Rotating an Image

Aim:

To Rotate the Image at any given Angle

Input:

Image and Angle to Rotate

Output:

Rotated Image

Edge Cases:

1. If the User Does not give an Image,

   Program will exit with appropriate Message.

Technologies Used:

Open CV, Visual Studio 2019, C++

How to run the Program:

Installation of Required Software's:

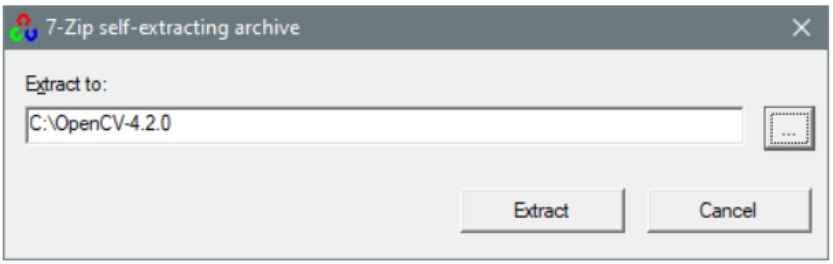
You can Refer to my YouTube Channel: https://youtu.be/riArxlmRMoQ

Step 1: Download and Install Open CV from official Website

Or Git Repository: https://github.com/opencv/opencv/releases/tag/4.1.2

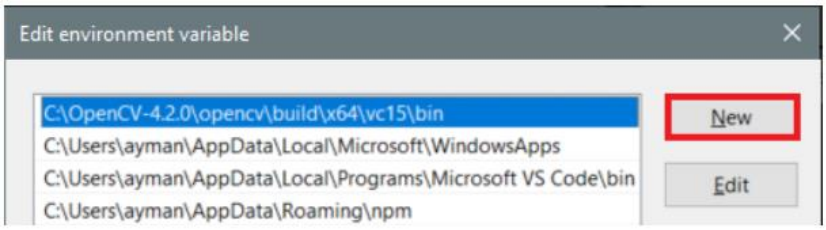
Download the .exe file

Extract it



Step 2: Add Open CV libraries to your System Path

Once OpenCV is correctly installed in your folder, you now have to add the binaries C:\OpenCV-4.2.0\opencv\build\x64\vc15\bin to your system path, so you can have access to OpenCV executables easily through your command line.



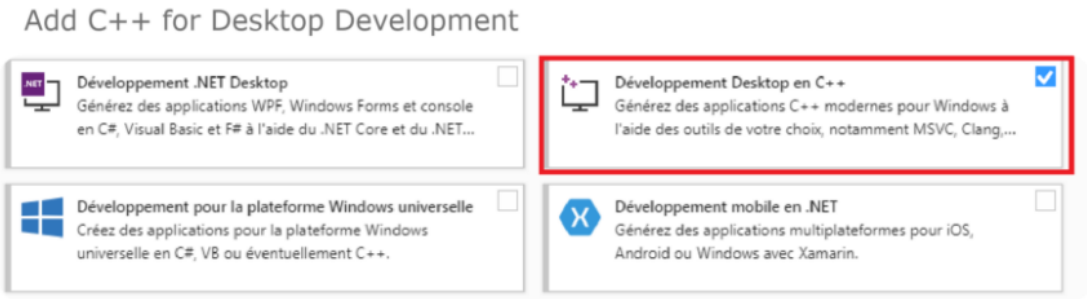
Editing the system path

Step 3: Downloading and Installing Visual Studio 2019

Download Community Version from official Website

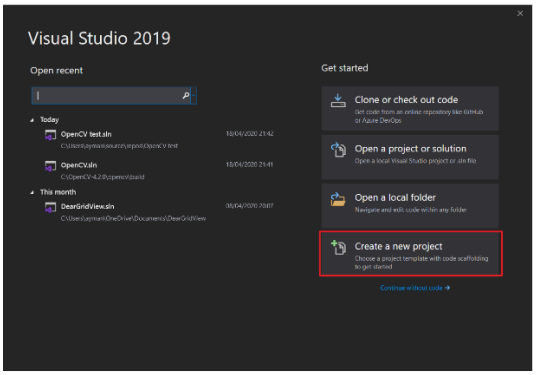
Link: https://visualstudio.microsoft.com/

