DIP Fall 2024 #58195
HW1B
Jeffrey Kerley
19 September 2024

Point Processes

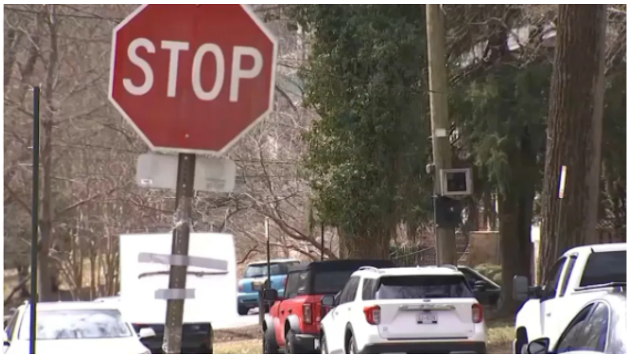
Abstract:

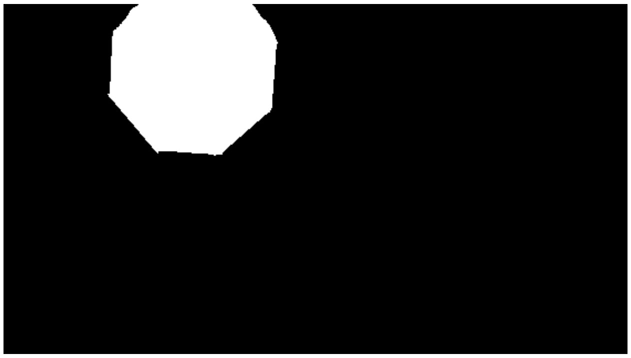
Classifying images and applying transformation on an image's pixels can be a daunting task. Converting a potentially multi-channel image into a histogram format can simplify and open the door for more complex image transformations. Applying a linear stretch results in a brightened image, while discarding these lower and upper bounds seems to result in a scaling factor for the stretch intensity.

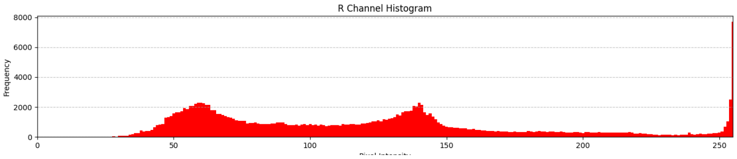
Introduction:

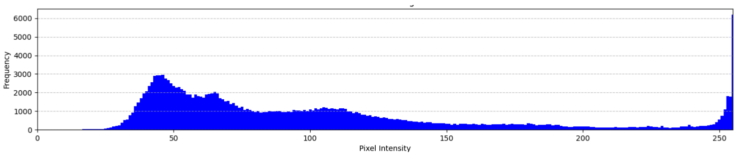
The goal of HW1B is to start to explore more operations to apply to an image, as well as understanding how certain statistical objects can make image operations easier and explain more about the operation itself. For example, a min-max stretch does seem to be intuitive for what will happen to the image, but coupling the image and histogram results in a clear connective picture between the stretch, discard, and resulting image transformation. The histograms themselves serve the purpose of explaining more fundamental properties of an image, such as its intensity.

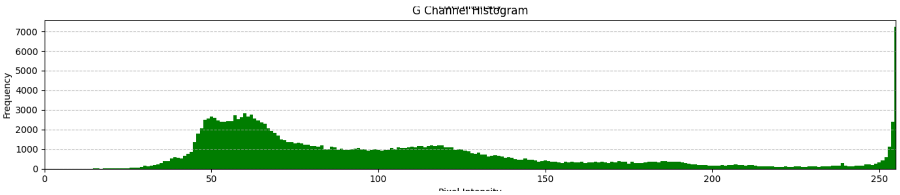
Experiments and Results:



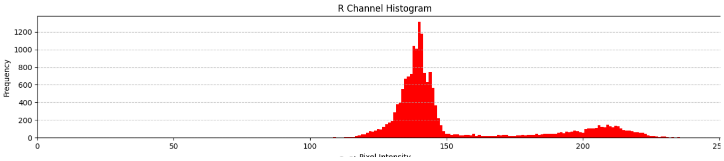


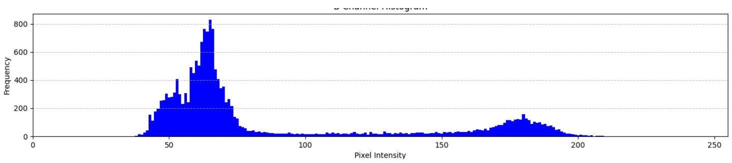


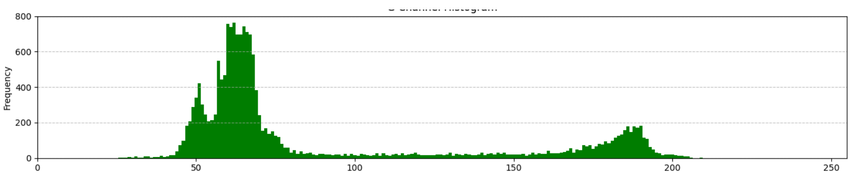




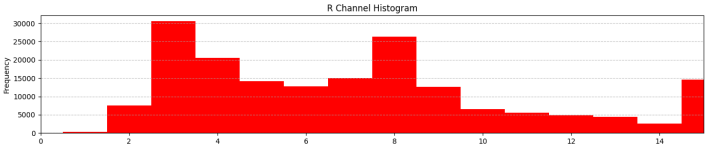
Regional(256):

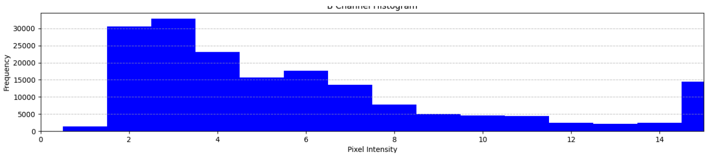


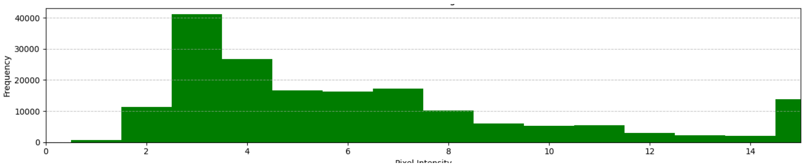




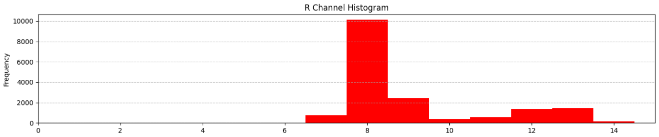
Full Image (16):

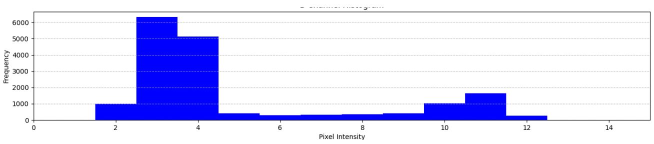


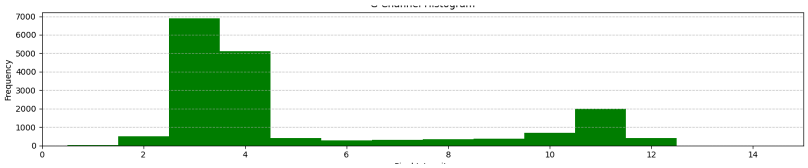




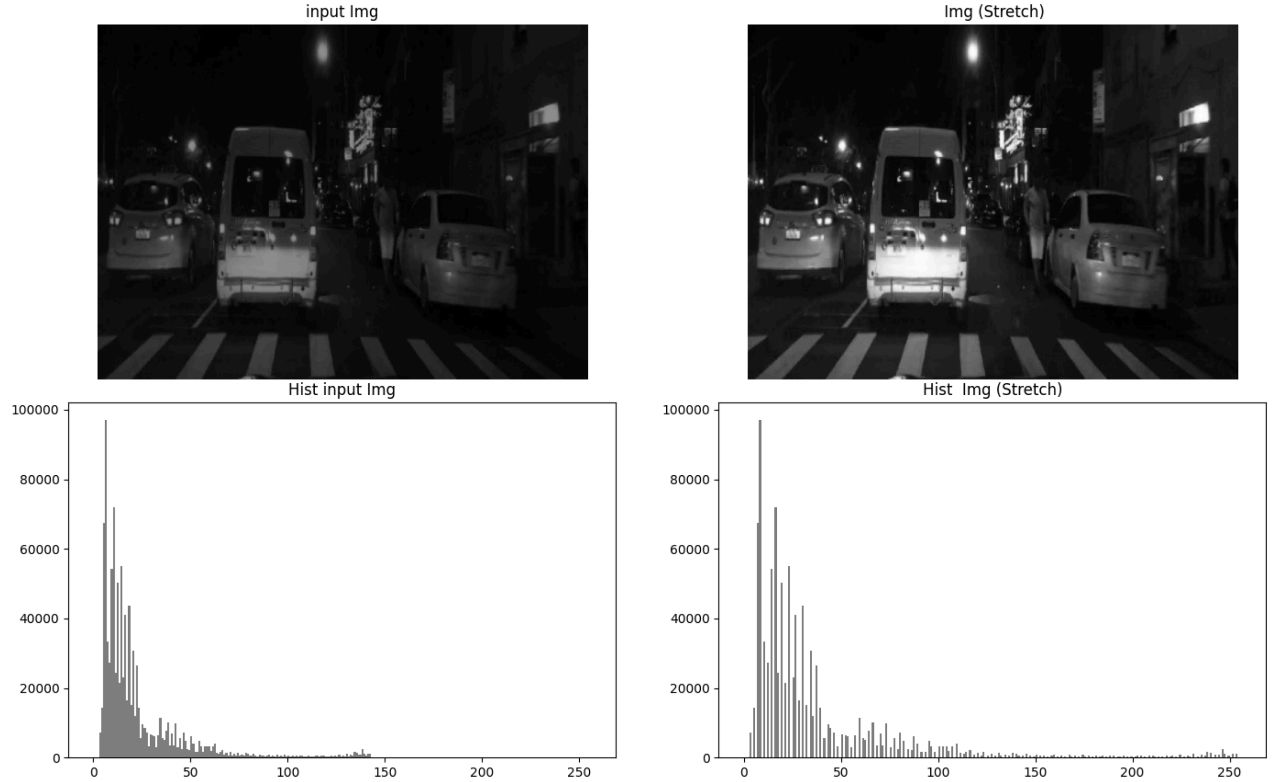
Regional(16):







