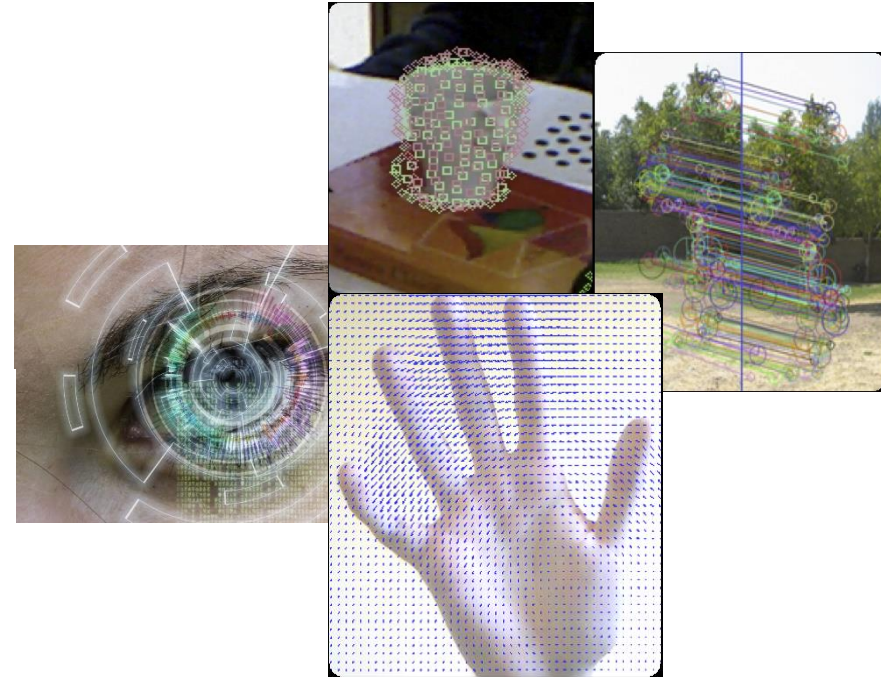


2023 Autumn

COMPUTER VISION

비전
프로그래밍



1.2 Open CV Installation (Full modules)

1.2 Open CV Build and Installation

■ Open CV Modules

OpenCV modules

Main (Basic) modules

core. [Core functionality](#)
imgproc. [Image processing](#)
imgcodecs. [Image file reading and writing](#)
videoio. [Video I/O](#)
highgui. [High-level GUI](#)
video. [Video Analysis](#)
calib3d. [Camera Calibration and 3D Reconstruction](#)
features2d. [2D Features Framework](#)
objdetect. [Object Detection](#)
dnn. [Deep Neural Network module](#)
ml. [Machine Learning](#)
flann. [Clustering and Search in Multi-Dimensional Spaces](#)
photo. [Computational Photography](#)
stitching. [Images stitching](#)
:
superres. [Super Resolution](#)
videostab. [Video Stabilization](#)
viz. [3D Visualizer](#)

Extra modules

aruco. [ArUco Marker Detection](#)
bgsegm. [Improved Background-Foreground Segmentation Methods](#)
bioinspired. [Biologically inspired vision models and derived tools](#)
ccalib. [Custom Calibration Pattern for 3D reconstruction](#)
cnn_3dobj. [3D object recognition and pose estimation API](#)
cvv. [GUI for Interactive Visual Debugging of Computer Vision Programs](#)
datasets. [Framework for working with different datasets](#)
dnn_modern. [Deep Learning Modern Module](#)
dnn_objdetect. [DNN used for object detection](#)
dpm. [Deformable Part-based Models](#)
face. [Face Analysis](#)
:
xfeatures2d. [Extra 2D Features Framework](#)
ximgproc. [Extended Image Processing](#)
xobjdetect. [Extended object detection](#)
xphoto. [Additional photo processing algorithms](#)

1.2 Open CV Build and Installation

■ Open CV Modules

- **Main modules** : Basic functions for general computer vision problems ([Releases - OpenCV](#)).
- **Extra modules** : Advanced functions with patent or license legally for many recognition problems such as feature matchings (SIFT, FAST, SURF and so on) and deep learning techniques (https://github.com/opencv/opencv_contrib).



We need all modules for our class....!!!

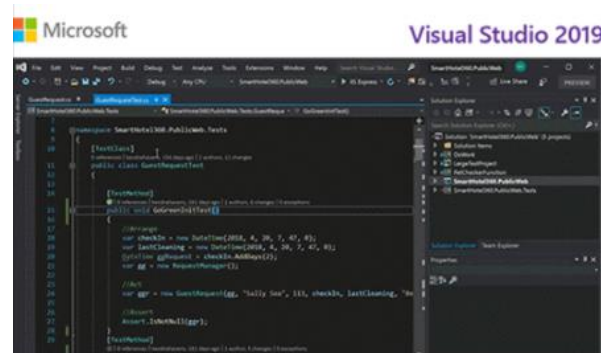
1.2 Open CV Build and Installation

■ Preparation

cmake 3.18.2



OpenCV 4.4



Visual Studio 2019 community

1.2 Open CV Build and Installation

■ Open CV Installation

■ Preparation

- Download source **extra modules** (opencv_contrib-4.1.1) and **original source** (opencv-4.8.0) from <https://opencv.org/releases> and https://github.com/opencv/opencv_contrib.
- To build, we need CMake for Windows (<http://www.cmake.org/download/>) and Microsoft Visual Studio (in this case, I used Visual Studio 2017/2019 community in 64-bit PC).

4.8.0 is possible.

The screenshot shows the CMake website's download page. It features two tables: one for source distributions and one for binary distributions. The binary distributions table has a red box highlighting the 'Windows win64-x64 Installer' row, which includes a warning: 'Installer tool has changed. Uninstall CMake 3.4 or lower first!'.

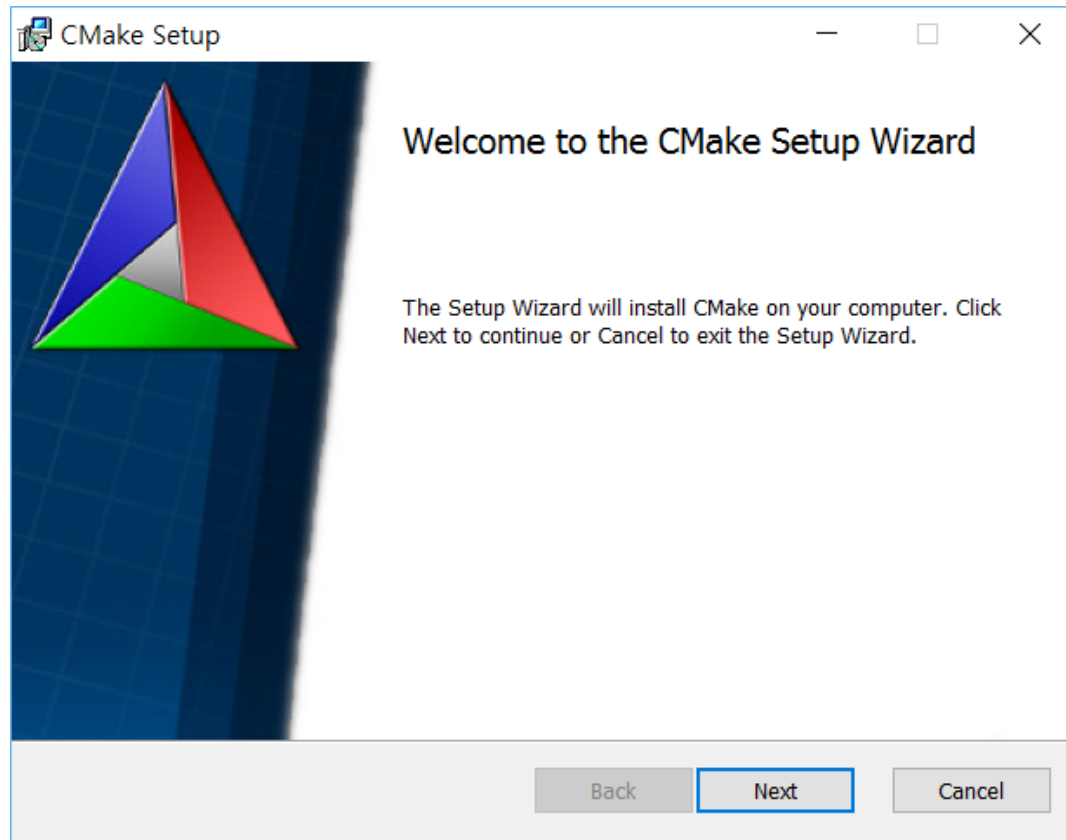
Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.15.2.tar.gz cmake-3.15.2.tar.Z
Windows Source (has \r\n line feeds)	cmake-3.15.2.zip

Binary distributions:

Platform	Files
Windows win64-x64 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first!	cmake-3.15.2-win64-x64.msi
Windows win64-x64 ZIP	cmake-3.15.2-win64-x64.zip
Windows win32-x86 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first!	cmake-3.15.2-win32-x86.msi
Windows win32-x86 ZIP	cmake-3.15.2-win32-x86.zip
Mac OS X 10.7 or later	cmake-3.15.2-Darwin-x86_64.dmg cmake-3.15.2-Darwin-x86_64.tar.gz
Linux x86_64	cmake-3.15.2-Linux-x86_64.sh cmake-3.15.2-Linux-x86_64.tar.gz

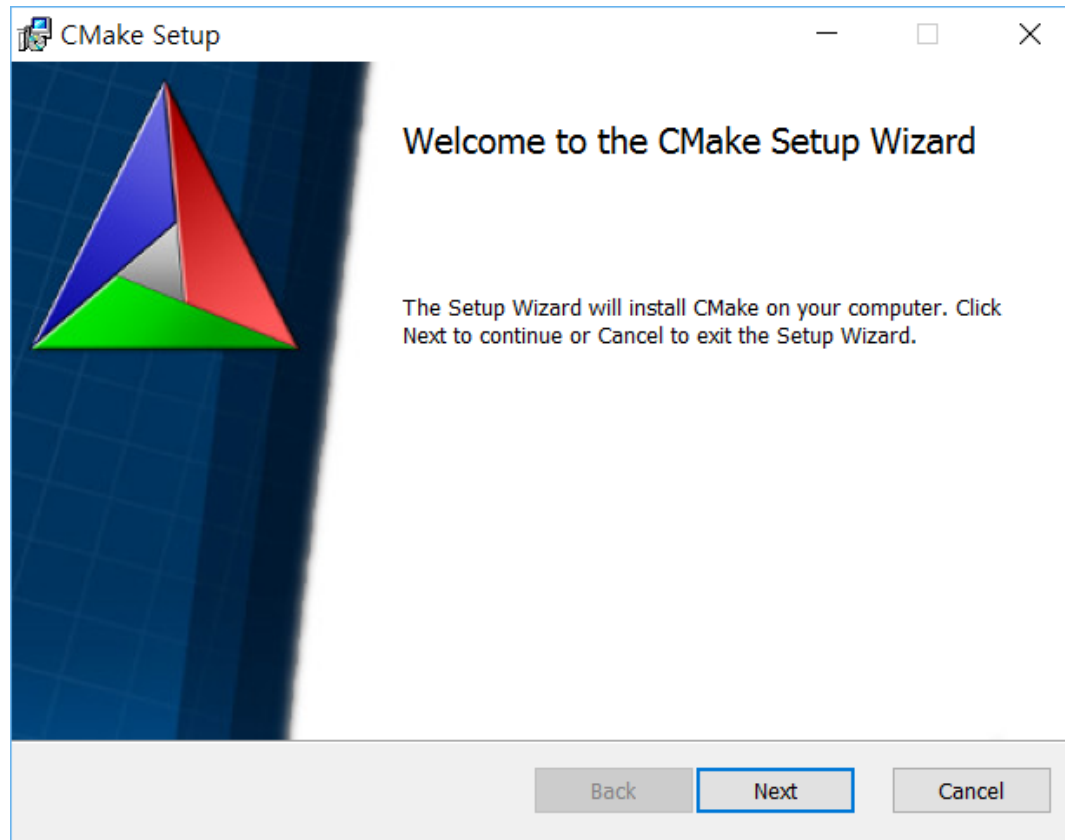
1.2 Open CV Build and Installation

- Open CV Installation
 - Install C-Make first...!!!

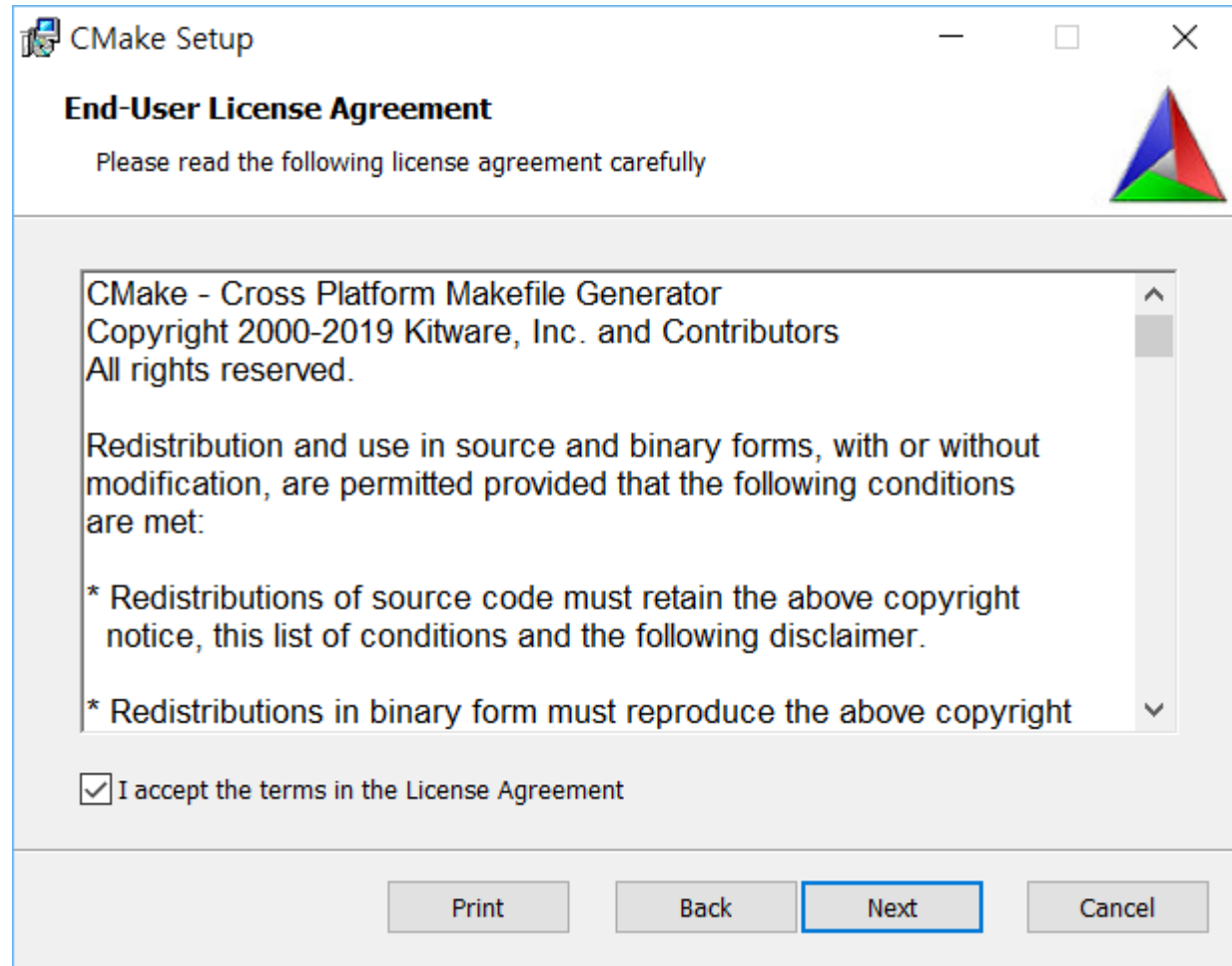


1.2 Open CV Build and Installation

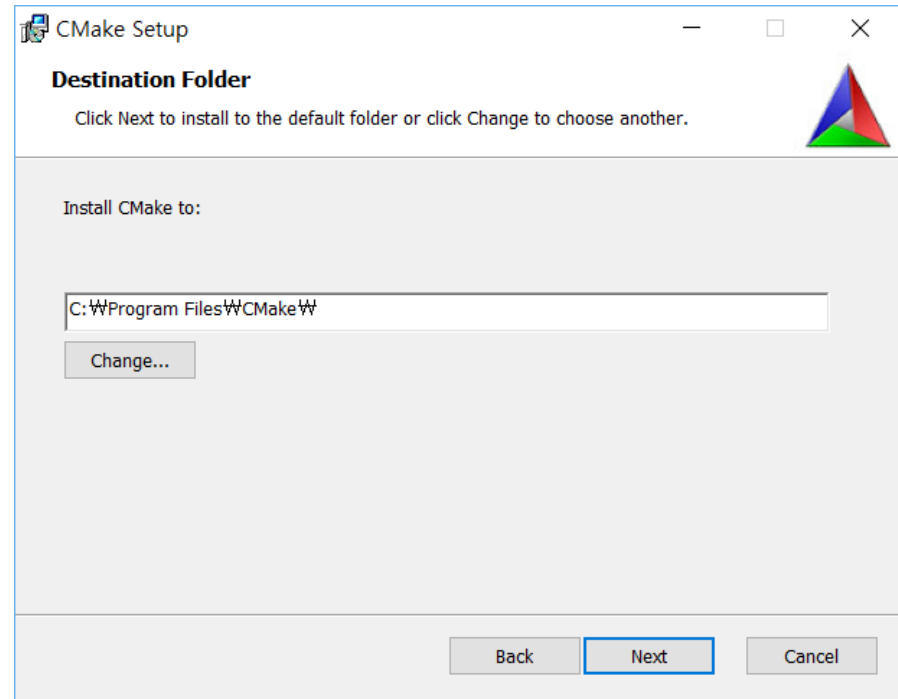
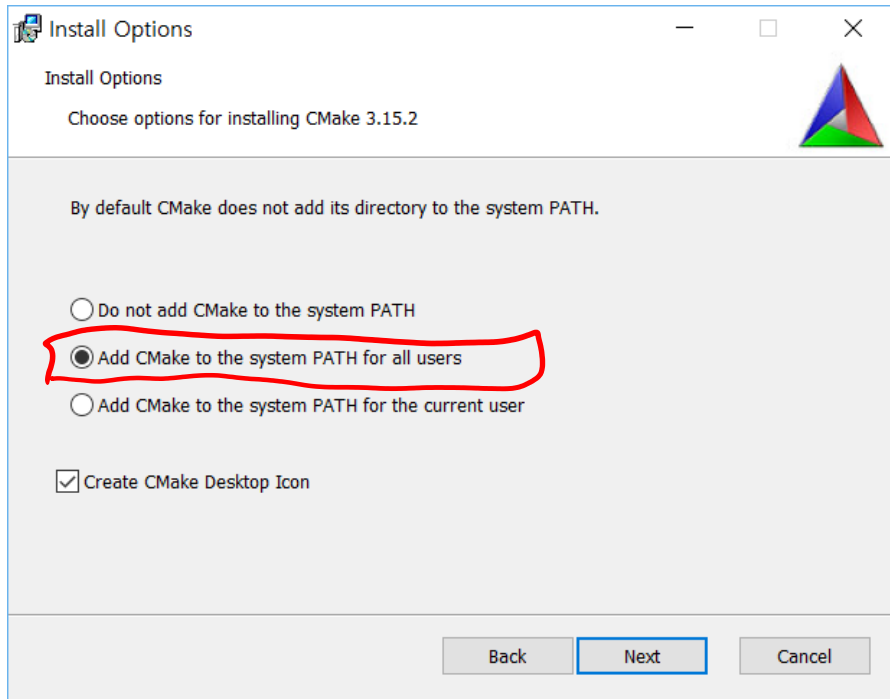
- Open CV Installation
 - Install C-Make first...!!!



