Pablo Medrano
A204107

cs512
Assignment 2

1. Problem statement

1.1. The image to be processed by the program should be read from a file if a file name is specified in the command line or capture it from the camera if a filename is not specified in the command line

1.2. The read image should be read as a 3 channel color image

1.3. The program should work for any size image

1.4. Special keys on the keyboard should be used to modify the display image as follows:

a) 'i': reload the original image.
b) 'w': save the current image into the file 'ouput.jpg.'
c) 'g' : convert the image to grayscale using the OpenCV conversion function.
d) 'G': convert the image to grayscale using your implementation of conversion function.
e) 'c': cycle through the color channels of the image showing a different channel every time the key is pressed.
f) 's': convert the image to grayscale and smooth it using the openCV function. Use a track bar to control the amount of smoothing.
g) 'S': convert the image to grayscale and smooth it using your function which should perform convolution with a suitable filter. Use a track bar to control the amount of smoothing.
h) 'd': downsample the image by a factor of 2 without smooting.
i) 'D': downsample the image by a factor of 2 with smoothing.
j) 'x': convert the image grayscale and perform convolution with an x derivative filter. Normalize the obtained values to the range [0,255].
k) 'y': convert the image to grayscale and perform convolution with a y derivative filter. Normalize the obtained values to the range [0,255].
l) 'm': show the magnitude of the gradient normalized to the range [0,255]. The gradient is computed base don the x and y derivatives of the image.
m) 'p': convert the image to grayscale and plot the gradient vectors of the image every N pixel and let the plotted gradient vectors have a lenght of K. Use a track bar to control N. Plot the vectors as short line segments of length K.
n) 'r': convert the image to grayscale and rotate it using an angle of teta degrees. Use a track bar to control the rotation angle.
o) 'h': display a short description of the program, its command line arguments, and the keys it supports.

2. Proposed solution

2.1. The user is asked to write a filename to load an image to perform operations on it or press enter key to capture an image from the camera.

2.2. If the image is 1 channel only (it is gray) it is converted to 3 channel image.

2.3. If the image is too big to fit the screen it is reduced until it fits it.

2.4. Special keys functions:

a) To reload the image a function is called that shows the original image if it was from a file or the camera takes a new picture if a filename was not given
b) The image is saved to "output.jpg" using openCV function imwrite().
c) The image is converted to gray using openCV function cvtColor()
d) The image is converted to gray converting each pixel to gray from the bgr values using the formula: g = 0.299*b + 0.587*g + 0.114*r
e) To cycle through the 3 channels bgr each time the key is pressed two of the channels are made zero.
f) The image is converted to gray using openCV function cvtColor() and track bar that controls a smoothing filter is created in the window to control the amount of smoothing.
g) I tried to convolve a filter with the image creating a function that performed convolution but I was not able to get it done correctly. Some errors appeared when trying to convolve the image and the filter.
h) To downsample the image without smoothing its dimensions are divided by 2.
i) To downsample the image with smoothing the openCV function pyrDown() is used.
j) The image is converted to gray using openCV function as before then a Sobel filter of size 5 is applied to image only in x direction. Finally, it is normalized.
k) The image is converted to gray using openCV function as before then a Sobel filter of size 5 is applied to image only in y direction. Finally, it is normalized.
l) Both Sobel filters in x and y directions are applied to calculate the magnitude doing the square root of the sum of them to the power of 2. The magnitude is normalized afterwards.
m) Image is converted to gray and Sobel filter in both directions is applied. For every N pixel the angle of its gradient is calculated as the inverse of the tangent of Sobely divided by Sobel x. Then knowing the angle, both x and y components can be calculated and the arrow drawn.
n) Image is converted to gray and rotate through the track bar using openCV functions getRotationMatrix2D() and warpAffine().
o) Information about keys' functionally is print on terminal.

3. Implementation details

First of all, I had some problems installing openCV since I had already anaconda3 installed with python 3.6 for another course. After a couple days trying I was able to install openCV but now spyder is not compiling anything I don't know why so I used Atom and the macOS terminal to compile.

I have had some problems with global and local variables for functions I was getting errors that variables were not declared. Finally, I figured it out but I think not in the best way. I am not sure if it is a python issue since I have just started coding in python this semester and I was using C and C++ before but I will look deeper into it for new assignments.

Apart from that, most openCV functions were straight forward to use.

I had some issues as well with the track bar because I wasn't reloading the image correctly but once I discover the problem it was the same solution for all the functions that used the track bar.

As I said I have not been able to implement the convolution. I searched the internet and find some code that it was supposed to do the convolution but none of it worked well.

I have not found a method to eliminate the track from the window when a new button is pressed if it was previously there. A possible solution could have been to create two windows, one without the track bar and another one if the track bar was used, but I did not like it very much so I didn't program it that way.

4. Results and discussion

Below I am going to show the output of all the functions:

a) This is the original image and it is the one that is reloaded when ‘i’ key is pressed.

b) All images used in this document have been exported from the program using the key ‘w’ that is the one that saves the image to a file.

c) When ‘g’ key is pressed the image is converted to gray using the openCV function.

d) When ‘G’ is pressed the same conversion to gray is done but using my own code instead of openCV function. The result is indistinguishable from the previous one but it takes some a couple seconds to do it since there is a for loop.

e) When ‘c’ is pressed several times the image displays the 3 channel colors bgr.

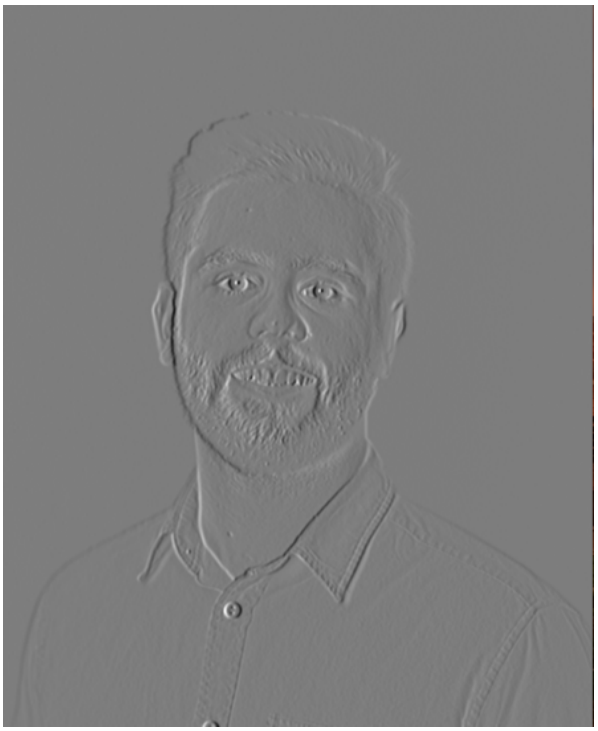
f) When ‘s’ is pressed image is converted to gray and smooth is controlled with the track bar.

g) When ‘S’ is pressed it was supposed to do the same as the previous one but I could not get it done.

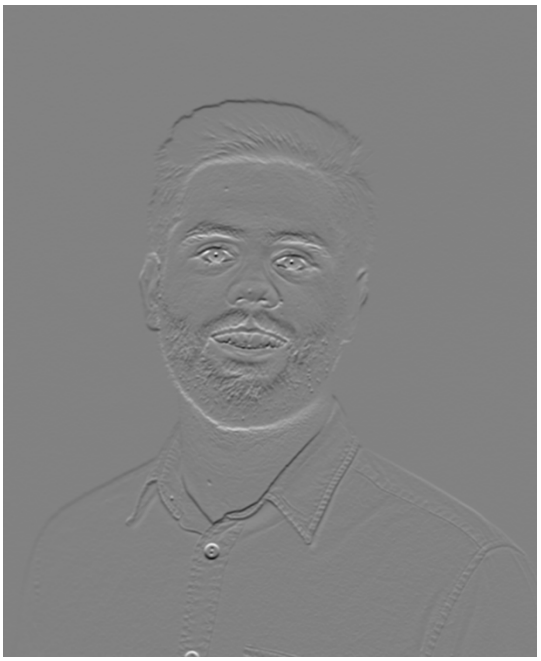
h) When ‘d’ is pressed the image is downsampled by 2 without smoothing. Since I am reducing the size of the image I cannot show that it is downsample by 2.

i) When ‘D’ is pressed the image is downsampled by 2 with smoothing. Again I cannot show the downsample but the smoothing can be seen in the image compared to the previous one.

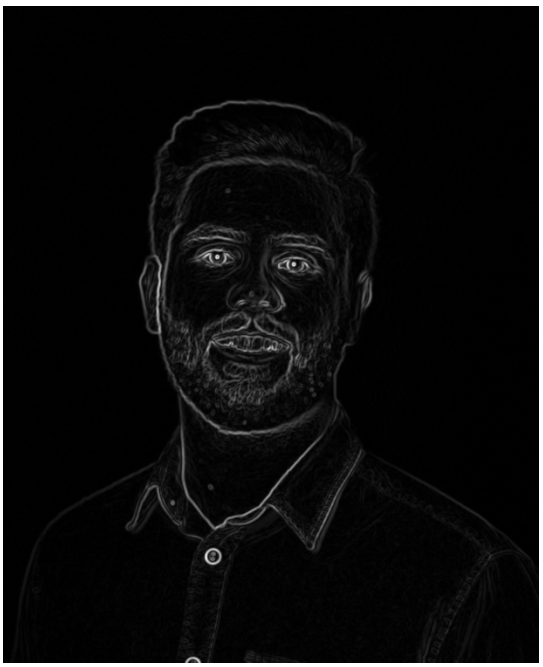
j) When ‘x’ is pressed the image is converted to gray and convolve with an x derivative filter (Sobel).



k) When ‘x’ is pressed the image is converted to gray and convolve with an x derivative filter (Sobel).



l) ‘m’: show the magnitude of the gradient normalized to the range [0,255]. The gradient is computed base don the x and y derivatives of the image.



m) ‘p’: convert the image to grayscale and plot the gradient vectors of the image every N pixel and let the plotted gradient vectors have a lenght of K. Use a track bar to control N. Plot the vectors as short line segments of length K.

n) ‘r’: convert the image to grayscale and rotate it using an angle of teta degrees. Use a track bar to control the rotation angle.

5. References

https://stackoverflow.com/questions/3614163/convert-single-channle-image-to-3-channel-image-c-opencv

https://stackoverflow.com/questions/43373521/how-to-do-convolution-matrix-operation-in-numpy

https://stackoverflow.com/questions/30079740/image-gradient-vector-field-in-python

http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/sobel_derivatives/sobel_derivatives.html

https://stackoverflow.com/questions/13003949/faster-way-to-loop-through-every-pixel-of-an-image-in-python

http://docs.opencv.org/3.3.0/da/d6e/tutorial_py_geometric_transformations.html

http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_core/py_basic_ops/py_basic_ops.html