

CS512 Assignment 2 Report

Akshay Jain

Department of Computer Science

Illinois Institute of Technology

February 17, 2022

Abstract

In this assessment, we have performed various operation on an image like noise and filtering, edge detection, corner detection etc.

1. Problem statement

In this assignment, our aim was to perform various transformations on images The program should load the image and convert it to grayscale and perform the respective operations on pressing following keys-

- a) 's' -- Smooth the image using OpenCV filter2D function and use a track bar to control the amount of smoothing.
- b) 'S' -- smooth the function using your own implementation of convolution function.
- c) 'D' -- downsample the image by a factor of 2
- d) 'U' -- upsample the image by a factor of 2
- e) 'x' -- compute the x-derivative
- f) 'm' -- compute the magnitude of image gradient
- g) 'p' -- plot the image gradient vectors
- h) 'c' -- detect corners in the image using the OpenCV Harris corner detection function
- i) 'C' -- detect corners in the image using your own implementation of the OpenCV Harris corner detection algorithm

2. Proposed solution

- Create a trackbar to get the amount of smoothing to be applied and use the OpenCV filter2D function with the kernel value from trackbar.
- Define a function in Cython file which takes the image and kernel as input arguments. Finally, convolve by multiplying and taking the sum.
- Use the above defined function to smooth and then downsample using OpenCV resize function. Do the same for upsampling (smooth after upsampling).

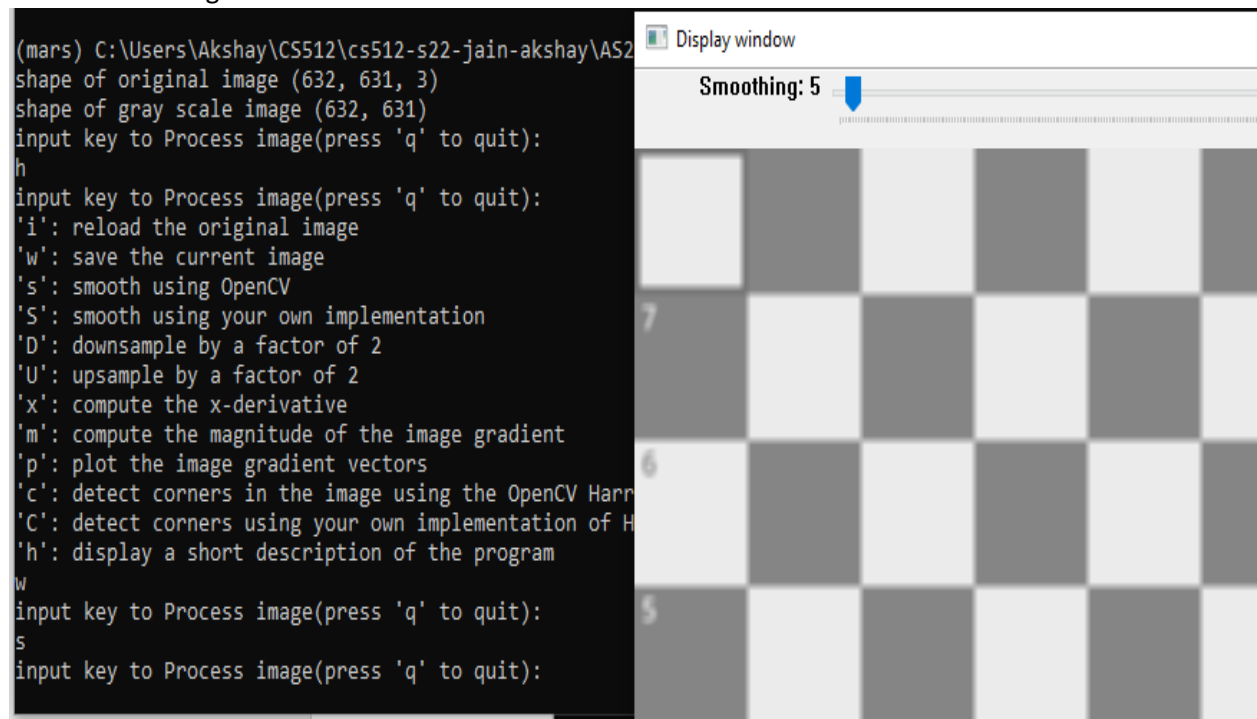
- Use the OpenCV Sobel function for computing the x-derivative and magnitude of vector gradients.
- Compute the angle using x and y derivative, and calculate the gradients for each region. Finally, plot the arrows using arrowedLine function of OpenCV.
- Use the OpenCV cornerHarris function.

