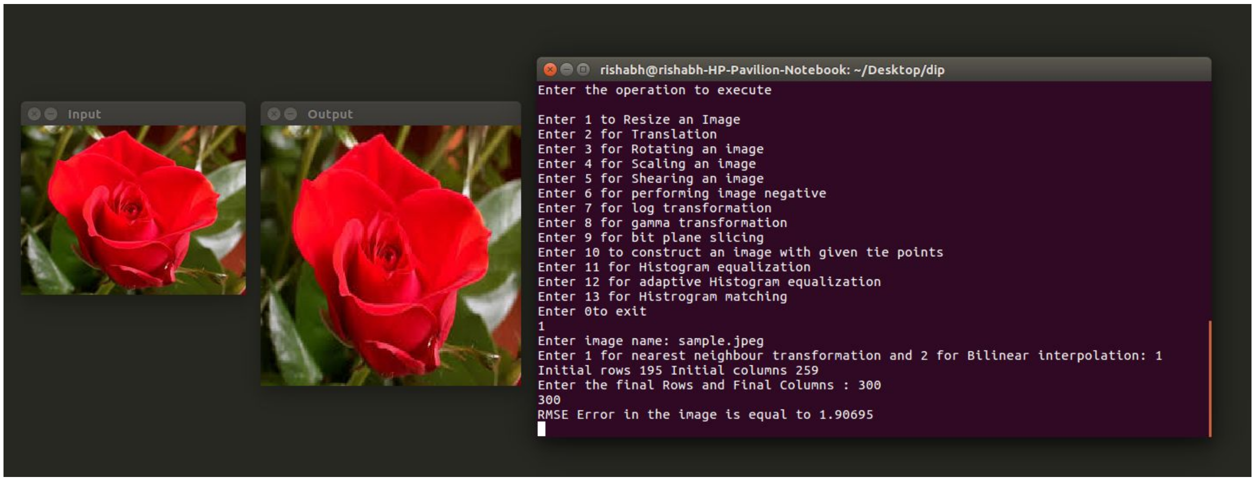
CSL-461
Rishabh Singh

# Digital Image Analysis-Report

## Image Resize

### Resizing Image

In the nearest neighbour technique use compute the nearest pixel matching the new size of Image The RMSE error observed in this case was found to be around zero and It is actually zero for some of the images.



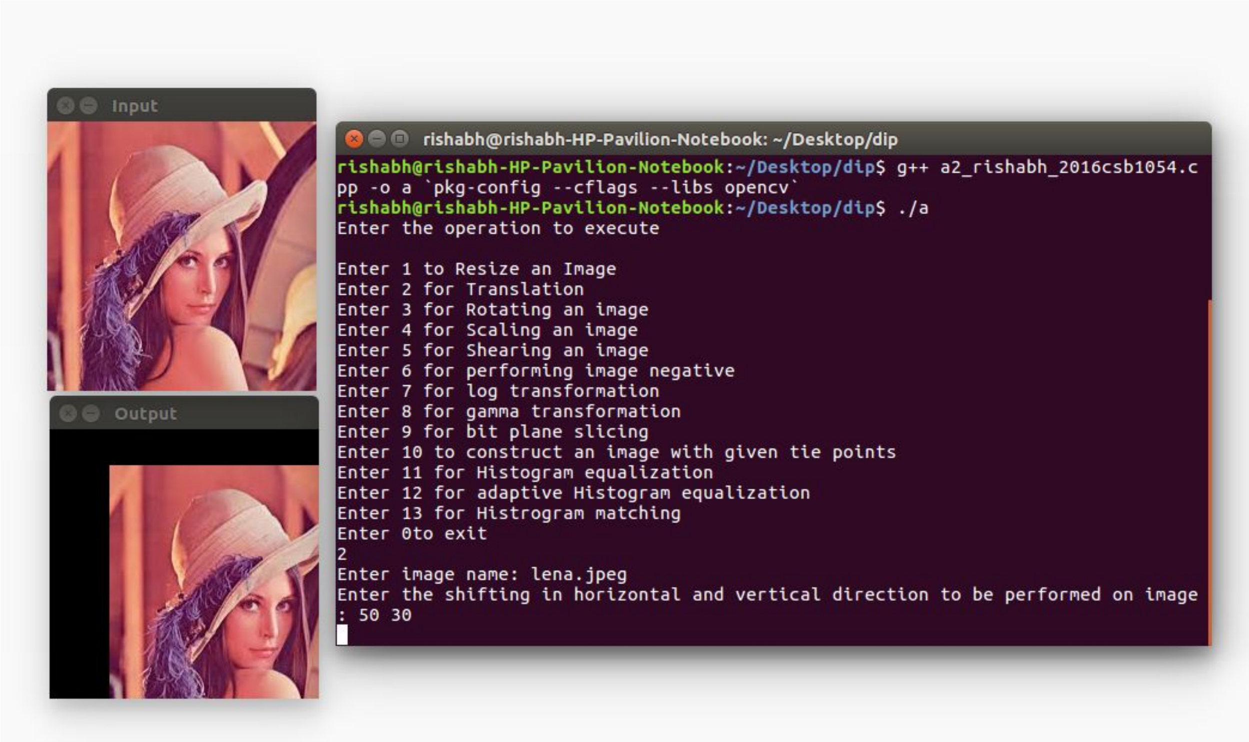
```
Enter the operation to execute

Enter 1 to Resize an Image
Enter 2 for Translation
Enter 3 for Rotating an image
Enter 4 for Scaling an image
Enter 5 for Shearing an image
Enter 6 for performing image negative
Enter 7 for log transformation
Enter 8 for gamma transformation
Enter 9 for bit plane slicing
Enter 10 to construct an image with given tie points
Enter 11 for Histogram equalization
Enter 12 for adaptive Histogram equalization
Enter 13 for Histrogram matching
Enter 0to exit
1
Enter image name: sample.jpeg
Enter 1 for nearest neighbour transformation and 2 for Bilinear interpolation: 1
Initial rows 195 Initial columns 259
Enter the final Rows and Final Columns : 300
300
RMSE Error in the image is equal to 1.90695
```

