

Code reproduction documentation

Author: Ting Liu

Catalogue

1. Software and library requirements	1
2. Download software	1
2.1. MATLAB	1
2.2. Matconvnet	1
2.3. Visual Studio	2
2.4. CUDA	2
2.5. CUDNN	2
3. Experimental environment construction process	3
3.1 CPU compilation	3
3.2 GPU compilation	7
4. Run the WSWTNN-PnP method code	16
4.1 Code download:	16
4.2 Run Demo_WSWTNN_PnP	16

1. Software and library requirements

The following software needs to be installed before running Demo_WSWTNN_PnP:

Matlab 2018a

Matconvnet

Visual Studio 2015

CUDA 9.0

CUDNN v7.6.5

Computer system: Windows 10, Intel Core i7-10870H CPU (2.20 GHz), RTX 2060

(Note: For computers with different performance and different versions of matlab, the CUDA version should be different. You need to select the version of CUDA and CUDNN according to your computer.)

2. Download software

2.1. MATLAB

MATLAB 2018a: https://pan.baidu.com/s/1Rl2pv0cGv27skG_02Dtjgw
[key]: z6so

2.2. Matconvnet

Matconvnet: <https://www.vlfeat.org/matconvnet/>

Place the matconvnet-1.0-beta25 file under MATLAB/R2018a/toobox in the MATLAB 2018a installation location as shown in Fig. 2.1 below:

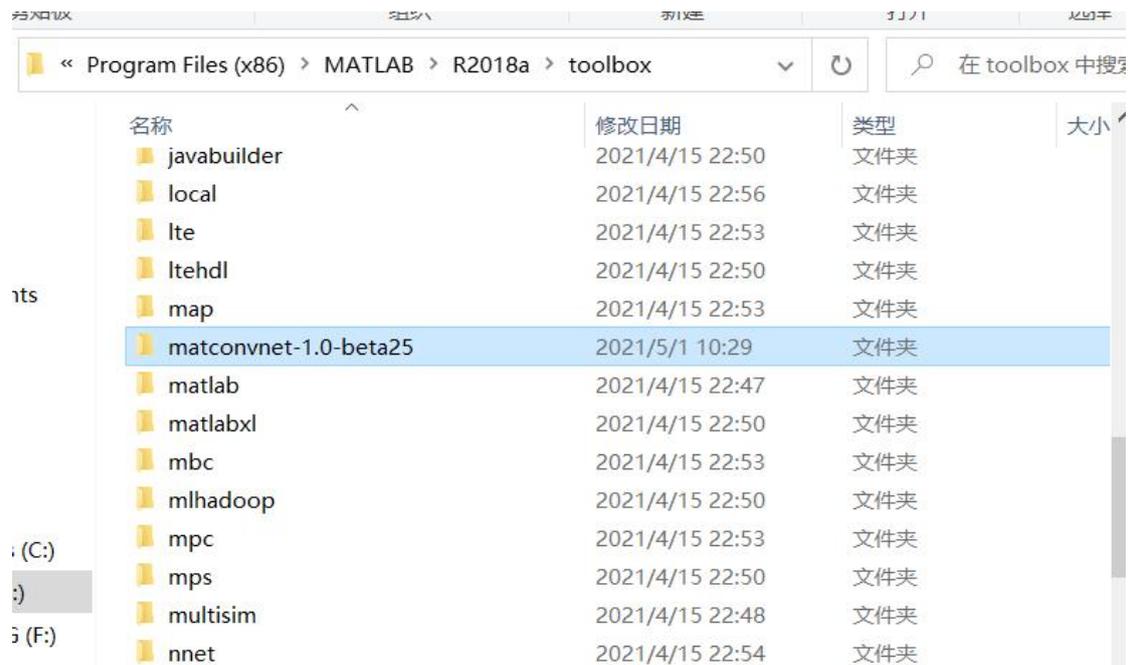


Figure 2.1

2.3. Visual Studio

Visual Studio 2015:

<https://pan.baidu.com/s/1bkEexqA8rMiEaTcYkcQdfg>
[key]: 5pd5

2.4. CUDA

CUDA 9.0

<https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html>

The steps for configuring CUDA are as follows:

<https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html>

2.5. CUDNN

CUDNN v7.6.5: <https://developer.nvidia.com/rdp/cudnn-archive>

The steps for configuring CUDNN are as follows:

<https://docs.nvidia.com/deeplearning/cudnn/install-guide/index.html#install-windows>

3. Experimental environment construction process

3.1 CPU compilation

(1) Add `matconvnet-1.0-beta25` and its sub files to `D:\Program Files (x86)\MATLAB\R2018a\toolbox`

