Why do we need image processing

- Improve the visual effect of people by improving the quality of the image.
- Machine/Robot vision
- Medical field
- Efficient storage & transmission.

So digital image processing is:

- Set of algorithms applied to digital images in order to improve quality, storage, transmission and representation.

What's an Image

- Image is a 2D projection of 3D object.
- So image can be defined as 2D function F(x,y)
  Where:
  - x,y represent spatial coordinates
  - The amplitude of F at any point is called the intensity at that point.

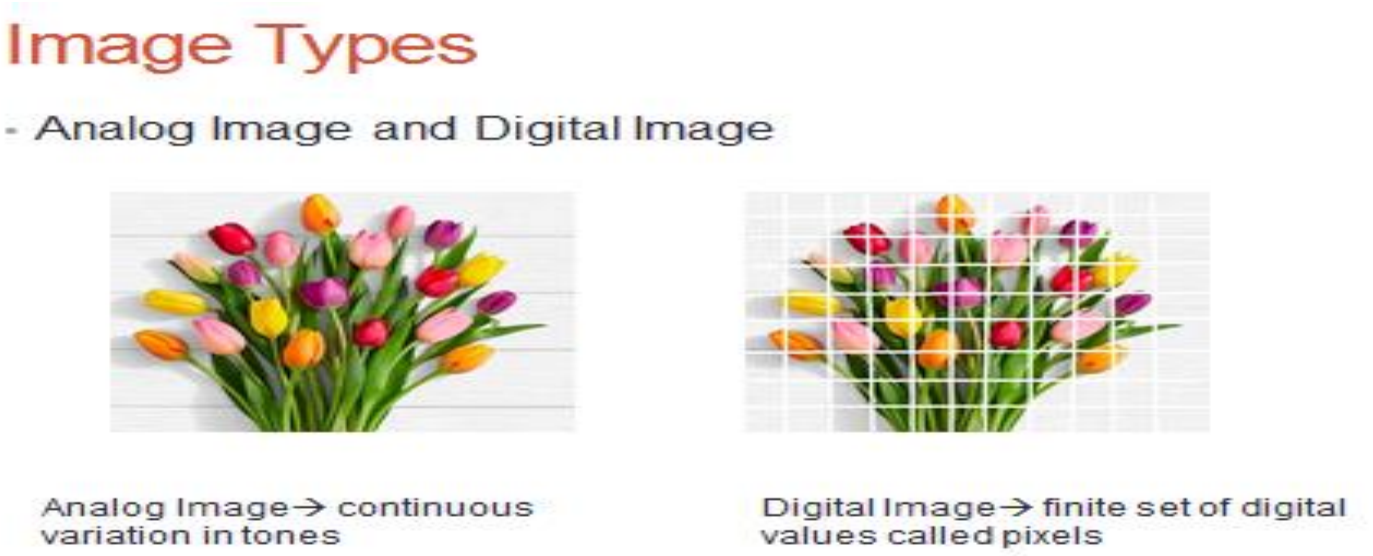


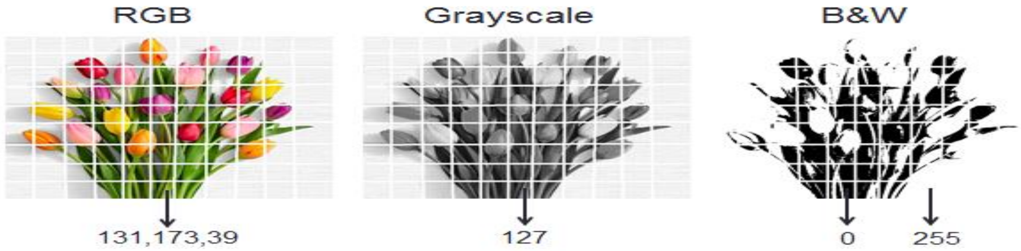
Image Types

- Analog Image and Digital Image

Analog Image→ continuous variation in tones

Digital Image→ finite set of digital values called pixels

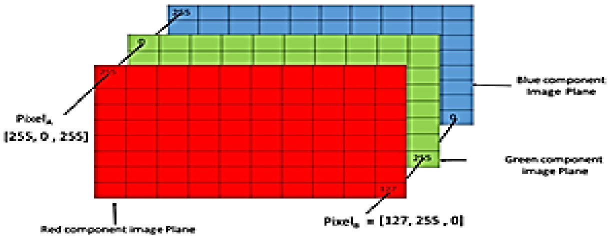
Digital Image color models



Digital image is composed of a finite number of pixels **"picture elements"**, each of which has a particular location and value