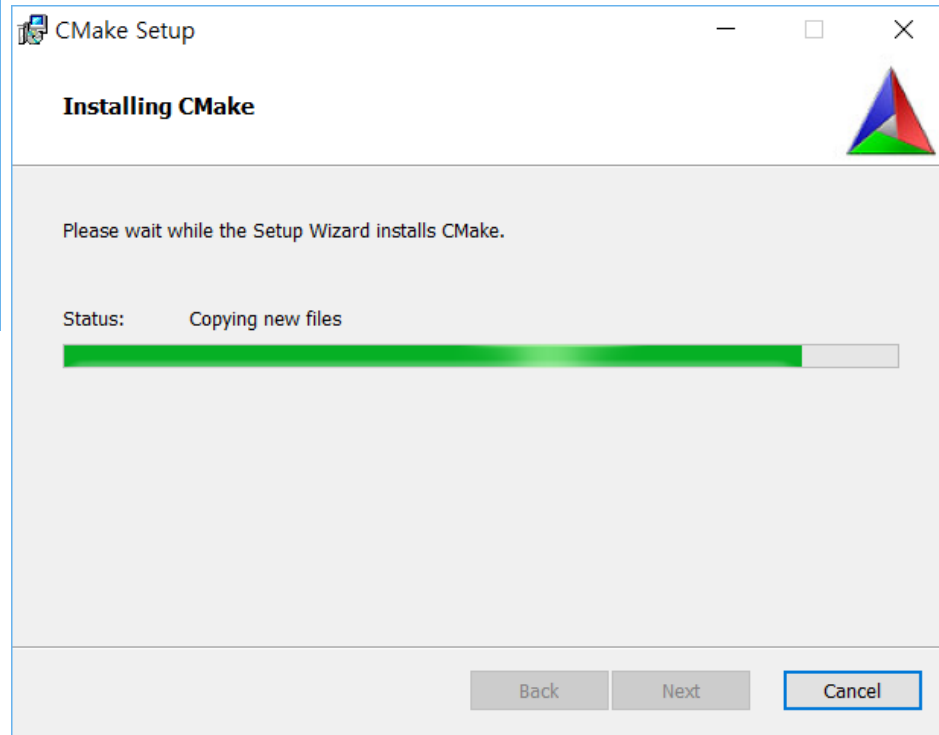
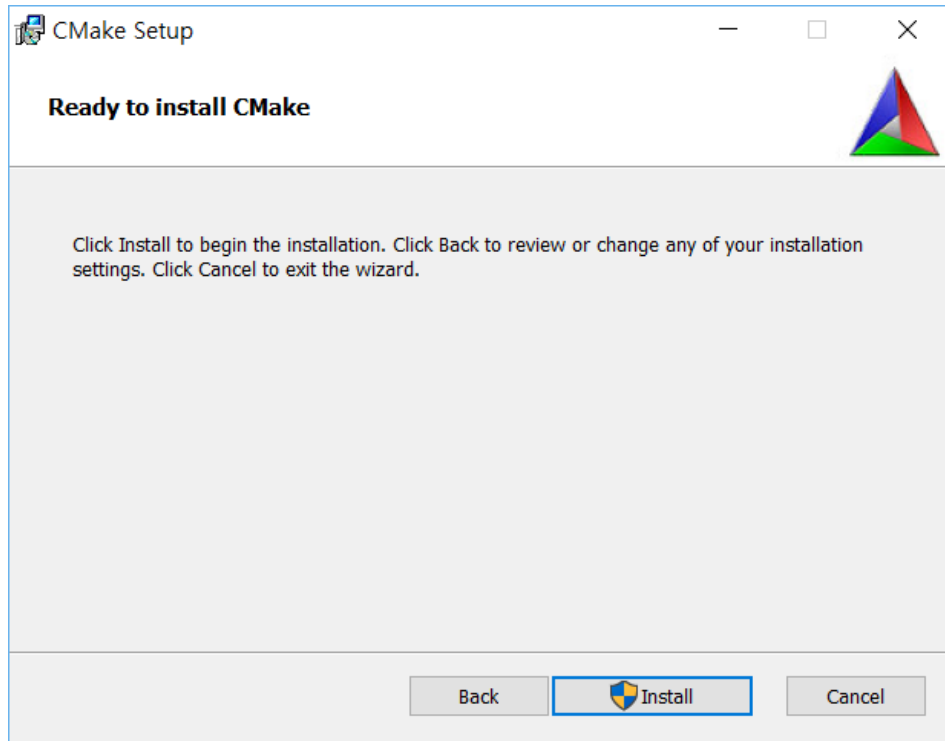
1.2 Open CV Build and Installation



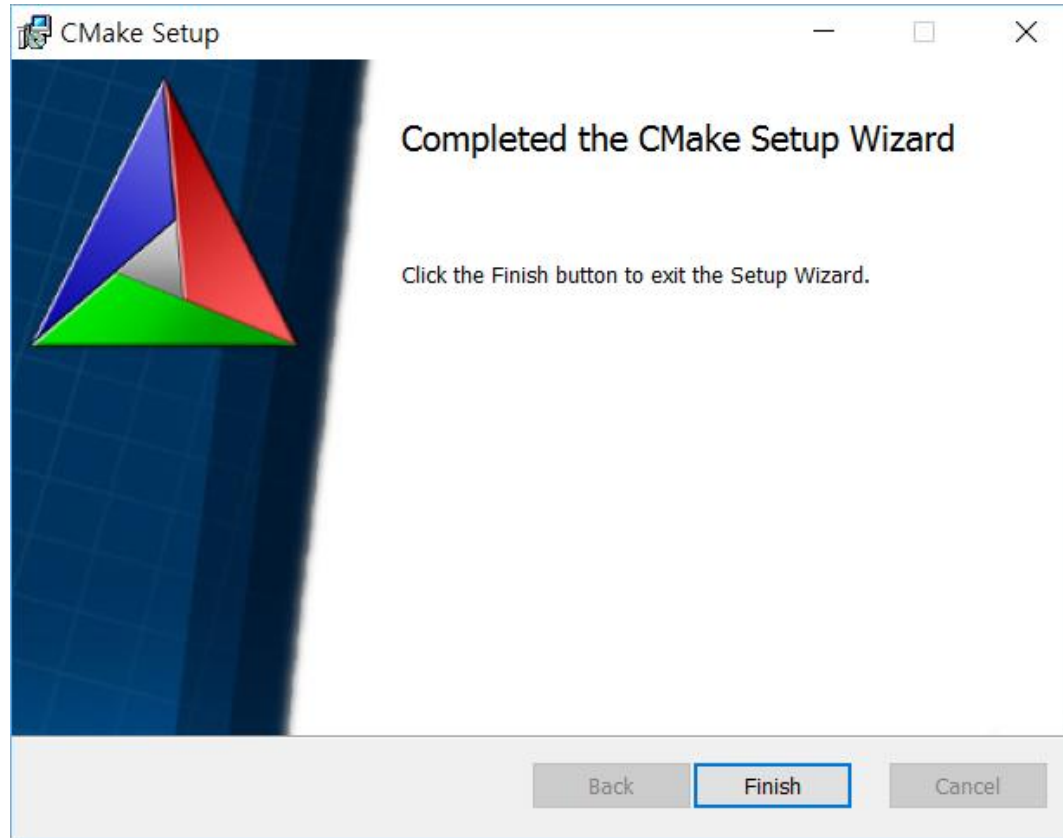
1.2 Open CV Build and Installation



1.2 Open CV Build and Installation

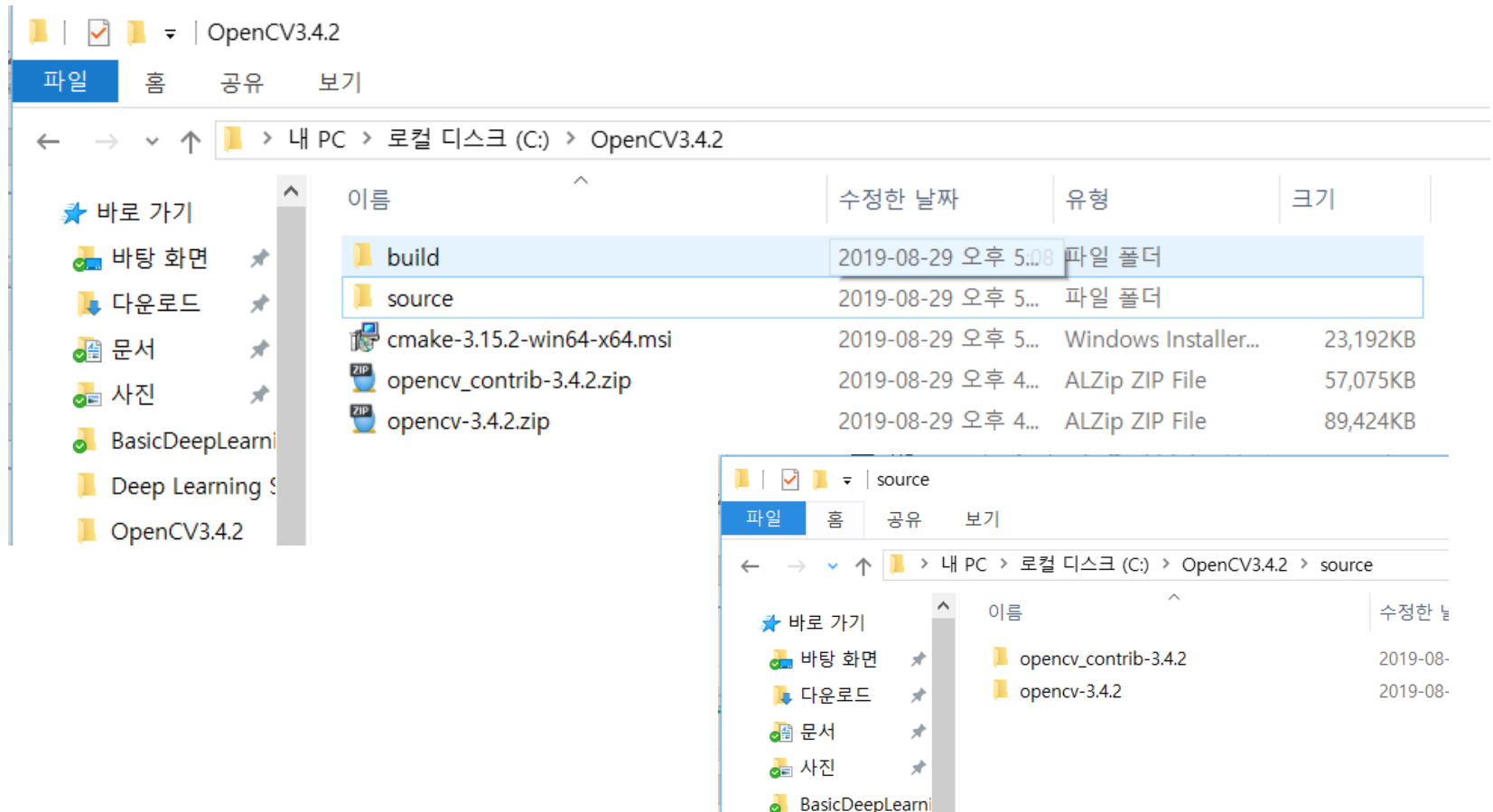


1.2 Open CV Build and Installation



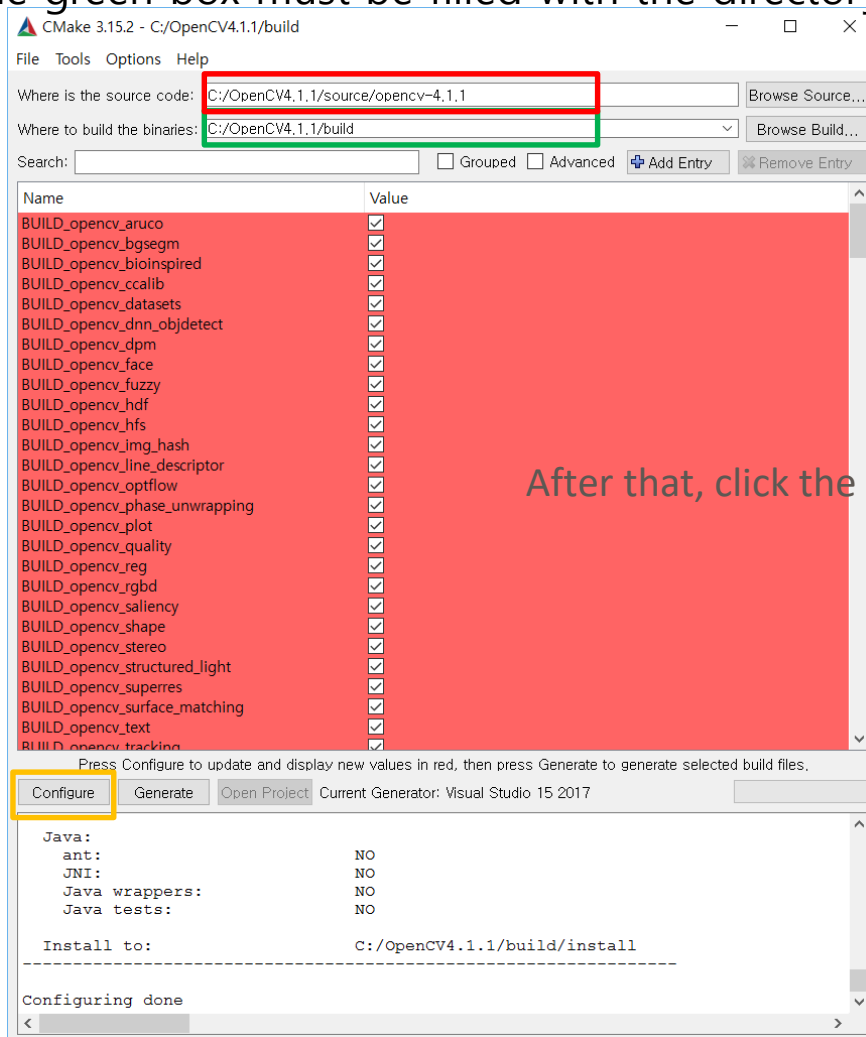
1.2 Open CV Build and Installation

- Make folders
 - build: for saving the result of building process
 - source : contain all of the OpenCV and its extra modules sources.



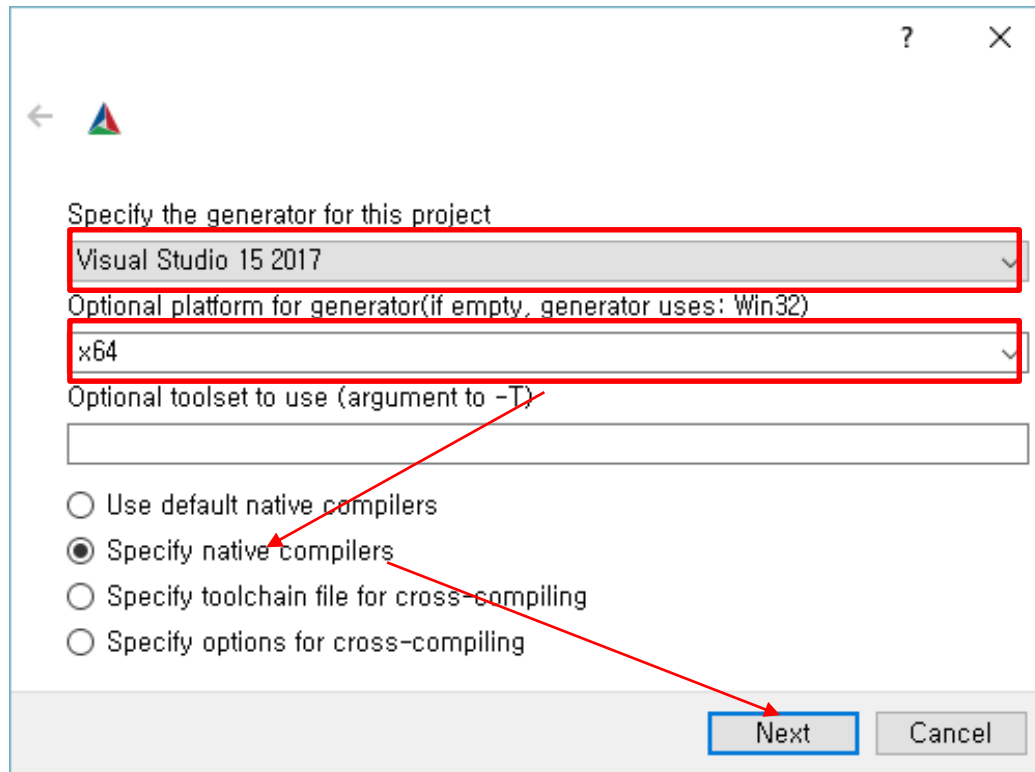
1.2 Open CV Build and Installation

- launch CMake application and then specify the source and build directory as shown in figure below. The red box must be filled with the directory path of OpenCV source, and the green box must be filled with the directory path of designated *build* folder.



1.2 Open CV Build and Installation

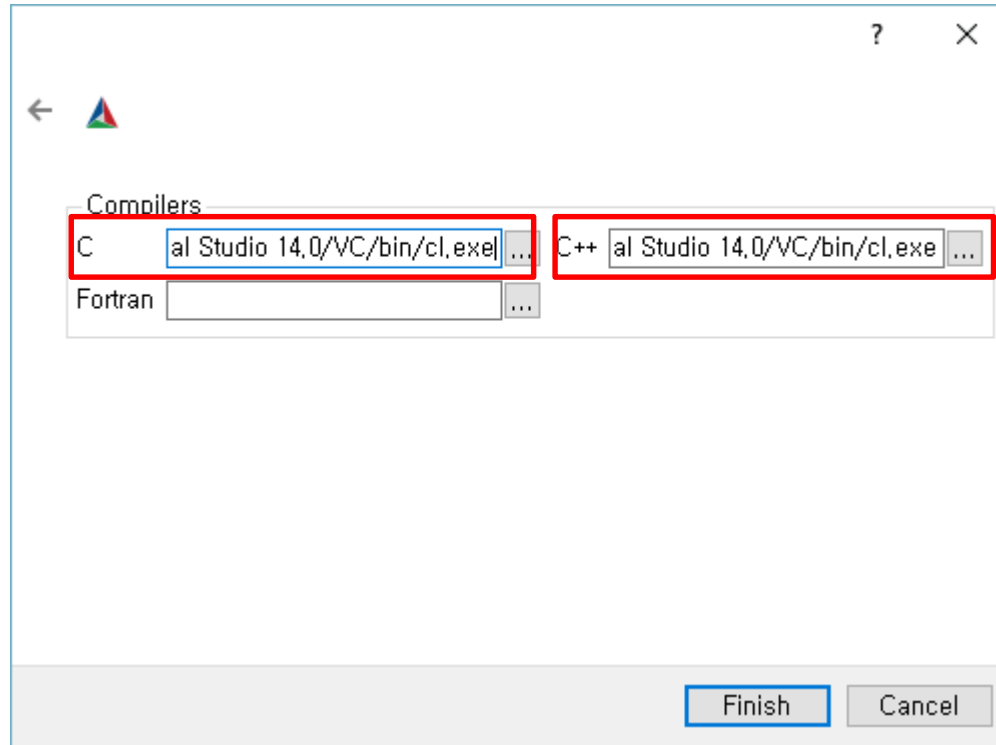
- Set Visual Studio (Compiler) version and machine version and check "Specify native compilers". (**Usually compiler name is "cl.exe".**)
- Then press "next" button.



15 2017 로 선택할 것
또는
16 2019 로 선택할 것

1.2 Open CV Build and Installation

- Specify C/C++ compilers from your system (set the same compiler).



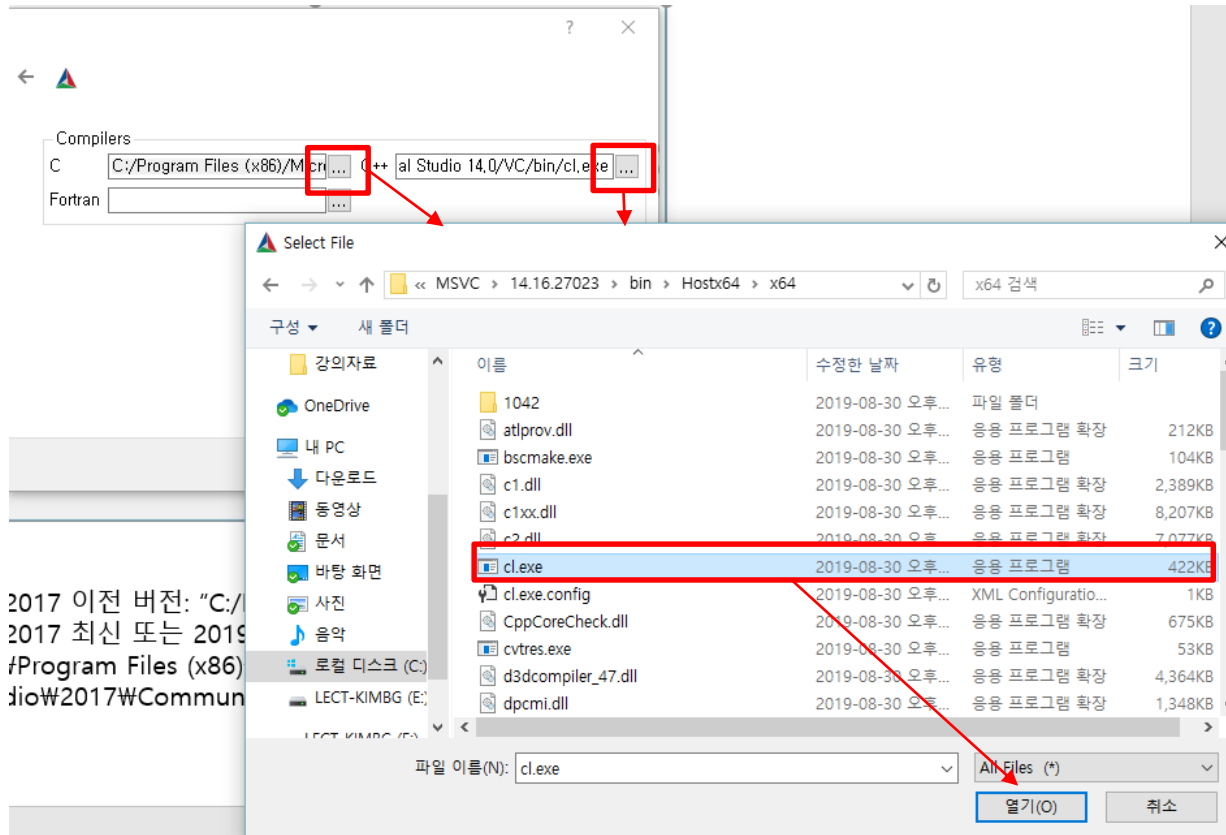
VS 2017 이전 버전: "C:/Program Files (x86)/Microsoft Visual Studio 14.0/VC/bin/cl.exe"

VS 2017 최신 또는 2019 버전:

"C:\Program Files (x86)\Microsoft Visual
Studio\2017\Community\VC\Tools\MSVC\14.16.27023\bin\Hostx64\x64/cl.exe"

