

Visual studio 2017

OpenCV-3.4.3

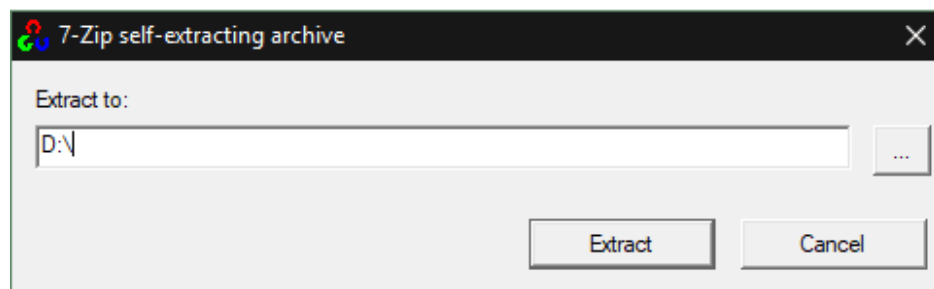
Dlib-19.15

Install visual studio 2017

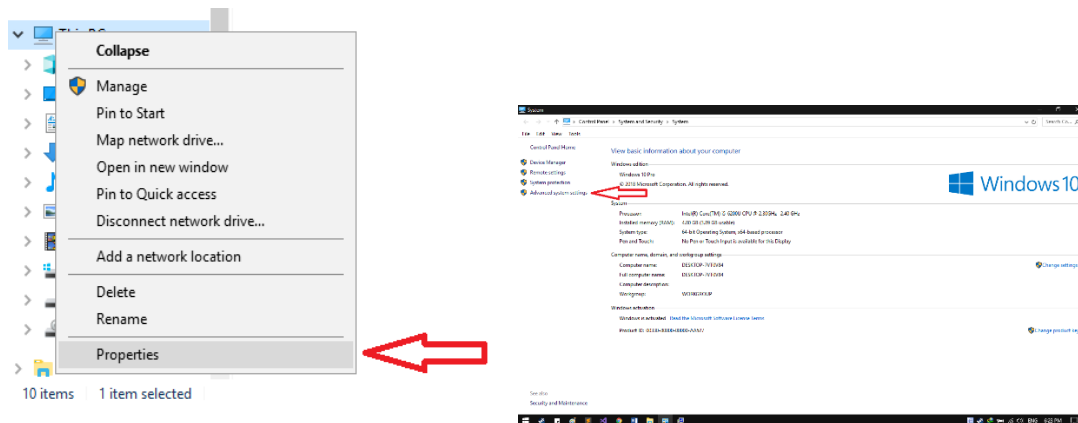
- Install visual studio (download from official website: <https://visualstudio.microsoft.com>)
- Install C++