Digital Image color models

RGB


255
0
255
Pixel$_A$
[255, 0 , 255]
Blue component image Plane
0
255
Green component image Plane
127
Red component image Plane
Pixel$_B$ = [127, 255 , 0]

Grayscale

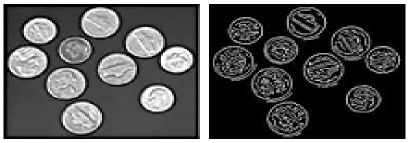
$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

In our work we mostly work on grayscale system as RGB image has 3D while grayscale has just one, so, everything tend to be more costly in RGB system

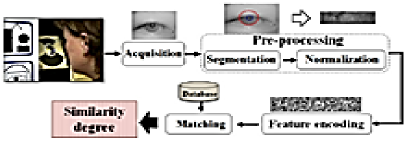
What type of processing can be done on digital image?

- Low level processing
  - Input image
  - Output image
    - Example: noise removal

- Mid level processing
  - Input image
  - Output Attributes(edges, contours)
    - Example: object segmentation



- High level processing
  - Input attributes
  - Output understanding
    - Example: scene understanding



Needed environment

- Hardware:
  - CPU (Intel i5/ i7/ Xeon recommended).
  - 8 GB RAM, 10 GB HDD Free Space.
- Software
  - Windows 8, 10, 64 bits
  - Opencv → latest release
  - Visual studio → latest release

OpenCv

- OpenCv stands for open source library for image processing and computer vision.
- OpenCv supports Windows, Linux, Android and Mac OS.
- OpenCv supports a wide variety of programming languages such as C++, Python, Java, etc..
- Primary interface of OpenCV is in C++
- Opencv 4.3.0 Download link→

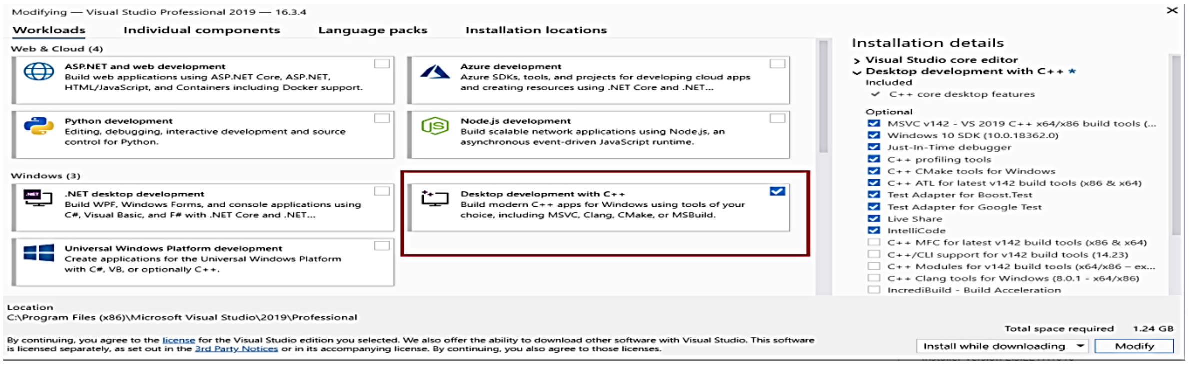
https://sourceforge.net/projects/opencvlibrary/files/4.3.0/opencv-4.3.0-vc14_vc15.exe/download

visual studio

visual studio 2019 community free download link→

https://visualstudio.microsoft.com/downloads/

- There are different kinds of component. What we will need is desktop development with c++ check it and goes with the install process



Modifying — Visual Studio Professional 2019 — 16.3.4

Web & Cloud (4)

□ **ASP.NET and web development**
Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker support.

□ **Azure development**
Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET Core and .NET...

□ **Python development**
Editing, debugging, interactive development and source control for Python.

□ **Node.js development**
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Windows (3)

□ **.NET desktop development**
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET Core and .NET...

☑ **Desktop development with C++**
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.