"C:\Program Files (x86)\Microsoft Visual
Studio\2019\Community\VC\Tools\MSVC\14.27.29110\bin\Hostx64\x64\cl.exe"

1.2 Open CV Build and Installation



VS 2017 이전 버전: "C:/Program Files (x86)/Microsoft Visual Studio 14.0/VC/bin/cl.exe"

VS 2017 최신 또는 2019/2022 버전:

"C:\Program Files (x86)\Microsoft Visual

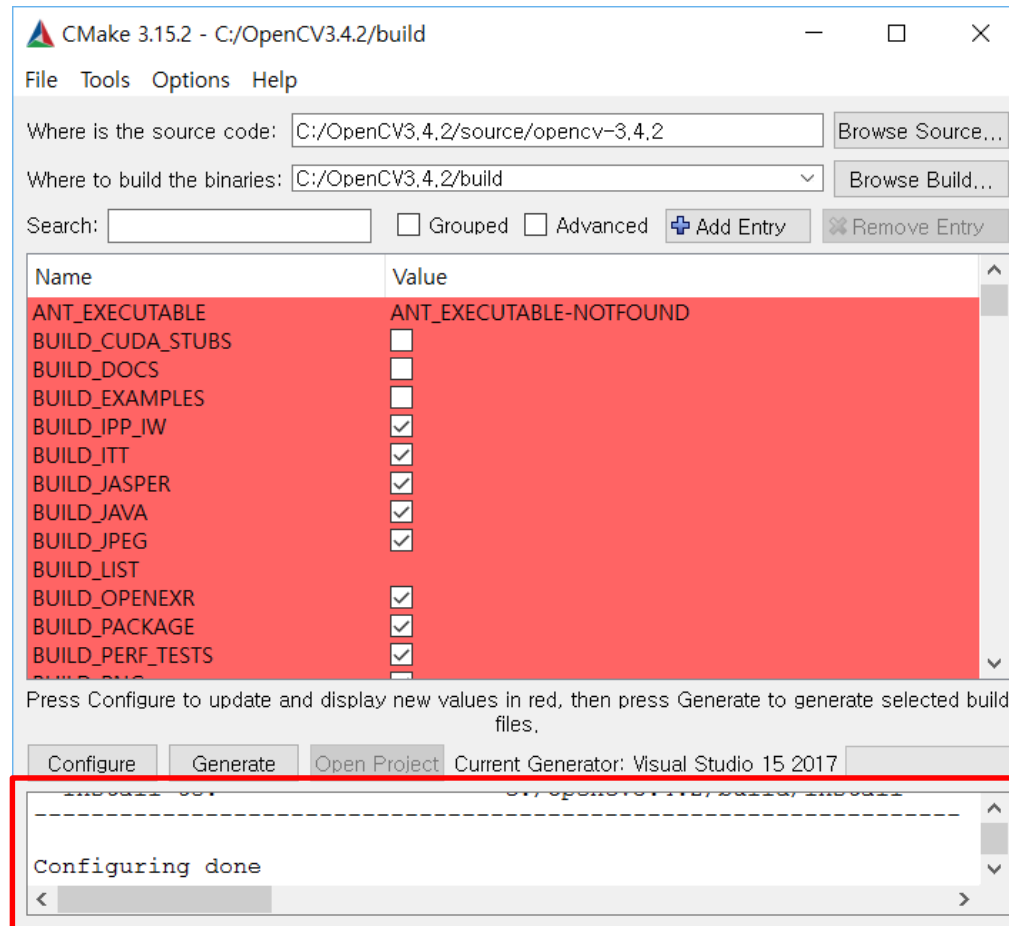
Studio\2017\Community\VC\Tools\MSVC\14.16.27023\bin\Hostx64\x64\cl.exe"

"C:\Program Files\Microsoft Visual

Studio\2022\Community\VC\Tools\MSVC\14.35.32215\bin\Hostx64\x64" (VS2022 버전)

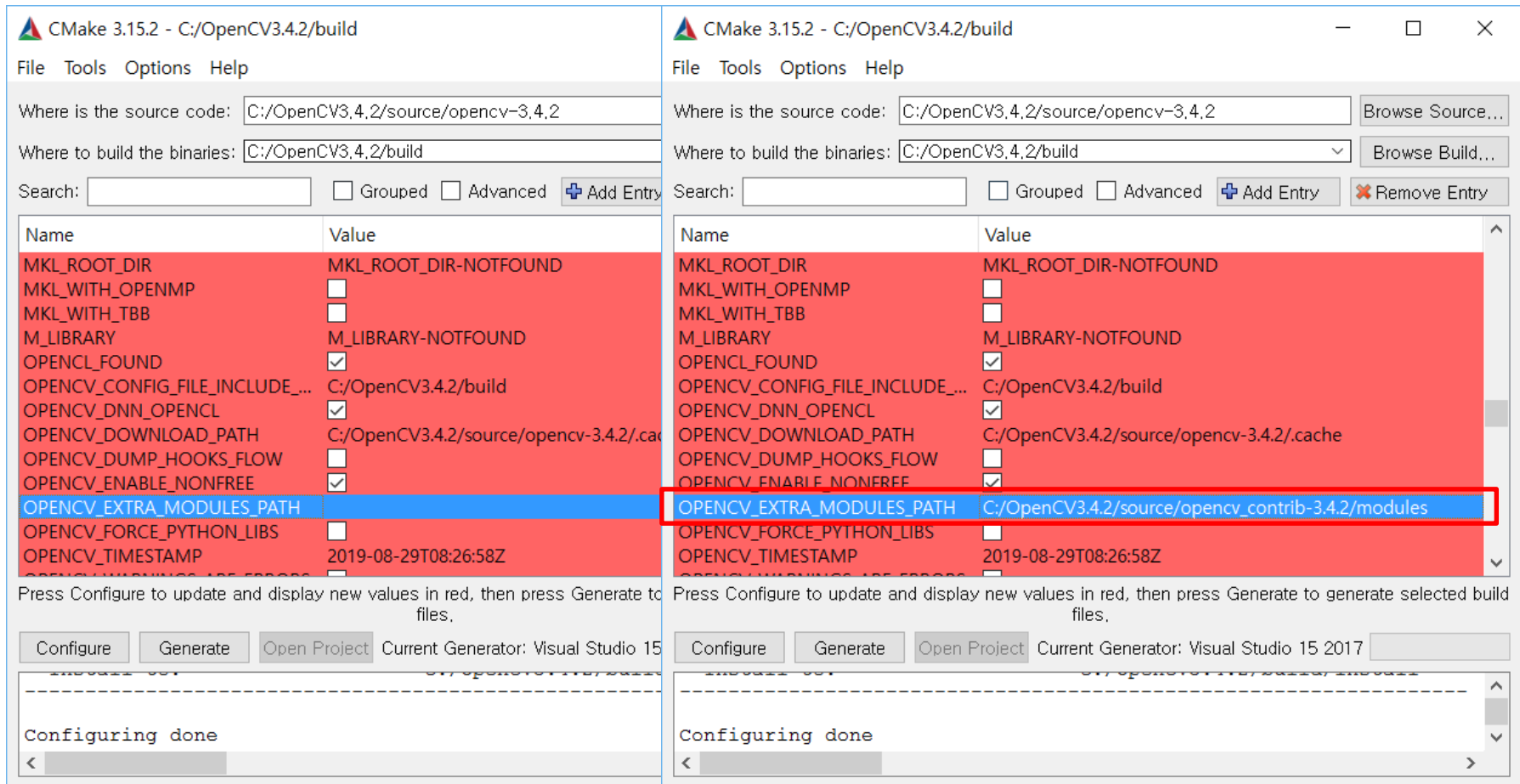
1.2 Open CV Build and Installation

- Then the configuring with compiler will be started automatically.
- Let's wait until CMake finishes its initial configuration. Whenever, the process has finished, we can see the configuration log denoting that it's done as shown in figure below:



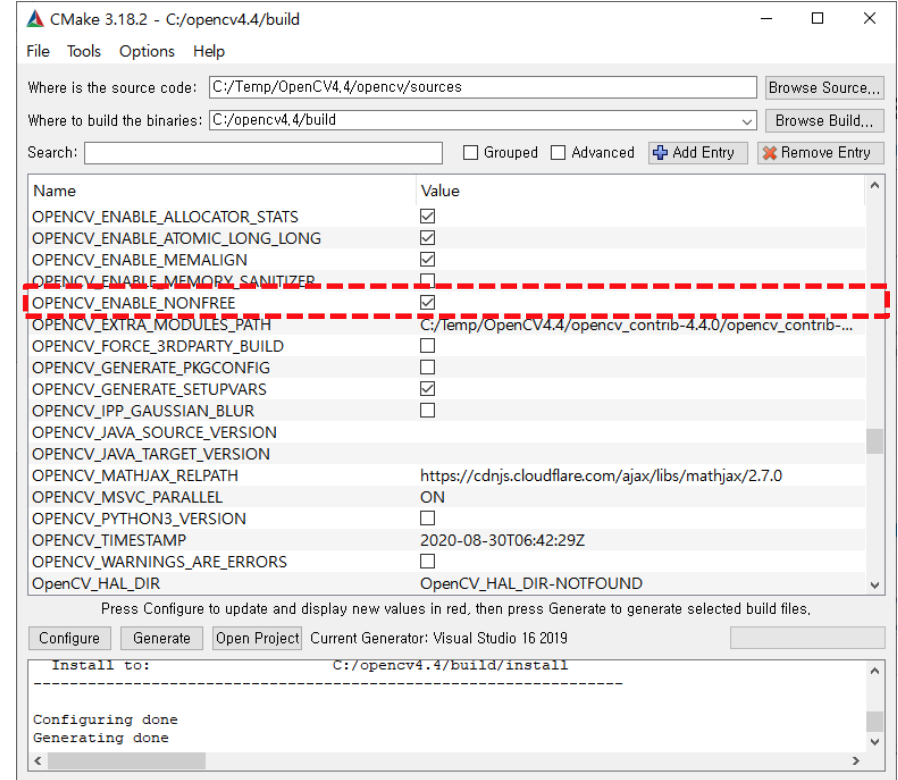
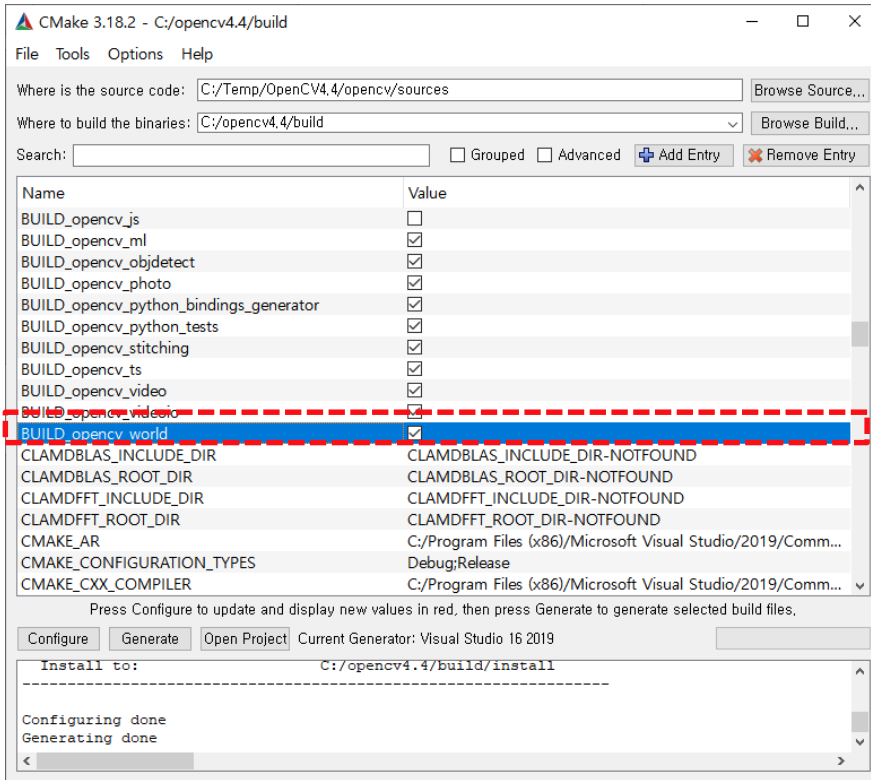
1.2 Open CV Build and Installation

- To add extra module in the original version, we need to specify the extra modules path which is {opencv-4.x}/source/**opencv_contrib/modules** as depicted in figure below:



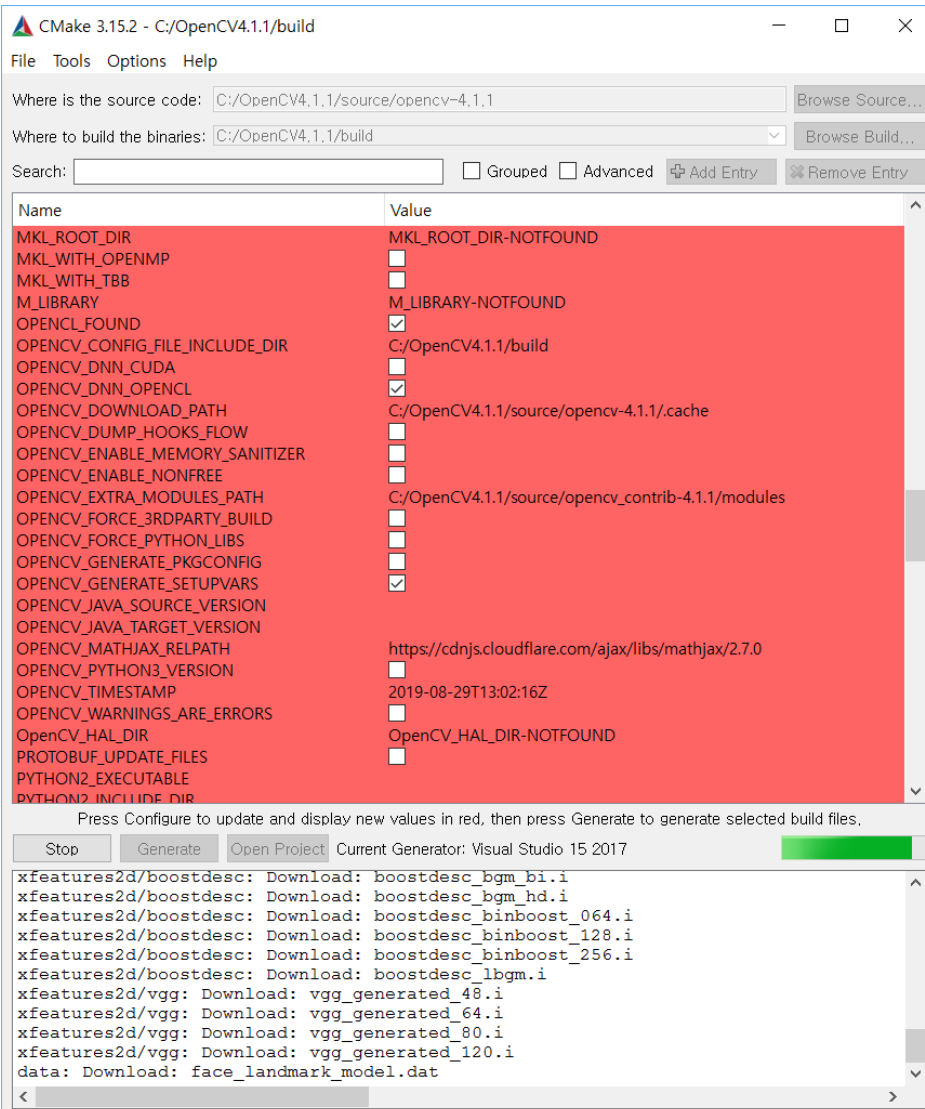
1.2 Open CV Build and Installation

- Check on two flag (option): BUILD_opencv_world, OPENCV_ENABLE_NONFREE



1.2 Open CV Build and Installation

- *After we have finished re-configuring, we need to hit the "Configure" button once again.*



CMake 3.15.2 - C:/OpenCV4.1.1/build

Where is the source code: C:/OpenCV4.1.1/source/opencv-4.1.1

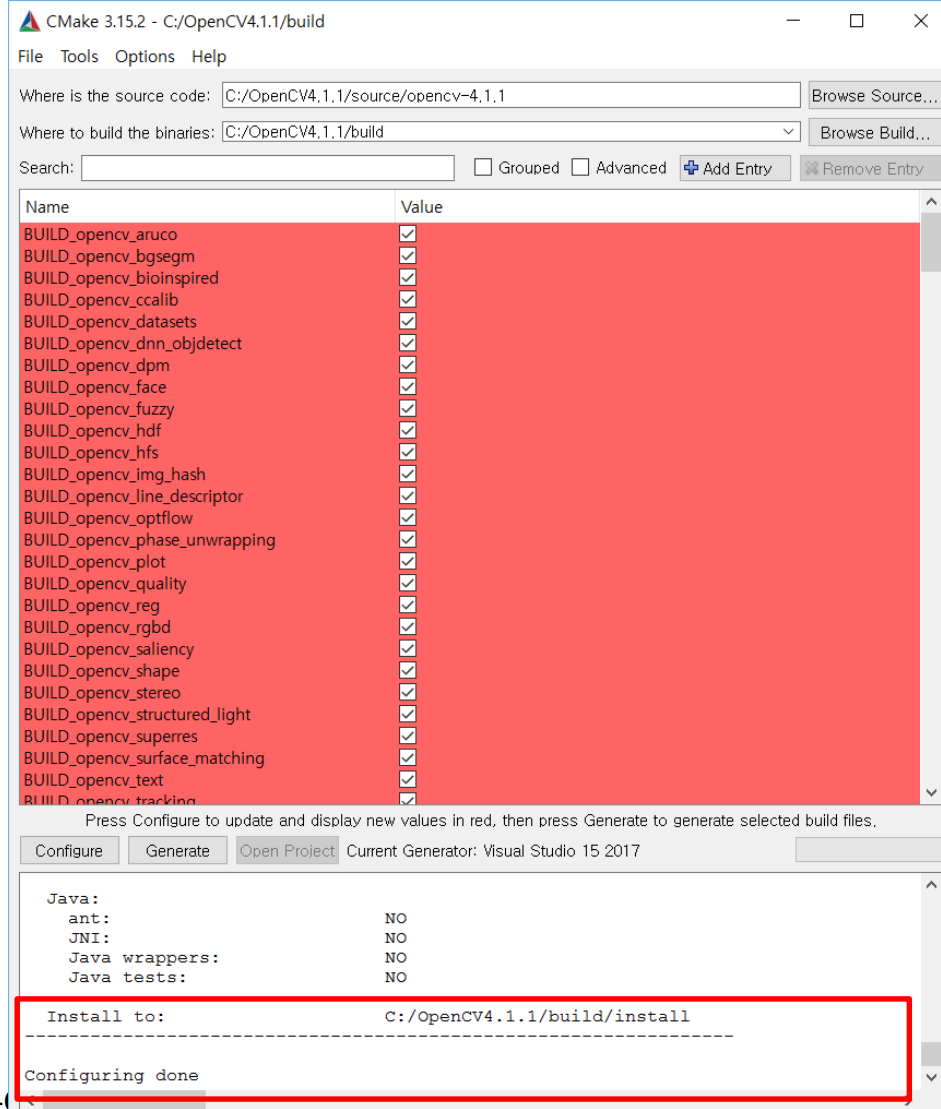
Where to build the binaries: C:/OpenCV4.1.1/build

Name	Value
MKL_ROOT_DIR	MKL_ROOT_DIR-NOTFOUND
MKL_WITH_OPENMP	<input type="checkbox"/>
MKL_WITH_TBB	<input type="checkbox"/>
M_LIBRARY	M_LIBRARY-NOTFOUND
OPENCV_FOUND	<input checked="" type="checkbox"/>
OPENCV_CONFIG_FILE_INCLUDE_DIR	C:/OpenCV4.1.1/build
OPENCV_DNN_CUDA	<input type="checkbox"/>
OPENCV_DNN_OPENCV	<input checked="" type="checkbox"/>
OPENCV_DOWNLOAD_PATH	C:/OpenCV4.1.1/source/opencv-4.1.1/cache
OPENCV_DUMP_HOOKS_FLOW	<input type="checkbox"/>
OPENCV_ENABLE_MEMORY_SANITIZER	<input type="checkbox"/>
OPENCV_ENABLE_NONFREE	<input type="checkbox"/>
OPENCV_EXTRA_MODULES_PATH	C:/OpenCV4.1.1/source/opencv_contrib-4.1.1/modules
OPENCV_FORCE_3RDPARTY_BUILD	<input type="checkbox"/>
OPENCV_FORCE_PYTHON_LIBS	<input type="checkbox"/>
OPENCV_GENERATE_PKGCONFIG	<input checked="" type="checkbox"/>
OPENCV_GENERATE_SETUPVARS	<input checked="" type="checkbox"/>
OPENCV_JAVA_SOURCE_VERSION	
OPENCV_JAVA_TARGET_VERSION	
OPENCV_MATHJAX_RELPATH	https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.0
OPENCV_PYTHON3_VERSION	<input type="checkbox"/>
OPENCV_TIMESTAMP	2019-08-29T13:02:16Z
OPENCV_WARNINGS_ARE_ERRORS	<input type="checkbox"/>
OpenCV_HAL_DIR	OpenCV_HAL_DIR-NOTFOUND
PROTOBUF_UPDATE_FILES	<input type="checkbox"/>
PYTHON2_EXECUTABLE	
PYTHON2_INCLUDE_DIR	

