CVIP PA1

Charanya Sudharsanan

Ubit Name : csudhars

UB Person Number : 50245956

Problem (1) **(1D and 2D Convolution on Images)**

Perform 2D convolution on grayscale Image to obtain gradient images G_x and G_y. Include report.

We convert the image to grayscale and perform 2D Convolution on image with a 3X3 sobelx filter and 3X3 sobely filter.

```
sobelx = np.array([[-1,0,1],[-2,0,2],[-1,0,1]],dtype = np.float)
sobely = np.array([[-1,-2,-1],[0,0,0],[1,2,1]],dtype = np.float)
```

We do zero padding of width 1 on all edges to not lose the edge row information.

We calculate the time taken for 2D Convolution using the time package.

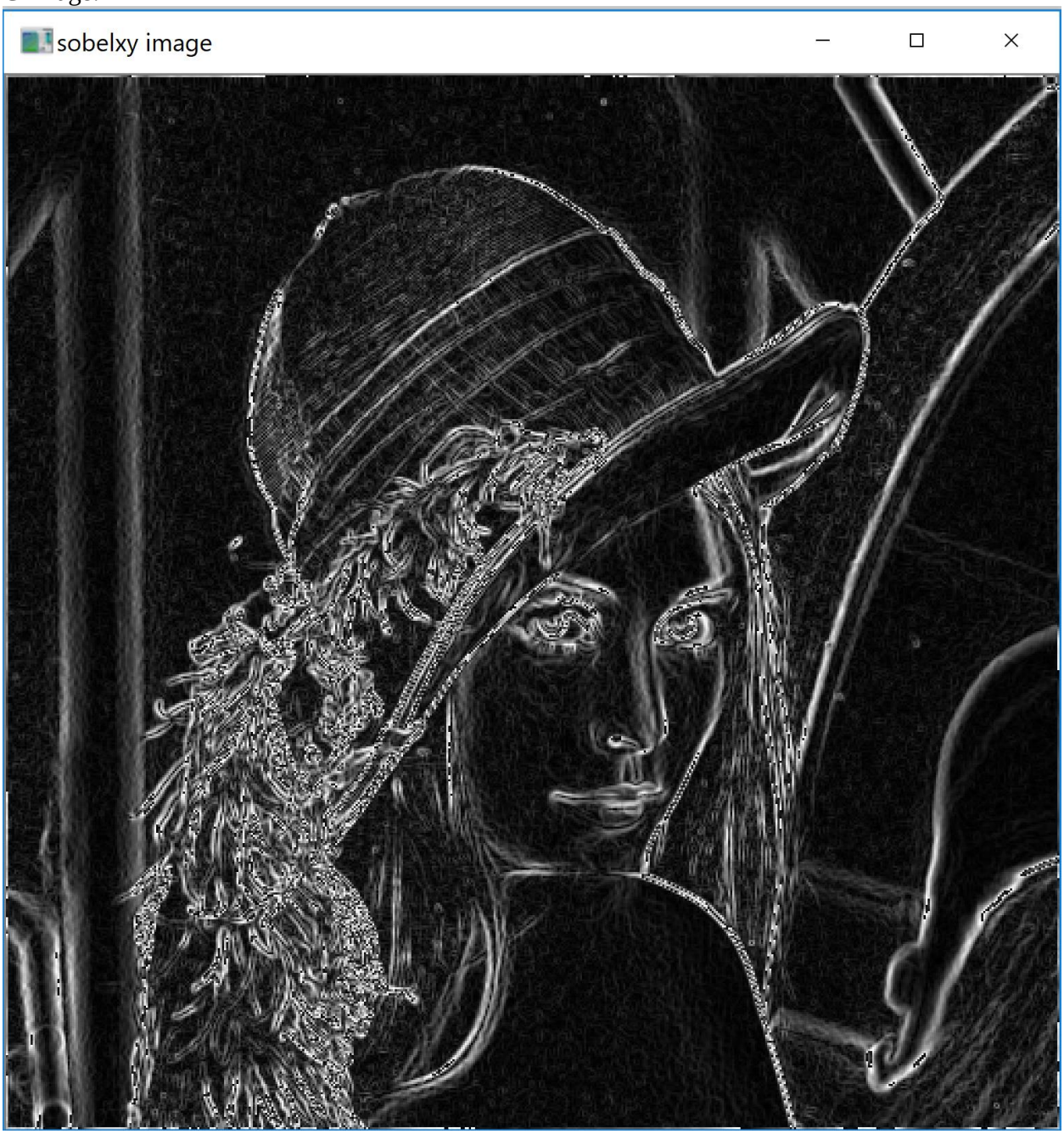
The Output Images obtained after 2D Convolution are:

Input Image:

Gx Image :

Gy Image:

G Image:



Time Taken for 2D Convolution:

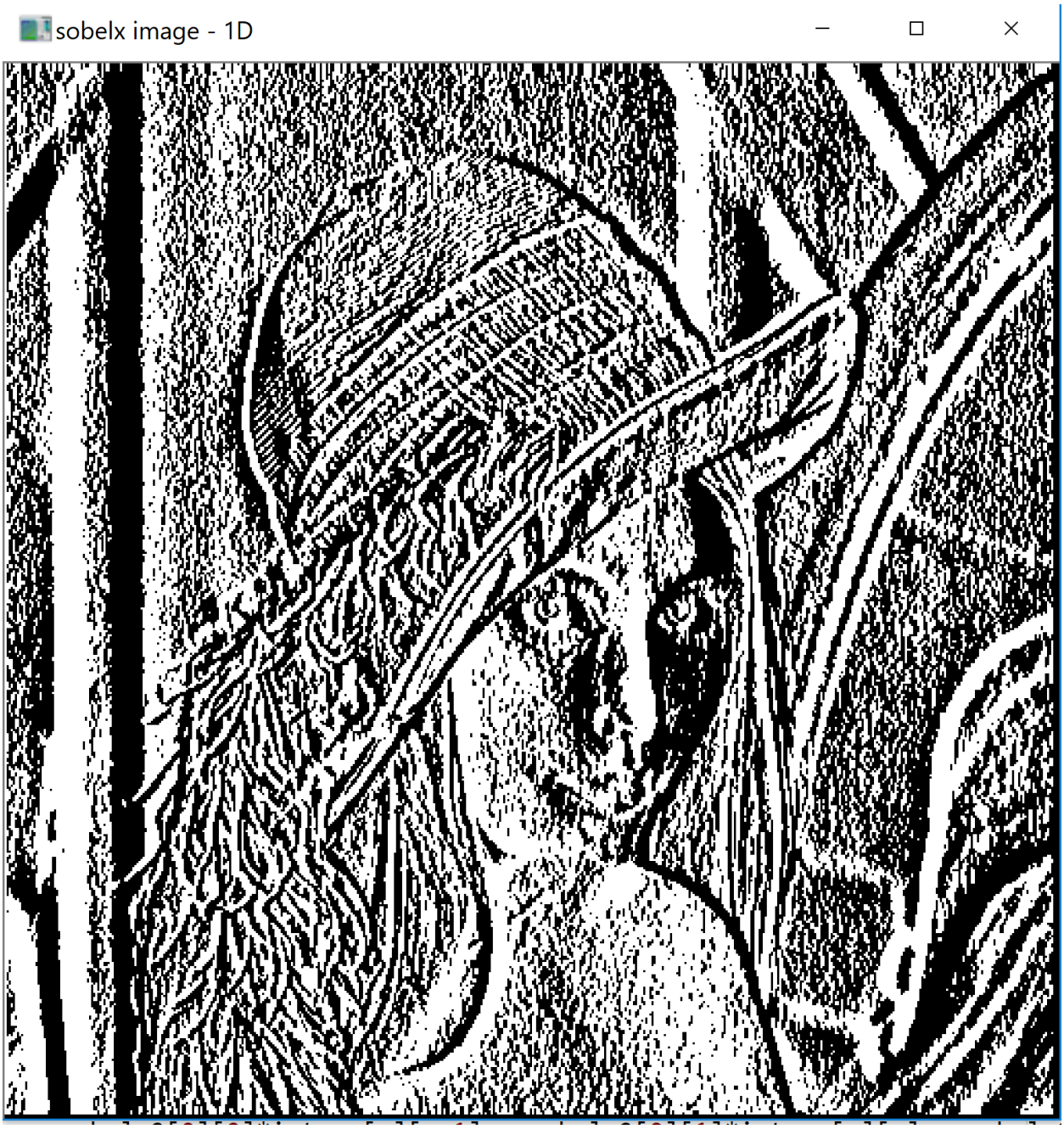
```
Time taken for 2D Convolution : 3.16461414394729
```

