

Digital Image Processing

Assignment 2

Jashandeep Singh

sing8282@mylaurier.ca

169018282

1. 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 to 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 is observed in the two images. Again, the OpenCV's version tends to be faster than the former one. This discrepancy in output could be due to differences in algorithmic details, such as kernel design or precision optimizations within OpenCV.

```
Task 1A: 0:00:01.190718
Task 1B: 0:00:04.918189
Task 1C: 0:00:01.599914
Task 2A: 0:00:00.000108
Task 2A: 0:00:00.000944
Task 2C: 0:00:00.002131
Task 3A: 0:00:00.000920
Task 3B: 0:00:00.000478
Task 4A: 0:00:01.691197
```

Figure 1: Run time of tasks

2. Edge Detection



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.

3. Group Adjacent Pixels