□ **Universal Windows Platform development**
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

Installation details

> Visual Studio core editor
⌄ Desktop development with C++ *

Included

✓ C++ core desktop features

Optional

☑ MSVC v142 - VS 2019 C++ x64/x86 build tools (...
☑ Windows 10 SDK (10.0.18362.0)
☑ Just-In-Time debugger
☑ C++ profiling tools
☑ C++ CMake tools for Windows
☑ C++ ATL for latest v142 build tools (x86 & x64)
☑ Test Adapter for Boost.Test
☑ Test Adapter for Google Test
☑ Live Share
☑ IntelliCode
□ C++ MFC for latest v142 build tools (x86 & x64)
□ C++/CLI support for v142 build tools (14.23)
□ C++ Modules for v142 build tools (x64/x86 – ex...
□ C++ Clang tools for Windows (8.0.1 - x64/x86)
□ IncrediBuild - Build Acceleration

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Professional

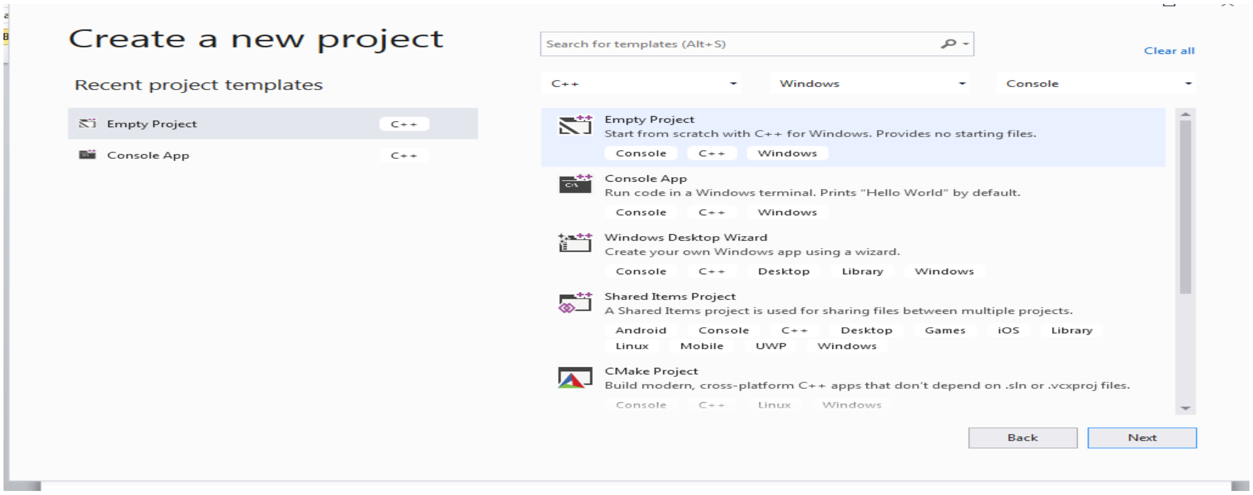
By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 1.24 GB

Add opencv dll files path

1. environment variables →system variables →double click on path→ write the path of the bin folder on your device. for example “E:\opencv\build\x64\vc15\bin”

2. visual studio →create new project



Create a new project

Recent project templates

Empty Project
Start from scratch with C++ for Windows. Provides no starting files.
Console C++ Windows

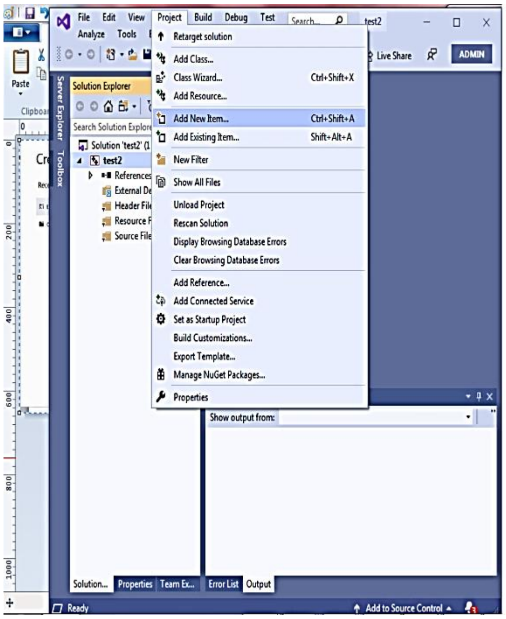
Console App
Run code in a Windows terminal. Prints "Hello World" by default.
Console C++ Windows