In the bilinear Interpolation technique the we use the nearest four coordinates of the image pixel to compute the pixel value at any point. In this case the RMSE error was around 5.

## Affine Transformation

### Translation

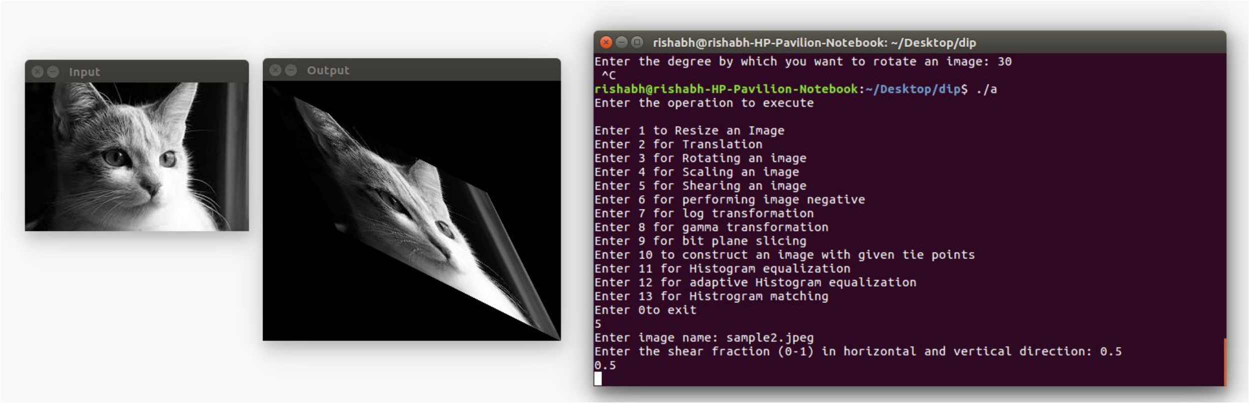
Translation refers to translating the image in the horizontal or vertical direction. It is simply one to one mapping of the input image to output image displaced by some distance.



```
rishabh@rishabh-HP-Pavilion-Notebook:~/Desktop/dip$ g++ a2_rishabh_2016csb1054.c
pp -o a `pkg-config --cflags --libs opencv`
rishabh@rishabh-HP-Pavilion-Notebook:~/Desktop/dip$ ./a
Enter the operation to execute

Enter 1 to Resize an Image
Enter 2 for Translation
Enter 3 for Rotating an image
Enter 4 for Scaling an image
Enter 5 for Shearing an image
Enter 6 for performing image negative
Enter 7 for log transformation
Enter 8 for gamma transformation
Enter 9 for bit plane slicing
Enter 10 to construct an image with given tie points
Enter 11 for Histogram equalization
Enter 12 for adaptive Histogram equalization
Enter 13 for Histrogram matching
Enter 0to exit
2
Enter image name: lena.jpeg
Enter the shifting in horizontal and vertical direction to be performed on image
: 50 30
```

