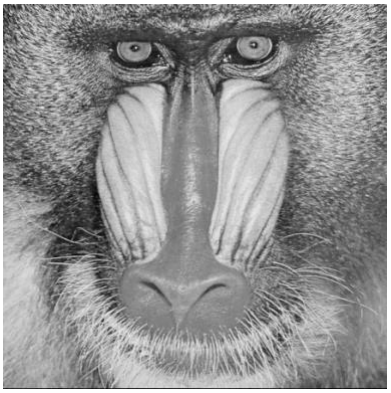
HW4 Report

Histogram Modification

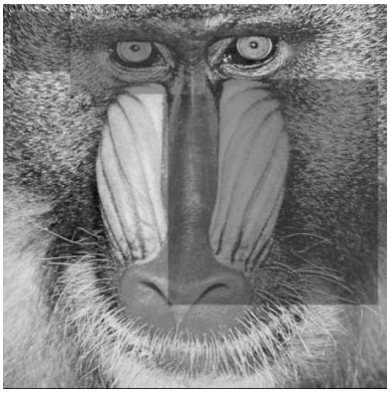
Histogram Stretching

Histogram stretching function stretches the histogram of the input image into a specific range. Due to the parameter file shows the function does not take any user input, the stretched range is default to 100 to 200.

Examples:



Original Image



Stretched Image

Original Image

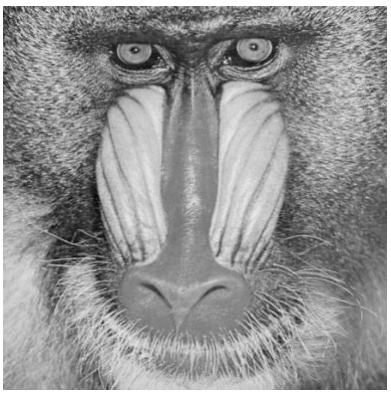
Stretched Image

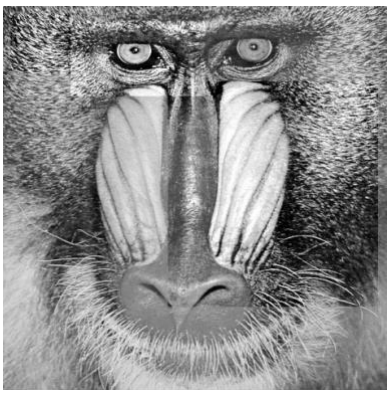
Histogram stretching flats the histogram curve of the region of interest and also constraints the pixel intensities in the stretched range.

Histogram Equalization

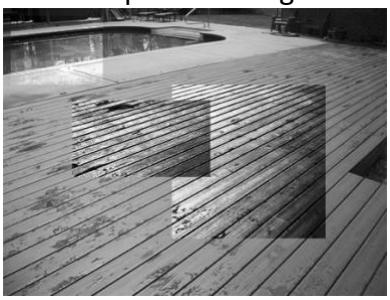
Histogram equalization function is similar to histogram stretching, except it stretches the histogram to the full range which is from 0 to 255.

Examples:





Original Image Equalized Image

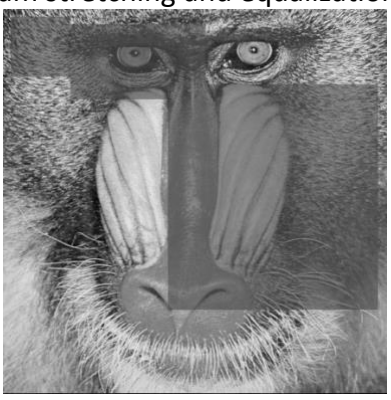
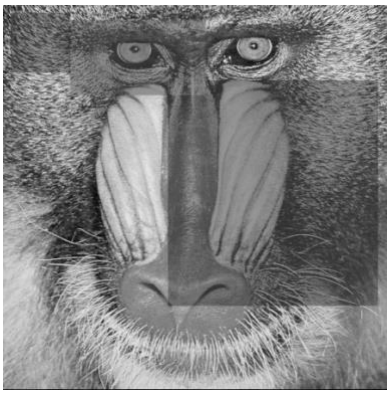



Original Image Equalized Image

Histogram equalization flats the histogram of the regions of interest and also stretches it to the full range from 0 to 255. This increases the contrast of the image.

