Andrew Ramirez

September 10, 2021

Project 3

LSB

For the LSB implementation, I used the code from https://github.com/kelvins/steganography/blob/main/steganography.py with some edits in order to track MSE (I'm not completely sure I did it correctly, but I did try).

For testing, I randomly used 2 images to merge and unmerge, and did so 20 times, so of course there was a chance for repeats. The error I got was around 200 for most pairs, though there was one that fell just bellow 100, which was interesting. I'll admit I struggled a bit in the set up for testing the kodak set from not fully understanding the required input path that certain modules wanted, but I did eventually figure it out.

NN

For the NN implementation, I used the code from https://github.com/fpingham/DeepSteg/blob/master/DeepSteganography.ipynb with some edits in order to actually get it to run. I spend a lot of time with quite a few different versions of this method, and at some point I just could not fix them. I did figure out how to actually install the different libraries and modules that I need, but I would always run into some sort of error(I couldn't for project 1. In hindsight, it was obvious how, but better late than never I suppose). That was the case with this as well, though thankfully it was not too difficult to fix it in the end (I guess most issues revolved around archaic methods of calling modules that were no longer valid, so I had to find workarounds for both that, as well as version conflicts, though that was relatively simple after spending a bit of time looking things up).

For this method, there was a significantly lower error (though part of that can probably be attributed to the way that error was calculated, in case I did it wrong for the LSB).

With testing the neural network on the kodak set, we see that the results are quite good, and that the main issue is simply the loss of a bit of color (which looks pretty cool).