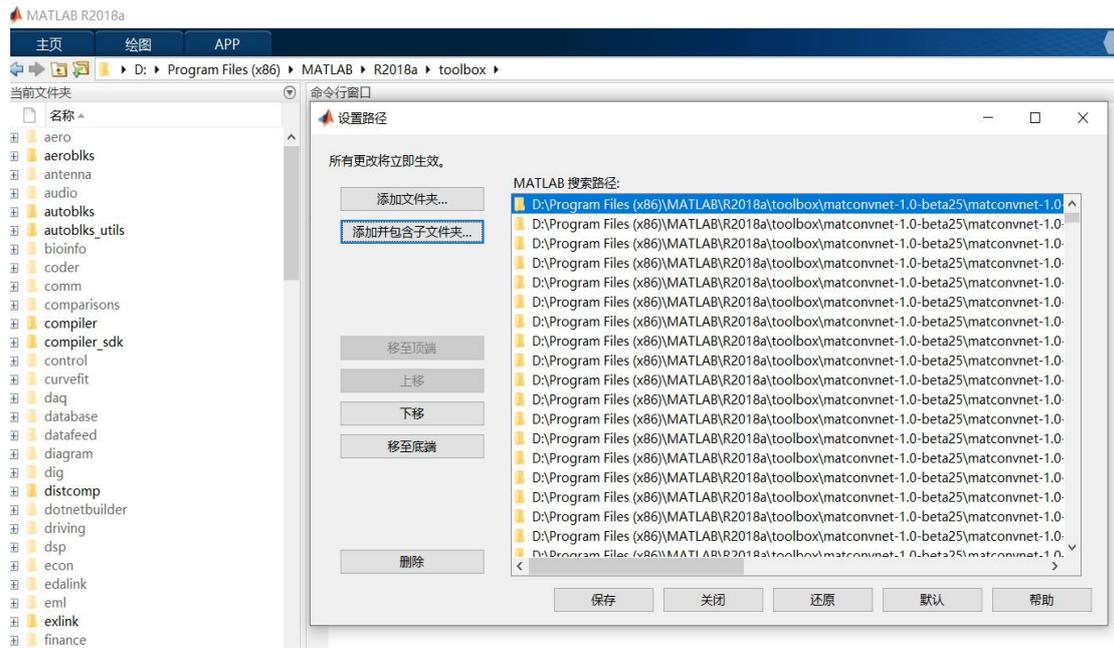


Figure 3.1

(2) Enter `mex-setup` in the MATLAB command line window to see if `mex-setup C++` appears. If no `mex-setup C++` appears, Visual Studio 2015 is not successfully installed

```

>> clear all
>> mex -setup
MEX 配置为使用 'Microsoft Visual C++ 2015 (C)' 以进行 C 语言编译。
警告: MATLAB C 和 Fortran API 已更改, 现可支持
包含 2^32-1 个以上元素的 MATLAB 变量。您需要
更新代码以利用新的 API。
您可以在以下网址找到更多的相关信息:
https://www.mathworks.com/help/matlab/matlab\_external/upgrading-mex-files-to-use-64-bit-api.html。

要选择不同的语言, 请从以下选项中选择一种命令:
mex -setup C++
mex -setup FORTRAN
MEX 配置为使用 'Microsoft Visual C++ 2015' 以进行 C++ 语言编译。
警告: MATLAB C 和 Fortran API 已更改, 现可支持
包含 2^32-1 个以上元素的 MATLAB 变量。您需要
更新代码以利用新的 API。
您可以在以下网址找到更多的相关信息:
https://www.mathworks.com/help/matlab/matlab\_external/upgrading-mex-files-to-use-64-bit-api.html。
MEX 配置为使用 'Microsoft Visual C++ 2015' 以进行 C++ 语言编译。
警告: MATLAB C 和 Fortran API 已更改, 现可支持
包含 2^32-1 个以上元素的 MATLAB 变量。您需要
更新代码以利用新的 API。
您可以在以下网址找到更多的相关信息:
https://www.mathworks.com/help/matlab/matlab\_external/upgrading-mex-files-to-use-64-bit-api.html。
>> v1_compiler

```

Figure 3.2

My Visual Studio 2015 installation path: C:\Program Files (x86)\VS 2015\professional

(Note: Add the installation directory to the Path of the environment variable → system variable.)

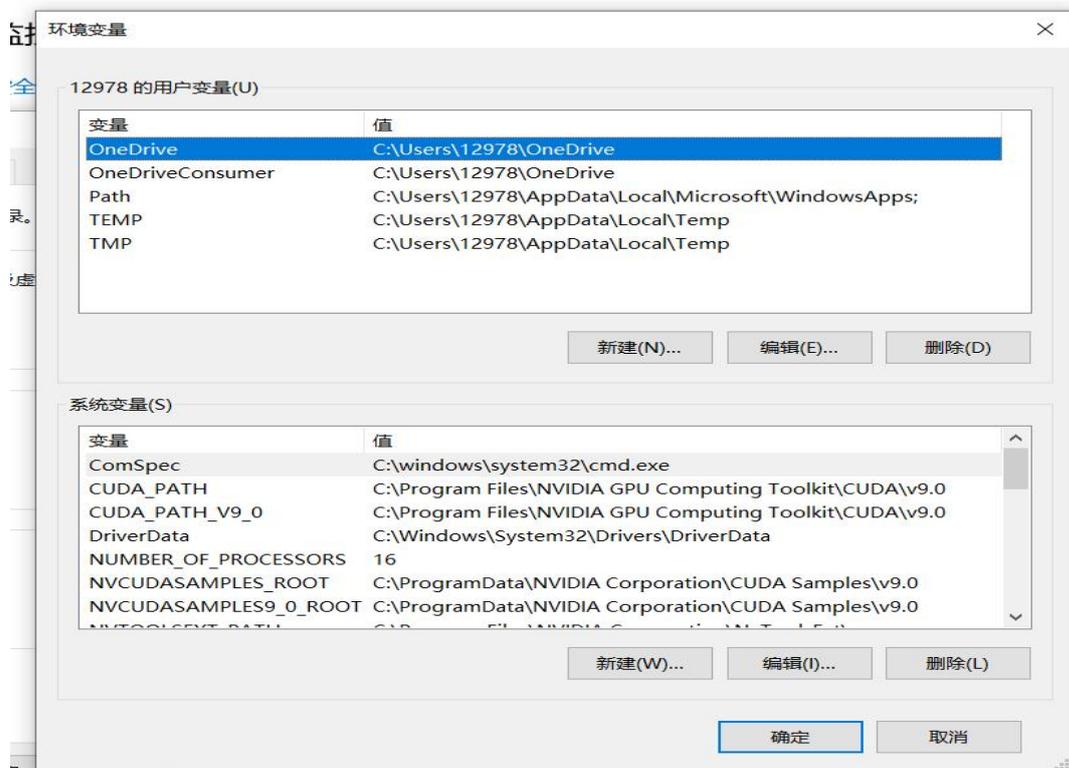


Figure 3.3

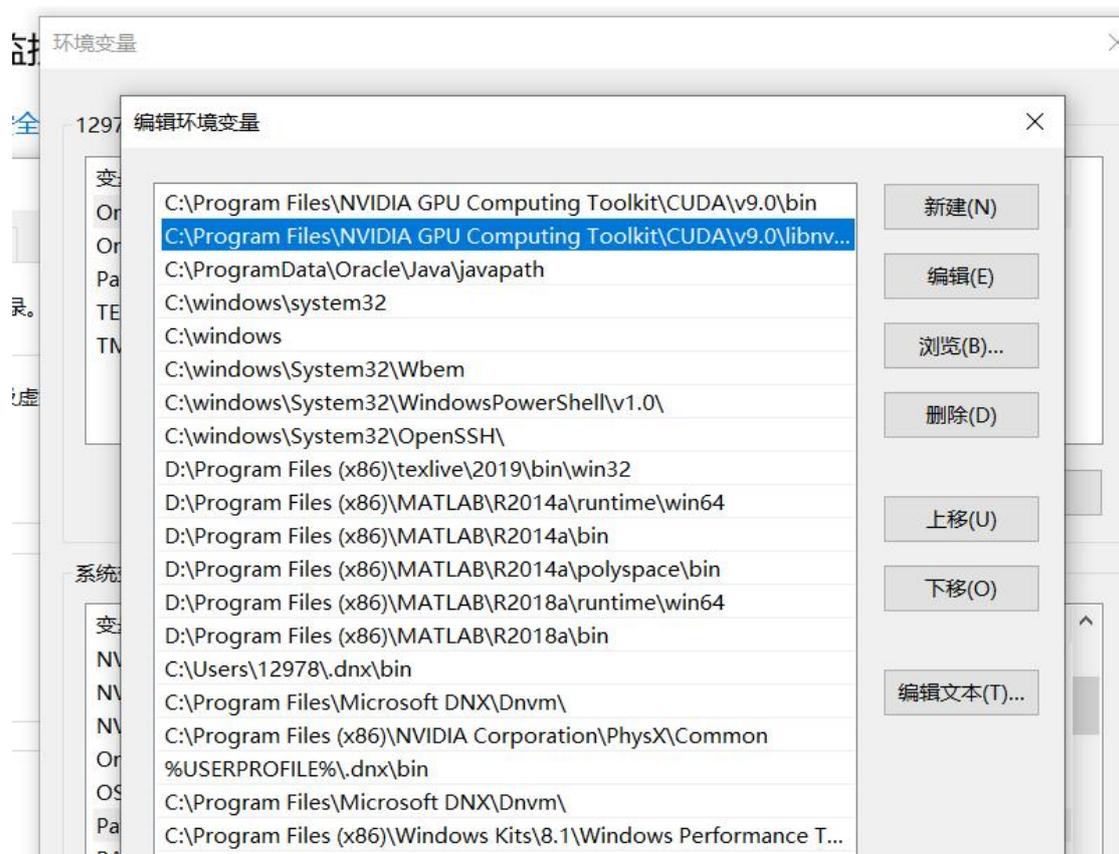


Figure 3.4

(3) Enter `vl_compilenn` in the command line window of MATLAB, `vl_compilenn` is successfully compiled.

```
>> vl_compilem
使用 'Microsoft Visual C++ 2015' 编译。
MEX 已成功完成。
fx MEX 已成功完成。
```

