So in the final position, the output produced is

~~{0 0 8 2 3 0 0}~~. {0 0 3 2 8 0 0}

[# note:-
Correlation follows the same steps as convolution, the only difference is that the mask or the kernel is not rotated].

Q) Let $I = \begin{pmatrix} 3 & 3 \\ 3 & 3 \end{pmatrix}$ be an image and $K = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ be a kernel (mask). Perform convolution and correlation.

1) <u>Convolution</u>.

<u>Step 1</u>:-
rotate kernel by 180° (reverse the rows then reverse the columns)

$\therefore K' = \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix}$

$\begin{matrix} 3 & 4 \\ 1 & 2 \end{matrix} \Rightarrow \begin{matrix} 4 & 3 \\ 2 & 1 \end{matrix}$

<u>Step 2</u>:-
pad with zeros on all the side.

$$\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{matrix} \quad \text{and } K' = \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix}$$

<u>Step 3</u>:-
mask the kernel on the padded image from the top left corner. and generate the new image by generating the output and writing it on the ~~first padded zero~~ upper right corner.

(a) mask the kernel

$$\begin{array}{c|cccc} & 4 & 3 & & \\ \hline & \boxed{\begin{matrix} 0 & 0 \\ 0 & 3 \end{matrix}} & & 0 & 0 \\ 2 \; 1 & & & 3 & 0 \\ & 0 & 3 & 3 & 0 \\ & 0 & 0 & 0 & 0 \end{array}$$

$\therefore (4\times0)+(3\times0)+(2\times0)+(3\times1)$

$= 3$

Thus the new image is

(b)

$$\begin{matrix} 3 & \boxed{0} & \boxed{0} & 0 \\ 0 & \boxed{3} & \boxed{3} & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$$

now repeat the same steps by shifting the kernel and masking the same way.

The output is $(3\times2)+(3\times1)=9$

(c)

$$\begin{matrix} 3 & 9 & \boxed{0} & \boxed{0} \\ 0 & 3 & \boxed{3} & \boxed{0} \\ 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$$

The output $=(3\times2)=6$

(new image)

(d)

$$\begin{matrix} 3 & 9 & 6 & 0 \\ \boxed{0} & \boxed{3} & 3 & 0 \\ \boxed{0} & \boxed{3} & 3 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$$

The output
$\Rightarrow (3\times3)+(3\times1)$
$\Rightarrow 9+3 \Rightarrow 12$

$\Rightarrow$

$$\begin{matrix} 3 & 9 & 6 & 0 \\ 12 & \boxed{3} & \boxed{3} & 0 \\ 0 & \boxed{3} & \boxed{3} & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$$

(new image)

The output:-
$(4\times3)+(3\times3)+(3\times2)+(3\times1) = 30$

$\Downarrow$ (new image)

$$\begin{matrix} 3 & 9 & 6 & 0 \\ 12 & 30 & \boxed{3} & \boxed{0} \\ 0 & 3 & \boxed{3} & \boxed{0} \\ 0 & 0 & 0 & 0 \end{matrix}$$

The output $\Rightarrow 12+6 \Rightarrow 18$

$\Leftarrow$

(new image)

$$\begin{matrix} 3 & 9 & 6 & 0 \\ 12 & 30 & 18 & 0 \\ \boxed{0} & \boxed{3} & 3 & 0 \\ \boxed{0} & \boxed{0} & 0 & 0 \end{matrix}$$

The output $\Rightarrow 9$

$\Leftarrow$

(new image)

$$\begin{matrix} 3 & 9 & 6 & 0 \\ 12 & 30 & 18 & 0 \\ 9 & \boxed{3} & \boxed{3} & 0 \\ 0 & \boxed{0} & \boxed{0} & 0 \end{matrix}$$

The output $\Rightarrow 12+9$
$\Rightarrow 21$

$\Downarrow$

(new image)

$$\begin{matrix} 3 & 9 & 6 & 0 \\ 12 & 30 & 18 & 0 \\ 9 & 21 & \boxed{3} & \boxed{0} \\ 0 & 0 & \boxed{0} & \boxed{0} \end{matrix}$$

output $\Rightarrow 4\times3=12$

$\Rightarrow$

$$\begin{matrix} 3 & 9 & 6 & 0 \\ 12 & 30 & 18 & 0 \\ 9 & 21 & 12 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$$

This is the final image.

