Homework #3
SOLUTION

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.

SOLUTION

$G =$

-2	-3	-2	-1	0	0
0	0	0	0	0	0
2	3	1	-1	-3	-2
2	3	1	-1	-3	-2
0	0	0	0	0	0
0	0	1	2	3	2

(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"?

SOLUTION

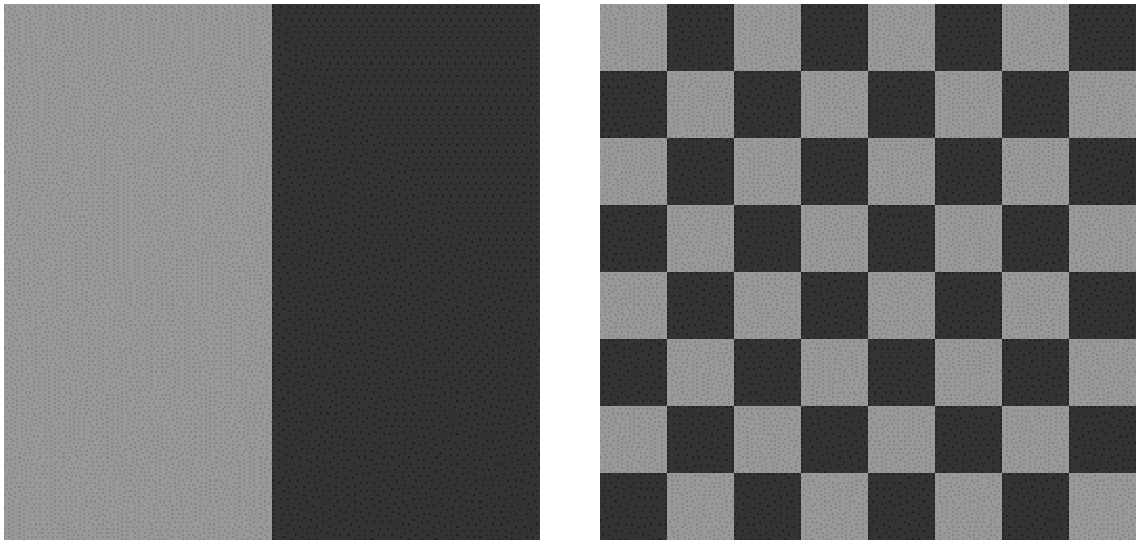
Detects large intensity changes in the vertical direction.

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

SOLUTION

The result in G has large (+ or -) values at the transition between 0 and 1 in the vertical direction.

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.

SOLUTION

Yes, they have the same histogram. They have the number of pixels with intensity value 50 and the same number of pixels with intensity value 150 and they have no other pixels. Thus, their histograms will have only two non-zero values, the count of the number of pixels with intensity value 50 and the count of the number of pixels with intensity value 150. And, these two counts will be equivalent. (Not that both their unnormalized histograms and normalized histograms are equal.)

(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.

SOLUTION

Yes, the resulting images will have the same histogram. Since the transformation involved in histogram equalization depends only on an image's histogram, and since the original images have the same histogram, then the transformations for image (a) and image (b) will be equal. Histogram equalization maps a pixel's intensity value regardless of its spatial location. So the pixels with intensity value 50 will be mapped to the same intensity value in both images. The same is true for pixels with intensity value 150. Thus, there will be same number of pixels of a particular intensity in the transformed images and their histograms will again be the same.

(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.

SOLUTION

No. To answer this, we focus on the boundaries between the regions of different intensity in the images. It is here that averaging mask will result in pixels with intensity values other than 50 or 150. Since the total boundary in image (a) is less than the total boundary in image (b) there will be fewer pixels with these other values in image (a). Thus the histograms of the blurred images will be different. Also note that the boundary in image (a) will result in pixels that have intensity value 83.3 or 116.7 (ignoring image boundary effects). However, the boundaries in image (b) will also result in pixels with intensity value 94.4 or 105.6.

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.

SOLUTION 1

(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).

SOLUTION

Let G_{old} be the result already computed for location (x-1,y). To compute G_{new} at location (x,y) using G_{old}, we need to subtract the sum of pixels under the first column of the mask before it was moved (call this sum C_1) and add the sum of pixels under the last column of the mask after it was moved (call this sum C_n). So,

$$G_{new} = G_{old} - C_1 + C_n$$

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

SOLUTION

We have already computed C_1. To compute C_n requires n-1 additions. Thus, to compute G_{new} from G_{old} requires **n+1** 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.

SOLUTION

The standard approach requires $\mathbf{n^2-1}$ additions for each output pixel.

(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.

SOLUTION

The computational advantage is

$$\frac{n^2-1}{n+1} = \frac{(n-1)(n+1)}{n+1} = n-1$$

ALTERNATE SOLUTION

(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).

SOLUTION

Let G_{old} be the result already computed for location (x-1,y). To compute G_{new} at location (x,y) using G_{old}, we need to subtract the sum of pixels under the first column of the mask before it was moved (call this sum C_1) and add the sum of pixels under the last column of the mask after it was moved (call this sum C_n). So,

$$G_{new} = G_{old} - C_1 + C_n$$

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

SOLUTION

It takes n-1 additions to compute C_1. To compute C_n also requires n-1 additions. Thus, to compute G_{new} from G_{old} requires **2n** 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.

SOLUTION

The standard approach requires $\mathbf{n^2-1}$ additions for each output pixel.

(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.

SOLUTION

The computational advantage is

$$\frac{n^2 - 1}{2n}$$