## Shearing

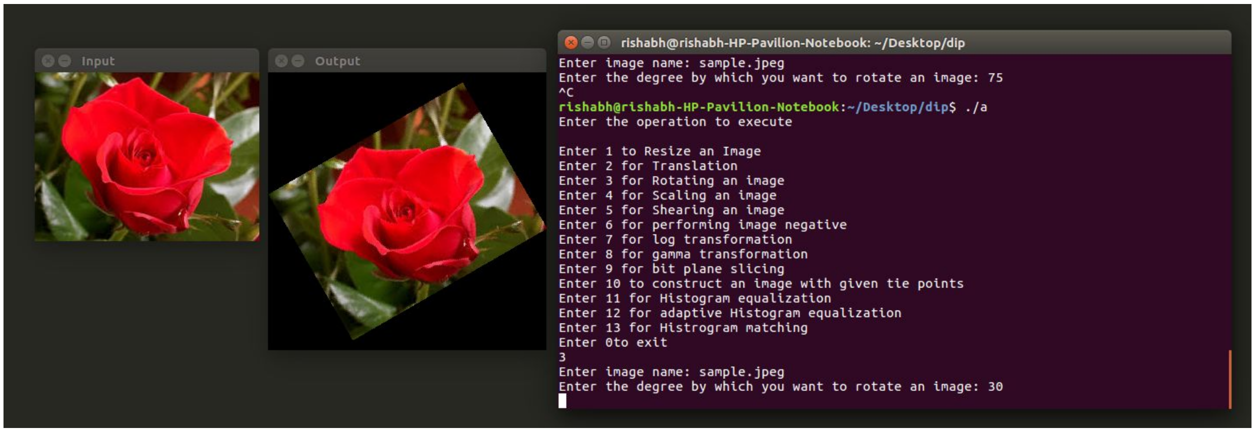
A shear mapping is a linear map that displaces each point in fixed direction, by an amount proportional to its  signed distance from a line that is parallel to that direction.



```
Enter the degree by which you want to rotate an image: 30
 ^C
rishabh@rishabh-HP-Pavilion-Notebook:~/Desktop/dip$ ./a
Enter the operation to execute

Enter 1 to Resize an Image
Enter 2 for Translation
Enter 3 for Rotating an image
Enter 4 for Scaling an image
Enter 5 for Shearing an image
Enter 6 for performing image negative
Enter 7 for log transformation
Enter 8 for gamma transformation
Enter 9 for bit plane slicing
Enter 10 to construct an image with given tie points
Enter 11 for Histogram equalization
Enter 12 for adaptive Histogram equalization
Enter 13 for Histrogram matching
Enter 0to exit
5
Enter image name: sample2.jpeg
Enter the shear fraction (0-1) in horizontal and vertical direction: 0.5
0.5
```

## Rotation

In this case a image is rotated by some degree in the anticlockwise sense. The output image size is fixed such that it actually fits in the new window.



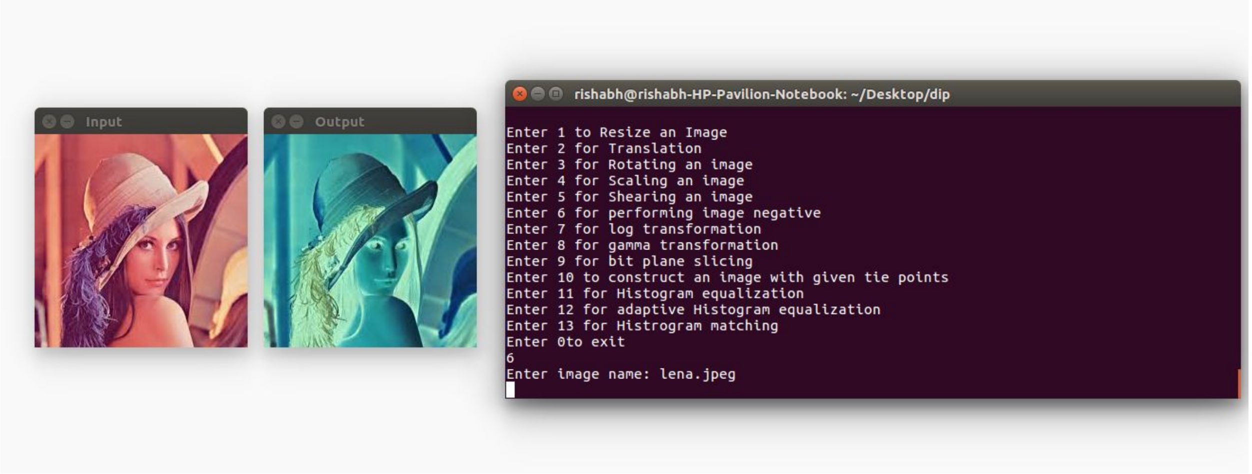
```
Enter image name: sample.jpeg
Enter the degree by which you want to rotate an image: 75
^C
rishabh@rishabh-HP-Pavilion-Notebook:~/Desktop/dip$ ./a
Enter the operation to execute

Enter 1 to Resize an Image
Enter 2 for Translation
Enter 3 for Rotating an image
Enter 4 for Scaling an image
Enter 5 for Shearing an image
Enter 6 for performing image negative
Enter 7 for log transformation
Enter 8 for gamma transformation
Enter 9 for bit plane slicing
Enter 10 to construct an image with given tie points
Enter 11 for Histogram equalization
Enter 12 for adaptive Histogram equalization
Enter 13 for Histrogram matching
Enter 0to exit
3
Enter image name: sample.jpeg
Enter the degree by which you want to rotate an image: 30
```

## Scaling

It is similar to the case in which we were the resizing of the image. So we can say that Image resizing is basically part of affine transformation.

