User’s guide of mEDSR-STORM

November 13，2022

# Description

mEDSR-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细节进行描述：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。

# How to install

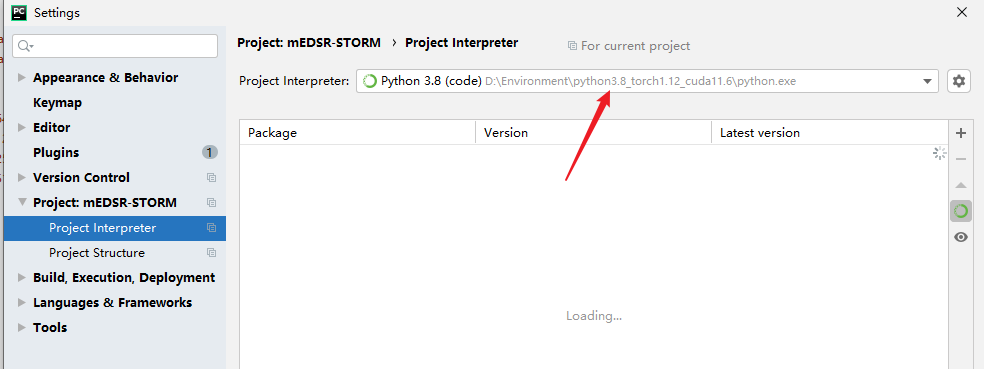
## 安装训练环境

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
* Cuda 11.6
* Opencv

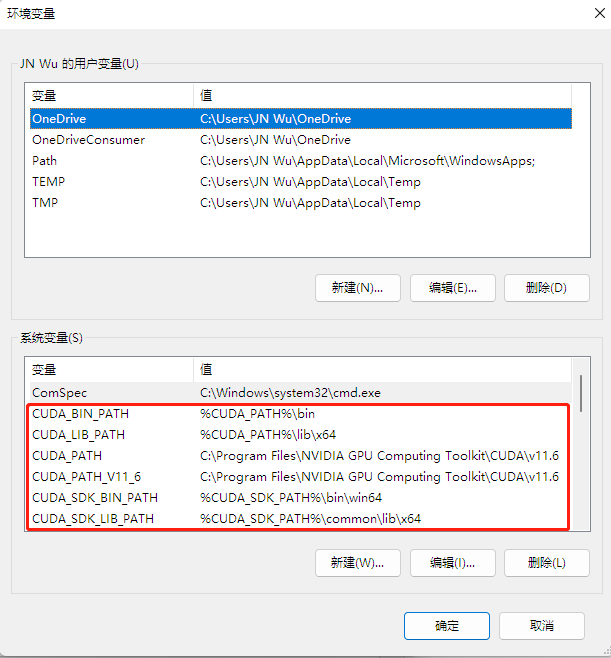
1. 使用pycharm去打开mEDSR-STORM python工程，并选择好虚拟环境新建的虚拟环境



