DIGITAL IMAGE PROCESSING-1

Unit 3: Lecture 31-32

Dr. Shikha Tripathi
Department of Electronics &
Communication Engineering

DIGITAL IMAGE PROCESSING-1

Unit 3: Image Enhancement

Dr. Shikha Tripathi
Department of Electronics & Communication Engineering

Last Session

- Point Operations cont..
  - Histogram Processing
    - Histogram stretching
    - Histogram equalization

Today's Session

- Spatial / Neighborhood Operations
  - Convolution
  - Correlation

Types of Enhancement Techniques

Image Enhancement

Point Operations

- Contrast Stretching
- Noise Clipping
- Window Slicing
- Histogram Modeling

Spatial Operations

- Noise Smoothing
- Median Filtering
- Unsharp Masking
- LP,HP,BP filtering
- Zooming

Transform Operations

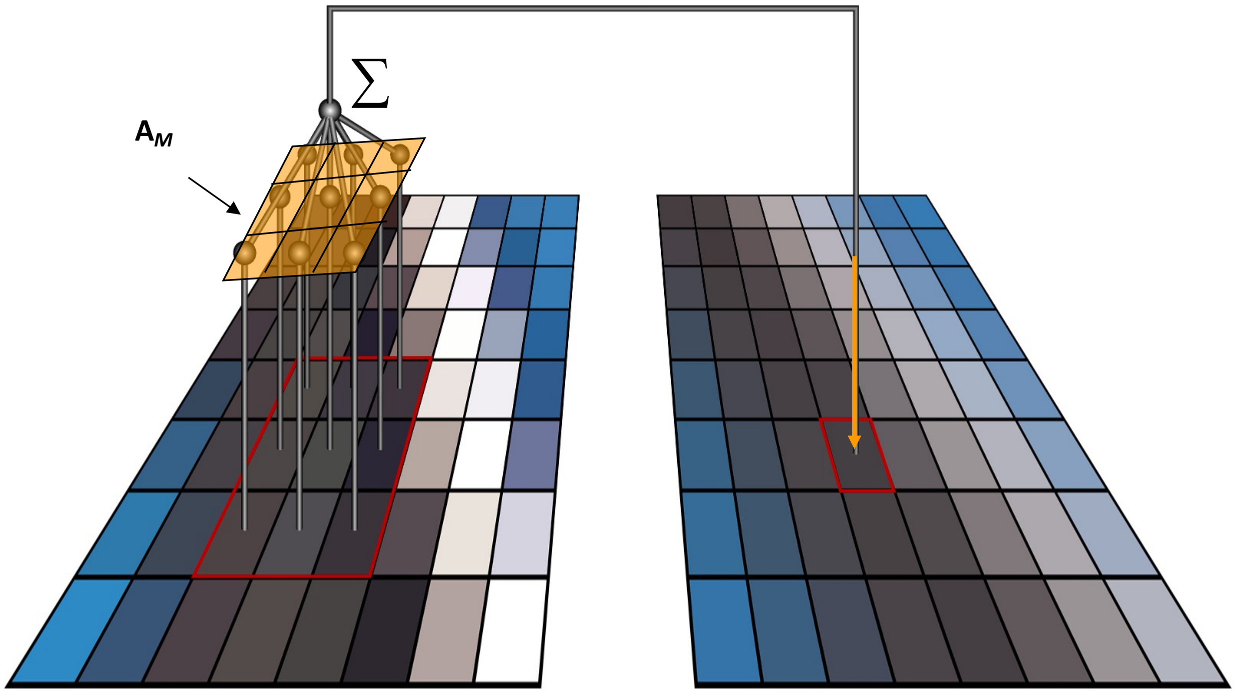
- Linear filtering
- Root Filtering
- Homomorphic Filtering

Pseudocoloring

- False Coloring
- Pseudocoloring

Spatial Domain

Spatial Operations (Neighbourhood Processing)



Local or Neighborhood Operations

- The pixels are modified based on some functions of the pixels in their neighborhood
- Work with the values of image pixels in the neighborhood and the corresponding values of a subimage that has the same dimension as the neighborhood
  - The subimage is called filter, mask, kernel, template or window
- Values in filter subimage (kernel) are referred to as coefficients rather than pixels

- **3 x 3 Neighborhood filter / Mask / Window / Kernel / Template:**

	(y - 1)	y	(y + 1)	Y
(x - 1)	W1 g(x-1, y-1)	W2 g(x-1, y)	W3 g(x-1, y+1)	
x	W4 g(x, y-1)	W5 g(x, y)	W6 g(x, y+1)	
(x + 1)	W7 g(x+1, y-1)	W8 g(x+1, y)	W9 g(x+1, y+1)	
X				

Spatial Filtering

- Filtering operations performed directly on pixels (not frequency domain filtering)
- Process consists of moving the filter mask from point to point in an image
- At each point (x,y), the response of the filter at that point is calculated using a predefined relationship in the neighborhood
  - Ex. For a linear spatial filter, response is sum of products of filter coefficients and corresponding image pixels in the area spanned by filter mask

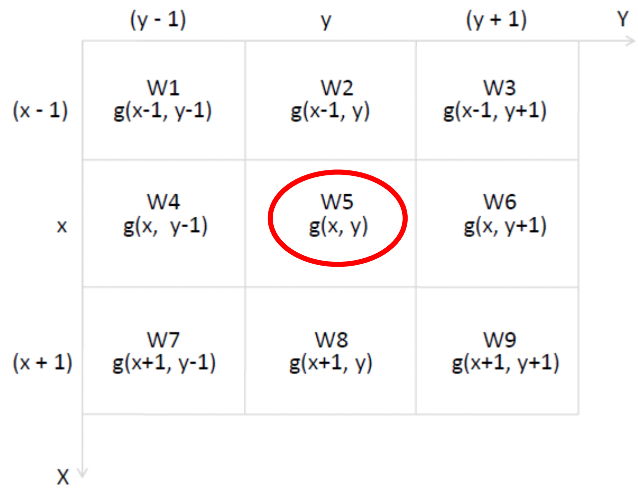
Steps Involved

- To achieve neighborhood processing:

1. Place the mask on the image.
2. Multiply each mask component with the pixel component.
3. Add them and place value at the center. Similar to *CONVOLUTION.*
    - here we need not flip the mask as it is symmetric.

- If g is original image & f is modified image, then:

$$f(x, y) = g(x-1,y-1).w1 + g(x-1,y).w2 + g(x-1,y+1).w3 + g(x,y-1).w4 + g(x, y).w5 + g(x,y+1).w6 + g(x+1,y-1).w7 + g(x+1,y).w8 + g(x+1,y+1).w9$$

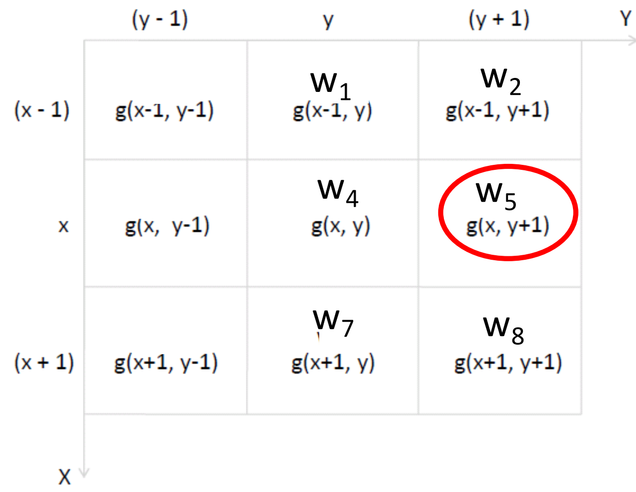


Steps Involved Cont…

5. Once f(x, y) is calculated, shift mask by 1 step to right.

- **Now, W_5 coincides with g(x, y+1)**

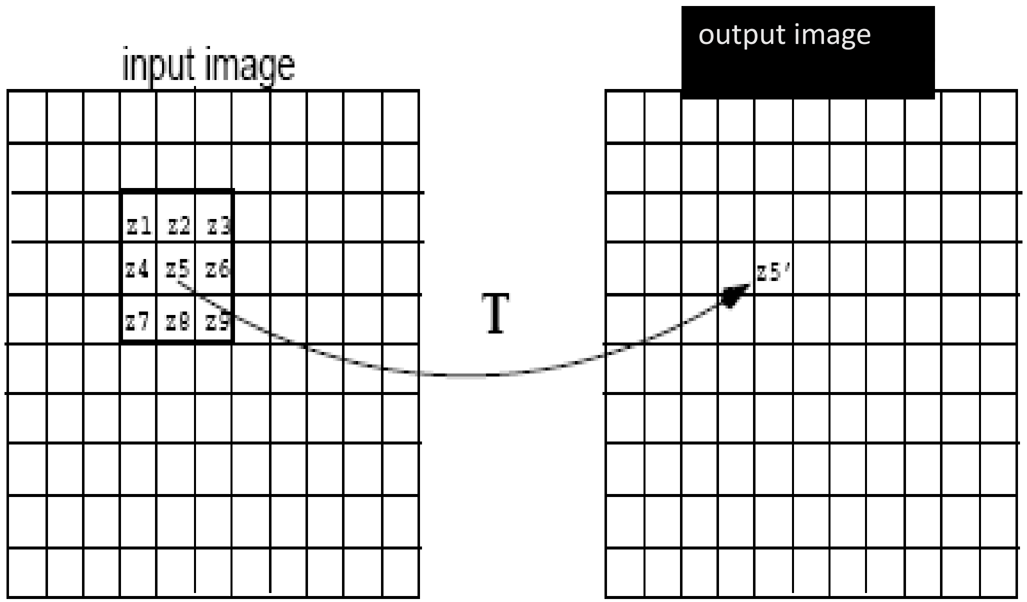
6. Repeat steps 2 to 5 till all pixels in original image are traversed



- For a mask of size m x n, we assume m = 2a+1 and n = 2b+1 (a and b are positive integers)
  - We consider masks of odd sizes with smallest size being 3 x 3 where a=b=1.

Spatial Filters

Area or Mask Processing Methods



w1	w2	w3
w4	w5	w6
w7	w8	w9

$$g(x,y) = T[f(x,y)]$$

T operates on a neighborhood of pixels

$$z5' = R = w1z1 + w2z2 + \dots + z9w9$$

A filtered image is generated as the center of the mask moves to every pixel in the input image.

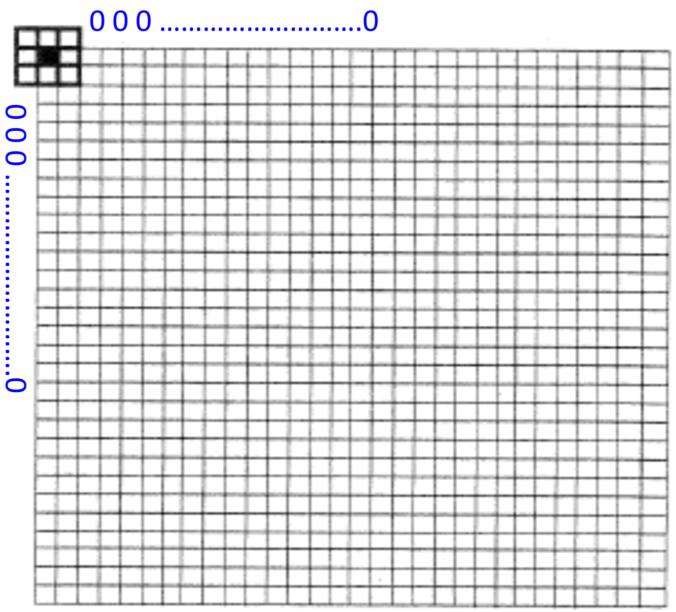
Applications of Neighbourhood Processing

- Image Filtering:

  Ex. LPF, HPF, BPF, BRF

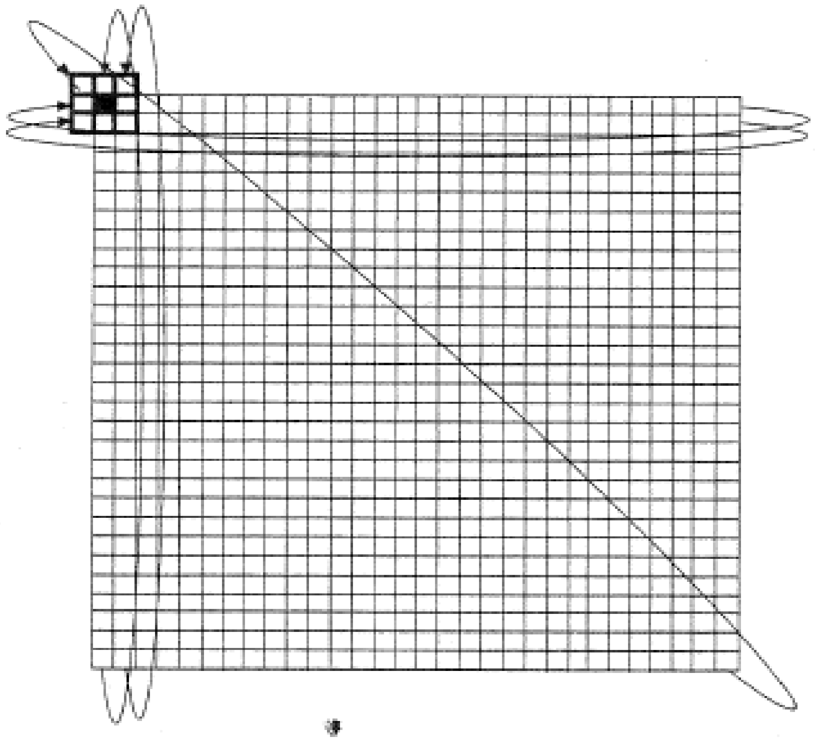
  - In 1D signals, like speech, audio, EEG,ECG etc. how fast the signal changes is indication of frequency
  - Same concept is applied to images where we have gray levels instead
    - **If gray level changes slowly over a region then LF area (Ex. Background)**
    - **If gray level changes abruptly over a region then HF area (Ex. Edges, Boundaries)**

Handling pixels close to boundaries



pad with zeroes

or



Linear Filters

- Each pixel is replaced by linear combination of intensities of neighboring pixels
- Each pixel value in output image is weighted sum of pixels in the neighborhood of the corresponding pixel in the input image

w1	w2	w3
w4	w5	w6
w7	w8	w9

$$z5' = R = w1z1 + w2z2 + \ldots + z9w9$$

- Can be used to smoothen or sharpen the image
- A spatially invariant linear filter can be implemented using a convolution mask
- Spatially varying filter

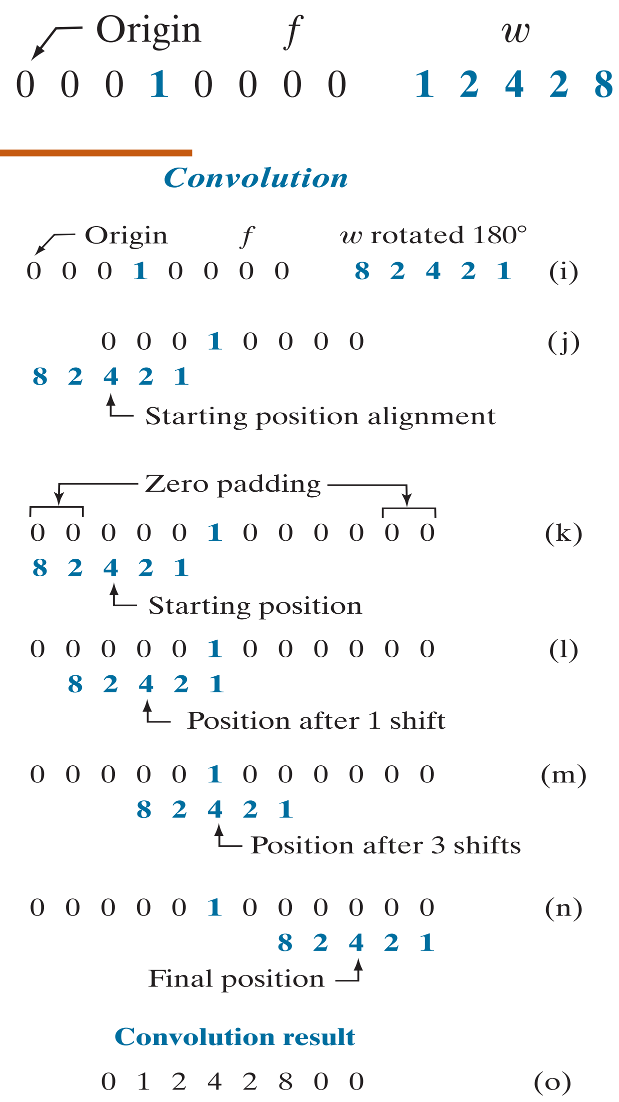
Image Convolution

- Convolution and correlation are used to extract information from images
- They are linear and shift invariant operations
  - Linear: Pixel is replaced by linear combination of its neighbors
  - Shift Invariant:

Image Convolution

- **Convolution** is a mathematical operation where each value in the output is expressed as the sum of values in the input multiplied by a set of weighting coefficients
  - Depending on the weighting coefficients, convolution operation is used to perform spatial domain lowpass and highpass filtering of the image.
  - **An image can be smoothened or sharpened by convolving the image with respect to lowpass and highpass mask respectively**
  - Convolution has many applications like: image filtering, image enhancement, image restoration, feature extraction and template matching

1D Convolution



- 1 D Convolution

$$g(x) = \sum_{s=-a}^{a} w(s)f(x-s)$$

- The only difference here is that the kernel is *pre-rotated* by 180° prior to performing the shifting/sum of products operations.
- As the convolution in Figure shows, the result of pre-rotating the kernel is that now we have an *exact* copy of the kernel at the location of the unit impulse.
- In fact, a foundation of linear system theory is that convolving a function with an impulse yields a copy of the function at the location of the impulse.

1D Correlation

- 1 D Correlation is given by $g(x) = \sum_{s=-a}^{a} w(s) f(x+s)$
- Correlation consists of moving the center of a kernel over an image, and computing the sum of products (MAC) at each location.
- The mechanics of *spatial convolution* are the same, except that the convolution kernel is rotated by 180° (folded)
- **Thus, when the values of a kernel are symmetric about its center, correlation and convolution yield the same result.**
- w.k.t for a kernel of size $m \times n$, $m = 2a + 1$ and $n = 2b + 1$
- Consider f(x,y)= [0 0 0 1 0 0 0 0] and w = [1 2 4 2 8]

center

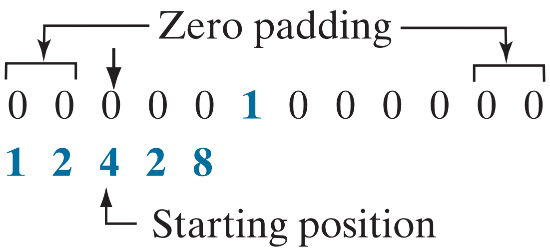
- The kernel w is of size 1 × 5, so $a = 0$ and $b = 2$

```
    0 0 0 1 0 0 0 0
1 2 4 2 8
    Starting position alignment
```

We notice that part of w lies outside *f*, so the summation is undefined in that area.

Correlation

- A solution to this problem is to *pad* function *f* with enough 0’s on either side.
- In general, if the kernel is of size 1 × *m*, we need (*m* − 1)/ 2 zeros on either side of *f* in order to handle the beginning and ending configurations of w with respect to *f*



$$\boxed{g(x) = \sum_{s=-a}^{a} w(s) f(x+s)}$$

- In this starting configuration, all coefficients of the kernel overlap valid values.

Correlation

- The first correlation value is the sum of products in this initial position, computed using correlation equation with x = 0 giving g(0)

$$\boxed{g(x) = \sum_{s=-a}^{a} w(s) f(x+s)}$$

Zero padding

0 0 0 0 0 **1** 0 0 0 0 0 0

1 2 4 2 8

Starting position

g(0) = 0

- To obtain the second value of correlation, we shift the relative positions of w and f one pixel location to the right [i.e., we let x = 1] and compute the sum of products again.

0 0 0 0 0 **1** 0 0 0 0 0 0

1 2 4 2 8

Position after 1 shift

g(1) = 8

Correlation

- Similarly, we obtain other output values shifting the window to right by 1 and computing MAC operations

$$g(x) = \sum_{s=-a}^{a} w(s) f(x+s)$$

0 0 0 0 0 **1** 0 0 0 0 0 0
1 2 4 2 8
Position after 3 shifts

g(3) = 4

0 0 0 0 0 **1** 0 0 0 0 0 0
1 2 4 2 8
Final position

g(7) = 0

Correlation result

0 8 2 4 2 1 0 0

Note that it took 8 values of x (i.e., $x = 0, 1, 2, \ldots, 7$) to fully shift w past f so the *center* coefficient in w visited *every* pixel in f.

- Sometimes, it is useful to have every element of w visit every pixel in f.
- For this, we have to start with the rightmost element of w coincident with the origin of f, and end with the leftmost element of w being coincident the last element of f

Extended (full) correlation result

0 0 0 8 2 4 2 1 0 0 0 0

Correlation

$$g(x) = \sum_{s=-a}^{a} w(s) f(x+s)$$

- Two important points to note from this
  - **Correlation is a function of *displacement* of the filter kernel relative to the image**
    - In other words, the first value of correlation corresponds to zero displacement of the kernel, the second corresponds to one unit displacement, and so on.
  - **Correlating a kernel w with a function that contains all 0's and a single 1 yields a *copy* of w, but *rotated* by 180°**
    - A function that contains a single 1 with the rest being 0's is called a *discrete unit impulse*.
    - **Correlating a kernel with a discrete unit impulse yields a *rotated* version of the kernel at the location of the impulse.**

w

1 2 4 2 8

Origin f

0 0 0 **1** 0 0 0 0

Correlation result

0 8 2 4 2 1 0 0

- 2D Convolution

The *convolution* of a kernel w of size $m \times n$ with an image $f(x, y)$, denoted by $(w \star f)(x, y)$,

$$(w \star f)(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

Basic operations: Folding, Shifting, Multiplying, Adding

- 2D Correlation

The correlation of a kernel w of size $m \ x \ n$ with an image $f(x, y)$, denoted as $(w \star f)(x, y)$

$$(w \star f)(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

2D Convolution

- Define correlation and convolution so that *every* element of w (instead of just its center) visits *every* pixel in f.
- This requires that the starting configuration be such that the right, lower corner of the kernel coincides with the origin of the image.
- Similarly, the ending configuration will be with the top left corner of the kernel coinciding with the lower right corner of the image.
- If the kernel and image are of sizes $m \times n$ and $M \times N$, respectively, the padding would have to increase to $(m - 1)/2$ padding elements above and below the image, and $(n - 1)/2$ elements to the left and right.
- Under these conditions, the size of the resulting full correlation or convolution array will be of size $S_v \times S_h$, where. $S_v = m + M - 1$ and $S_h = n + N - 1$

- 2D Correlation

$$(w \star f)(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

- 2D Convolution

$$(w \bigstar f)(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

2D Convolution

- Basic operations: Folding, Shifting, Multiplying, Adding

Example 1: Perform linear convolution between the two matrices

$$x(m,n) = \begin{bmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad h(m,n) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

2D Convolution

- Convolved matrix is

$$y(m,n)=\begin{bmatrix} 4 & 5 & 6 \\ 11 & 13 & 15 \\ 11 & 13 & 15 \\ 7 & 8 & 9 \end{bmatrix}$$

- Dimension of resultant matrix = {No. of rows of x(m,n) + No. of rows of h(m,n) -1}x {No. of columns of x(m,n) + No. of columns of h(m,n) -1}
- Dimension of the given convolution =(2+3-1)x(3+1-1) = 4 x 3

2D Convolution

Example2: Perform linear convolution between the two matrices

$$x = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} ; h = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$f(x,y) = \begin{bmatrix} 5 & 16 & 12 \\ 22 & 60 & 40 \\ 21 & 52 & 32 \end{bmatrix}$$

Dimension of the given convolution

=(2+2-1)x(2+2-1)=3 x 3

Correlation

Example 3: Perform correlation between the two matrices

$$x = \begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix} ; h = \begin{bmatrix} 1 & 5 \\ 2 & 3 \end{bmatrix}$$

$$\gamma_{xh} = \begin{bmatrix} 9 & 9 & 2 \\ 21 & 24 & 9 \\ 10 & 22 & 4 \end{bmatrix}$$

Properties of Correlation and Convolution

- 2D Correlation

$$(w \star f)(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

- 2D Convolution

$$(w \bigstar f)(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

Property	**Convolution**	**Correlation**
Commutative	$f \bigstar g = g \bigstar f$	—
Associative	$f \bigstar (g \bigstar h) = (f \bigstar g) \bigstar h$	—
Distributive	$f \bigstar (g + h) = (f \bigstar g) + (f \bigstar h)$	$f \star (g + h) = (f \star g) + (f \star h)$

Linear Filtering

- In general linear filtering of an image of size M x N with a filter mask of size m x n is given by (Since kernels are generally symmetric correlation and convolution give same result) :

$$g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t)$$

where x =0,1,2, ……, M-1 and y = 0,1,2,….,N-1

a = (m-1)/2 and b = (n-1)/2

- Linear filtering is also called as Convolution
  - Convolving a mask with an image

Next Session

- Spatial Filters
- Image Sharpening Filters
- Sharpening using Laplacian operator

THANK YOU

Dr. Shikha Tripathi
Department of Electronics &
Communication Engineering

shikha@pes.edu
+91 9482219115