ECE 6310 – **Lab1** Name: **Atef Emran**

Convolutions, Separable Filters and Sliding Windows

Problem Statement

In this project implement three versions of a 7x7 mean filter. The first version should use basic 2D convolution. The second version should use separable filters (1x7 and 7x1). The third version should use separable filters and a sliding window.

Any pixel for which the convolution extends outside the image boundary "edge cases" should be given an output value of zero.

All three versions of the filter should produce the exact same output. This must be verified by comparing the images and showing the method used and result. Also, each version should be timed to be compared with the other versions.

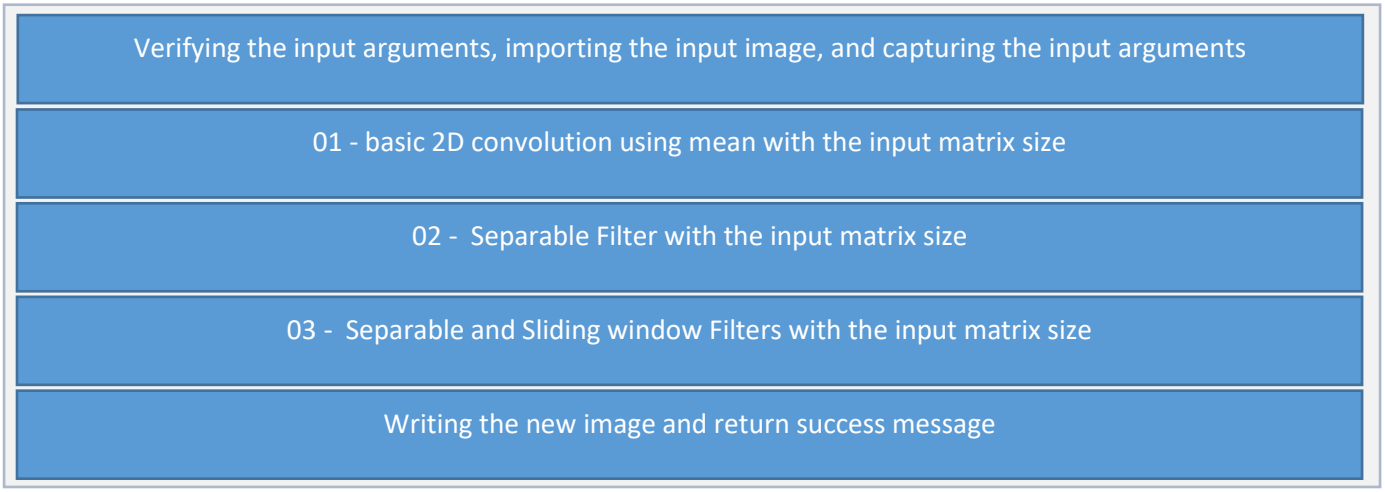
Program / Code Summary

The program brings .ppm image, reads it, and smooth/filter it using one of three differing smoothing techniques – based on the user input -, and returns the filtered/smoothed image.

The input arguments are: [0-excutable_file] [1-image_to_be_filtered] [2-filter_size] [3-filtering_method]

Where filtering methods: (1) basic 2D convolution using mean, (2) Separable Filter, (3) Separable and Sliding Window Filters

The **code structure** is as below:



Results

The expected result is a smoothed image which has a black border with width = rounddown(matrix size / 2); because for the border pixels, the convolution matrix will extend outside the image and will be given value of zero as per the requirements.