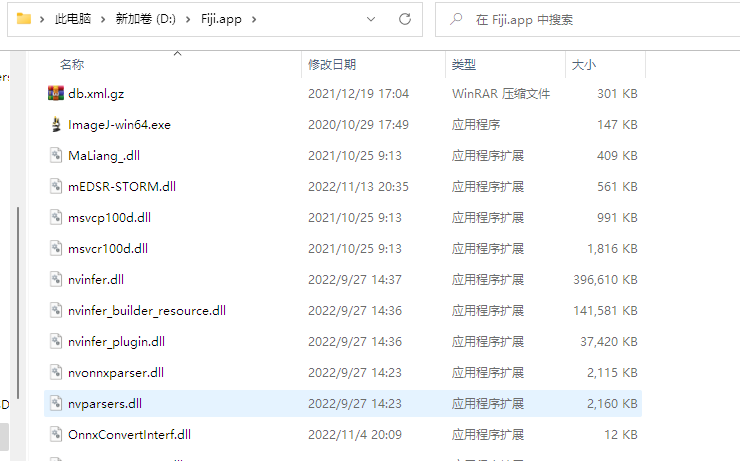
## 安装模型推理ImageJ运行环境

mEDSR-STORM 被构建为一个在ImageJ中处理图像的插件。对于该插件的安装过程主要分为以下几步：

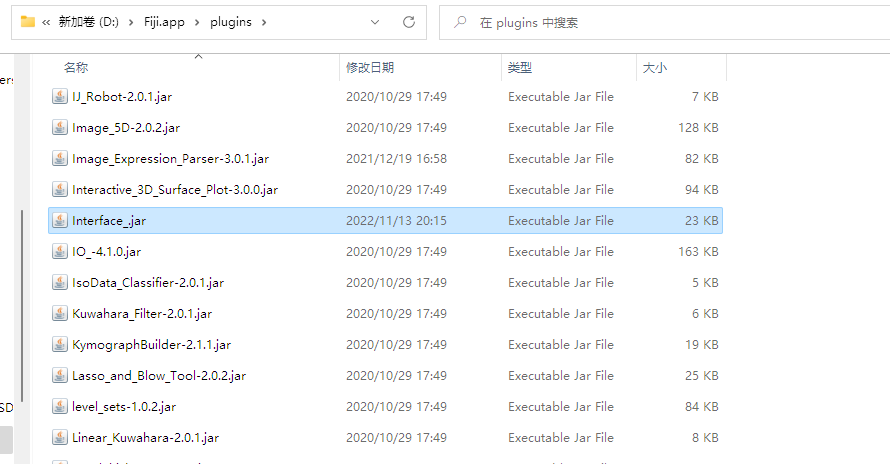
Ⅰ. 正确安装CUDA11.6并配置环境变量。



Ⅱ. 将所需的动态库.dll文件复制到ImageJ的安装目录下，包括nvinfer.dll、nvinfer\_builder\_resource.dll、nvinfer\_plugin.dll、nvonnxparser.dll、nvparsers.dll以及mEDSR-STORM.dll。



Ⅲ. 将interface.jar文件放到ImageJ的plugins目录下



## 安装模型推理代码调试环境

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

# How to train our model

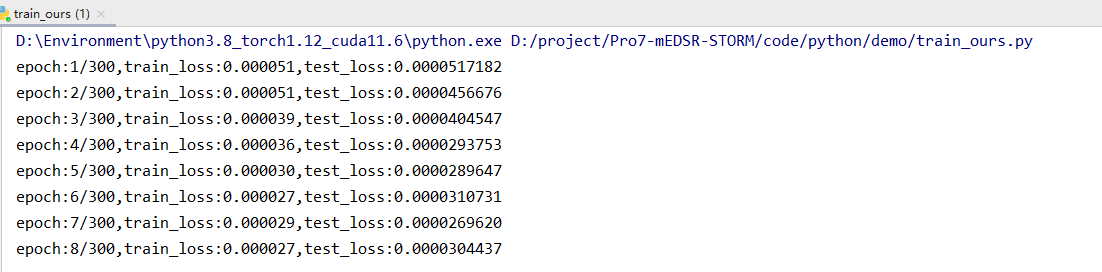
## 生成训练数据

1. 打开python\demo\dataset\dataPrepare\GenerateTrainingData\_fromQC\_STORM\_main.m
2. 设置好parameters setting下的相关参数
3. Datapath : data path
4. ouverlapFactor : the nums of raw images that are overlaped
5. density : filter, the density below 1.0 of raw image will be remove
6. camera\_pixelsize: camera pixel size in [nm]
7. upsampling\_factor : upsampling factor, raw image will be upsampled x(factor) times
8. kernelSize : kernel size
9. Gaussian\_sigma : using for heatmap，standard error, unit is pixels
10. 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。



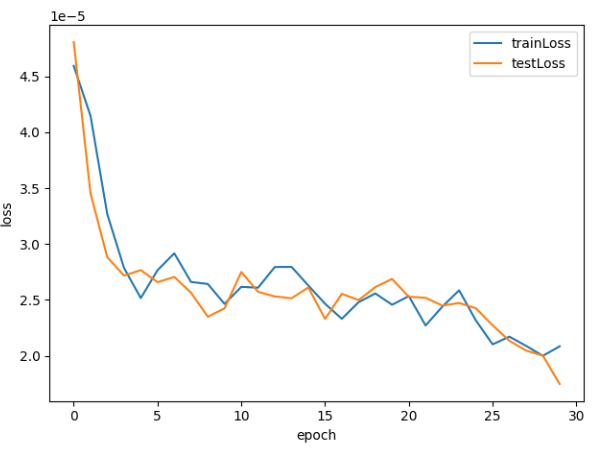
## 训练mEDSR-STORM模型

1. 打开train\_ours.py，设置好以下参数：
2. rawImgPath,代表生成的训练数据
3. savePath, 代表训练生成的模型路径
4. saveTestPath, 代表训练过程中测试图像保存路径
5. EPOCH, 代表在训练数据上迭代次数
6. lr, 代表学习率
7. 设置好参数后，执行train\_ours.py,开始训练；



