Caitlin Kimbrell
Nikhil Raina

HW 04

a) Since this was a group project, we decided to name the file as 'HW_04_Kimbrell_Caitlin_Raina_Nikhil_Edge_Stats_and_Display' and followed the alphabetical order for it.

b) Code works as required.

c) Code has been documented entirely

d) 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. From the resulting image, one could not tell what the old miage was, however, there are a few highlighted areas that allow some resemblance to be shown. These highlighted areas are the edges that exist in and above the 95th percentile of all possible edges in the image, in terms of its contrast. The others are being removed or being set at 0.

e) The red and blue channels are being removed along with the edges that are not in or above the 95th percentile, in terms of contrast.

f) The resulting image shows the most contrasting edges of the image. The entire image appears to be blue with the highlights in white.

g) The expectation was that the pants would be outlined too since the pants color and the color of the background are very different, but the program doesn't pick up on these edges.

h) The edges show up very well on this image because the contrast is 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.

i) No, there doesn't seem to be any edges missing that should have been noticed. The edges are showing up well as the boundary clearly contrasts the 2 images that are being shown side by side.

j) 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.

k) It does quantization differently; there's a different number of bins. The first runs 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.

l) The general shape of the entire histogram shows a Rayleigh Distribution graph. Here we can understand that the light shades or the higher contrasting edges are less compared to the ones that are darker or lower contrasting edges. Thus the bell curve is

much higher for the data below the 50th percentile itself. We can also see the spread of the data among all the bins in the histogram.

m) Most of the images have the same trend for the distribution except the relaxing_jaguar, camo_failure and the best_traffic_cone distributions. Their distributions have a larger spread in terms of bin specific values across all the bins. As a result the bell curve does not appear to be sharp and more like a hill or a plateau. This is because the edges that were found are more homogenous with one another, that is the contrasting effect, being shown with boundaries of different parts of the images, is less.

n) The line represents the 95th percentile. The values in the bins on the right are above the 95th percentile and the ones below the line are below the 95th percentile.

o) No, all the edges that were being shown were as expected as the difference in the contrast from the kite and the sky, being separated by the boundary is high.

p) 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 isn't necessarily edges like leaves of the trees or the texture of a wall.

q) Noise is greatly reduced, but so is the sensitivity to actual edges. While the unnecessary texture may be gone, the boundaries between edges are starting to look faint because we are now too selective in how we consider which values should count as proper edges.

r) 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

Conclusion

One of the things we learned is exactly how binning in practice is used and two different ways someone can bin data. We learned the built in Matlab function histcounts is used to bin data in specified ways. We also learned how the Cut-Off percentile affects the results of algorithms to find edges and how a balance needs to be struck between including edges that are important, but disregarding edges that add to the noise of the final image. Figuring out how the overall distribution of values in an image can affect how well the computer can find edges was also interesting. One example is in the picture that showed the sign reading “Old Landmark.” In the original image, it is very easy to view the picture and read the sign, but the program has issues trying to find the edges because the values are not as well distributed as some of the other pictures. This makes the cutoff percentile not as impactful, and results in an image that is unreadable.