Read me

OpenCV project for image rotation-->

About the project:

This project "image rotation using opencv" is a simple program written in c++ using open cv. The purpose or objective of the project is to rotate any given image of appropriate format to a user defined angle, and thus display it to the output screen.

Technology/platform/software used

For accomplishing its objective the program uses opencv, which is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS.

All the coding is done in a rich and powerful IDE, which goes by the name Visual studio.

Visual studio is developed by Microsoft and apart from being used to develop computer programs it is also used in developing websites, web apps, web services and mobile apps.

Platforms

We have used visual studio as our platform to build and run the basic c++ code, aided with rich libraries of openCV.

Thus the primary platform i.e. visual studio should be installed in the device also the the libraries which allow us to work with images are available in openCV, so both of the software should be downloaded and installed in the system before going further.

Downloading open CV

OpenCV is open source software library so it can be easily downloaded from its official site.

Steps for downloading open CV-->

step1: visit official site and download the file

Step 2: Locate the downloaded file in your system

Step3 open the downloaded file and extract the files to any location.

Step4 after extraction is completed; you will find a folder with name opencv

Basic installation of open CV is completed!!!

Setting up the environment

In order to use the open CV in our visual studio, we need to give the location of the bin files, which are located in our opencv directory.

Step 1: Go to your extracted directory opencv and open it.

Step 2: now open build directory > x64 > vc15 > bin

Step 3: Now copy the path of the bin directory,

Step 4: open control panel

Step 5: locate system setting, and from choose 'Advanced system setting' from the available options in the left

Step 6: 'system properties' dialog box will appear

Step 7: click on 'advanced' tab and then click on 'Environment Variables'

Step 8: 'Environment variables' dialog box will appear

Step 9: go to system variables section > path > press edit button.

Step 10: Now press new from the options available on right and paste the address of the directory copied in step 3.

Step 11: click on apply and ok.

Downloading Visual code

For downloading visual studio

Step 1: go to https://visualstudio.microsoft.com/

Step 2: now look for visual studio and click on the drop down button bellow visual studio.

Step 3: choose 'Community 2019' and an automatic download will start.

Step 4: locate downloaded installer file and run it

Step 5: The installer file will take a while to install and download some files

Step 6: A window will appear asking to choose multiple features or workloads under the workloads tab, click on 'Desktop development with c++'

NOTE: You can also choose other options depending upon your need, but here we are only concerned with running the c++ code. So choosing the above mention workload will be enough.

Step 7: navigate through different tabs to add additional features if you want.

Step 8: look for installation locations tab and choose a location where you want to install the visual code.

Step 9: click on install button and the installation will begin. It will take 1 hour or so depending on your system.

Step 10: Your basic installation is done

How to run the program

Visual studio settings for running the program

Setting visual code for using open CV

A) Including directory

Step 1: Go to extracted directory 'open cv'

Step 2: click on build > include

Step 3: Copy the path of this 'include' directory and now open the visual studio

Step 4: click on project tab and choose properties (last option in the drop down list)

Step 5: From property page and click on vc++ directories

Step 6: in the 'general' section click on 'include directories', a drop down button will appear on right hand side

Step 7: click on the drop down button and click on edit

Step 8: 'Include Directories' box will appear

Step 9: paste the path copied in the step 3 and click on ok

B) Library directory

Step 1: go to extracted directory 'open cv'

Step 2: click on build > x64 > vc15 > lib

Step 3: Copy this directory path

Step 4: Now again open visual code and open your project

Step 5: go to 'Setting’ tab and again choose property

Step 6: From the property page and click on vc++ directories > general section

Step 7: click on 'Library Directories' and choose 'edit' from the drop down menu

Step 8: Now paste the path address you copied in the step 3

C) Debugger

Step 1: go to extracted directory 'open cv'

Step 2: click on build > x64 > vc15 > lib

Step 3: look for debugger file and copy its name.

Step 4: open visual code and open your project

Step 5: click on setting and click on property

Step 6: In property page click on

>'Linker' drop down button > choose 'Input'

Step 7: Now click on 'Additional Dependencies' and on the right choose 'Edit ' from the drop down button

Step 8: paste the debugger file name you copied in the step 3.

Step 9: click ok and you are done with all settings

Running the program

To get the desired output the program must be compiled, in order to create an application file.

If application file or .exe file is already available then program can be accessed directly by opening this file.

The program on compilation or execution will load the inputted image and will ask the user to enter some angle to rotate the image.

If the image location or image will be valid then the program will run and will rotate the image as desired.

Different function used

The objective of the project, i.e. to rotate the image is mainly achieved by inbuilt functions like getRotationMatrix2D( ). We made only one function from our side i.e. rotate_it

Main()

In the main function we basically create Mat object , read image and input the angle user desire to rotate the image at. The image along with the angle is passed to the rotate_it( ) function and finally after the rotation both the original and the rotated image is displayed.

rotate_it( )

This function first figure out the centre of image , and then pass that point in the getRotationMatrix2D( ) function, which Calculates an affine matrix of 2D rotation.

At the end warpAffine( ) function is used which apply affline transformation on the image.

Then the a Mat object I back to calling function