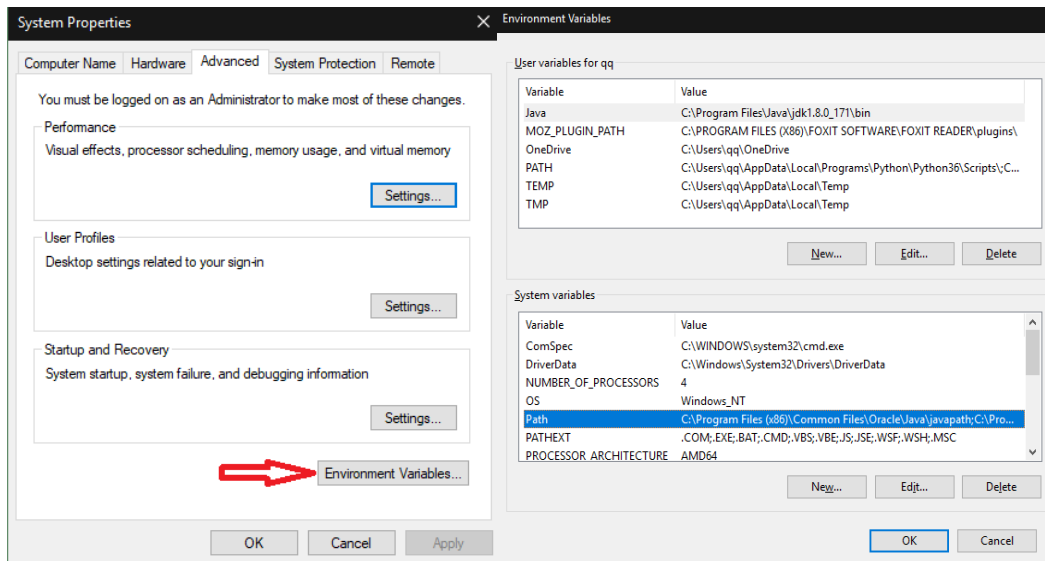
Install opencv

- Download opencv from official website : <https://opencv.org>
- Install openCV (install after openCV 3.0 for eye glance project)
- Extract openCV in your favorite path(e. g. D:\)

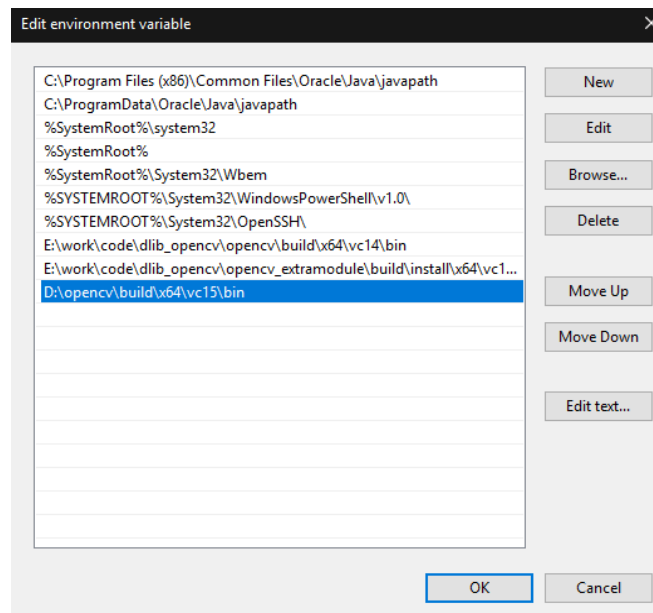


- Setup openCV in enviroment: Right click This PC -> Properties -> Advanced system settings -> Enviroment Variables, find Path in System variables