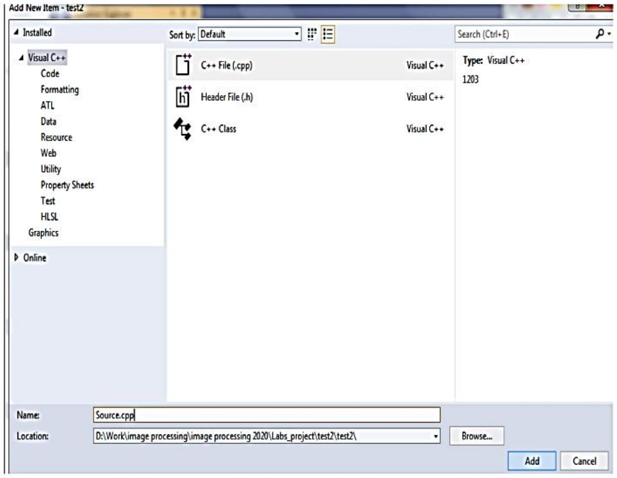
Windows Desktop Wizard
Create your own Windows app using a wizard.
Console C++ Desktop Library Windows

Shared Items Project
A Shared Items project is used for sharing files between multiple projects.
Android Console C++ Desktop Games iOS Library Linux Mobile UWP Windows

CMake Project
Build modern, cross-platform C++ apps that don't depend on .sln or .vcxproj files.
Console C++ Linux Windows

3. Add C++ file





Add New Item - test2

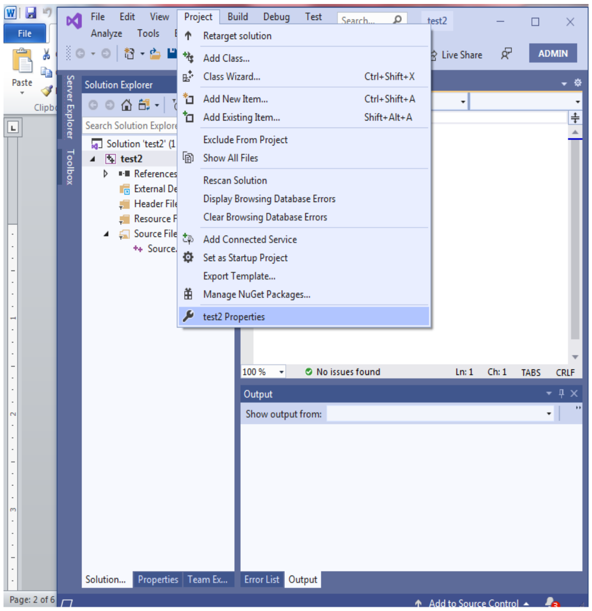
Type: Visual C++
1203

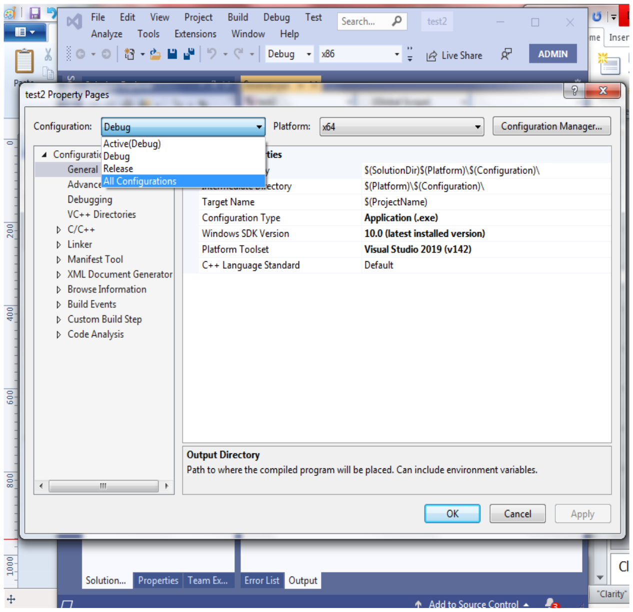
Name: Source.cpp

Location: D:\Work\image processing\image processing 2020\Labs_project\test2\test2\

