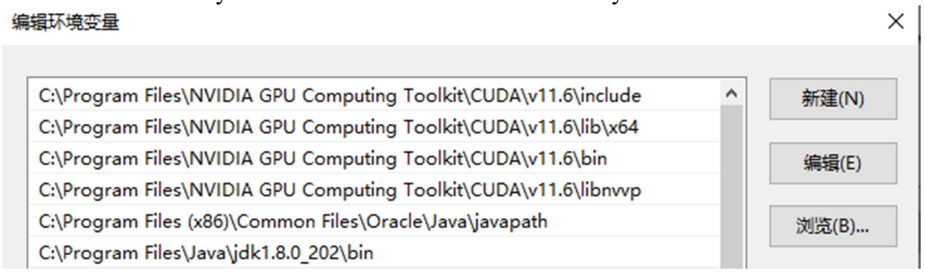

Figure 1. Environment settings of cuda 11.6

4) Install cudnn8.4 on 64-bit windows 10, and copy the files of bin, lib and include library in the cudnn 8.4 to the bin, lib, and include directory of cuda 11.6 respectively.

5) Install cudnn8.4 on 64-bit windows 10, and copy the files of bin, lib and include library in the tensorRT-8.4.2.4 to the bin, lib, and include directory of cuda 11.6 respectively.

Notice: When the program is running, there may be problems that dll can not be loaded. You can try to copy zlibwapi.dll to bin library of cuda 11.6. The zlibwapi.dll can be downloaded at https://www.dll-files.com/zlibwapi.dll.html.

2.2 Network inference of FID-STORM in ImageJ plugin

Now you can do a inference based on our ImageJ plugin after all running environment are installed.

1) 打开 FID-STORM ImageJ 插件，点击 plugin->FID-STORM，得到 FID-STORM 的 GUI 界面, 如图所示。

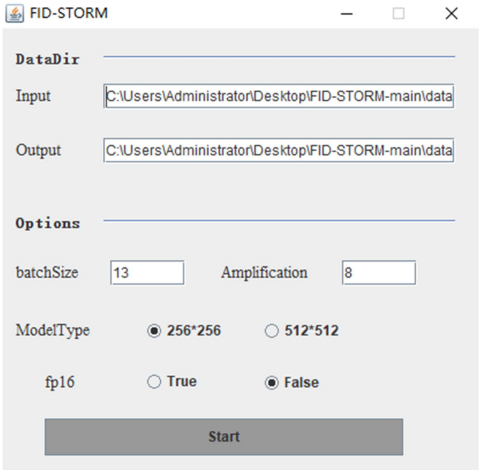

Figure 2. The GUI of FID-STORM

2) DataDir 中的参数为输入、输出数据的路径，Input 文件夹中存放原始图像、.onnx 模型文件，Output 文件夹中保存执行结果

3) 设置好 GUI 中的相关参数

Table 2. The corresponding parameters in GUI

Items	Description
Input	The index address of the folder where the raw image is located.
Output	The index address of the results folder.
batchSize	The number of samples selected for one training. It affects the optimization degree and speed of the model.
Amplification	Amplification of the raw images, default 8.
ModelType	Deep learning models of different sizes, including 256*256，512*512.
Fp16	Indicates the type of the .trt “false” indicates 32 bits and “true” indicates 16 bits.