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

Stop Generate Open Project Current Generator: Visual Studio 15 2017

```
xfeatures2d/boostdesc: Download: boostdesc_bgm_bi.i
xfeatures2d/boostdesc: Download: boostdesc_bgm_hd.i
xfeatures2d/boostdesc: Download: boostdesc_binboost_064.i
xfeatures2d/boostdesc: Download: boostdesc_binboost_128.i
xfeatures2d/boostdesc: Download: boostdesc_binboost_256.i
xfeatures2d/boostdesc: Download: boostdesc_lbgm.i
xfeatures2d/vgg: Download: vgg_generated_48.i
xfeatures2d/vgg: Download: vgg_generated_64.i
xfeatures2d/vgg: Download: vgg_generated_80.i
xfeatures2d/vgg: Download: vgg_generated_120.i
data: Download: face_landmark_model.dat
```



CMake 3.15.2 - C:/OpenCV4.1.1/build

Where is the source code: C:/OpenCV4.1.1/source/opencv-4.1.1

Where to build the binaries: C:/OpenCV4.1.1/build

Name	Value
BUILD_opencv_aruco	<input checked="" type="checkbox"/>
BUILD_opencv_bgsegm	<input checked="" type="checkbox"/>
BUILD_opencv_bioinspired	<input checked="" type="checkbox"/>
BUILD_opencv_ccalib	<input checked="" type="checkbox"/>
BUILD_opencv_datasets	<input checked="" type="checkbox"/>
BUILD_opencv_dnn_objdetect	<input checked="" type="checkbox"/>
BUILD_opencv_dpm	<input checked="" type="checkbox"/>
BUILD_opencv_face	<input checked="" type="checkbox"/>
BUILD_opencv_fuzzy	<input checked="" type="checkbox"/>
BUILD_opencv_hdf	<input checked="" type="checkbox"/>
BUILD_opencv_hfs	<input checked="" type="checkbox"/>
BUILD_opencv_img_hash	<input checked="" type="checkbox"/>
BUILD_opencv_line_descriptor	<input checked="" type="checkbox"/>
BUILD_opencv_optflow	<input checked="" type="checkbox"/>
BUILD_opencv_phase_unwrapping	<input checked="" type="checkbox"/>
BUILD_opencv_plot	<input checked="" type="checkbox"/>
BUILD_opencv_quality	<input checked="" type="checkbox"/>
BUILD_opencv_reg	<input checked="" type="checkbox"/>
BUILD_opencv_rgbd	<input checked="" type="checkbox"/>
BUILD_opencv_saliency	<input checked="" type="checkbox"/>
BUILD_opencv_shape	<input checked="" type="checkbox"/>
BUILD_opencv_stereo	<input checked="" type="checkbox"/>
BUILD_opencv_structured_light	<input checked="" type="checkbox"/>
BUILD_opencv_superres	<input checked="" type="checkbox"/>
BUILD_opencv_surface_matching	<input checked="" type="checkbox"/>
BUILD_opencv_text	<input checked="" type="checkbox"/>
BUILD_opencv_tracking	<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

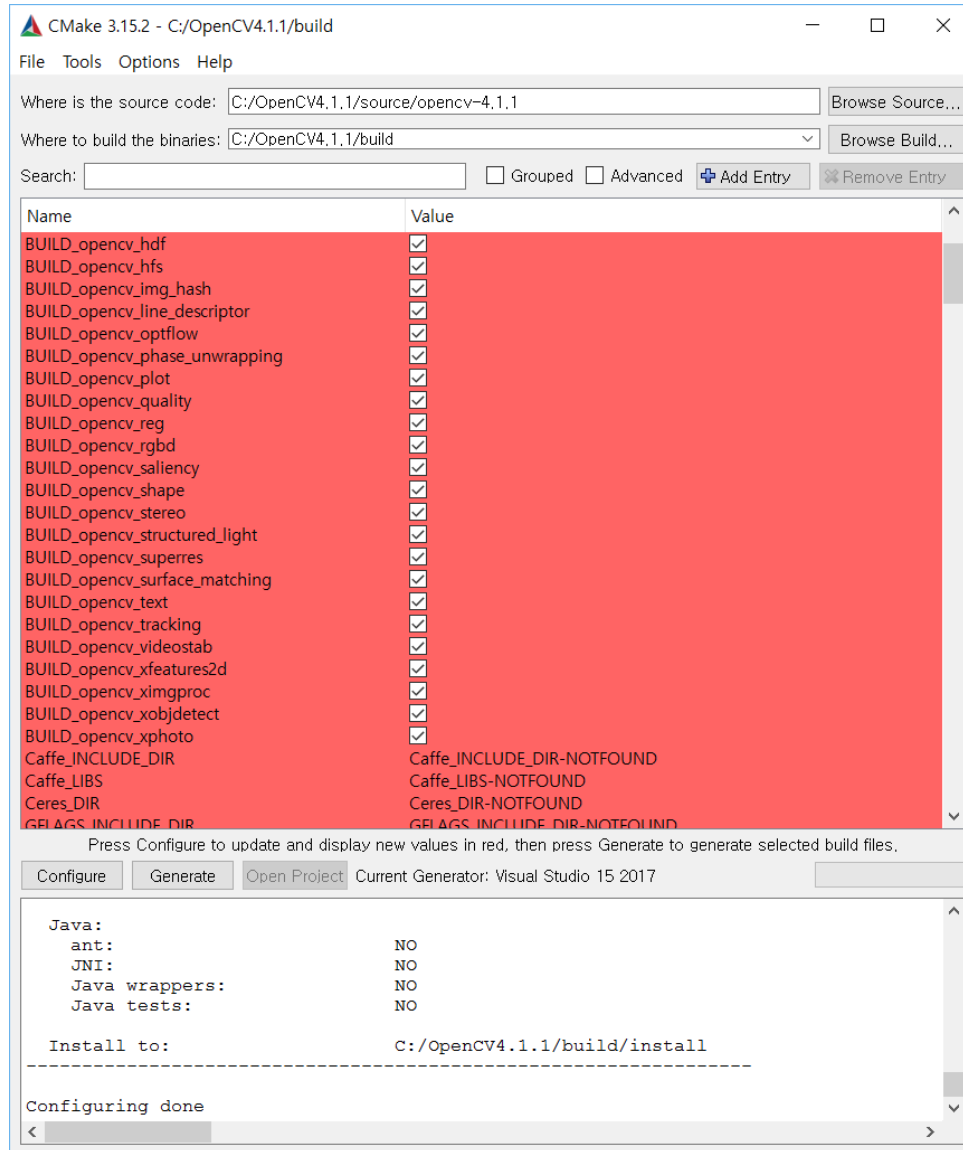
```
Java:
ant: NO
JNI: NO
Java wrappers: NO
Java tests: NO

Install to: C:/OpenCV4.1.1/build/install

-----
Configuring done
```

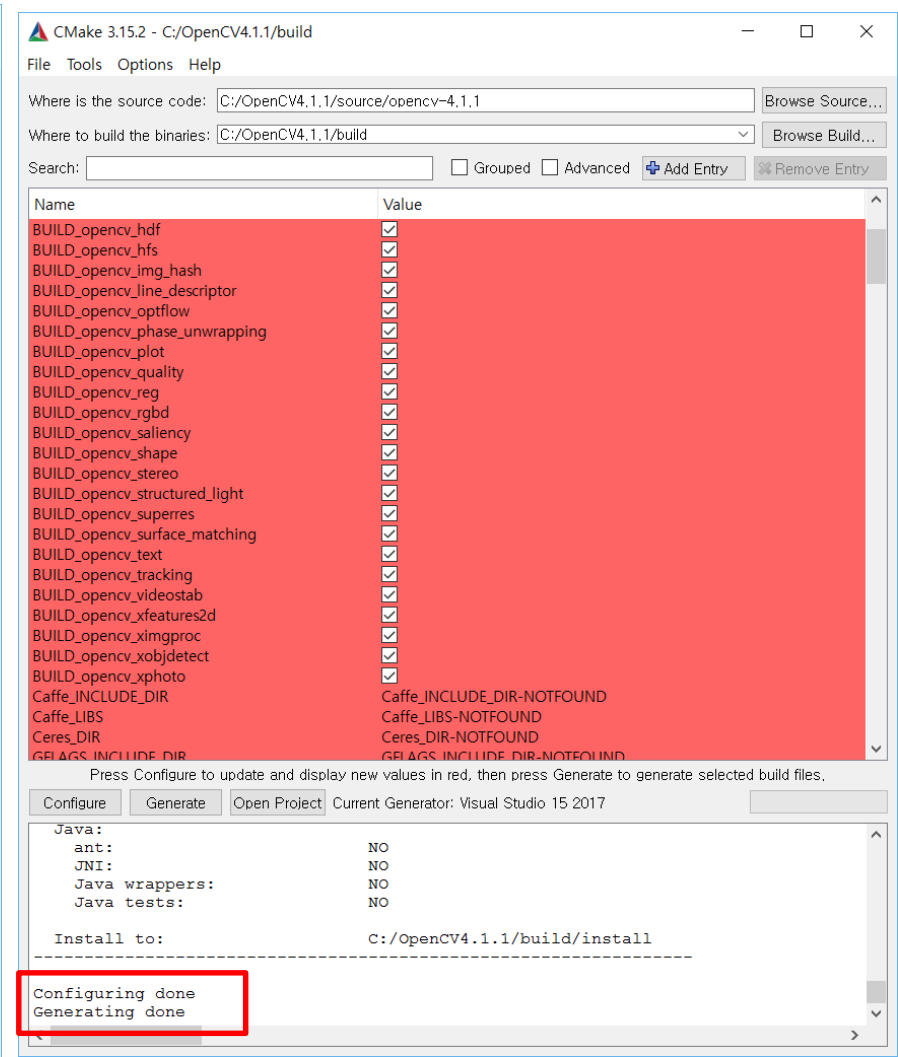
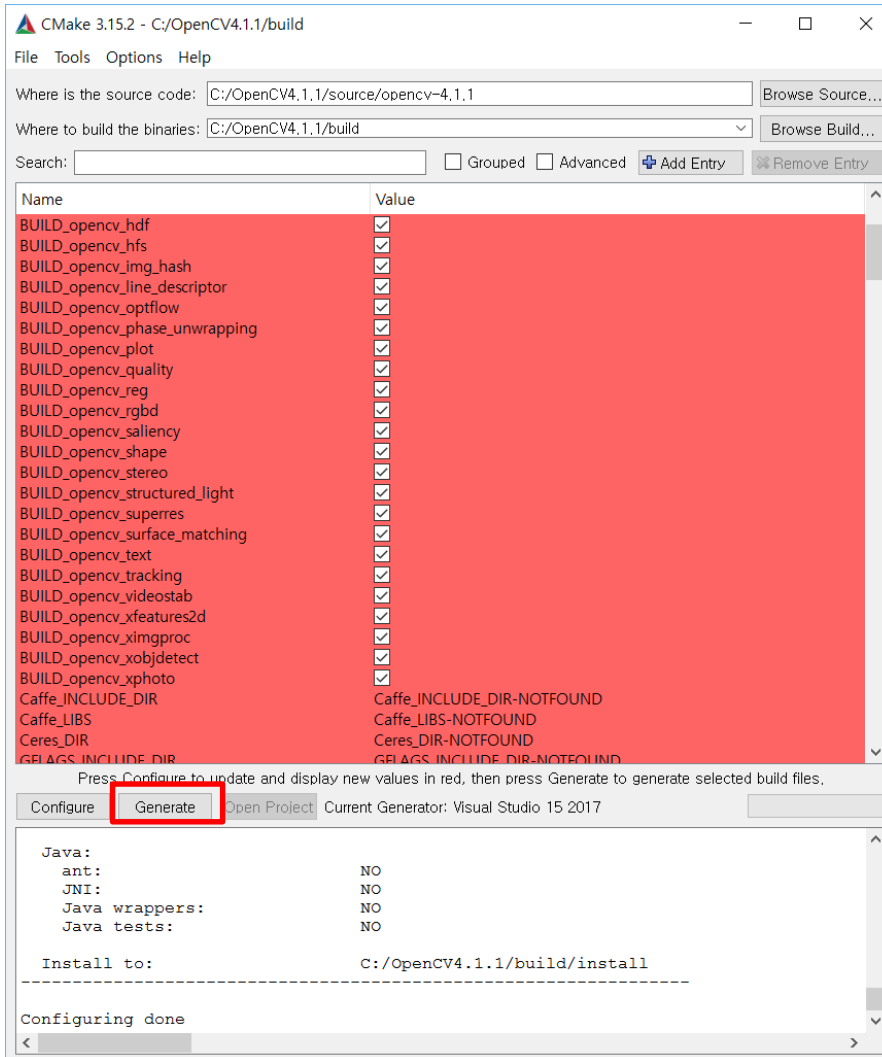
1.2 Open CV Build and Installation

- Now, let's double check whether the desired extra modules are on the list.



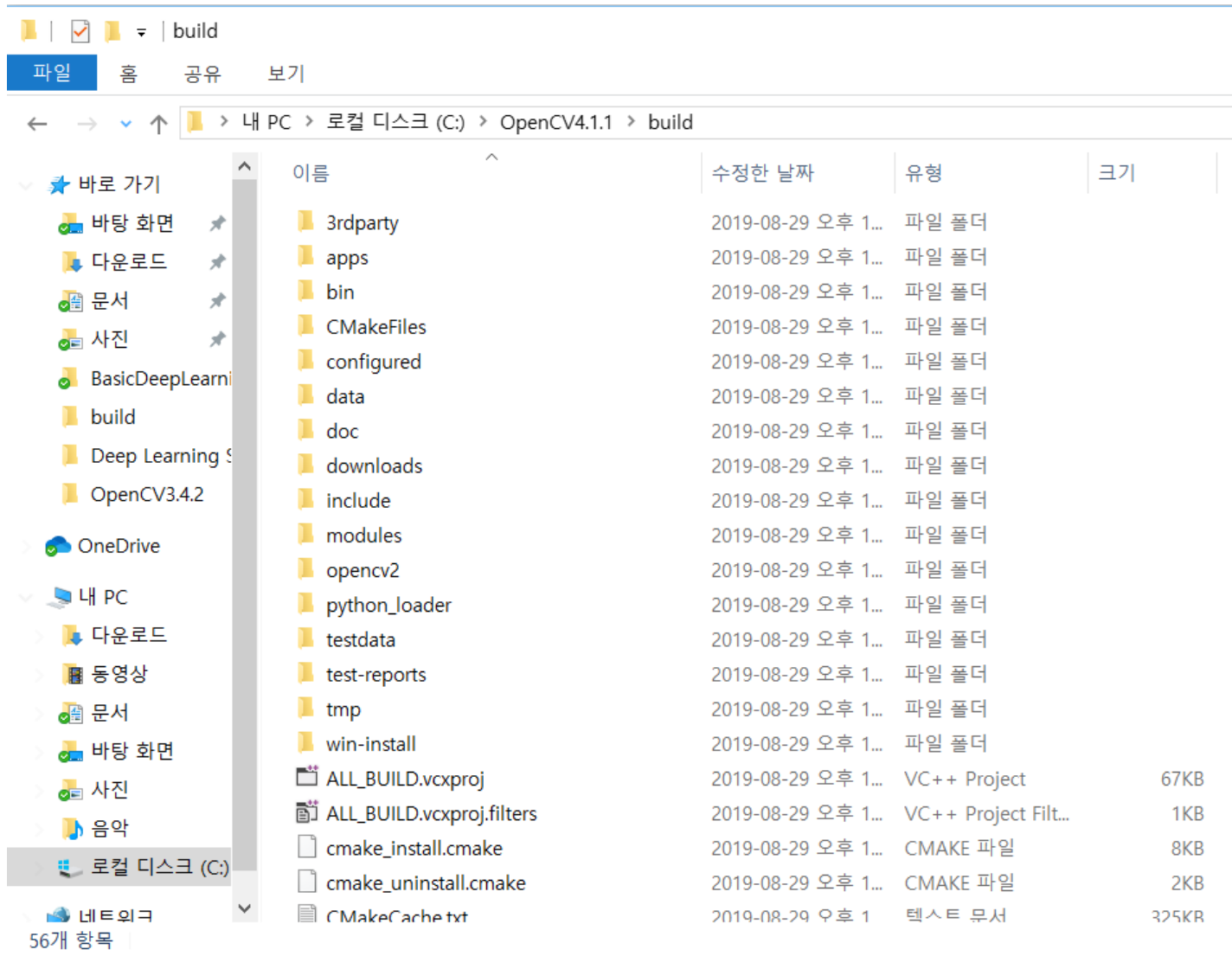
1.2 Open CV Build and Installation

- After that, we can start generating the Visual Studio Project by pushing the "Generate" button. And your build directory will be fulfilled with OpenCV Visual Studio Project.



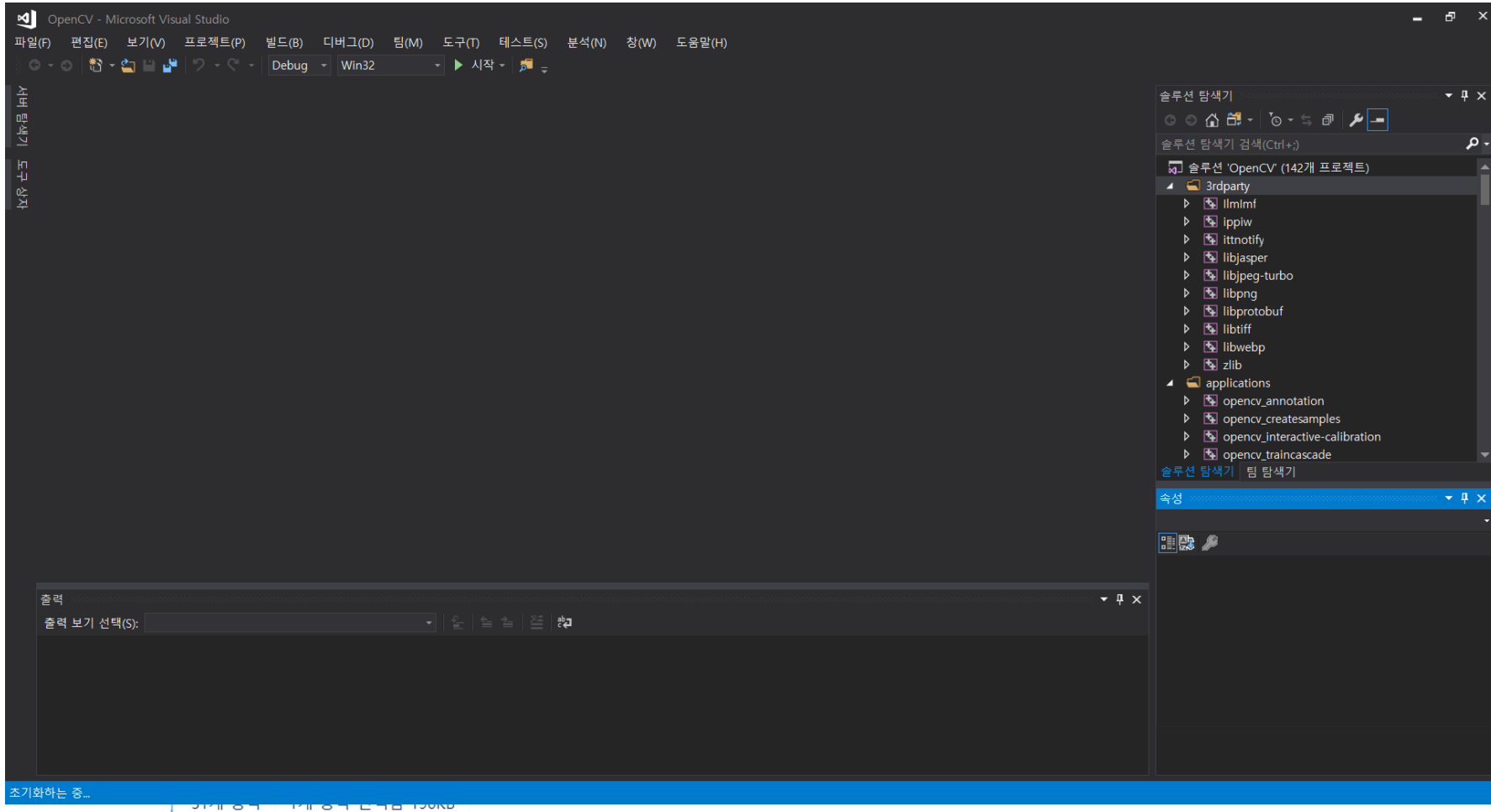
1.2 Open CV Build and Installation

- In "build" fold, you can find the compiled source.



