ECE 6310 Introduction to Computer Vision

Lab 1 – Convolution, Separable Filters, Sliding Windows

Prajval Vaskar

C20664702

September 1, 2020

CLEMSON®
UNIVERSITY

In this project each student must 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.

A. Convolution using basic 2D 7x7 mean filter

Image convolution can be calculated using following formula

$$O[r,c] = \sum_{dr=-W}^{+W} \sum_{dc=-W}^{+W} I[r+dr, c+dc] * f[dr, dc]$$

Equation 1 Basic 2D convolution

7x7 mean filter is shown below

$$f = \begin{bmatrix} 1/49 & 1/49 & 1/49 & 1/49 & 1/49 & 1/49 & 1/49 \\ 1/49 & 1/49 & 1/49 & 1/49 & 1/49 & 1/49 & 1/49 \\ 1/49 & 1/49 & 1/49 & 1/49 & 1/49 & 1/49 & 1/49 \\ 1/49 & 1/49 & 1/49 & 1/49 & 1/49 & 1/49 & 1/49 \\ 1/49 & 1/49 & 1/49 & 1/49 & 1/49 & 1/49 & 1/49 \\ 1/49 & 1/49 & 1/49 & 1/49 & 1/49 & 1/49 & 1/49 \\ 1/49 & 1/49 & 1/49 & 1/49 & 1/49 & 1/49 & 1/49 \end{bmatrix}$$

For any edge cases, output should be the value of zero as asked. For this nothing has to be done because we are allocating memory to the image using calloc () function which initializes the requested memory to zeros so after allocating memory we get an array of ROWS*COLS of all zeros.

Result of the image

Figure 1: Original provided image and image after applying mean filter 2-D convolution

B). Separable filters (1x7 and 7x1)

In separable filters. We have to apply vertical and horizontal filter separately meaning first apply vertical or horizontal filter on provided image and then apply remaining filter on the image generated by the first filter. We can apply any filters first because of the associativity of the convolution.

$$O_1[r,c] = \sum_{dc=-W}^{+W} I[r,c+dc] * f_c[dc]$$
$$O[r,c] = \sum_{dr=-W}^{+W} O_1[r+dr,c] * f_r[dr]$$

Equation 2 Separable convolution using vertical and horizontal filter

Separable filters are shown below

$$fc = \begin{bmatrix} 1/7 \\ 1/7 \\ 1/7 \\ 1/7 \\ 1/7 \\ 1/7 \\ 1/7 \end{bmatrix}$$

$$fr = [1/7 \quad 1/7 \quad 1/7 \quad 1/7 \quad 1/7 \quad 1/7 \quad 1/7]$$

Result of the image

Figure 2: Original provided image and image after applying mean filter convolution using separable filters

C). Separable filter with sliding window

This can be done by combining separable filter and sliding window. As we know during convolution the mean filter running in for loop with increment of one.

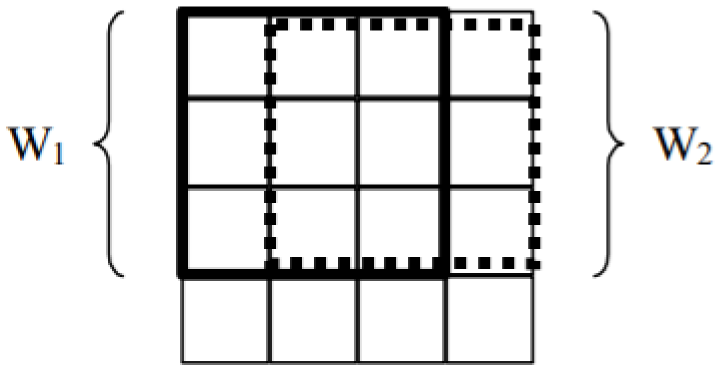

Figure 3 Separable filters

In the above window, we can see the difference between first convolution operation with dark border filter and successive convolution by dotted border filter. So, for the successive operation we can use the value of the intersection of 2 filters from the previous convolution and calculate only the last column values. This will save computational effort and time.

Result of the image

Figure 3: Original provided image and image after applying mean filter convolution using separable filters

Summary

For the conclusion part, similarity of the output image is compared using FC function on windows machine.

Command for comparing the image comparison in command prompt

FC ‘file_name’ ‘file_name’

1. Comparing the basic 2D convolution and separable filters convolution output image.

```
C:\AuE Fall 20\computer vision\Assignment1>FC smoothed_2D_filter.ppm smoothed_1D_filter.ppm
Comparing files smoothed_2D_filter.ppm and SMOOTHED_1D_FILTER.PPM
FC: no differences encountered
```

FC returns no difference encountered which means the output image is same for both the cases.

2. Comparing the basic 2D convolution and separable filters with sliding window convolution output image.

```
C:\AuE Fall 20\computer vision\Assignment1>FC smoothed_2D_filter.ppm smoothed_1D_sliding_filter.ppm
Comparing files smoothed_2D_filter.ppm and SMOOTHED_1D_SLIDING_FILTER.PPM
FC: no differences encountered
```

FC returns no difference encountered which means the output image is same for both the cases.

From above we can say that all three-image output are similar and output image is same in all cases.

Time in nanoseconds for each method

Sr No.	Time for basic 2D convolution	Time for separable 1D convolution	Time for separable 1 D with sliding window convolution
1	24802100	12230800	4041900
2	24867500	12200800	4097000
3	24857100	12270600	4091400
4	25175400	12250400	4019400
5	24808100	12257200	3991800
6	24938900	12238400	4030900
7	24889400	12248200	4018700
8	24927000	12265200	4037500
9	24837800	12241900	4098800
10	24931900	12246800	4032700
Avg.	24903520 ns	12245030 ns	4046010 ns
Avg.	0.02490352 s	0.01224503 s	0.00404601 s

Table 1 Comparison of time required for each program in nanoseconds

Table shows the time required to run the code in nanoseconds. After finding average time, time required for separable convolution with sliding window is less than time required for separable 1D convolution. Time required for basic 2D convolution is highest of all the three methods.