MP3 - Histogram Equalization

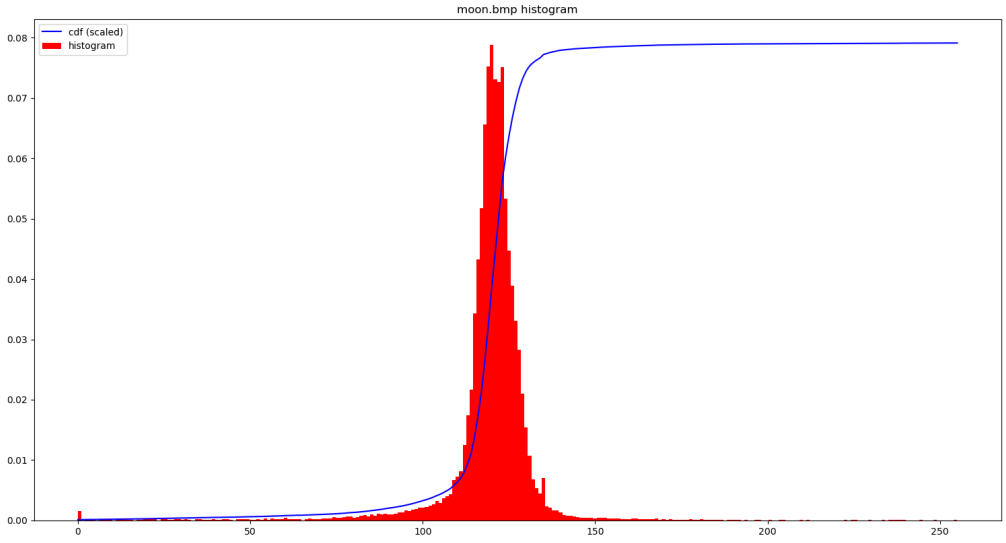
Logan Boswell

Overview

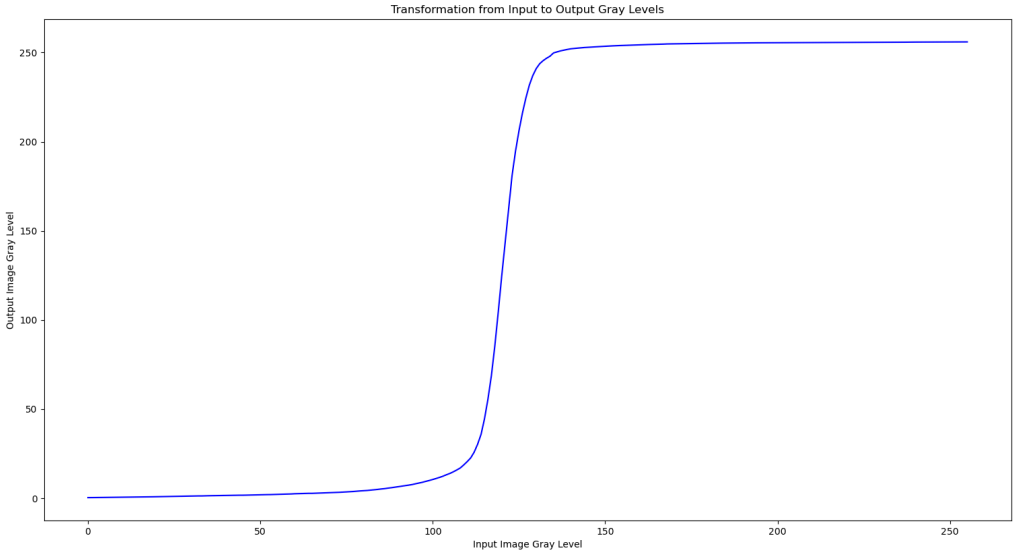
This MP is focused on implementing a histogram equalization algorithm to adjust the contrast of an image. In order to implement the algorithm, a function was written that takes in an image as an argument and performs the algorithm.

Algorithm Description

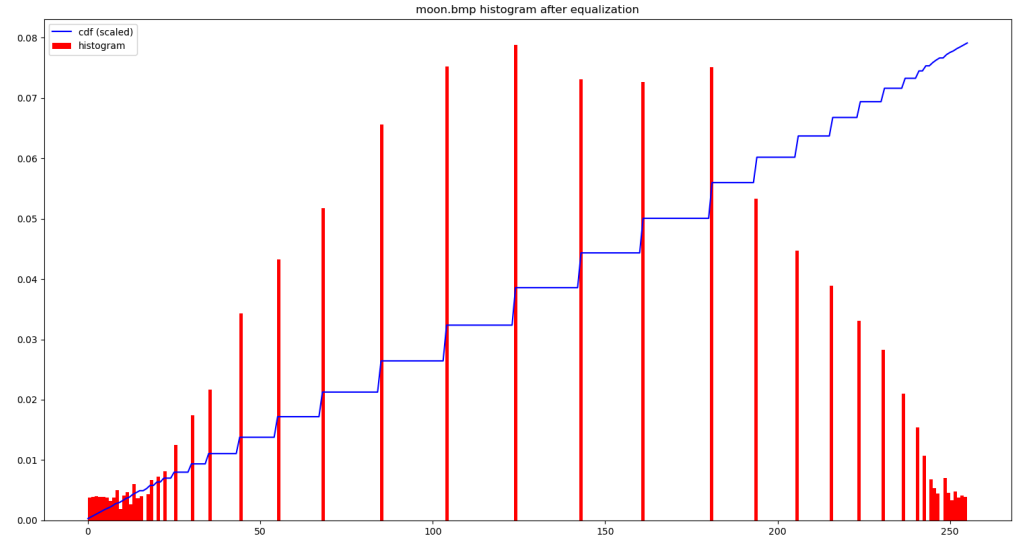
For my histogram equalization function, the first step taken is to convert the input image to a one-channel grayscale image and determine the number of rows and columns. From here, a histogram is generated based on the grayscale image pixel values using the np.histogram() function with the optional density argument set to true so the max value of the cumulative distribution function (cdf) of the histogram will be equal to 1. The cdf of the histogram is calculated by using the np.cumsum() function with the generated histogram from np.histogram(). The histogram with its cdf is shown in the plot below. One important note is that the max value of the cdf is scaled solely for this plot so it overlays nicely with the histogram.



From here, a transformation between the gray levels of the input and output images is determined based on cdf. The transform is given by the 256-element cdf array multiplied by the max gray pixel value of 255. A plot of the transform between gray levels is shown below.



From here, the transform is applied to every pixel in the image. For each pixel, the intensity is indexed from the image and used to index the transform array from above to obtain the new intensity, which replaces the current pixel intensity. To show the difference made after the transform is applied to every pixel in the image, a plot of the new histogram and cdf is below.



Lastly, the image is converted back to a 3-channel image and returned.

Results

Here are the testing results with the moon.bmp image. The image on the left is the original image and the image on the right is the image after the histogram equalization is applied.