

Image Reader

November 1, 2019

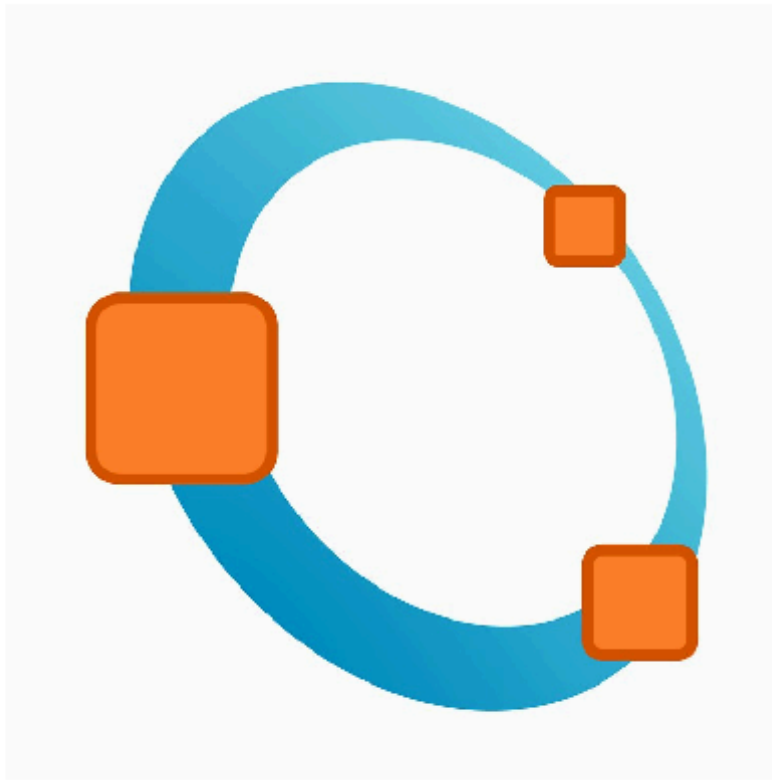
1 Homework 9 - Q1

1.1 Problem statement:

Pick an image – say your own passport size photograph – and convert it to gray scale using a tool such as gimp. Load it in octave as a matrix. 1. Reshape the matrix to a 1 D array. Make a histogram of this image using 16 bins that span over the 256 levels of gray scale. 2. Add a constant integer such as 50 to every element of this matrix. Make the histogram again and also visualize the image using `imshow`. 3. Subtract 50 from every element of this matrix. Make the histogram again and view it using `imshow`. 4. Compare the original matrix before 1. with the matrix in case 3. and comment on what an increase or decrease of brightness control does to an image. What are the other histogram operations you could think of – eg., what is contrast enhancement or gamma correction?

First, we'll import the image into a matrix, using `imread`, and then show the image using `imshow`.

```
[1]: t = imread("../Pics/octave_logo.jpg");  
      imshow(t)
```



To convert it into Grayscale, I've used the `rgb2gray` algorithm that is used by Octave, in a different library, available on SourceForge. There is a question on this topic on StackOverflow, which I have taken help from: [Implementing rgb2gray in Octave](#)

```
[2]: function y = rgbToGray(image)
      y = (0.298936 * image(:,:,1) + 0.587043 * image(:,:,2) + 0.114021 * image(
      ↪,:,3));
      endfunction;
```

```
[3]: t_grey = rgbToGray(t);
```

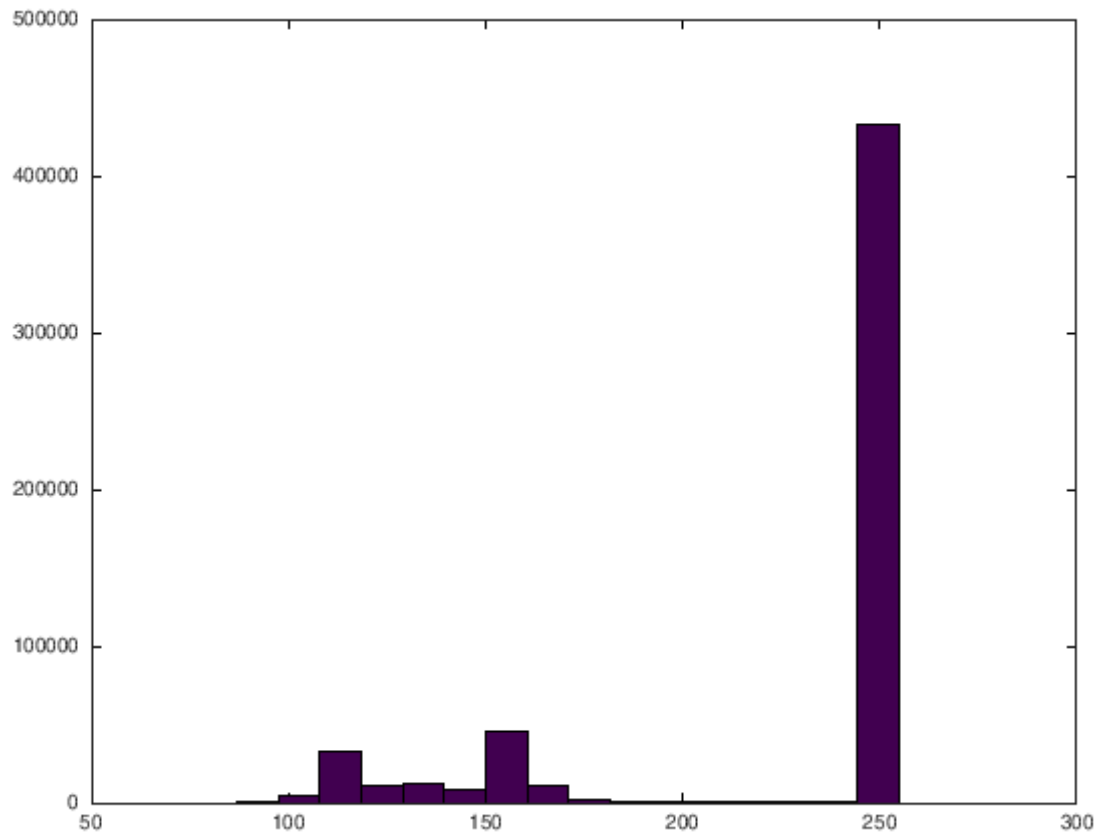
Now, we're reshaping this matrix into a single row:

```
[4]: t_grey_line = reshape(t_grey,1,size(t_grey)(1)**2);
```

We're plotting a histogram of the numbers we have, using 16 bins, with the `hist` command.