# Negative

As the name suggest Image negative is the reverse values of the image pixels of any image i.e. if image pixel is x then the value of that pixel in the output image will be 255-x.

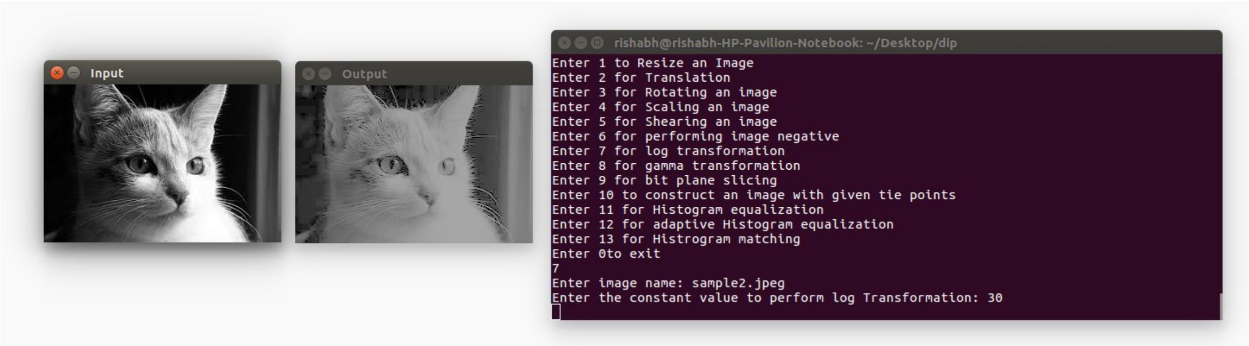


```
Enter 1 to Resize an Image
Enter 2 for Translation
Enter 3 for Rotating an image
Enter 4 for Scaling an image
Enter 5 for Shearing an image
Enter 6 for performing image negative
Enter 7 for log transformation
Enter 8 for gamma transformation
Enter 9 for bit plane slicing
Enter 10 to construct an image with given tie points
Enter 11 for Histogram equalization
Enter 12 for adaptive Histogram equalization
Enter 13 for Histrogram matching
Enter 0to exit
6
Enter image name: lena.jpeg
```

## Log Transformation

In this case we calculate the value of output image pixel by applying the log on the input image multiplied by some constant.

$$outputImage(r, c) = C * log(inputImage(r, c) + 1)$$

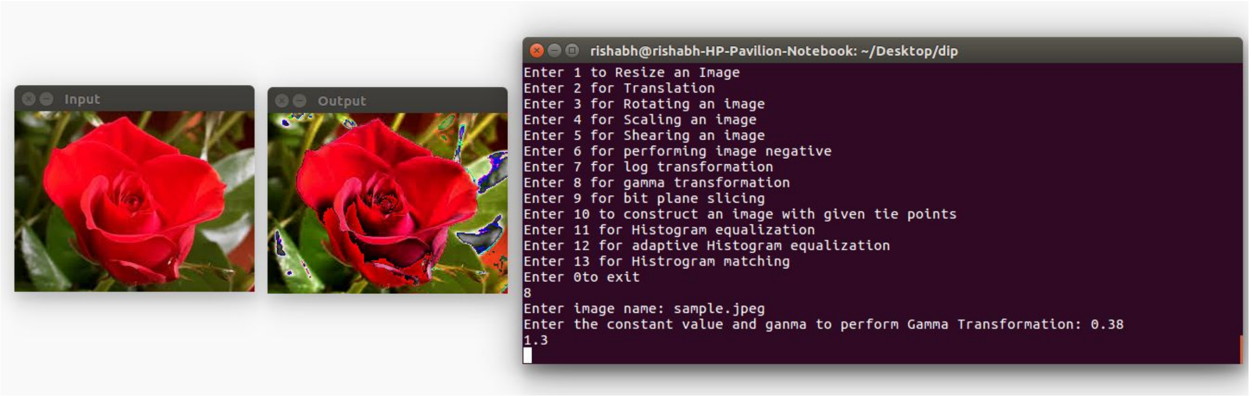


```
Enter 1 to Resize an Image
Enter 2 for Translation
Enter 3 for Rotating an image
Enter 4 for Scaling an image
Enter 5 for Shearing an image
Enter 6 for performing image negative
Enter 7 for log transformation
Enter 8 for gamma transformation
Enter 9 for bit plane slicing
Enter 10 to construct an image with given tie points
Enter 11 for Histogram equalization
Enter 12 for adaptive Histogram equalization
Enter 13 for Histrogram matching
Enter 0to exit
7
Enter image name: sample2.jpeg
Enter the constant value to perform log Transformation: 30
```

## Gamma Transformation

It is similar to log transformation except in this case instead of taking log we take power of the value of input image pixel values.

