

\*\* 미리 **visual\_studio 2015** 가 깔려있어야 합니다 ! 인스톨시 가급적, **전체설치** 권장합니다. 거의 50GB 되니 용량 충분히 확보후 인스톨합니다.

(전체설치 안할 경우, 디버깅시 오류가 많이 뜰 수 있습니다)

\*\* OpenCV 라이브러리는.. 환경변수 설정으로 간단 사용 가능합니다.

다만 .. OpenCV 는 오픈소스 이기 때문에 전세계 개발자들에 의해.. 하루가 다르게 최신라이브러리가 추가 됩니다. (특히, OpenCV\_contrib 폴더의 파일들)

이것을 포함하여 추가하고자 하는 것입니다^^

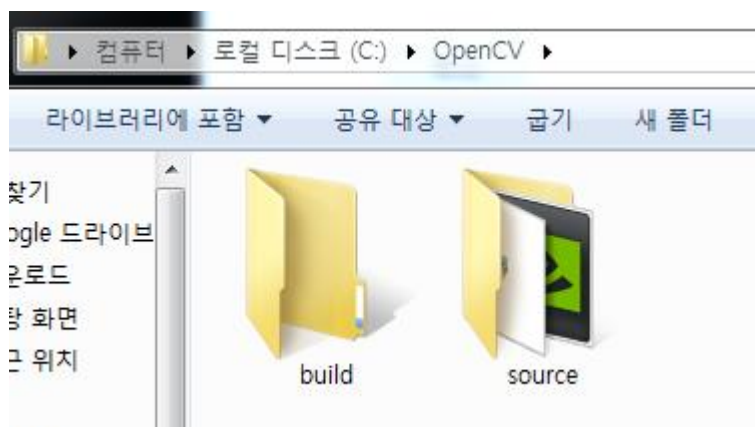
- 빨간색 글자는, 강조할 사항, 주의할 사항, 실행해야 하는 사항, 표시한것 입니다
- 연두색 글자는, 그저...참고사항입니다. 읽어두시면 도움이 됩니다
- 파랑색 글자는, 진행하기 위한 사항 또는 실행하기 위해 필수로 알아야하는 내용입니다.
- 검정색 **강조된** 글자는, 키워드 입니다

=====

1. C 드라이브 밑에 **OpenCV** 폴더를 하나 만듭니다

그리고, OpenCV 폴더의 하위폴더로, **build** 폴더와 **source** 폴더를 생성합니다

- build 폴더는 나만의 라이브러리 구성하기 위한 폴더입니다
- source 폴더는, github 등 에서 Download 한, 최신라이브러리 보관하는 폴더입니다



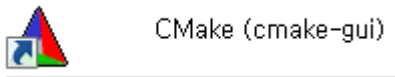
2. **source** 폴더 안에 **다음파일들**을 **압축해제**하여 옮깁니다

- opencv\_contrib-master
- opencv-master
- tbb2017\_20170412oss\_win

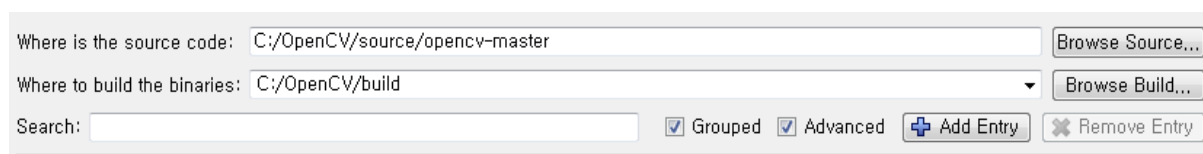
3. 인스톨 해야하는 파일은 **인스톨** 합시다

- cmake 인스톨 (무조건 next 넘겨버립니다)
- cuda 인스톨 (무조건 next 넘겨버립니다)

4. 인스톨이 잘 되었다면, **cmake** 를 **실행**시킵니다.