4. Add opencv files to your project

- Project→ properties
- **Configuration→ all configuration and platform→x64**



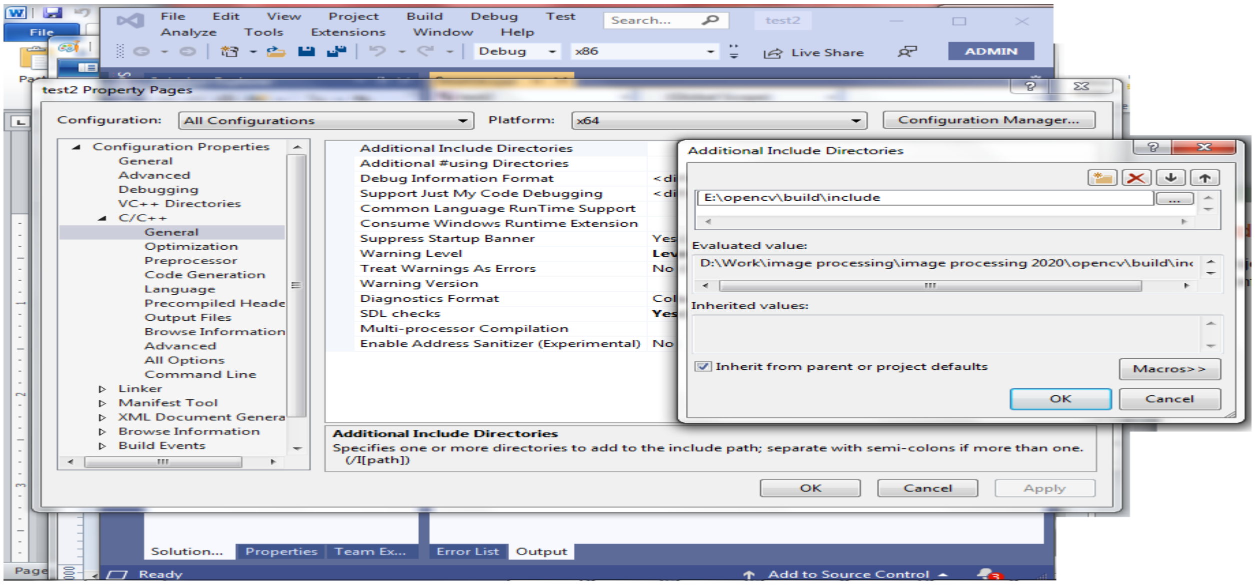

test2 Property Pages

Configuration: Debug | Platform: x64 | Configuration Manager...

Output Directory	$(SolutionDir)$(Platform)\$(Configuration)\
Intermediate Directory	$(Platform)\$(Configuration)\
Target Name	$(ProjectName)
Configuration Type	Application (.exe)
Windows SDK Version	10.0 (latest installed version)
Platform Toolset	Visual Studio 2019 (v142)
C++ Language Standard	Default

Output Directory
Path to where the compiled program will be placed. Can include environment variables.

- C/C++→General
  - Additional include directory→write the path of the include folder on your device. For example"**E:\opencv\build\include**"


test2 Property Pages

Configuration: All Configurations | Platform: x64 | Configuration Manager...

Additional Include Directories
Specifies one or more directories to add to the include path; separate with semi-colons if more than one. (/I[path])

