ASSIGNMENTS

Assignment #4: Gradients and Edges - Returned

Title	Assignment #4: Gradients and Edges
Student	Tran, Ngoc T
Submitted Date	Sep 20, 2015 9:14 pm
Grade	95.0 (max 100.0)

Instructions

Assignment #4: Gradients and Edges

Basic Description:

In the first part of this assignment you will program three functions. The first two functions will involve you differentiating an image in the x and y axis. The third function will be a much more generic function which will use cross correlation to apply a kernel to an input image (View lectures 2-05, 2-06 and 2-07 for a refresher on this).

In the second part of the assignment we will allow you to perform edge detection on an image. You will then describe the different things you attempted to get the edges of an image, and include your output in the PDF.

Part 1: Programming Gradients

The first part of the assignment requires you to program three functions. You may download the assignment files (zip file) that include the python template and a basic test file that will verify your functions output the correct type after you have written them (Note: simply because the tests succeed **does not mean** your function is doing the correct thing, you must verify this yourself, the test serves as a very basic check). Download the assignment files from T-Square Resources / Assignments / Assignment 4.

This zipped folder contains the following files in a folder labeled assignment4:

- **assignment4.py** - This is where you will write the code for the three functions we provide.
- **assignment4_test.py** - This can be used to test if your code is outputting the correct data type.
- **README** - Some basic instructions on how to run the unit test and what things you should and shouldn't do. We became a bit stricter after Assignment 2 in order to make grading smoother so please read this!
- **test_image.jpg** - A basic test image of a butterfly.

The function documentation details exactly what you need to do, but a basic description is provided here:

1. **imageGradientX**: This function takes in a grayscale image, and you must differentiate the image in the X direction. Make sure to take the absolute value of the difference. Refer to the function description for more details

details.

2. **imageGradientY:** This function takes in a grayscale image and you must differentiate the image in the Y direction (very similar to #1). Make sure to take the absolute value of the diffrence. Refer to the function description for more details.
3. **computeGradient:** This function applies a 3x3 input kernel to the input image. The output is the computed gradient for the image depending on the kernel used. Refer to the function description for more details.

Part 2: Edge Detection

For the second part of this assignment, we have an open-ended task. We want you to produce an edge detected image of an image of your choosing (choose / take an image that has clear edges to get better results). We do not care how you produce it, but if you want to get the most out of this assignment, here are some suggestions:

- Try using different kernels with your computeGradient function. After you do that, apply the convertToBlackAndWhite function you wrote in Assignment 2. See what results you get, and play with the threshold of 128, maybe the thresholding should be more / less depending on the image you use!
- Look into OpenCV's Canny Edges for classic edge detection results to compare to if you try the above so you get an idea as to how to get an image that simply displays the edges.

After you find a result that you like, add that to the PDF, and describe what you did to accomplish these results!

Please turn in the following files. **Keep the size limit of each file to 6MB**. If you need to compress your PDF before submitting, you can use http://smallpdf.com/compress-pdf:

- **assignment4.py**
  - Please put your name at the top of this assignment. Please do not include print statements, or statements like imread or imwrite in your file. You are free to do that to get your output but when you submit it, please comment these out as it makes grading a bit more difficult for us.
- **assignment4.pdf**
  - In the PDF we want you to write how you approached the first two functions (you can describe both of these together since they are very similar), and how you approached the third function. After this, we want you to include your edge-detected result for Part 2, and describe what you did to obtain this image.

Submitted Attachments

assignment4.py (7 KB; Sep 20, 2015 9:07 pm)

assignment4.pdf (320 KB; Sep 20, 2015 9:14 pm)

Additional instructor's comments about your submission

Programming: 75/75

- imageGradient X,Y: 30/30
- computeGradient: 45/45

Edge Detection: 15/15