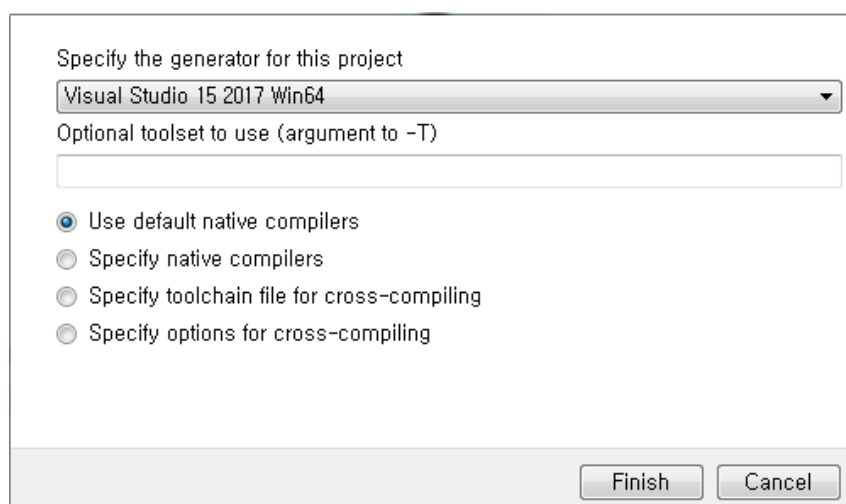
5. cmake 의 빈칸을 (**Browse... 열어서**) 다음과 같이 채웁니다



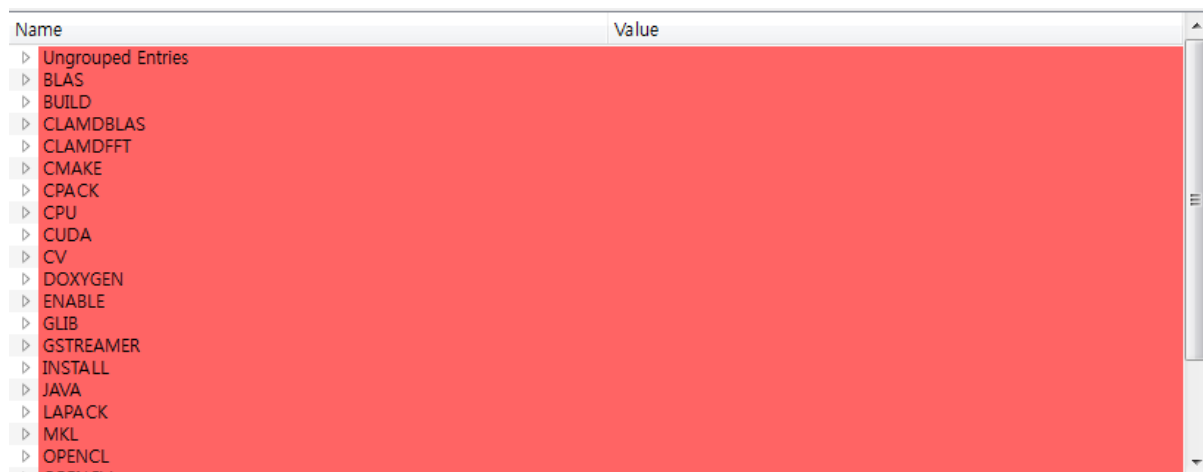
입력완료 후, 하단의 **Configure** 를 클릭합니다.

**자신이 64bit 컴퓨터이다.** 하면 , 반드시 **Win64** 버전의 컴파일러를 매칭시켜 줘야합니다. 컴파일러를 영뚱한거 가져다가 cmake 하고.. 에러가 엄청 날수 있습니다.

**Use default native compilers** 선택 > **Finish** 클릭



그러면 다음과 같이 위에는 빨간창이...



하단에는 이러한 정보가 쪽~출력 됩니다

이는 버전이 맞게 깔렸는지, 설정한 내용이 맞는지 확인하는 용도로 사용하세요.

cmake 가 아래와 같이 자동으로 찾아서 넣어준 것입니다.

```
General configuration for OpenCV 3.2.0-dev

=====

Version control: unknown

Platform:

Host: Windows 6.1.7601 AMD64

CMake: 3.8.1

CMake generator: Visual Studio 15 2017 Win64

CMake build tool:

C:\Program Files (x86)\Microsoft Visual Studio 2017\Community\MSBuild\15.0\Bin\MSBuild.exe

C/C++: Built as dynamic libs?: YES

C++ Compiler: C:/Program Files (x86)/Microsoft Visual Studio/2017/Community/VC/Tools/MSVC/
14.10.25017/bin/HostX86/x64/cl.exe (ver 19.10.25019.0)

C Compiler: C:/Program Files (x86)/Microsoft Visual Studio/2017/Community/VC/Tools/MSVC/
14.10.25017/bin/HostX86/x64/cl.exe

Linker flags (Release): /machine:x64 /INCREMENTAL:NO /debug
```