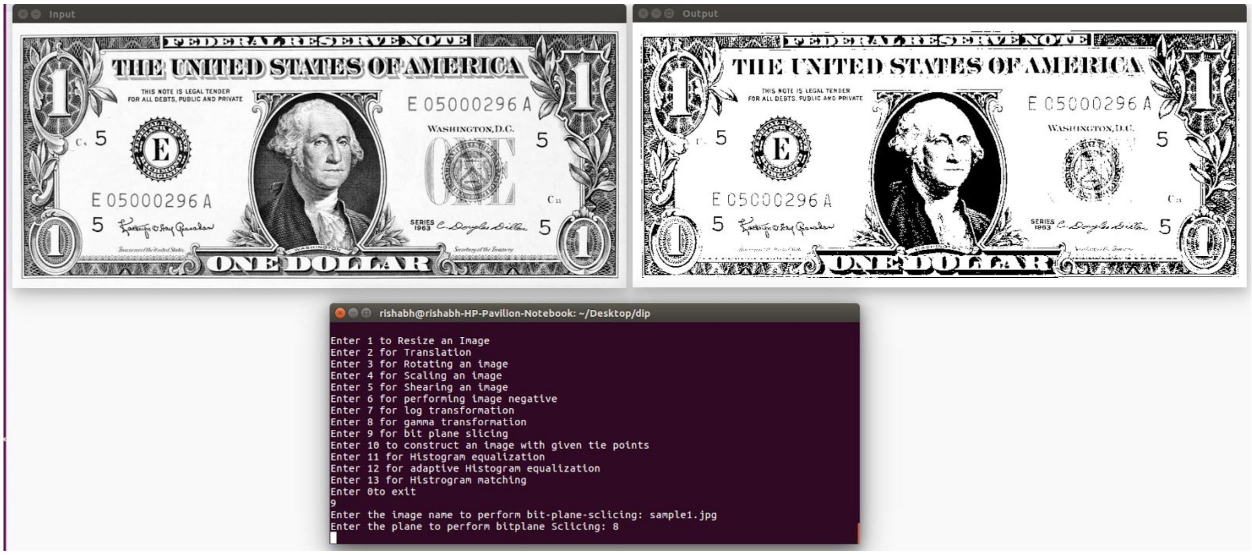
$$outputImage(r, c) = C * (inputImage(r, c))^{gamma}$$



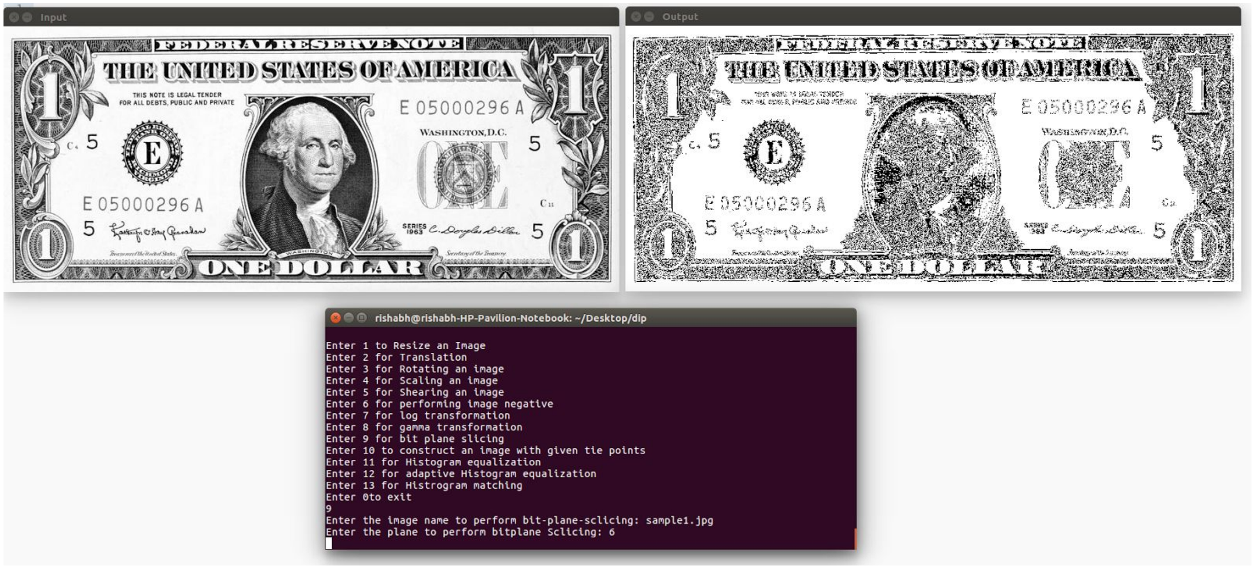
```
Enter 1 to Resize an Image
Enter 2 for Translation
Enter 3 for Rotating an image
Enter 4 for Scaling an image
Enter 5 for Shearing an image
Enter 6 for performing image negative
Enter 7 for log transformation
Enter 8 for gamma transformation
Enter 9 for bit plane slicing
Enter 10 to construct an image with given tie points
Enter 11 for Histogram equalization
Enter 12 for adaptive Histogram equalization
Enter 13 for Histrogram matching
Enter 0to exit
8
Enter image name: sample.jpeg
Enter the constant value and gamma to perform Gamma Transformation: 0.38
1.3
```

