Eric Kearney
CS 390S – Digital Image Processing
Professor Feng Jiang
26, August 2018

Assignment 1

I decided to try and use OpenCV with Python for this course. If I'm ever unable to complete an assignment with OpenCV, I will switch to Matlab, but for my personal future prospects being able to write Python will give me an enormous boost, so for now I'm taking every opportunity available to me to sharpen my Python skills.

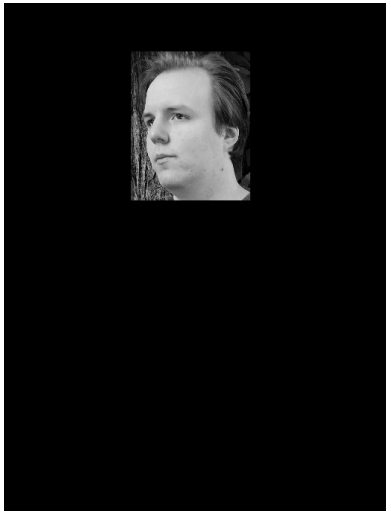
I used a portrait of myself that a friend took for this assignment. First, I have to resize the portrait, as the actual image size was enormous (2584x3446), so I had to scale it down to be able to work with it. I then converted the image to grayscale. I experimented with using a box and a circle as my regions of interest. To implement the box region of interest. To chose all the points and values for this assignment, I displayed the resized image in a window, then I moved my mouse so that it was approximately where I wanted my box's corner or my circle's center to be. I chose the point (65,170) for the top-left corner, then I chose a value of 160 of the width of the box and 200 for the height of the box. Instead of setting the value of pixels outside of the box to 0, I created a blank black image of the same size as my image, then I copied every pixel within the box to the new image.

The circle was a little more complicated, I chose my center (250,160) and my radius (100) using the mouse-method I described above. Then I wrote a nested `for` loop, so that I could iterate over every pixel in the image. Then, for every pixel, I used the distance formula, sqrt((x-x_i)+(y-y_i)), where (x,y) is the center of the circle, and (x_i,y_i) is the given pixel's coordinates. If the distance was greater than the radius, that pixel is outside of the circle, so I set the pixel's value to 0.

The code I wrote, which includes with it the sources I used to help me with this assignment, is included with this folder, '1assignment.py'.

Original Image

Grayscale



Box ROI

Circle ROI