CP467

**Digital Image Processing**

**Assignment 2**

**Jashandeep Singh**

sing8282@mylaurier.ca

169018282

# Image Convolution from Scratch

|  |  |  |
| --- | --- | --- |
|  |  | |
| Source Image |
|  | | |
|  |  |  |
| Task 1A: Average Smoothing Filter | Task 1B: Gaussian Smoothing | Task 1C: Sobel Sharpening Filter |
|  | | |
|  |  |  |
| Task 2A: Average Smoothing Filter | Task 2B: Gaussian Smoothing | Task 2C: Sobel Sharpening Filter |

When comparing the results of the average smoothing filter, both the custom implementation and the OpenCV built-in function produce visually identical outputs. This observation also applies to the Gaussian filter, where the results from the self-implemented version are same as that of those generated by OpenCV's built-in function.

However, a significant difference is observed in terms of runtime performance. OpenCV's optimized implementations are considerably faster, while low-level custom implementations are slower. This could be due the lack of optimized routines and potential overhead from manual computation.

On comparing the results of self-implemented Sobel sharpening filter and that of OpenCV’s built in function, a significant difference in observer in the two images. Again, the OpenCV’s version tends to faster than former one. This discrepancy in output could be due to differences in algorithmic details, such as kernel design or precision optimizations within OpenCV.

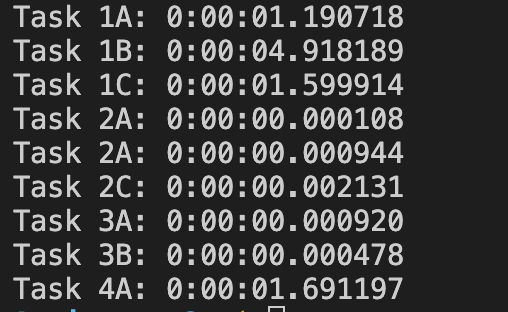


Figure : Run time of tasks

# Edge Detection

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Source Image | Task 3A: Marr-Hildreth edge detector | Task 3B: Canny edge detector |

The results produced by the both edge detector i.e., Marr-Hildreth and Canny edge detector, are different. These differences arise from the because of the distinct approaches used by each algorithm. The Marr-Hildreth method relies on the Laplacian of Gaussian (LoG) to detect edges, which often results in broader, less precise edge boundaries. In contrast, the Canny edge detector uses a multi-stage process involving gradient calculation, non-maximum suppression, and edge tracking, producing sharper, more accurate edges.

# Group Adjacent Pixels

|  |  |
| --- | --- |
|  |  |
| Source Image | Task 4A: Group Adjacent Pixels |