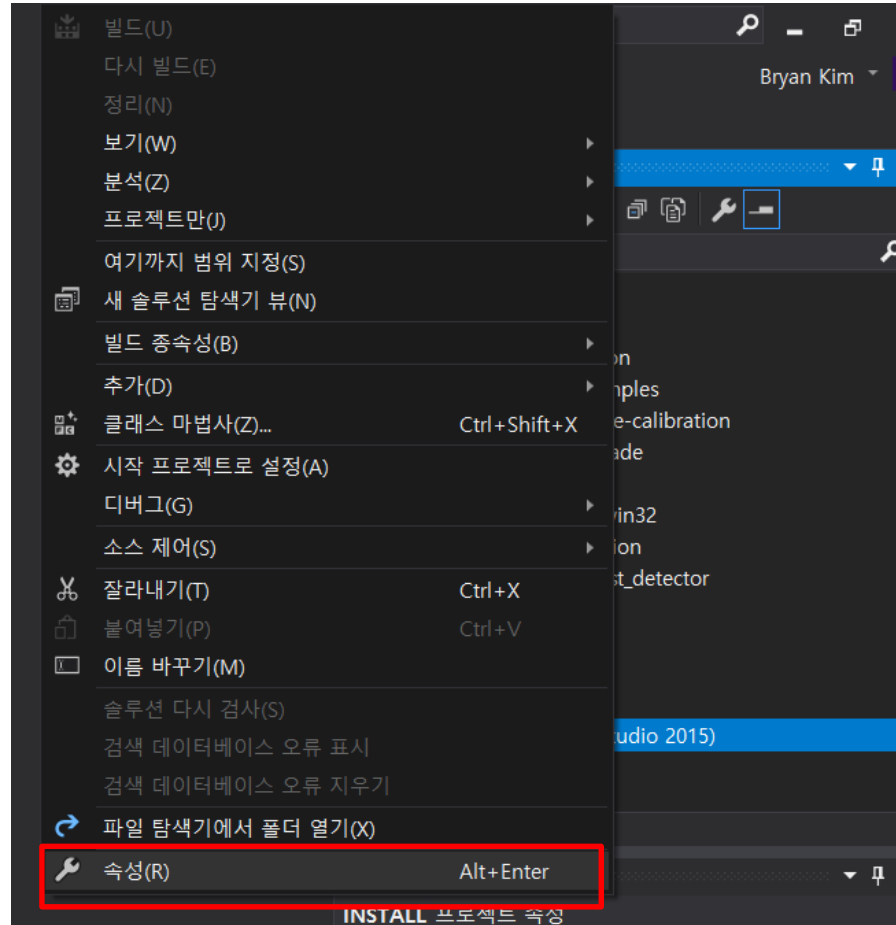
1.2 Open CV Build and Installation

- Building OpenCV Project
 - launch the Visual Studio application open the **OpenCV.sln** file located in the *build* directory.



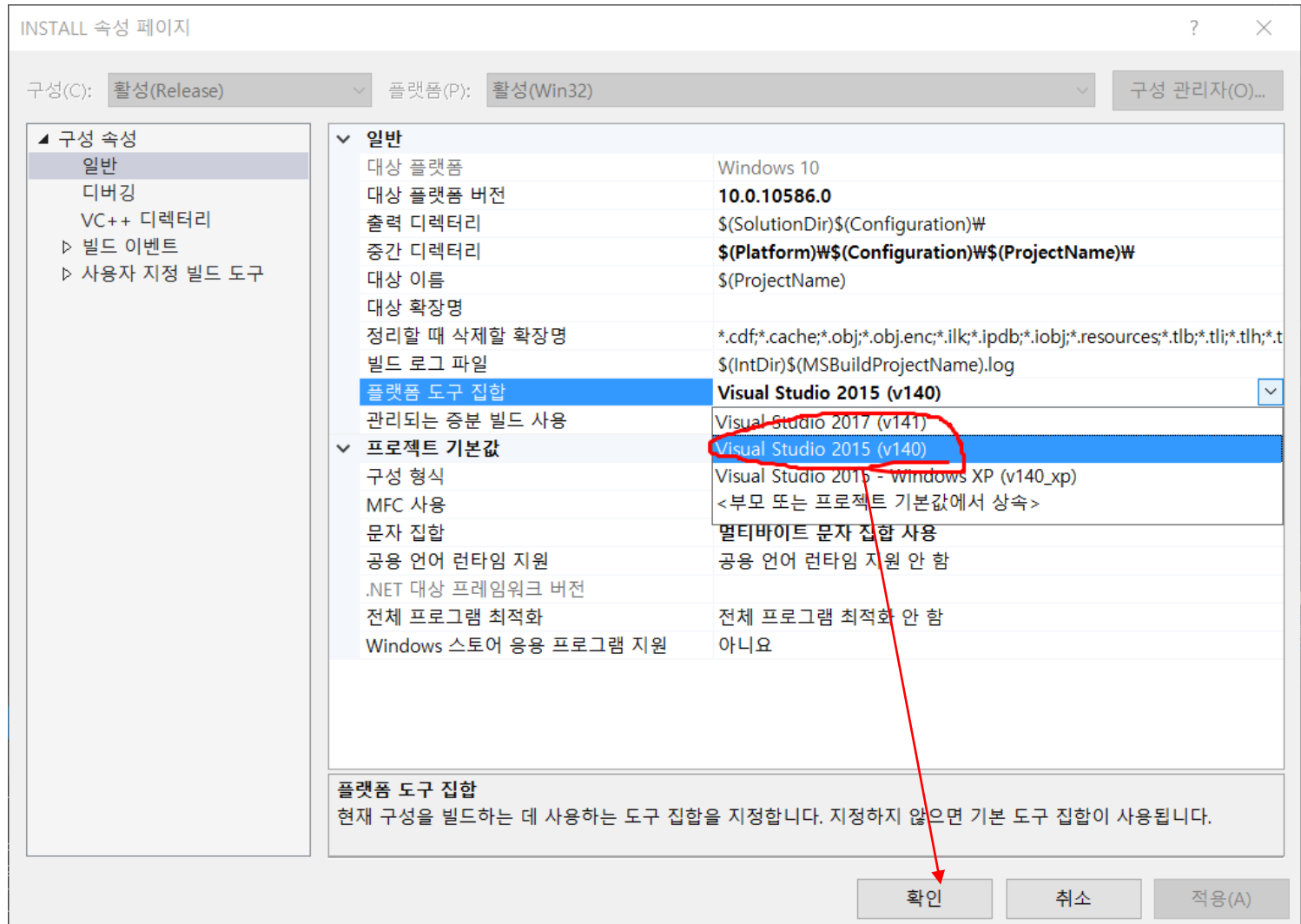
1.2 Open CV Build and Installation

- After loading all of the project files, go to Solution Explorer, find the *INSTALL* project inside the *CMakeTargets* directory, and open the property of *INSTALL* project. It may take several minutes for the whole building process to finish, just be patient.



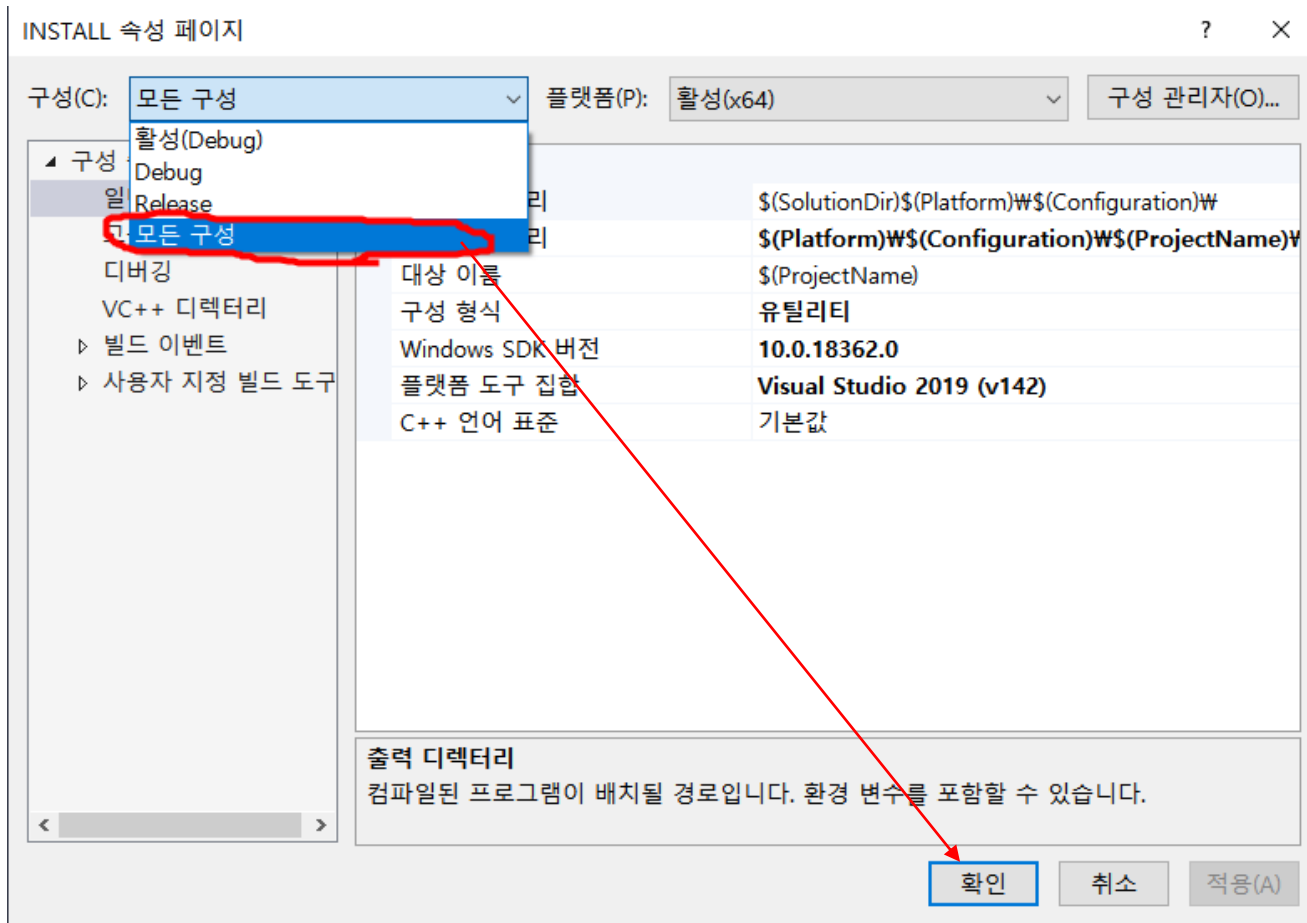
1.2 Open CV Build and Installation

- Change the compiler version as "VS 2015(v140)" if you need it.



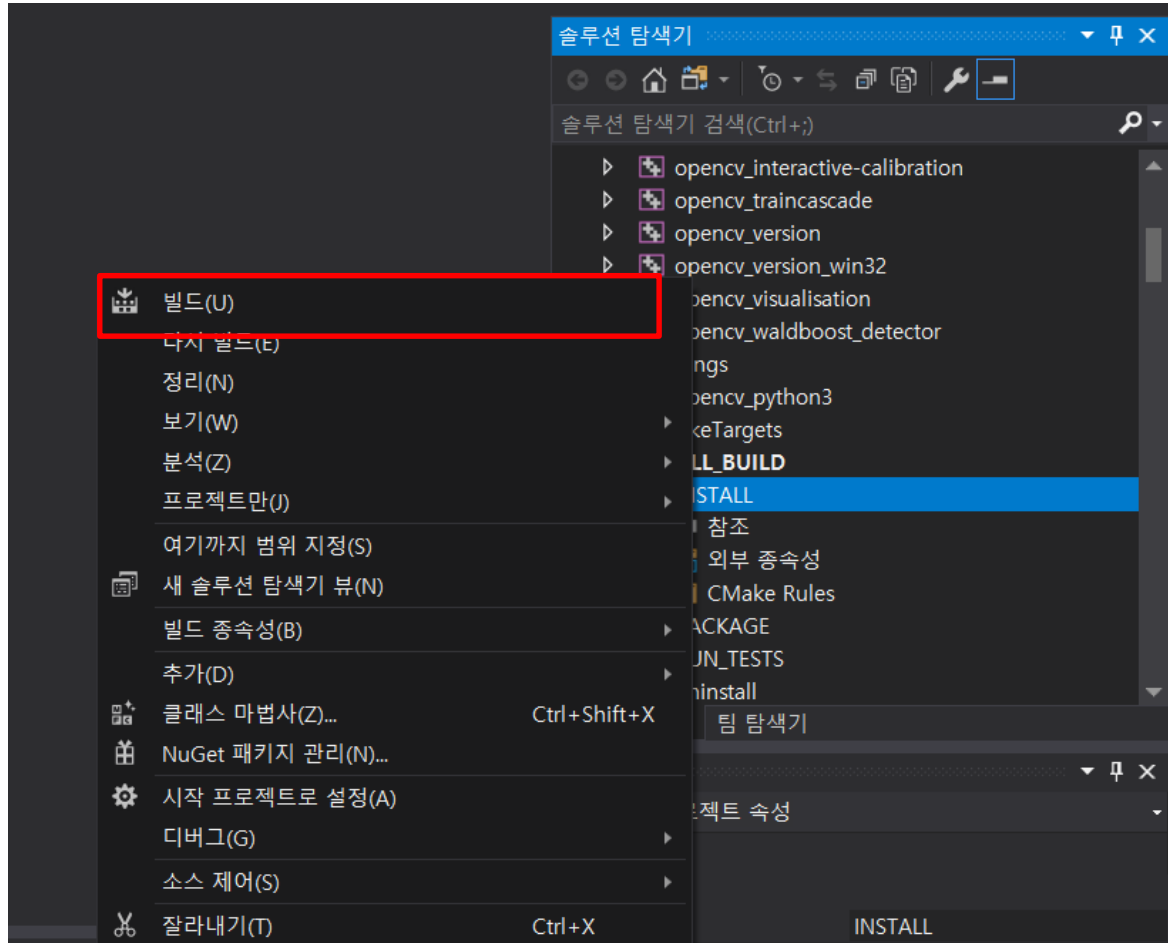
1.2 Open CV Build and Installation

- Change "구성" as "모든 구성" if you need debug and release modes both.



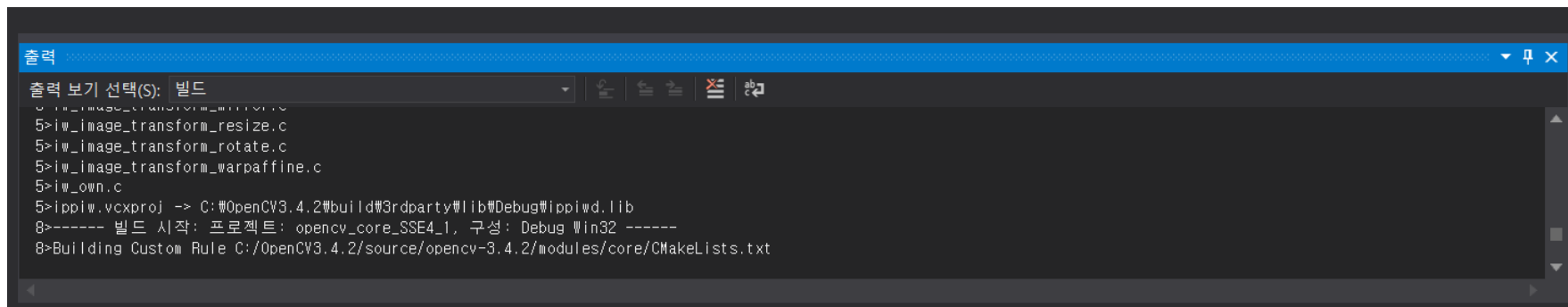
1.2 Open CV Build and Installation

- After loading all of the project files, go to Solution Explorer, find the *INSTALL* project inside the *CMakeTargets* directory, and build that project. It may take several minutes for the whole building process to finish, just be patient.

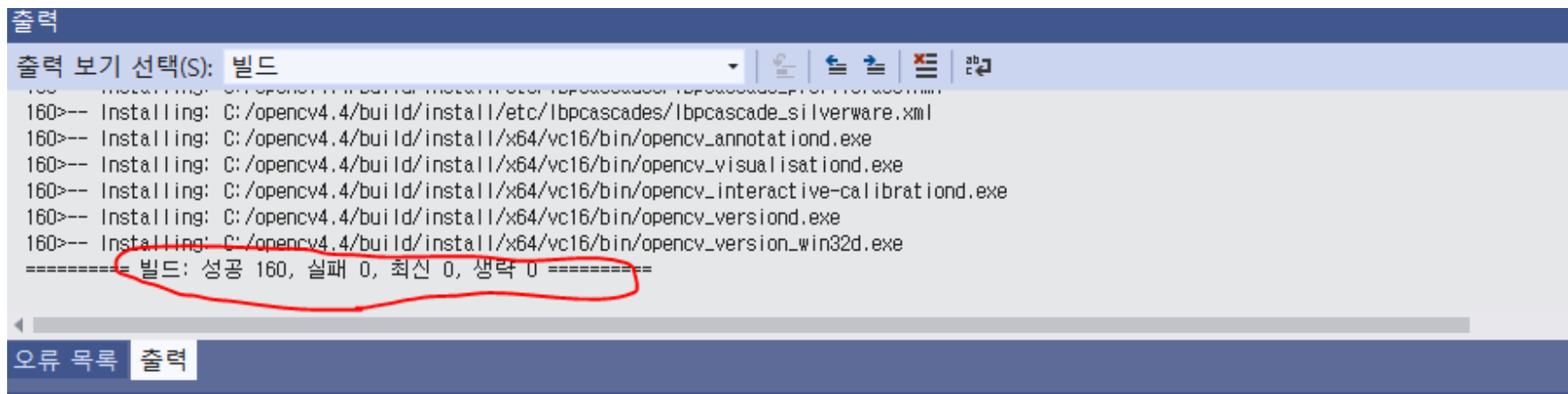


1.2 Open CV Build and Installation

- Building the Visual Studio Project....!!!!



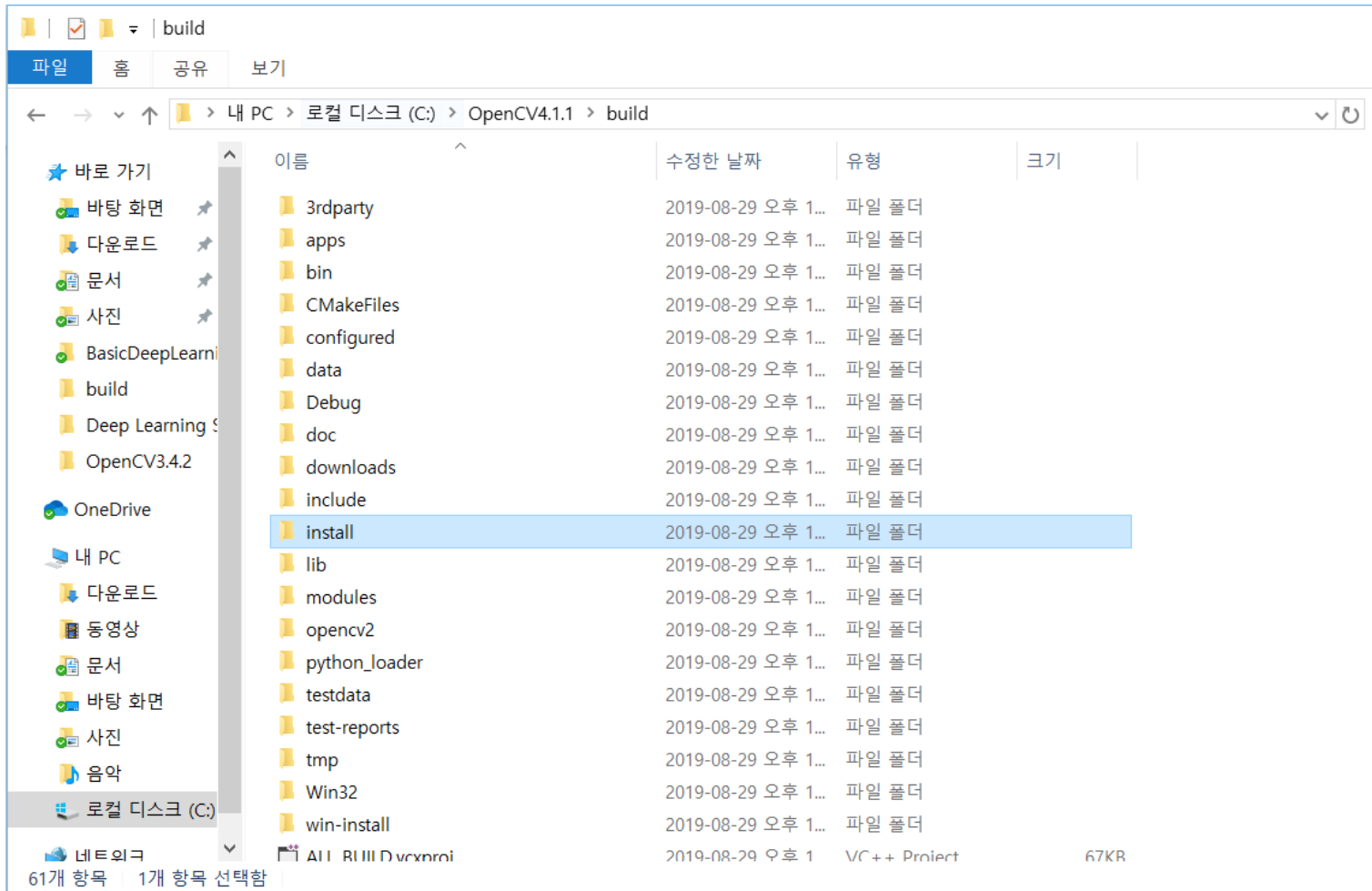
```
출력
출력 보기 선택(S): 빌드
C:\opencv\build\install\etc\lbpcascades\lbpcascade_silverware.xml
5>iw_image_transform_resize.c
5>iw_image_transform_rotate.c
5>iw_image_transform_warpaffine.c
5>iw_owm.c
5>ippiw.vcxproj -> C:\OpenCV3.4.2\build\3rdparty\lib\Debug\ippiw.lib
8>----- 빌드 시작: 프로젝트: opencv_core_SSE4_1, 구성: Debug Win32 -----
8>Building Custom Rule C:/OpenCV3.4.2/source/opencv-3.4.2/modules/core/CMakeLists.txt
```



