CGRA352 Assignment 1 Report

Running the program:
To run the program, the program uses an argument to find the location of the input images. For Example res/Flower.jpg will load the Flower image if it is located in the res folder. For this assignment that does mean that the Building image will have to be tested individually to the flower.
When the program runs, it will run each function then load each image in a separate window.

Overview of Functions:
cgraHsvImage() is a function that takes an image, separates the RGB values as well as separating the HSV values. The function then combines all images together and returns the new combined image. It is also important to note that the newly created image is ordered Blue,Green,Red from left to right. With white representing a large amount of that value in the image and black representing a small amount of the value.

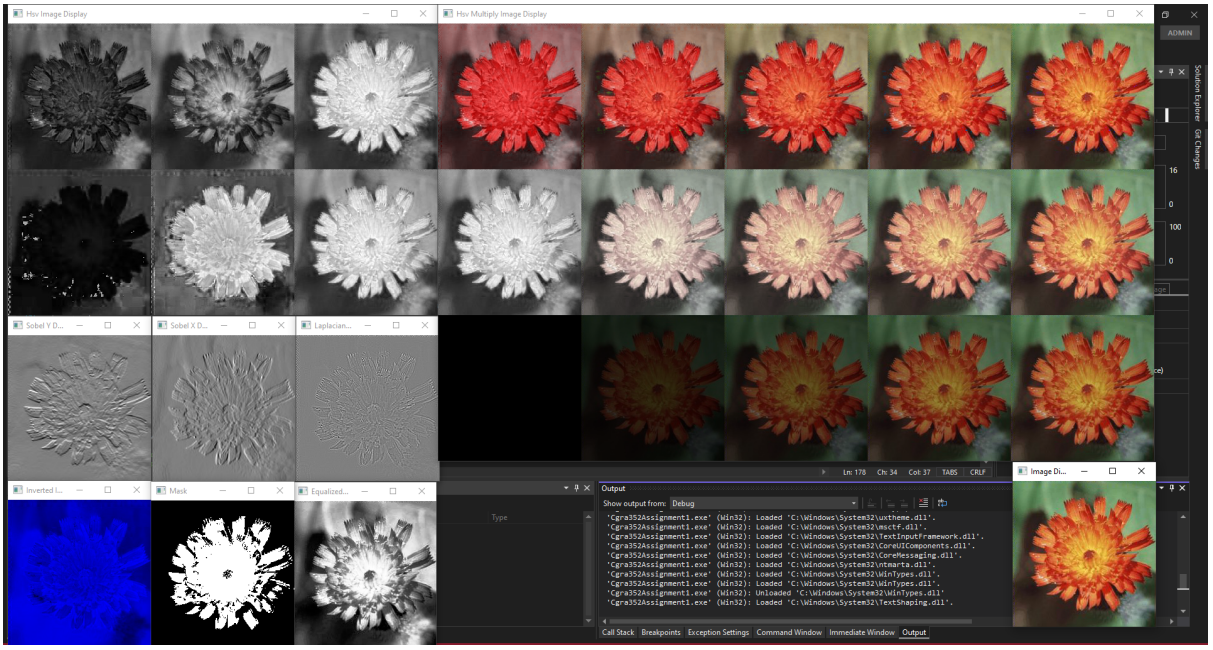
cgraHsvMultiplyImage() is a function that takes an image, converts it into HSV and then multiplies the HSV values by (from left to right) 0.0 , 0.2 , 0.4 , 0.6 and 0.8.It then converts the image back into the RGB colorspace.