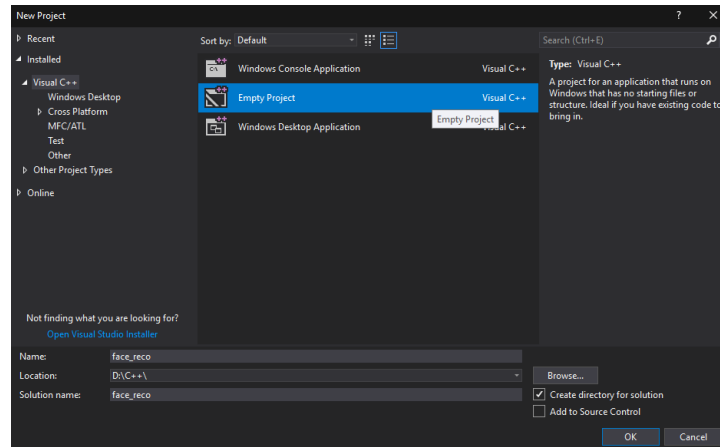


- Set the environment of openCV : D:\opencv\build\x64\vc15\bin

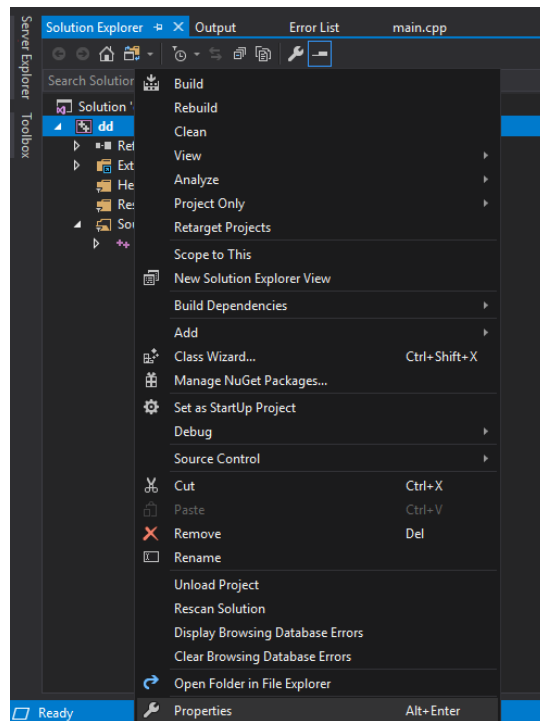


Add opencv to your project

- Open visual studio, creat a new Empty project, chose your favorite path.



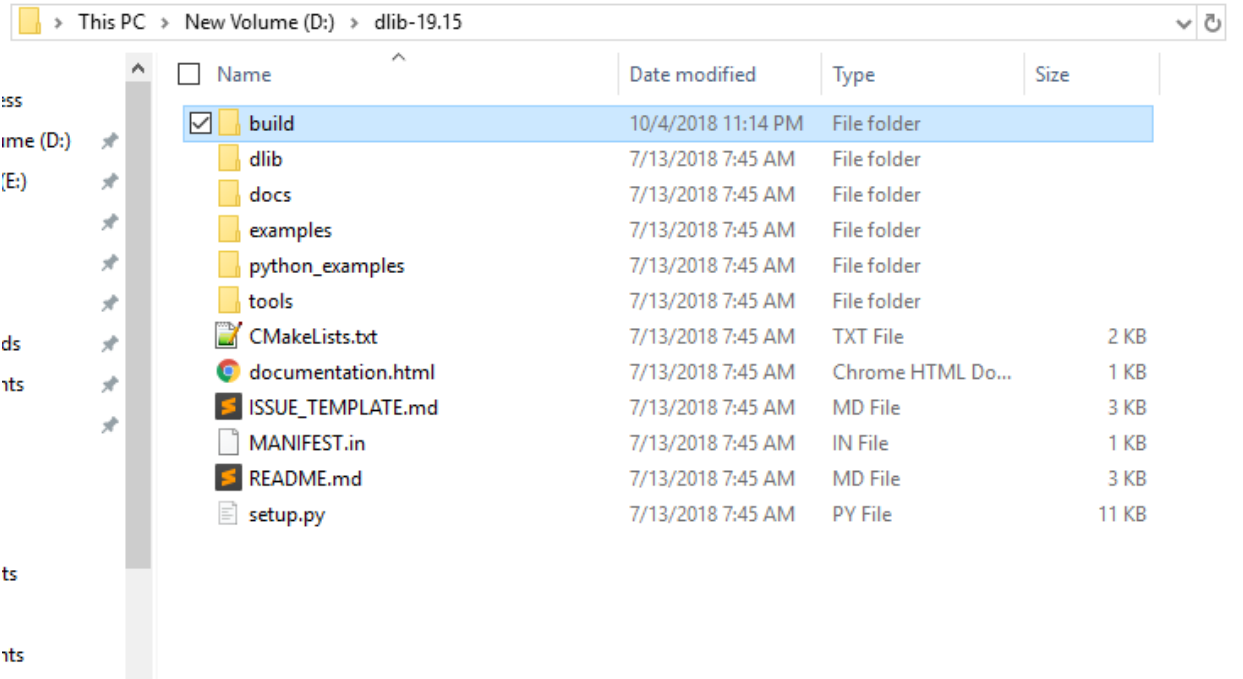
- Link the opencv to your project: Right click in your project -> Properties



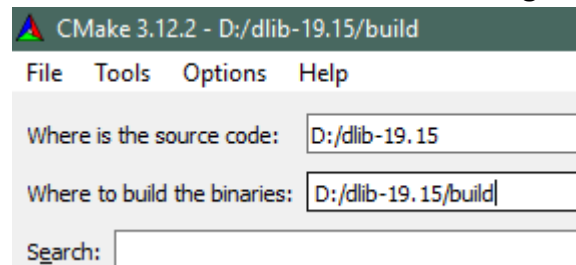
- add a source code to your project
- VC++ Directories:
 - add Include Directories: D:\opencv\build\include
 - add Library Directories: D:\opencv\build\x64\vc15\lib
- Linker -> Input add Additional Dependencies:
 - “opencv_world343d.lib” for Debug mode
 - “opencv_world343.lib” for Release mode.

