Image Compression Assignment 2 Report

Samuel Near

251102998

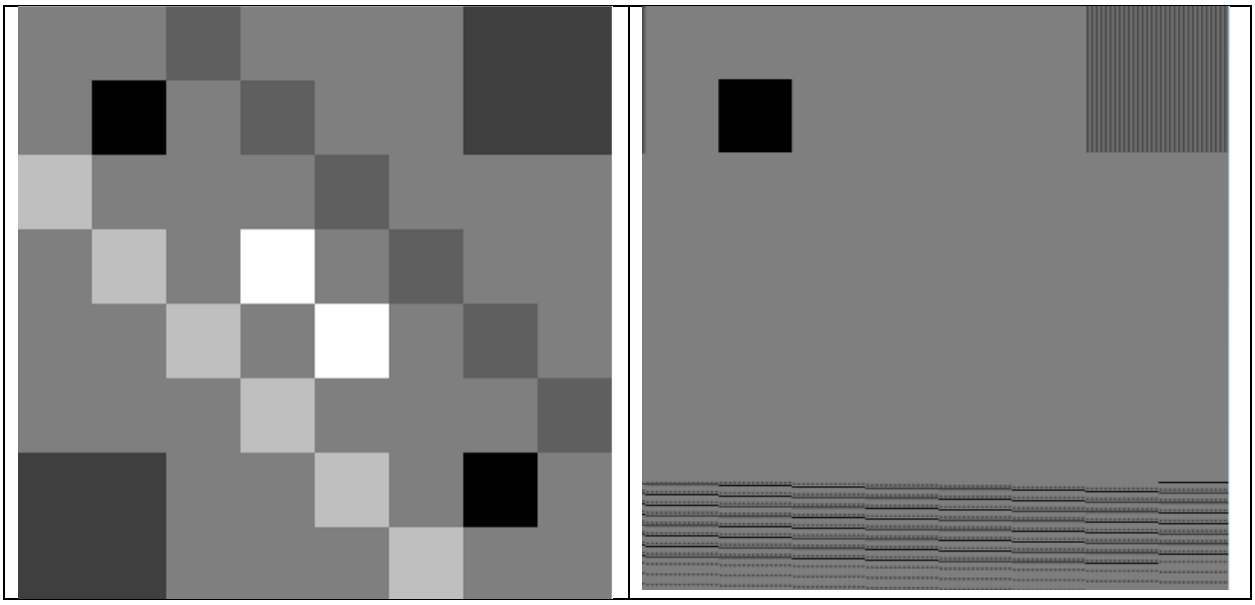
CS4481 – Image Compression

Dr. Mahmoud El-Sakka

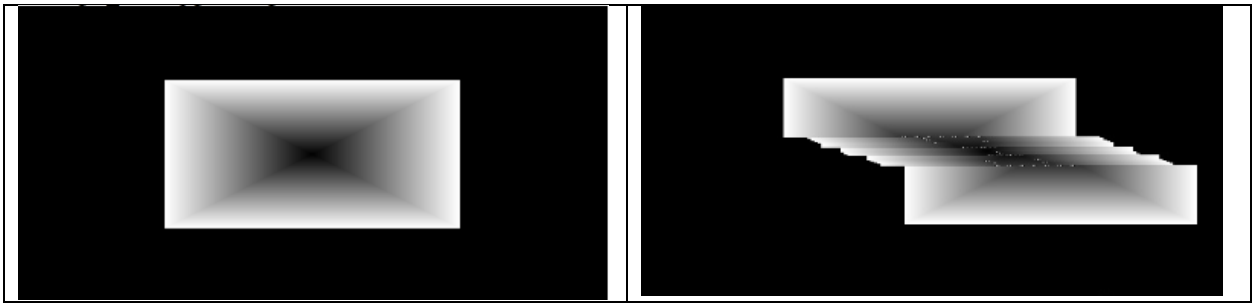
February 29, 2024

Unfortunately, I couldn't get my program working properly, I was able to encode and decode perfectly with the example that we did in lecture, all the Huffman nodes were correct, and the Huffman tree was also correct. However, when I tried to encode and decode real images, I couldn't get the images to come out nice. I still want to hand in all my code as I spent MANY hours on it and I do think it is just a small bug that I can find, but I am out of time to debug! Ill go over each image before and after compression. My mean absolute error program works as intended.