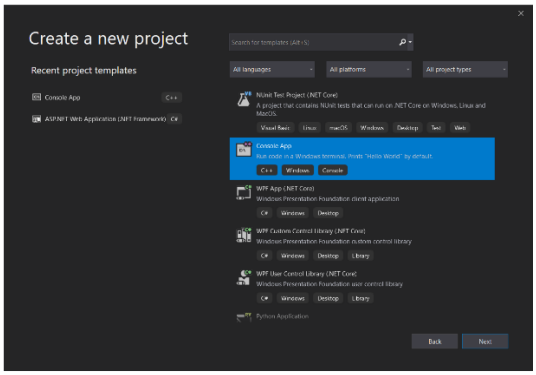
Check the box to install “Development Desktop on C++”



Step 4: Configure a Visual Studio project to run OpenCV

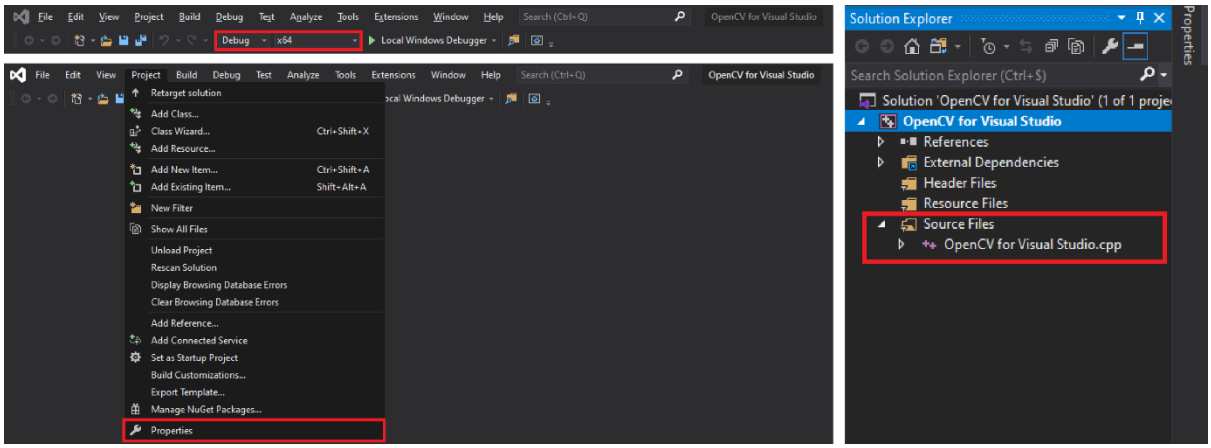
Open Visual Studio 2019, choose to create a new project and go for the C++ Console App template.





Once the project created you should have a layout with a solution explorer to the right having one source file and, in the window, upper ribbon you should see Debug for x86 platforms, meaning that the project will build and run-in debug mode targeting x86 windows architectures.