Linker flags (Debug): /machine:x64 /debug /INCREMENTAL

ccache: NO

Precompiled headers: YES

Extra dependencies: comctl32 gdi32 ole32 setupapi ws2\_32 vfw32 cudart nppc nppi npps cufft -

LIBPATH:C:/Program Files/NVIDIA GPU Computing Toolkit/CUDA/v8.0/lib/x64

3rdparty dependencies:

OpenCV modules:

To be built: cudev core cudaarithm flann imgproc ml objdetect video cudabgsegm cudafilters cudaimgproc

cudawarping imgcodecs photo shape videoio cudacodec highgui ts features2d calib3d cudafeatures2d cudalegacy

cudaobjdetect cudaoptflow cudastereo stitching superres videostab

### **Disabled: world**

Disabled by dependency: -

Unavailable: java python2 python3 viz

Windows RT support: NO

GUI:

Win32 UI: YES

Media I/O:

ZLib: build (ver 1.2.8)

JPEG: build (ver 90)

WEBP: build (ver encoder: 0x020e)

PNG: build (ver 1.6.24)

TIFF: build (ver 42 - 4.0.2)

JPEG 2000: build (ver 1.900.1)

OpenEXR: build (ver 1.7.1)

Video I/O:

Video for Windows: YES

FFMPEG: YES (prebuilt binaries)

avcodec: YES (ver )

avformat: YES (ver )

avutil: YES (ver )

swscale: YES (ver )

avresample: YES (ver )

DirectShow: YES

Parallel framework: Concurrency

Other third-party libraries:

Use Cuda: YES (ver 8.0)

Use OpenCL: YES

NVIDIA CUDA

Use CUFFT: YES

NVIDIA GPU arch: 20 30 35 37 50 52 60 61

OpenCL: <Dynamic loading of OpenCL library>

Include path: C:/OpenCV/source/opencv-master/3rdparty/include/opencv/1.2

Tests: YES

Performance tests: YES

Install path: C:/OpenCV/build/install

cvconfig.h is in: C:/OpenCV/build

-----

Configuring done // 성공적으로 완료했음

## 5.1 아래 설정을 하고 **Configuring** 할 것입니다

### 5.1.1 위의 빨간 창에서 **WITH** 를 찾는다

- **WITH\_CUDA** 항목에 **체크**
- **WITH\_TBB** 항목에 **체크**

### 5.1.2 위의 빨간 창에서 **BUILD** 를 찾는다

- 아래 항목을 제외한 **나머지를 모두 체크**합니다

BUILD\_CUDA\_STUBS

BUILD\_TBB

BUILD\_WITH\_DYNAMIC\_IPP

BUILD\_opencv\_world

BUILD\_contrib\_world(이 항목은 없을수도 있음)

### 5.1.3 이번에는 **OPENCV** 를 찾는다

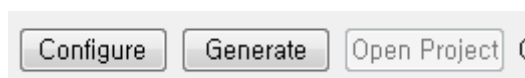
- **OPENCV\_EXTRA\_MODULES\_PATH** 의 경로를 다음과같이 **Browse** 창 띄워서 **찾아넣습니다**

(contrib 라이브러리를 추가하여 코드분석 해야 겠죠..? ^^)

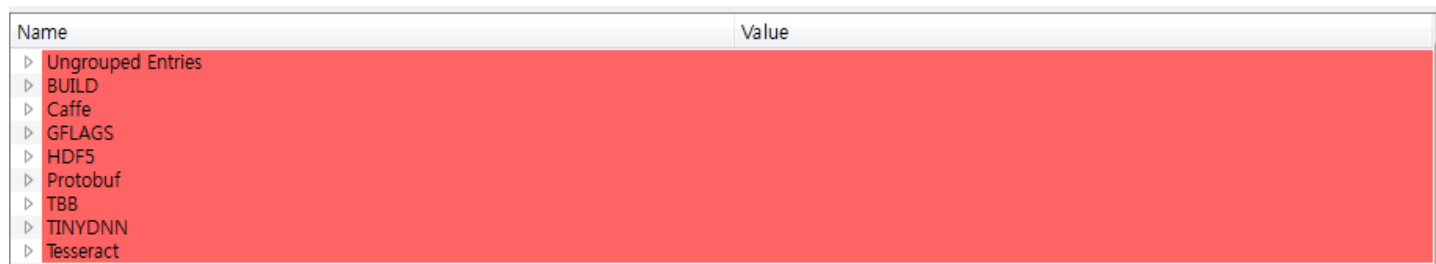
**C:/OpenCV/source/opencv\_contrib-master/modules**

