



# CHAPTER 1: OPENCV INSTALLATION

In this chapter, we will learn how to install OpenCV and use it in Microsoft Visual Studio. Installing OpenCV and using it in Visual Studio may seem very complex to the beginners. But if you follow the step by step procedure of this chapter properly, you will not face any kind of trouble. Let's get started.

For a beginner, processing OpenCV libraries and binaries is a complex task. And integrating them with Visual Studio makes it more challenging. In most of the cases, beginners stop trying after taking several attempts to use OpenCV. Because new problems keep showing up and things keep getting more complex. To make it easier to start with OpenCV, I have tried my best to represent the installation process in the simplest possible way. I hope the step by step procedure and visual support (photos) of this book will be enough for learners of any level (beginner, intermediate or advanced) to successfully run OpenCV in Visual Studio.

During installation process, if you make any mistake, you may get compile time error or other errors and complexities may arise. That is why, I recommend you to strictly follow all of the steps written in this chapter. After you have successfully completed all of the steps mentioned here, you can design your own installation procedure. Then you can modify few steps or you can do whatever you want. But during this learning period, follow the steps provided in this chapter strictly.

Installing OpenCV and configuring it for Visual Studio is a series of tasks. To make it easier for you, the entire process has been divided into three different steps. They are:

- Step 1:** Downloading all required software and install them,
- Step 2:** Processing OpenCV for Visual Studio and
- Step 3:** Linking OpenCV with Visual studio.

In this book OpenCV3.3.0, Visual Studio 2017 (community version) and Windows 10 (64 bit) operating system have been used. But this installation process will work for all previous version of OpenCV, any version of Visual Studio and Windows operating system.

## **1.1 Downloading and Installing Required Software (Step 1):**

We will use OpenCV in Microsoft Visual Studio. So we must have to download Visual Studio and OpenCV. To make OpenCV usable in Visual Studio, we need another software named Cmake. So, for entire procedure, we need three software. They are:

1. Visual Studio,
2. OpenCV and
3. Cmake.

Let's learn how to install these software first.

### **Installing Visual Studio:**

First download the Visual Studio 2017 or any other version you want. However, if you are an absolute beginner (what I am assuming), it is better to strictly follow what have been described in this book. Once you will have become expert in using OpenCV with Visual Studio, then you can do whatever you want. But for this learning period, it is recommended to strictly follow the instructions of this book. That means use Visual Studio 2017 (Community Version – because it is free). Go to the webpage to download Visual Studio: <https://www.visualstudio.com/downloads/>

At the beginning, an installer file will be downloaded. Run that file and the installation process will start. During Visual Studio 2017 installation, make sure you select the following packages (shown in figure 1.1):

1. Universal Windows Platform development,
2. .NET desktop development and
3. Desktop development for C++.

These packages are mandatory to use OpenCV in Visual Studio. There are other packages too. If you want to download some of them or all of them, there is no problem. But make sure you are not missing these three packages (mentioned above).

For better understanding, take visual support from the figure 1.1. Make sure you have selected 1, 2 and 3 (marked in red color), before clicking on the 'Install' button.

## Open Source Computer Vision for Beginners by Nuruzzaman Faruqui

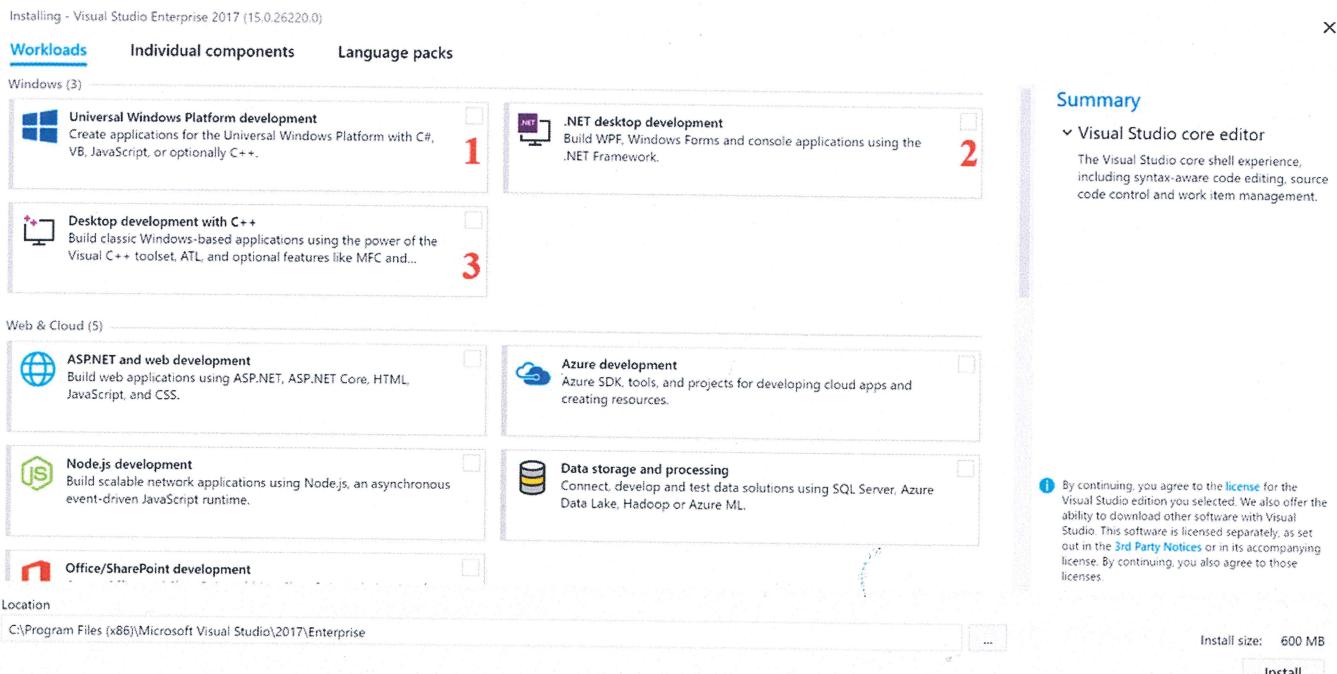


Figure 1.1: Necessary Packages of Visual Studio

If you want to use Visual Studio for free, you can use the Visual Studio Community version. The OpenCV runs in Visual Studio community version without any problem.

### Installing OpenCV3.3.0:

First download OpenCV3.3.0. Go to this webpage to download it: <https://opencv.org/releases.html>. You will see different releases of OpenCV. Either download OpenCV3.3.0 or download any release you want. But make sure you are downloading windows version. The windows version is named as 'Win pack' (figure 1.2). Again I like to mention that it is recommended to download OpenCV3.3.0 for this learning period. Once you will have understood the concept of installing and running OpenCV in Visual Studio, you can choose any version you want. However, to keep you away from facing any kind of compilation error or other complexities, I recommend to use OpenCV3.3.0.

For better understanding, take visual support from Figure 1.2. The windows version is enclosed by a red ellipse in the figure.