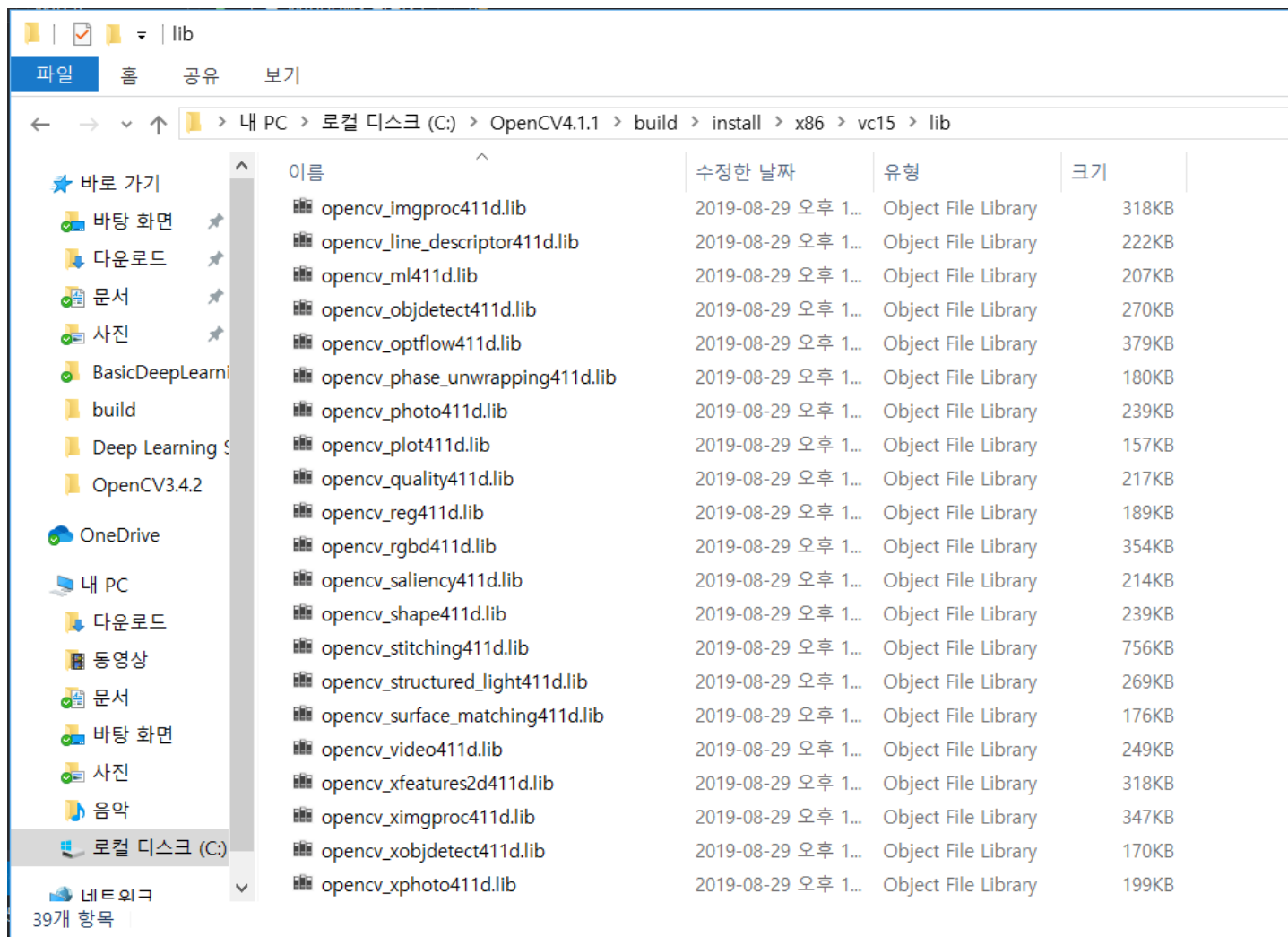
```
출력
출력 보기 선택(S): 빌드
160>-- Installing: C:/opencv4.4/build/install/etc/lbpcascades/lbpcascade_silverware.xml
160>-- Installing: C:/opencv4.4/build/install/x64/vc16/bin/opencv_annotationd.exe
160>-- Installing: C:/opencv4.4/build/install/x64/vc16/bin/opencv_visualisationd.exe
160>-- Installing: C:/opencv4.4/build/install/x64/vc16/bin/opencv_interactive-calibrationd.exe
160>-- Installing: C:/opencv4.4/build/install/x64/vc16/bin/opencv_versiond.exe
160>-- Installing: C:/opencv4.4/build/install/x64/vc16/bin/opencv_version_win32d.exe
===== 빌드: 성공 160, 실패 0, 최신 0, 생략 0 =====
```

1.2 Open CV Build and Installation

- Whenever the whole building process has been done, there will be several resulting new directories, such as: *install*, *bin*, and *lib* directory inside the *build* directory. We will use the *install* directory for building our own computer vision application.



1.2 Open CV Build and Installation



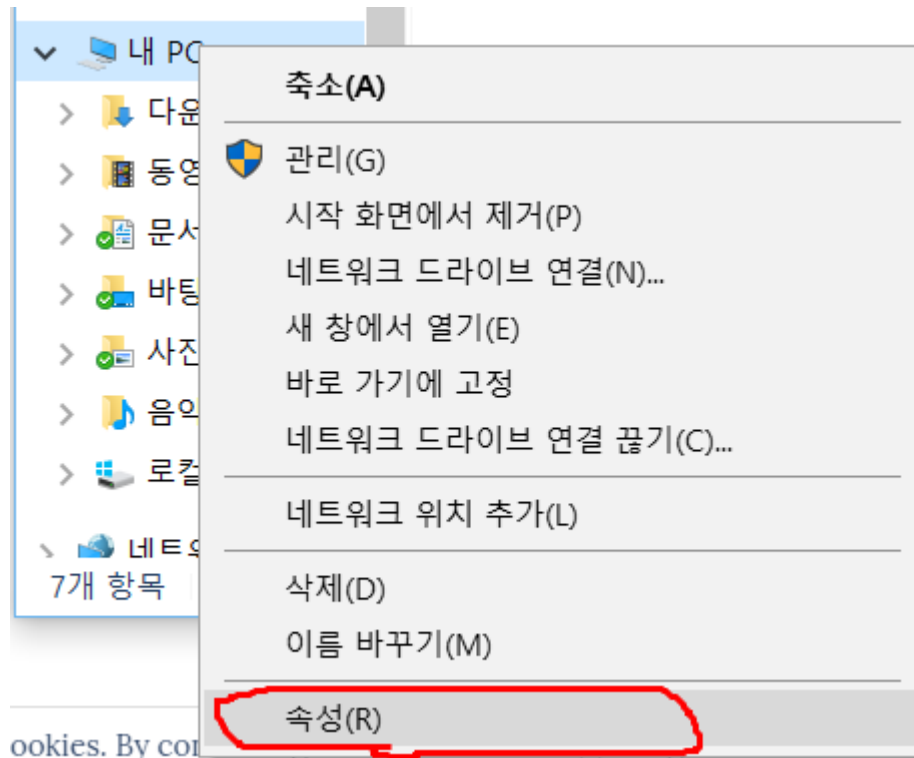
1.2 Open CV Build and Installation

- **Developing Application using OpenCV**

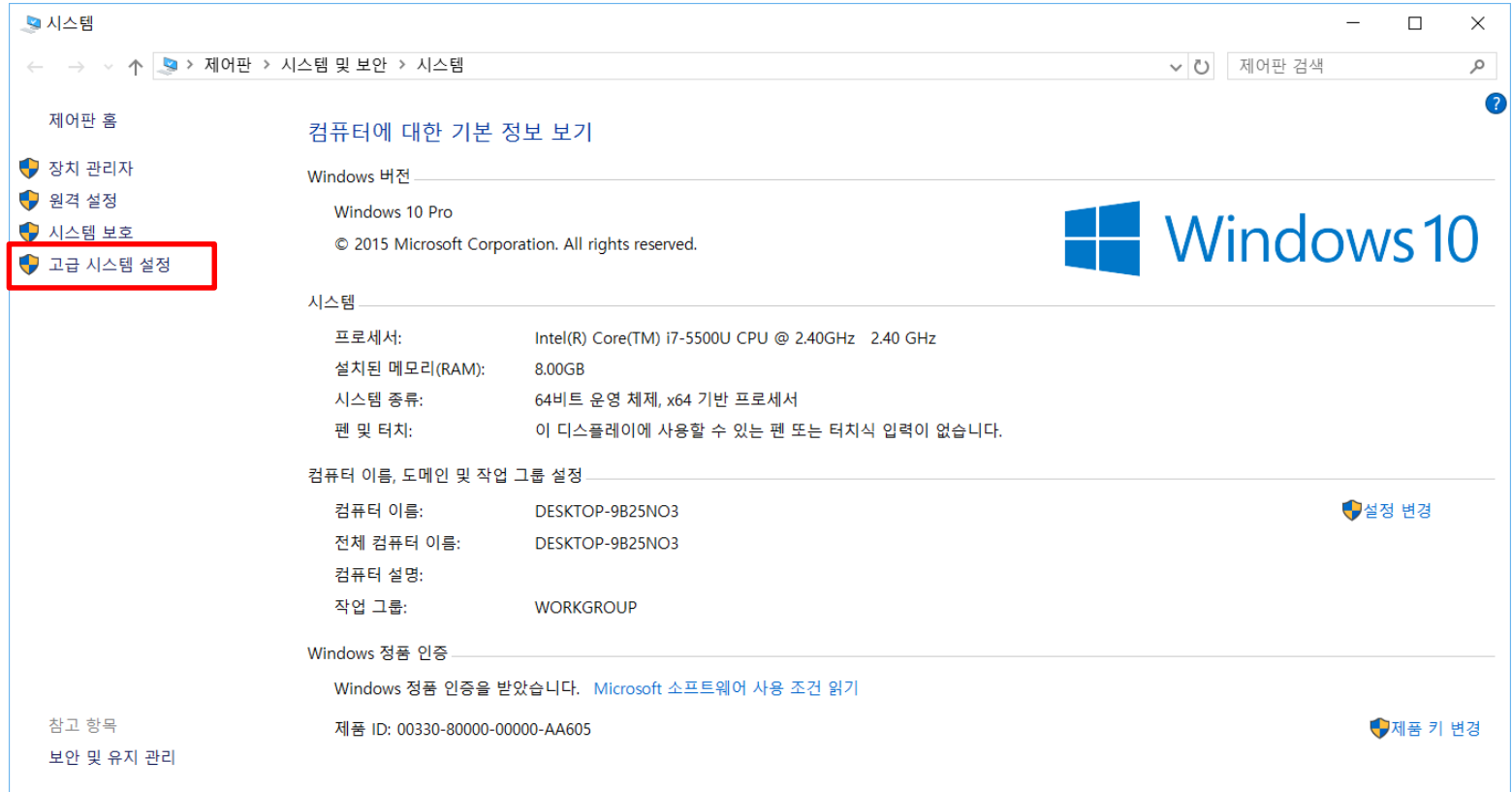
There are two main steps that are required:

1. Registering OpenCV Environment Variables
2. Creating OpenCV Project Property Sheet

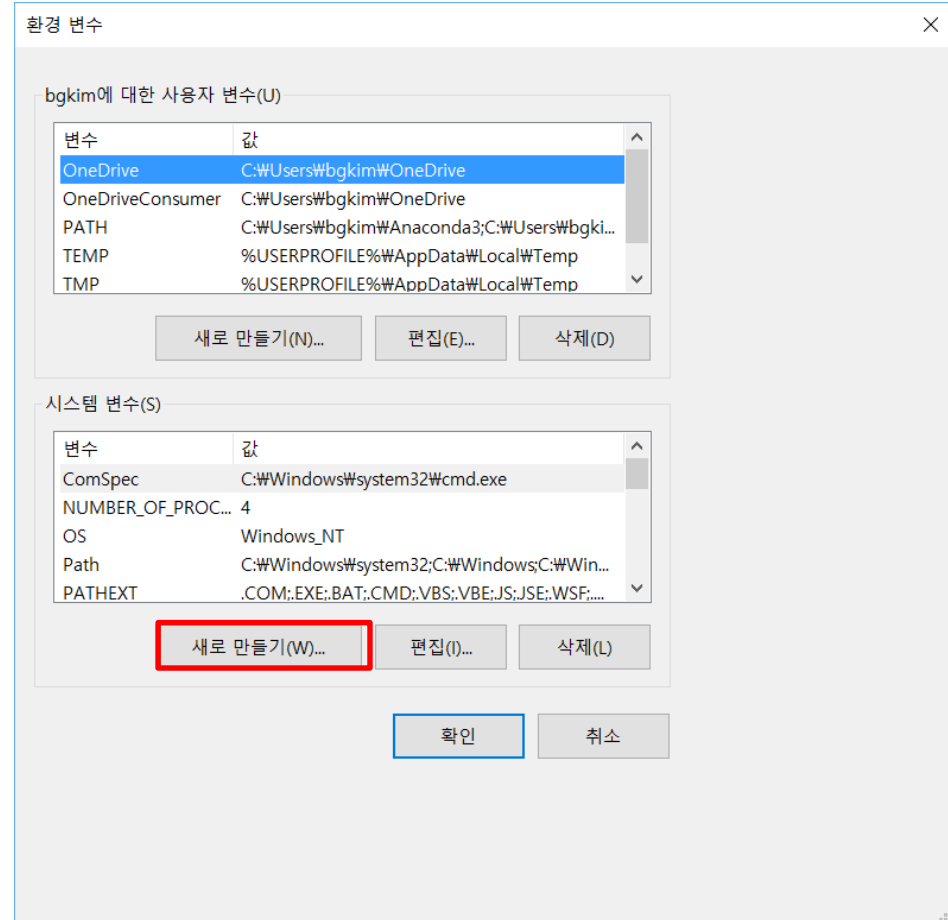
- **1. Registering OpenCV Environment Variables**



1.2 Open CV Build and Installation

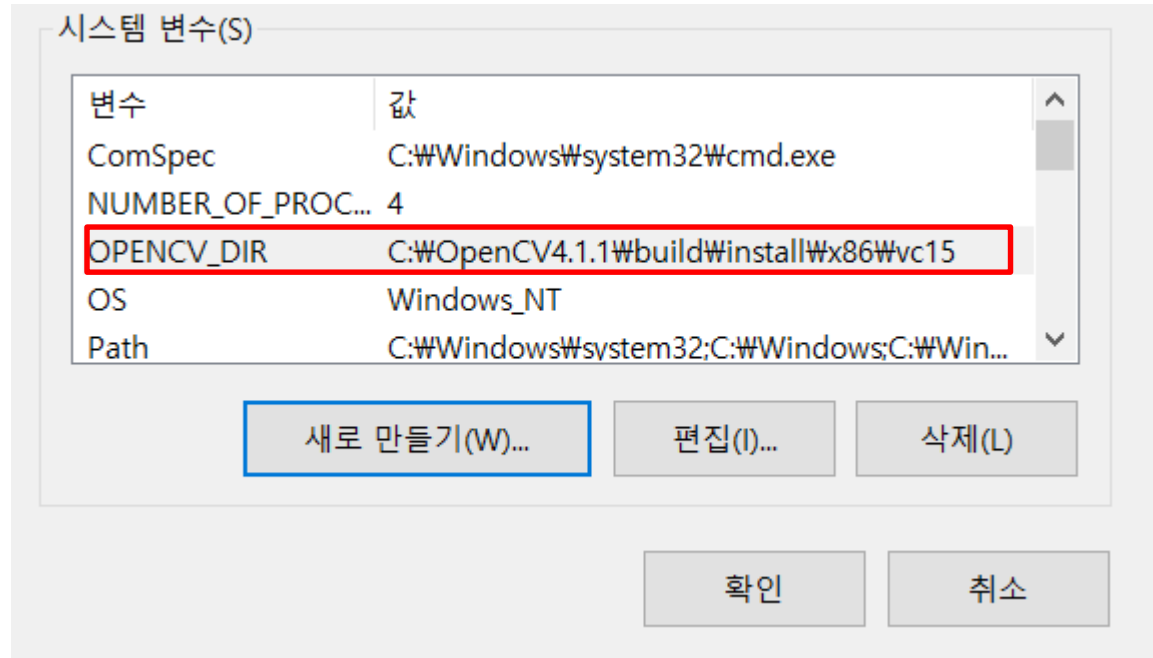
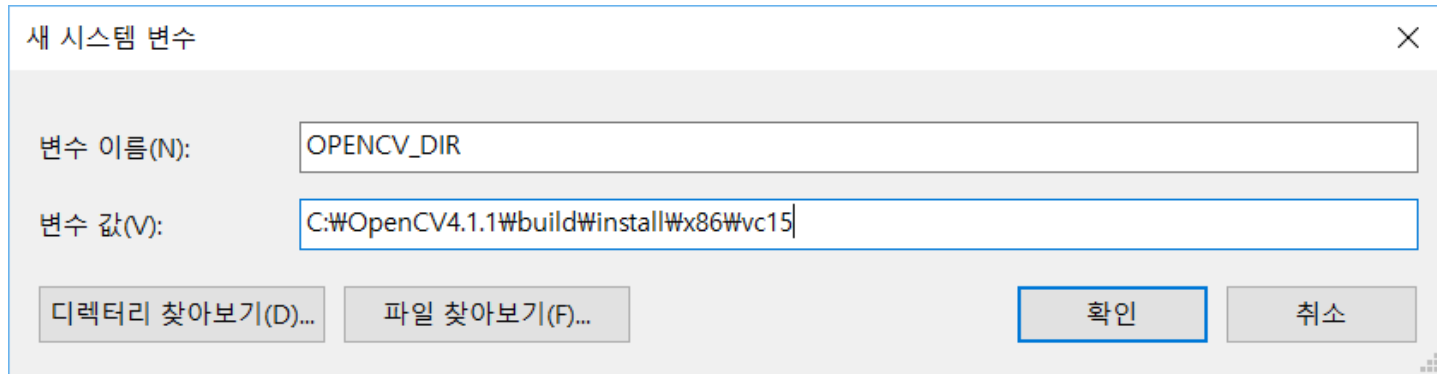


1.2 Open CV Build and Installation



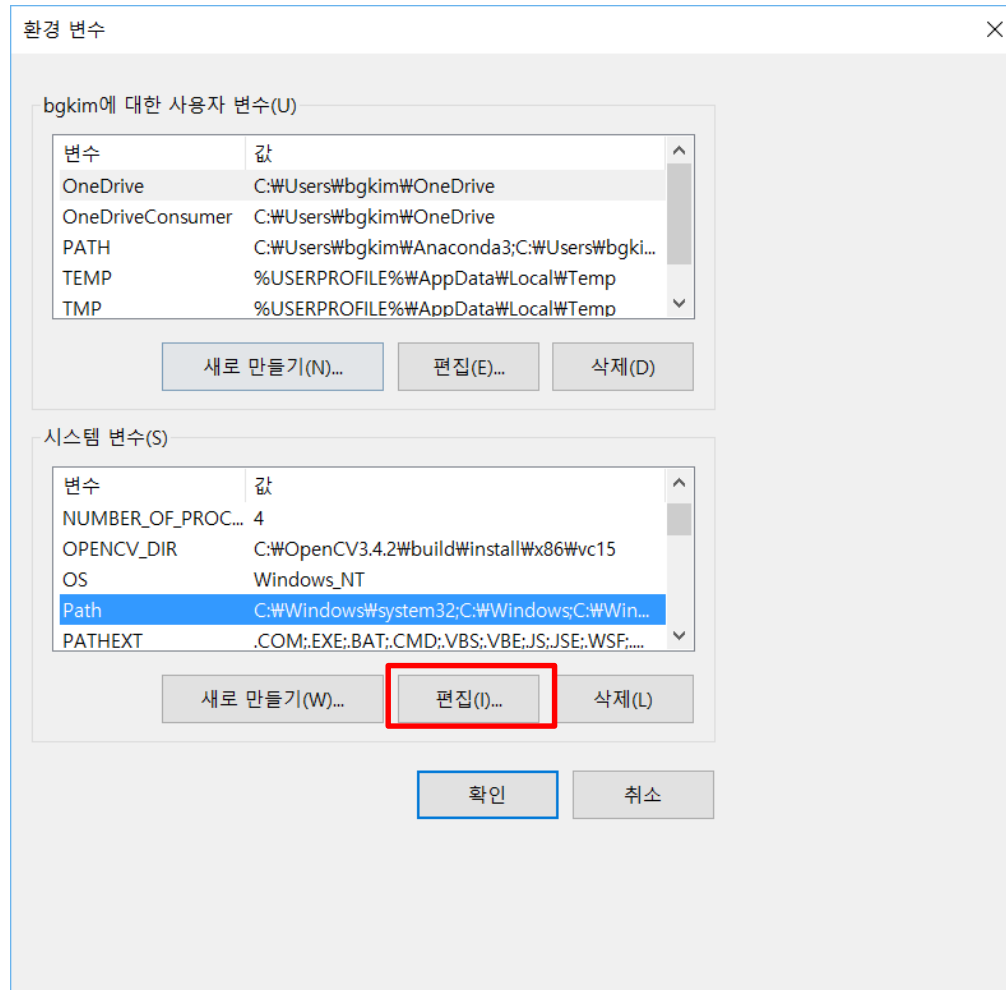
1.2 Open CV Build and Installation

- In "새로만들기" window, the value should be "vc12 directory inside *install* x64 or x86 directory"

