جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

قسم الامن السيبراني

Department of Cyber Security

Subject:

Image Processing

Class:

Third

Lecturer:

Asst. Prof. Dr. Ali Kadhum Al-Quraby

Lecture: (3)

The Convolution Process

3.The Convolution Process:

this method requires a mathematical process to enlarge an image. This method required **two** steps:

1. Extend the image by adding rows and columns of zeros

between the existing rows and columns. The image is extended as follows:

Original Image Array

$$\begin{pmatrix} 3 & 5 & 7 \\ 2 & 7 & 6 \\ 3 & 4 & 9 \end{pmatrix}$$

Image extended with zeros

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 5 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 7 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 4 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

2. Perform the convolution.

Next, we use convolution mask, which is slide a cross the extended image, and perform simple arithmetic operation at each pixel location .

$$\begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}$$

The convolution process requires us to overlay the mask on the image, multiply the coincident values and sum all these results. This is equivalent to finding the vector inner product of the mask with underlying sub image. The vector inner product is found by overlaying mask on subimage. Multiplying coincident terms, and summing the resulting products.

For example, if we put the mask over the upper-left corner of the image, we obtain (from right to left, and top to bottom):

1/4(0) +1/2(0) +1/4(0) +1/2(0) +1(3) +1/2(0) + 1/4(0) +1/2(0) +1/4(0)

=3

Note that the existing image values do not change. The next step is to slide the mask over by on pixel and repeat the process, as follows:

1/4(0) +1/2(0) +1/4(0) +1/2(3) +1(0) +1/2(5) + 1/4(0) +1/2(0) +1/4(0)

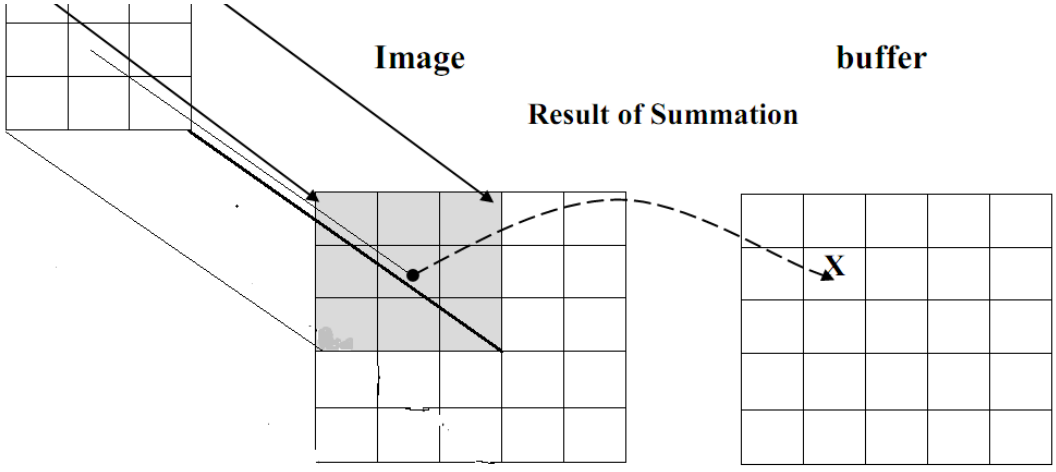
=4

Note this is the average of the two existing neighbors. This process continues until we get to the end of the row, each time placing the result of the operation in the location corresponding to center of the mask.

When the end of the row is reached, the mask is moved down one row, and the process is repeated row by row. This procedure has been performed on the entire image, the process of sliding, multiplying and summing is called convolution.

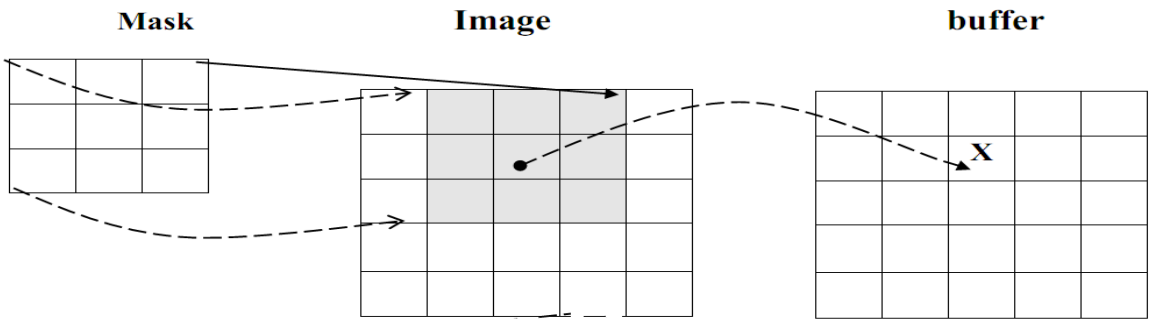
Note that the output image must be put in a separate image array called a buffer, so that the existing values are not overwritten during the convolution process.

a. Overlay the convolution mask in the upper-left corner of the image. Multiply coincident terms, sum, and put the result into the image buffer at the location that corresponds to the masks current center, which is (r,c)=(1,1).

