**OpenCV**:

Installation: <https://opencv.org/>

//添加compile option

CMakeList:

cmake\_minimum\_required(VERSION 2.8)

project( main)

find\_package( OpenCV REQUIRED )

include\_directories( ${OpenCV\_INCLUDE\_DIRS} )

SET(CMAKE\_EXE\_LINKER\_FLAGS “-no-pie”)

Add\_compile\_options(-fPIC)

add\_executable( main main.cpp )

target\_link\_libraries( main ${OpenCV\_LIBS} )

#include <opencv2/opencv.hpp>

#include <iostream>

Using namespace std;

Using namespace cv;

Int main(){

VideoCapture cap(“/dev/video0”);

While(1){

Mat frame;

Cap>>frame;

If(frame.empty()) break;

Imshow(“Frame”,frame);

Char c=(char)waitKey(1);

If(c==27)break;

}

}

Cmake ..

make

Error: OpenCV Error: Unspecified error (The function is not implemented. Rebuild the library with Windows, GTK+ 2.x or Carbon support. If you are on Ubuntu or Debian, install libgtk2.0-dev and pkg-config, then re-run cmake or configure script) in cvNamedWindow

Solution: turn on those backends in installation

Sudo aptitude search libgtk2.0-dev

Mkdir Release (in opencv directory)

Cd Release

cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=/usr/local -D WITH\_TBB=ON -D BUILD\_NEW\_PYTHON\_SUPPORT=ON -D WITH\_V4L=ON -D INSTALL\_C\_EXAMPLES=ON -D INSTALL\_PYTHON\_EXAMPLES=ON -D BUILD\_EXAMPLES=ON -D WITH\_QT=ON -D WITH\_GTK=ON -D WITH\_OPENGL=ON ..

make

sudo make install

If independency has lost, run

Sudo aptitude install libgtk2.0-dev

Error: This plugin does not support propagateSizeHints()

Solution: Delete export QT\_QPA\_PLATFORM='offscreen' in ~/.bashrc

Error: QXcbConnection: Could not connect to display

Solution:

echo ${DISPLAY}

if not empty, run

export DISPLAY=’:1.0’

OR

export DISPLAY=’:2.0’

if empty, run

export DISPLAY=’:0.0’

**Embedding python into C++:**

CMakeList:

cmake\_minimum\_required(VERSION 3.12)

project(cpppy)

set(CMAKE\_CXX\_STANDARD 14)

add\_compile\_options(-lpython)

include\_directories(C:\\Users\\69009\\Anaconda3\\include)

link\_libraries(C:/Users/69009/Anaconda3/libs/python37.lib)

link\_directories(C:/Users/69009/Anaconda3/libs)

add\_executable(cpppy main.cpp)

find\_package(PythonLibs)

target\_link\_libraries(cpppy "C:/Users/69009/Anaconda3/Lib/site-packages")

set(Python3\_ROOT\_DIR C:/Users/69009/Anaconda3)

find\_package(Python3 3.7 COMPONENTS Interpreter Development)

if (Python3\_FOUND)

message("Python include directory: " ${Python3\_INCLUDE\_DIRS})

message("Python version is: " ${Python3\_VERSION})

include\_directories(${Python3\_INCLUDE\_DIRS})

target\_link\_libraries(cpppy ${Python3\_LIBRARIES})

endif (Python3\_FOUND)

In the above, change the directories to the environment where packages are installed

Adding python into environment variable

Make sure that the version of MINGW and python are the same (i.e. both 64 bit)

If you are using conda environment, please make sure that you add anaconda into system environment path. If no, adding: whole directory (/Anaconda3/), Scripts(/Anaconda3/Scripts), Library(/Anaconda3/Library/bin)