Figure 3.5

(4) Enter test command `vl_testnn` in the command line window of MATLAB. `vl_testnn` is successfully compiled.

```
>> vl_testnn
fx
```

```
正在运行 nnsnorm
正在设置 nnsnorm[dataType=single, device=cpu]
设置 nnsnorm[dataType=single, device=cpu] 在 0.0031351 秒内完成
正在运行 nnsnorm[dataType=single, device=cpu]/basic
nnsnorm[dataType=single, device=cpu]/basic 在 0.75398 秒内完成
正在拆解 nnsnorm[dataType=single, device=cpu]
拆解 nnsnorm[dataType=single, device=cpu] 在 0 秒内完成
nnsnorm 在 0.75711 秒内完成
```

result =

1×1793 [TestResult](#) 数组 - 属性:

```
Name
Passed
Failed
Incomplete
Duration
Details
```

总计:

```
1793 Passed, 0 Failed, 0 Incomplete.
135.0925 秒测试时间。
```

Figure 3.6

Fig 3.6 shows that without using CUDA and CUDNN to drive the GPU, the CPU is used alone to run the program compiled by matconvnet.

3.2 GPU compilation

First, look at the computer's CUDA device properties and enter gpuDevice in the MATLAB command line window:

```
>> gpuDevice

ans =

  CUDADevice - 属性:

      Name: 'NVIDIA GeForce RTX 2060'
      Index: 1
      ComputeCapability: '7.5'
      SupportsDouble: 1
      DriverVersion: 11.6000
      ToolkitVersion: 9
      MaxThreadsPerBlock: 1024
      MaxShmemPerBlock: 49152
      MaxThreadBlockSize: [1024 1024 64]
      MaxGridSize: [2.1475e+09 65535 65535]
      SIMDWidth: 32
      TotalMemory: 6.4421e+09
      AvailableMemory: 5.2690e+09
      MultiprocessorCount: 30
      ClockRateKHz: 1200000
      ComputeMode: 'Default'
      GPUOverlapsTransfers: 1
      KernelExecutionTimeout: 1
      CanMapHostMemory: 1
      DeviceSupported: 1
```

Figure 3.7

3.1.1 Install CUDA

Follow the steps on the official website to install CUDA:
<https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html>

Quick Start Guide (PDF) - v11.8.0 (older) - Last updated October 3, 2022 - [Send Feedback](#)

CUDA Quick Start Guide
 Minimal first-steps instructions to get CUDA running on a standard system.

1. Introduction
 This guide covers the basic instructions needed to install CUDA and verify that a CUDA application can run on each supported platform. These instructions are intended to be used on a clean installation of a supported platform. For questions which are not answered in this document, please refer to the [Windows Installation Guide](#) and [Linux Installation Guide](#). The CUDA installation packages can be found on the [CUDA Downloads Page](#).

2. Windows
 When installing CUDA on Windows, you can choose between the Network Installer and the Local Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. For more details, refer to the [Windows Installation Guide](#).

2.1. Network Installer
 Perform the following steps to install CUDA and verify the installation.

1. Launch the downloaded installer package.
2. Read and accept the EULA.
3. Select "next" to download and install all components.
4. Once the download completes, the installation will begin automatically.
5. Once the installation completes, click "next" to acknowledge the Nsight Visual Studio Edition installation summary.
6. Click "close" to close the installer.
7. Navigate to the Samples' nbody directory in https://github.com/NVIDIA/cuda-samples/tree/master/Samples/5_Domain_Specific/nbody.
8. Open the nbody Visual Studio solution file for the version of Visual Studio you have installed, for example, nbody_vs2019.sln.

Figure 3.8

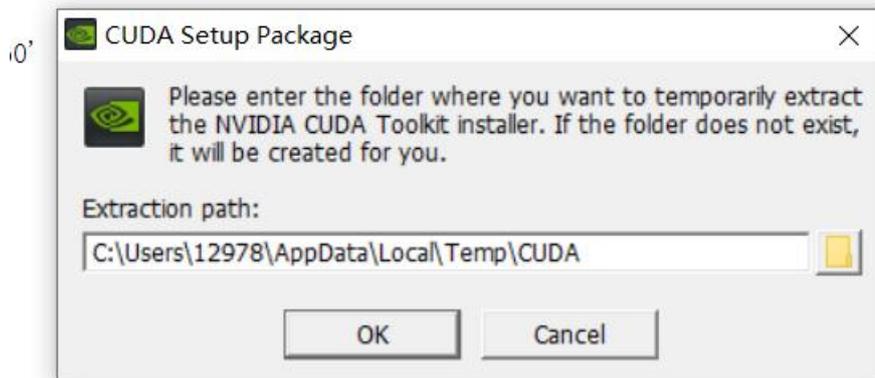


Figure 3.9



GPU0overlapsTransfers: 1

Figure 3.10

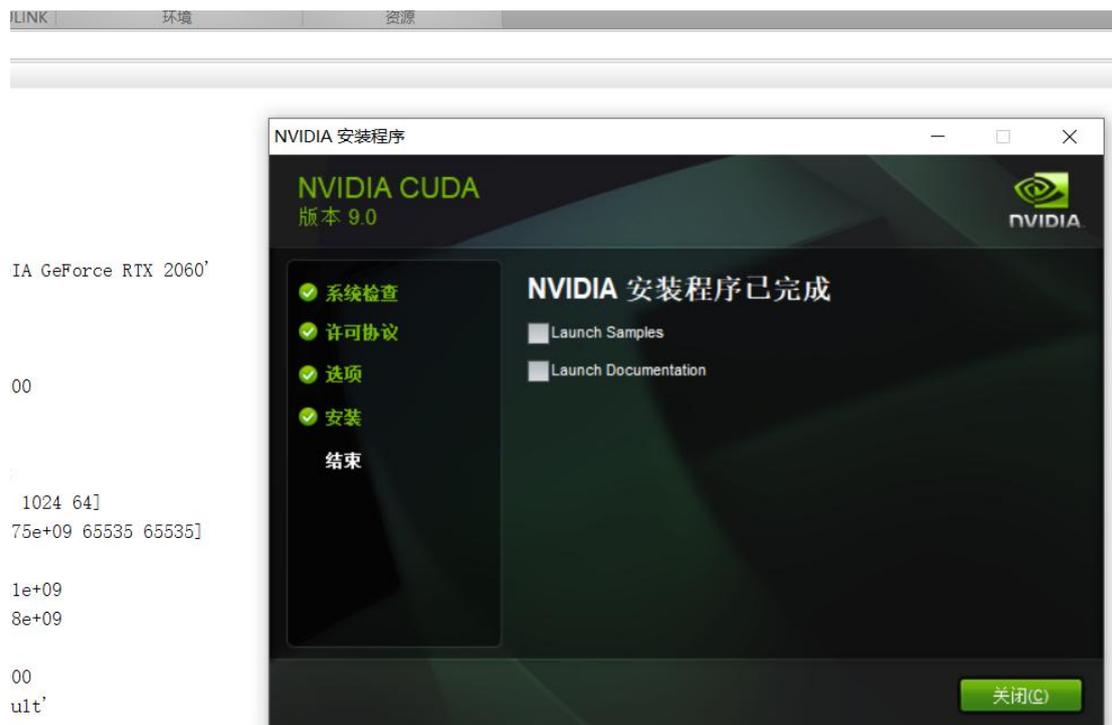


Figure 3.11

Open the nbody Visual Studio solution file for the version of Visual Studio you have installed. Then, Open the 'Build'menu within Visual Studio and click 'Build Solution'.