OPENCV	
OPENCV_CONFIG_FILE_INCLUDE_DIR	C:/OpenCV/build
OPENCV_DOWNLOAD_PATH	C:/OpenCV/source/opencv-master/.cache
OPENCV_ENABLE_NONFREE	<input type="checkbox"/>
OPENCV_EXTRA_MODULES_PATH	C:/OpenCV/source/opencv_contrib-master/modules
OPENCV_WARNINGS_ARE_ERRORS	<input type="checkbox"/>

### 5.1.4 **Configure** 를 클릭합니다



5.2 그러면 다음과 같이 빨간 화면이 또 뜹니다.. 설정해달라.. 라는 의미겠죠 ㅎㅎ



### 5.2.1 주소 찾아 넣기전에.. 먼저 폴더를 하나 지웁시다

(VS 에서 컴파일시 에러나는 부분이라 삭제하는것 권장)

- OpenCV / source / opencv\_contrib-master / modules / **bioinspired**      **폴더삭제**

### 5.2.2 그다음으로 TBB 항목을 설정해줍시다

TBB 와 관련된 설정을 자동으로 찾지 못해서 빨간색으로 뜬 것이므로, **수동으로 설정을 해줘야함.**

TBB	
TBB_ENV_INCLUDE	TBB_ENV_INCLUDE-NOTFOUND
TBB_ENV_LIB	TBB_ENV_LIB-NOTFOUND
TBB_ENV_LIB_DEBUG	TBB_ENV_LIB_DEBUG-NOTFOUND

- **TBB\_ENV\_INCLUDE**    C:\OpenCV\source\tbb2017\_20170412oss\_win\include
- **TBB\_ENV\_LIB**    C:/OpenCV/source/tbb2017\_20170412oss\_win/lib/intel64/vc14/tbb.lib
- **TBB\_ENV\_LIB\_DEBUG**    C:/OpenCV/source/tbb2017\_20170412oss\_win/lib/intel64/vc14/tbb\_debug.lib
- **Configure** 를 클릭합니다

### 5.2.3 그러면, 또 다시 빨간 화면이 뜨는데,

이것은 기존과는 다르게.. 초기설정이어서 경로가 맞는지 한번 더 확인하는 것입니다.

Name	Value
Ungrouped Entries	
<b>TBB_VER_FILE</b>	C:/OpenCV/source/tbb2017_20170412oss_win/include/tbb/tbb...

경로가 맞는지 한번 더 체크합니다

- **TBB\_VER\_FILE**    C:/OpenCV/source/tbb2017\_20170412oss\_win/include/tbb/tbb\_stddef.h
- **Configure** 를 클릭합니다

#### 5.2.4 그러면.. 빨간창이 뜰수도 안뜰수도 있는데, 일단은 무시해도 좋습니다

여기에서 Search 란에 [test] 를 입력하여 검색 -> **뜨는 것 체크 해제** 합니다

Search: test	<input checked="" type="checkbox"/> Grouped	<input checked="" type="checkbox"/> Advanced	<input type="button" value="+ Add Entry"/>	<input type="button" value="✕ Remove Entry"/>
Name	Value			
▴ BUILD				
BUILD_PERF_TESTS	<input type="checkbox"/>			
BUILD_TESTS	<input type="checkbox"/>			
▾ INSTALL				

- **Configure** 를 클릭합니다
- **configure 후 (generate 전) :**

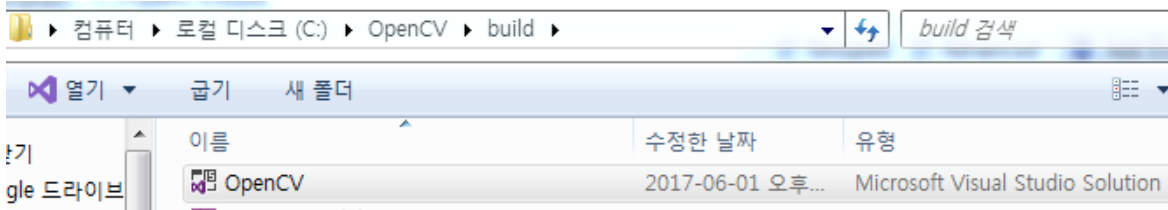
**opencv\_contrib/modules/bioinspired 폴더 삭제** - 오류를 사전제거

- 성공적으로 끝났다면, **Generating** 을 클릭합니다
- 맨 하단에 **Configuring done / Generating done** 이라는 글자가 확인된다면, 제대로 cmake 수행이 된 것입니다.