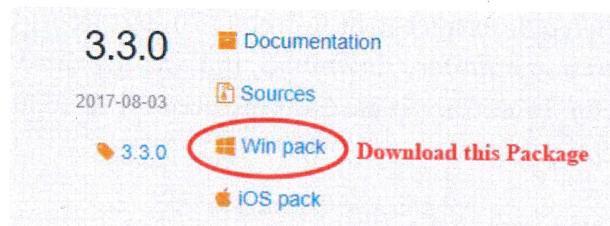


Figure 1.2: OpenCV Downloading

Once the download is completed, run the file and extract it in C drive. I recommend to extract it in C drive to reduce the complexity in upcoming steps. The following image will help you to understand better.

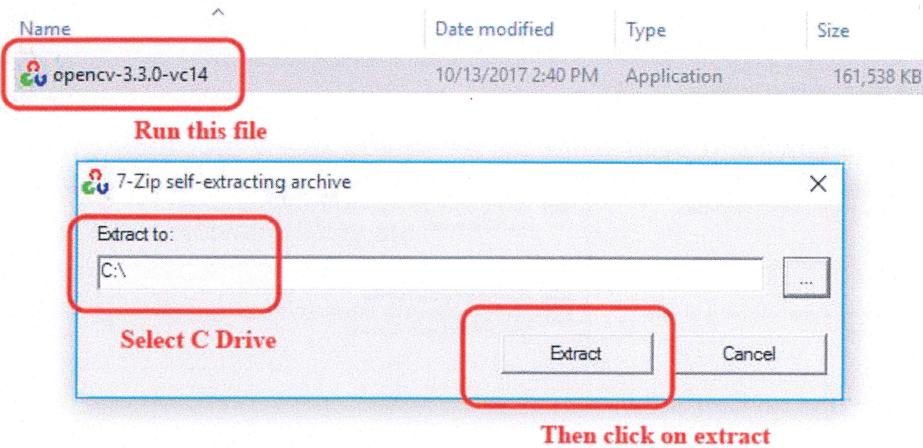


Figure 1.3: OpenCV Extraction

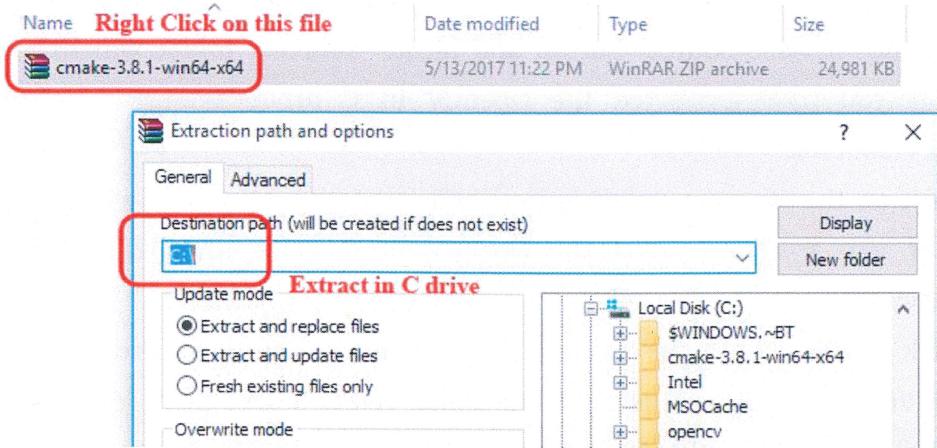
After the extraction, you will see that two folders named ‘build’ and ‘sources’ have been created. The ‘sources’ folder contains all the modules, include directories, examples and everything. The ‘build’ folder contains the binaries. Once you will have processed OpenCV libraries (Shown in step 2), most of the things we will use, will be inside this ‘build’ folder.

### Installing Cmake:

Cmake is an open source software tool. It is used to control compilation process of software system. We use Cmake because it can generate platform-independent configuration files of software modules. In our case, we will use Cmake to generate OpenCV libraries, PDB (Program Database) and DLL (Dynamic Link Library) for Visual Studio.

First download Cmake. If you are using 32 bit operating system, then download 32 bit Cmake and if you are using 64 bit operating system, then download 64 bit Cmake. Download the ZIP version. You can download the installer version if you want. But I prefer downloading the ZIP version. Go to this webpage to download the Cmake: <https://cmake.org/download/>

Once the download is completed, extract it in C drive. (To extract it, you need WinRAR. If you don't have WinRAR installed in your computer, download and install the WinRAR). You can choose any location you want. But I prefer to extract it in C drive. Because it reduces the complexity. Figure 1.4 illustrates the Cmake extraction process.



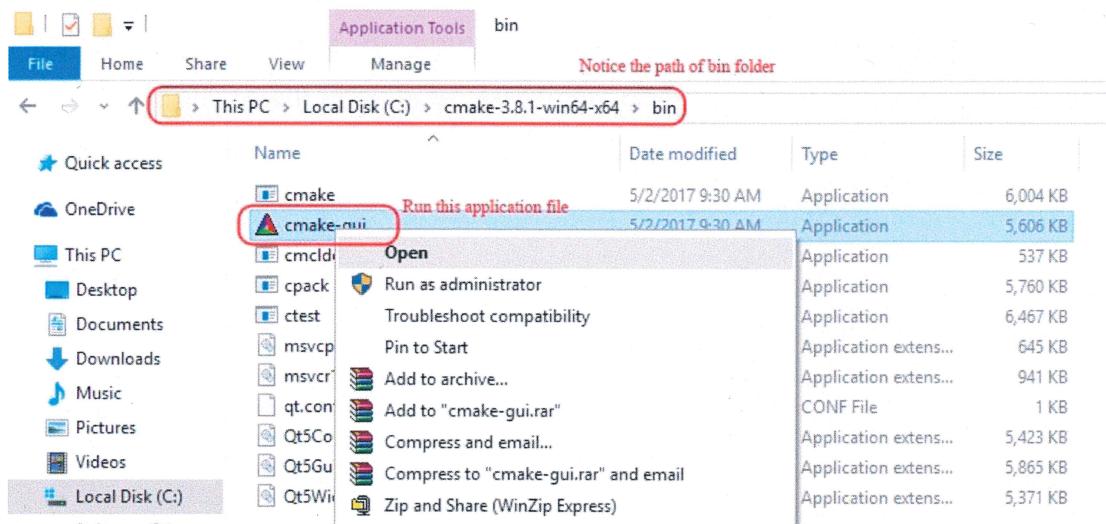
**Figure 1.4:** Cmake Extraction

There are different versions of Cmake. Download the latest version. Cmake has different installer for 32 bit operating system and 64 bit operating system. So, download the appropriate version.

*This is the end of step 1. You have downloaded and installed all the necessary software. Now proceed to step 2.*

## 1.2 Processing OpenCV for Visual Studio (Step 2)

In this step, we will process OpenCV libraries and binaries to make them usable in Visual Studio. First of all, go to the folder where you extracted Cmake. Get inside the Cmake folder. You will see several folders there. Locate and get inside of 'bin' folder. Then run 'cmake-gui.exe'. Figure 1.5 shows the icon that you have to use.