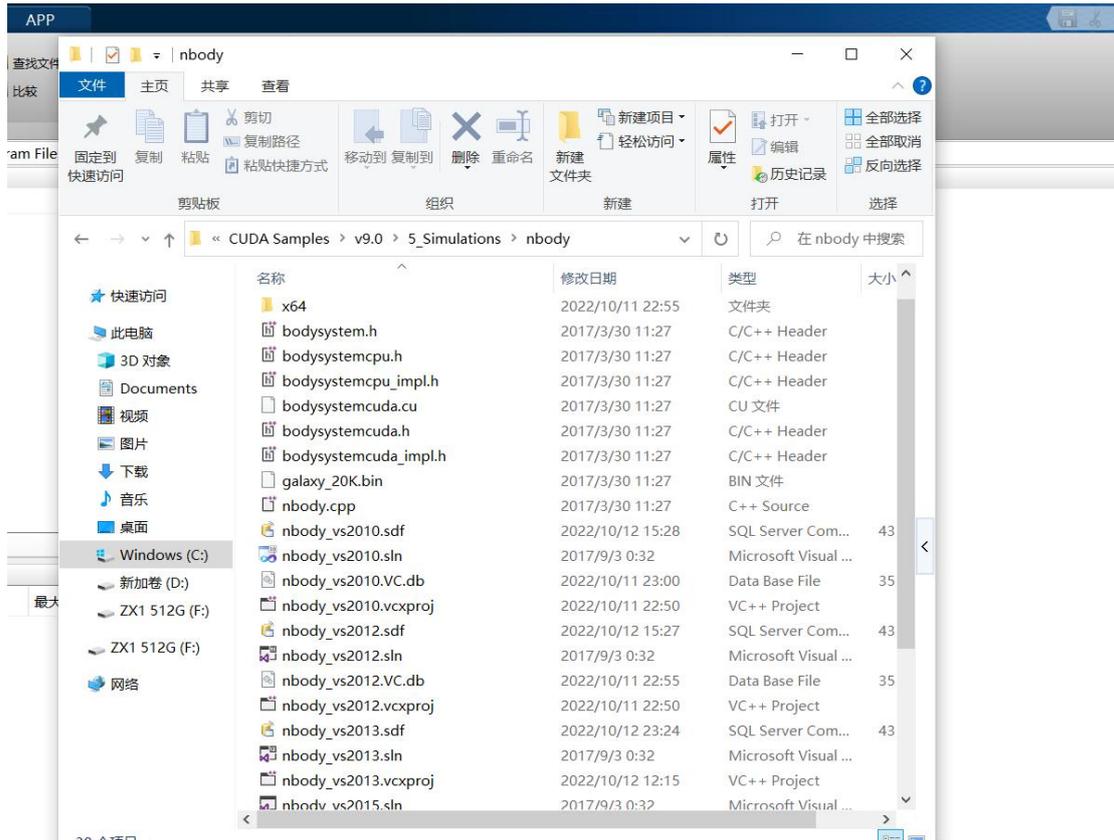


Figure 3.12

Open `nbody_Vs2010.sln` as follows:

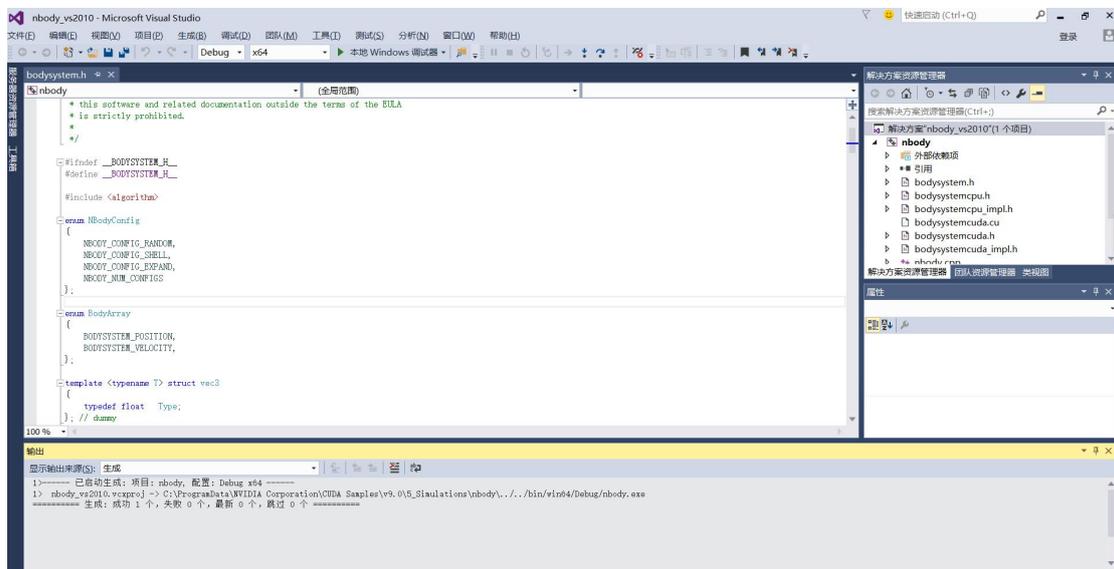


Figure 3.13

Open `nbody_vs2012.sln` as follows:

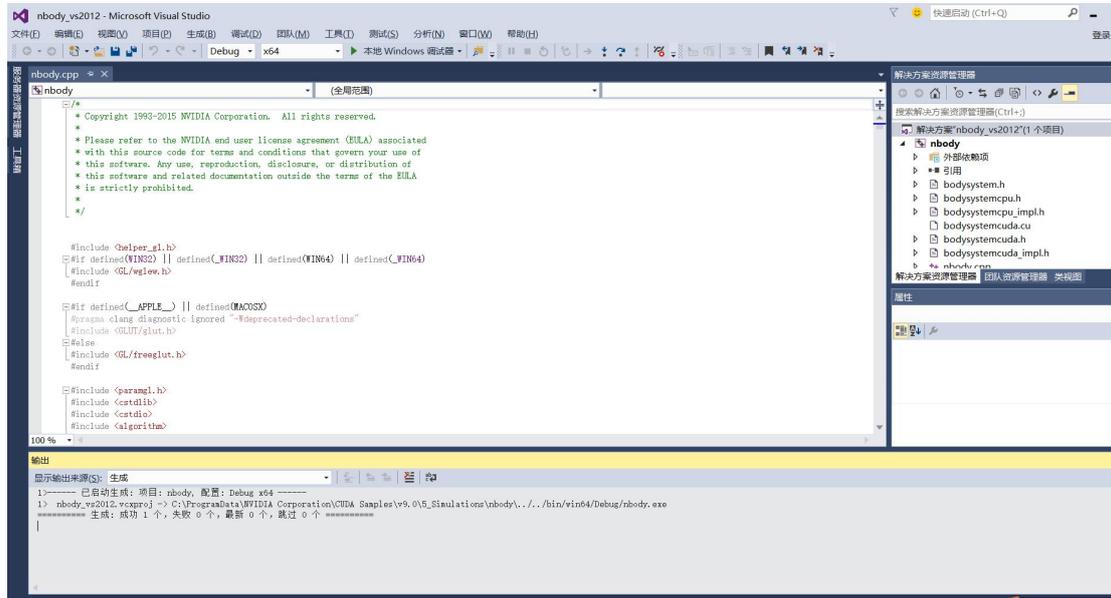


Figure 3.14

Open `nbody_vs2013.sln` as follows:

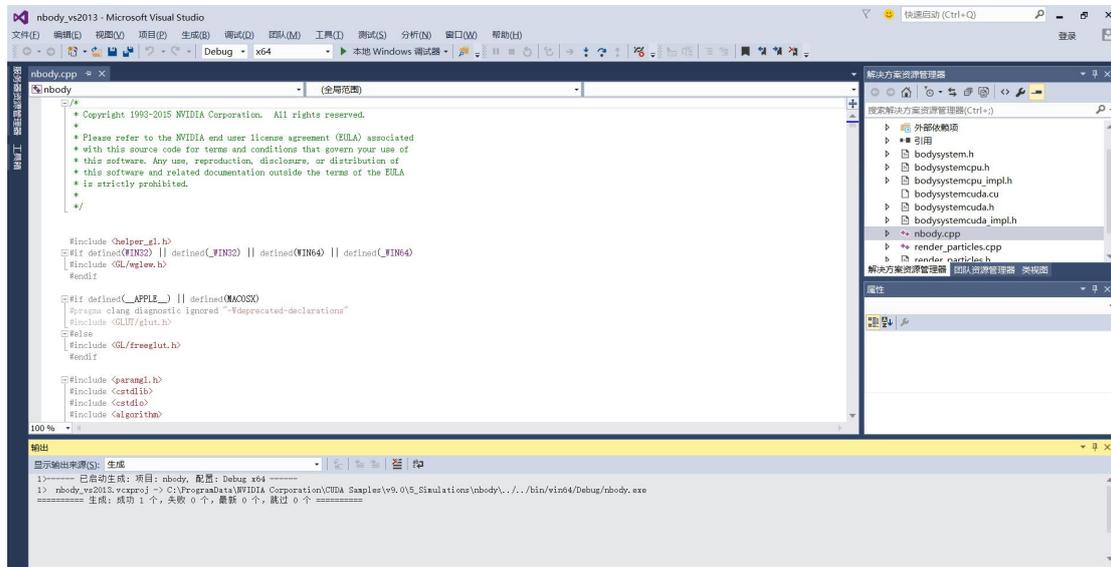


Figure 3.15

Navigate to the CUDA Samples' build directory and run the `nbody` sample. Note: Run samples by navigating to the executable's location, otherwise it will fail to locate dependent resources.

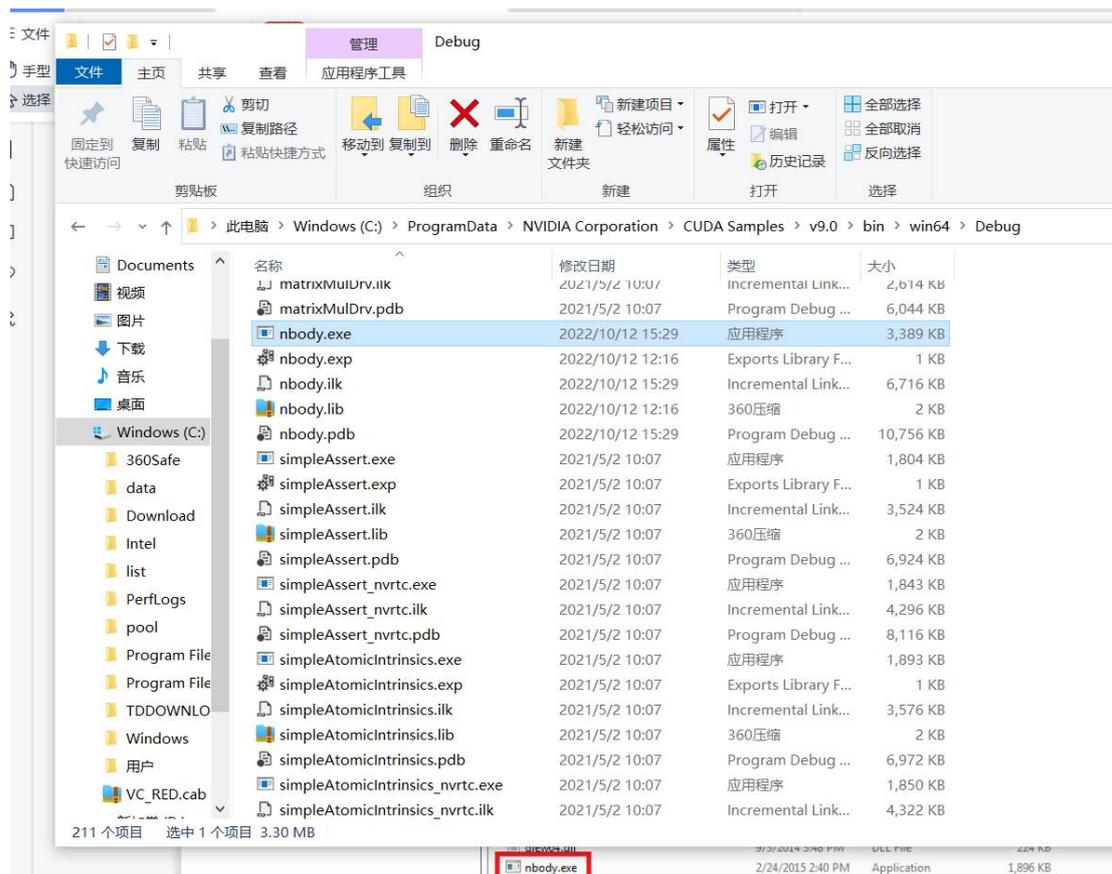


Figure 3.16 (图 3.16)

The execution results are as follows:

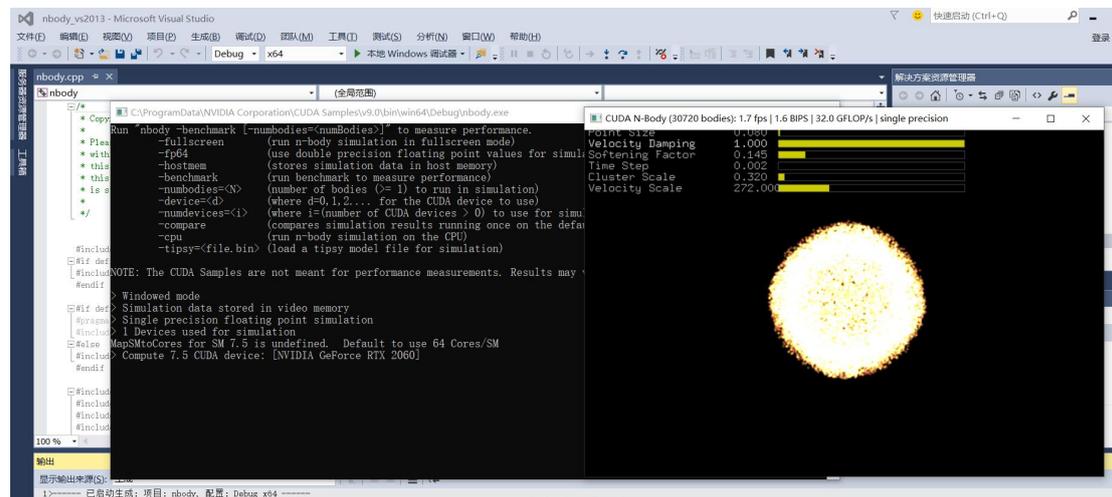
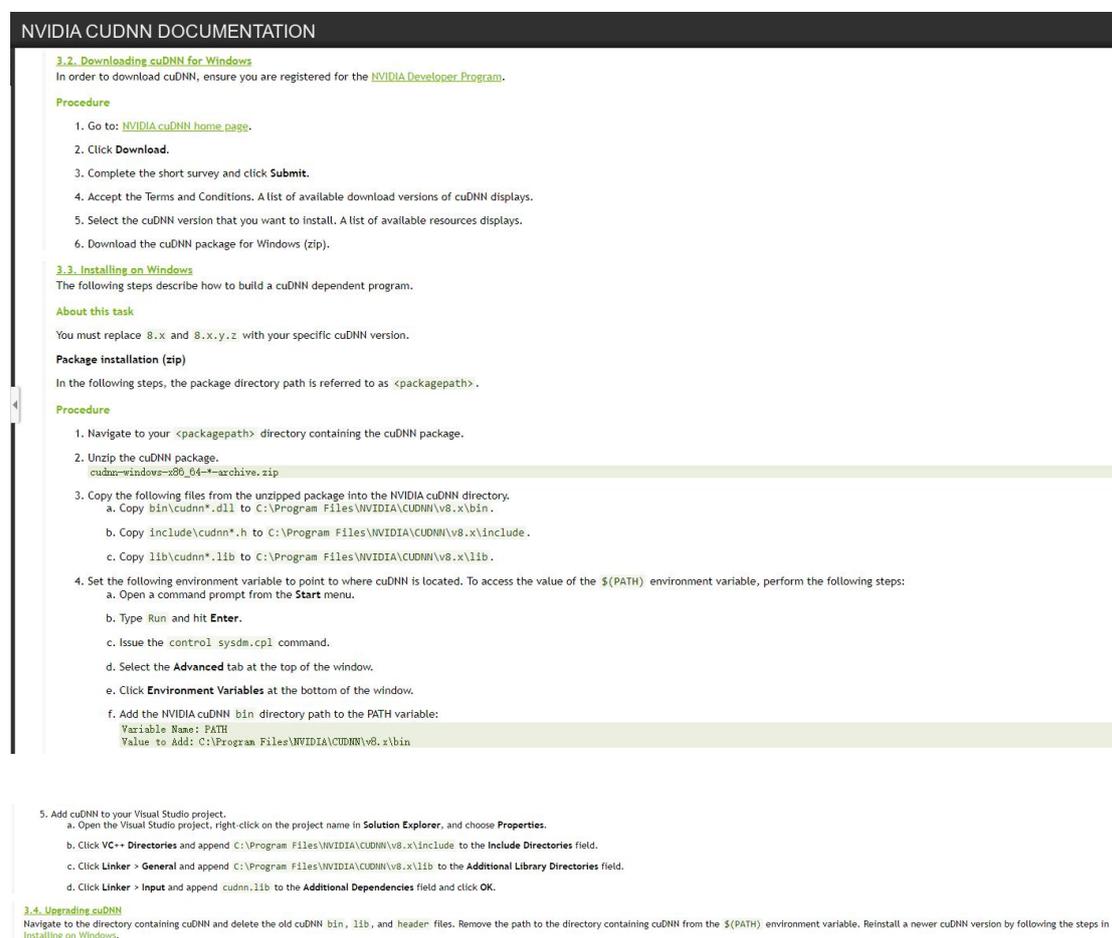


Figure 3.17

The results in Figure 3.17 show that the CUDNN library is installed successfully, and the GPU is used to speed up the operation.

3.1.2 Install CUDNN

Follow the steps on the official website to install CUDNN:
<https://docs.nvidia.com/deeplearning/cudnn/developer-guide/index.html>



The screenshot shows the NVIDIA CUDNN documentation page. It is divided into two main sections: 3.2. Downloading cuDNN for Windows and 3.3. Installing on Windows. Section 3.2 includes a procedure with six steps: 1. Go to the NVIDIA cuDNN home page. 2. Click Download. 3. Complete the short survey and click Submit. 4. Accept the Terms and Conditions. 5. Select the cuDNN version. 6. Download the cuDNN package. Section 3.3 includes instructions on how to build a cuDNN dependent program, a note about replacing version numbers, and a procedure for package installation (zip) with 11 steps. Step 3 details copying files to the NVIDIA cuDNN directory. Step 4 details setting the PATH environment variable. Step 5 details adding cuDNN to a Visual Studio project. Step 6 details upgrading cuDNN.

3.2. Downloading cuDNN for Windows
In order to download cuDNN, ensure you are registered for the [NVIDIA Developer Program](#).

Procedure

1. Go to: [NVIDIA cuDNN home page](#).
2. Click **Download**.
3. Complete the short survey and click **Submit**.
4. Accept the **Terms and Conditions**. A list of available download versions of cuDNN displays.
5. Select the cuDNN version that you want to install. A list of available resources displays.
6. Download the cuDNN package for Windows (zip).

3.3. Installing on Windows
The following steps describe how to build a cuDNN dependent program.

About this task
You must replace 8.x and 8.x.y.z with your specific cuDNN version.

Package installation (zip)
In the following steps, the package directory path is referred to as <packagepath>.

Procedure

1. Navigate to your <packagepath> directory containing the cuDNN package.
2. Unzip the cuDNN package.
`cmd> windows-x86_64-archivé.zip`
3. Copy the following files from the unzipped package into the NVIDIA cuDNN directory.
 - a. Copy `bin\cudnn*.dll` to `C:\Program Files\NVIDIA\CUDNN\v8.x\bin`.
 - b. Copy `include\cudnn*.h` to `C:\Program Files\NVIDIA\CUDNN\v8.x\include`.
 - c. Copy `lib\cudnn*.lib` to `C:\Program Files\NVIDIA\CUDNN\v8.x\lib`.
4. Set the following environment variable to point to where cuDNN is located. To access the value of the \$(PATH) environment variable, perform the following steps:
 - a. Open a command prompt from the **Start** menu.
 - b. Type **Run** and hit **Enter**.
 - c. Issue the control `sysdm.cpl` command.
 - d. Select the **Advanced** tab at the top of the window.
 - e. Click **Environment Variables** at the bottom of the window.
 - f. Add the NVIDIA cuDNN `bin` directory path to the **PATH** variable:
Variable Name: `PATH`
Value to Add: `C:\Program Files\NVIDIA\CUDNN\v8.x\bin`

5. Add cuDNN to your Visual Studio project.

- a. Open the Visual Studio project, right-click on the project name in **Solution Explorer**, and choose **Properties**.
- b. Click **VC++ Directories** and append `C:\Program Files\NVIDIA\CUDNN\v8.x\include` to the **Include Directories** field.
- c. Click **Linker > General** and append `C:\Program Files\NVIDIA\CUDNN\v8.x\lib` to the **Additional Library Directories** field.
- d. Click **Linker > Input** and append `cudnn.lib` to the **Additional Dependencies** field and click **OK**.