Image 1

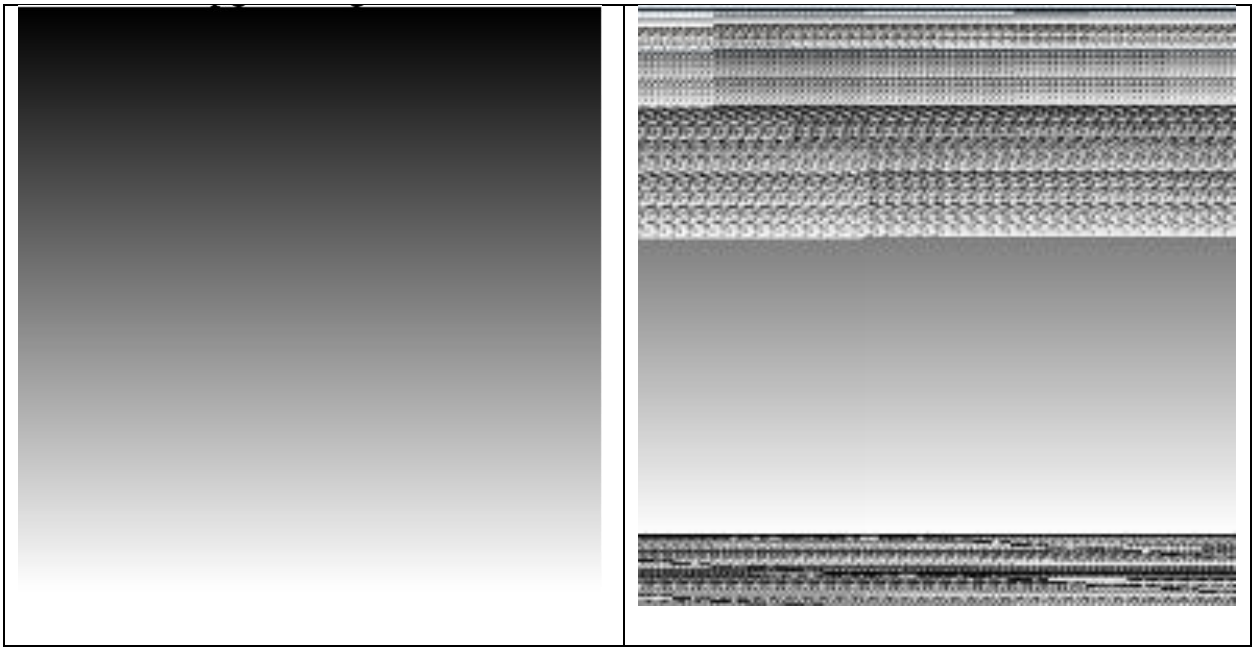


Mean absolute error: 23.173580. test_square.raw.pgm → test_square.decoded.pgm

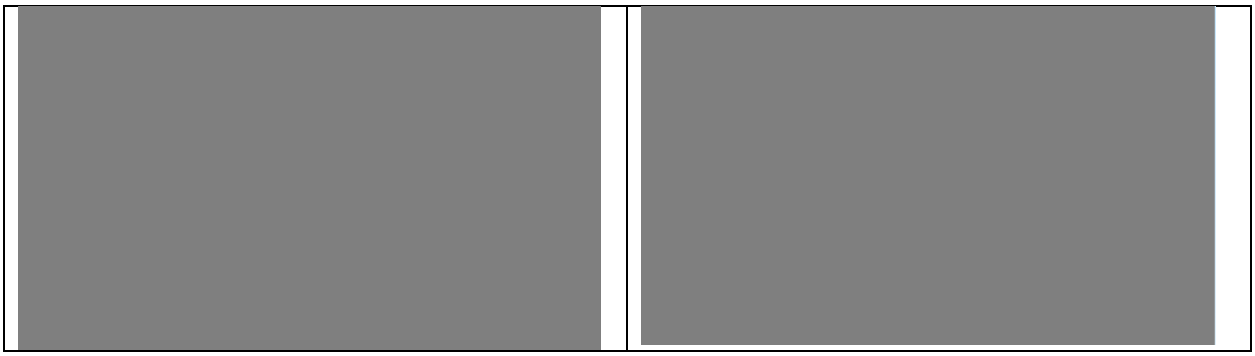


Mean absolute error: 25.544601. rectangle_2.raw.pgm → rectangle_2.decoded.pgm

Mean absolute error: 54.965744. boats.raw.pgm → boats.decoded.pgm



Mean absolute error: 70.907944. smooth.raw.pgm → smooth.decoded.pgm



Mean absolute error: 0.00000. flat.raw.pgm → flat.decoded.pgm

In this case my program kind of got “Lucky” and I think this only works because of it’s simplicity to not break.

When comparing two of the exact same images, as in same file name, the mean absolute error function returns 0.

As I stated before and you can see from the images, I think it is a very small error that is causing problems with the encoding and decoding. Since it is hard to verify each step of the process with larger sets of data, I couldn’t pinpoint what my error was. I was able to break down every step of the process and confirmed that everything was working correctly with the example data I implemented from the lecture. Unfortunately I have ran out of time to debug.