Additional Include Directories

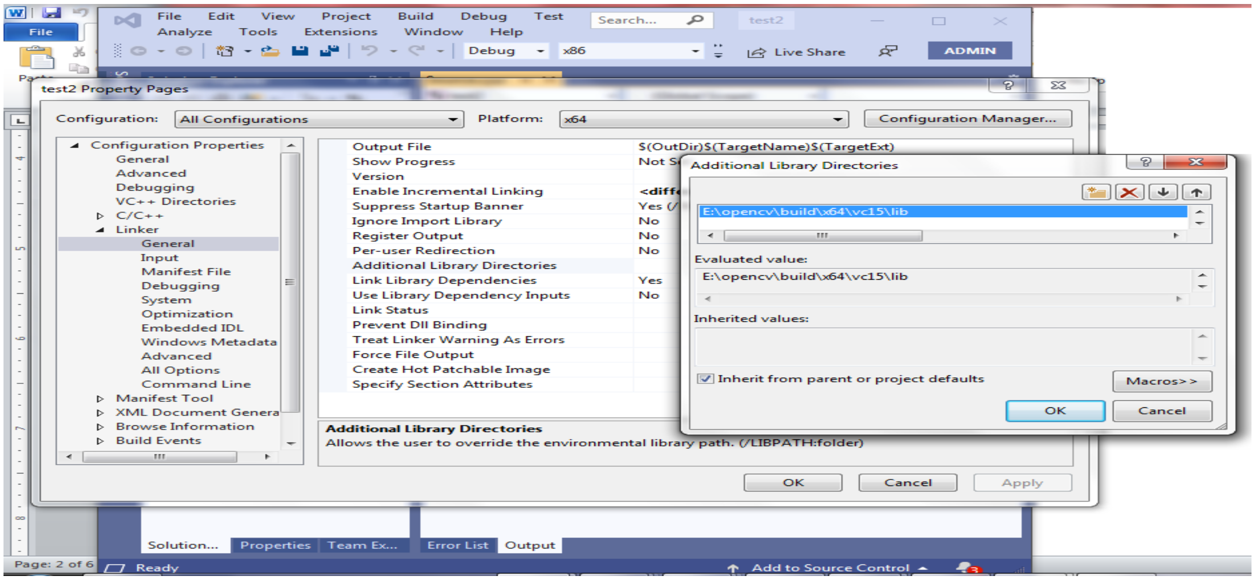
E:\opencv\build\include

Evaluated value:
D:\Work\image processing\image processing 2020\opencv\build\inc

Inherited values:

☑ Inherit from parent or project defaults

- Linker→ General
    - Additional library directories→ write the path of the lib folder on your device. for example “**E:\opencv\build\x64\vc15\lib**”



test2 Property Pages

Configuration: All Configurations — Platform: x64 — Configuration Manager...

Output File	$(OutDir)$(TargetName)$(TargetExt)
Show Progress	Not S
Version	
Enable Incremental Linking	<diff
Suppress Startup Banner	Yes (/
Ignore Import Library	No
Register Output	No
Per-user Redirection	No
Additional Library Directories	
Link Library Dependencies	Yes
Use Library Dependency Inputs	No
Link Status	
Prevent Dll Binding	
Treat Linker Warning As Errors	
Force File Output	
Create Hot Patchable Image	
Specify Section Attributes	

Additional Library Directories
Allows the user to override the environmental library path. (/LIBPATH:folder)

Additional Library Directories

E:\opencv\build\x64\vc15\lib

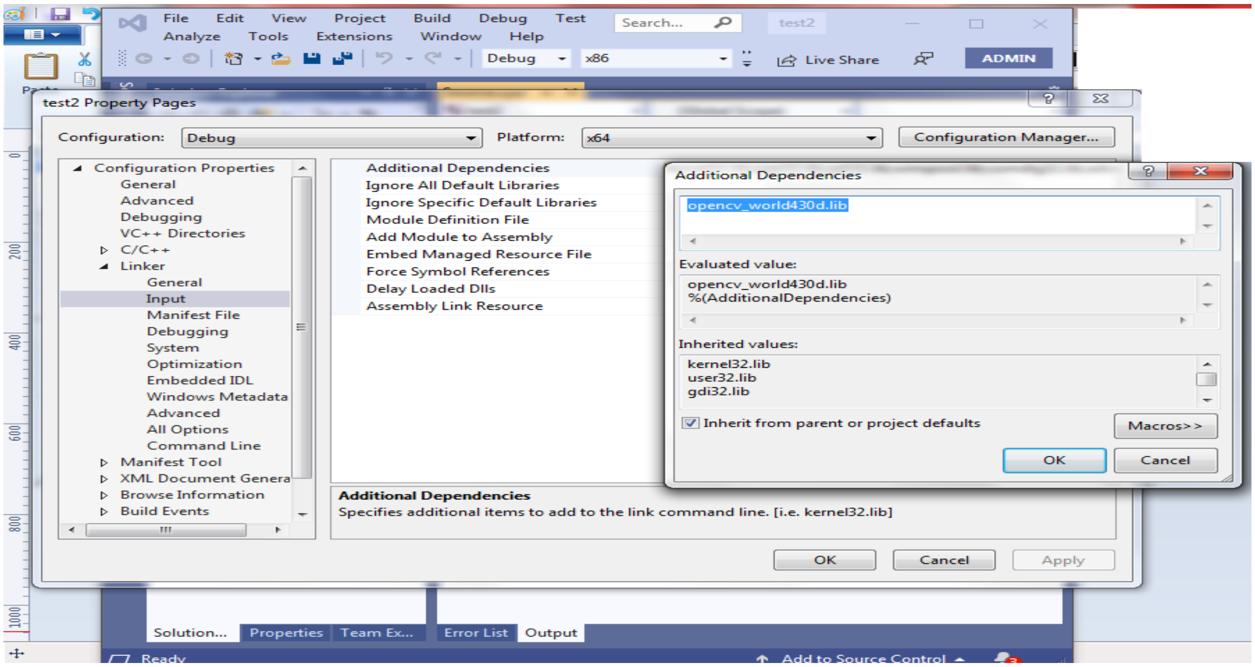
Evaluated value:

E:\opencv\build\x64\vc15\lib

Inherited values:

☑ Inherit from parent or project defaults

- **Configuration→ Debug and platform→x64**
    - Linker→input
        - Additional dependencies→opencv_world430d.lib



test2 Property Pages

Configuration: Debug — Platform: x64 — Configuration Manager...

Additional Dependencies
Ignore All Default Libraries
Ignore Specific Default Libraries
Module Definition File
Add Module to Assembly
Embed Managed Resource File
Force Symbol References
Delay Loaded Dlls
Assembly Link Resource

Additional Dependencies
Specifies additional items to add to the link command line. [i.e. kernel32.lib]

Additional Dependencies

opencv_world430d.lib

Evaluated value:

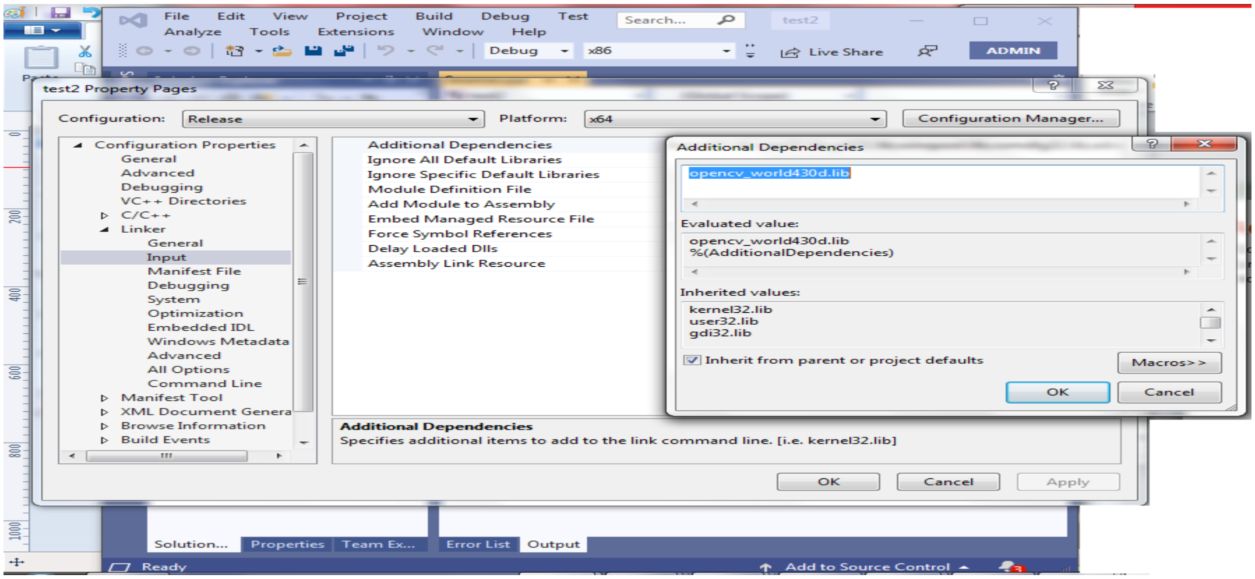
opencv_world430d.lib
%(AdditionalDependencies)

Inherited values:

kernel32.lib
user32.lib
gdi32.lib

☑ Inherit from parent or project defaults

- **Configuration→   Release and platform→x64**
    - Linker→   input
        - Additional dependencies→opencv_world430.lib



test2 Property Pages

Configuration: Release   Platform: x64

Additional Dependencies

opencv_world430d.lib

Evaluated value:

opencv_world430d.lib
%(AdditionalDependencies)

