CPS621 Winter2022
Lab01 Report
Name: Tusaif Azmat, Student#: 500660278.

Part 1: Noise reduction by image averaging:
1.1:


← Original Image I1

1.2:


←Gray Image I2

1.3:

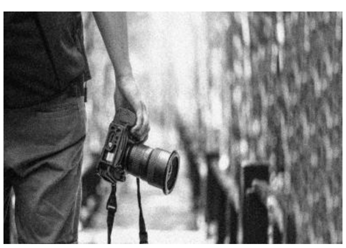
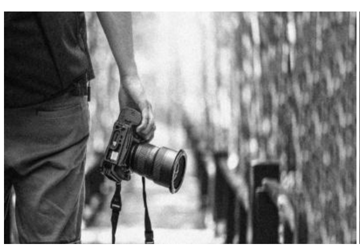
   

Image average for 2 | Image average for 5 | Image average for 10 | Image average for 30

The above 4 images shows the noise reduction results by taking the average of any 2, 5, 20, and 30 images from N1 to N30 noise-contaminated images. I have noticed the more noise in images with less number of images while averaging.

1.4:

The Gaussian assumption makes the added noise easy to remove. The mean of a standard Gaussian is 0.

So,

$(I + Noise_1) + (I + Noise_2) + ... + (I + Noise_n) / n$

$\Rightarrow (n * I) + (Noise_1 + Noise_2 + ... + Noise_n) / n$

$\approx> I + 0$

The above experiment proof that the averaging of noise-contaminated images gives the results same as the original gray scaled. Actually it made the original image sharper.
Gaussian noise addition provide extra white light to the image and the random white noise to many images and then averaging out the image provides extra quality to the image.

Part 2: Power-law transformation

2.1:

Log Transformation: The general form of the log transformation is s = c * log (1 + r). The log transformation maps a narrow range of low input grey level values into a wider range of output values. The inverse log transformation performs the opposite transformation. Log functions are particularly useful when the input grey level values may have an extremely large range of values. s = log (1 + r), as we usually set c to 1. Grey levels must be in the range [0.0, 1.0]. Hence, log transformation provides images to transform to reveal more details.

Power law transformation: The nth power and nth root curves shown in fig. A can be given by the expression, $s = cr^{\gamma}$ this transformation function is also called as gamma correction. For various values of **γ** different levels of enhancements can be obtained. This technique is quite commonly called as Gamma Correction.

The difference between the log transformation function and the power-law functions is that using the power-law function a family of possible transformation curves can be obtained just by varying the γ (gamma value). These are the two basic image enhancement functions for grey scale images that can be applied easily for any type of image for better contrast and highlighting. Using the image negation formula given above, it is not necessary for the results to be mapped into the grey scale range [0, L-1]. Output of L-1-r automatically falls in the range of [0, L-1]. But for the Log and Power-Law transformations resulting values are often quite distinctive, depending upon control parameters like γ and logarithmic scales. So the results of these values should be mapped back to the grey scale range to get a meaningful output image. For example, Log function s = c log (1 + r) results in 0 and 2.41 for r varying between 0 and 255, keeping c=1. So, the range [0, 2.41] should be mapped to [0, L-1] for getting a meaningful image.

2.2:

Original gray Image (I2)

Image T1 Gamma value 0.3

Image T2 Gamma value 3

2.3:

After the power law transformation of the original image, I have notice that with smaller γ (gamma) (0.03) values compress higher intensities and stretch the lower intensities. As can be seen in image two above that it has more light as the image got compressed with higher intensity values and starched the lower intensity.

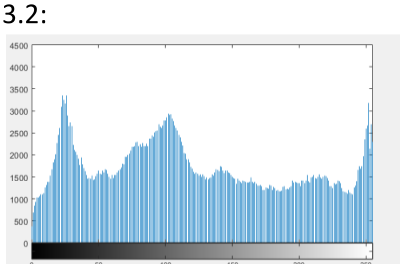
With gamma value 3 the image is darker as can be seen above because the high γ (gamma) (3) values stretch high-intensity levels and compress the low-intensity levels.

Hence, with proper values of γ (gamma) we can make the image properly observe for further clarifications.

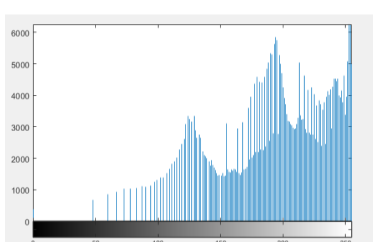
3.1:

Histogram equalization is a common technique for enhancing the appearance of images. The basic principle of histogram equalization provides with the method of contrast adjustment using the image histogram and it fixes the contrast of image under observation. It is a process of spreading out the most frequent intensity values by effectively stretching out the intensity of the image that improves the overall contrast of the image.