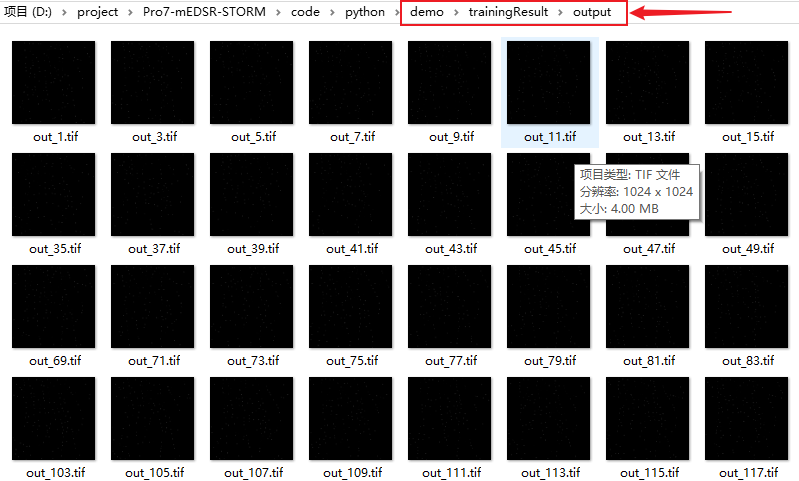
1. 训练完成后的模型被保存到了参数savePath指定的文件夹。

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。



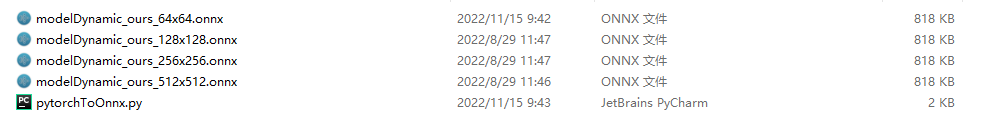
## 测试mEDSR-STORM模型

1. 打开test\_ours.py
2. 设置好以下参数
3. rawImgPath: raw image directory
4. modelPath: model path, best.pkl will be loaded
5. savePath: save path, output images and timeList will be saved
6. subDir: sub directory, using for saving output images
7. 执行代码，推理图像将被保存在trainingResult\output文件夹下



## 使用pytorchToOnnx.py转换为.onnx模型

1. 打开onnxConvert\pytorchToOnnx.py，设置好两个参数：
2. modelName：训练好的模型名称
3. shape：导出模型的输入限制尺寸
4. 运行，导出onnx模型。下图为导出的4种不同尺寸的onnx模型。



# How to infer our super-resolution image based our trained model

## 使用mEDSR-STORM的ImageJ plugin推理

1. 打开mEDSR-STORM ImageJ插件，点击plugin->mEDSR-STORM，得到mEDSR-STORM的GUI界面, 如图所示.