Install dlib

- Download dlib from official page: <http://dlib.net>
- Unzip it in your favorite path(D:\dlib-19.15)
- Go to the dlib folder , creat a new folder build



- Download cmake-gui: <https://cmake.org/download>
- Unzip it in your favorite path(e.g D:\cmake-3.12.2-win64-x64)
- Open cmake D:\cmake-3.12.2-win64-x64\bin\cmake-gui.exe



- Choose source D:/dlib-19.15
- Choose build : D:/dlib-19.15/build
- Click configure -> Fininsh

Specify the generator for this project

Visual Studio 15 2017 Win64

Optional toolset to use (argument to -T)

☒ Use default native compilers
☐ Specify native compilers
☐ Specify toolchain file for cross-compiling
☐ Specify options for cross-compiling

Finish Cancel

Name	Value
CMAKE_BUILD_TYPE	Release
CMAKE_CONFIGURATION_TYPES	Debug;Release;MinSizeRel;RelWithDebInfo
CMAKE_INSTALL_PREFIX	C:/Program Files/Project
CUDA_HOST_COMPILER	\$(VCInstallDir)Tools/MSVC/\$(VCToolsVersion)/bin/Host\$(Platform)/\$(PlatformTarget)
CUDA_SDK_ROOT_DIR	CUDA_SDK_ROOT_DIR-NOTFOUND
CUDA_TOOLKIT_ROOT_DIR	CUDA_TOOLKIT_ROOT_DIR-NOTFOUND
DLIB_ENABLE_ASSERTS	<input type="checkbox"/>
DLIB_ENABLE_STACK_TRACE	<input type="checkbox"/>
DLIB_GIF_SUPPORT	OFF
DLIB_ISO_CPP_ONLY	<input type="checkbox"/>
DLIB_JPEG_SUPPORT	<input checked="" type="checkbox"/>
DLIB_LINK_WITH_SQLITE3	OFF
DLIB_NO_GUI_SUPPORT	<input type="checkbox"/>
DLIB_PNG_SUPPORT	<input checked="" type="checkbox"/>
DLIB_USE_BLAS	OFF
DLIB_USE_CUDA	OFF
DLIB_USE_LAPACK	OFF
DLIB_USE_MKL_FFT	OFF
DLIB_USE_MKL_SEQUENTIAL	<input type="checkbox"/>
USE_AVX_INSTRUCTIONS	<input type="checkbox"/>
USE_SSE2_INSTRUCTIONS	<input checked="" type="checkbox"/>
USE_SSE4_INSTRUCTIONS	<input checked="" type="checkbox"/>

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Configure Generate Open Project Current Generator: Visual Studio 15 2017 Win64

```

Looking for pthread.h
Looking for pthread.h - not found
Found Threads: TRUE
CUDA_TOOLKIT_ROOT_DIR not found or specified
Could NOT find CUDA (missing: CUDA_TOOLKIT_ROOT_DIR CUDA_NVOC_EXECUTABLE CUDA_INCLUDE_DIRS CUDA_CUDART_LIBRARY) (Required is at least version "7.5")
Disabling CUDA support for dlib. DLIB WILL NOT USE CUDA
C++11 activated.
Configuring done
  
```

- Click use USE_SSE2_INSTRUCTIONS, USE_SSE4_INSTRUCTIONS, Click configure again

```

Using CMake version: 3.12.2
Compiling dlib version: 19.15.0
Enabling SSE4 instructions
C++11 activated.
Configuring done
  