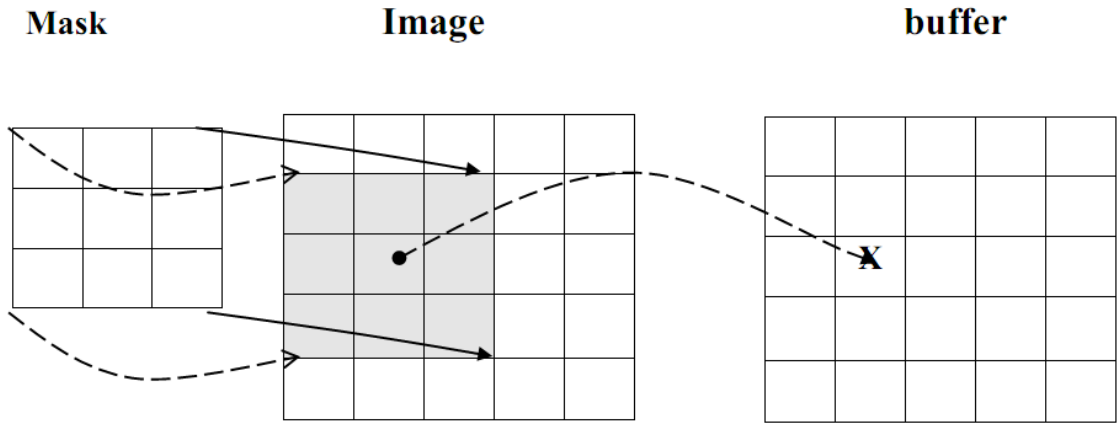


b. Move the mask one pixel to the right , multiply coincident terms sum , and place the new results into the buffer at the location that corresponds to the new

center location of the convolution mask which is now at (r,c)=(1,2), continue to the end of the row.



c. Move the mask down on row and repeat the process until the mask is convolved with the entire image. Note that we lose the outer row(s) and columns(s).



At this point a good question would be Why we use this convolution method when it require, so many more calculation than the basic averaging of the neighbors method?

The answer :is that many computer boards can perform convolution in hardware, which is generally very fast, typically much faster than applying a faster algorithm in software.

