**IMAGE PROCESSING PROJECT**

Name- Mansi Agarwal Semester - IV

Course – B.Tech CSE (DS AND AI)

University Roll no.- 2015176

The project is about using opencv to process an image in C++.In the following project I have displayed an image and rotated the image anti clockwise by any angle given as input by the user.

The tools and technology used are as follows:

1. Visual Studio
2. OpenCv

The project has been made as follows:

1. The first step is installing Visual studio and opencv on our system using Internet.
2. Once both of them got installed I modified the path in the system settings so the code that is written in Visual studio can be executed.
3. Now I created an empty console type C++ project on Visual studio and created a new source file which contains the code for my project.
4. Then code is as follows:

#include<opencv2/opencv.hpp>

#include<iostream>

using namespace std;

using namespace cv;

Mat rotate(Mat img, double angle)

{

Mat after;

Point2f pt(img.cols / 2., img.rows / 2.);

Mat r = getRotationMatrix2D(pt, angle, 1.0);

warpAffine(img, after, r, Size(img.cols, img.rows));

return after;

}

int main()

{

int a;

Mat img = imread("coding.jpg");

Mat after;

if (img.empty())

{

cout << "Error!!";

cin.get();

return -1;

}

cout << "Enter Angle by which the image should be rotated:"<<endl;

cin >> a;

after = rotate(img, a);

namedWindow("Image", WINDOW\_NORMAL);

imshow("Image", img);

namedWindow("Image After Rotation", WINDOW\_NORMAL);

imshow("Image After Rotation", after);

waitKey(0);

destroyWindow("Image");

destroyWindow("Image After Rotation");

return 0;

}

1. So first of all we include the header files for opencv and create a function rotate which has the image and angle as arguments. Inside the function the point of rotation is specified at which the image rotates through the given angle in anti clockwise direction.

warpAffine is used for the size of the output image.

imread is used to read/load the image.

img.empty() is used to check if the image is loaded properly ,if not it terminates the program with a message so that our time is saved.

2 windows are created for the images , one before and after rotation using namedWindow .

WINDOW\_NORMAL shows that the size of the window can be changed afterwards also.

imshow is used to display the images.

waitKey(0) is used to that the windows are closed when the user presses a button.

destroyWindow is used at the end of the program so that once the windows have performed their task they can be destroyed.

1. The above program will work only when we have made some additional settings .

First of all we should check that the solution platform is the right configuration ( like I changed it to x64).

Then click on the project properties and add additional include directories, additional library directories and additional dependencies.

1. Now Build the program and Run to get the output which is as follows:

