EE420 – Digital Image Processing – Homowork 1

Question 1 - Modeling the imaging process:

The hand written answer is attached to the end of this PDF.

Question 2 - Image denoising:

1. The MATLAB code attached to this PDF (HW1_Denoising_by_Averaging.m).

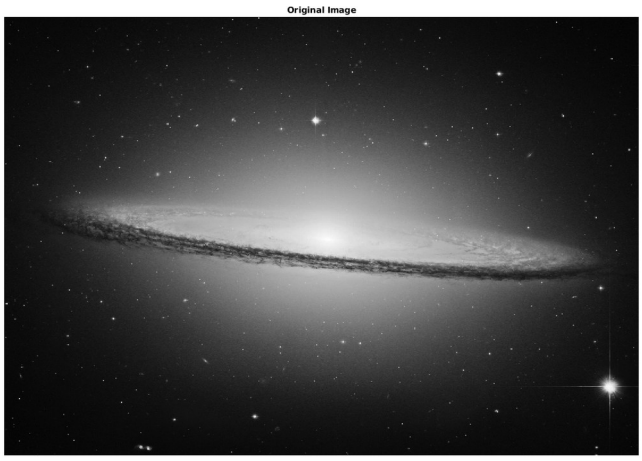
2. The largest intensity in the original image is 255 (white), and the smallest intensity is 0 (black).

3. I defined the noise to be Gaussian noise using the function randn() with zero meanand a standard deviation of 16 (line 18):

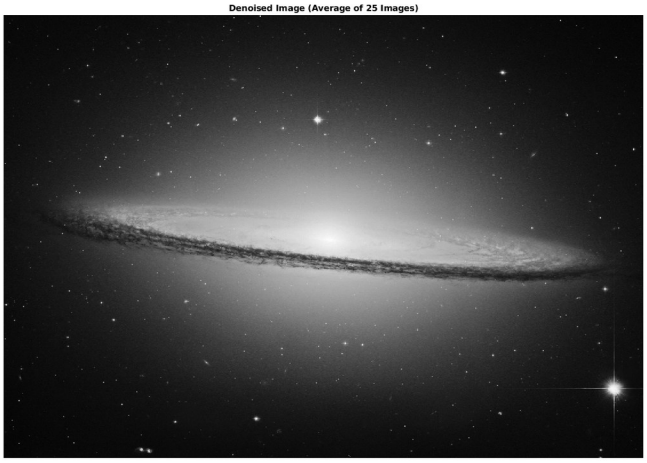
```
im_noised(:, :, i) = (sqrt(16) * randn(size(im_original))) + 0 + im_original;
```

4. We can see that the results of denoising improved with averaging more realizations/images when we look at the MSE results for averaging 25, 50, and 100 images. For averaging of 25 images, we got MSE=0.6400, for averaging of 50 images, we got MSE=0.3196, and for averaging of 100 images, we got MSE=0.1599. From those results we can see that the Mean Squared Error decreased when we increased the number of images in the averaging process.

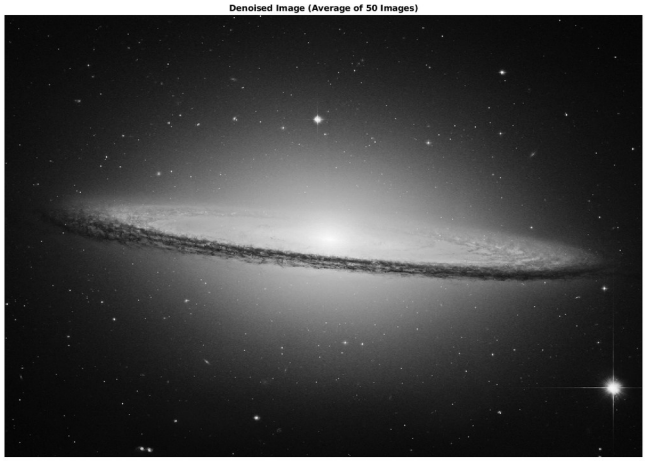
5. We can see the 4 different output images:
The original image:



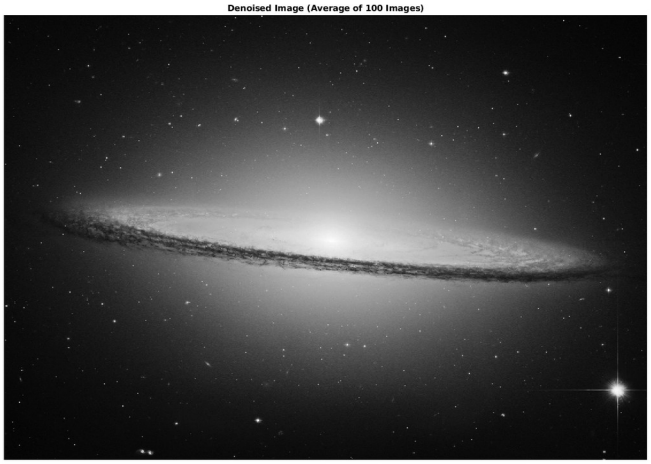
The image denoised using average of 25 images:



The image denoised using average of 50 images:



The image denoised using average of 100 images:



Question 3 - Detecting image tampering by image differencing:

1. The MATLAB code attached to this PDF (HW1_ImageTampering.m).

2. The message hidden in my print is “23-Jan-2021 14:36:21” (The time on my computer’s clock).

3. A screenshots of the original image, the tampered image, the difference image with typecasting:

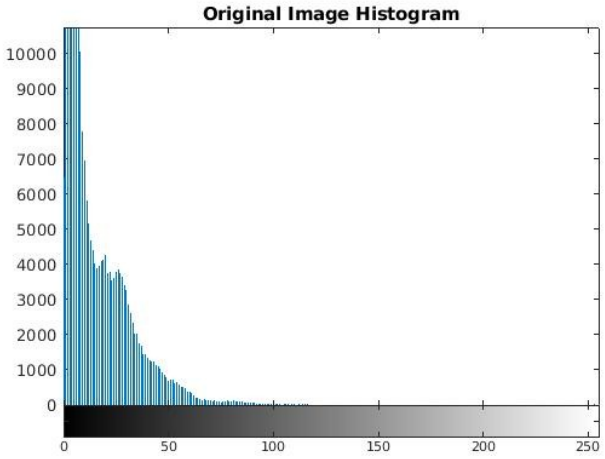
A screenshots of the difference image without typecasting:

A screenshots of the difference image with typecasting:

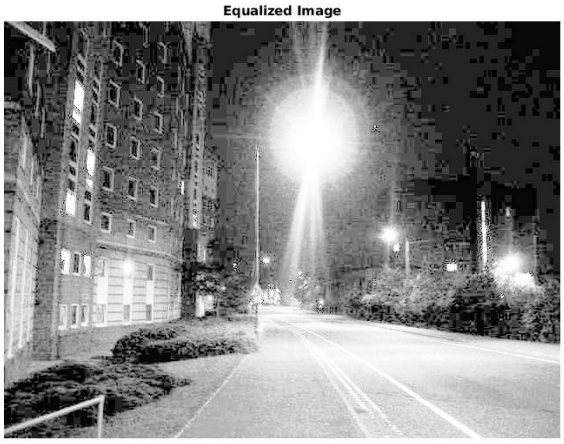
Question 4 - Histogram Equalization – Part 1:

1. The MATLAB code attached to this PDF (HW1_HistogramEqualization_Part1.m.).

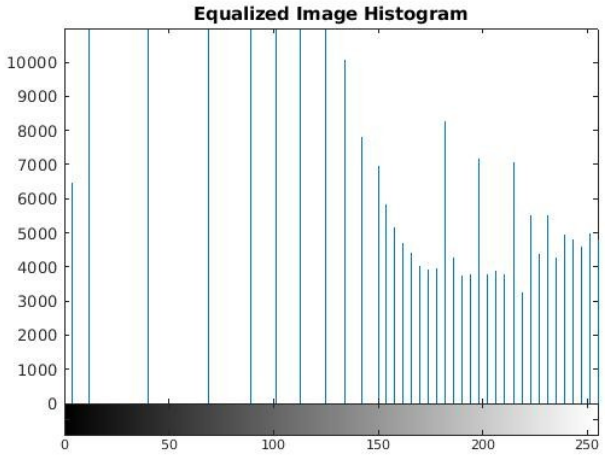
2. The histogram of the original Dark_Road image is:



3. The equalized image is:



4. The histogram of the equalized image is:



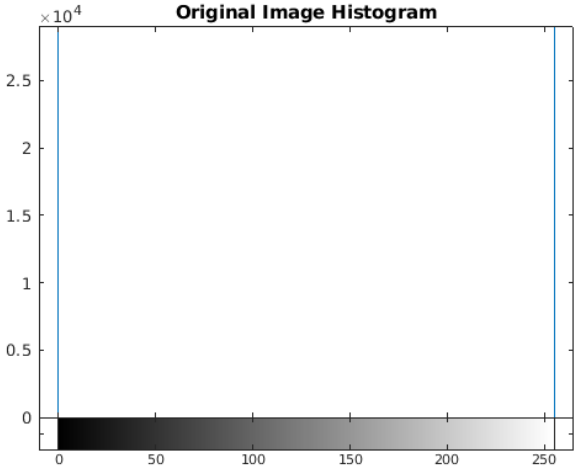
5. Desirable/undesirable enhancements in the image:

#	Desirable enhancements	Undesirable enhancements
1	We can see objects that we didn't see before, like the bushes on the road side and the building windows	The streetlights are too bright
2		Objects that we could see clearly on the original image, like the road lines and marking, are too bright now and we can't detect them
3		We can detect the pixels – the equalized image has pure quality

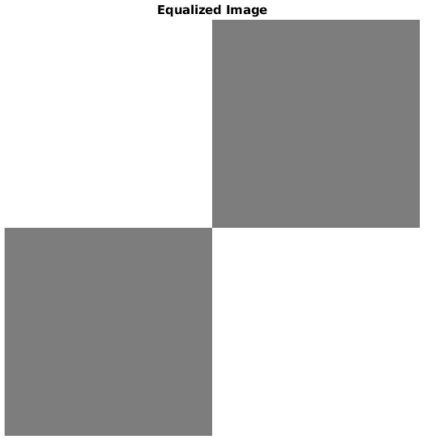
<u>Question 4 - Histogram Equalization – Part 2:</u>

1. The MATLAB code attached to this PDF (HW1_HistogramEqualization_Part2.m.).

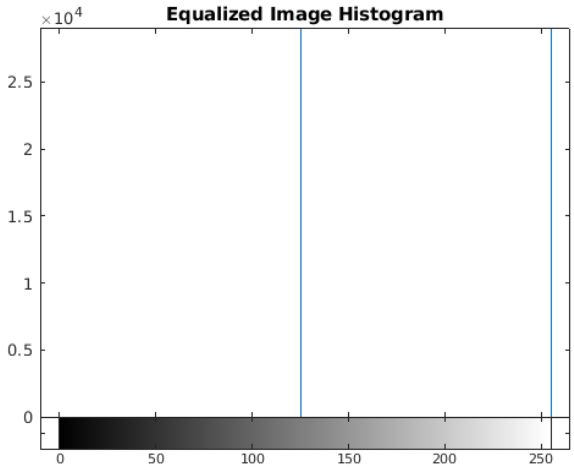
2. The histogram of the original Checkerboard image is (x axis is between -10 to 265 for better visualization of the gray levels 0 and 255):



3. The equalized image is:



4. The histogram of the equalized image is (x axis is between -10 to 265 for better visualization of the gray levels 0 and 255):



5. The white squares stayed the same, but the black squares became lighter (from 255 to 125).

6. We can see that the result make sense from looking at the mathematical expression of slide 22 (lecture 4):

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{L-1}{MN}\sum_{j=0}^{k} n_j$$

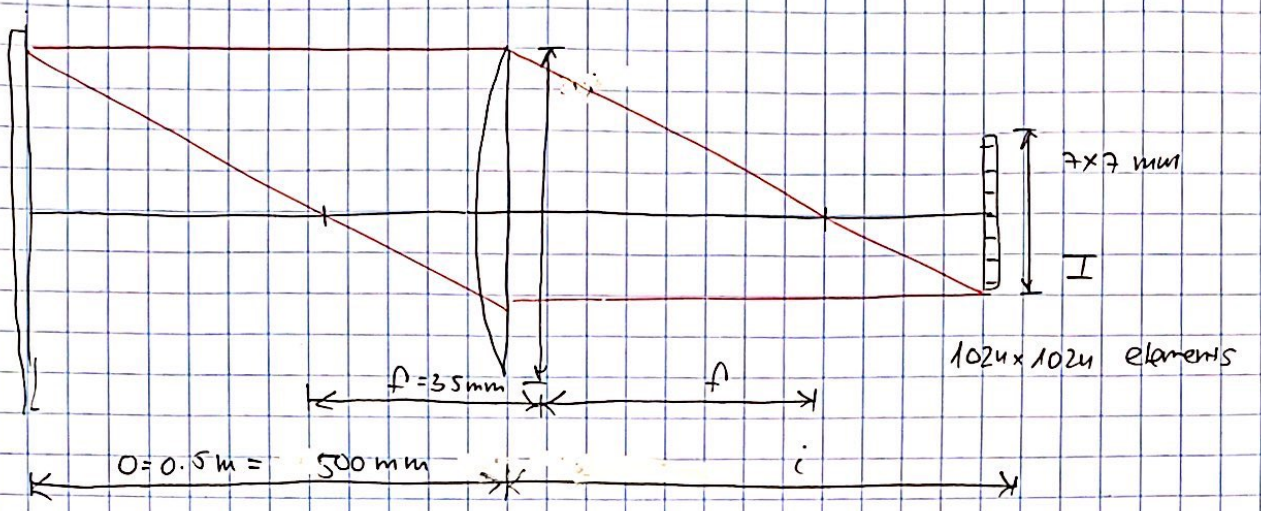
We can see here, that when we equalize the first histogram. In the original image we have 2 colors, so we will get 2 colors in the accumulate histogram (2 lines):

s_0 = 255/256 x (256/2) = 255/2 = ~127

s_1 = 255/256 x (256/2 + 256/2) = 255

EE 420 - Homework 1

Question 1 :



Thin lens formula :

$$\frac{1}{f} = \frac{1}{O} + \frac{1}{i} \quad \Rightarrow \quad \frac{1}{i} = \frac{1}{f} - \frac{1}{O} = \frac{1}{35} - \frac{1}{500} = \frac{107}{3500}$$

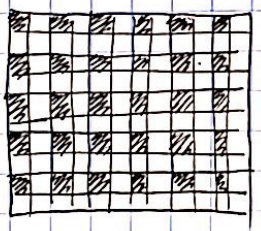
$$i = \frac{3500}{107} \text{ mm.}$$

Magnification

$$\beta = \frac{I}{O} = \frac{i}{o} \quad \Rightarrow \quad O = \frac{I_o}{i} = \frac{7 \cdot 500}{\frac{3500}{107}} = \frac{3500}{\frac{3500}{107}} = 107 \text{ mm.}$$

(one dimensional)

The camera is able to resolve $\frac{1024}{2}$ = pixels (with empty line between every line of pixels)



In 2D : 512 x 512 pixels (elements)

So, in 1D, we have 512 pixels over 107 mm

$\Rightarrow$ ~ 4.78 pixels per mm.

line pairs