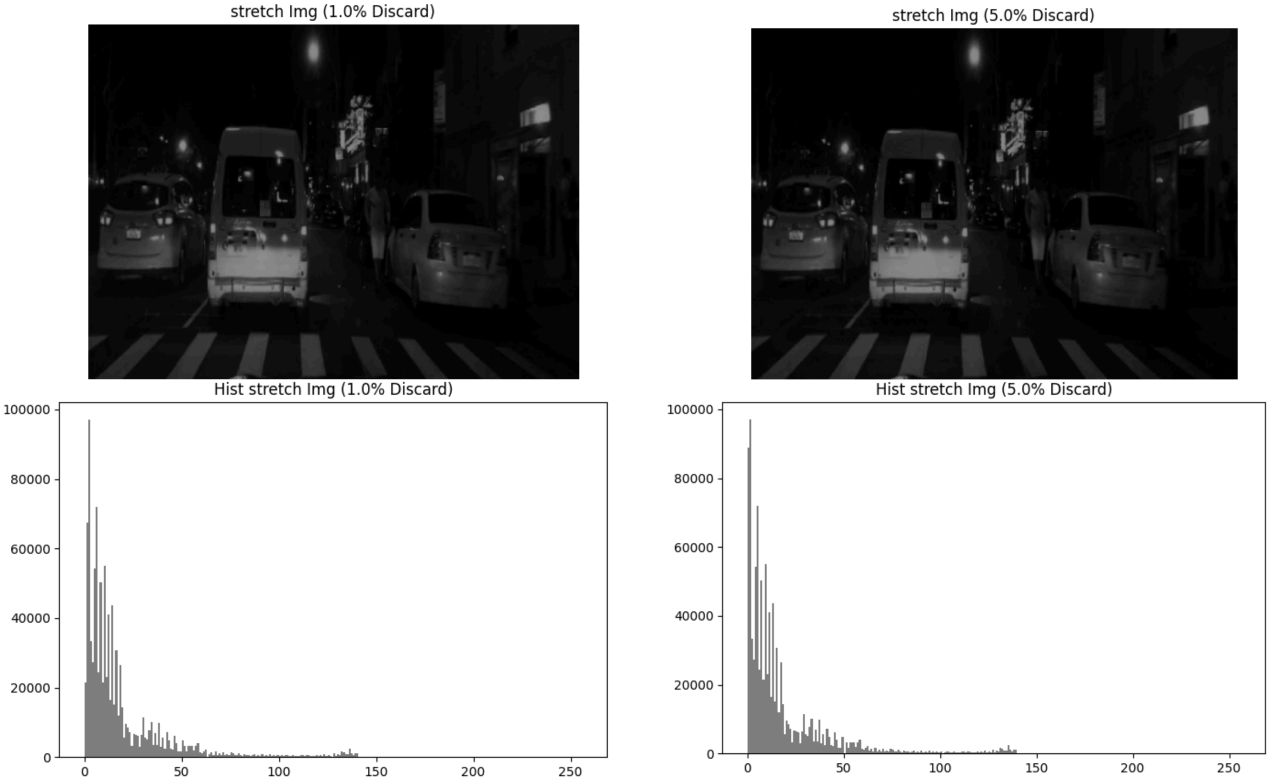
Histogram Type	Time To Process
Entire Image (256 bins)	0.7612609169445932
Regional Image (256 bins)	0.07326112501323223
Entire Image (16 bins)	0.7614475837908685
Regional Image (16 bins)	0.07187970820814371


input Img

Img (Stretch)

Hist input Img

Hist Img (Stretch)


stretch Img (1.0% Discard)

stretch Img (5.0% Discard)

Hist stretch Img (1.0% Discard)

Hist stretch Img (5.0% Discard)

Image	Min Intensity	Max Intensity
Original	1	143
Stretch min-max	0	255
1% Discard	0	141
5% Discard	0	140
60% Discard	0	135

Discussion/Conclusions:

I learned more about python list operations, such as easy list reversals, searching, and ravel operations to work with the image pixels. I also learned more image operations such as working with just the image type (taking it as a float), doing various cv operations like merge and split, as well as doing conversions between rgb, png with an alpha channel, etc. Working with the bins was easy as python lets you do conditional search using the mask (only get the mask pixels == 1), and then just put them into the bin that their pixel value is proportional to (such as bins = 16, the bin size for pixel will be 256/16). The stretch was interesting as it resulted in a dimming, when discarding large values of the pixels, and a brightening when applying the stretch as is, min-max.

References/Appendix: (As needed)

OpenCV Documentation:

https://docs.opencv.org/3.4/d8/dbc/tutorial_histogram_calculation.html

NumPy:

https://numpy.org/doc/stable/reference/generated/numpy.ravel.html