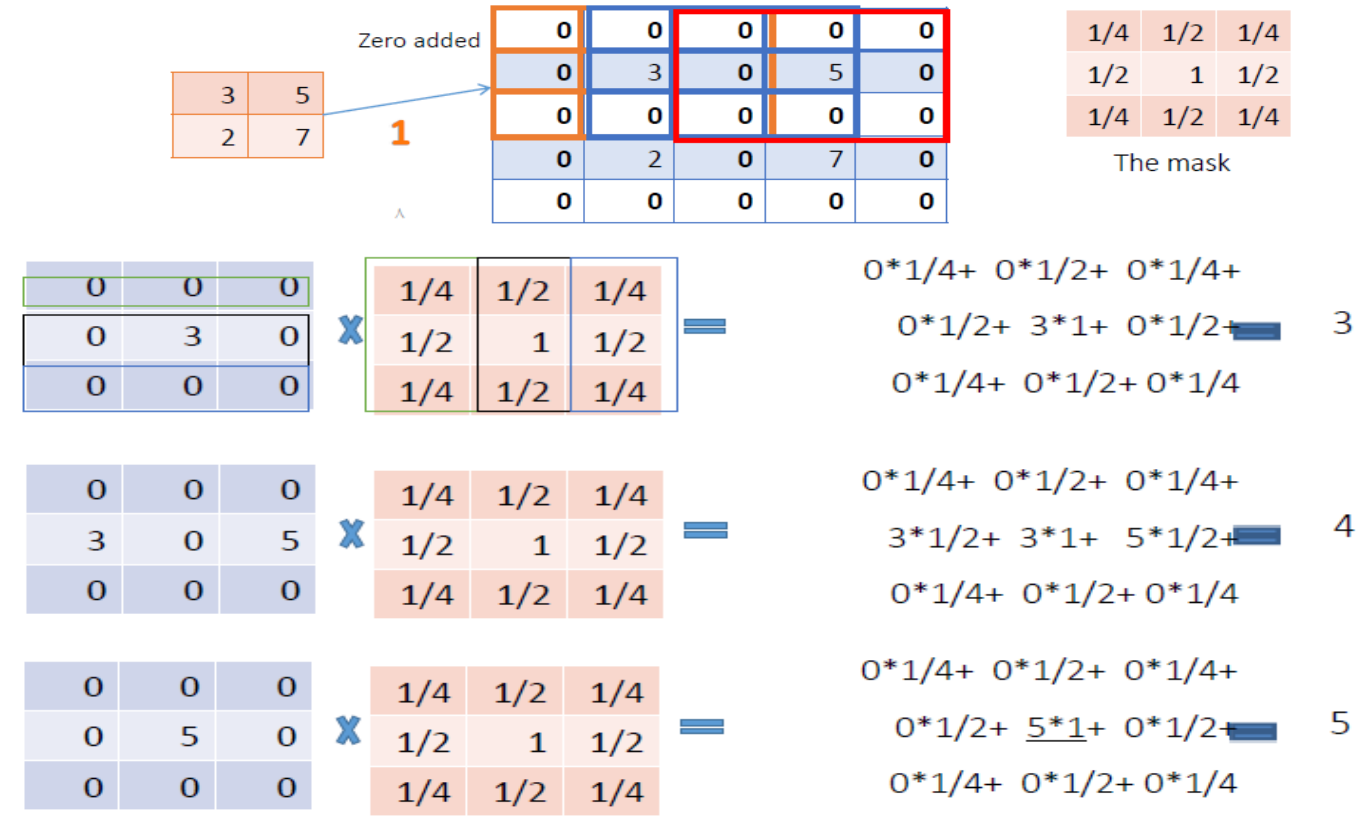
Not only first-order hold be performed via convolution, but zeroorder hold can also achieved by extending the image with zeros and using the following convolution mask.

Zero-order hold convolution mask

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Note that for this mask we will need to put the result in the pixel location corresponding to the lower-right corner because there is no center pixel.

Example: Convolution



0	0	0	0	0
0	3	0	5	0
0	0	0	0	0
0	2	0	7	0
0	0	0	0	0

0	3	0	x	1/4	1/2	1/4	=
0	0	0		1/2	1	1/2	
0	2	0		1/4	1/2	1/4	

0*1/4+ 3*1/2+ 0*1/4+
0*1/2+ 0*1+ 0*1/2+ = 2.5
0*1/4+ 2*1/2+ 0*1/4

3	0	5	x	1/4	1/2	1/4	=
0	0	0		1/2	1	1/2	
2	0	7		1/4	1/2	1/4	

3*1/4+ 4*1/2+ 5*1/4+
0*1/2+ 0*1+ 0*1/2+ = 4.25
2*1/4+ 0*1/2+ 7*1/4

4	5	0	x	1/4	1/2	1/4	=
0	0	0		1/2	1	1/2	
0	7	0		1/4	1/2	1/4	

4*1/4+ 5*1/2+ 0*1/4+
0*1/2+ 0*1+ 0*1/2+ = 6
0*1/4+ 7*1/2+ 0*1/4

0	0	0	0	0
0	3	0	5	0
0	0	0	0	0
0	2	0	7	0
0	0	0	0	0

0	0	0	x	1/4	1/2	1/4	=
0	2	0		1/2	1	1/2	
0	0	0		1/4	1/2	1/4	

0*1/4+ 0*1/2+ 0*1/4+
0*1/2+ 2*1+ 0*1/2+ = 2
0*1/4+ 0*1/2+ 0*1/4

0	0	0	x	1/4	1/2	1/4	=
2	0	7		1/2	1	1/2	
0	0	0		1/4	1/2	1/4	

0*1/4+ 0*1/2+ 0*1/4+
2*1/2+ 0*1+ 7*1/2+ = 4.5
0*1/4+ 0*1/2+ 0*1/4

0	0	0	x	1/4	1/2	1/4	=
0	7	0		1/2	1	1/2	
0	0	0		1/4	1/2	1/4	

0*1/4+ 0*1/2+ 0*1/4+
0*1/2+ 7*1+ 0*1/2+ = 7
0*1/4+ 0*1/2+ 0*1/4

3	4	5
2.5	4.25	6
2	4.5	7

Note: These methods will only allows us to enlarge an image by a factor of (2N-1), but **what if we want to enlarge an image by something other than a factor of (2N-1)?**

To do this we need to apply a more general method. We take two adjacent values and linearly interpolate more than one value between them. This is done by define an enlargement number k and then following this process:

K-Times zooming: K times is a zooming method we are going to discuss. It is one of the most perfect zooming algorithm discussed so far. It caters the challenges of both nearest neighbors and pixel replication. K in this zooming algorithm stands for zooming factor.