```

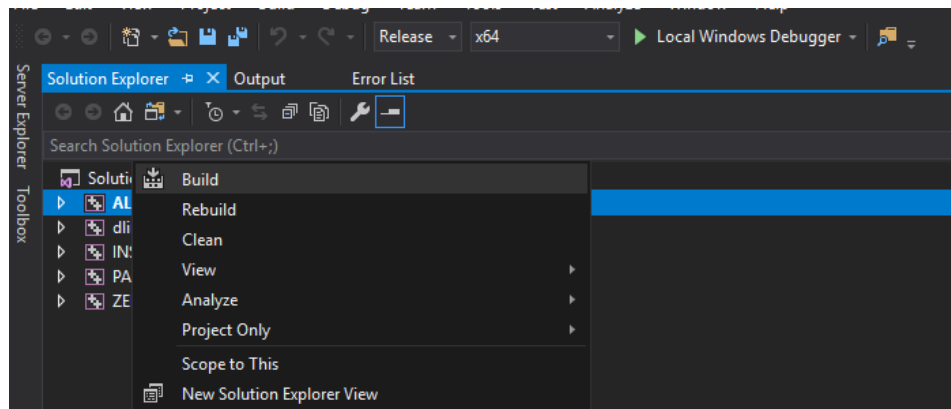
- Click generate

```
Using CMake version: 3.12.2
Compiling dlib version: 19.15.0
Enabling SSE4 instructions
C++11 activated.
Configuring done
Generating done
```

Done

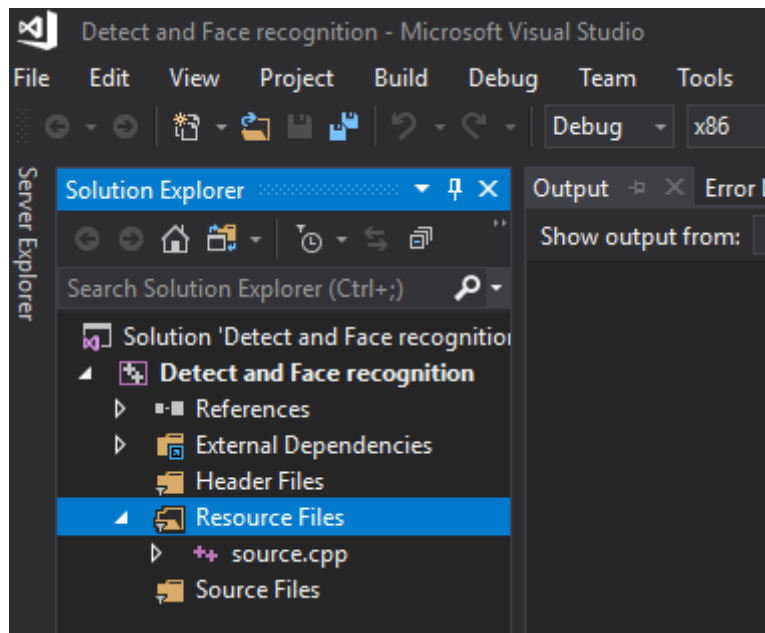
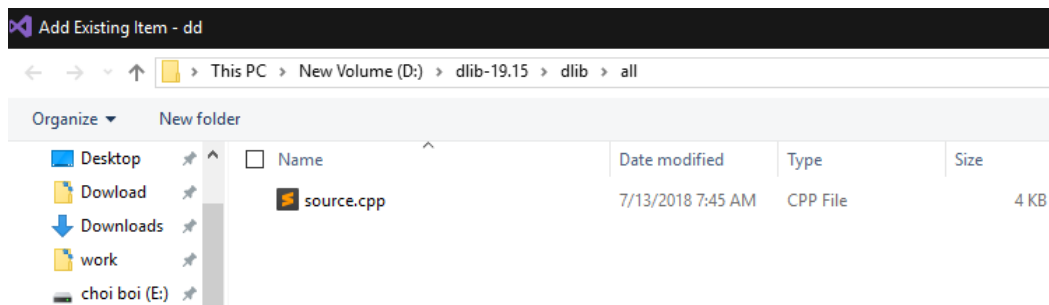
Build dlib with visual studio

- Open the project D:\dlib-19.15\build\Project.sln
- Choose build in Release and x64 then build