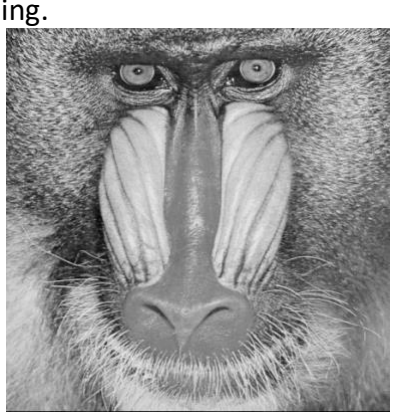
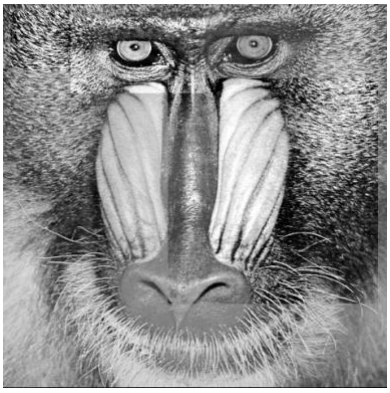
Comparison

Comparing the manual histogram stretching and histogram equalization with OpenCV histogram stretching and equalization.

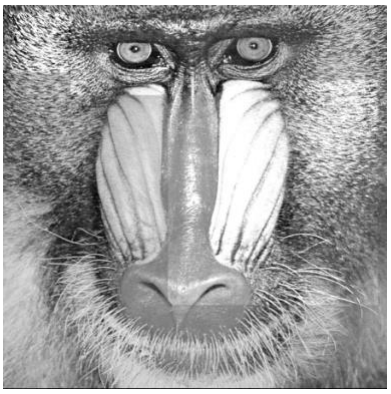
Manual Stretch OpenCV Stretch

The histogram stretching produce similar results between manual stretching and OpenCV stretching.

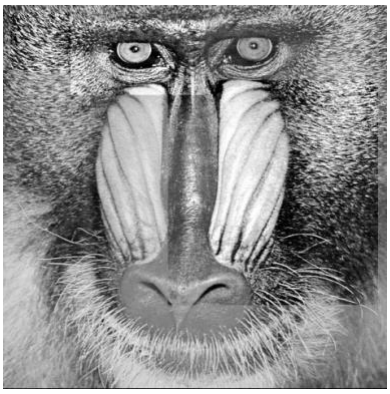



Manual Equalization OpenCV Equalization

The histogram equalization produces different result, most likely due to outlier pixels. When using 80% alternate equalization which ignores some of the outlier pixels, the result becomes closer to OpenCV result.



Alternate Equalization

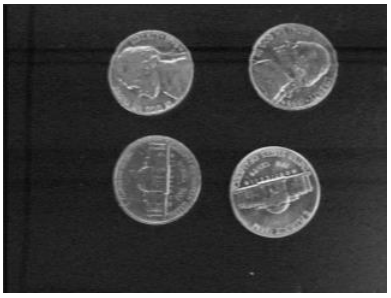


OpenCV Equalization

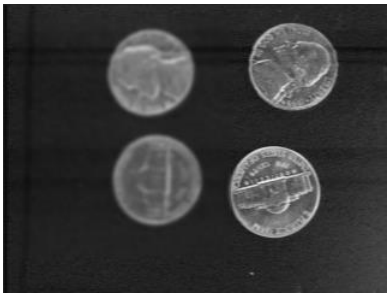
Hough Transform

Hough transform function uses OpenCV to do Hough transform for circles in the regions of interest.