Smoothing Spatial filters (ch 5) Noise & filter

→ Smoothing filters are used for blurring & noise reduction

→ Blurring is used in preprocessing tasks, such as removal of small details from an image prior to (large) object extraction

→ Noise reduction can be acomplished by blurring with a linear ~~and~~ filter and also by non linear filtering.

Smoothing Spatial filter

- linear filter
  - Mean/Box filter
  - Weighted avg. filter
  - Gaussian filter
- non-linear filters. (order - static filters)
  - Median filter
  - Max filter
  - Min filter

Smoothing Linear filter

They are also known as averaging filter (or) lowpass ~~box~~ filters as they are simply the average of the pixels contained in the neighbourhood of the filter mask.

The process result in an image with reduced 'sharp' transitions in intensities which ultimately leads to noise reduction.

Adaptive filters

* Linear filter

1) Box/mean filter - all coefficients are equal.

(5.3)

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \rightarrow \text{Mask}$$

3x3

Arithmetic mean
geometric mean
Harmonic mean.
Contraharmonic mean

2) Weighted avg. - give more (less) weight to pixels near (away from) the output location.

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \rightarrow \text{Mask.}$$

3) Gaussian filter - the weights are samples of 2D Gaussian function:-

$$\boxed{G_0(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}}$$ (2D Gaussian function)

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \rightarrow \text{Mask.}$$

→ used to blur edges and reduce contrast

→ Similar to median filter but is faster.

* Non-linear (Order - Statistic)

Their response is based on ordering (ranking) the pixels ~~the pixels~~ contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.

1) Median filters - Find the median of all the pixel values.

2) Min filter - Find the minimum of all the pixel values.

3) Max filter - Find the max of all the pixel values

Q Consider the image below and calculate the output of the pixel (2,2) if smoothing is done using 3×3 neighbourhood using all the filter below:-

(a) Box/Mean filter
(b) Weighted average filter
(c) Median filter.
(d) Min filter
(e) Max filter

1	8	8	0	7
4	7	9	5	7
5	4	6	8	6
4	2	0	1	5
0	1	0	2	0

3X3

Solution:-

(a) Box/Mean filter.

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\frac{1}{9}\left[7+9+5+4\times6+8\times2+0+1\right]$$

$$\Rightarrow \frac{1}{9}[42] = 4.66 \simeq 5.$$

(b) Weighted avg. filter.

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\Rightarrow \frac{1}{16}\left[7+18+5+8+24+16+2+0+1\right]$$

$$\Rightarrow \frac{1}{16}(81) = 5.0625 \simeq 5.$$

(c) Median filter :- [more efficient.

arrange in ascending order
Pixels

0 1 2 4 5 6 7 8 9

(d) Min filter.

0

(e) Max filter (used for finding the brightest noise pixel).

9.

Theories

1) Why median filter is better than mean filters?

Ans) Median filter is normally used to reduce noise in an image, similar to the mean filter. However, it often does a better job than mean filters in preserving useful details in an image.

Median filters has 2 main advantages :-

- The median is more robust avg. than the mean and so a single very unrepresentative pixels in a neighborhood will not affect the median value significantly

- Since the median value must actually be the value of one of the pixels in the neighborhood the median filter does not create new unrealistic pixel values when the filter straddles an edge. Therefore it is much better at preserving sharp edges than the mean filter.

That is because when we calc. median. It always gives output a pixel value of the images whereas mean filters generates a value that may or may not exist in the image which is unrealistic

NOTE

— Mean filter is better at dealing with Gaussian Noise than median filter.

- Median filter is better at dealing with salt and pepper noise than mean filter.