3. Implementation details

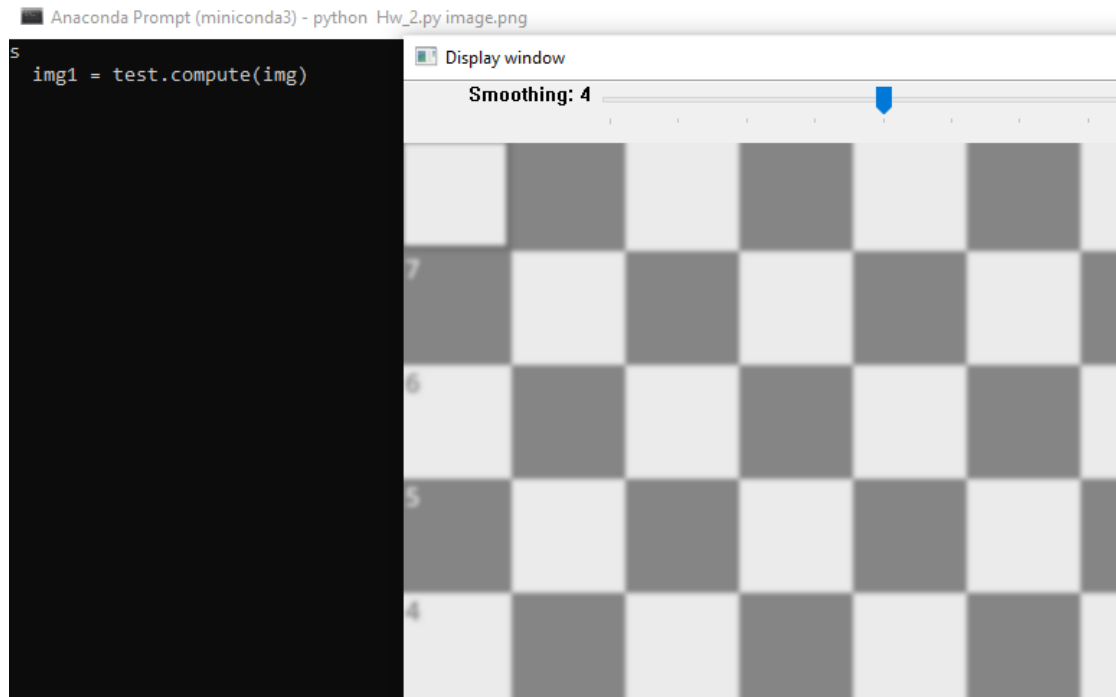
- The way my own smoothing function needs to be optimized. Got operand type errors due to which I had to use numpy sum and multiply functions instead of using four for loops.
- Had to refer various sources to get to display arrows of the gradient vectors.
- I have used images from the source directory itself.