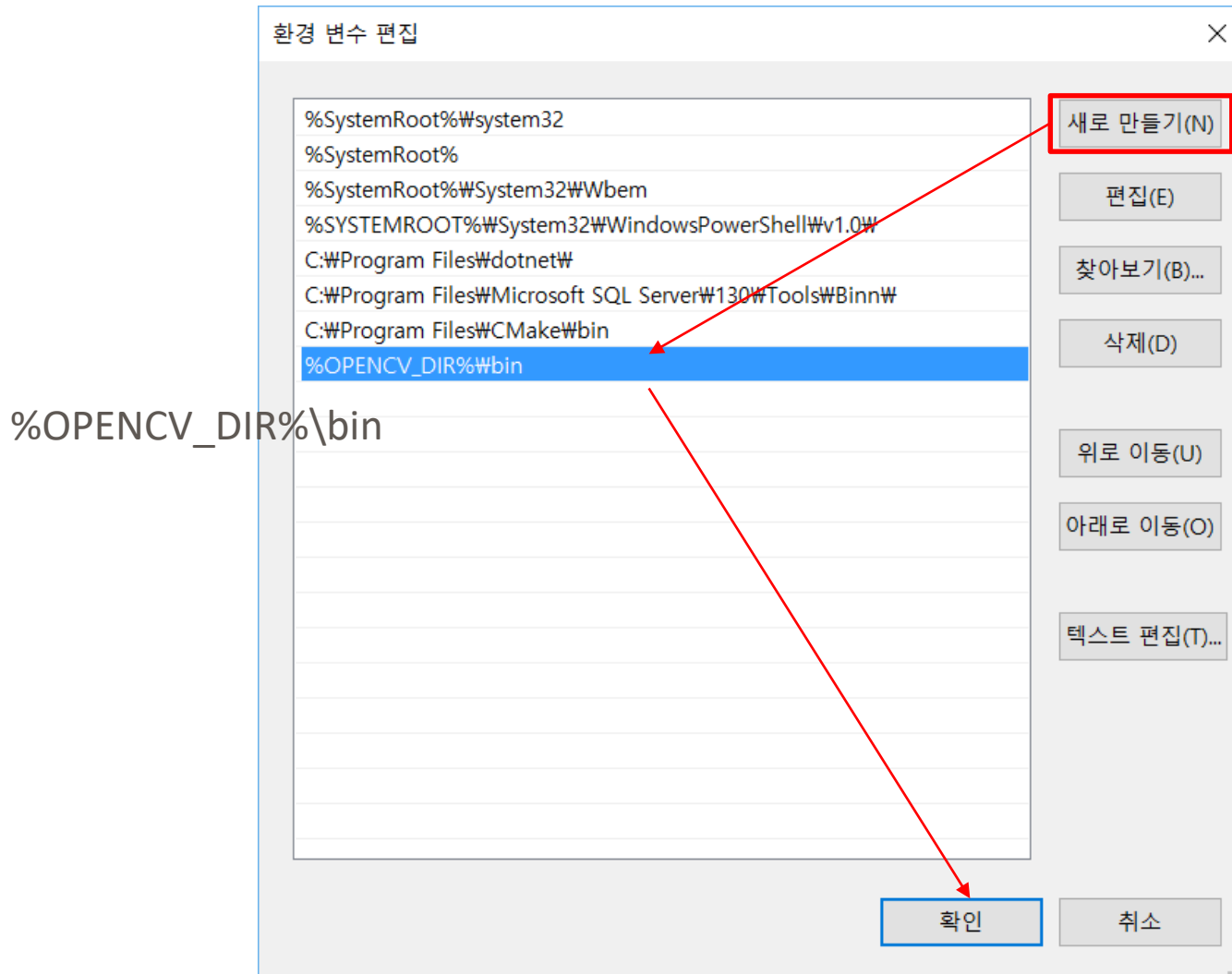


1.2 Open CV Build and Installation

- After that, we also need to modify the "Path" variable to register our OpenCV .dll directory. This can be done by adding %OPENCV_DIR%\bin entry at the last part of Path variable value.



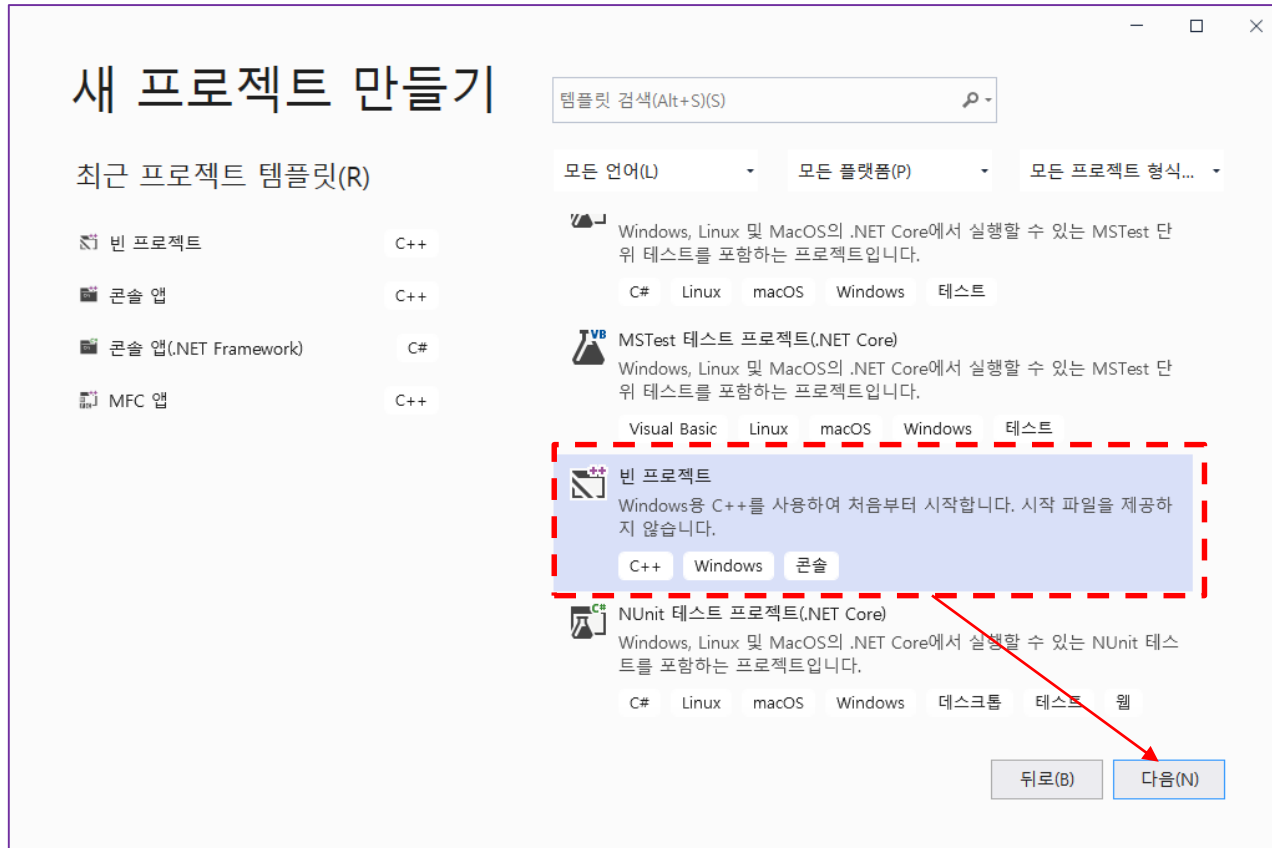
1.2 Open CV Build and Installation



*Finished registering OpenCV
Environment Variables*

1.2 Open CV Build and Installation

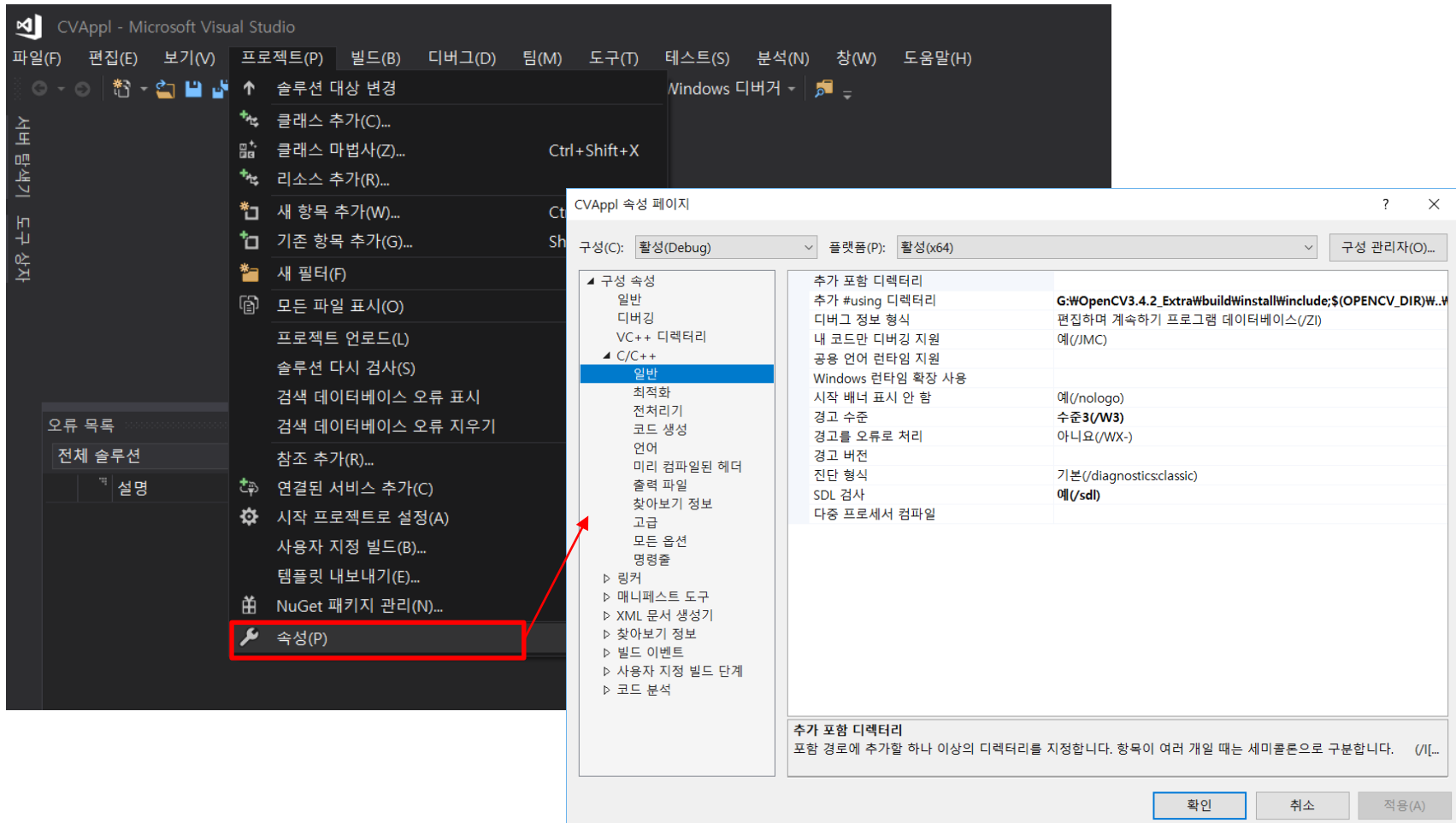
- 2. Creating OpenCV Project Property Sheet (**빈프로젝트 또는 Win32 Console mode**)



– After selecting project mode, you can set where you want to save this project....!!!

1.2 Open CV Build and Installation

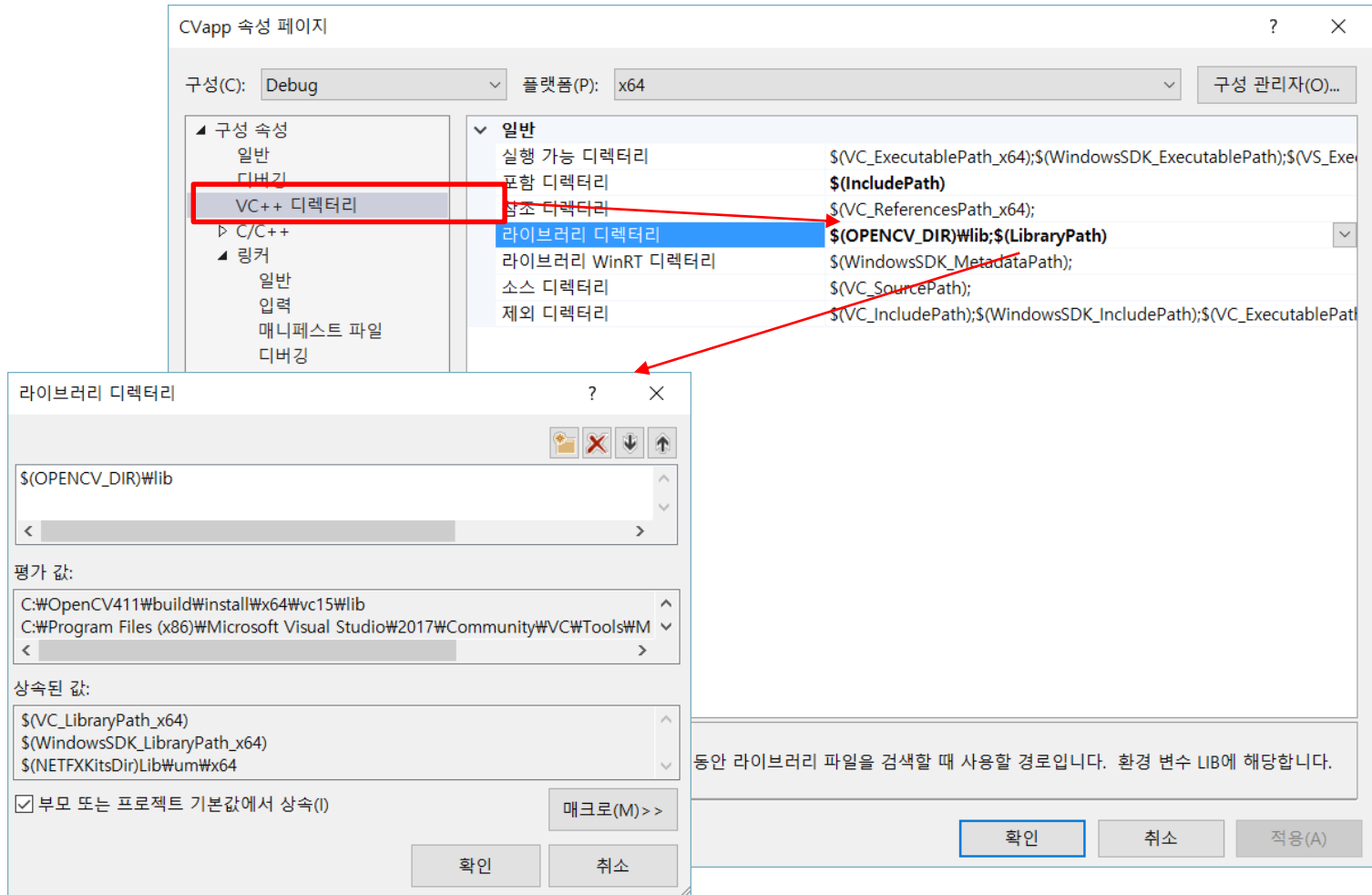
- 2. Creating OpenCV Project Property Sheet (**빈프로젝트 또는 Win32 Console mode**)
 - Configure the Additional Include Libraries (Common Properties->C/C++/General) by adding the following entry: **\$(OPENCV_DIR)\w..w..winclude**



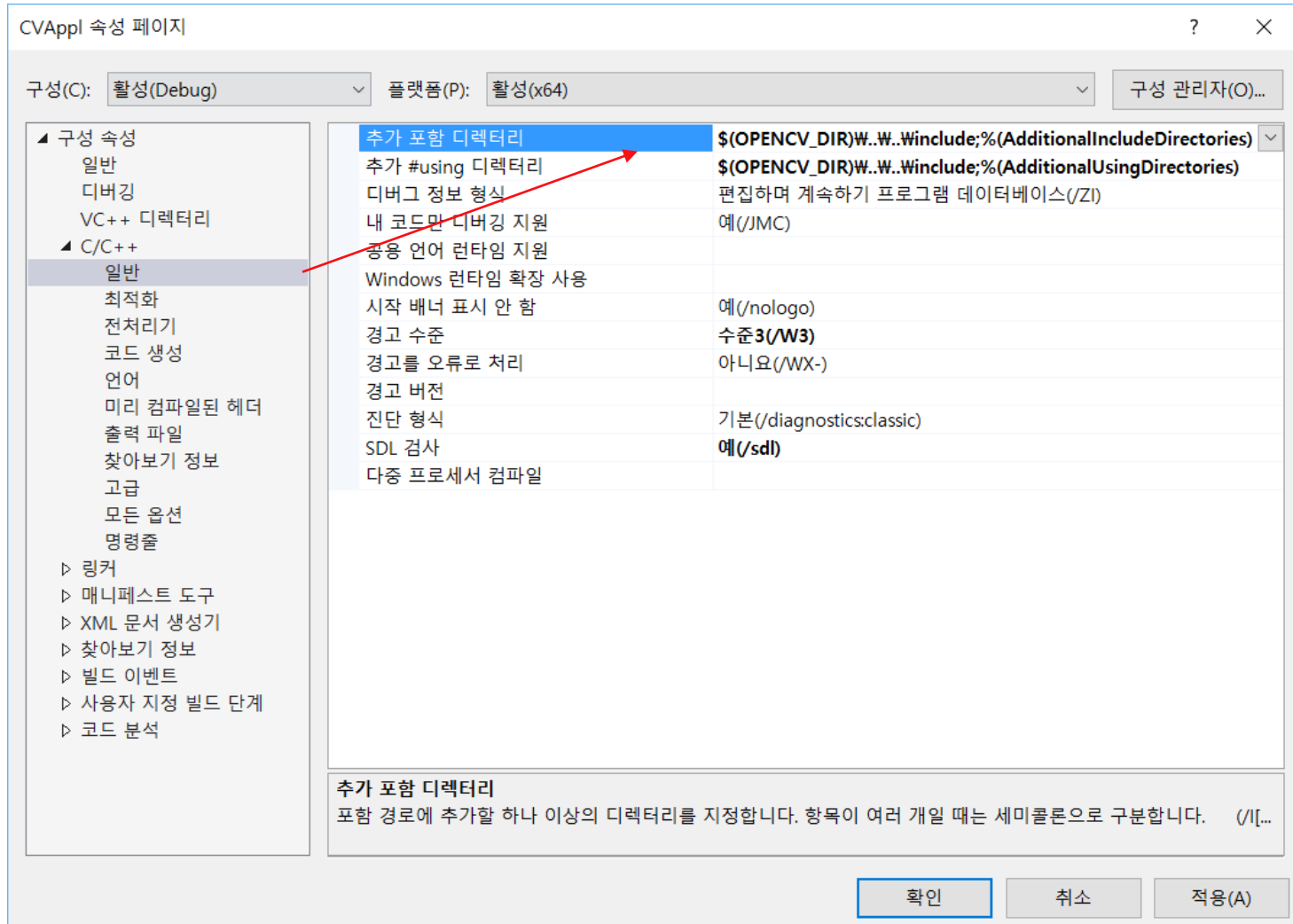
1.2 Open CV Build and Installation

• 2. Creating OpenCV Project Property Sheet

- Configure the Additional Include Libraries (Common Properties->VC++/library directory) by adding the following entry: **\$(OPENCV_DIR)\wlib 추가**