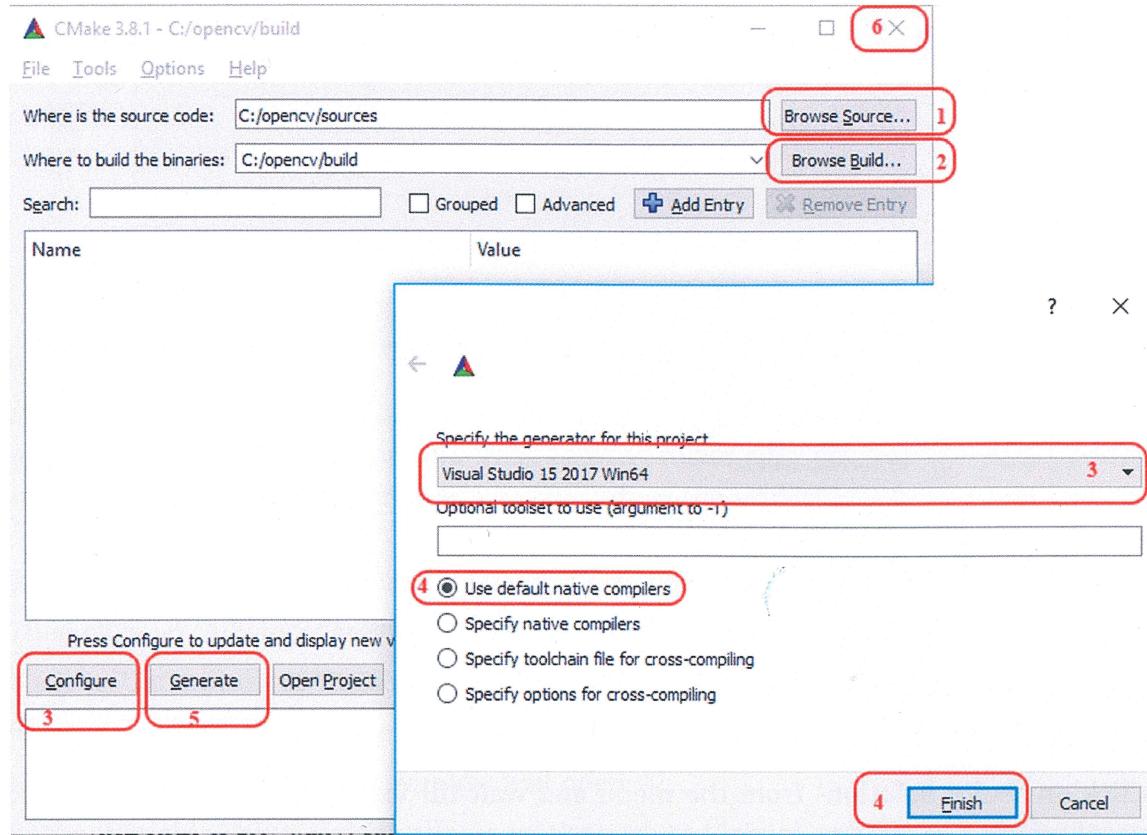


**Figure 1.5:** Running Cmake

After running the cmake-gui.exe, Graphical User Interface (GUI) will show up (figure 1.6). Now –

1. Click on the ‘Browse Source’ button and locate ‘sources’ folder which is inside the OpenCV folder. If you have extracted the OpenCV in C drive, then the path to the folder is: C:\opencv\sources.
2. Now click on the ‘Browse Build’ button and locate ‘build’ folder which is inside the OpenCV folder. If you have extracted the OpenCV in C drive, then the path to the folder is: C:\opencv\build.
3. Now click on the ‘Configure’ button. A new window will open. From there, select version of Visual Studio that you are using. I am using Visual Studio 2017 (64 bit) that is why I have selected ‘Visual Studio 15 2017 Win64’ (Figure: 1.6). The version you are using may be different. Choose the appropriate version.
4. And mark ‘Use default native compilers’ located in the same window. Now press ‘Finish’ button. Cmake will start the generating process. Wait till the process is completed.
5. Now click on the ‘Generate’ button and wait till the process is completed.
6. Now close Cmake.

Figure 1.6 shows the steps. If you face any trouble in following the steps above, take visual support from the figure 1.6.



**Figure 1.6:** Using Cmake

Now we will build solution of OpenCV libraries using Visual Studio. Follow the steps to easily complete this process:

1. Go to the folder where you have installed OpenCV. Get inside the folder.
2. Now get inside the 'build' folder.
3. Now find out a file named 'OpenCV'. The file type is 'Microsoft Visual Studio Solution' (figure 1.7). Run this file. It will open in Visual Studio. Wait till Visual Studio is fully loaded and ready. It may take 5-10 minutes. To locate the exact file, take help from figure 1.7:

Name	Date modified	Type	Size
CMakeVars	11/1/2017 10:08 PM	Text Document	175 KB
CPackConfig.cmake	11/1/2017 10:08 PM	CMAKE File	10 KB
CPackSourceConfig.cmake	11/1/2017 10:08 PM	CMAKE File	10 KB
CTestTestfile.cmake	11/1/2017 10:08 PM	CMAKE File	1 KB
custom_hal.hpp	11/1/2017 10:08 PM	C/C++ Header	1 KB
cv_cpu_config.h	11/1/2017 10:08 PM	C/C++ Header	1 KB
cvconfig.h	11/1/2017 10:08 PM	C/C++ Header	6 KB
INSTALL.vcxproj	11/1/2017 10:08 PM	VC++ Project	7 KB
INSTALL.vcxproj.filters	11/1/2017 10:08 PM	VC++ Project Filters File	1 KB
LICENSE	1/23/2017 4:44 PM	File	3 KB
OpenCV	11/1/2017 10:09 PM	Microsoft Visual Studio Solution	94 KB
opencv_modules.vcxproj	11/1/2017 10:08 PM	VC++ Project	56 KB
opencv_modules.vcxproj.filters	11/1/2017 10:08 PM	VC++ Project Filters File	1 KB
opencv_perf_tests.vcxproj	11/1/2017 10:08 PM	VC++ Project	55 KB
opencv_perf_tests.vcxproj.filters	11/1/2017 10:08 PM	VC++ Project Filters File	1 KB
opencv_tests.vcxproj	11/1/2017 10:08 PM	VC++ Project	56 KB
opencv_tests.vcxproj.filters	11/1/2017 10:08 PM	VC++ Project Filters File	1 KB
opencv_tests_config.hpp	11/1/2017 10:08 PM	C/C++ Header	1 KB
OpenCVConfig.cmake	8/4/2017 6:28 AM	CMAKE File	12 KB
OpenCVConfig-version.cmake		CMAKE File	1 KB
OpenCVModules.cmake	11/1/2017 10:09 PM	CMAKE File	19 KB
PACKAGE.manifest	11/1/2017 10:09 PM	Manifest	7 KB

Figure 1.7: Finding OpenCV Solution

4. Now select ‘Debug’ from the ribbon.
5. Click on ‘Build’ tab. A menu will open.
6. Click on ‘Build Solution’ from the menu and wait till Visual Studio builds all the solutions. may take 10–15 minutes. The figure 1.8 illustrates the steps–