## Bit Plane Slicing

In this case of transformation we convert the pixel values in the binary and take only one bit as the pixels values of output image. If the pixel value in the bit-plane of some particular input image is greater than zero the in output image the result is 255 else it is 0.



```
Enter 1 to Resize an Image
Enter 2 for Translation
Enter 3 for Rotating an image
Enter 4 for Scaling an image
Enter 5 for Shearing an image
Enter 6 for performing image negative
Enter 7 for log transformation
Enter 8 for gamma transformation
Enter 9 for bit plane slicing
Enter 10 to construct an image with given tie points
Enter 11 for Histogram equalization
Enter 12 for adaptive Histogram equalization
Enter 13 for Histrogram matching
Enter 0to exit
9
Enter the image name to perform bit-plane-sclicing: sample1.jpg
Enter the plane to perform bitplane Sclicing: 8
```



```
Enter 1 to Resize an Image
Enter 2 for Translation
Enter 3 for Rotating an image
Enter 4 for Scaling an image
Enter 5 for Shearing an image
Enter 6 for performing image negative
Enter 7 for log transformation
Enter 8 for gamma transformation
Enter 9 for bit plane slicing
Enter 10 to construct an image with given tie points
Enter 11 for Histogram equalization
Enter 12 for adaptive Histogram equalization
Enter 13 for Histrogram matching
Enter 0to exit
9
Enter the image name to perform bit-plane-sclicing: sample1.jpg
Enter the plane to perform bitplane Sclicing: 6
```

## Piecewise Transformation

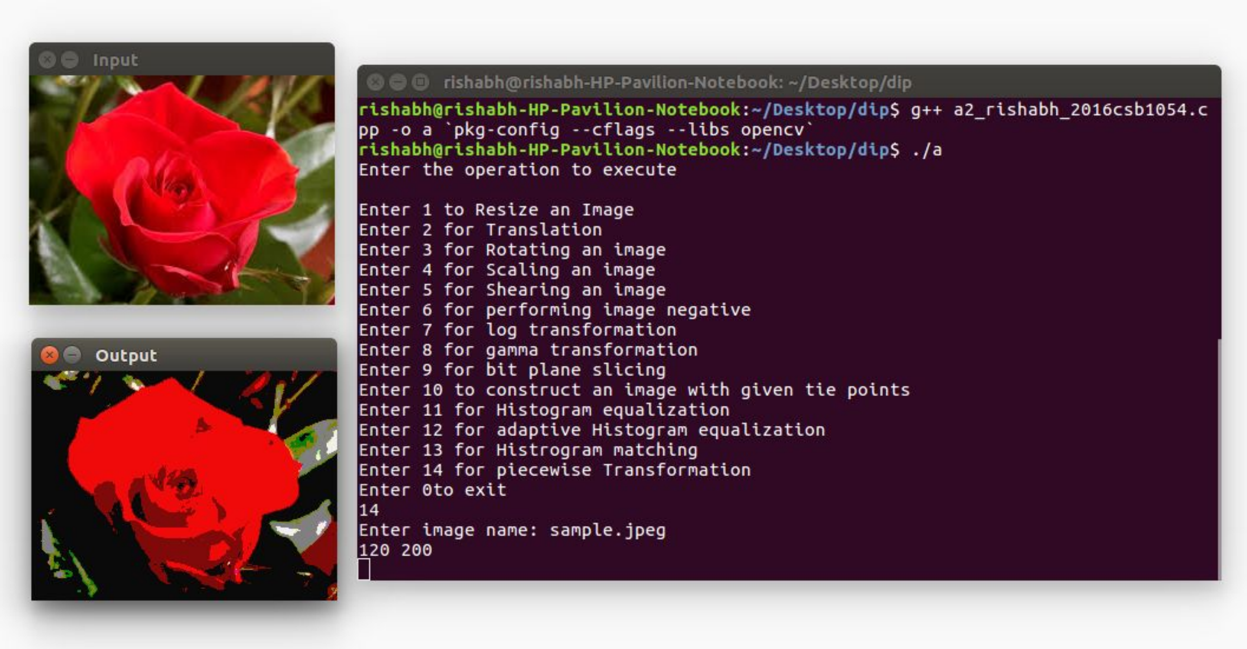
In this type of image Transformation we do piecewise operations like suppose user enters the values 100 , 150 then we do something like this:

$$if\ (\ inputImage(r,c)\ <\ 100)\ do\ outputImage(r,c\ )\ =\ c1$$

$$else\ if\ (\ inputImage(r,c)\ >\ 150)\ do\ outputImage(r,c\ )\ =\ c3$$

$$else\ \ do\ outputImage(r,c\ )\ =\ c2$$

Where c1,c2,c3 are constants lying between 0-255.