It works like this way.

1. First of all , you have to take two adjacent pixels as you did in the pixel repetition. Then you have to subtract the smaller from the greater one.
2. Divide the output with the zooming factor(K). We call this output (OP)
3. Now you have to add the result (Op)to the smaller value and put the result in between those two values.
4. Add the value OP again to the value you just put and place it again next to the previous putted value. You have to do it till you place k-1 values in it.
5. sort the inserted values in ascending order , so there remains a symmetry between them
6. Repeat the same step for all the rows and the columns , and you get a zoomed images.

For example: Suppose you have an image of 2 rows and 3 columns , which is given below. And you have to zoom it three times. K in this case is 3. K = 3.
The number of values that should be inserted is k-1 = 3-1 = 2.

15	30	15
30	15	30

Row wise zooming

- Take the first two adjacent pixels. Which are 15 and 30.
- Subtract 15 from 30. 30-15 = 15.

- Divide 15 by k. 15/k = 15/3 = 5. We call it OP.(where op is just a name)
- Add OP to lower number. 15 + OP = 15 + 5 = 20.
- Add OP to 20 again. 20 + OP = 20 + 5 = 25.

We do that 2 times because we have to insert k-1 values. Now repeat this step for the next two adjacent pixels. It is shown in the first table. After inserting the values , you have to sort the inserted values in ascending order, so there remains a symmetry between them. It is shown in the second table

15	20	25	30	20	25	15
30	20	25	15	20	25	30

15	20	25	30	25	20	15
30	25	20	15	20	25	30

Column wise zooming

The same procedure has to be performed column wise. The procedure include taking the two adjacent pixel values, and then subtracting the smaller from the bigger one. Then after that , you have to divide it by k. Store the result as OP. Add OP to smaller one, and then again add OP to the value that comes in first addition of OP. Insert the new values. Here what you got after all that

15	20	25	30	25	20	15
20	21	21	25	21	21	20
25	22	22	20	22	22	25
30	25	20	15	20	25	30

The best way to calculate the formula for the dimensions of a new image is to compare the dimensions of the original image and the final image. The dimensions of the original image were 2 X 3. And the dimensions of the new image are 4 x 7. The formula thus is:

(K *(number of rows - 1) + 1) X (K *(number of cols -1) + 1)

Advantages and disadvantages

The one of the clear advantage that k time zooming algorithm has that it is able to compute zoom of any factor which was the power of pixel replication algorithm also it gives improved result (less blurry) which was the power of zero order hold method. So hence It comprises the power of the two algorithms. The only difficulty this algorithm has that it has to be sort in the end , which is an additional step , and thus increases the cost of computation.

Digital Image Processing: MCQs- Lecture 3

1. What is the first step in the convolution process for enlarging an image?

2. What is the second step in the convolution process?

3. What is the main arithmetic operation in convolution?

4. The convolution process involves finding which of the following?

5. What happens to the existing image values during convolution?

6. Where is the result of convolution stored during processing?

7. What is the purpose of using an image buffer in convolution?

8. What is the position (r,c) of the mask center in the first convolution operation?

9. What happens when the end of a row is reached in convolution?

10. What part of the image is lost after convolution?

11. Why is convolution often preferred despite more calculations?

12. Which order hold can also be achieved using convolution?

13. Why is the result of zero-order hold convolution placed at the lower-right corner?

14. The convolution-based methods allow enlargement only by what factor?

15. What method is used to enlarge an image by a factor other than (2N–1)?

16. What does "K" represent in the K-times zooming method?

17. In K-times zooming, what is done first?

18. After subtracting the smaller from the larger pixel, what is the next step?

19. What is the symbol "OP" used for in K-times zooming?

20. How many values are inserted between two original pixels in K-times zooming?

21. Why are inserted values sorted in ascending order in K-times zooming?

22. The K-times zooming process is applied to:

23. What formula gives the dimensions of the new image after K-times zooming?

24. One advantage of K-times zooming is that it:

25. What is the main disadvantage of K-times zooming?