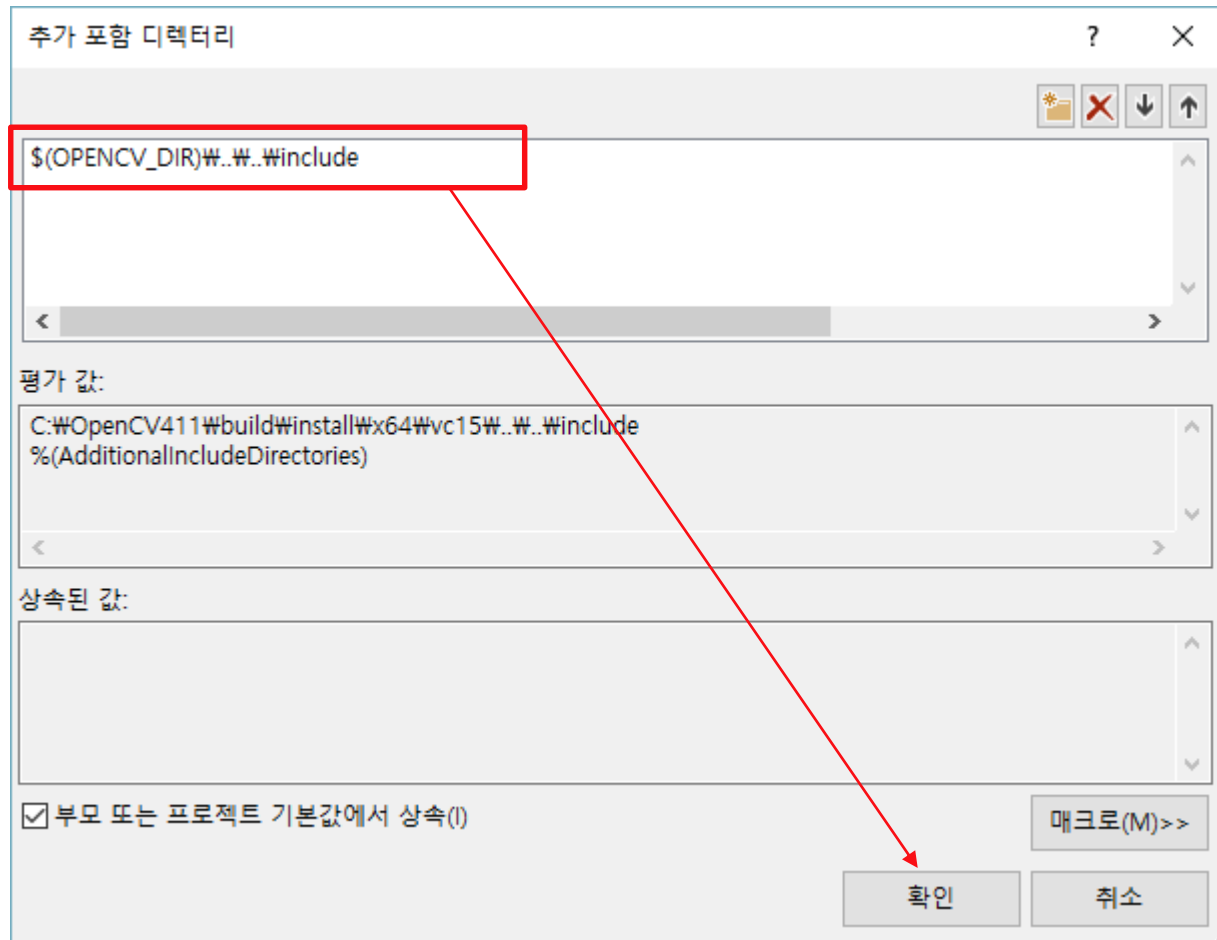


1.2 Open CV Build and Installation



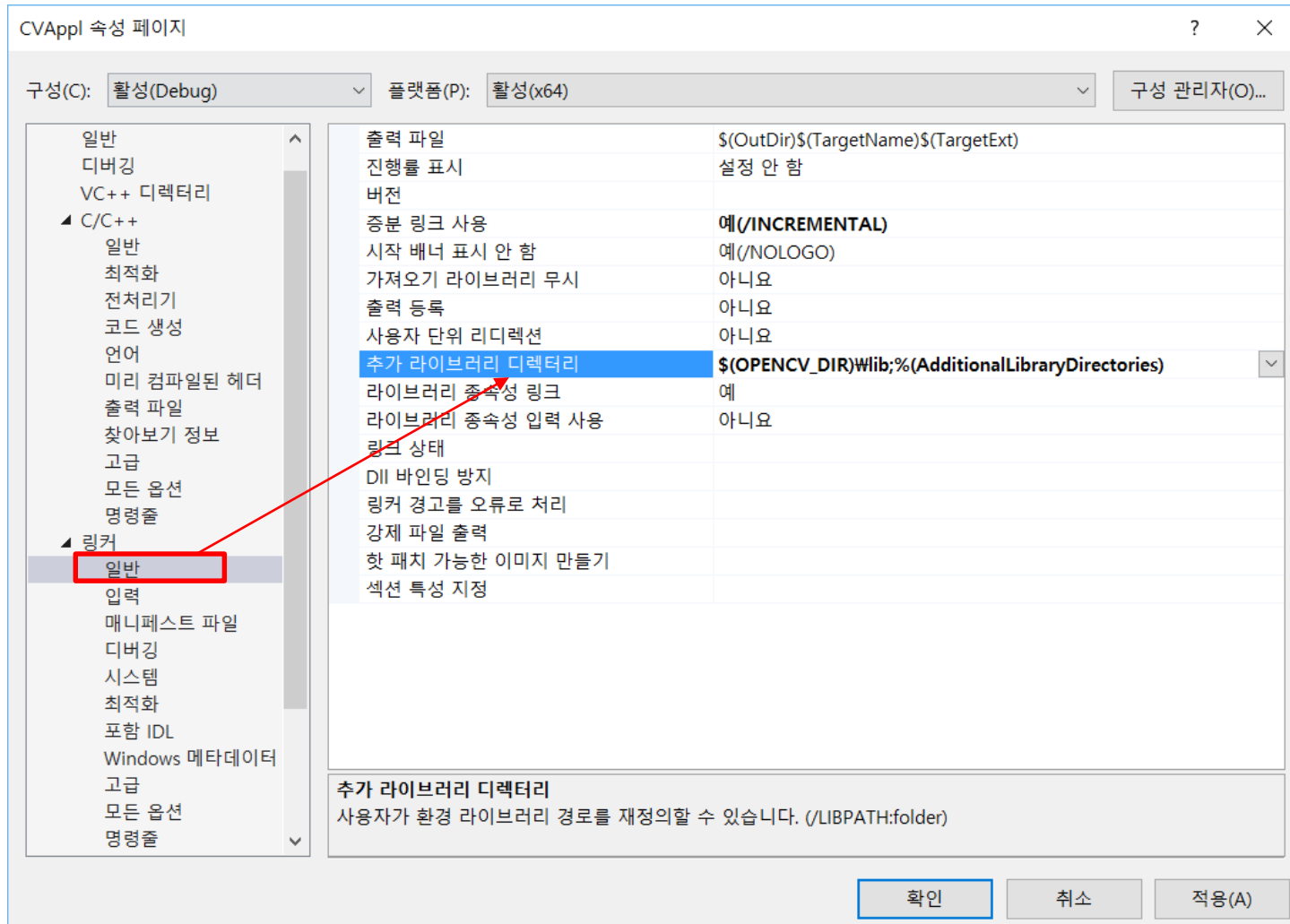
1.2 Open CV Build and Installation

- Add "\$ (OPENCV_DIR) \. \. \include" and click OK



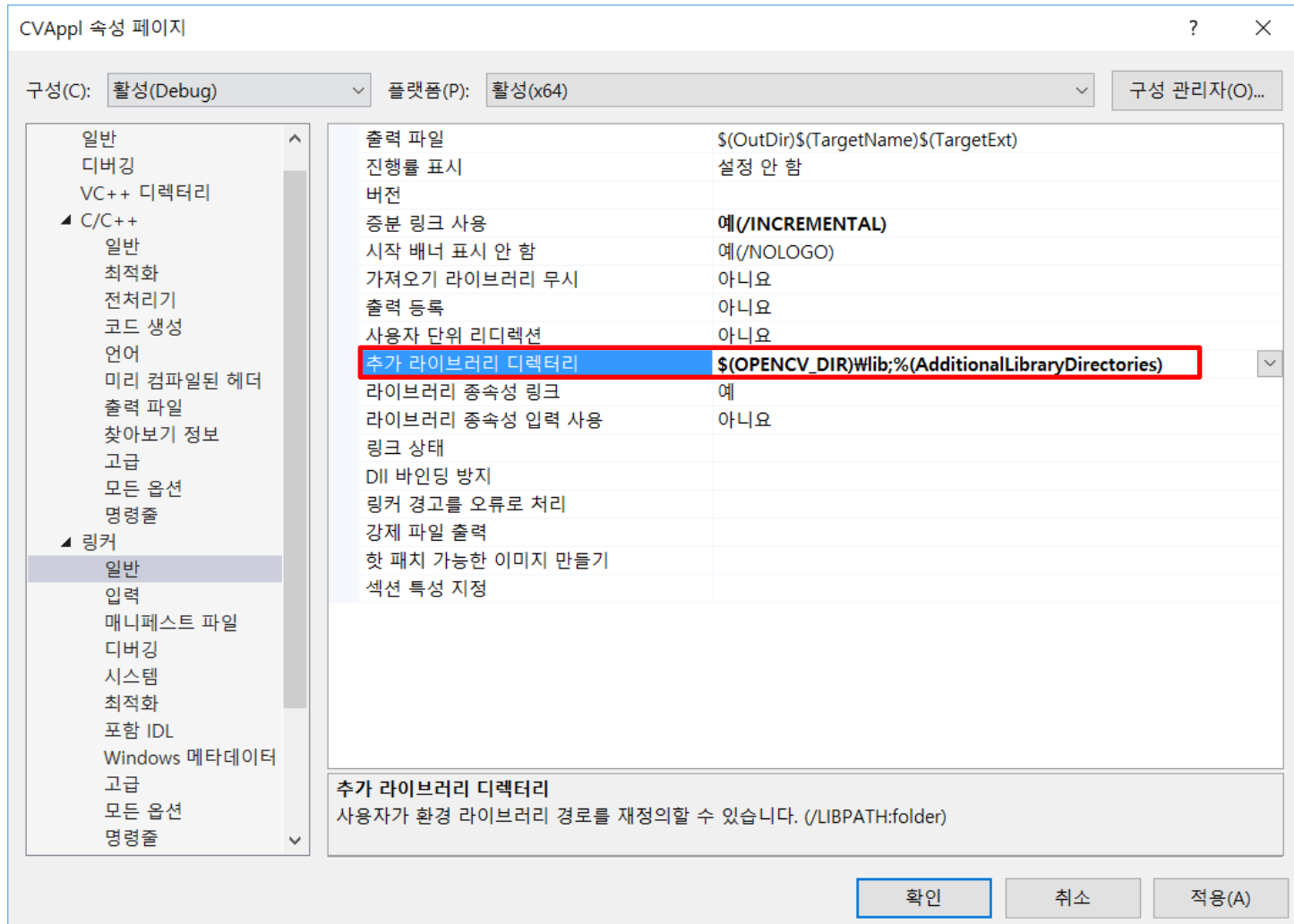
1.2 Open CV Build and Installation

- Configure the Additional Library Directories (Common Properties->Linker->General) by adding the following entry: **\$(OPENCV_DIR)\lib**



1.2 Open CV Build and Installation

- Configure the Additional Library Directories (Common Properties->Linker->General) by adding the following entry: **\$(OPENCV_DIR)\lib**



1.2 Open CV Build and Installation

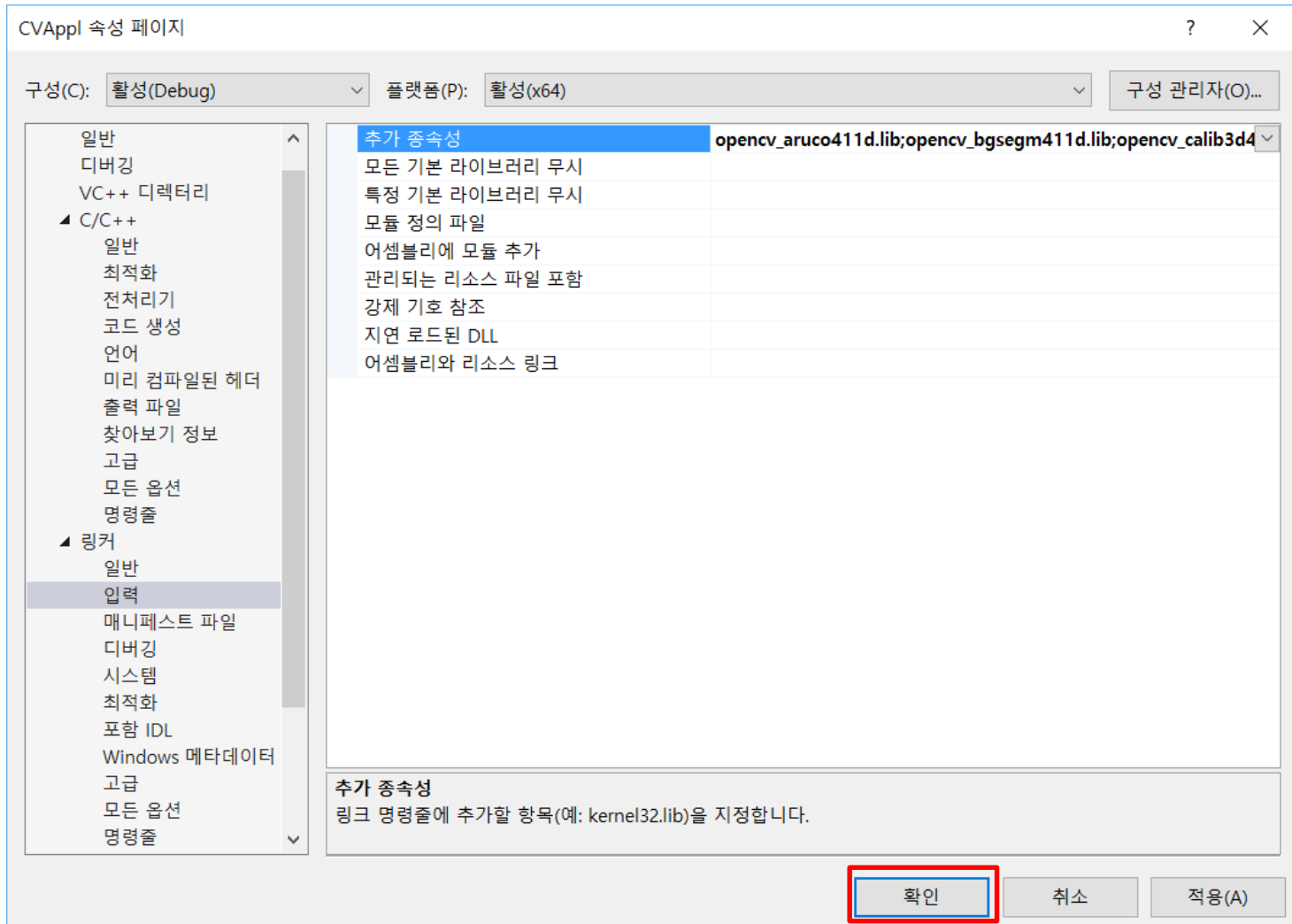
– List is the following:

opencv_world480.lib
opencv_world480d.lib

opencv_aruco411d.lib
opencv_bgsegm411d.lib
opencv_calib3d411d.lib
opencv_core411d.lib
opencv_dnn411d.lib
opencv_face411d.lib
opencv_features2d411d.lib
opencv_flann411d.lib
opencv_fuzzy411d.lib
opencv_gapi411d.lib
opencv_hdf411d.lib
opencv_hfs411d.lib
opencv_highgui411d.lib
opencv_img_hash411d.lib
opencv_imgcodecs411d.lib
opencv_imgproc411d.lib
opencv_line_descriptor411d.lib
opencv_ml411d.lib
opencv_objdetect411d.lib
opencv_optflow411d.lib
opencv_phase_unwrapping411d.lib
opencv_photo411d.lib
opencv_plot411d.lib
opencv_quality411d.lib
opencv_reg411d.lib
opencv_rgbd411d.lib
opencv_saliency411d.lib
opencv_shape411d.lib
opencv_stitching411d.lib
opencv_structured_light411d.lib
opencv_surface_matching411d.lib
opencv_video411d.lib
opencv_videoio411d.lib
opencv_videoio_ffmpeg411_64.lib
opencv_xfeatures2d411d.lib
opencv_ximgproc411d.lib
opencv_xobjdetect411d.lib
opencv_aruco342d.lib
opencv_bgsegm342d.lib
opencv_calib3d342d.lib
opencv_core342d.lib
opencv_dnn342d.lib
opencv_face342d.lib
opencv_features2d342d.lib
opencv_flann342d.lib
opencv_fuzzy342d.lib
opencv_gapi342d.lib
opencv_hdf342d.lib
opencv_hfs342d.lib
opencv_img_hash342d.lib
opencv_imgcodecs342d.lib
opencv_imgproc342d.lib
opencv_line_descriptor342d.lib
opencv_ml342d.lib
opencv_objdetect342d.lib
opencv_optflow342d.lib
opencv_phase_unwrapping342d.lib
opencv_photo342d.lib
opencv_plot342d.lib
opencv_quality342d.lib
opencv_reg342d.lib
opencv_rgbd342d.lib
opencv_saliency342d.lib
opencv_shape342d.lib
opencv_stitching342d.lib
opencv_structured_light342d.lib
opencv_surface_matching342d.lib
opencv_video342d.lib
opencv_xfeatures2d342d.lib
opencv_ximgproc342d.lib
opencv_xobjdetect342d.lib
opencv_xphoto342d.lib

1.2 Open CV Build and Installation

– Press "OK" button to finish the configuration.....!



1.3 Open CV Application Programming (1)

- Add C++ file and writing Image show program.....!

```
#include <opencv2/core.hpp>
#include <opencv2/videoio.hpp>
#include <opencv2/highgui.hpp>
#include <opencv2/opencv.hpp>
#include <opencv2/xfeatures2d.hpp>

using namespace cv;
using namespace std;

int main(int argc, char** argv){
    Mat image;

    char* imageName = argv[1];

    image = imread(imageName,
        IMREAD_COLOR);
    if (argc != 2 || !image.data){
        printf(" No image data \n ");
        return -1;
    }
}
```

```
Mat gray_image;
cvtColor(image, gray_image,
    COLOR_BGR2GRAY);

namedWindow( imageName,
    WINDOW_AUTOSIZE);
namedWindow( "Gray image",
    WINDOW_AUTOSIZE);

imshow( imageName, image);
imshow( "Gray image", gray_image);

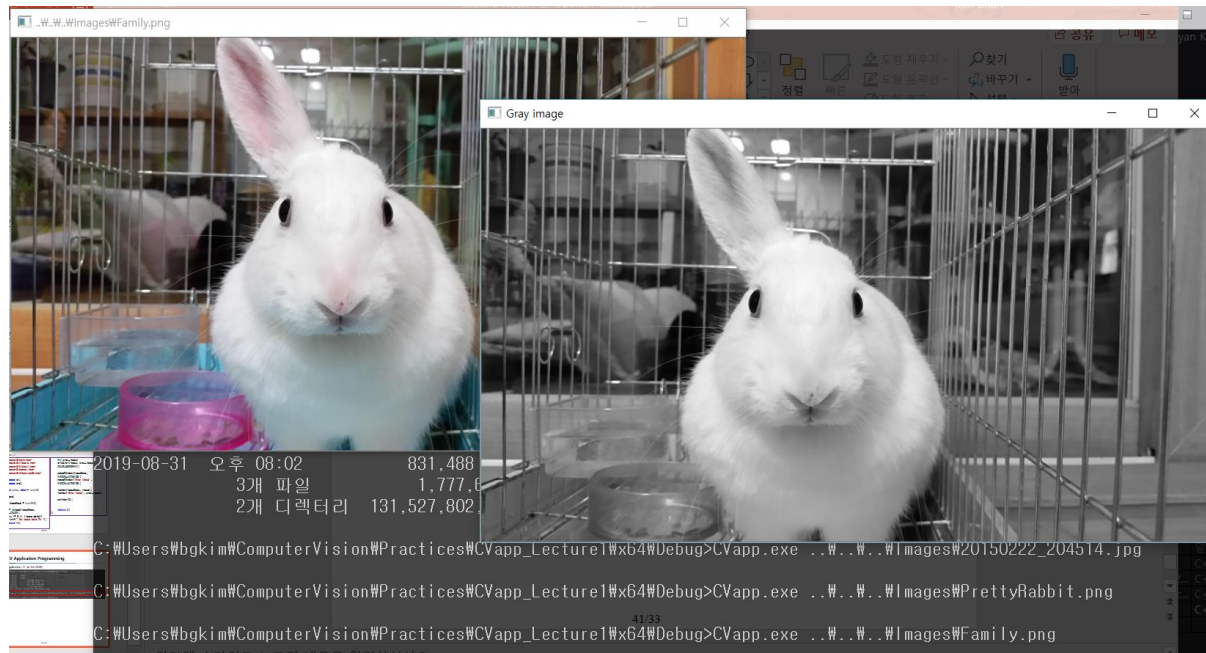
waitKey(0);

return 0;
}
```


1.3 Open CV Application Programming (2)

- Run the Application...!!! (in the CMD): >>실행 파일 영상파일 (enter)

```
C:\Users\bgkim\ComputerVision\Practices\CVapp_Lecture1\wx64\Debug 디렉터리
2019-08-31 오후 08:02 <DIR> Mat image;
2019-08-31 오후 08:02 <DIR> imshow("Gray image", gray_image);
2019-08-31 오후 08:02 char* imageName = argv[1];
2019-08-31 오후 08:02 153,088 CVapp.exe
2019-08-31 오후 08:02 793,092 CVapp.pdb
2019-08-31 오후 08:02 831,488 CVapp.pdb
3개 파일 1,777,668 바이트
2개 디렉터리 131,527,802,880 바이트 남음
C:\Users\bgkim\ComputerVision\Practices\CVapp_Lecture1\wx64\Debug>CVapp.exe ..\..\..\Images\20150222_204514.jpg
C:\Users\bgkim\ComputerVision\Practices\CVapp_Lecture1\wx64\Debug>
```



1.3 Open CV Application Programming (2)

- Webcam 구동 응용 프로그램

```
#include <opencv2/core.hpp>
#include <opencv2/videoio.hpp>
#include <opencv2/highgui.hpp>
#include <iostream>
#include <stdio.h>

using namespace cv;
using namespace std;

int main(int, char**)
{
    Mat frame;
    //--- INITIALIZE VIDEOCAPTURE
    VideoCapture cap;
    // open the default camera using default API
    // cap.open(0);
    // OR advance usage: select any API backend
    int deviceId = 0;          // 0 = open
    default camera
    int apiID = cv::CAP_ANY;   // 0 = autodetect
    default API
    // open selected camera using selected API
    cap.open(deviceID + apiID);
    // check if we succeeded
    if (!cap.isOpened()) {
        cerr << "ERROR! Unable to open camera\n";
        return -1;
    }
}
```

```
//--- GRAB AND WRITE LOOP
cout << "Start grabbing" << endl
<< "Press any key to terminate" << endl;
for (;;)
{
    // wait for a new frame from camera and
    store it into 'frame'
    cap.read(frame);
    // check if we succeeded
    if (frame.empty()) {
        cerr << "ERROR! blank frame
        grabbed\n";
        break;
    }
    // show live and wait for a key with
    timeout long enough to show images
    imshow("Live", frame);
    if (waitKey(5) >= 0)
        break;
}
// the camera will be deinitialized
automatically in VideoCapture destructor
return 0;
}
```

1.3 Open CV Application Programming (2)

- Run the Application...!!! (in the CMD): >>실행 파일 (enter)

COMPUTER VISION 비전 프로그래밍

Thank you and question?