4. Results and discussions

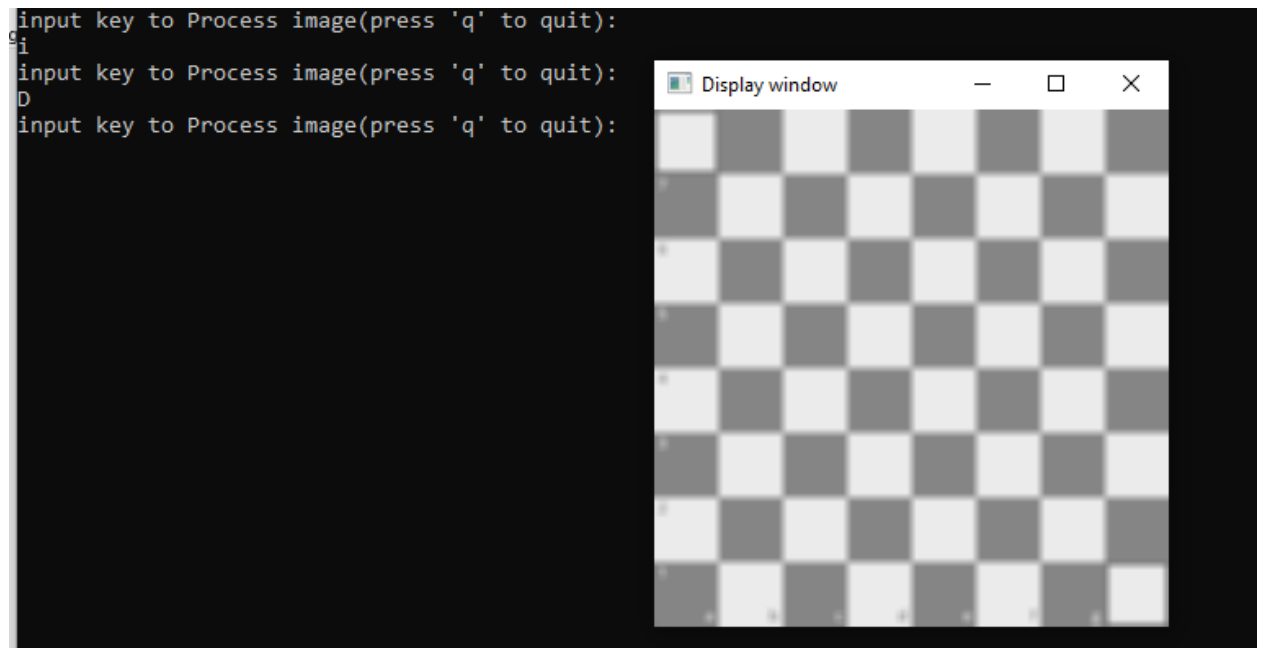
- Using 's' -- Smooth the image using OpenCV filter2D function and use a track bar to control the amount of smoothing.



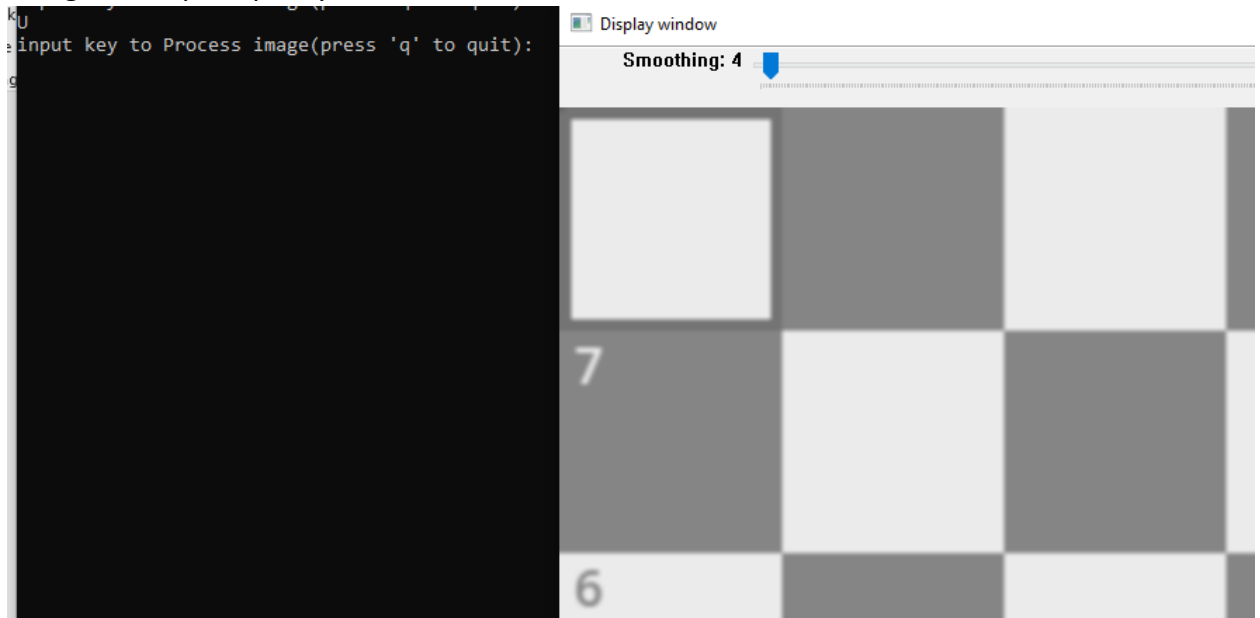
- Using 'S' – smooth the function using your own implementation of convolution function.