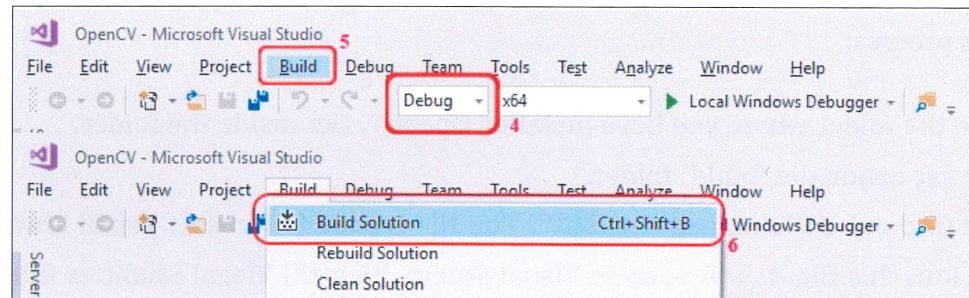
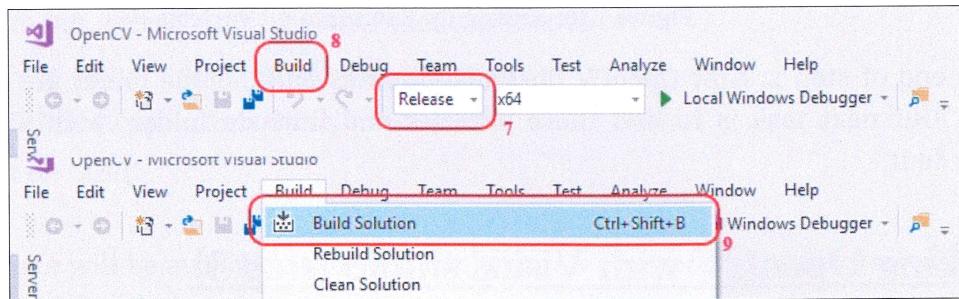


Figure 1.8: Building Debug Solution

7. Now select ‘Release’ from the ribbon
8. Click on the ‘Build’ tab. A menu will open.
9. Click on ‘Build Solution’ from the menu and wait till Visual Studio builds all the solutions. may take 10–15 minutes. Close Visual Studio when the process will have completed successfully. If you are confused, take help from figure 1.9 :

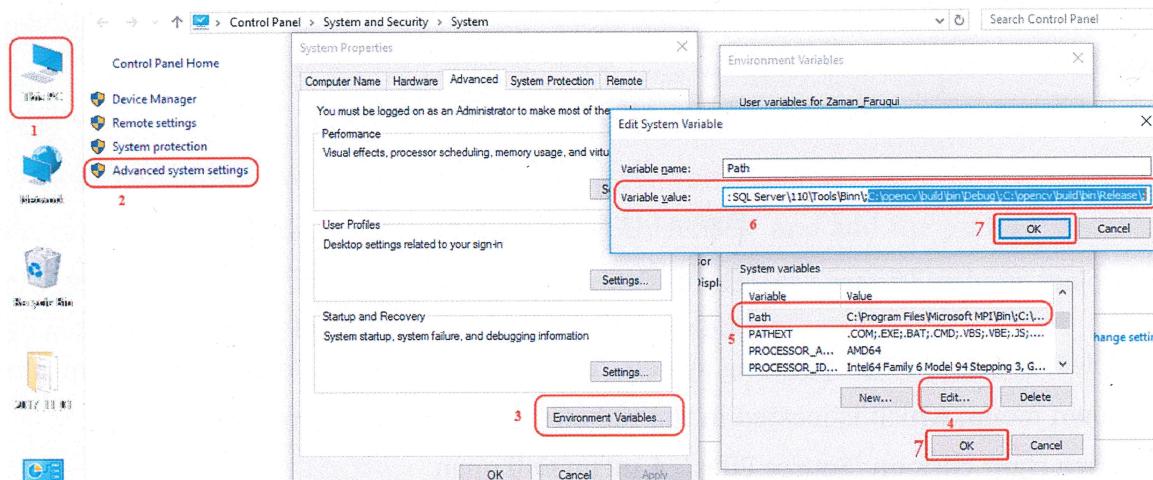


**Figure 1.9: Building Release Solution**

Next task is to set up Environmental Variables (figure 1.10). If you do not set up the environmental variable properly, the OpenCV will not run. That is why, make sure you are not making any mistake while setting up the environmental variable. To do it, follow the steps:

1. Right click on 'This PC', mostly known as 'My Computer', then click on properties.
2. Then click on 'Advanced system settings'. A new window will open.
3. Click on 'Environment Variable' from there. Another window will open.
4. In the 'System variables' use scroll bar and locate and select 'path'.
5. Now click on 'Edit'. Another window will open.
6. In the 'Variable Value' field, write down the path to 'debug' and 'release' folder located inside 'bin' folder of 'build' folder of 'OpenCV' (highlighted in Figure: 1.10).
7. And of course save all the changes by clicking 'ok' buttons.

If you have installed OpenCV in C drive, then the path to debug and release folders will be 'C:\opencv\build\bin\Debug' and 'C:\opencv\build\bin\Release' respectively. Make sure you separate these two lines like this: C:\opencv\build\bin\Debug\;C:\opencv\build\bin\Release\; inside 'Variable value' field. The entire process has been shown in figure 1.10 to make it easier for you.



**Figure 1.10: Setting up Environment Variable**

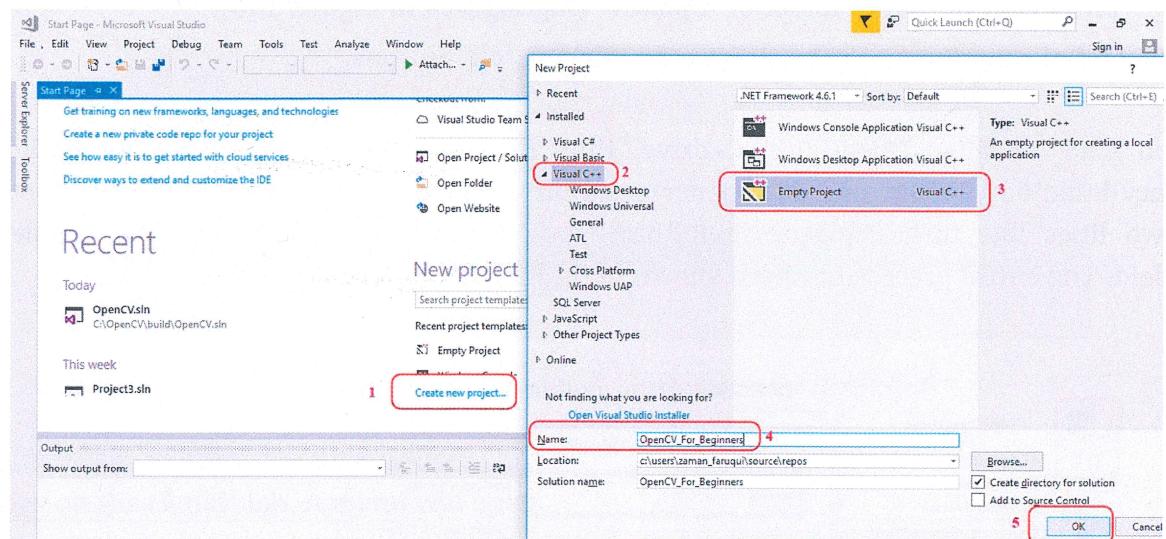
This is the end of step 2. Now OpenCV libraries, binaries and include folder are ready to be used in Visual Studio. Our next task is to link these libraries and ‘include folder’ with Visual Studio. Step 3 shows how to do it.