4) 点击 start 开始执行程序

5) 如果程序正常运行，你将会看到推理过程、时间等窗口界面，如图。

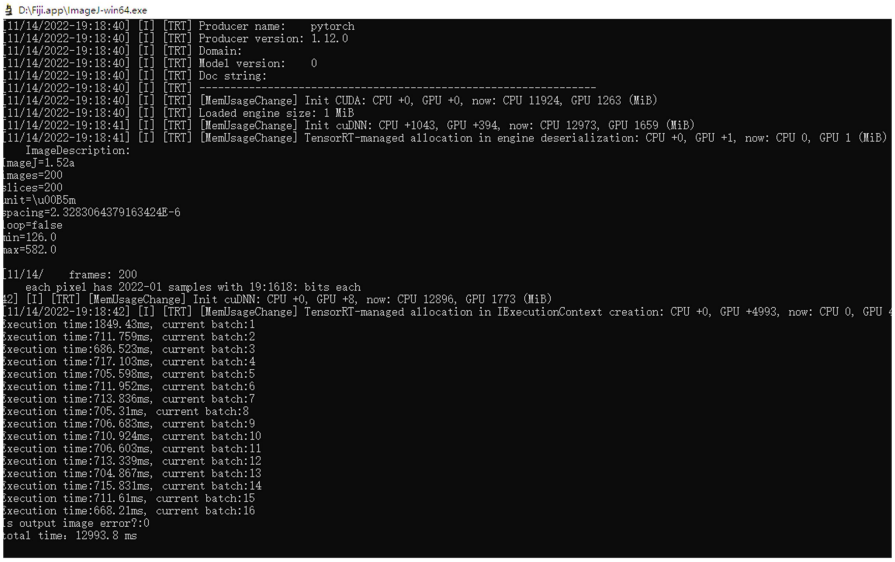

Figure 3. 程序运行的日志打印

6) 能在 output 文件夹下获得一张推理的超分辨图像

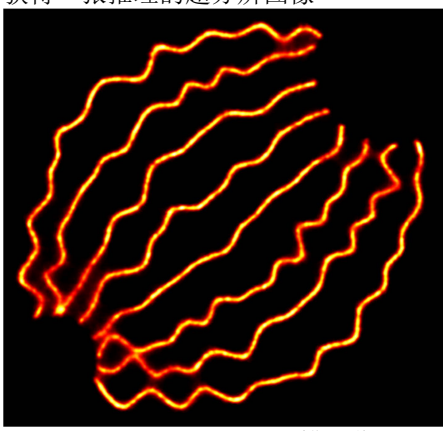

Figure 4. 重建的超分辨图像

3. How to train model based on the data sets

3.1 Environment preparation for model training

我们推荐使用 ANACONDA（https://www.anaconda.com/ ）去管理 python 虚拟环境，并基于 pycharm 去编写，编译及调试代码。

1) 先基于 ANNACONDA 去新建 python 虚拟环境，环境所需库包含：

- Python 3.8
- Numpy
- Pytorch 1.12.0
- Opencv

2) 使用 pycharm 去打开 FID-STORM python 工程，并选择好虚拟环境新建的虚拟环境

3.2 The generation of Training data sets

1) 打开 python\demo\dataset\dataPrepare\GenerateTrainingData_fromQC_STORM_main.m

2) 设置好 parameters setting 下的相关参数

- Datapath : data path
- ouverlapFactor : the nums of raw images that are overlaped
- density : filter, the density below 1.0 of raw image will be remove
- camera_pixelsize: camera pixel size in [nm]
- upsampling_factor : upsampling factor, raw image will be upsampled x(factor) times
- kernelSize : kernel size
- Gaussian_sigma : using for heatmap，standard error, unit is pixels

3) 开始执行，执行结果会在指定的 datapath 路径下生成 result 文件夹，文件夹中存放着 HeatmapImg 和 rawImgUp，HeatmapImg 文件夹存放着训练用真实图像、rawImg 文件夹存放着训练用的原始图像。

Figure 5. 数据文件夹

3.3 Training

1) 打开 train_ours.py，设置好以下参数：

- rawImgPath,代表生成的训练数据
- savePath, 代表训练生成的模型路径
- saveTestPath, 代表训练过程中测试图像保存路径
- EPOCH, 代表在训练数据上迭代次数
- lr, 代表学习率

2) 设置好参数后，执行 train_ours.py,开始训练；

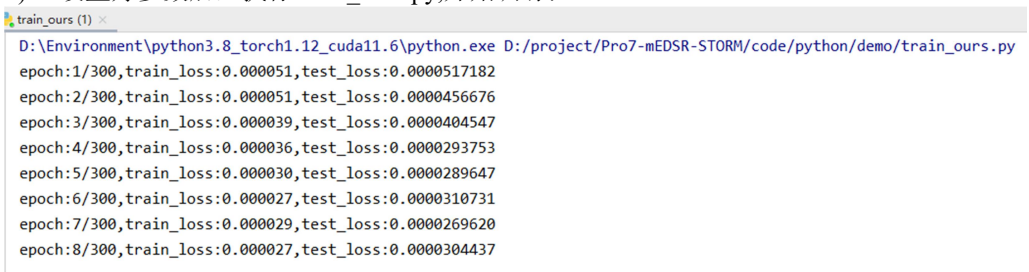

Figure 6. 训练日志

3) 训练完成后的模型被保存到了参数 savePath 指定的文件夹。

模型保存格式为.pkl，其中 best.pkl 为损失值最小的模型；trainLoss_CNN.npy 和 testLoss_CNN.npy 分别存放着训练和测试损失；训练结束后，可以看到训练和验证损失函数的曲线，下图为 30 个 epoch 的训练和测试损失曲线图。

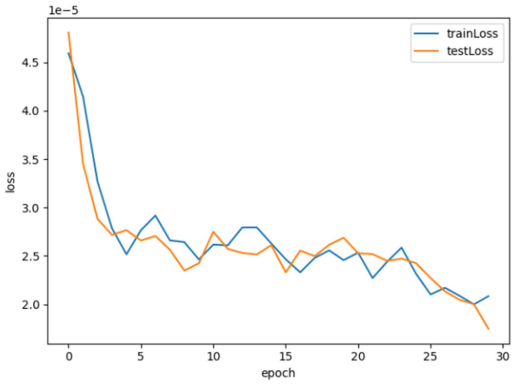

Figure 7. 训练损失和测试损失

3.4 Testing

1) 打开 test_ours.py

2) 设置好以下参数

- rawImgPath: raw image directory
- modelPath: model path, best.pkl will be loaded
- savePath: save path, output images and timeList will be saved
- subDir: sub directory, using for saving output images

3) 执行代码，推理图像将被保存在 trainingResult\output 文件夹下

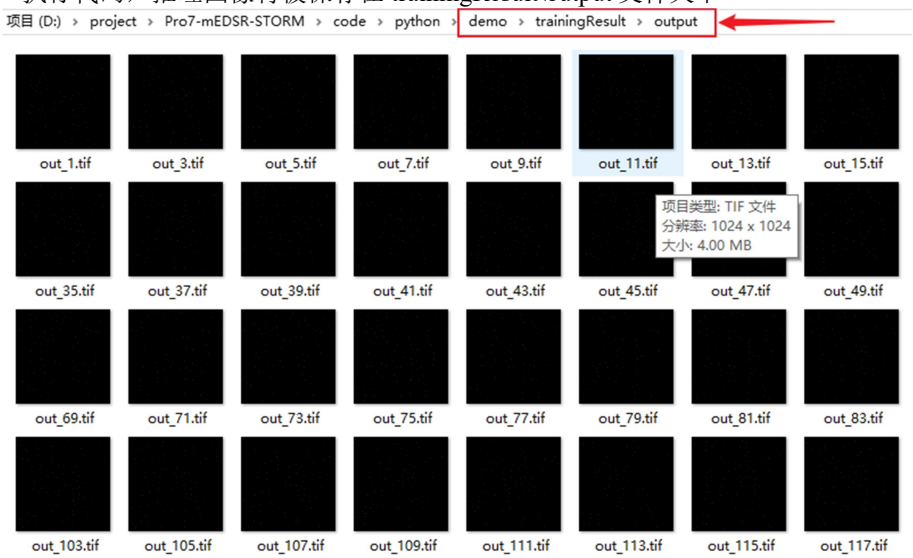

Figure 8. 推理后的图像输出

3.5 Converting trained model into .onnx format

1) 打开 onnxConvert\pytorchToOnnx.py，设置好两个参数：

- modelName：训练好的模型名称
- shape：导出模型的输入限制尺寸

2) 运行，导出 onnx 模型。下图为导出的 4 种不同尺寸的 onnx 模型。

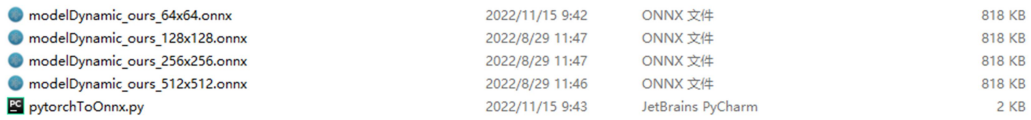

Figure 9. 转换为 onnx 格式后的模型输出

4. How to debug the proposed tensorRT code

4.1 Environment preparation for debug in visual studio 2017

如果想对基于 TensorRT 的模型推理代码进行调试，可以安装以下环境：

- Visual studio 2017 community
- Cuda 11.6
- Cudnn 8.4
- tensorRT-8.4.2.4

Notice: The detailed description can be seen in Section 2.1

4.2 Modifications of the configuration in visual studio 2017

1) 使用 Visual studio 2017 打开 FID-STORM，修改项目->配置属性为应用程序(exe);