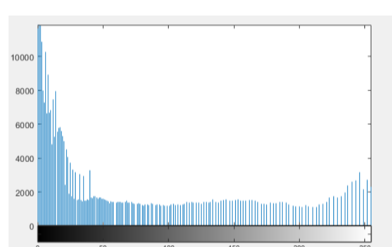
3.2:



I2 Histogram

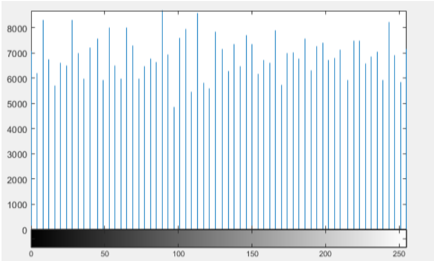


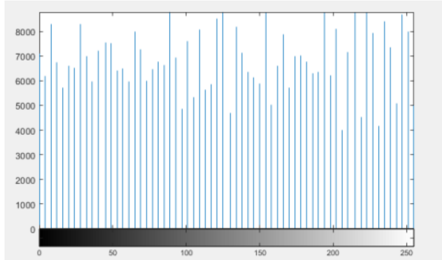
T1 Histogram

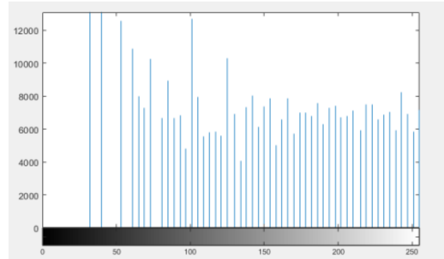


T2 Histogram

3.3:


I2 histogram-equalized


T1 histogram-equalized

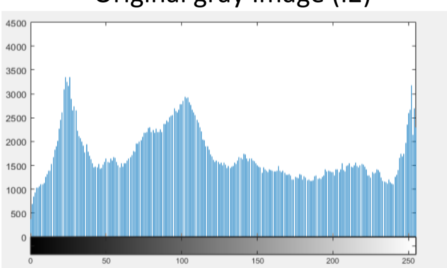

T2 histogram-equalized

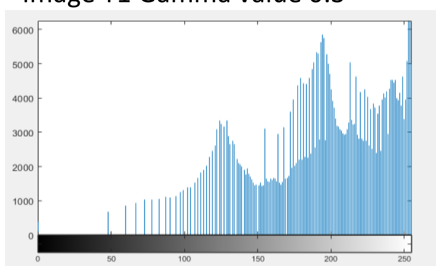
3.4:

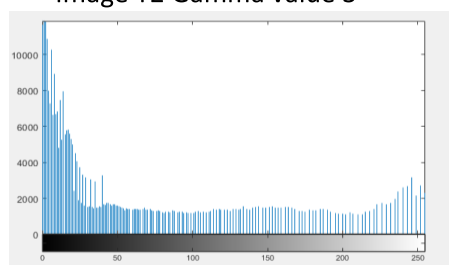

Original gray Image (I2)


Image T1 Gamma value 0.3


Image T2 Gamma value 3


I2 Histogram

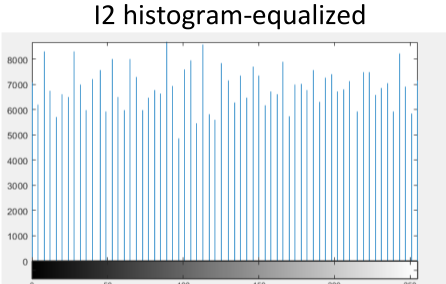

T1 Histogram

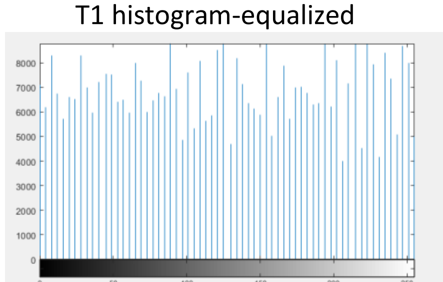

T2 Histogram

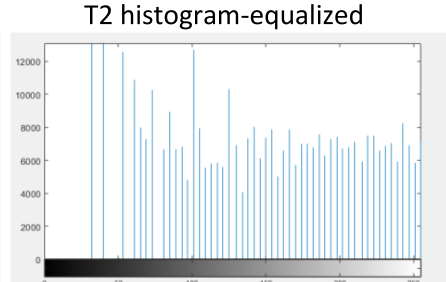

I2 histogram-equalized


T1 histogram-equalized


T2 histogram-equalized


I2 histogram-equalized


T1 histogram-equalized


T2 histogram-equalized

As can be see above the histogram equalization technique has improved the contrast of the above three images quite well and results of histogram-equalization significant. Histogram equalization has enhanced the appearance all the three images.
We have used three images which are original, light and predominantly dark. The histograms of all three images are skewed towards the lower end of the grey scale and all the image detail is compressed into the dark end of the histogram. Histogram equalization `stretch out' the grey levels and make images much smoother and also for the dark images to produce a more uniformly distributed histogram that the image would become much clearer.
Hence, Histogram equalization is a transformation that stretches the contrast by redistributing the gray-level values uniformly.