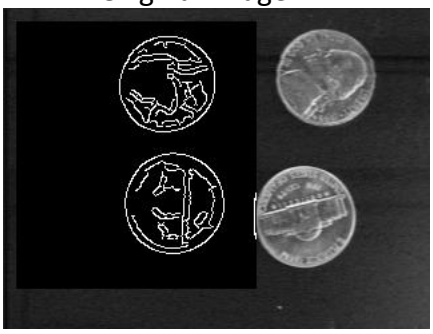
Examples:



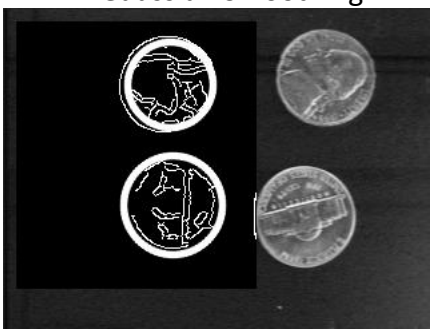
Original Image



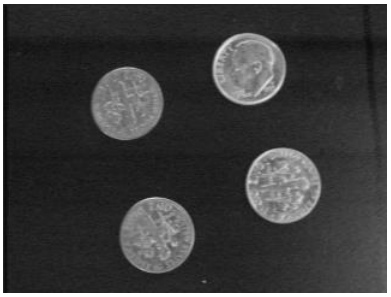
Gaussian Smoothing



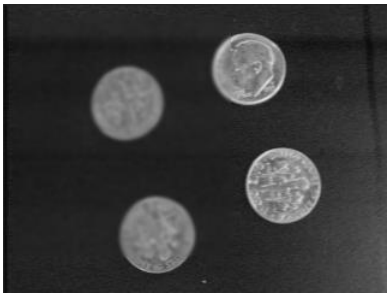
Canny Image



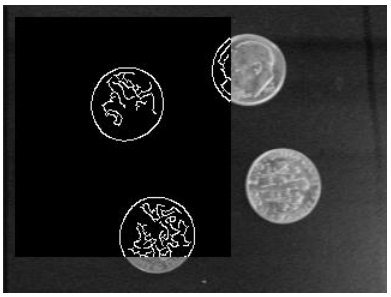
Hough Circle



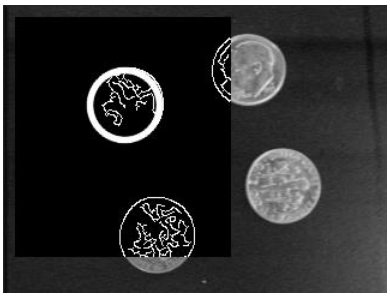
Original Image



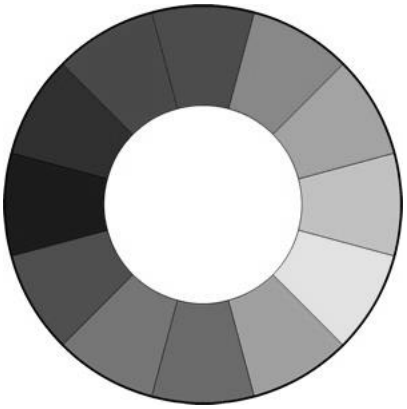
Gaussian Smoothing



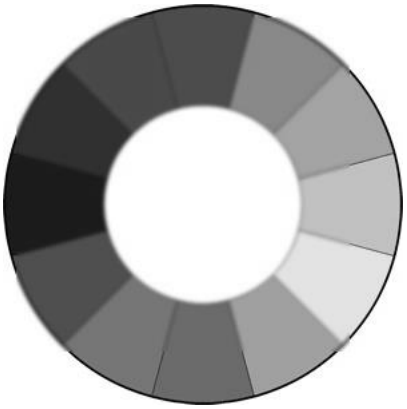
Canny Image



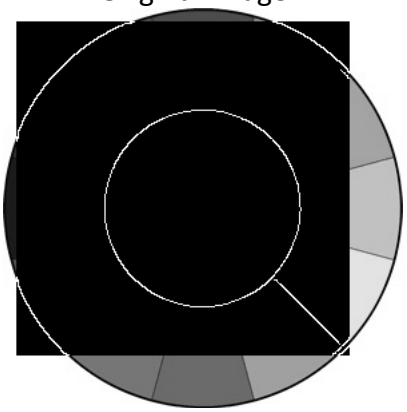
Hough Circle



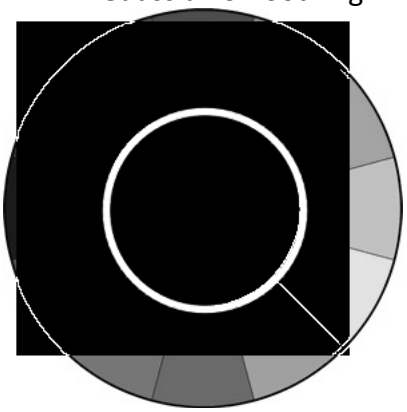
Original Image



Gaussian Smoothing



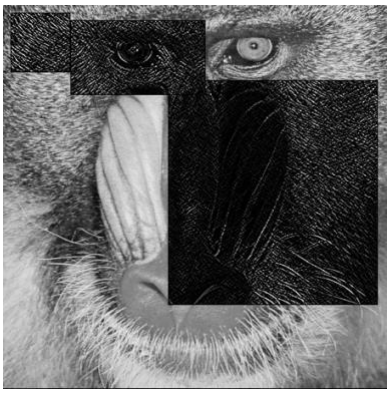
Canny Image



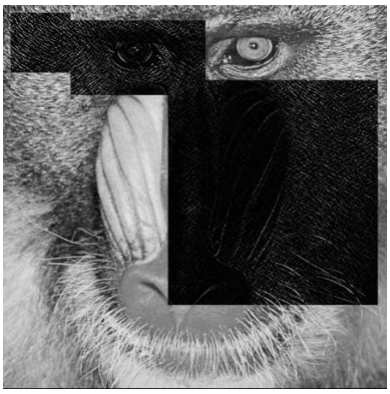
Hough Circle

Combining Operations

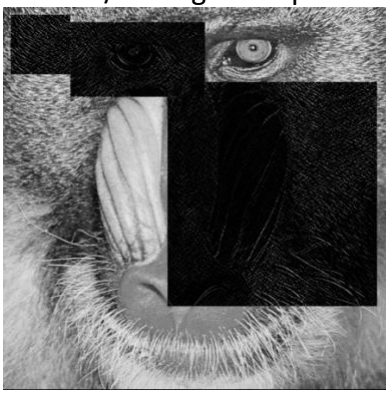
Perform histogram equalization followed by Sobel edge detector:



Sobel w/ Histogram Equalization

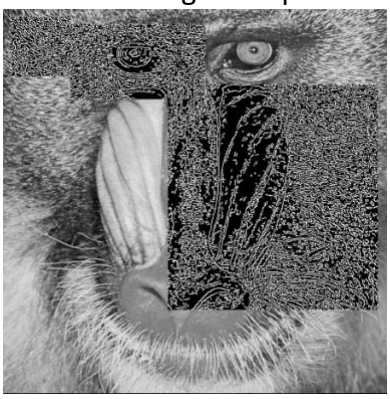


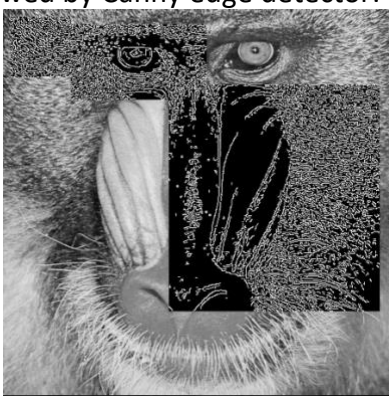
Sobel w/o Histogram Equalization



Sobel w/ HE subtract Sobel w/o HE

Perform histogram equalization followed by Canny edge detector:





Canny w/ Histogram Equalization Canny w/o Histogram Equalization

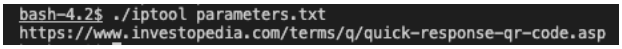
Canny w/ HE subtract Canny w/o HE

It is obvious that doing histogram equalization before performing edge detection will increase the number of edges being detected. As shown above, both Canny and Sobel edge detection produce a better result after doing histogram equalization.

QR Code Detection

QR code detection using OpenCV. If there is QR code in the region of interest, the program will detect and decode the code, then output the decoded message in console. The function still needs an input as output image file, even though it will simply copy the original image file. Example:

QR Code



Console Output

The function is able to read the downloaded QR Code image from the internet. However, the image taken by cell phone was extremely large and the program could not decode it correctly.

Performing histogram equalization prior QR reading does not help reading QR code, on the contrary, it makes the QR code harder to read. Because QR code relies on black and white contrast to be easily detected, histogram equalization flats the contrast and therefore make the QR code decoding more difficult. Thresholding will help with QR code detection and decode.