Original

Filtered image

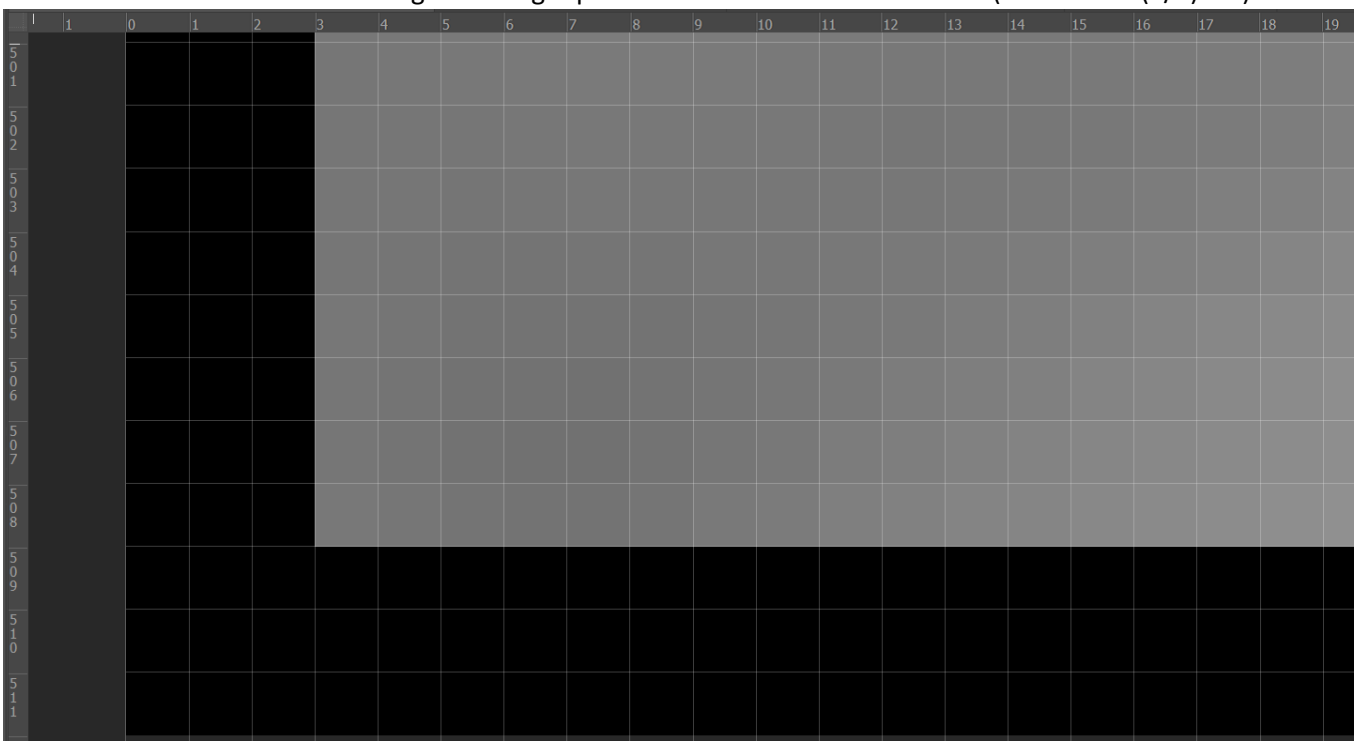
The resulted image from each filter method is as below:

01_filtered_2D_convolution

02_separable_filters

03_separable_and_sliding_window_filters

Zoom-in on the corners of the image showing 3-pixels black borders for 7x7 matrix (rounddown(7/2) = 3)



