Project 2

Zachary Taylor,John Dinofrio, Cristian Bueno

March 7, 2020

Problem 1

Here, we aim to improve the quality of the video sequence provided above. This is a video recording of a highway during night. Most of the Computer Vision pipelines for lane detection or other self-driving tasks require good lighting conditions and color information for detecting good features. A lot of pre-processing is required in such scenarios where lighting conditions are poor. Now, using the techniques taught in class your aim is to enhance the contrast and improve the visual appearance of the video sequence. You can use any in-built functions for the same.

The provided source video was extremely dark. We resize the image to both speed up our adjustments and make it easier to see the changes side by side with the image source.
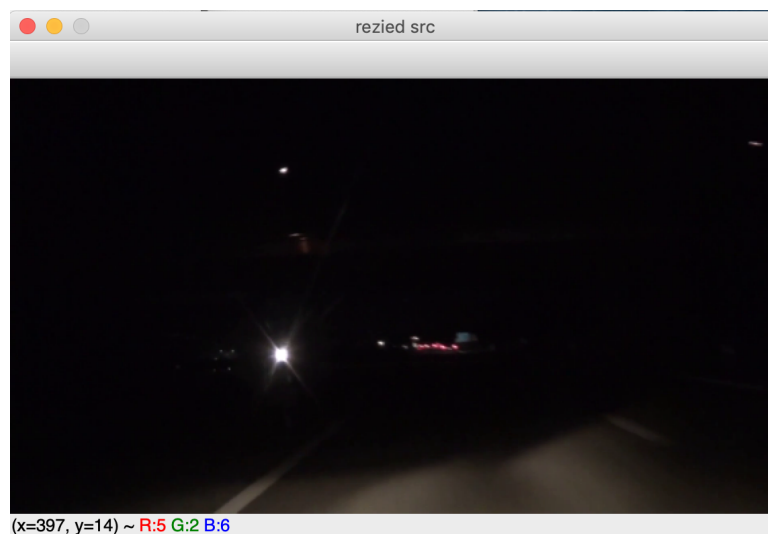

Figure 1: Source Image

The first approach we took to solve this problem was using the histogram equalization technique as the effect as demonstrated in the course materials looked powerful. We utilized the built in OpenCV function called equalizeHist() to do this operation. Initially we made each frame grayscale and then equalized, but we also separated the three color channels, equalized each individually, and merged them back together into one image to try to get an equalized color image. We also tried other techniques of converting the image into other color spaces and trying to equalize certain channels that might increase the brightness and contrast, all with little improvement over the other methods.
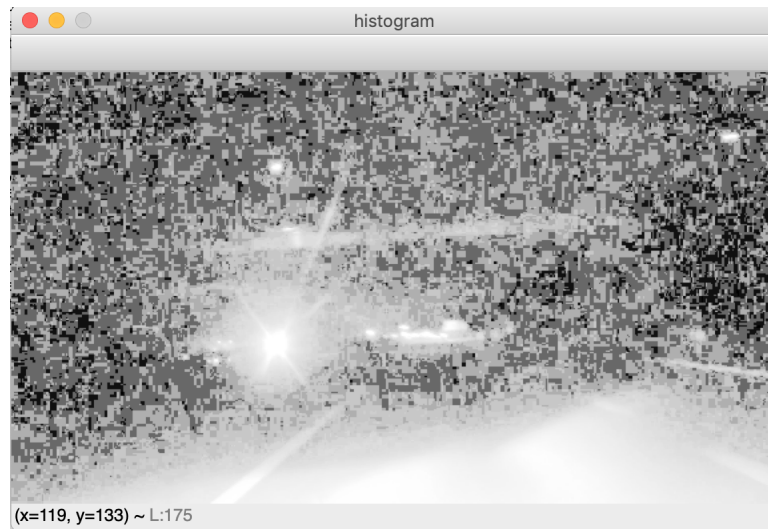