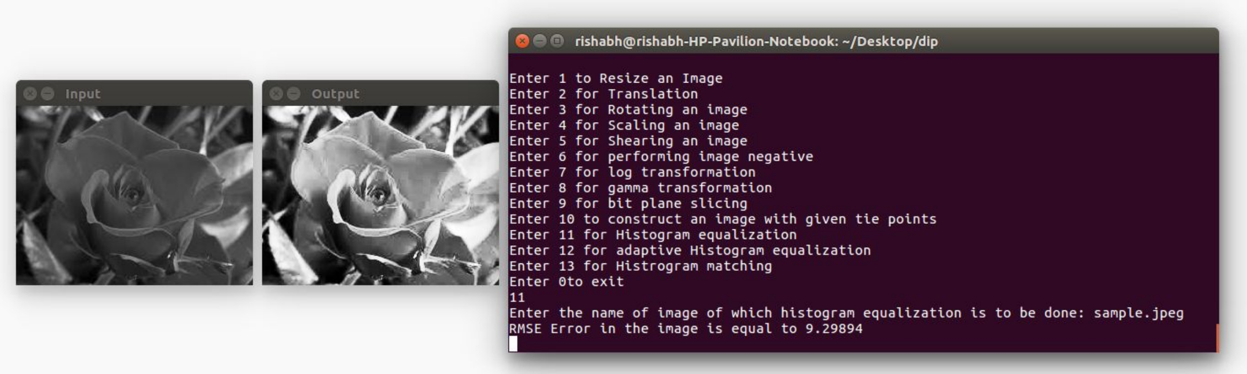
```
rishabh@rishabh-HP-Pavilion-Notebook:~/Desktop/dip$ g++ a2_rishabh_2016csb1054.c
pp -o a `pkg-config --cflags --libs opencv`
rishabh@rishabh-HP-Pavilion-Notebook:~/Desktop/dip$ ./a
Enter the operation to execute

Enter 1 to Resize an Image
Enter 2 for Translation
Enter 3 for Rotating an image
Enter 4 for Scaling an image
Enter 5 for Shearing an image
Enter 6 for performing image negative
Enter 7 for log transformation
Enter 8 for gamma transformation
Enter 9 for bit plane slicing
Enter 10 to construct an image with given tie points
Enter 11 for Histogram equalization
Enter 12 for adaptive Histogram equalization
Enter 13 for Histrogram matching
Enter 14 for piecewise Transformation
Enter 0to exit
14
Enter image name: sample.jpeg
120 200
```

## Tie point construction

In this case we have a deformed image and some coordinates of the actual image related with the deformed image. With these coordinates we calculate the transformation matrix of the input image. With this matrix we backtrace the input image.

## Histogram Equalization

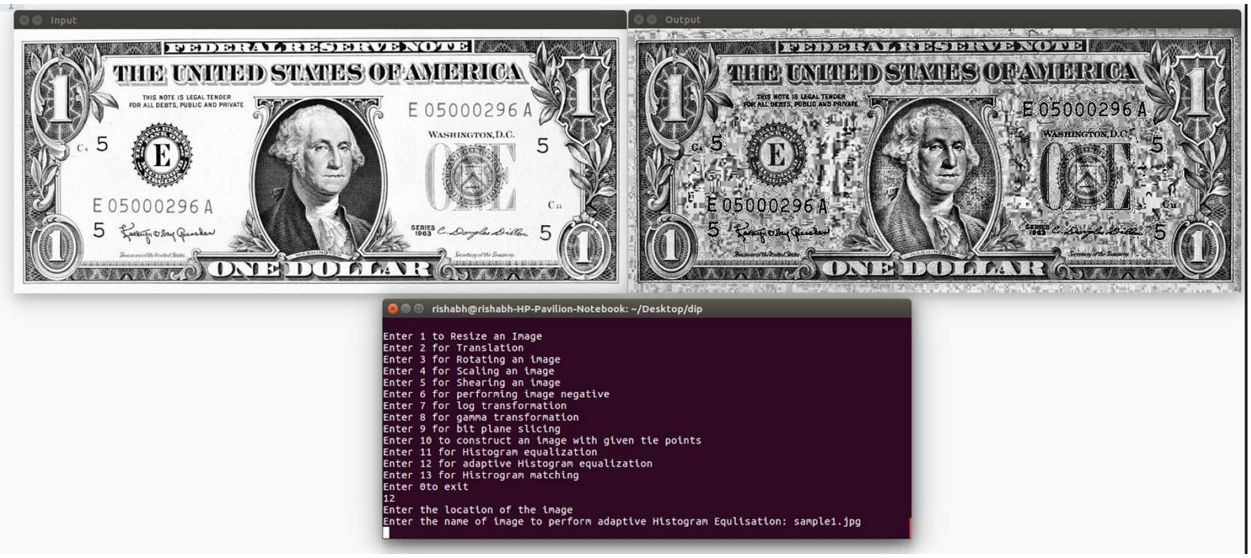
As the name suggest Histogram Equalization is the image processing technique in which we Equalize the histogram of the input image to form the output image. In my case the RMSE error calculated with actual OpenCV Function was close to 7.



```
Enter 1 to Resize an Image
Enter 2 for Translation
Enter 3 for Rotating an image
Enter 4 for Scaling an image
Enter 5 for Shearing an image
Enter 6 for performing image negative
Enter 7 for log transformation
Enter 8 for gamma transformation
Enter 9 for bit plane slicing
Enter 10 to construct an image with given tie points
Enter 11 for Histogram equalization
Enter 12 for adaptive Histogram equalization
Enter 13 for Histrogram matching
Enter 0to exit
11
Enter the name of image of which histogram equalization is to be done: sample.jpeg
RMSE Error in the image is equal to 9.29894
```