### **1.3 Linking OpenCV with Visual studio (step 3)**

This is the final step. In this step, we will link OpenCV ‘Include folder’ and libraries with Visual Studio. Follow the steps to successfully do it (figure 1.11):

1. Launch Visual Studio
2. Then click on ‘Create new project...’. A new window will open.
3. Click on Visual C++ and select ‘Empty Project’
4. Give a name to your project. In this project I am naming it as ‘OpenCV\_For\_Beginners’.
5. Now press ‘OK’ and the project will be created.

These steps have been illustrated in Figure 1.11. This figure will help you to understand better.



**Figure 1.11: Creating a C++ Project**

In the project window (figure 1.12), do the followings:

1. Select x86 for 32 bit operating system or select x64 for 64 bit operating system from the drop down menu, which is located in the ribbon (shown in the figure 1.12)
2. Then right click on ‘Source Files’. A menu will open.

3. From the menu, select 'Add'. Another menu will open.
4. Click on 'New Item..'. Another window will open.
5. Select 'C++ File (.cpp)' from there.
6. Then click on 'Add'.

The coding interface will be available. Now you can write C++ code here. In the next section, you will learn how to link OpenCV with this project.

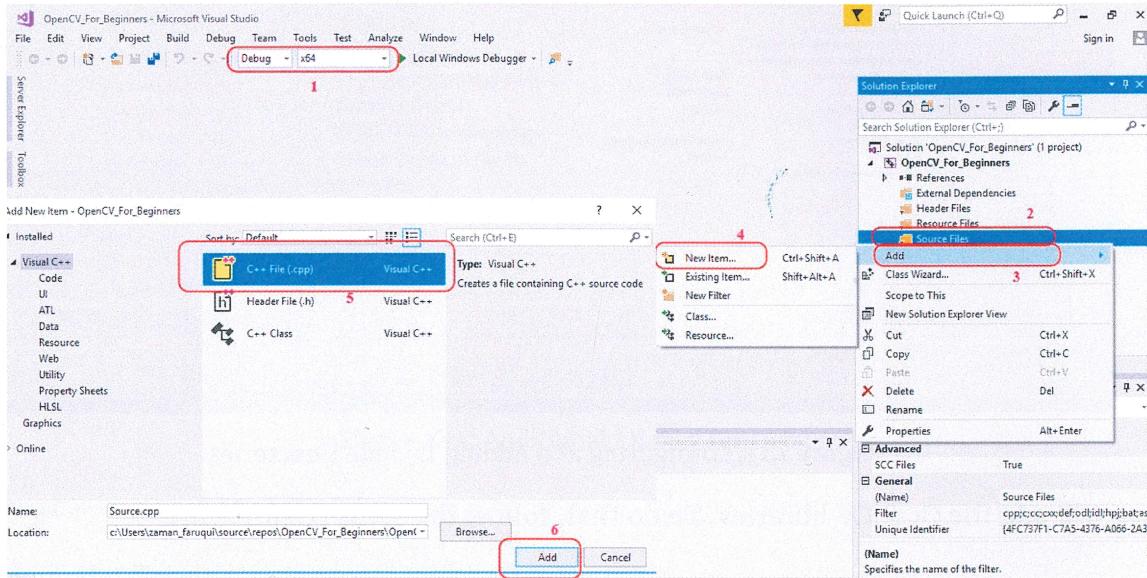
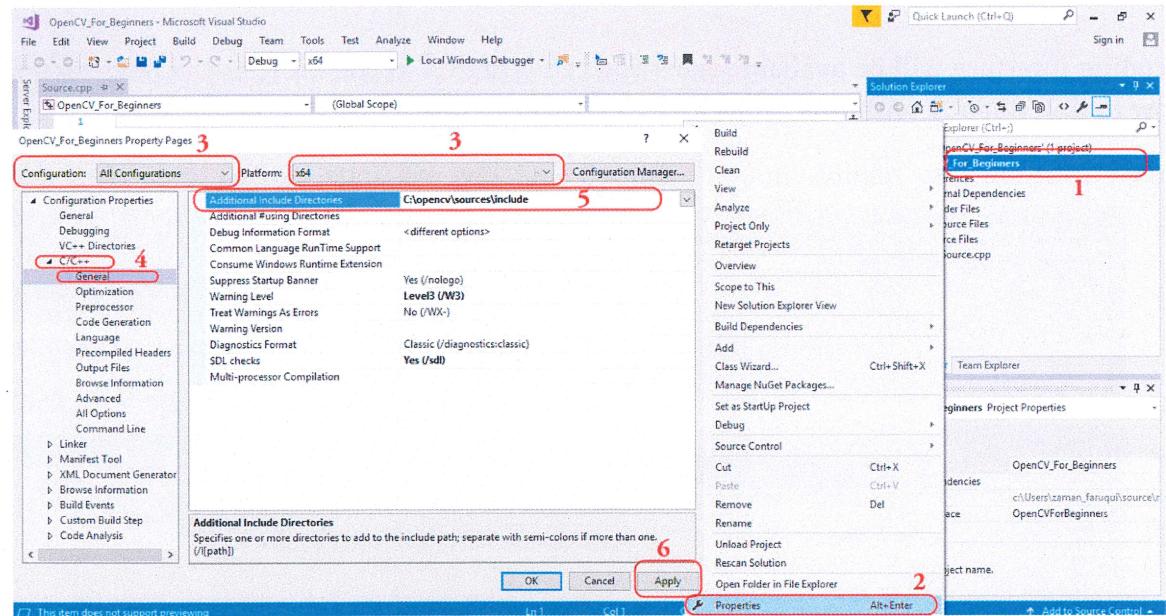


Figure 1.12: Adding New Source

To link OpenCV with this project follow the steps (figure 1.13):

1. Right click on the name of the project (Here the name is OpenCV\_For\_Beginners). A menu will appear.
2. Click on the 'Properties' from the menu. A new window will open.
3. From there select 'All Configuration' and appropriate version of your operating system (win32 for 32 bit and x64 for 64 bit).
4. Then click on dropdown arrow of 'C/C++' and click on 'General'.
5. On the right side, locate 'Additional Include Directories' and add the path of 'include' folder of OpenCV and press 'Enter' from keyboard (Figure: 1.13).
6. Then click on 'Apply' button. But do not close the window.

The ‘include’ folder is located at ‘C:\opencv\build\include’ if you have installed the OpenCV in C drive. If you have installed in other location, look for the ‘include’ folder inside the ‘build’ folder of OpenCV. Figure 1.13 illustrates these steps:



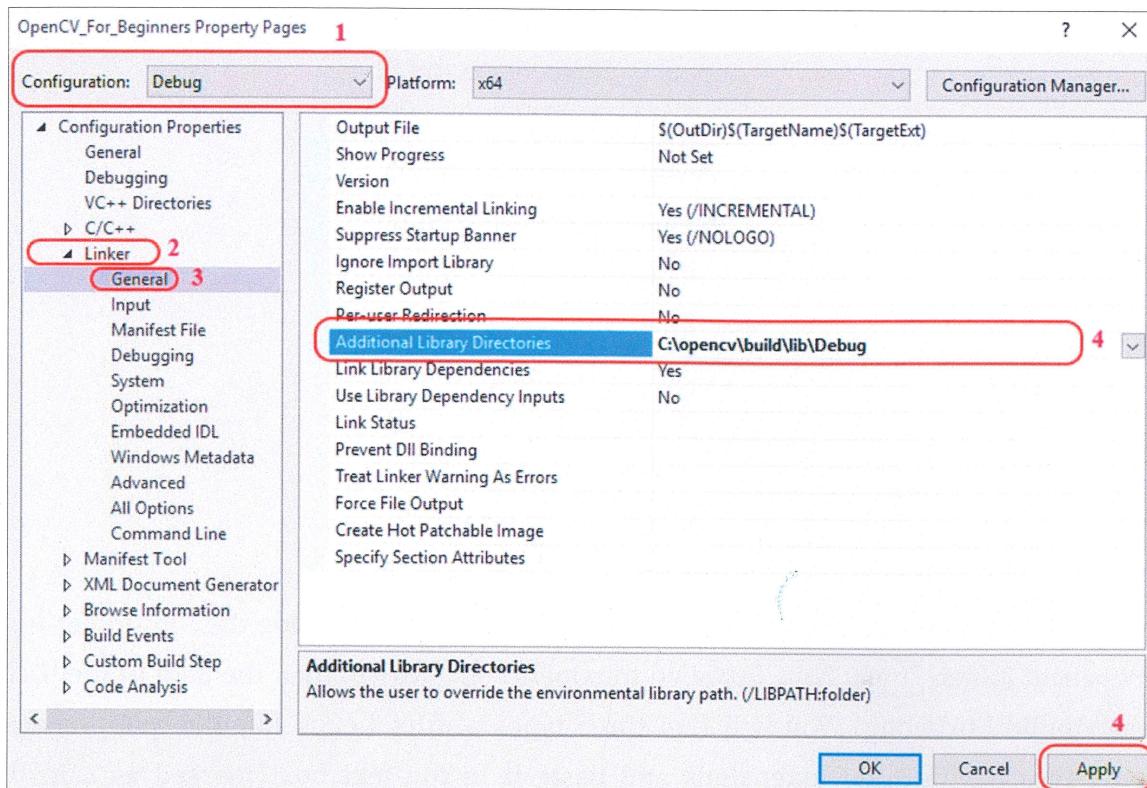
**Figure 1.13: Configuring and Adding Include Directories**

Now we will add the OpenCV libraries. To do that, follow the steps (figure 1.14):

1. Select ‘Debug’ from ‘Configuration’,
2. Then click on the dropdown arrow of ‘Linker’,
3. Click on ‘General’,
4. Now on the right side in ‘Additional Library Dependencies’ field add the path of ‘Debug’ folder and hit ‘Enter’ from Key board. Then click ‘Apply’ button. But do not close the window (Figure: 1.14).

The ‘Debug’ folder is located inside ‘lib’ folder, which is located inside ‘build’ folder of OpenCV. The path is ‘C:\opencv\build\lib\Debug’ if you have extracted OpenCV in C drive.

Figure 1.14 illustrates these steps. If you have confusion, visual support from figure 1.14 will help you to easily understand.



**Figure 1.14:** Additional Library for Debug

Now on the same window (figure 1.16):

1. Click on the 'Input'.
2. Then locate 'Additional Dependencies' on the right side and click on the dropdown icon.
3. Click on the '<Edit...>' option. A new window will open.
4. In this window, there is a text box. In the text box you have to write down the name of debug libraries with extension. If you are using OpenCV3.3.0, then the name of the debug libraries will be 'opencv\_name of the library330d.lib'. If you use OpenCV 2.4.7, the name of the debug libraries will be 'opencv\_name of the library247d.lib'. The format of debug library is – 'opencv\_name of the libraryx<sub>1</sub>x<sub>2</sub>x<sub>3</sub>d.lib' where x<sub>1</sub>x<sub>2</sub>x<sub>3</sub> represents the version of OpenCV, and the 'd' represents the 'debug library'. The list of the debug libraries of OpenCV3.3.0 has been listed below –

```

opencv_calib3d330d.lib
opencv_core330d.lib
opencv_dnn330d.lib
opencv_features2d330d.lib
opencv_flann330d.lib

```

```

opencv_highgui330d.lib
opencv_imgcodecs330d.lib
opencv_imgproc330d.lib
opencv_imgproc330d.lib
opencv_objdetect330d.lib
opencv_photo330d.lib
opencv_shape330d.lib
opencv_stitching330d.lib
opencv_superres330d.lib
opencv_video330d.lib
opencv_videio330d.lib
opencv_videostab330d.lib

```

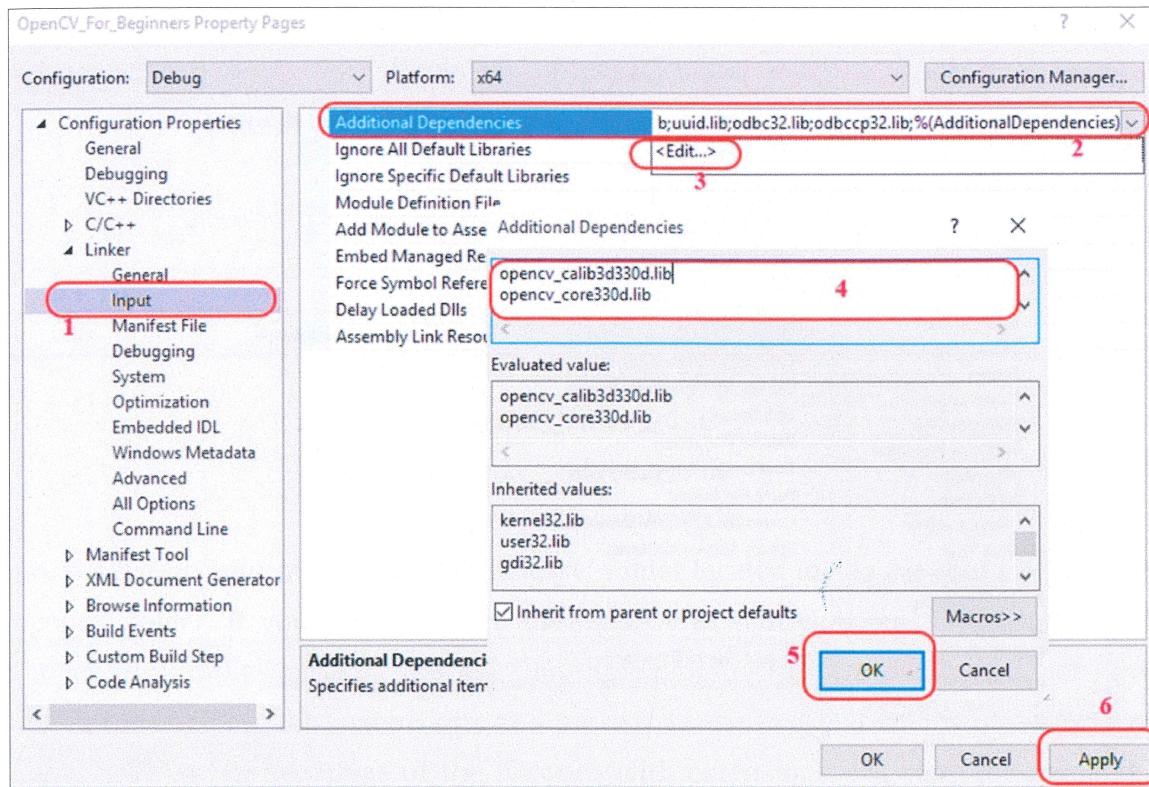
To prepare the list on your own, go to the ‘Debug’ folder located inside the ‘lib’ folder inside ‘build’ folder of ‘opencv’ folder. If you have installed the OpenCV in C drive, then the path of the ‘Debug’ folder is ‘C:\opencv\build\lib\Debug’. You will find three types of files (.exp, .lib, and .pdb) there. Copy the name of the files having ‘.lib’ extensions and paste it in the text box (marked as 4 in figure 1.16). Remember, you have to put the name of the libraries with extensions. So, copy the name of the files’ with extensions too. Figure 1.15 shows an example of copying a name of one of the debug libraries.

This PC > Local Disk (C:) > opencv > build > lib > Debug				
Name	Date modified	Type	Size	
opencv_calib3d330d.exp	11/1/2017 10:14 PM	Exports Library File	187 KB	
opencv_calib3d330d.lib	11/1/2017 10:14 PM	Object File Library	316 KB	
opencv_calib3d330d.pdb	11/1/2017 10:14 PM	Program Debug D...	2,532 KB	
opencv_core330d.exp	11/1/2017 10:11 PM	Exports Library File	453 KB	
opencv_core330d.lib	11/1/2017 10:11 PM	Object File Library	753 KB	

Figure 1.15: Making the List of Debug Libraries

5. Now press ‘OK’ button
6. Then press ‘Apply’ button. But do not close the window.

The following figure will help you to understand if you have any confusion about the steps mentioned above.



**Figure 1.16:** Debug Library List Adding

Now we have to do the same thing for release. To do it, follow the steps (figure 1.17):

1. Select 'Release' from 'Configuration',
2. Then click on the dropdown arrow of 'Linker',
3. Click on 'General',
4. Now on the right side in 'Additional Library Dependencies' field add the path of 'Release' folder and hit 'Enter' from Key board. Then click 'Apply' button. But do not close the window (Figure: 1.17).

The 'Release' folder is located inside 'lib' folder, located inside 'build' folder of OpenCV. If you have extracted the OpenCV in C drive he path of the 'Release' folder is 'C:\opencv\build\lib\Release'.

If there is any confusion, take visual support from Figure 1.17:

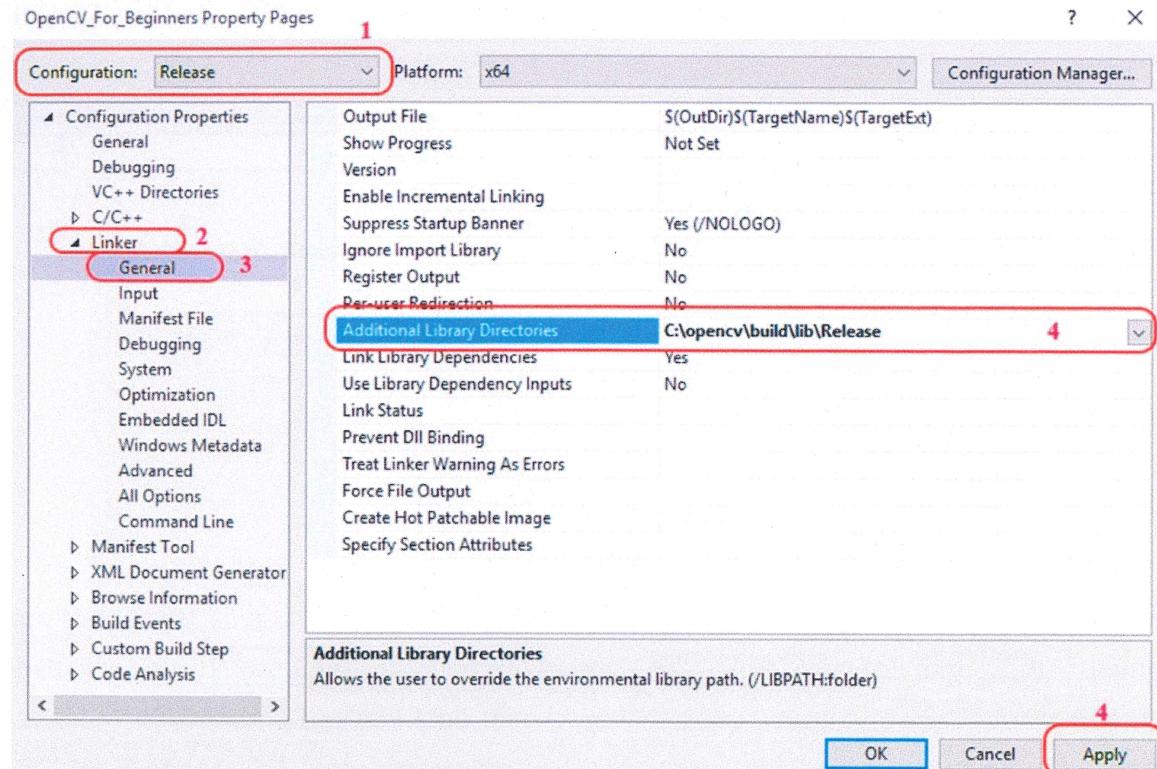


Figure 1.17: Additional Library for Release

Now on the same window (figure 1.18):

1. Click on the 'Input'
2. Then locate 'Additional Dependencies' on the right side and click on dropdown icon
3. Click on the '<Edit...>' option. A new window will open.
4. In this window, there is a text box. In the text box you have to write down the name of release libraries with extension. If you are using OpenCV3.3.0, then the name of the debug libraries will be 'opencv\_name of the library330.lib'. If you use OpenCV 2.4.7, the name of the debug libraries will be 'opencv\_name of the library247.lib'. The format of debug library is – 'opencv\_name of the libraryx<sub>1</sub>x<sub>2</sub>x<sub>3</sub>.lib' where x<sub>1</sub>x<sub>2</sub>x<sub>3</sub> represents the version of OpenCV. The list of the release libraries of OpenCV3.3.0 has been listed below–

```

opencv_calib3d330.lib
opencv_core330.lib
opencv_dnn330.lib
opencv_features2d330.lib
opencv_flann330.lib
opencv_highgui330.lib

