Ashutosh Sharma (190100027)
Kritti Sharma (180100060)
Shreyas Chandgothia (180260037)

Homework Assignment 2
CS 663: Digital Image Processing

2021-08-31

Exercise-5

We are given an image f of size $n \times n$. Consider the following mean filter, M, of size $((2a+1) \times (2a+1))$:

$$M = \frac{1}{(2a+1)^2} \begin{bmatrix} 1 & 1 & 1 & 1 & . & . & 1 \\ 1 & 1 & 1 & 1 & . & . & 1 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 1 & 1 & 1 & 1 & . & . & 1 \end{bmatrix}$$

Now, we convolve the image f with the mean filter M, which yields f_1. Then we convolve the resultant image f_1 with the mean filter M again, which yields f_2. We carry on these iterations k times, where f_{k-1} is convolved with the mean filter M, which yields f_k.
Mathematically:

$$f_1 = M * f$$
$$f_2 = M * f_1 \quad = M * (M * f)$$

Since the convolution operation is associative, we can write this as follows:

$$f_2 = M * f_1 = (M * M) * f$$

So, when the convolution operation is applied k times, we have:

$$f_k = M * f_{k-1} = (M * M * .... * M) * f$$

where M is convolved k times.

Thus, we can indeed express f_k in terms of a convolution of f with a kernel K, given by $K = (M * M * ... * M)$, where M is convolved k times. Thus K is expressed here as a single kernel equivalent k-fold application of the mean convolution operator.

Important Points to Note about the Above Constuction:

- We cannot obtain an analytical closed-form expression for the matrix entries of the kernel K. It can only be expressed numerically.
- The resultant convolutional kernel K will give the correct answer, equivalent to the answer obtained from the k-fold operation of the convolutional filter M, only if we retain the padded non-zero values on each iteration. That is to say, that at any iteration i, when we calculate the convolution of $M * M * ...M$ ($i-1$ times) with M, we must treat both filters as having infinite extent with padded zeroes. And after obtaining the resulting convolution given by $M * M * ...M$ (i times), we crop out only the zero rows and columns to get a finite-sized kernel, which would be larger in size than the previous iteration's kernel.
- The previous point implies that if M was of size $((2a+1) \times (2a+1))$, then each iteration will add a total of a rows, both to the top and bottom, as well as a columns both to the left and the right. Thus the size of the kernel will increase on each iteration. $M * M$ will be of size $((4a+1) \times (4a+1))$, $M * M * M$ will be of size $((6a+1) \times (6a+1))$ and $M * M * .... * M$ (k times) will be of size $((2ka+1) \times (2ka+1))$.
- At each iteration i, we must re-normalize the resulting convolutional kernel $M * M * ...M$ (i times), by the sum of its entries. We cannot simply multiply the normalization factors of the two kernels being used as the input to the convolution operation (i.e. $M * M * ...M$ ($i-1$ times) and M).
- In practice, the mean filter, when being applied at the edges and corners of an image, is different from when being applied in the middle of the image, due to the finite nature of the image (Since we do not take the padded zeroes to be having any weight at the edges and corners). If we want to incorporate this practical effect, then even the 1-fold mean filter M cannot be expressed as a convolutional matrix, since it's no longer space invariant. Thus, the above theoretical analysis in the question ignores this practical consideration.