MatLab Code:

```
%############################################
% CPS 621 Winter2022
% Lab01
% Name: Tusaif Azmat Student#: 500660278.
%############################################


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Part 1. Noise reduction by image averaging
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%1.1
I1 = imread("C:\Users\Zanara\Documents\Ryerson\Winter2022\CPS621\CPS621_winter2022\Labs\lab01\AdobeStock_01.jpeg");
figure, imshow(I1);
I2 = im2gray(I1);
figure, imshow(I2);

%1.2
N1 = imnoise(I2, 'gaussian');N2 = imnoise(I2, 'gaussian');N3 = imnoise(I2, 'gaussian');N4 = imnoise(I2, 'gaussian');
N5 = imnoise(I2, 'gaussian');N6 = imnoise(I2, 'gaussian');N7 = imnoise(I2, 'gaussian');N8 = imnoise(I2, 'gaussian');
N9 = imnoise(I2, 'gaussian');N10 = imnoise(I2, 'gaussian');N11 = imnoise(I2, 'gaussian');N12 = imnoise(I2, 'gaussian');
N13 = imnoise(I2, 'gaussian');N14 = imnoise(I2, 'gaussian');N15 = imnoise(I2, 'gaussian');N16 = imnoise(I2, 'gaussian');
N17 = imnoise(I2, 'gaussian');N18 = imnoise(I2, 'gaussian');N19 = imnoise(I2, 'gaussian');N20 = imnoise(I2, 'gaussian');
N21 = imnoise(I2, 'gaussian');N22 = imnoise(I2, 'gaussian');N23 = imnoise(I2, 'gaussian');N24 = imnoise(I2, 'gaussian');
N25 = imnoise(I2, 'gaussian');N26 = imnoise(I2, 'gaussian');N27 = imnoise(I2, 'gaussian');N28 = imnoise(I2, 'gaussian');
N29 = imnoise(I2, 'gaussian');N30 = imnoise(I2, 'gaussian');

%1.3
IA1 = (double(N1)+double(N30))/2;
IA1 = uint8(rescale(IA1, 0, 255));
figure, imshow(IA1);

IA2 = (double(N5)+double(N10)+double(N15)+double(N20)+double(N25))/5;
IA2 = uint8(rescale(IA2, 0, 255));
figure, imshow(IA2);

IA3 = (double(N2)+double(N3)+double(N4)+double(N6)+double(N7)+double(N8)+double(N9) ...
    +double(N11)+double(N12)+double(N13)+double(N14)+double(N16)+double(N18)+double(N19) ...
    +double(N21)+double(N22)+double(N23)+double(N26)+double(N27)+double(N28))/20;
IA3 = uint8(rescale(IA3, 0, 255));
figure, imshow(IA3);

IA4 = (double(N1)+double(N2)+double(N3)+double(N4)+double(N5)+double(N6)+double(N7)+double(N8)+double(N9)+double(N10) ...
    +double(N11)+double(N12)+double(N13)+double(N14)+double(N15)+double(N16)+double(N17)+double(N18)+double(N19) ...
    +double(N20)+double(N21)+double(N22)+double(N23)+double(N24)+double(N25)+double(N26)+double(N27)+double(N28) ...
    +double(N29)+double(N30))/30;
IA4 = uint8(rescale(IA4, 0, 255));
figure, imshow(IA4);
```

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Part 2. Power-law transformation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%2.2
T1 = imadjust(I2,[],[],0.3);
T2 = imadjust(I2,[],[],3);

figure, imshow(I2);

figure, imshow(T1);
figure, imshow(T2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Part 3. Histogram equalization
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%3.2
imhist(I2);
imhist(T1);
imhist(T2);
%3.3
figure, imshow(I2);
figure, imshow(T1);
figure, imshow(T2);

HeI2 = histeq(I2);
HeT1 = histeq(T1);
HeT2 = histeq(T2);

figure, imshow(HeI2);
figure, imshow(HeT1);
figure, imshow(HeT2);

imhist(HeI2);
imhist(HeT1);
imhist(HeT2);
```