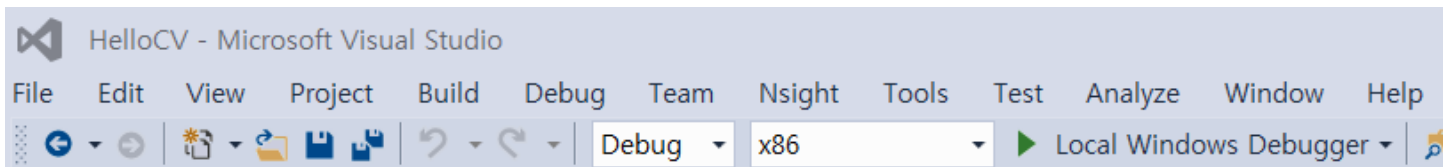
6. 그러면 이제 프로젝트를 열어봅시다.

OpenCV / build / **OpenCV.sln** 더블클릭 (확장자가 안보이면, 아래의 아이콘을 찾자)



## 6.1 visual studio 창에서

6.1.1 (메뉴 표시줄 글자들 아래에 있는) **debug** 선택 & **그 옆에는** 자신의 컴에 맞는 **x64** 또는 **x86** 설정



6.1.2 솔루션 탐색기에서 install 찾습니다 -> **install** 위에 마우스커서를 놓고 **오른쪽 클릭**

-> 맨 위 **Build** 클릭

**\*\*참고\*\***

**\*\* 이 작업은, 이전에 우리가 받아놓은 소스코드들이 있는**

**opencv / source / opencv\_master** 폴더에서 ,

**(컴이 디버깅 또는 릴리즈 컴파일을 수행하면서)**

**우리가 쓸 opencv / build** 폴더 안에 **install** 이라는 폴더를 생성하여,

**이 폴더에, include 파일들 복사, lib 파일들 복사, dll 파일들 복사 - 한군데에 모아주는 수행을 하고,**

**이 폴더를 우리의 프로젝트의 설정창에서 경로설정을 해줌으로써, 실질적으로 사용할수 있게 한다**

**\*\*\*\*\***

만약 빌드가 완료되었는데, 에러가 하나라도 뜬다면,

1 . 반드시 Visual Studio 2015 를 설치했는지 확인합니다

2 . 5 번에서 CMake 수행시 Visual Studio 2015 x64 (자신의 컴퓨터 상황에 맞게 선택했는지) 확인합니다

3 . CMake 에서 선택항목을 잘 체크해제하고, 체크해야 할 것은 했는지 확인합니다

4 . CMake 에서 경로설정을 잘 했는지 확인합니다

5 . 컴퓨터를 껐다가 켜서 다시한번 수행해봅니다

5 . 그래도 안 될 경우에는 .. 컴터를 밀고 다시 처음부터 수행합니다

(컴파일러가 잘못 설치되서 잘못 사용하게 될 경우.. 굉장히 많은 오류가 발생합니다)

6.1.3 빌드가 완료되고, 에러가 하나도 뜨지 않는다면 , 다음을 수행합니다

맨 위 메뉴줄의 File -> New -> Project

=> visual C++ > win32 console application > name 임의로 입력 > next

> empty project 체크 > OK

Solution Explorer 창에서

=> Source Files 위에서 오른쪽 클릭 > add > new item

> c++.cpp 파일 만들기 선택하고, name 변경

6.1.4 다음의 예제코드가 잘 실행되는지 확인하겠습니다

새로 생성한 .cpp 파일에 복-붙 합니다

```
#include "opencv2/opencv.hpp"
#include <iostream>

using namespace cv;
using namespace std;

int main(void)
{
    Mat img = imread("lenna.bmp");

    if (img.empty()) {
```

```

        cerr << "Image load failed!" << endl;
        return -1;
    }

    cout << "width: " << img.cols << "height: " << img.rows << endl;

    namedWindow("image");
    imshow("image", img);

    waitKey(0);

    return 0;
}

```

그리고, "lenna.bmp" 이러한 이름의 bmp 파일을 하나 build 폴더에 넣읍시다.

그러면 이렇게 빨간 줄이 보이는데,

```

#include "opencv2/ope
#include <iostream>

```

이는 경로설정이 되질 않아, opencv2 폴더를 찾지 못해서 입니다.

설정해주도록 합시다

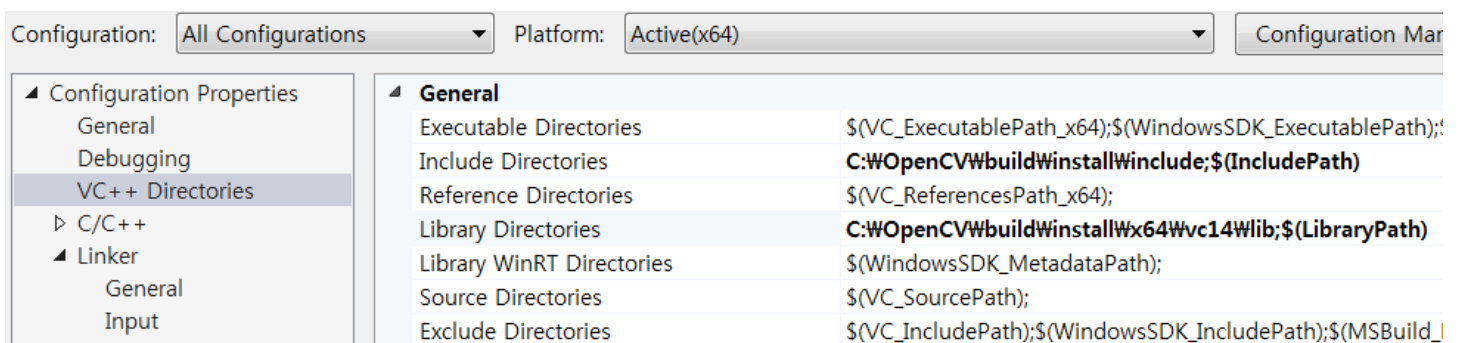
[솔루션 익스플로러] 의 **프로젝트 이름** 위에서 , **마우스 오른쪽 클릭** -> **properties(속성)** 클릭

#### 6.1.5 우리가 만든 build 폴더의 경로를 설정해줍시다

Configuration : **All Configuration** 상태에서

**VC++ Directories > Include Derectories** : **OpenCV / build / install / include**

**VC++ Directories > Library Derectories** : **OpenCV / build / install / x64 / vc14 / lib**



설정을 일단 **적용**해줍시다.

6.1.6 그리고, 일단 당장 쓸 라이브러리 추가하겠습니다.

다시한번, [솔루션 익스플로러] 의 **프로젝트 이름** 위에서 , **마우스 오른쪽 클릭** -> **properties(속성)** 클릭  
이번에는 **Configuration : Active(Debug) 상태** 로 두고 ..

**Linker > Input > Additional Dependencies** 에 **다음의 라이브러리 목록을 붙여넣기** 합시다

```
opencv_core320d.lib  
opencv_highgui320d.lib  
opencv_imgcodecs320d.lib  
opencv_imgproc320d.lib  
opencv_videoio320d.lib
```

두번째로 **Configuration : Active(Release) 상태** 로 두고 ..

**Linker > Input > Additional Dependencies** 에 **다음의 라이브러리 목록을 붙여넣기** 합시다

```
opencv_core320.lib  
opencv_highgui320.lib  
opencv_imgcodecs320.lib  
opencv_imgproc320.lib  
opencv_videoio320.lib
```

설정을 **적용**해줍니다.

#### 6.1.7 내컴퓨터 오른쪽클릭 > 속성 > 고급시스템설정 > 환경변수

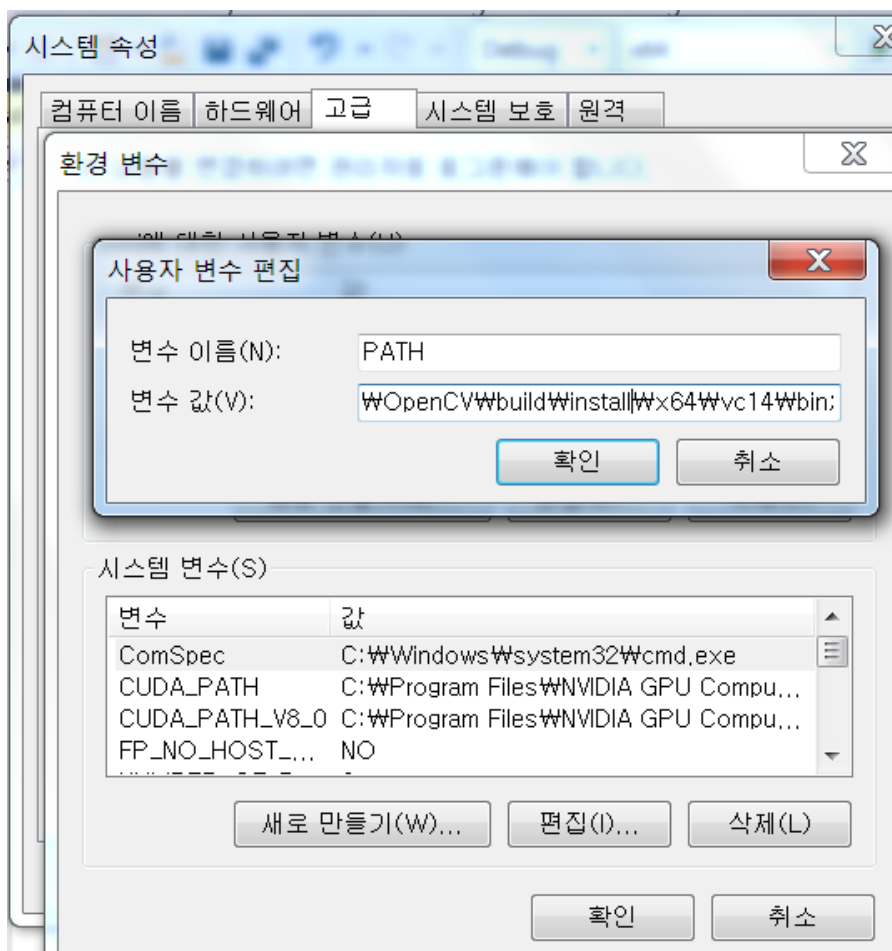
> PATH 더블클릭

이 곳에, 다음 경로를 탐색기에서 찾은 뒤 .. 경로복사하여 넣습니다

**OpenCV / build / install / x64 / vc14 / bin;**

**이 설정은 초기에 한번만 수행합니다. 참고하세요.**

**이 설정을 수행하고 컴퓨터를 Reboot 하여 사용합니다.**



#### 6.1.8 TBB 관련 dll 파일을 복사해서 붙여넣기 합니다

복사하기 : OpenCV / source / tbb2017\_20170604oss\_win / bin / intel64 / vc14 의 모든 파일

붙여넣을 곳 : OpenCV / build / install / x64 / vc14 / bin

#### 6.1.9 컴파일이 정상적으로 되는지 확인합니다

**Local Windows Debugger 클릭 (▶ 디버거 클릭)**

레퍼런스 : <http://darkpgmr.tistory.com/102> 참고하세요.

작성한대로 그대로 따라하시면, [최신 논문을 통해 구현된 라이브러리들] 을 사용할 수 있게 됩니다

만약 에러가 발생한다면, 그것을 찾아보는 것 또한 자신의 공부가 됩니다

빌드되는 과정을 이해하고, 어떤 폴더에 어떤 파일들이 모이고, 에러가 발생하는 이유가 링크가 연결되지

않아서인지, 코드상에 문제가 있어서 일부분 삭제를 한 뒤에 컴파일을 해야하는 지 .. 등등

찾아가면서 배우시게 됩니다

만약 찾아보았는데도, 문제해결이 되지 않았다면

1. 질문할 경우에는 , 에러코드 나 에러메세지 , 그 당시의 상황 등 구체적인 사항을 가지고 질문하면 좋겠습니다. 나를 위해 상대방이 봐주는 그 시간 또한 배려하는 마음을 가집시다.
2. 꼭 해야 하는 것이 아니라면, 다른 방법으로 우회하는 것 또한 하나의 좋은 방법입니다
3. 에러를 해결했다면, 공유문서화 시키는 것도 .. 이 문서를 보는 다음사람을 위한 큰 배려입니다.

=====

=====

< 에러를 발견 / 혹은 해결하신 분은, 아래에 해당 내용을 공유해주시면 감사하겠습니다 >