2.) Write down a few approaches to deal with missing edge pixels.

Ans) A few approaches to dealing with missing edge pixels are :-

1) Omit missing pixels.
   - only works with some filters.
   - Can add extra code and slow down processing
2) Pad the image.
   - Typically with either all white or all black pixels
3) Replicate border pixels.
4) Truncate the image.
5) Allow pixels to wrap around the image
   - Can cause some strang image artifacts.

* Zero Padding

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

* Pixel Replication.

1	1	2	3	3
1	1	2	3	3
4	4	5	6	6
7	7	8	9	9
7	7	8	9	9

* Wrapping of Pixels

9	7	8	9	7
3	1	2	3	1
6	4	5	6	4
9	7	8	9	7
3	1	2	3	1

Sharpening Spatial Filter

— The principal objective of shapening is to highlight transitions in intensity.

— Application of image sharpening include electronic printing, medical imaging, industrial inspection and autonomous guidance in military systems.

Bluring → pixel averaging

Sharpening → Spatial differentiation.

→ The strength of the responce of the derivation operation is proportional to the degree of intensity discontinuity of the image at the part of which operator is applied.

Foundation of sharpening filters.

1) First-order derivative of an one-dimentional function $f(x)$:

$$\frac{dy}{dx} = f(x+1) - f(x)$$

2) Second-order derivative of a one-dimentional function $f(x)$:

$$\frac{d^2y}{dx^2} = f(x+1) + f(x-1) - 2f(x).$$

Laplacian filter :-

→ It highlights gray-level discontinuities in an image.

→ It deemphasize regions with slowly varying gray levels

→ Formula : $\nabla^2 f = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2}$

where;

$$\frac{d^2 f}{dx^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{d^2 f}{dy^2} = f(x, y+1) + f(x, y-1) - 2f(x, y).$$

Thus.

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Laplacian mask

0	1	0	1	1	1		$f(x-1, y-1)$	$f(x, y-1)$	$f(x+1, y-1)$
1	-4	1	1	-8	1		$f(x-1, y)$	$f(x, y)$	$f(x+1, y)$
0	1	0	1	1	1		$f(x-1, y+1)$	$f(x, y+1)$	$f(x+1, y+1)$

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

Q) Apply Laplacian filter on the given image on the center pixel.

$$\begin{matrix} 8 & 5 & 4 \\ 0 & 6 & 2 \\ 1 & 3 & 7 \end{matrix} \ast \begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}$$

Input image

⇒ $5 + 2 + 3 - 24$

⇒ $\underline{-14}$

output to replace the center pixel

Enhanced laplacian Filter

Laplacian filter

$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix} \xrightarrow{\text{Enhanced}} \begin{matrix} 0 & 1 & 0 \\ 1 & -5 & 1 \\ 0 & 1 & 0 \end{matrix}$$

$\frac{1}{100} \times 1000$

Enhanced laplacian filter.

$$\begin{matrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{matrix} \xrightarrow{\text{Enhanced}} \begin{matrix} 1 & 1 & 1 \\ 1 & -9 & 1 \\ 1 & 1 & 1 \end{matrix}$$

2) Apply enhanced laplacian filter on the given image on the center pixel.

input img. + Laplacian filter

$$\begin{bmatrix} 8 & 5 & 4 \\ 0 & 6 & 2 \\ 1 & 3 & 7 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & -9 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$\Rightarrow 8 + 5 + 4 + 2 + 7 + 3 + 1 - 54$

$\Rightarrow 30 - 54 = -24$

replace center pixel with output to get output image.

Unsharp masking and Highboost Filtering.

→ Primarily used in the printing & publishing industry to sharpen images.

→ The process involves subtracting an unsharp (smoothed) version of an image from the original image. This process is called unsharp masking consists of the following steps:-

1) Blur the original image.
2) Subtract the blurred image from the original (the resulting difference is called the mask).
3) Add the mask to the original.

Sharpend = Original + (Original - blurred) x amount.

Unsharp masking (mathematically).

$$f_s(x,y) = f(x,y) - \bar{f}(x,y) \qquad \bar{f}(x,y) = f(x,y) - f_s(x,y)$$

sharpened image = original image - blurred image.

Subtracting a blurred ~~image~~ version of an image from the original produces a sharpened image.

Highboost filtering. (that is adding an amount A as scaling factor.)

$\therefore f_{hb} = A f_s$

$$f_{hb}(x,y) = Af(x,y) - \bar{f}(x,y)$$

$$= Af(x,y) - [f(x,y) - f_s(x,y)]$$

$$\boxed{f_{hb} = (A-1)\, f(x,y) + f_s(x,y)}$$

This is the generalized equn. of unsharp masking. where $A \geq 1$, that is this equation is used for both highboost and unsharp masking.

If $A = 1$, then it is the equn. for unsharp masking and if $A > 1$ then it is the equn. for highboost filterning.

Thus, A specifies the amount of sharpening of the image. (but only to a certain point).

Now if we use Laplacian filter to create the sharpened image (f_s) with addition of the original image, we get,

$$f_s(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) \\ f(x,y) + \nabla^2 f(x,y) \end{cases}$$

$$f_{hb}(x,y) = \begin{cases} Af(x,y) - \nabla^2 f(x,y) \\ Af(x,y) + \nabla^2 f(x,y) \end{cases}$$

Highboost Mask :-

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & A+4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ -1 & A+8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Q1.) Apply highboost filter on the image given below on the center pixel. Use the mask with A = 1.7.

Soln:-

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} * \begin{bmatrix} -1 & -1 & -1 \\ -1 & A+8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Input image

$$\Rightarrow -1(1+2+3+4+6+7+8+9) + 5(1.7+8)$$

$$\Rightarrow -1(40) + 5(9.7)$$

$$\Rightarrow 8.5$$

∴ Thus the output image after masking will be

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 8.5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

output image

First Order Derivative Filters

The first order derivative filters are used for Edge detection

Roberts Operators (Cross gradient)

→ One of the first edge detectors used for diagonal edge detection.

→ Works using 2×2 mask.

→ It is also known as Cross gradient and it works on the concept of Cross diagonal differences. (this it only focuses on the diagonal elements and not on the neighbouring pixels).

21	22	23
24	25	26
27	28	29

Image.

2×2 masks.

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Problems with Roberts Cross.

1) 2×2 mask are not easy to implement. (as it is an even sized kernal which makes it difficult to convolute on image's as it doesn't contain any center pixel or doesn't have symetry which make it difficult to preserve the spatial resolution of the image)

2) No. of calculations are more (due to the small size of the kernel)

3) No. of neighbouring pixels considered in one go are less

4) To solve these problems, we made the following changes:-

→ change in size of the mask.

→ change in the no. of neighbouring pixels considered.

So, the other two Operator were introduced,

a) Sobel Operators :-

- used for vertical and horizontal edge detection.
- More Importance is given to the pixels near the center.

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Masks for the sobel operators.

(b) Prewitt Operators :-

- Just similar to Sobel operator.
- Mainly used for vertical or horizontal edges of images.

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Q) Apply Roberts, Sobel and Prewitt operators in the pixel (1, 1) in the following image.

Solun:-

50	50	100	100
50	50	100	100
50	50	100	100
50	50	100	100

Input image.

pixel Replication

Roberts operator

origin

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

mask

Masking:-

50	50	100	100
50	50	100	100
50	50	100	100
50	50	100	100

$\Rightarrow 50 \times (-1) + 100 \times (1)$

$\Rightarrow 100 - 50 = \underline{\underline{50}}.$

Sobel Operator

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$\Rightarrow 50(-1) + 50(-2) + 100(-1) + 50(1) + 50(2) + 100(01)$

$\Rightarrow \underline{\underline{0}}$

Prewitt Operator

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

~~⇒ -50 -50 -50 +100 +100 +100~~

$\Rightarrow -50 - 50 - 100 + 50 + 50 + 100$

$\Rightarrow 0$