Project on

Image Rotation

Image rotation is an image processing practice where an image to rotate and the rotation angle θ (point about which rotation is done) are input and output is a rotated image at the desired angle.

Prerequisites

- Visual Studio 2019(latest version)/Visual Studio Code
- Open CV

Installation

Visual Studio 2019

1. Install Visual Studio 2019
Check the system requirements to know whether your computer supports Visual Studio 2019.

2. Download Visual Studio bootstrapper file
Visual Studio 2019 can be downloaded through the following link: https://visualstudio.microsoft.com/downloads/

3.Install the Visual Studio installer
From your Downloads folder, double-click the bootstrapper that matches or is similar to one of the following files:

->vs_community.exe for Visual Studio Community (recommended)
->vs_professional.exe for Visual Studio Professional
->vs_enterprise.exe for Visual Studio Enterprise

If you receive a User Account Control notice, choose Yes.

4.Choose workloads
Find the workload you want in the Visual Studio Installer.
"Desktop development with C++" is a required workload for the project.

5.Start download and Install the Visual Studio 2019 setup

- Choose individual components (Optional)
- Install language packs (Optional)
- Select the installation location (Optional)
- Select "Install"

6.Start developing
After Visual Studio installation is complete, choose the Launch button to get started developing with Visual Studio.

On the start window, choose Create a new project.

OpenCV

NOTE: *If you have previous/other manually installed (= not installed via pip) version of OpenCV installed (e.g. cv2 module in the root of Python's site-packages), remove it before installation to avoid conflicts.*

1.Download and Install OpenCV-4.2.0

Download OpenCV 4.2.0 latest stable release (opencv-4.2.0-vc14_vc15.exe) for Windows platform. Go to the official OpenCV website: *https://opencv.org/*

Go to *Resources -> Releases* and click on the *Windows platform*.
You will be redirected to *SourceForge* and download will automatically start.

2.Add OpenCV binaries 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.

3.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.

4.Change Settings

- Set platform target to x64 — Pre-built binaries are built for x64 Windows platforms.
- Add to Include Directories — Tell the compiler how the OpenCV library looks. This is done by providing a path to the header files (build/include).[1]
- Add to Library Directories — Tell the linker where it can find the lib files for different modules.
- Add Additional Dependencies — List .lib files for different modules. Note that we're only going to list a single all-in-one file named OpenCV_world.
- 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*
- Go to Edit the VC++ project linker with the opencv_world420d.lib OpenCV dynamic library. You will find the DLL (Dynamic Link Library) here: *C:\OpenCV-4.2.0\opencv\build\x64\vc15\lib*
- Now copy the name of the file opencv_world420d.lib and paste it in the dependency box.

You are now ready to build your first project!

Working

1. **Import necessary libraries-**

   ```
   #include <opencv2/opencv.hpp> //File that defines what modules where included during the build of OpenCV
   #include <iostream>
   ```

2. **Import namespaces**

   ```
   using namespace cv;
   using namespace std;
   ```

3. **Create a function**

Syntax: <return_type> <function_name> ()

Example: Mat rotate(Mat src, double angle) // rotate function returns mat object with parameters image file and angle.

4. **Inside the Function**

- Mat dst;
  Mat object for output image file

- Point2f pt(src.cols/2., src.rows/2.);
  Point from where to rotate

- getRotationMatrix2D(pt, angle, 1.0);
  Mat object for storing after rotation

- warpAffine(src, dst, r, Size(src.cols, src.rows));
  Apply an affine transforation to image.

- return dst;
  returning Mat object for output image file

5. Inside the main function()

```
int main()
  {
    int angle;
    Mat src = imread("IMAGE_PATH");   //import image

    Mat dst;
    cout << "Enter an angle for anti-clockwise rotation(in degrees):";
    cin >> angle;            //input value
    dst = rotate(src, angle);        //function call

    string srcname = "Source Image";     //create window1: Source Image
    namedWindow(srcname);
    string windowname = "Rotated Image";        //create window2: Rotated Image
    namedWindow(windowname);

    imshow(srcname, src);         //display window1: Source Image
    imshow(windowname, dst);             //display window2: Rotated Image
    waitKey(0);             //to exit press any key
    destroyAllWindows();          //to destroy all windows created after key press
    return 0;
  }
```

6. Build and Run the Project

Select Build->Build Solution.
Select Debug->Start without debugging.

In the Terminal Window, input the desired angle for image rotation.
Press Enter.

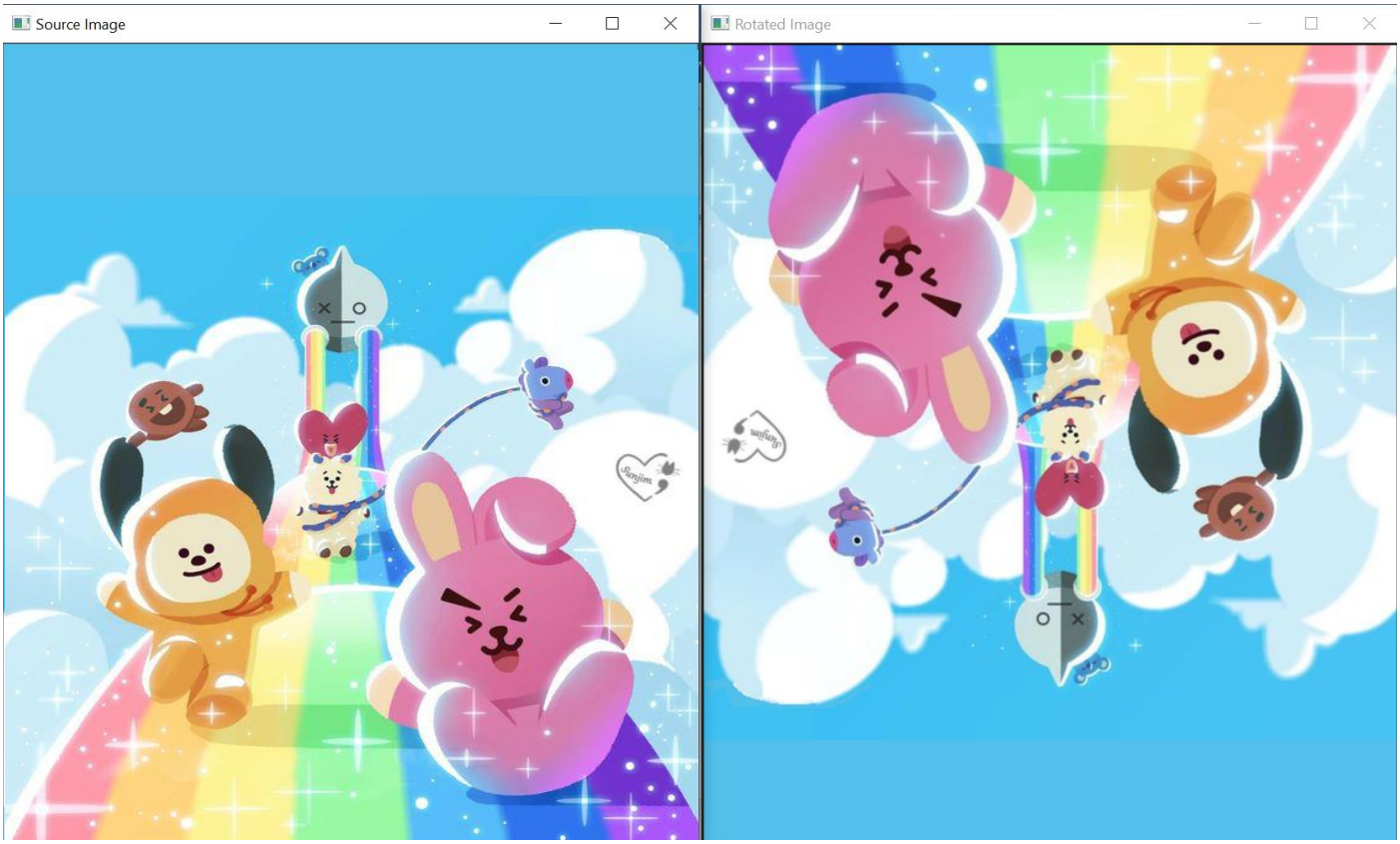
Two windows, displaying the source image and rotated image will pop up!

Through this project, we are now able to rotate an image of any size by any desired angle.

Sample Input

```
Enter an angle for anti-clockwise rotation(in degrees):180
```

Sample Output



The output image will be rotated at any desired angle, here angle for rotation is 180 degrees.

Author

Name: Khushi Kumar
University: Graphic Era deemed to be University, Dehradun