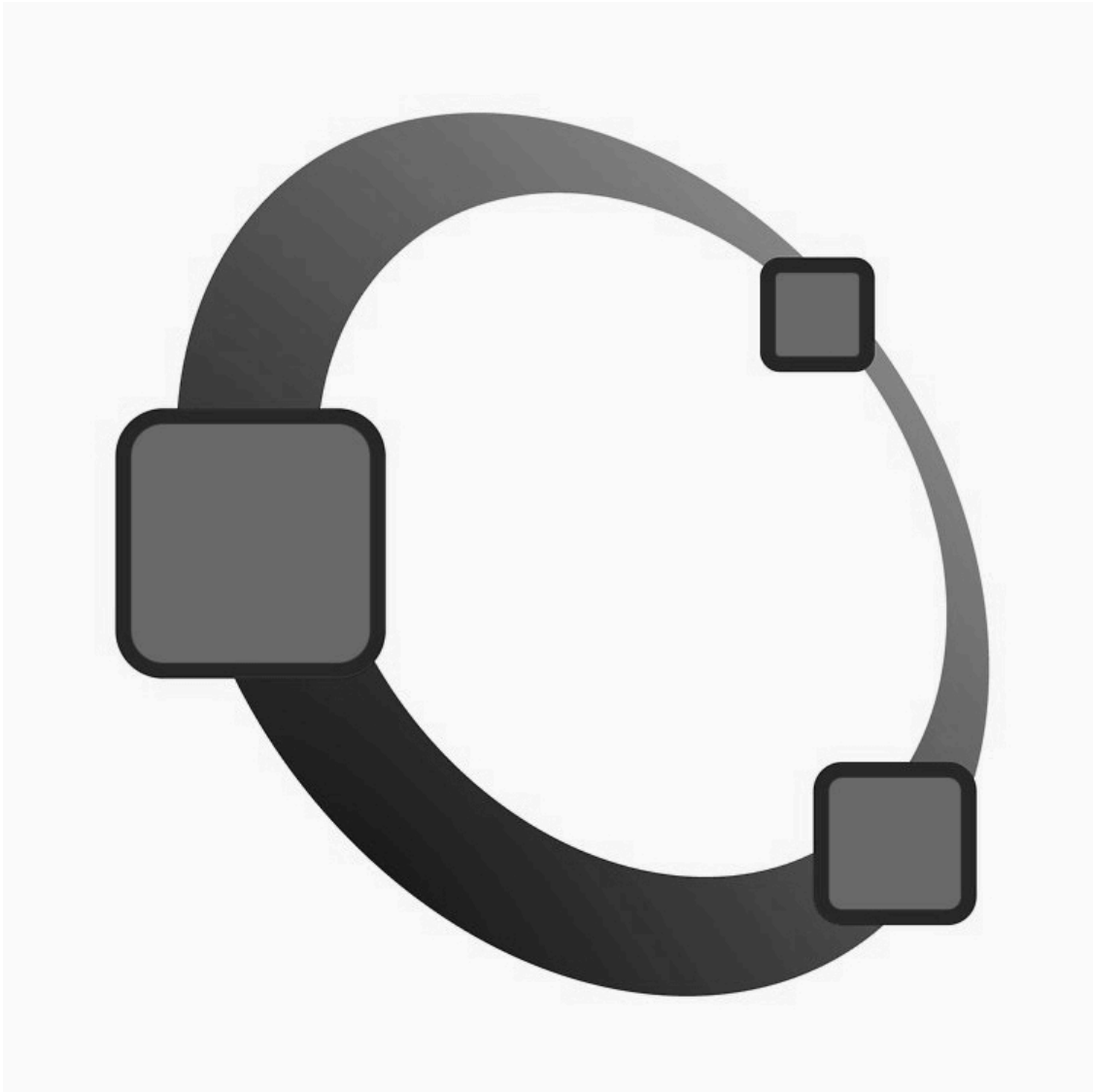
In the histogram below, we see that the most populated number is 250, which corresponds to white, and it logically makes sense.

```
[5]: hist(t_grey_line,16)
```



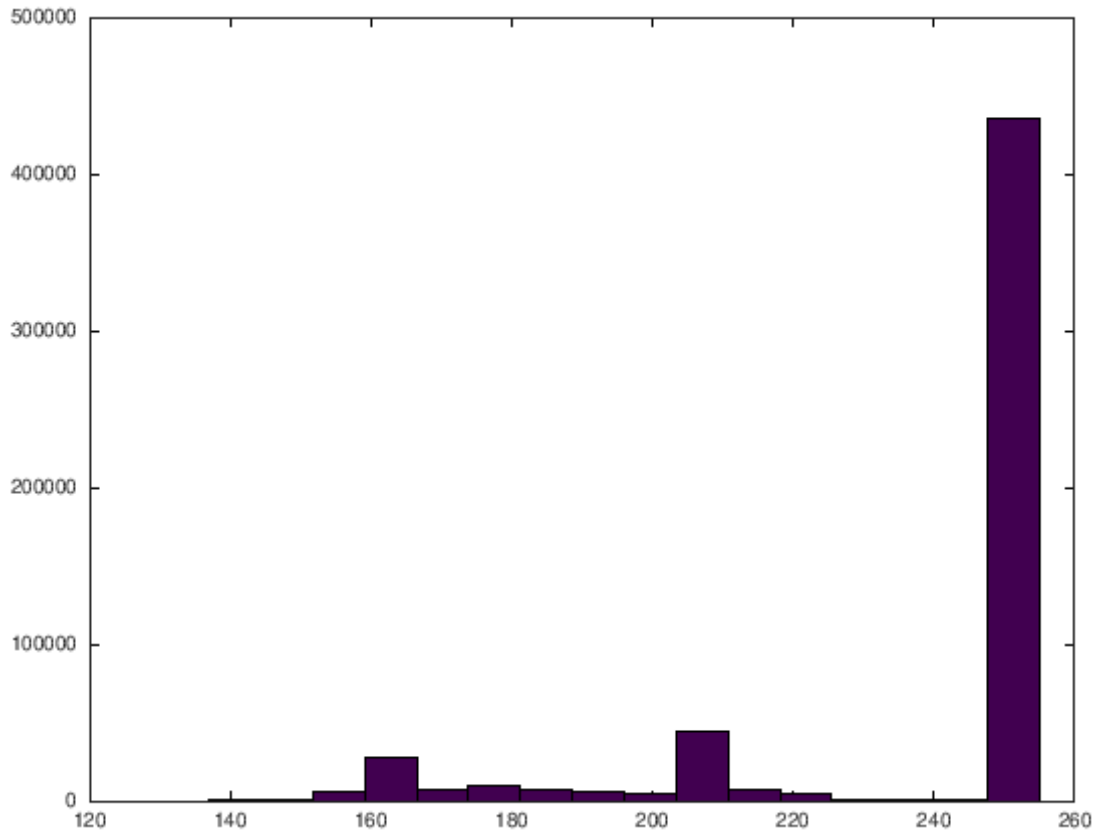
Showing the grayscale version of the image:

```
[6]: imshow(t_grey)
```