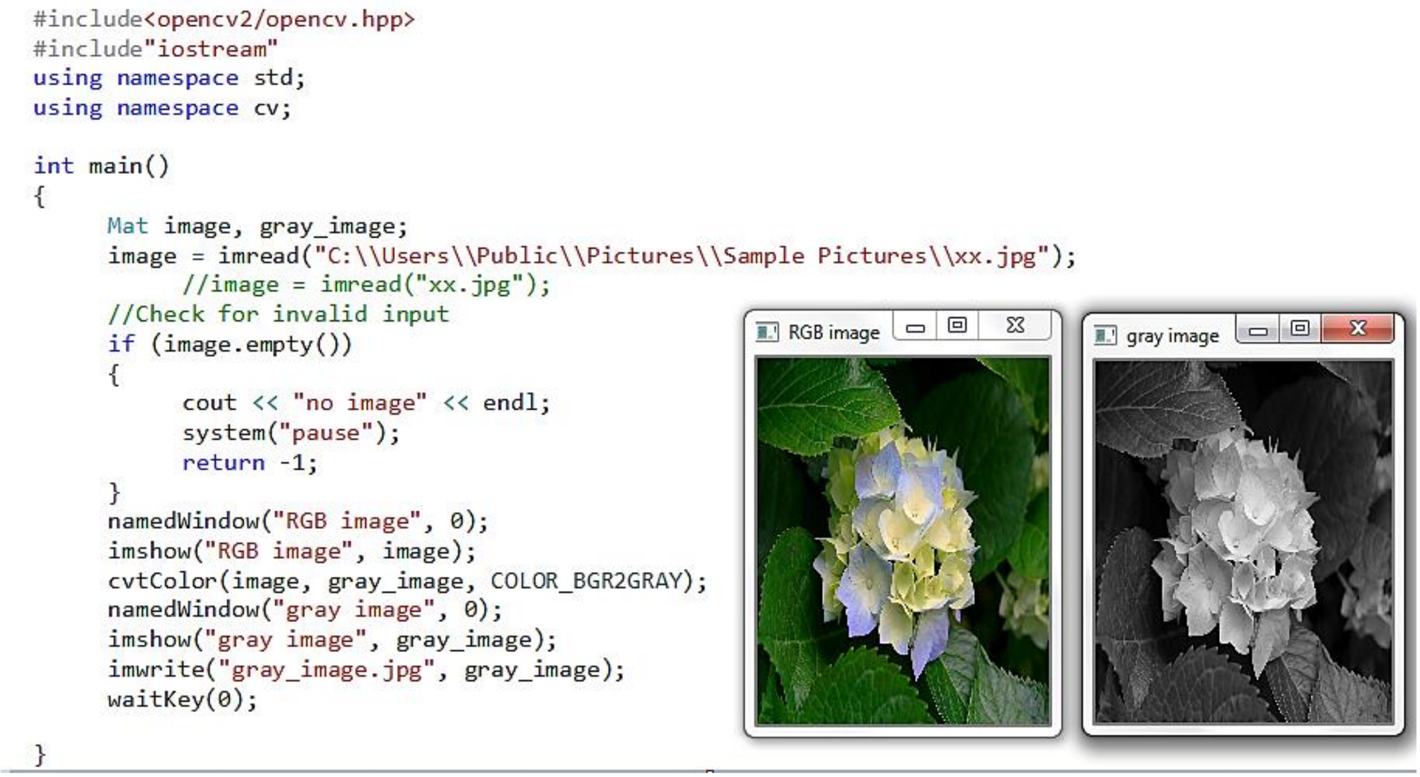
Inherited values:

kernel32.lib
user32.lib
gdi32.lib

☑ Inherit from parent or project defaults

Additional Dependencies
Specifies additional items to add to the link command line. [i.e. kernel32.lib]

Display image



```
#include<opencv2/opencv.hpp>
#include"iostream"
using namespace std;
using namespace cv;

int main()
{
	Mat image, gray_image;
	image = imread("C:\\Users\\Public\\Pictures\\Sample Pictures\\xx.jpg");
		//image = imread("xx.jpg");
	//Check for invalid input
	if (image.empty())
	{
		cout << "no image" << endl;
		system("pause");
		return -1;
	}
	namedWindow("RGB image", 0);
	imshow("RGB image", image);
	cvtColor(image, gray_image, COLOR_BGR2GRAY);
	namedWindow("gray image", 0);
	imshow("gray image", gray_image);
	imwrite("gray_image.jpg", gray_image);
	waitKey(0);

}
```