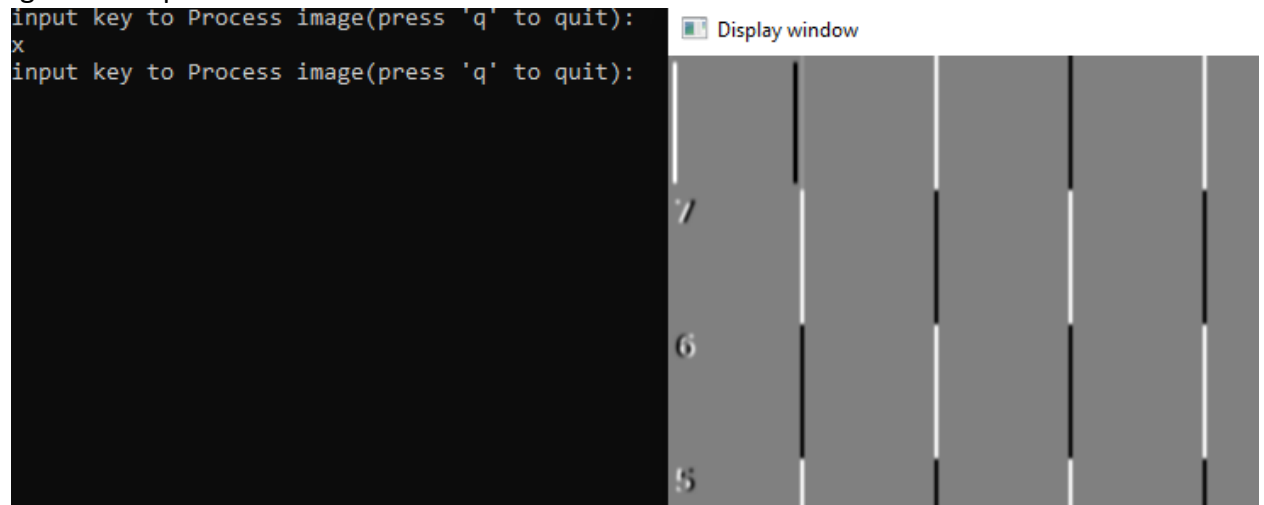
- Using 'D' to downsample by a factor of 2



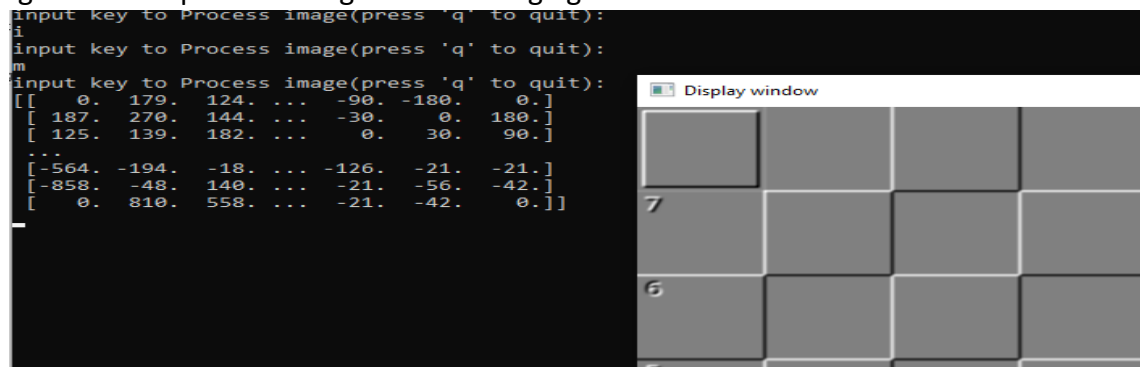
- Using 'U' to upsample by a factor of 2