(b) Perform 1D convolution on grayscale Image **lena_gray.jpg** with 1D-filters specified above to obtain gradient images G_x and G_y. Include these two images in your report. Verify the result after 1D convolutions is same as the one obtained from 2D convolution from (a)

Filters are linear seperable. 1D Convolution is based on that. It reduces the computational complexity of fairly large kernels.

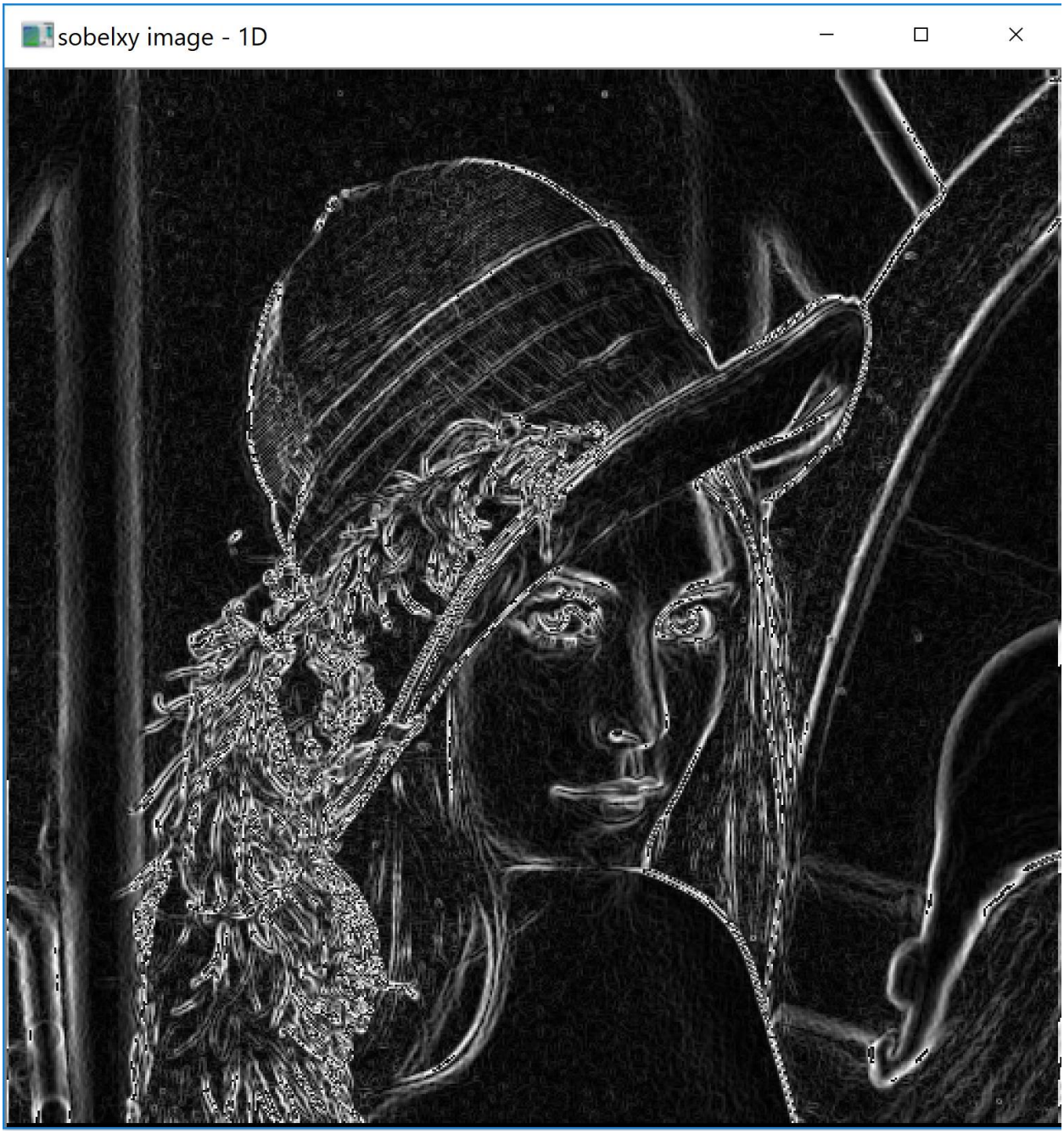
Output:

Sobel filter Gx – 1D:



Sobel Gy Image : 1D

Sobel G Image: 1-D



Time Taken for 1D Convolution:

```
Time taken for 1D Convolution : 5.763205581498255
```

As we can see 1-D Convolution takes more time than 2D Convolution. In question 1d, on using a larger kernel , there will be visible difference in the Computation between 1D and 2D.

Result Verification between 1D and 2D Convolution Outputs:

sobelx_output1, sobely_output1, sobelxy_output1 – Output for 1D
sobelx_output, sobely_output, sobelxy_output – Output for 2D

```
139
140print(np.array_equal(sobelx_output,sobelx_output1))
141print(np.array_equal(sobely_output,sobely_output1))
142print(np.array_equal(sobelxy_output,sobelxy_output1))
143
```

```
[   0.  -43.  -46. ... -107. -107.    0.]
[   0.    0.    0. ...    0.    0.    0.]]
True
True
True
```

(c)

Given an MxN Image and a PxQ filter, Report the computational complexity of performing 2D convolution vs using separable filters with 1D convolution.

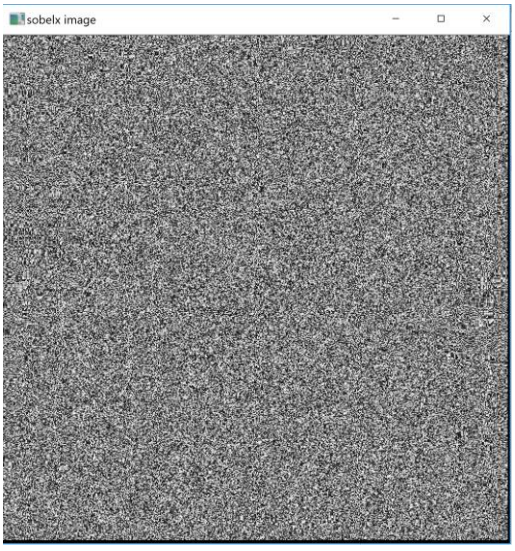
Complexity of 2D Convolution = MNPQ multiplies and adds

Complexity of 1D Convolution = First step requires MNP multiplies and adds, second step requires MNQ multiplies and adds. Totally it requires – MN (P + Q)

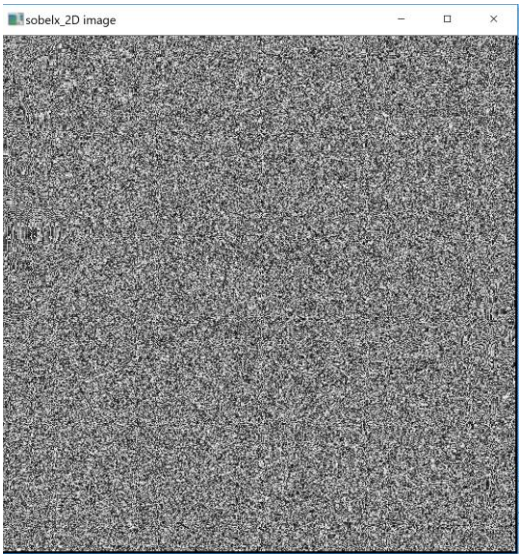
The computational advantage of separable convolution versus nonseparable convolution is therefore: PQ/ (P + Q)

(d) Create a separable filter of 100x100 dimension of your choice and apply it on Lena lena_gray.jpg using both 2D convolution function and 1D convolutions. Compare the execution time of these two routines using Python time libraries and report the execution times (Note: with just 3x3 filter, the speedup observed will be negligible)

Output Screenshots for 2D and 1D:
1D Convolution:



2D Convolution:

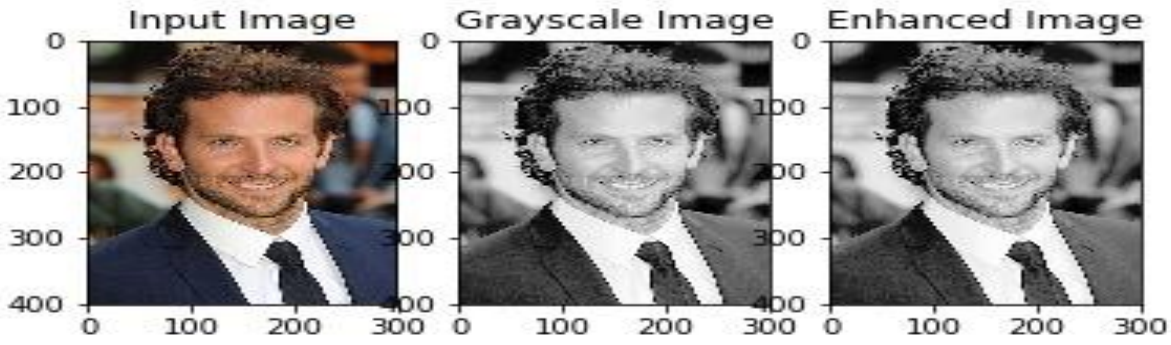


Time taken for 2D Convolution Vs 1D Convolution:

```
Time taken for 1D Convolution : 5.318465722664996
Time taken for 2D Convolution : 22.21383942754801
```

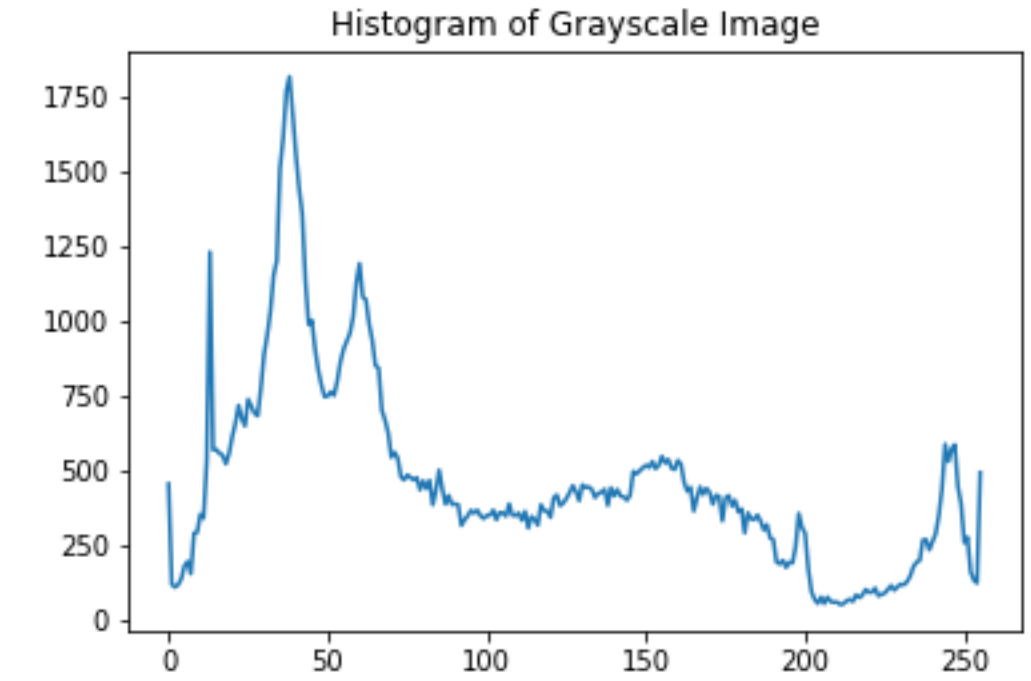
Problem (2) **(Histogram Equalization)**

Output:

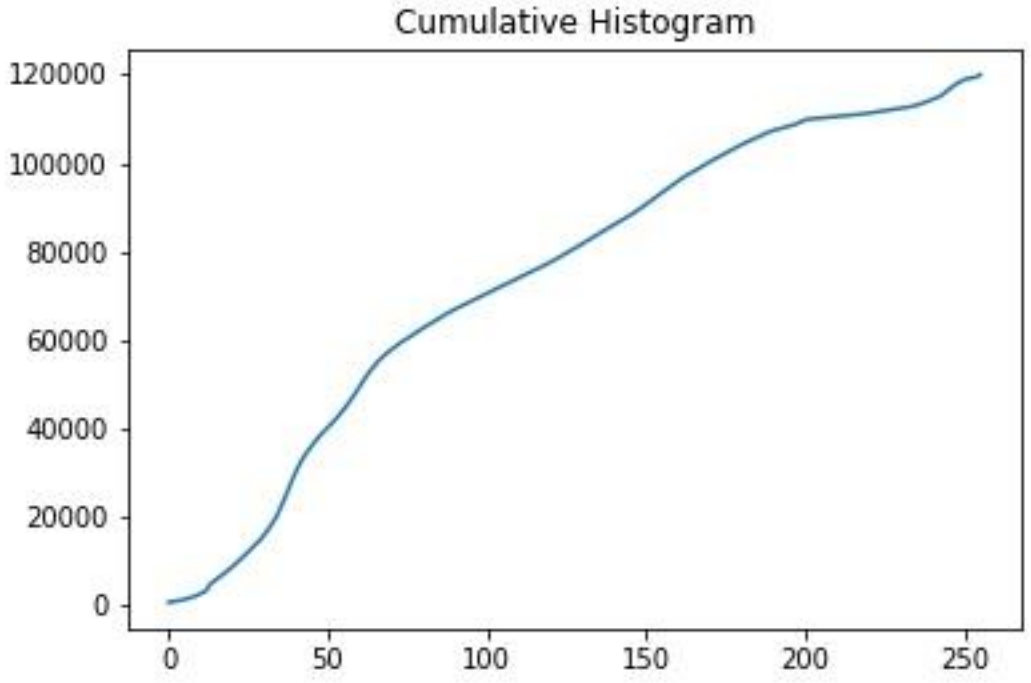


Histograms:

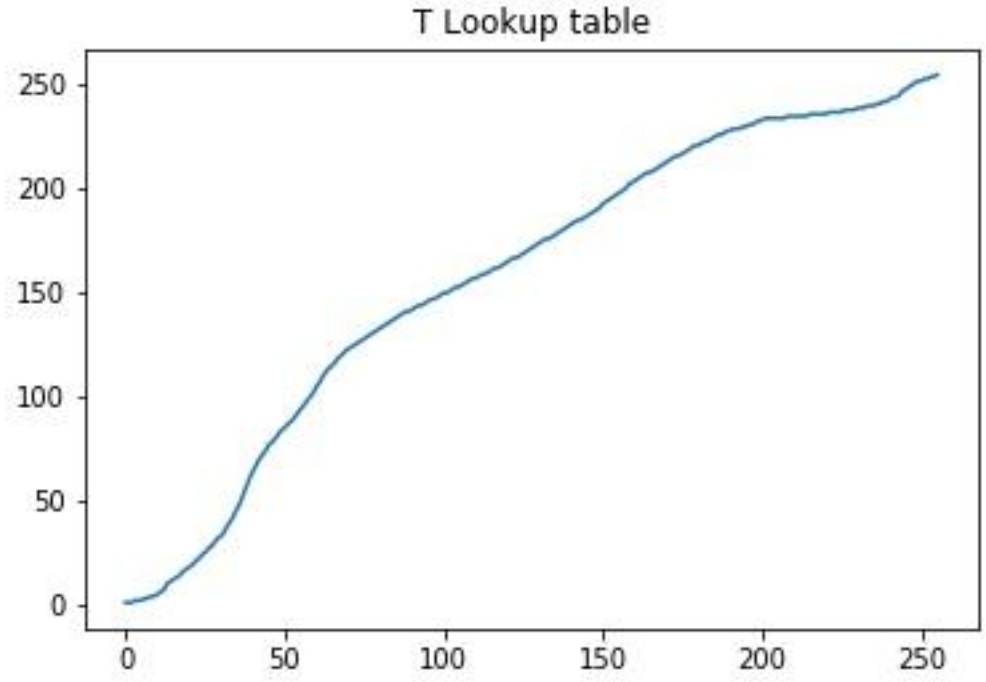
Histogram of H<Grayscale Image>:



Histogram of Hc<Cumulative histogram>:



Look Up Table/Transformation Function:



Histogram of Final Image: