Homework #3

PDF version due Mon. November 21 11:59pm through CatCourses

1) Let W be the following 3x3 spatial filter

$W =$

1	1	1
0	0	0
-1	-1	-1

and let F be the following 6x6 image

$F =$

1	1	1	0	0	0
1	1	1	0	0	0
1	1	1	0	0	0
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1

(a) Compute G, the result of applying the filter W to the image F using standard spatial filtering:

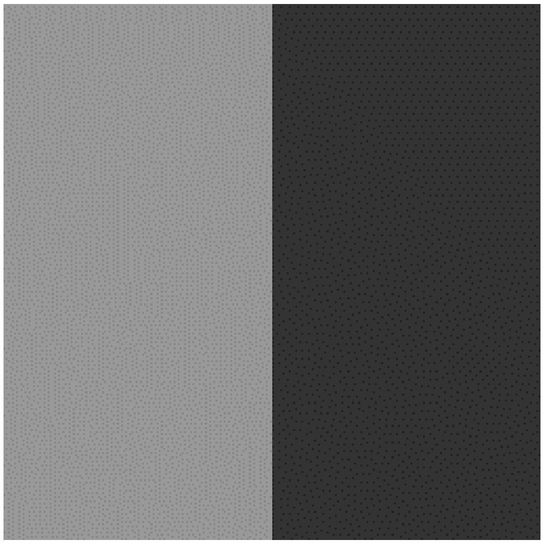
$$G(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} W(s,t) F(x+s, y+t)$$

G should have the same size F. Assume the zero-padding approach is used to deal with the case when the filter extends past the edge of the image.

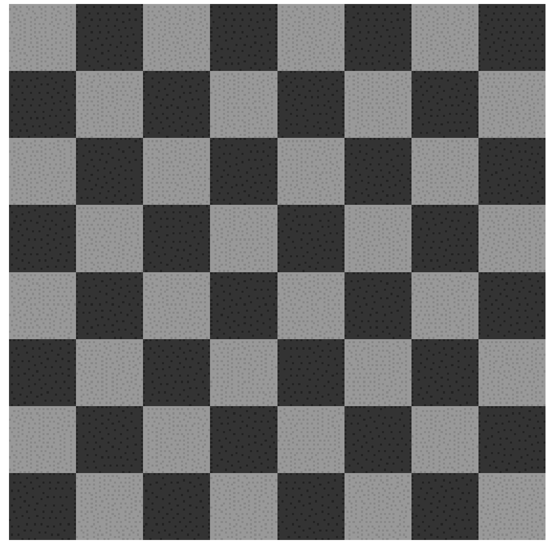
(b) Give a brief description of what the spatial filter W does (in the general case, not just when applied to the image above). What does it "detect"?

(c) Describe how your result G above demonstrates this.

2) Consider the two grayscale images below:



(a)



(b)

They are both 256x256 pixels in size. Image (a) has intensity value 150 in its left half and intensity value 50 in its right half. Image (b) is a checkerboard with squares measuring 32x32 pixels, half of intensity value 150 and the other half of intensity value 50.

(a) Do these two images have the same histogram? Briefly state why or why not.

(b) Now, suppose you perform histogram equalization on the two images. Do the resulting images have the same histogram? Briefly state why or why not.

(c) Suppose you blur each of the original images with a 3x3 averaging mask. Do the resulting images have the same histogram? Briefly state why or why not.

3) In this problem, you will investigate a more efficient way to implement spatial filtering when all the filter coefficients have the same value. The motivation comes from the observation that as you slide the filter one pixel at a time over the image and compute the sum-of-products of image and filter values, you can use the results from the previous computation in the current computation.

Although the method can be generalized, we will consider the case in which all the filter coefficients have the value 1. And, we will also ignore the $1/n^2$ scaling factor that typically accompanies an averaging filter of size nxn.

(a) Describe the algorithm you would use to compute the output value at location (x,y) given that you have already computed the result for location (x-1,y), for example, for an averaging filter of size nxn (think about what changes when you shift the filter by one pixel).

(b) How many additions (in terms of n) does this require for each output pixel. Count subtractions as additions.

(c) Now, let's compare this with the standard approach of not using previous results. How many additions are required for each output pixel in this case. This should be in terms of n.

(d) Compute the *computational advantage* of the more efficient approach. This is simply the ratio of the number of additions required by the standard approach to the number of additions required by the more efficient approach. Again, this should be in terms of n.