**COMPUTER VISION**

**Intro:**

A digital image is composed of a two-dimensional array of pixels. If the images has r rows and c columns, then there are r\*c pixels in the image. Each pixel has a location and value. The location is given by the row and column of the pixel.

In RGB imaging (red, green, blue), each pixel in a color image has three intensities, one for red, one for green, and one for blue (these are the primary colors of light). Three bytes (0 to 255) are used for each pixel in a color image. Images take values (call the intensity) from 0 to 255 (1 byte = 8 bits of information). 0 is the darkest black and 255 is the brightest white.

**In this project:**

“ImageLib.lib”, “ImageLib.pdb”, and “ImageLib.h” are abstractions and implementation for the ImageLib class, representing an image object (only in GIF format). All methods and their specifications are listed in “ImageLib.h”, please give it a look.

To put my spin on the original ImageLib class, I have created another class that uses the ImageLib class abstractions. I name my class Image.

“Image.h” and “Image.cpp” are my own abstractions and implementation for the Image class. All methods and their specifications are listed in both files, please give it a look. My driver file is “main.cpp”. I use this file to simulate the methods I have written and display the results performed by the class files.

There are two GIFs – “test2.GIF” which is an example input image (you can just choose your own GIF file to test out the functionalities of the Image class); and “output.GIF” which is the output file performed by my driver file. Please look at “main.cpp” as well.

**Disclaimer:**

I do not own the following files: “ImageLib.lib”, “ImageLib.pdb”, “ImageLib.h” and “test2.GIF”. They are provided to me by my Data Structures professor at the University of Washington.