Link dlib to your project

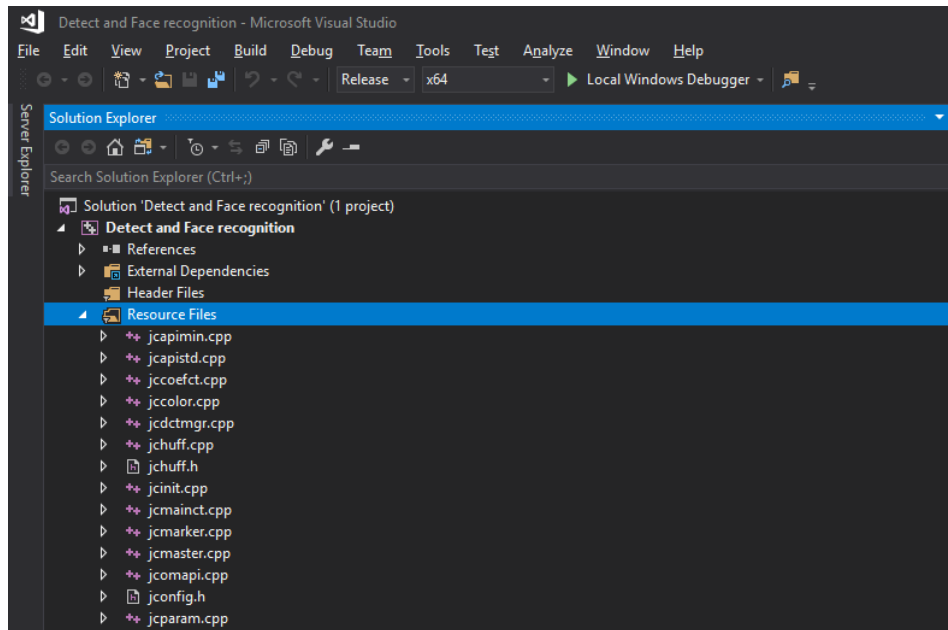
- Open your project you made before, open Property Pages(Alt + Enter)
- VC++ Directories:
 - add Include Directories: D:\dlib-19.15
 - add Library Directories: D:\dlib-19.15\build\dlib\Release
- Linker -> Input:
 - add Additional Dependencies: dlib19.15.0_release_64bit_msvc1915.lib
- Add D:\dlib-19.15\dlib\all\Source.cpp to your Source Files(Header Files, Resource Files or Source Files)



- Run project with dlib in Release mode, x64.

Use image jpg in project with dlib

- Add all file in folder D:\dlib-19.15\dlib\external\libjpeg to your project(Header Files, Resource Files or Source Files)



- Right click in your project -> Properties:
-C/C++ -> Preprocessor -> Preprocessor Definitions -> Add: DLIB_JPEG_SUPPORT
- If you want use image png file, do the same.

Detect and Face recognition

- Link source code: <https://github.com/freedom1810/Detect-and-Face-recognition>
- How to use:
 - Download source code to your computer
 - Create a new empty project
 - Link project to opencv and dlib you installed.
 - Copy all code file(.h, .cpp) you downloaded to your project
- Build it in Release mode and x64.
- Copy folder data_train, dlib_face_recognition_resnet_model_v1.dat, shape_predictor_5_face_landmarks.dat to folder: ..\x64\Release
- Open command line in this folder and call this program like this:
"Detect and Face recognition.exe" data_train

```
C:\Windows\System32\cmd.exe - "Detect and Face recognition.exe" data_train
Microsoft Windows [Version 10.0.17134.472]
(c) 2018 Microsoft Corporation. All rights reserved.

D:\study\20181\Project1\code\Detect and Face recognition\x64\Release>"Detect and Face recognition.exe" data_train
Do you want to train ? [Y/n]:
```


- True Y to train your model, it will be save to svm.dat. If you have trained it, no need to do it again.
- Camera will be opened, turn off the camera it will be detect and recognition your face.(It will need data your face in folder data_train(which each person we need 11 picture, each picture has same size(150 x 150))
- This program will expect to be given a directory structured as follows:

data_train/

 person1/

 image1.jpg

 image2.jpg

 image3.jpg

 person2/

 image4.jpg

 image5.jpg

 image6.jpg

 person3/

 image7.jpg

 image8.jpg

 image9.jpg

- It already 6 people in folder data_train, add picture of who you want to recognition and retrain model.