```

```

opencv_imgcodecs330.lib
opencv_imgproc330.lib
opencv_ml330.lib
opencv_objdetect330.lib
opencv_photo330.lib
opencv_shape330.lib
opencv_stitching330.lib
opencv_superres330.lib
opencv_video330.lib
opencv_videoio330.lib
opencv_videostab330.lib

```

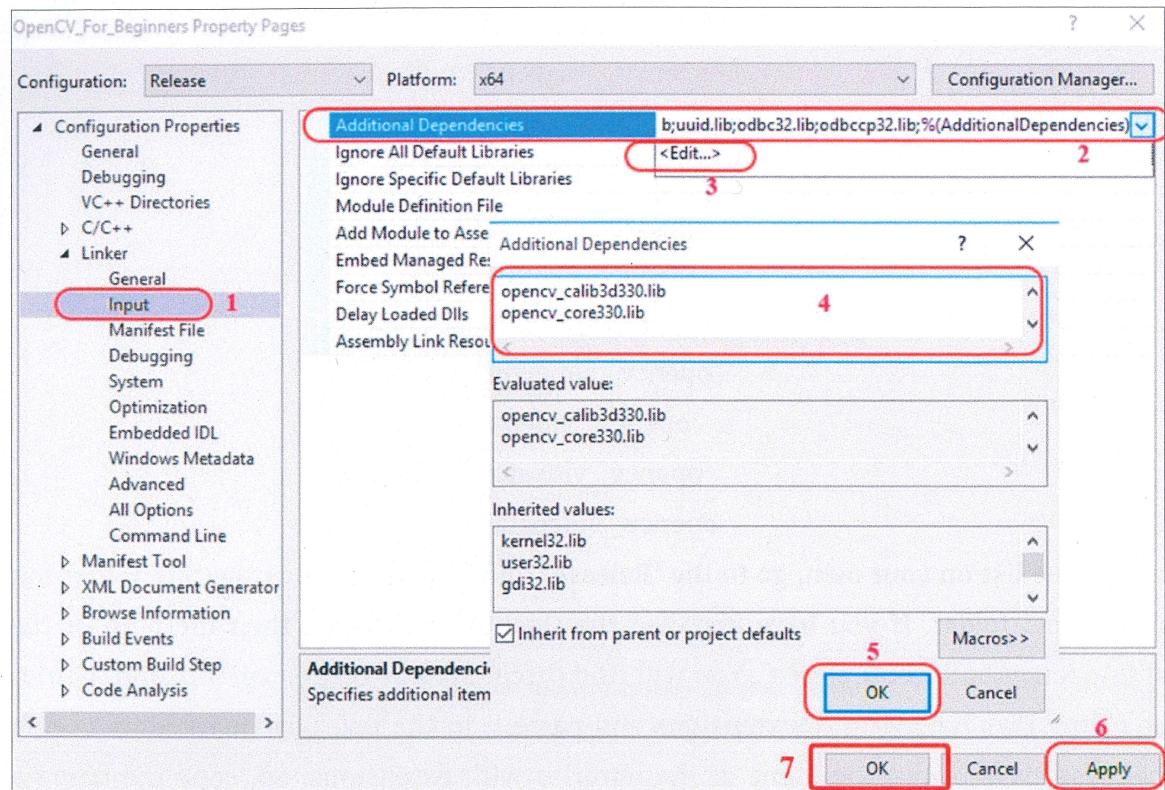
To prepare the list on your own, go to the ‘Release’ folder located inside the ‘lib’ folder inside ‘build’ folder of ‘opencv’ folder. If you have installed the OpenCV in C drive, then the path of the ‘Release’ folder is ‘C:\opencv\build\lib\Release’. You will find three types of files (.exp, .lib, and .pdb) there. Copy the name of the files having ‘.lib’ extensions and paste it in the text box (marked as 4 in figure 1.18). Remember, you have to put the name of the libraries with extensions. So, copy the name of the files with extensions. Figure 1.17 shows an example of copying a name of one of the release libraries.

This PC > Local Disk (C:) > opencv > build > lib > Release			
Name	Date modified	Type	Size
opencv_calib3d330.exp	11/1/2017 10:23 PM	Exports Library File	179 KB
<b>opencv_calib3d330.lib</b>	<b>11/1/2017 10:23 PM</b>	<b>Object File Library</b>	<b>302 KB</b>
opencv_calib3d330.pdb	11/1/2017 10:23 PM	Program Debug D...	2,252 KB
opencv_core330.exp	11/1/2017 10:19 PM	Exports Library File	437 KB
opencv_core330.lib	11/1/2017 10:19 PM	Object File Library	723 KB

**Figure 1.17:** Release Library List Making

5. Now press ‘OK’ button,
6. Then press ‘Apply’ button.
7. Press ‘OK’ and the Properties window will be closed.

If there is any problem in understanding these steps, please have a look at figure 1.18.



**Figure 1.18: Release Library Adding**

This is the end of 3<sup>rd</sup> step. We are ready to use OpenCV now. Every time you create a new project, you have to link OpenCV with Visual Studio again. To do it, repeat the step 3 only.

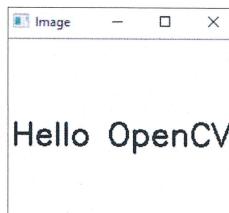
Now let's run an example. At this time, simply copy the following program word for word. You do not need to understand anything. This program is to check if everything is working properly or not.

```
#include<iostream>
#include<opencv2/highgui/highgui.hpp>
#include<opencv2/imgproc/imgproc.hpp>

using namespace std;
using namespace cv;

int main()
{
    Mat image = Mat(200, 220, CV_8UC3, Scalar(255, 255, 255));
    Point text_position(5, 100);
    int font_size = 1;
    Scalar font_color(0, 0, 0);
    int font_weight = 2;
    putText(image, "Hello OpenCV", text_position, FONT_HERSHEY_SIMPLEX, font_size, font_color,
            font_weight);
    imshow("Image", image);
    waitKey(0);
    return(0);
}
```

If you have not made any mistake during the OpenCV installation process (in all three of the steps), then the program will generate the following output –



**Figure 1.19:** Testing the Installing Process

If you get any error message, first of all check if you have written the code properly. Then review the installation process again. This installation procedure of this book has been tested with different versions of Visual studio in windows 7 (both 32 and 64 bit), Windows 8 (both 32 and 64 bit) and Windows 10 (both 32 and 64 bit). The OpenCV ran smoothly in all of the tests. It is normal for the beginners to make mistakes at the beginning. So, if you get any error message while trying to run OpenCV, carefully review the installation steps and try to figure out where you have made the mistake.

# End of Chapter 1