## Adaptive Histogram Equalization

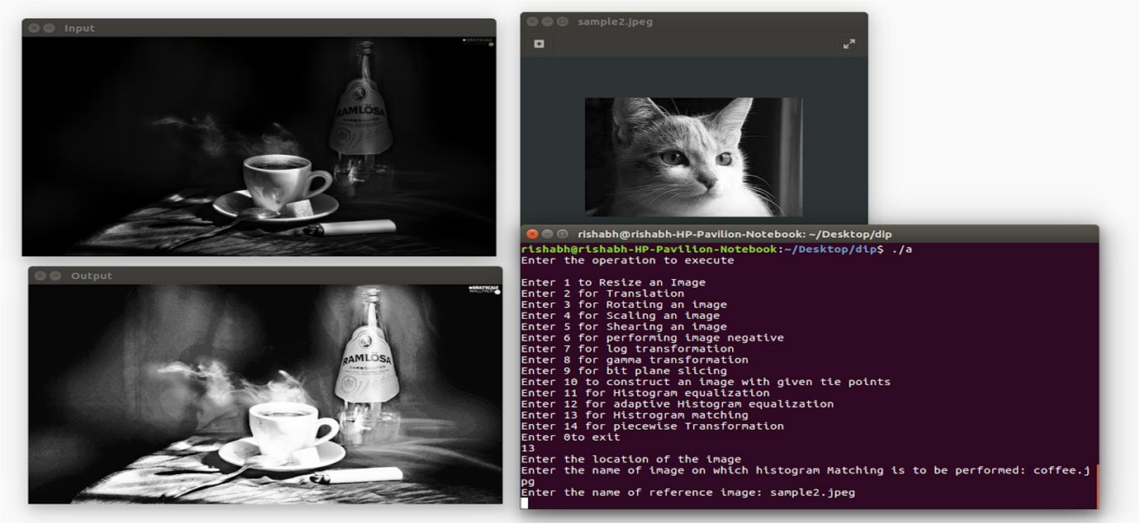
It is also the contrast stretching technique in which the edges in the image are highlighted properly. In its simplest form, each pixel is transformed based on the histogram of a square surrounding the pixel.



```
Enter 1 to Resize an Image
Enter 2 for Translation
Enter 3 for Rotating an image
Enter 4 for Scaling an image
Enter 5 for Shearing an image
Enter 6 for performing image negative
Enter 7 for log transformation
Enter 8 for gamma transformation
Enter 9 for bit plane slicing
Enter 10 to construct an image with given tie points
Enter 11 for Histogram equalization
Enter 12 for adaptive Histogram equalization
Enter 13 for Histrogram matching
Enter 0to exit
12
Enter the location of the image
Enter the name of image to perform adaptive Histogram Equlisation: sample1.jpg
```

# Histogram Matching

As the name suggest Histogram matching is an Image Processing technique in which we match the histogram of one image to another image. By this we can achieve different histogram of the input image as per our needs.



```
rishabh@rishabh-HP-Pavilion-Notebook:~/Desktop/dip$ ./a
Enter the operation to execute

Enter 1 to Resize an Image
Enter 2 for Translation
Enter 3 for Rotating an image
Enter 4 for Scaling an image
Enter 5 for Shearing an image
Enter 6 for performing image negative
Enter 7 for log transformation
Enter 8 for gamma transformation
Enter 9 for bit plane slicing
Enter 10 to construct an image with given tie points
Enter 11 for Histogram equalization
Enter 12 for adaptive Histogram equalization
Enter 13 for Histrogram matching
Enter 14 for piecewise Transformation
Enter 0to exit
13
Enter the location of the image
Enter the name of image on which histogram Matching is to be performed: coffee.j
pg
Enter the name of reference image: sample2.jpeg
```