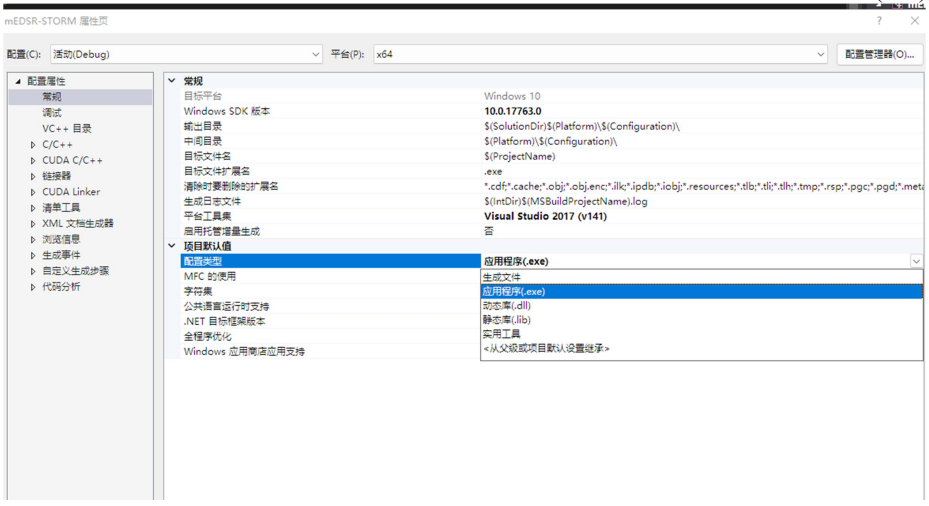

Figure 10. Visual studio 2017 的工程设置

2) 修改 main_test.cpp 文件中的相关参数

- inputDataDir: 原始图像和训练模型所在文件夹
- outputDataDir: 用于存放输出结果的文件夹
- fileName: 原始图像名称
- batchSize: 一次训练选取的样本数量
- fp16: .trt 文件类型，true 表示 16 位，false 表示 32 位
- modelType: 原始图像尺寸
- scaleFactor: 图像放大倍数

3) 运行代码， 相关日志打印见 Fig 11.

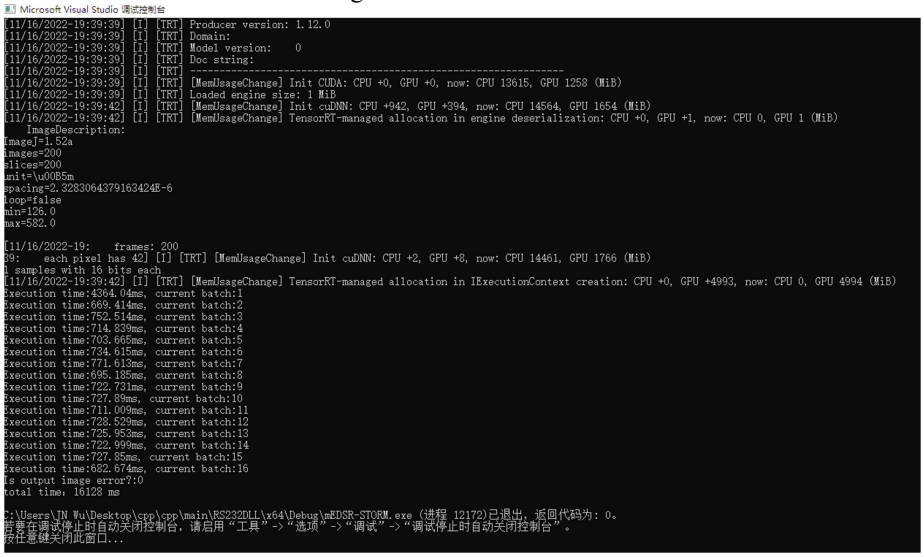

Figure 11. 日志输出

4) 推理后的重建图像将保存在 outputDataDir 参数的路径中。

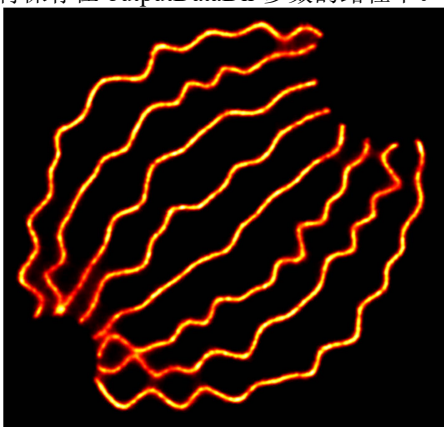

Figure 12. 重建图像