3.4. Upgrading cuDNN
Navigate to the directory containing cuDNN and delete the old cuDNN `bin`, `lib`, and `header` files. Remove the path to the directory containing cuDNN from the \$(PATH) environment variable. Reinstall a newer cuDNN version by following the steps in [Installing on Windows](#).

Figure 3.18

Execute the following code in MATLAB:

(1) Enter `mex-setup c++` in the MATLAB command line window.

```
>> mex -setup c++
MEX 配置为使用 'Microsoft Visual C++ 2015' 以进行 C++ 语言编译。
警告: MATLAB C 和 Fortran API 已更改, 现可支持
包含 2^32-1 个以上元素的 MATLAB 变量。您需要
更新代码以利用新的 API。
您可以在以下网址找到更多的相关信息:
https://www.mathworks.com/help/matlab/matlab\_external/upgrading-mex-files-to-use-64-bit-api.html
fx >> |
```

Figure 3.19

(2) Enter `vl_compile('enableGpu', true, ...'cudaRoot', 'C:\Program Files\NVIDIA`


```

使用 'Microsoft Visual C++ 2015 (C)' 编译。
MEX 已成功完成。
>> run vl_setupnn
>> vl_testnn('gpu', true)
fx

```

Figure 3.23

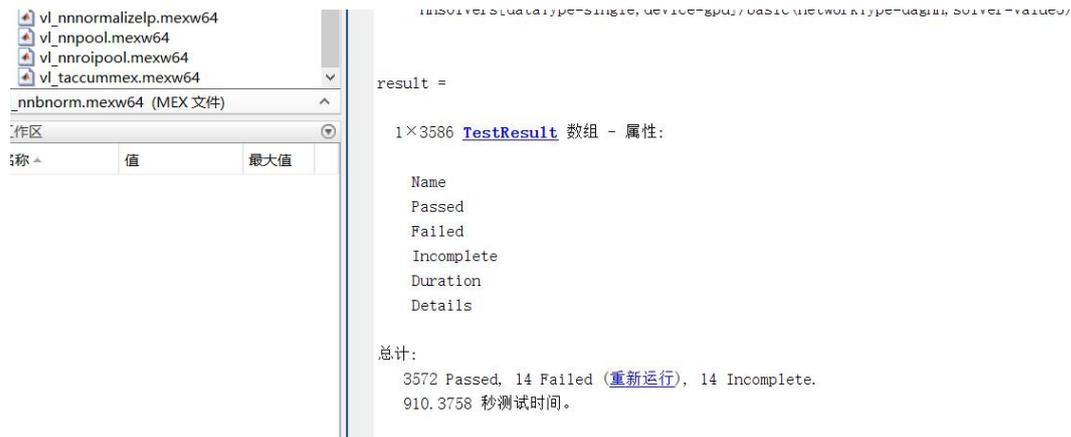


Figure 3.24

Fig 3.24 shows that the matconvnet test code runs successfully in the GPU environment.

After the compilation of CPU and GPU, you can run the code of WSWTNN-PnP.

4. Run the WSWTNN-PnP method code

4.1 Code download:

website: <https://github.com/LiuTing20a/WSWTNN-PnP1>

4.2 Run Demo_WSWTNN_PnP

The code is running::

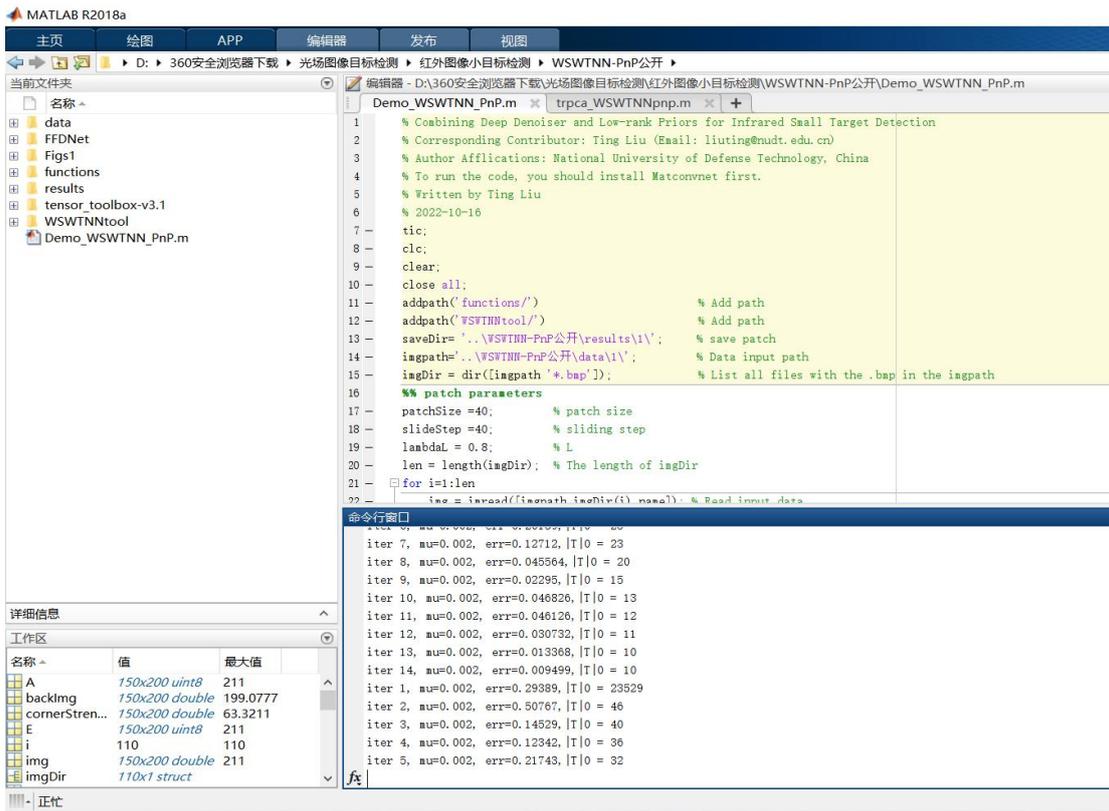


Figure 4.1

Experimental results in the process of code running are shown as follows:

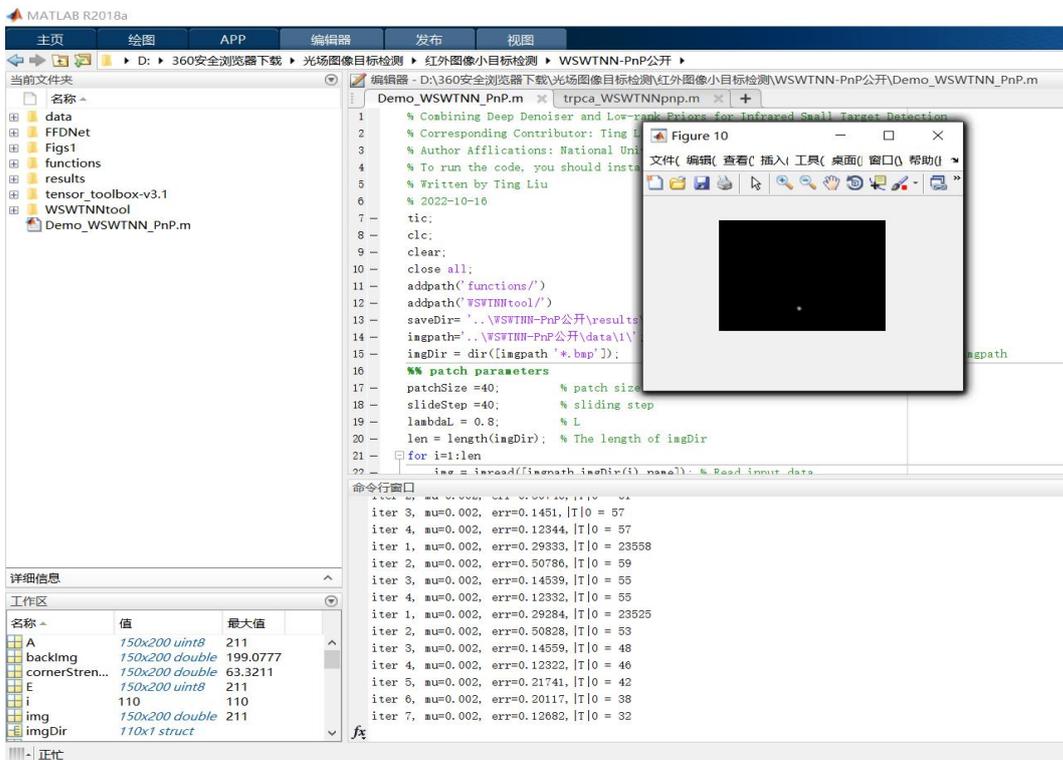


Figure 4.2

Code running completed:

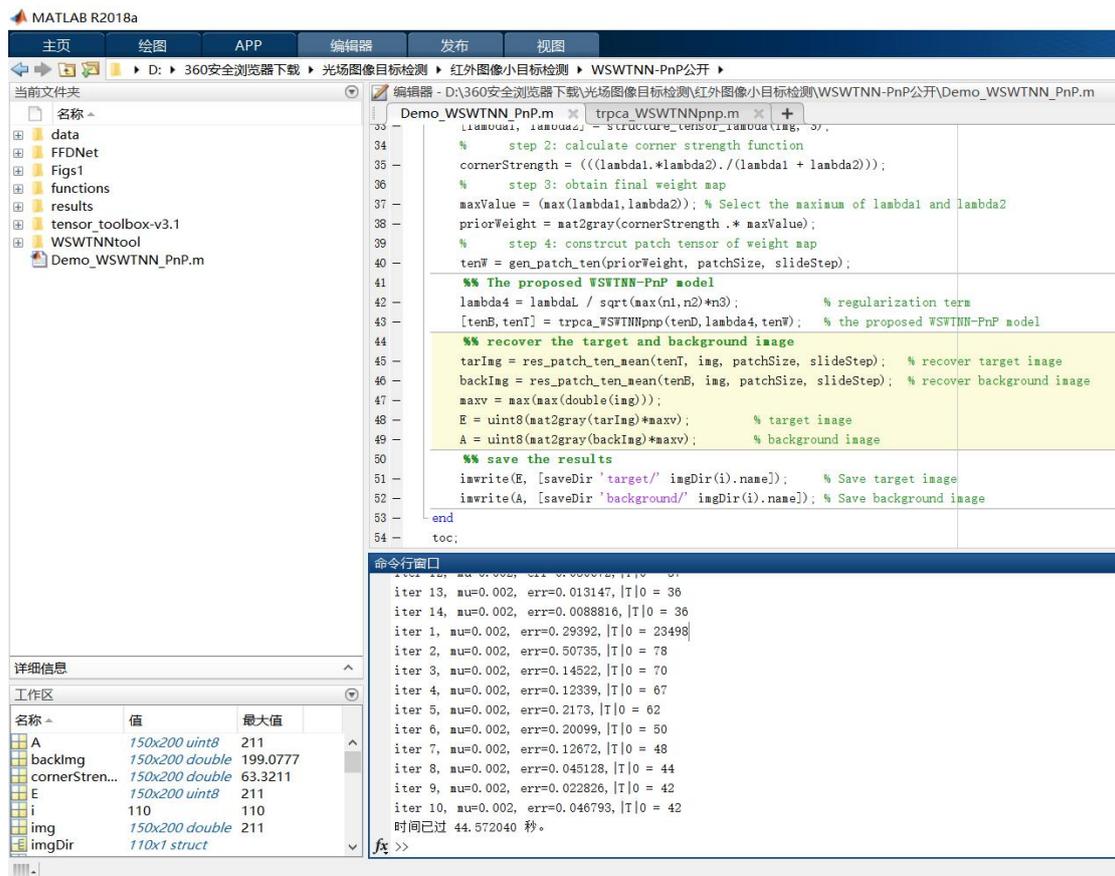


Figure 4.3

The experimental results are stored in the folder

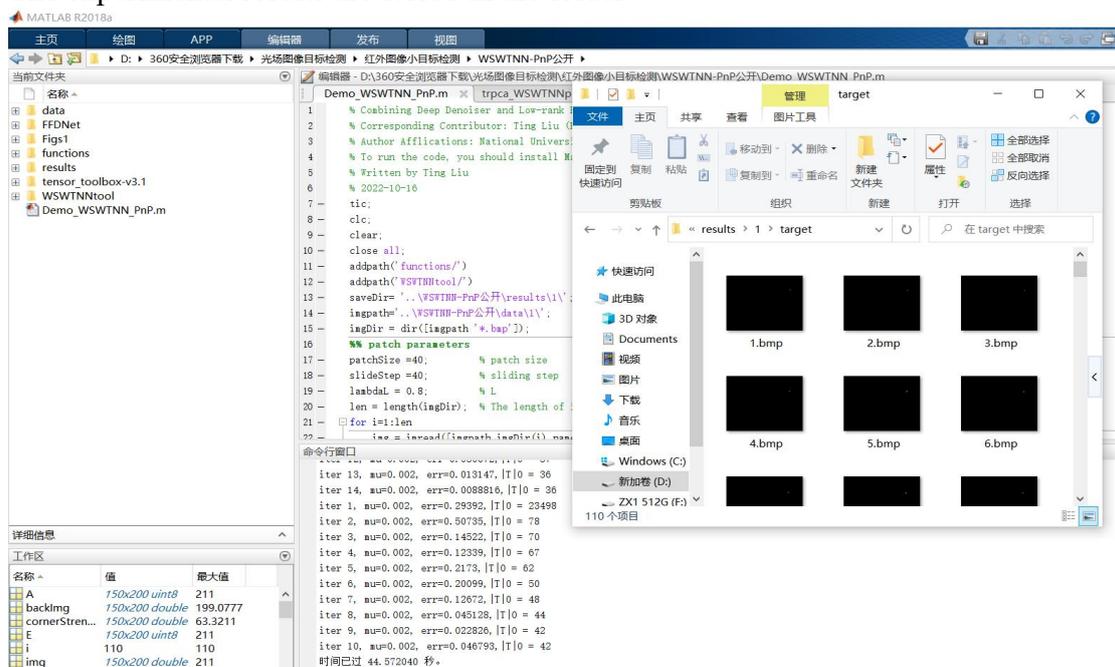


Figure 4.4