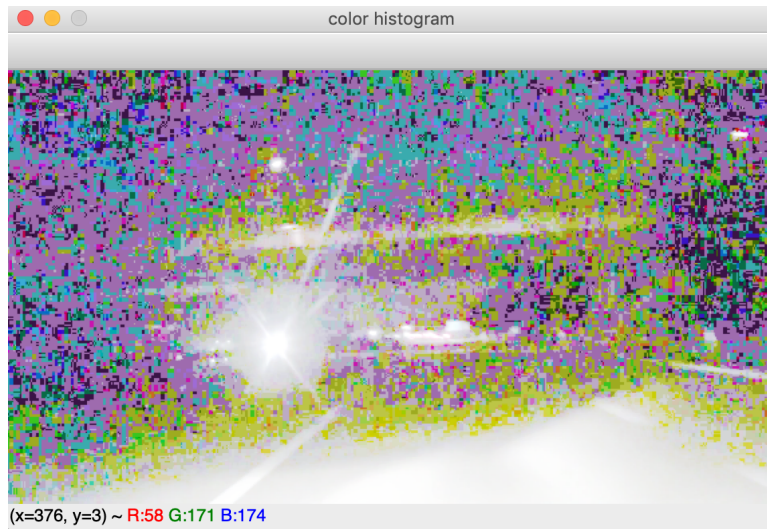Figure 2: Grayscale Histogram Equalization

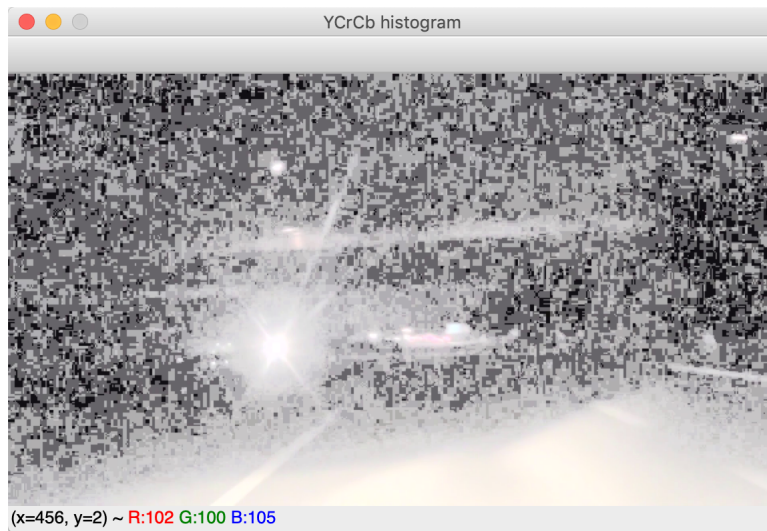Figure 3: 3 Channel Color Histogram Equalization

Figure 4: YCrCb Histogram Equalization

The consistent challenge we had using the built in histogram equalization is that we were getting a lot of noise in the image after equalizing. While it did make it slightly easier to see the lines on the road, the amount of noise in the image may make programmatically finding the lines on the road difficult.

We were expecting results similar to what was shown in the class slides that brightened the image and exposed details that otherwise couldn't be seen, but instead got an image with so much noise it was hard to see anything in it. In analyzing the first frame of the video as a sample, we found that the first 25 intensity values on the grayscale image made up 90% of the pixels in the image. Of those, there were 5 that represented 70% of the image. Since such a large portion of the image was concentrated in such a narrow range, we can expect some pixelation to occur. This is specifically because each pixel value can only map to one new value, or in other words, we can't add more fidelity to the image than what is already there. That means, since 70% of our image is populated by only 5 unique values, 70% of our image, after equalization, will be made up of only 5 different intensity values, just further distributed across the full 8 bit intensity range. Because these values are so spread out, the image will look pixelated instead of smooth. We can do some smoothing after the fact but then some detail is lost in the lane markings we are trying to uncover.
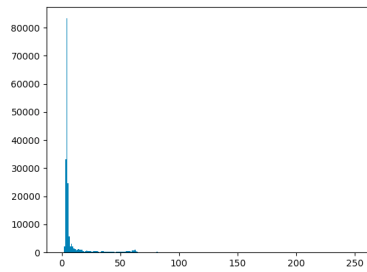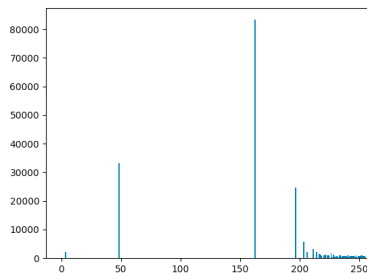

Figure 5: Source Grayscale Histogram

Figure 6: Equalized Histogram

Through some searching we found another technique that adjusts the gamma of the image. We found this provided the best result.

Figure 7: Gamma Adjustment