For my results, you may run ***main.py*** via a python IDE, but I would recommend running ***main.ipynb***. When you do, you can test the functions ***original('moon.bmp')*** and ***final('moon.bmp')*** which will return the original image from the file 'moon.bmp' and the equalized result of the image, respectively. Any image can be accepted as the input, but the result will be a grayscale copy.

---

Histogram Equalization is a common approach to **adjust the pixel intensities of an image** so that they are distributed more **uniformally**. While traditional histograms tend to have normal distributions, equalization tends to result in uniform distributions. The end result of the adjustment is increased contrast when the intensities are more level, and not so focused in one region. Many of the definitions from my code run various steps along the pathway to equalize, but the result you are looking for is in the **final()** function, which takes in a file name as an input parameter, such as 'moon.bmp'. Any image can be taken in as an input. I begin by reading the image as an array, and transform that array into a histogram, such as in **Figure 1** below. This allows me to view the intensities of each pixel and notice that they are quite concentrated in the middle of 0 and 255 (ie. a very gray image). I can count the sum of pixels per intensity, such as in **Figure 2**. The **transfer function** Prof. Wu showed us in class $T(r_k) = \Sigma_{(j=0 \text{ to } k)}P(r_j)$, is used for calculating the cumulative mass distribution, which sums the intensities from our image (shown in **Figure 3**). **Figure 4** represents the next step that requires normalizing the intensities away from 0 - 60,000, to 0 - 255 again. Now that the values are normalized to 0 - 255, we need to restore these intensities into the array we created from the image. The histogram we create from this new, normalized array is represented in **Figure 5**. A simple Image.fromarray() function transforms our new array back into the result we're looking for (don't forget to reshape first!). I did not finish correcting for lighting. I ran into errors while adjusting the plane of the intensities toward a flat plane. Now I'm sure that if we had more noise in the original image, we would encounter much more problems and less clarity. Results are on the following page, along with **Figure 6**, the equalized image:

.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.

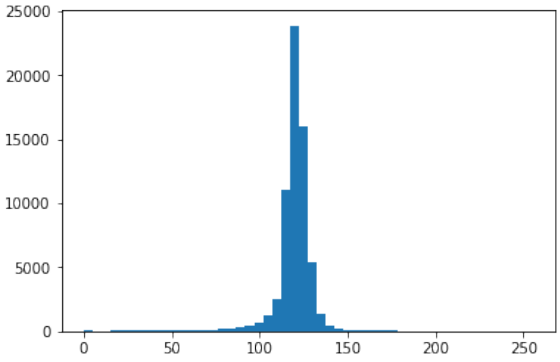


Figure 1 – Histogram of original image

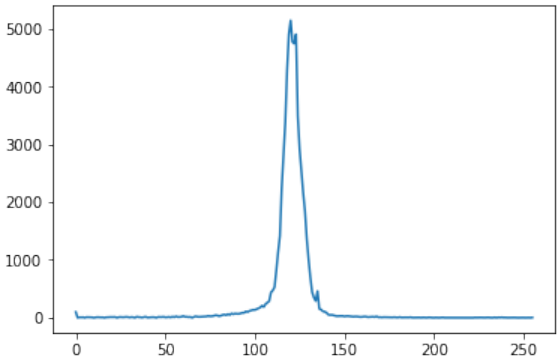


Figure 2 – Pixel sum per intensity

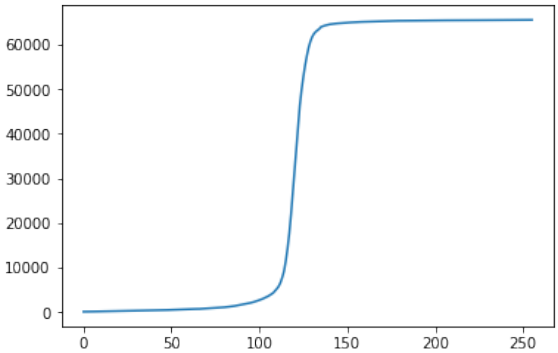


Figure 3 – Cumulative sum of intensities distribution

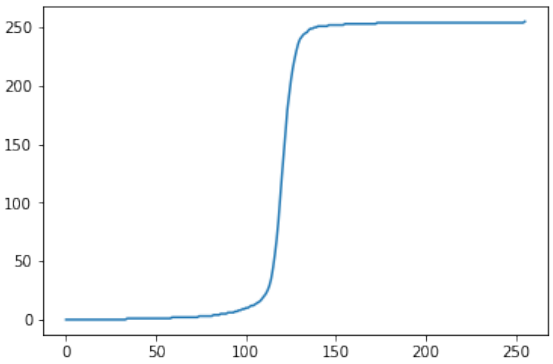


Figure 4 – Normalized intensity values to 0 - 255

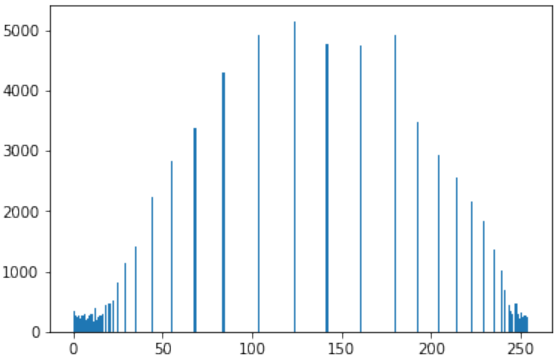


Figure 5 – Histogram of new, normalized result of original image

Figure 6 – Final result of histogram equalization