Now, adding 50 to each element, and then plotting the histogram with 16 bins:

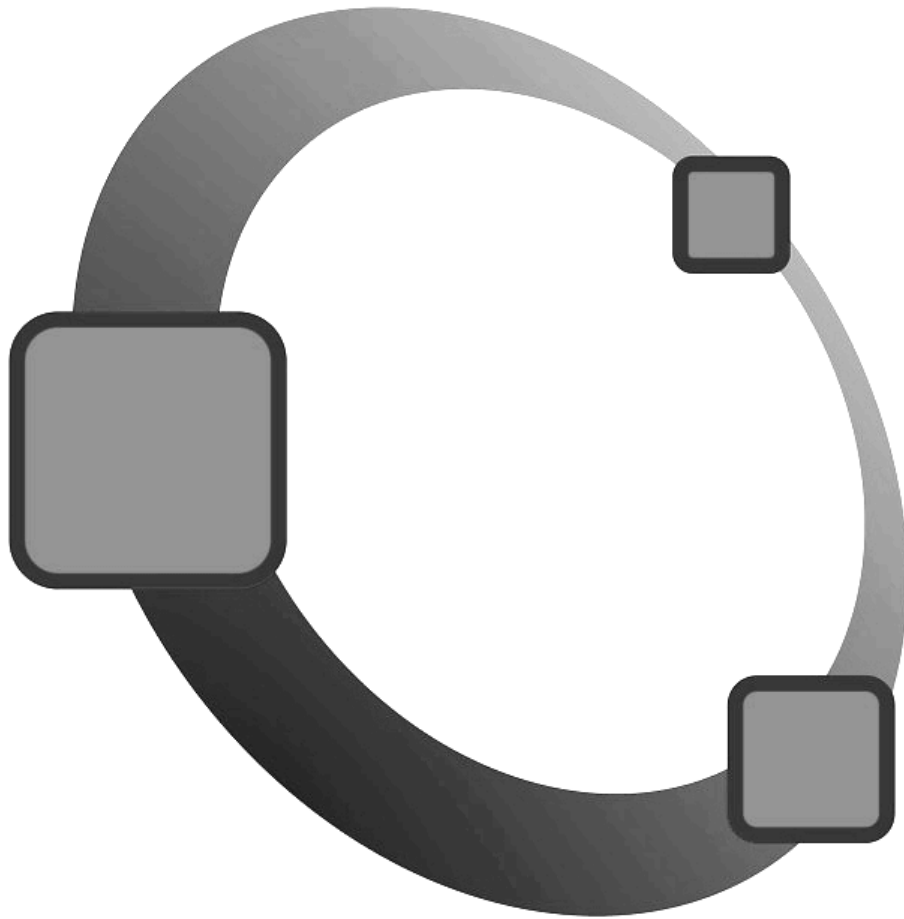
```
[7]: t_grey_added= t_grey_line.+50;  
hist(t_grey_added,16)
```



Now, we're converting the line matrix back to a 750x750 image, using the `reshape` command, and then plotting it:

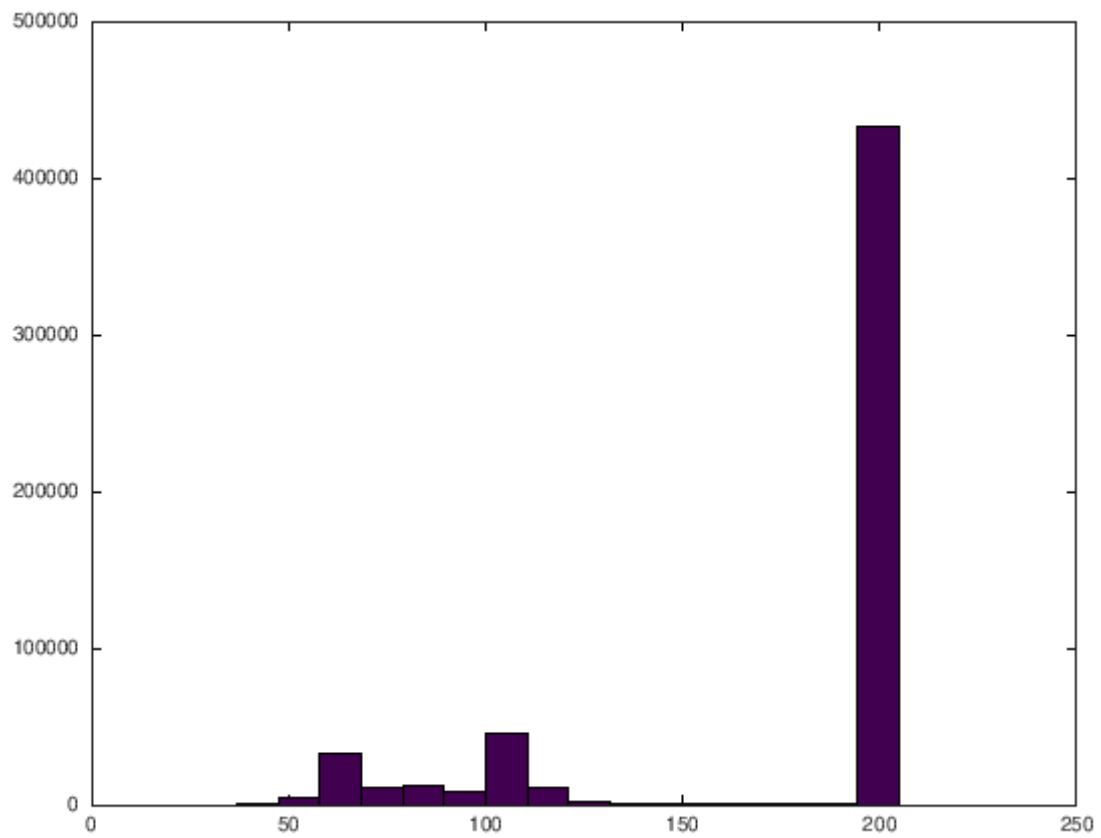
```
[8]: t_grey_bright = reshape(t_grey_added,750,750);
```

```
[9]: imshow(t_grey_bright)
```

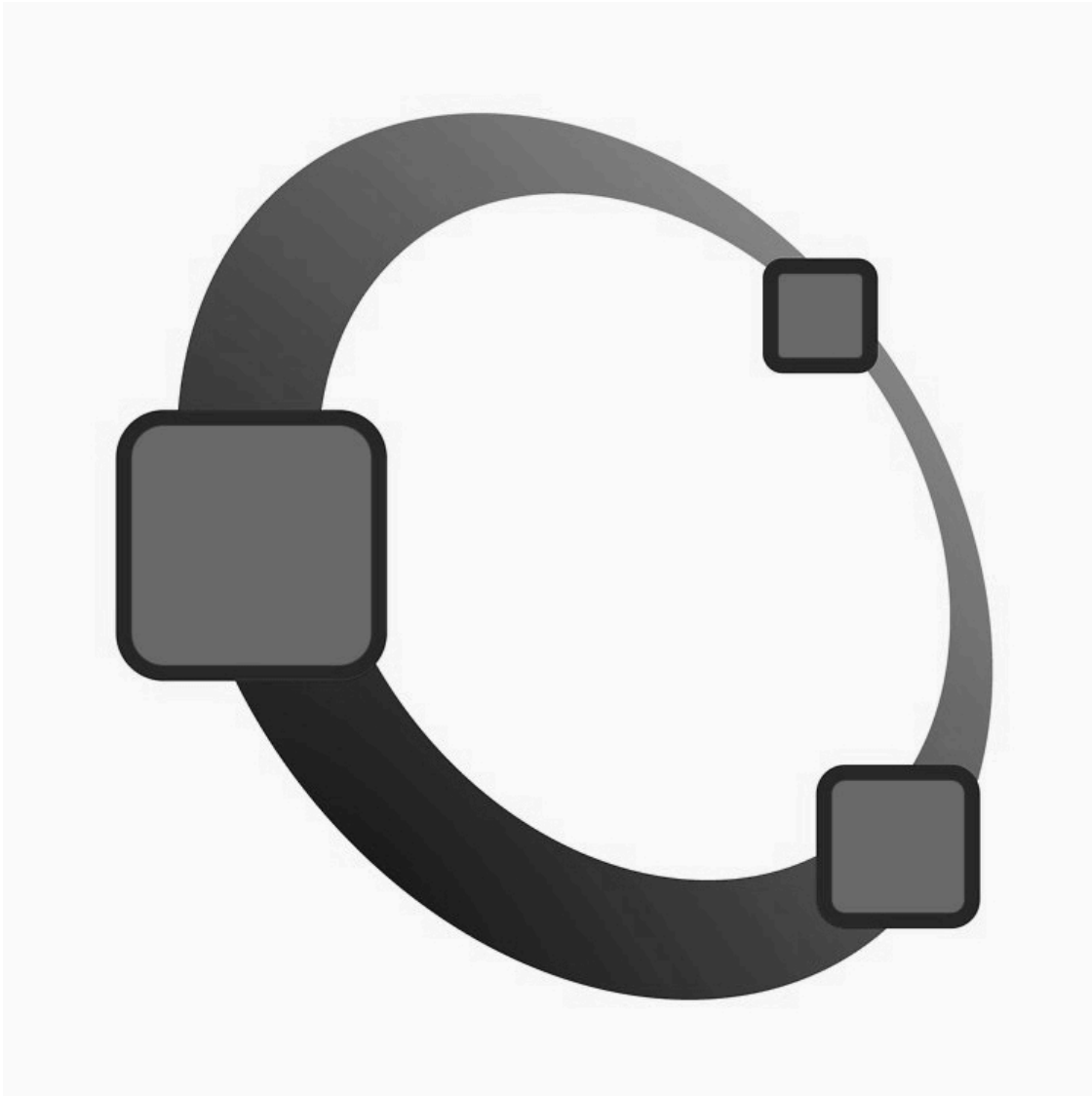


Now, we're repeating the same as above, but after subtracting 50 from each element of the matrix.

```
[10]: t_grey_subtracted = t_grey_line.-50;  
hist(t_grey_subtracted,16)
```



```
[11]: t_grey_dark = reshape(t_grey_subtracted,750,750);  
      imshow(t_grey_dark)
```



Now, we're finding the difference between the bright and the dark images:

```
[12]: difference = (t_grey_bright - t_grey_dark)(1);  
      difference
```

```
difference = 54
```

Next, the difference between the normal and the bright images:

```
[13]: difference2 = (t_grey_bright - t_grey)(1);  
      difference2
```

```
difference2 = 4
```

Finally, the difference between the normal and the dark images:


```
[14]: difference3 = (t_grey - t_grey_dark)(1);  
      difference3
```

```
difference3 = 50
```