Report : Histogram Equalization

Algorithm:

The **histogram equalization algorithm** is a computer image processing technique that is used to enhance the contrast of an image. It works by effectively redistributing the most frequent pixel values, i.e stretching out the intensity range, in the image to produce a more uniform distribution. The basic steps involved in histogram equalization algorithm are as follows:

1. Create the input image histogram: Histogram represents the distribution of pixel values in the image. It is computed by counting the number of pixels in the image that have each possible intensity value.
2. Compute the cumulative distribution function (CDF): Next, the cumulative distribution function (CDF) of the histogram is calculated. The CDF is computed by summing up the histogram values up to the current intensity level.
3. Normalize the CDF: The CDF values are then normalized by dividing each value by the total number of pixels in the image. This produces a probability distribution function (PDF) that represents the probability of a pixel having a specific intensity value.
4. Compute the new pixel values: The next step is to compute the new pixel values for each pixel in the image. This is done by multiplying the normalized CDF value for the current pixel intensity by the maximum possible pixel value (typically 255 for an 8-bit image). The result is rounded to the nearest integer to obtain the new pixel value.

[In my code for histogram equalization normalization(step 3) and multiplying the resulting value by the maximum intensity(step 4) is combined in one step, so that the only step remaining is getting the new pixel values using this.]

5. Create the new image: Finally, the new image is created by replacing each pixel in the original image with its corresponding new pixel value.

The lighting correction algorithm adjusts the bright pixels on the histogram equalized image to make the lighting more even throughout the image.

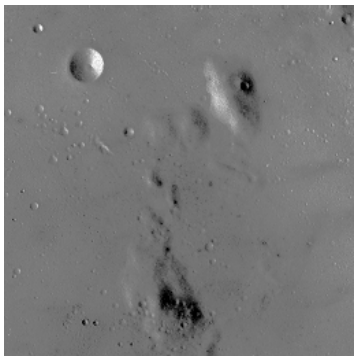
1. Flatten the input image: input pixel values are flattened to a 1D
2. Calculate the weights for a plane fit by calculating pseudo inverse [weights, x = A -1 * t].
3. Then use these weights to calculate new intensity values for each pixel in the flattened array.
4. Reconstructs the image by reshaping the new intensity values into the original image dimensions.

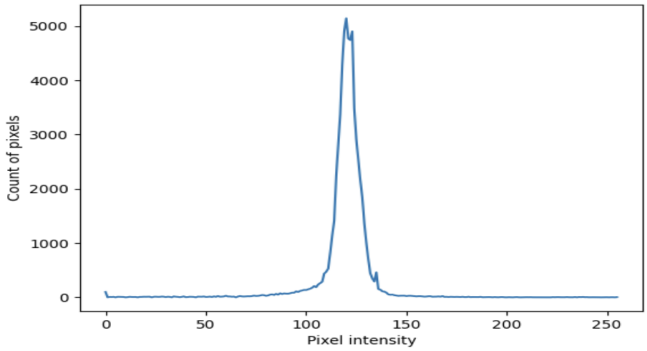
Results & Analysis:

The histogram equalization was able to successfully enhance the contrast of the input image by redistributing the pixels. **histo_equalization()** was able to compute & plot input image histogram, compute the CDF, normalize, compute output image, and compute & plot output image histogram.

The lighting correction function **light_correction()** was able to make the lighting more even on the histogram equalized image

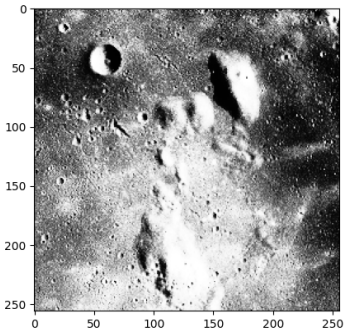
Initially the **input image** and input image **histogram** looks like this -

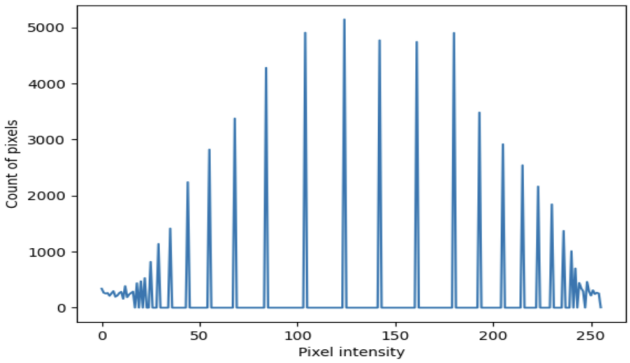




We can see from the input histogram that the most frequent pixels are concentrated in probably the **120-140 intensity region**. To enhance the image's contrast, the above mentioned histogram equalization technique is applied, and it spreads out the most frequent pixel intensity values or stretches out the intensity range of the image.

Finally we get the following **output** and **output histogram**.





From these we can see that the output image contrast is enhanced, and the histogram has been equalized.

After **lighting correction,** the final image obtained is below