PROJECT1: Image Rotation Using OpenCv

A C++ program that takes an image as input and an angle and rotate the image by the given angle then displays the rotated image as an output.

**TECHNOLOGY USED:**

Programming Language:C++

Tools:OpenCV 4.5.1

Soft-wares: Visual Studio 2019(community)-16.8.3

Platforms:

<https://github.com/opencv/opencv/releases>

<https://docs.opencv.org/3.4/d4/d61/tutorial_warp_affine.html>

**STEPS TO RUN THE PROGRAM :-**

Download ZIP and extract the file on your local system or clone repository using below command in command prompt :

Install OpenCV executable file (latest version) in your system. Download link:-

<https://github.com/opencv/opencv/releases>

Set up path of OpenCV in environment variables

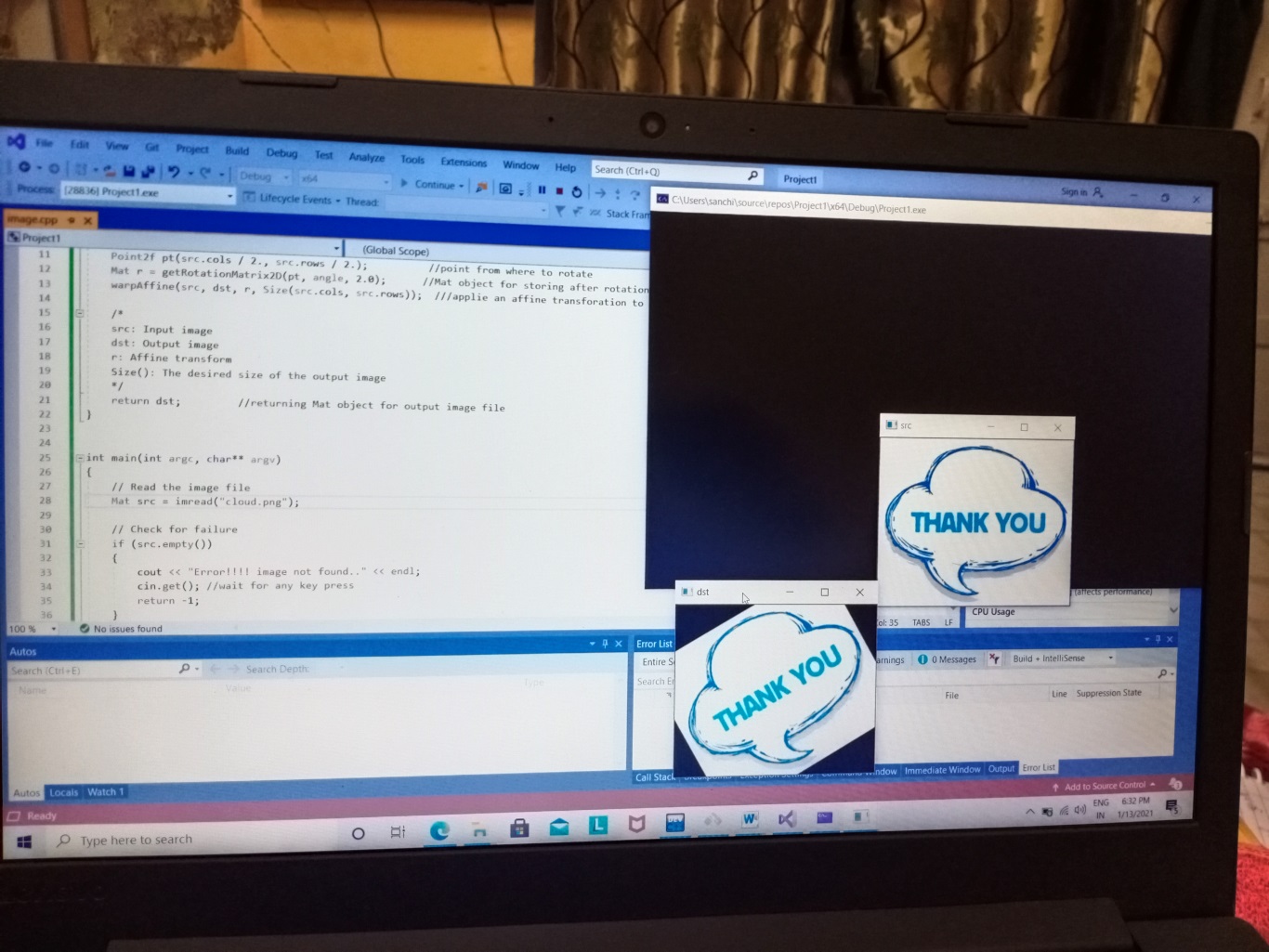
Open cloned project in visual studio

Now set up the environment for OpenCV in Visual Studio For Reference :-

<https://medium.com/@subwaymatch/opencv410-with-vs-2019-3d0bc0c81d96>

6)Now you are ready to build your c++ program. Select x64 debugger before running your program.

7)Run your code using run button provided above and you output can be seen as below.



**This is a C++ program to rotate the input file(image) by the use of wrafAffine() Function of OpenCV.**

funtion which is inbuilt in the opencv library.

wrafAffine takes 4 parametres:

First is source image file ,

second is destination image file ,

third parameter is mat object that is ouput after rotating with some point

taking into reference and last is taking the size.

**What is an Affine Transformation?**

1. A transformation that can be expressed in the form of a matrix multiplication (linear transformation) followed by a vector addition (translation).

2. From the above, we can use an Affine Transformation to express: a. Rotations (linear transformation)

b. Translations (vector addition)

c. Scale operations (linear transformation)

you can see that, in essence, an Affine Transformation represents a relation between two images.

3. The usual way to represent an Affine Transformation is by using a 2×3 matrix. A=[a00a10a01a11]2×2B=[b00b10]2×1

M=[AB]=[a00a10a01a11b00b10]2×3

Considering that we want to transform a 2D vector X=[xy] by using A and B, we can do the same with:

T=A⋅[xy]+B or T=M⋅[x,y,1]T

T=[a00x+a01y+b00a10x+a11y+b10]

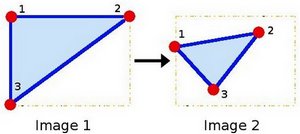
**How do we get an Affine Transformation?**

1. We mentioned that an Affine Transformation is basically a relation between two images. The information about this relation can come, roughly, in two ways:

a. We know both X and T and we also know that they are related. Then our task is to find M

b. We know M and X. To obtain T we only need to apply T=M⋅X. Our information for M may be explicit (i.e. have the 2-by-3 matrix) or it can come as a geometric relation between points.

2. Let's explain this in a better way (b). Since M relates 2 images, we can analyze the simplest case in which it relates three points in both images. Look at the figure below:



the points 1, 2 and 3 (forming a triangle in image 1) are mapped into image 2, still forming a triangle, but now they have changed notoriously. If we find the Affine Transformation with these 3 points (you can choose them as you like), then we can apply this found relation to all the pixels in an image.

Source :- Official Documentation OpenCV

<https://docs.opencv.org/3.4/d4/d61/tutorial_warp_affine.html>