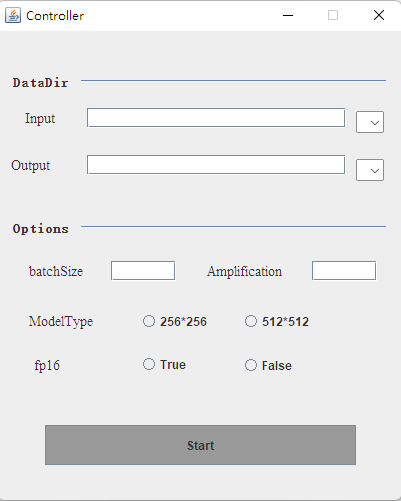


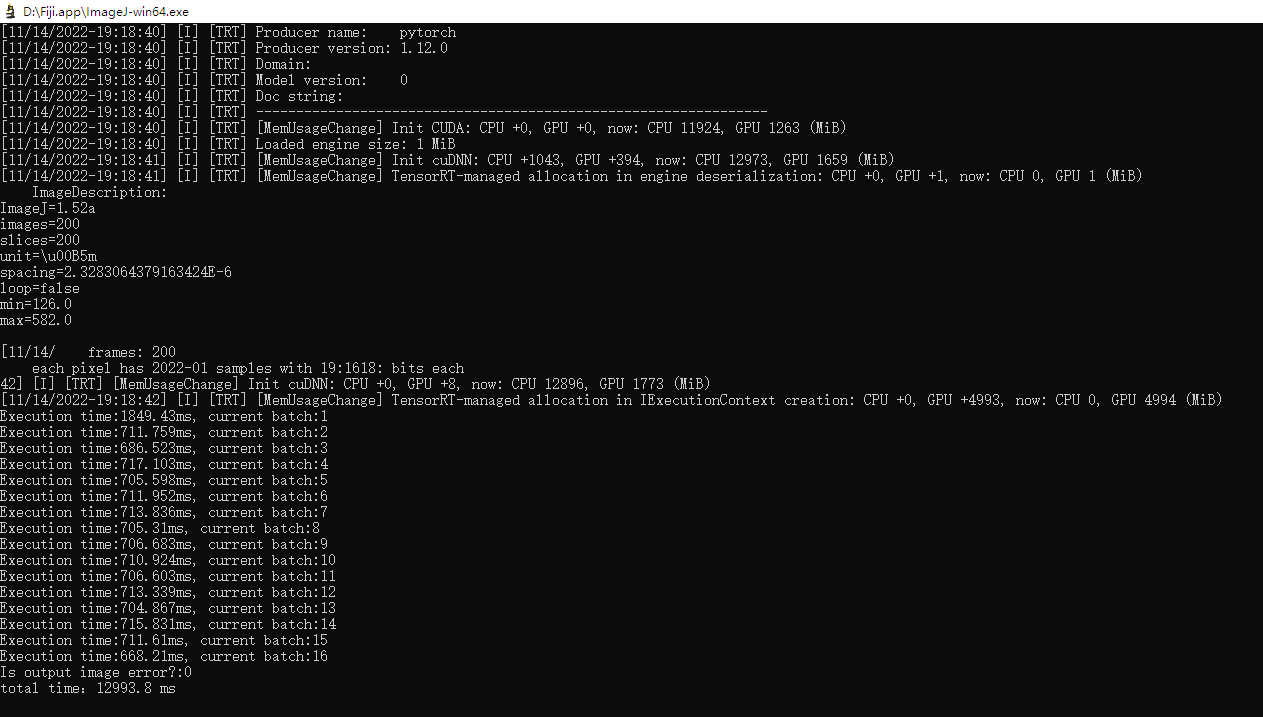
Figure 1. GUI interface of mEDSR-STORM

1. 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
2. 设置好GUI中的相关参数

|  |  |
| --- | --- |
| Item | Description |
| Input | The index address of the folder where the raw image is located. (eg: D:\TensorRT\raw\_image) |
| Output | The index address of the results folder.  (eg: D:\TensorRT\results) |
| 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. |

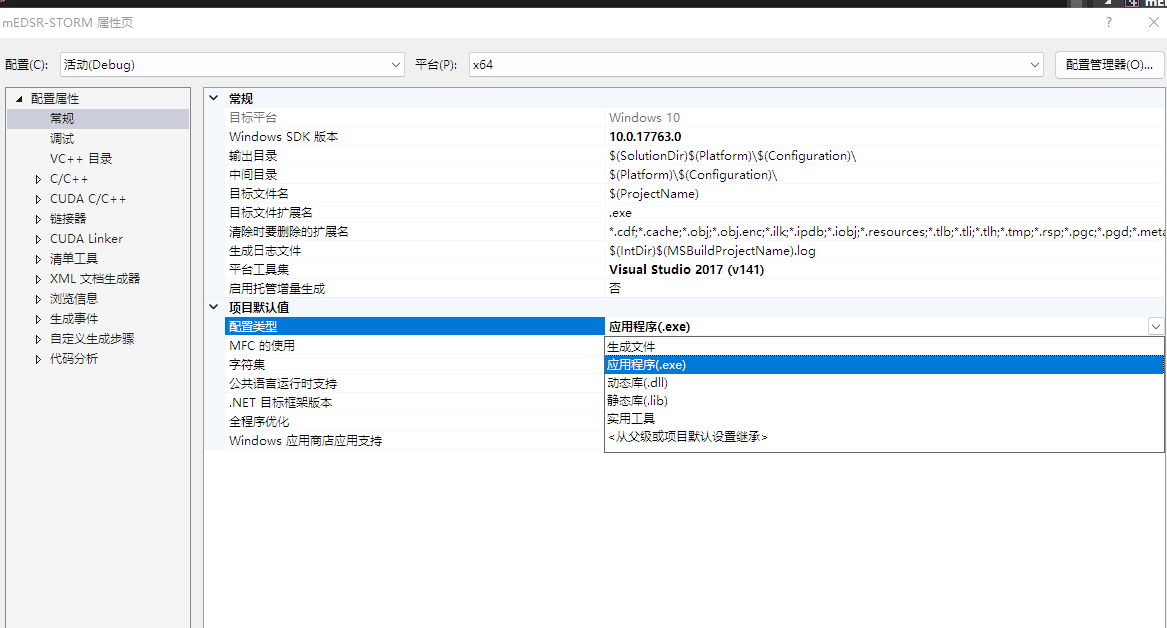
1. 点击start开始执行程序

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。



## 4.2 使用mEDSR-STORM的c++ code推理

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



2. 修改main\_test.cpp文件中的相关参数

A．inputDataDir: 原始图像和训练模型所在文件夹

B．outputDataDir: 用于存放输出结果的文件夹

C．fileName: 原始图像名称

D．batchSize: 一次训练选取的样本数量

E．fp16: .trt文件类型，true表示16位，false表示32位

F．modelType: 原始图像尺寸

G．scaleFactor: 图像放大倍数

3. 运行代码，结果将保存在outputDataDir参数的路径中。

