HW 04

1. To Do: Change file name
2. Code works
3. Half way through documenting...
4. I could mostly tell what was going on especially since edges were kept in high contrast areas. But, it was harder to see what was going on in the areas with lower contrast, since the slight variations in color was what was distinguishing them.
5. Color values and distribution of vals
6. I would have expected that the pants would be outlined too since the the pants color and the color of the background are vary different, but the program doesn’t pick up on these edges.
7. I think the edges show up very well on this image because the contrast is pretty high. The sky, the trees, the grass, the road, the marlins on the road, etc they all have an established boundary between them and the colors at that boundary are different and not muddy, therefore it is easier to detect and show edges.
8. No, certain texts were easier to read than others. The sign reading Old Landmark was hard to read because the values contained in the image are close together, making the distinction of the cut-off percent not valuable in determining what is an important edge and what isn’t
9. It does quantization differently; theres a different amount of bins. The first run through bins depending on the highest value of edge. The values are split into 256 bins and are even in size. The 2nd time, bins are just steps of .0001, this means that it isn’t guaranteed that bins are the same size.
10. We reduce the percentile of values of edges that we keep, which lets in edges detected that are not actually “edges” to remain in the final picture. As a result, these pictures are a lot noisier, and you can see a lot of texture that isnt necessarily edges like leaves of the trees or the texture of a wall.
11. Noise is greatly reduced, but so is the sensitivity to actual edges. While the unnecessary texture maybe gone, the boundary between edges are starting to look faint because we are now too selective in how we consider which values should count as proper edges.
12. Numel: returns the number of elementar there are in a given array

Histcounts: bins a given array’s values into bins specified by a number or vector

Cumsum: finds the cumulative sum of an array’s values

Find: finds indices and values of nonzero elements, in our program it returns the first index that has a value greater than the 95th percentile of pixel