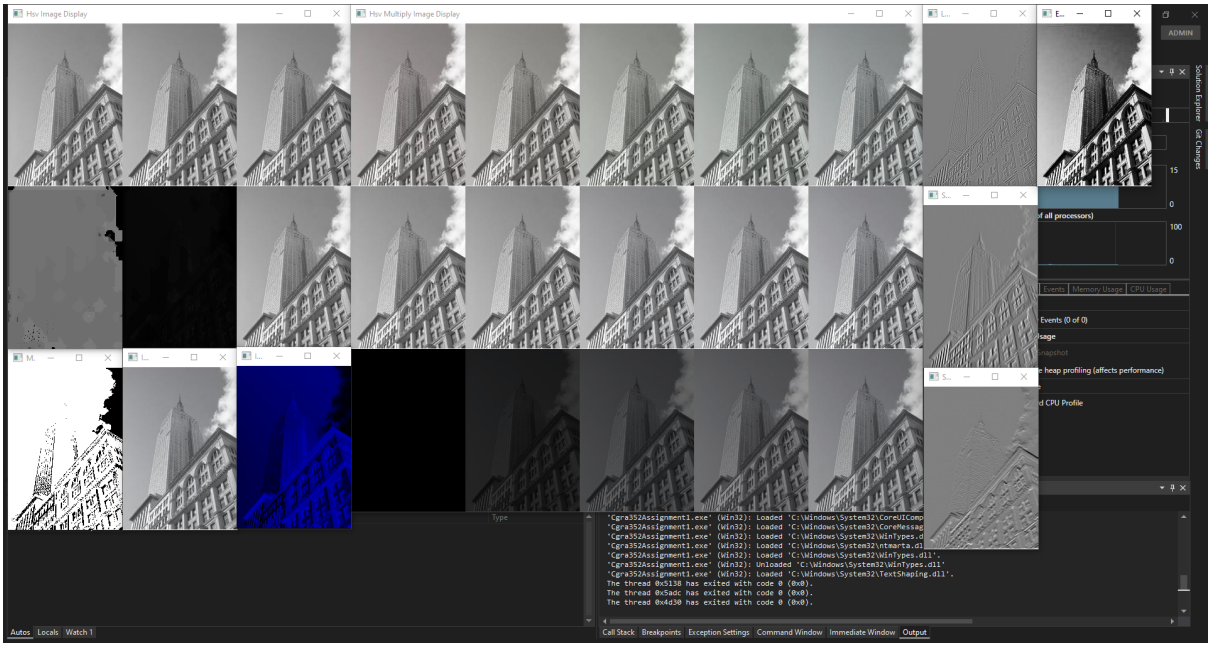
cgraMaskImage() is a function that takes an image. It then goes through every pixel in the image and checks if the euclidean distance from the point (80,80) is less than 100. If it is then it sets the pixel to white (255,255,255) else it sets the pixel to black (0,0,0).

cgraConvolutionImage() is a function that takes an image and a Matrix ‘K’. It uses the openCV function filter2D to convolute the input image and output a new image based on the kernel. The function then goes through and scales the image by using the function minMaxIdx which finds the maximum and minimum values of a matrix, we use this to scale the image appropriately. When we call this function we pass it the appropriate Matrix to generate either a Laplacian, Sobel X or Sobel Y image.

cgraEqualizedImage() is a function that takes an image and equilizes its histogram, that is we find how many pixels there are at each value (0-255) and we scale the distances in between the pixels. This means we do not change the number of pixels with a certain value but the image is now able to use a larger range of values. We do this using openCvV’s calcHist function which calculates how many pixels there are for each intensity between 0-255. We then scale the distance between pixels appropriately.

Results:


Hsv Image Display
Hsv Multiply Image Display
Sobel Y D...
Sobel X D...
Laplacian...
Inverted I...
Mask
Equalized...
Image Di...
Output
Show output from: Debug
'Cgra352Assignment1.exe' (Win32): Loaded 'C:\Windows\System32\uxtheme.dll'.
'Cgra352Assignment1.exe' (Win32): Loaded 'C:\Windows\System32\msctf.dll'.
'Cgra352Assignment1.exe' (Win32): Loaded 'C:\Windows\System32\TextInputFramework.dll'.
'Cgra352Assignment1.exe' (Win32): Loaded 'C:\Windows\System32\CoreUIComponents.dll'.
'Cgra352Assignment1.exe' (Win32): Loaded 'C:\Windows\System32\CoreMessaging.dll'.
'Cgra352Assignment1.exe' (Win32): Loaded 'C:\Windows\System32\ntmarta.dll'.
'Cgra352Assignment1.exe' (Win32): Loaded 'C:\Windows\System32\WinTypes.dll'.
'Cgra352Assignment1.exe' (Win32): Loaded 'C:\Windows\System32\WinTypes.dll'.
'Cgra352Assignment1.exe' (Win32): Unloaded 'C:\Windows\System32\WinTypes.dll'.
'Cgra352Assignment1.exe' (Win32): Loaded 'C:\Windows\System32\TextShaping.dll'.
Call Stack
Breakpoints
Exception Settings
Command Window
Immediate Window
Output


Hsv Image Display
Hsv Multiply Image Display
'Cgra352Assignment1.exe' (Win32): Loaded 'C:\Windows\System32\WinTypes.dll'.
'Cgra352Assignment1.exe' (Win32): Unloaded 'C:\Windows\System32\WinTypes.dll'
'Cgra352Assignment1.exe' (Win32): Loaded 'C:\Windows\System32\TextShaping.dll'.
The thread 0x5138 has exited with code 0 (0x0).
The thread 0x5a0c has exited with code 0 (0x0).
The thread 0x4d30 has exited with code 0 (0x0).
Autos
Locals
Watch 1
Call Stack
Breakpoints
Exception Settings
Command Window
Immediate Window
Output