Qiushi Academy for Advanced Studies, Zhejiang University

**Software Environment Construction Guide**

1. Install anaconda
2. Install the GPU driver and the corresponding cuda version. For example: GTX1060 driver and cuda 10.2
3. Open pytorch.org and use conda to install the corresponding version. For example: conda install pytorch==1.7.0 torchvision==0.8.0 torchaudio==0.7.0 cudatoolkit=10.2 -c pytorch
4. Install Pycharm, note that the configuration environment is Python in the anaconda directory
5. Use Pycharm to test registration.py, if the operation is successful, the Pytorch environment configuration is successful
6. Copy the compiled version of OpenCV corresponding to mingw64, for example: opencv3.43\_mingw\_64
7. Add system environment variable: MinGW 64-bit version

For example: C:\opencv3.43\_mingw\_64\install\x64\mingw\bin

1. Install the corresponding version of Qt, such as Qt 5.12.8
2. Click the .pro file to open the project, and select the Qt corresponding MINGW 64 compiler
3. Change the opencv and python paths in the .pro file as follows

#Environment opencv3.43\_mingw\_64

INCLUDEPATH += D:\ProgramData\opencv3.43\_mingw\_64\install\include

LIBS += -L D:\ProgramData\opencv3.43\_mingw\_64\install\x64\mingw\lib\libopencv\_\*.a \

# python enviroment

INCLUDEPATH += 'D:\ProgramData\Anaconda3\include'

INCLUDEPATH += 'D:\ProgramData\Anaconda3\Lib\site packages\numpy\core\include\numpy'

LIBS += 'D:\ProgramData\Anaconda3\libs\python3.lib'

LIBS += 'D:\ProgramData\Anaconda3\libs\python3\_d.lib'

LIBS += 'D:\ProgramData\Anaconda3\libs\python37.lib'

LIBS += 'D:\ProgramData\Anaconda3\libs\python37\_d.lib'

(If there is no python3\_d.lib, python37\_d.lib, copy the original files of python3.lib and python37.lib, and add the suffix of ‘\_d’ after them)

1. Click the compile button to test whether the application is output

**The problems encountered are as follows:**

1. Line 448 of object.h



The reason is that slots in python's object.h conflict with Qt's Slot.

solution:

**Original:**

typedef struct{

const char\* name;

int basicsize;

int itemsize;

unsigned int flags;

PyType\_Slot \*slots; /\* terminated by slot==0. \*/

} PyType\_Spec;

**Change:**

typedef struct{

const char\* name;

int basicsize;

int itemsize;

unsigned int flags;

#undef slots // Here to cancel the slots macro definition

PyType\_Slot \*slots; /\* terminated by slot==0. \*/

#define slots Q\_SLOTS　　// The restoration of the slots macro definition here is consistent with that in QObjectDefs.h in QT

} PyType\_Spec;

1. he **import\_array** function reports an error

{

import\_array();

}

Select the error message and jump to the error location: line 1531 of the \_multiarray\_api.h function

Remove the “return NULL” of this row;