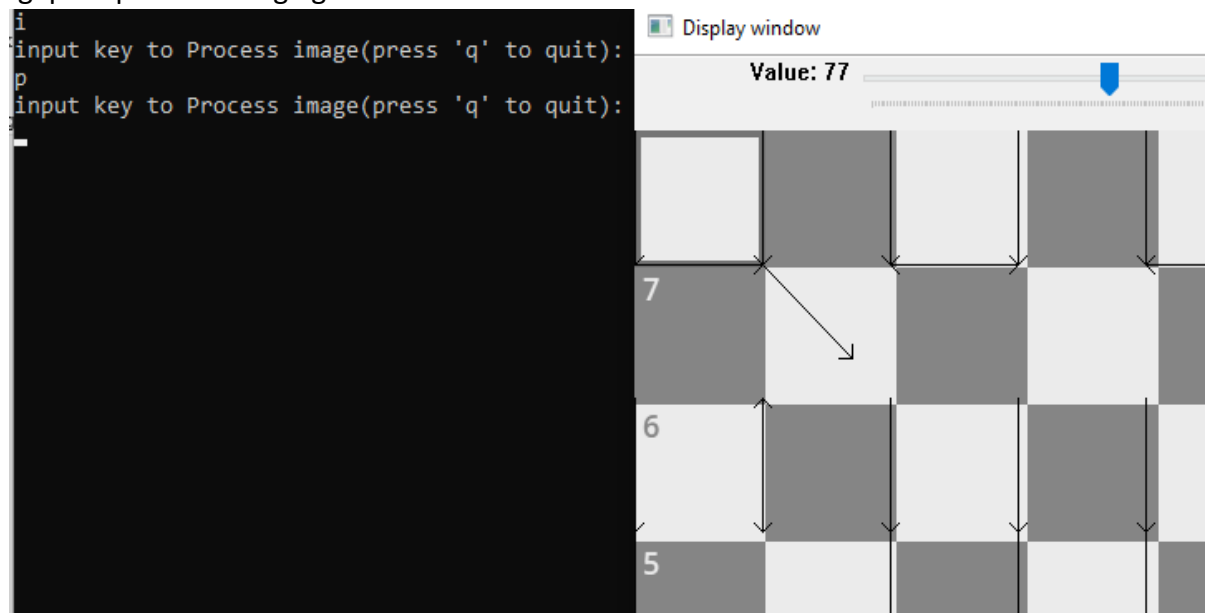
- Using 'x' to compute the x-derivative



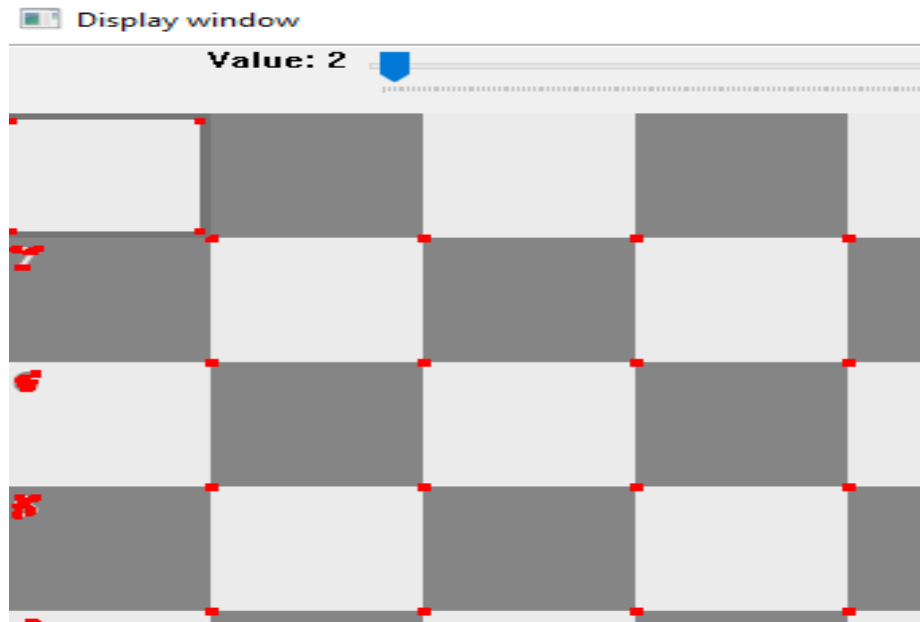
- Using 'm' to compute the magnitude of image gradient



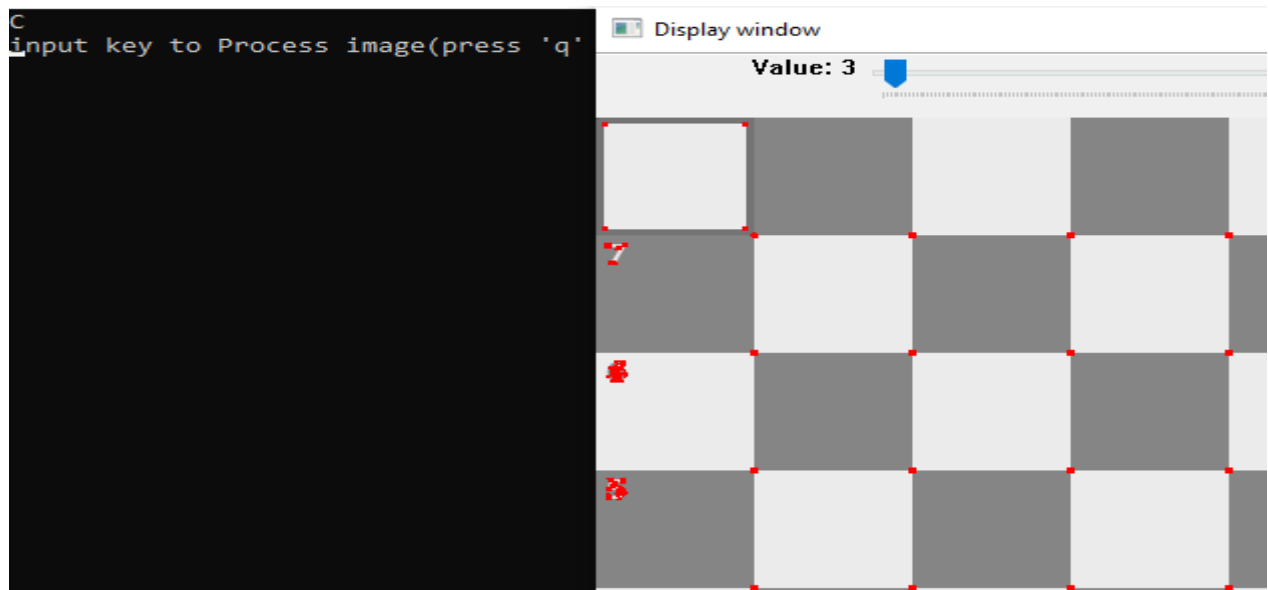
- Using 'p' to plot the image gradient vectors



- Using 'c' to detect corners in the image using the OpenCV Harris corner detection function



- Using 'C' to detect corners in the image using your own implementation



5. References

- <https://towardsdatascience.com/use-cython-to-get-more-than-30x-speedup-on-your-python-code-f6cb337919b6>
- https://docs.opencv.org/3.4/d6/d00/tutorial_py_root.html