Comparing the results

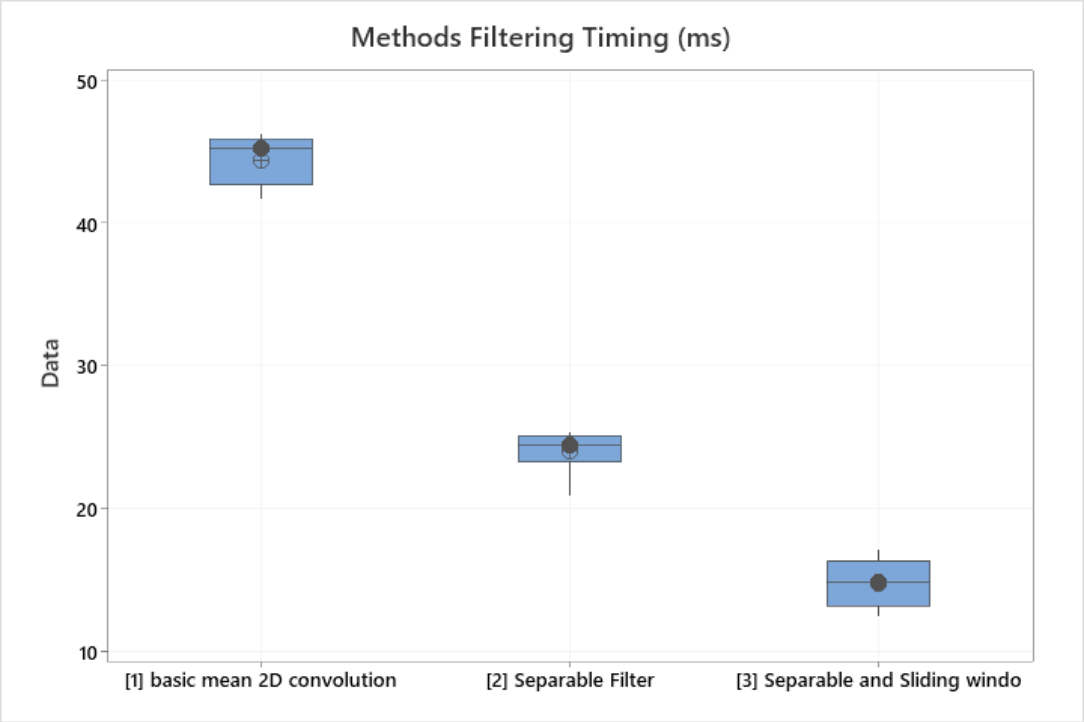
To compare the different results to each other, the compare.c was created to compare between these files. The program reads the two input images, and compare them pixel by pixel, and outputs the different pixels' coordinates or "Identical Message".

The result of comparing the first with second and first with third image show identical images as below:

```
PS E:\CU-ICAR Automotive Engineering Masters\3rd Semester\Computer vision\Labs\Lab 1> .\compare.exe .\01_filtered_2D_convolution.ppm .\02_separable_filters.ppm
Identical! Total number of identical pixels = 262144, and total number of pixels in first pic are 262144
PS E:\CU-ICAR Automotive Engineering Masters\3rd Semester\Computer vision\Labs\Lab 1> .\compare.exe .\01_filtered_2D_convolution.ppm .\03_separable_and_sliding_window_filters.ppm
Identical! Total number of identical pixels = 262144, and total number of pixels in first pic are 262144
PS E:\CU-ICAR Automotive Engineering Masters\3rd Semester\Computer vision\Labs\Lab 1>
```

Timing Data

Each filtering method was timed to be compared with the other filtering methods; the box plot below shows the summary of the 10 runs timing for each of the methods.



	[1] basic mean 2D convolution	[2] Separable Filter	[3] Separable and Sliding Window
Medians (ms)	45.22	24.49	14.84

As shown the timing dramatically decreases when using separable filter compared to the basic 2D convolution as the large mathematical calculations with the nested loops are avoided in this technique. Also, when combining the sliding window with the separable filter there is further improvement in the processing time due to the further logic simplicity.

	ns			ms		
run	1	2	3	1	2	3
1	45695079	24476694	16195122	45.70	24.48	16.20
2	41683584	23414200	16794699	41.68	23.41	16.79
3	43118888	23733459	14631104	43.12	23.73	14.63
4	44975734	25369037	12452481	44.98	25.37	12.45
5	46252783	25056933	13542144	46.25	25.06	13.54
6	45834858	20853981	15051454	45.83	20.85	15.05
7	42928171	25302011	12454515	42.93	25.30	12.45
8	45459988	24510156	13454545	45.46	24.51	13.45
9	46170232	24759905	17155465	46.17	24.76	17.16
10	41938886	22900387	15055155	41.94	22.90	15.06
	Median			**45.22**	**24.49**	**14.84**

Instructions

To run the program, create executable file first using your compiler, then in the terminal run it as below:

[0-excutable_file] [1-image_to_be_filtered] [2-filter_size] [3-filtering_method]

Example: .\lab1.exe bridge.ppm 7 3, which will use the image bridge.ppm as input to be filtered, with 7x7 matrix and using Separable and Sliding window Filters. Result will be file named with the selected filter in the directory.

To compare the files:

[0-excutable_file] [1-first_image] [2-second_image]

Example: .\compare.exe 01_filtered_2D_convolution.ppm 03_separable_and_sliding_window_filters.ppm