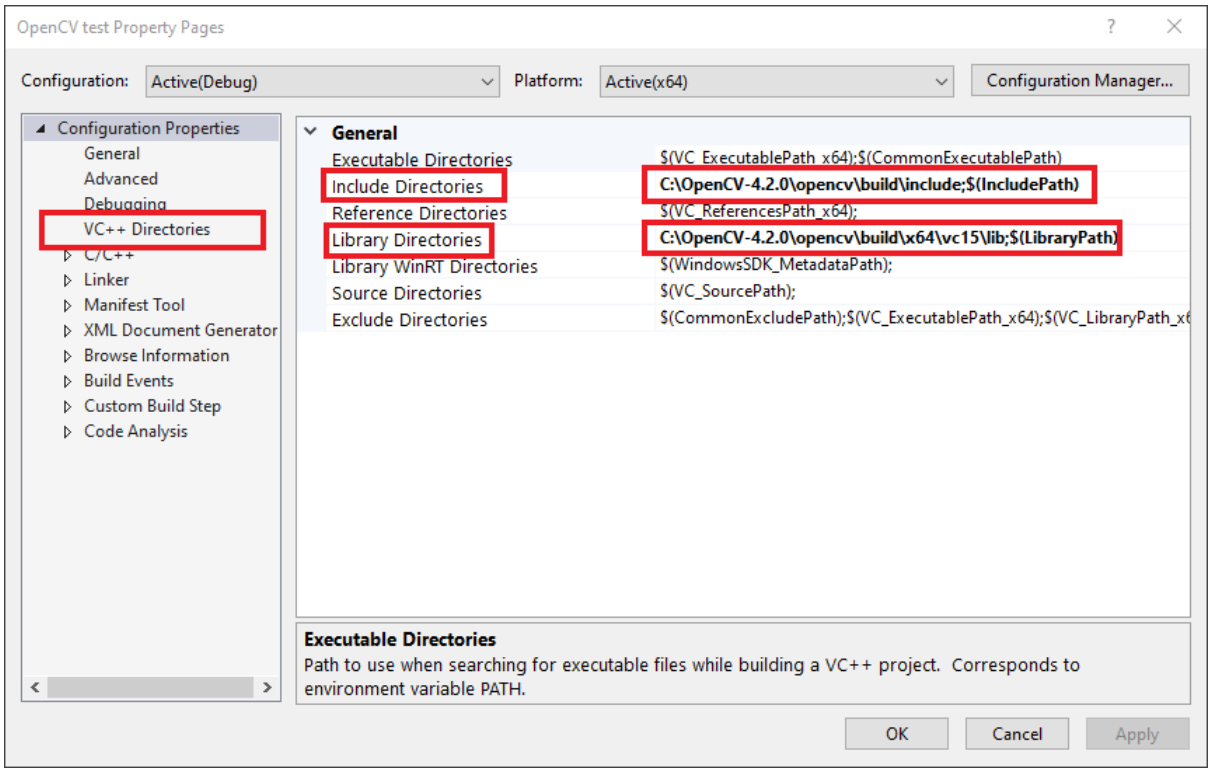
- First, you should change the solution platforms from x86 to x64
- Secondly, you have to change the **Project Properties** to add the OpenCV libraries



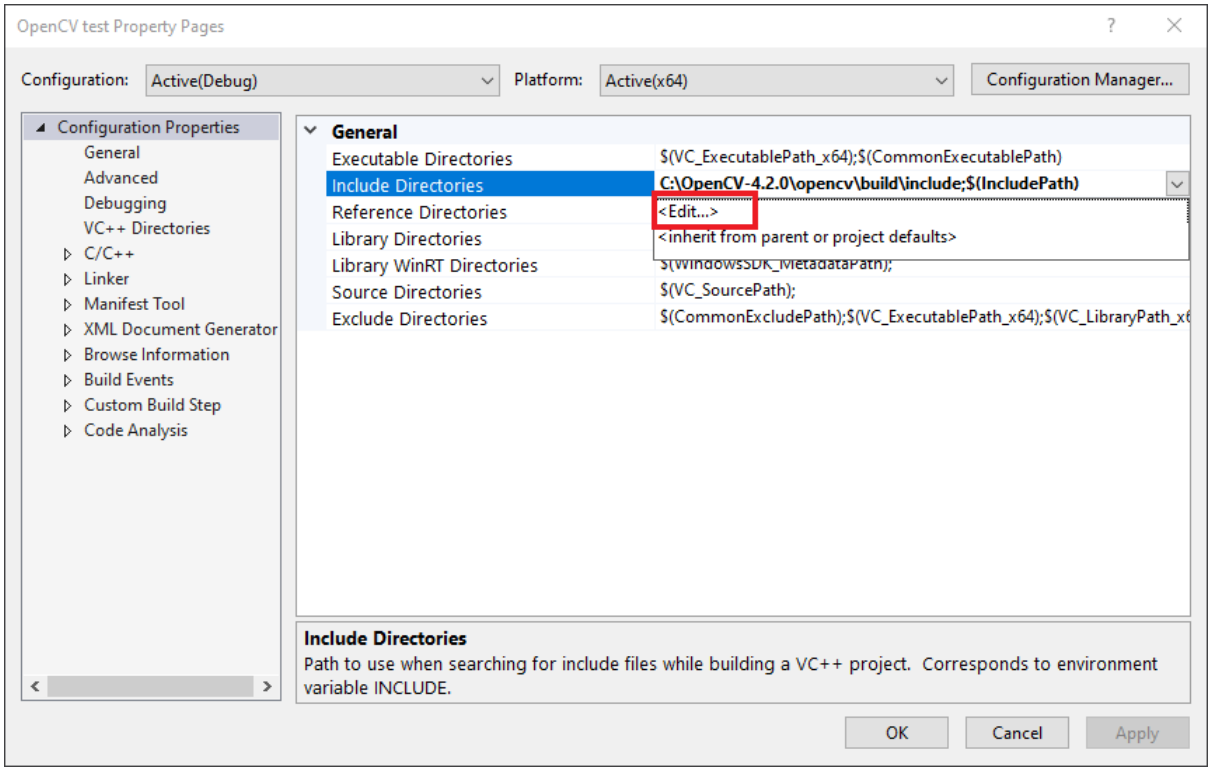
Change the target solution platform, and head to project properties

There are a bunch of properties to edit before being able to execute any code:

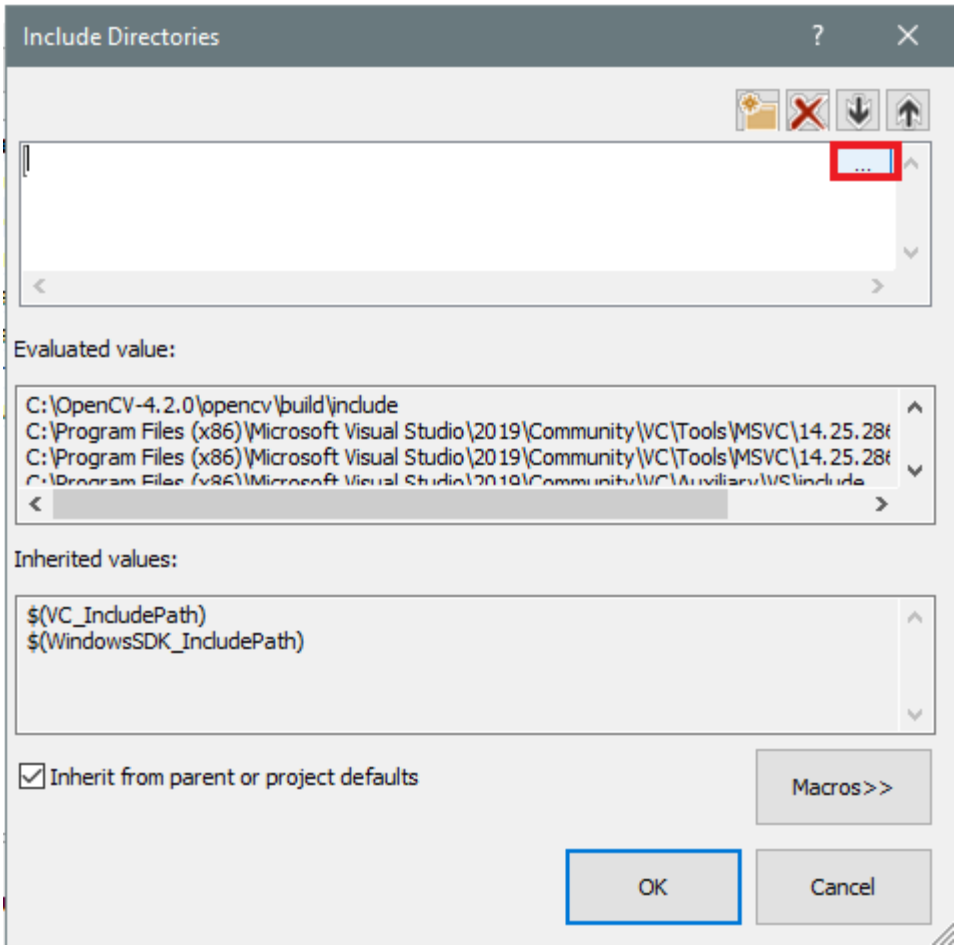
1. Go to **Configuration Properties/VC++ Directories** to add the include and library directories for OpenCV.



Edit both Include and Library Directories

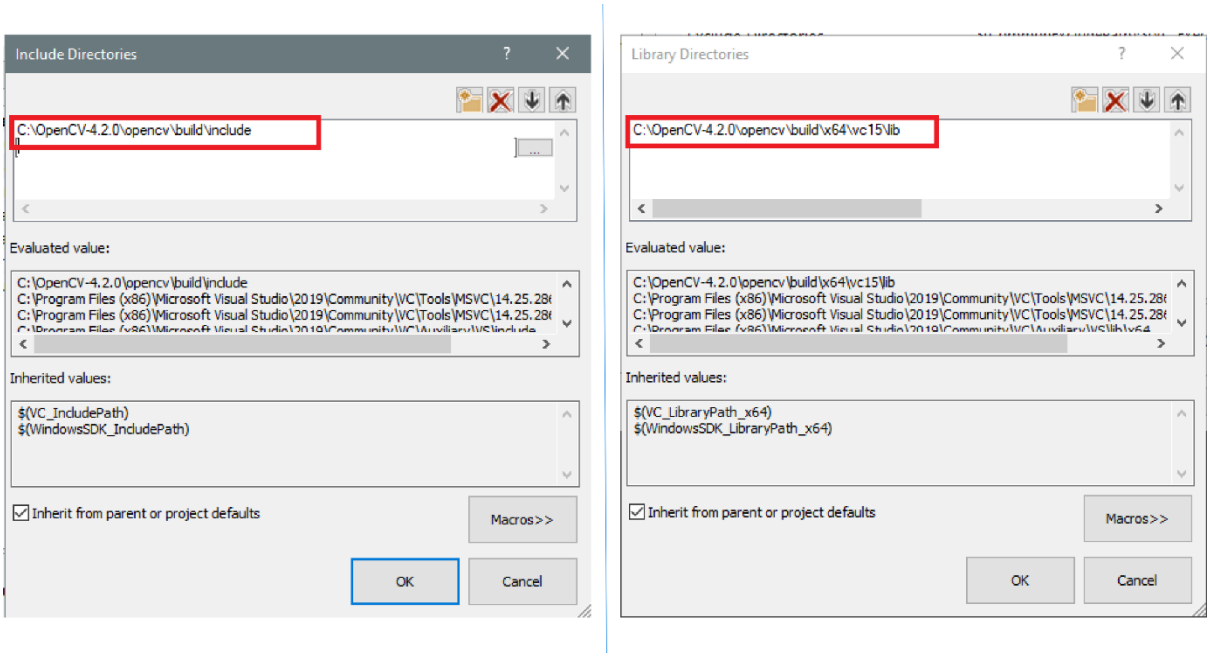


Click on the right arrow next to blue entry



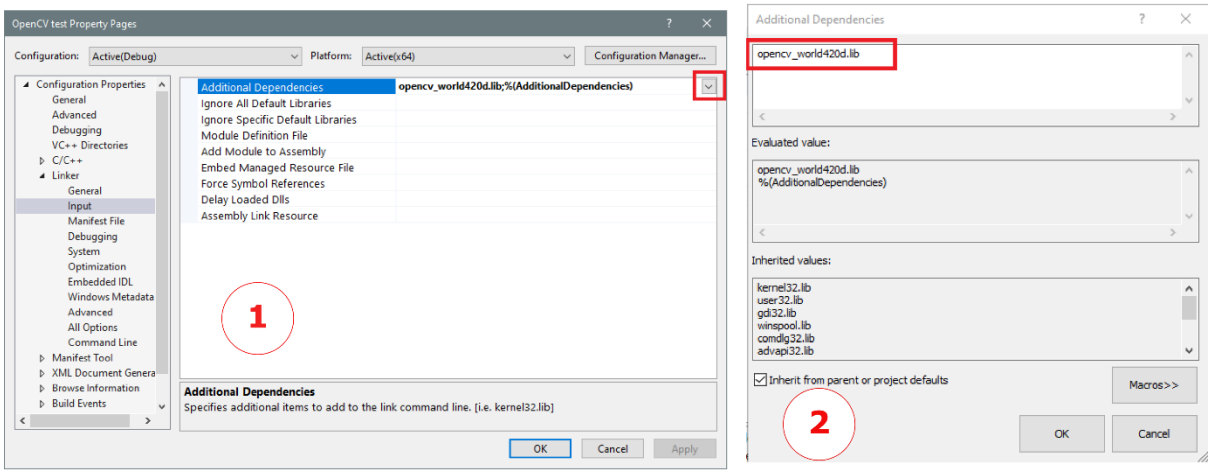
Include the internal system path

For the Include directory, you have to add the following path: C:\OpenCV-4.2.0\opencv\build\include. Do the same for the **Library Directories** adding this internal path: C:\OpenCV-4.2.0\opencv\build\x64\vc15\lib.



Include and Lib directories both added to the project

2. Edit the VC++ project linker with the opencv_world420d.lib OpenCV dynamic library. You will find the **DLL** (**D***ynamic* ***L****ink* ***L****ibrary*) here: C:\OpenCV-4.2.0\opencv\build\x64\vc15\lib copy the name of the file opencv_world420d.lib and paste it in the dependency box.



Click the OK button when finished

Final Step: Test a bit of code

Finally, you can run this demo code in your visual studio IDE to see if it’s all working fine.

The resulting image

And that’s all, you can now use OpenCV with ease. 😉

Working:

1. Reading the Image
   - Read in matrix format with data type Mat.

2. Reading Angle as integer to Rotate the Image.

3. Create AutoSize Window to Display Image and TrackBar.
   - Using namedWindow(string_name, Type);
   -
4. Rotating Image using built-in Function.
   - getRotationMatrix2D(Point2f center, angle, Scale);
   - center => takes value to rotate image from a (center)point.
   - Angle => take Anti Clockwise value to rotate Image
   - Scale => Zoom, 1 is Normal, +1 Values Zoom the Image.

5. Overcoming Problem of Image Cropping while Rotating Image.
   - We will take a Bounding Rectangle and add it to Translated Image
   - To make a bigger box to fit our Image without Cropping in our window.
   - RotatedRect(Point2f(), src.size(), -angle).boundingRect2f();
   - Point2f()
   - This OpenCV function applies affine transformation to an image.

6. void warpAffine(InputArray src, OutputArray dst, InputArray M, Size dsize)
   - src - Source Image
   - dst - Destination image which should have the same type as the source image(The transformed image is stored in this location)
   - M - 2x3 affine transformation matrix
   - dsize - Size of the destination image

7. int createTrackbar(string trackbarname, string winname, int* value, int count, TrackbarCallback onChange = 0, void* userdata = 0).
   - This OpenCV function creates a trackbar and attached that trackbar to a specified window
   - trackbarname - The name of the trackbar
   - winname - The name of the window to which the trackbar is attached
   - value - This integer, pointed by this pointer, holds the value associated with the position of the trackbar
   - count - The maximum value of the trackbar. The minimum value is always zero.
   - onChange - This function will be called everytime the position of the trackbar is changed. The prototype of this function should be "FunctionName(int, void*)". The "int" value is the value associate with the position of the trackbar. And "void*" is any pointer value which you pass as the "userdata" (See the next parameter).
   - userdata - This pointer variable will be passed as the second parameter of the above function

8. Displaying Image
   - imshow(string winName, InputArray() mat).
   - winName is name displayed on window
   - InputArray is image Matrix to be Displayed

9. Run Program and Enter value to Rotate

10. Window will be Displayed with Interactive Track Bar that can be used to rotate the Image.

11. Press “ESC” to exit Program Displaying Image
    - Number 27 key is for “ESC” button in keyboard so we can set it to exit the Program.

12. Press “CTRL” + “S” to save Image on your Storage Device.

Sample Input:

Sample Output:

Thank-you

Project 1: Rotating an Image in Open CV

Course: B.TECH CSE in Machine Learning and AI.

Student ID: 20011001

College: Graphic Era Hill University (Dehradun Campus)

Submitted By: Maskar Vishal