cs512 Assignment 1 : Program Report

Pushpendra Singh Chauhan

Department of Computer Science

Illinois Institute of Technology

October 6, 2020

Abstract

This is a report for programming question in cs512 Assignment 1. It will contain the description of the problem I am trying to solve, the algorithms I employed to solve the problem, implementation details like program design issues and the results obtained. The results obtained will be analyzed for correctness. The performance of the algorithm will be evaluated and discussed.

1 Problem Statement:

- Write a program to perform simple image manipulation using openCV. The program should load an image by either reading it from a file or capturing it directly from a camera. When the user presses a key perform the operation corresponding to the key on the original image (not the result of the last processing step). The program should satisfy the following specifications :
- The image to be processed by the program should be either read from a file or captured directly from a camera. If a filename is specified, the image should be read from it. Otherwise the program should attempt to capture an image from a camera. When capturing an image from the camera, continue to capture and process images continuously.
- The read image should be read as a 3 channel color image.

- The program should work for any size image.

2 Proposed Solution:

- Manually converting the 3 channel color image (RGB) to gray scale using formula from Wikipedia: gray = R * 0.2126 + G * 0.7152 + B * 0.0722.
- Used Gaussian Smoothing for smoothing an image using openCV conversion function 'GaussianBlur'. Gaussian Smoothing is the result of blurring an image by a Gaussian function.
- Used Median filter for smoothing the image manually. Here, I convolve the image with an nXn Median filter where, 'n' is an odd number. To get the resultant image the same size of the original image I performed zero-padding on the original image.
- Used 'resize' conversion function of openCV for Downsampling the image by a factor of 2.
- I convolved the gray scale image using Central Difference derivative filter for obtaining x-derivative and y-derivative of the image. I manually implemented it in cython.
- Plotted the gradient vectors of the image using 'quiver plots' of matplotlib library and then embedded it in the openCV trackbar window which is used to control the number of pixels.
- Used 'getRotationMatrix2D' function of openCV for creating the rotation matrix. And then used 'warpAffine' function of openCV for applying 'Affine' transformation on the image using the rotation matrix in order to rotate the gray scale image by an angle of Θ degrees. Used trackbar window of openCV for controlling the value of Θ.
- Manually wrote the function in cython for finding the magnitude of the gradient normalized to the range [0,255]. The gradient is computed based on the x and y derivative of the image.

3 Implementation Details:

- The main issue I faced in program design is that to control the amount of smoothing, to control the amount of rotation of an image and to control the plotting of gradient vectors for every N pixel. I resolved them all using 'createTrackbar' function of openCV.
- Special Keys on the keyboard used to modify the displayed image are as follows:
- 'i' : reload the original image (i.e. cancel any previous peocessing)
- 'w': save the current (possibly processed) image into the file 'out.jpg'
- 'g' : convert the image to gray scale using the openCV conversion function
- 'G' : convert the image to gray scale using your implementation of conversion function
- 'c' : cycle through the color channels of the image showing a different channel every time the key is pressed
- 's' : convert the image to grayscale and smooth it using the openCV function. Use a track bar to control the amount of smoothing
- 'S' : convert the image to gray scale and smooth it using your function which should perform convolution with a suitable filter. Use a track bar to control the amount of smoothing
- 'd' : down sample the image by a factor of 2 without smoothing
- 'D' : down sample the image by a factor of 2 with smoothing
- 'x' : convert the image to gray scale and perform convolution with an x-derivative filter. Normalize the obtained values to the range [0,255]
- 'y' : convert the image to gray scale and perform convolution with a y-derivative filter. Normalize the obtained values to the range [0,255]
- 'm' : show the magnitude of the gradient normalized to the range [0,255]. The gradient is computed based on the x and y derivatives of the image
- 'p' : convert the image to gray scale and plot the gradient vectors of the image every N pixels and let the plotted gradient vectors have a length of K. Use a track bar to control N.

- ‘r’ : convert the image to gray scale and rotate it using an angle of Θ degrees. Use a track bar to control the rotation angle
- ‘h’ : display a short description of the program, its command line arguments and the keys it supports.
- ‘exit’ : exit the program
- Command Line instructions for using the program: (1). python main.py ‘image_filename.jpg’ (2). python main.py
- Folder structure: AS1 (main folder) →

  (1). src (sub-folder) – It contains source code files (‘main.py’, ’cython_smooth.pyx’, ‘setup.py’)

  (2). data (sub-folder) – It contains test files (images) and saved output file ‘out.jpg’ also resides in this sub-folder

  (3). doc (sub-folder) – It contains two pdf files (‘review questions answers.pdf’ and ‘program report.pdf’).

4 Results and discussion:

- As the size of smoothing filter in both ‘Gaussian filter smoothing’ and ‘Median filter smoothing’ increases the image becomes more blur.
- When we perform convolution with an x-derivative filter on the gray scale image then it detects edges in the x-axis.
- When we perform convolution with a y-derivative filter on the gray scale image then it detects edges in y-axis.

5 References:

- https://en.wikipedia.org/wiki/Gaussian_blur
- https://en.wikipedia.org/wiki/Median_filter
- https://docs.opencv.org/3.4/da/d6a/tutorial_trackbar.html

- https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_gui/py_trackbar/py_trackbar.html