So in the final position, the output produced is \$ 100822003. 300328003

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

It Let
$$T = \begin{pmatrix} 3 & 3 \\ 3 & 3 \end{pmatrix}$$
 be an image and $K = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ be

a kernel (mask). Perform convolution and correlation.

Step 1:rotate kernel by 180° (renerse the rows then)
renerse the colourns)

: K = (4 3)
2 1)

a Step 2:-

pad with zeros on all the side.

00 00 and
$$K' = (43)$$
03 30 and $K' = (43)$

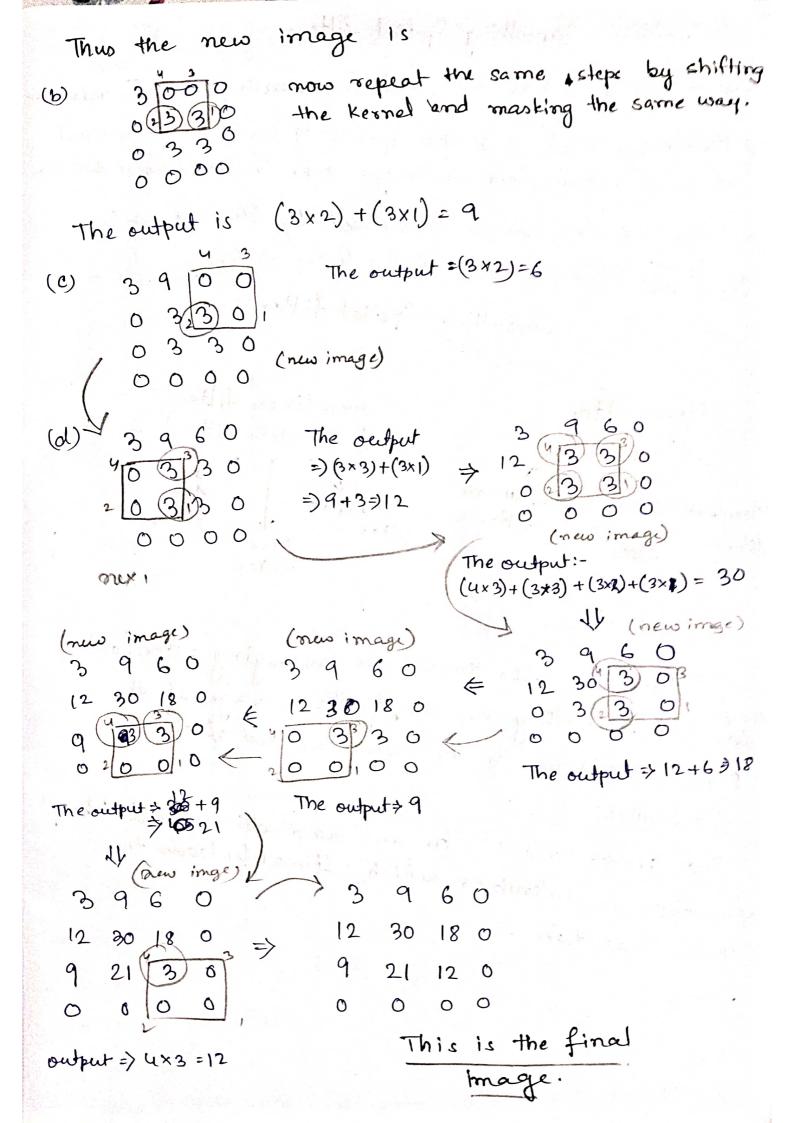
Step 30mask the kernel on the padded image from the top left corner, and general the new image by generaling the autput and writing it on the first padded torso.

(a)
$$(2\times0) + (2\times0) + (2\times0) + (3\times1)$$

mask+ne $(2\times0) + (3\times1)$

Kernel $(3\times0) + (2\times0) + (3\times1)$

A STATE OF THE STA

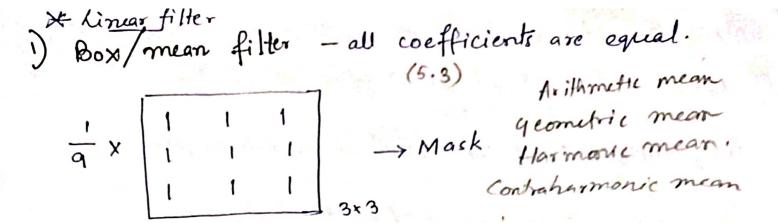


Smoothing Spatial filters (ch 5) Noise -> Smoothing filters are used for blurring A noise reduction -> Blurring is used in preprocessing tasks, such as removal of small details from an image prior to (large) object entaction -> Noise reduction can be accomplisted by blurring with a linear and filter and also by non linear filtering. Smoothing Spatial filter linear filter non-linear filters. Mean/Box Weighted gaussian avg. filter filter (order - Static filters) Median Min filter Max filter (* Smoothing linear filter

They are also known as averaging filter (or) cowpans down 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.

Adaptintilters



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

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

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

$$G_0(x,y) = \frac{1}{2\pi\sigma^2} \left(\frac{-(x^2+y^2)}{2\sigma^2}\right) \left(\frac{20}{4\pi}\right) G_0(x,y)$$

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

tured to blur edges and reduce contrast

y Similar to median filter but is faster.

* Non-linear (order-Statisfic)

Their response is based on ordering (nanking) the pixels the pixels contained in the image area encomposed by the filter, and then replacing the value of the center pixel with the value determined by the

ranking result.

- Median filters- Find the median of all the pixel values.
- 2) Min filter Find the minimum of all the pixel values.
- 3) Max filler Find the mox of all the pixel values

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
- @ Median filter.
- (d) Min filter
- (e) Max filter

Solur: (a) Box/Mean filter.

$$\frac{1}{9} \left[\frac{42}{9} \right] = 4.66 \approx 5.$$

(b) Weighted aug. filter.

$$\frac{1}{16}(81) = 5.0625 \approx 5$$

1	8	8	0	7
4	7	9	5	7
5	4	6	8	B
4	2	0	1	5 3×3
0	l	0	2	0

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

(c) Median filter :- [more efficient.

arrange in ascending order 0124658789 (d) Min filter. (e) Max filter (used for Andrythe)

9. bightes notre pixel).

Theories) why median filter is better than mean filters? And 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 fitters in preserving useful détails 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 neighbojarhood 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 gives output a pixel value # NOTE of the image; whereas mean filters generales a - Mean filter is better at dealing with is unrealistic 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:

i) Omit missing pixels.

- only works with some filters.

- Can add extra cocle and slow down processin

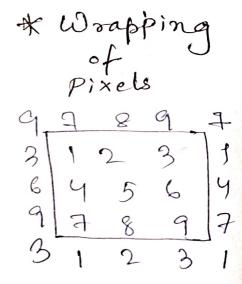
2) Pad the image.

- Typically with either all white or all black pixels

3) Replicate border pixels.

u) Truncate the image.

5) Allow pixels to wrap around the image -Can cause some strang image arctifacts.



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 authonomous guidance in military systems.

Bluring > pixel averaging Shorpening > Spatial differentiation.

The strength of the response of the derivation operation is discontinuity of the image at the part of which operator is applied.

Foundation of sharpening filters.

i) 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 highlighte gray-level discontinuities in an image.

of It deemphasize regions with slowly varying gray levels

$$\rightarrow$$
 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

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

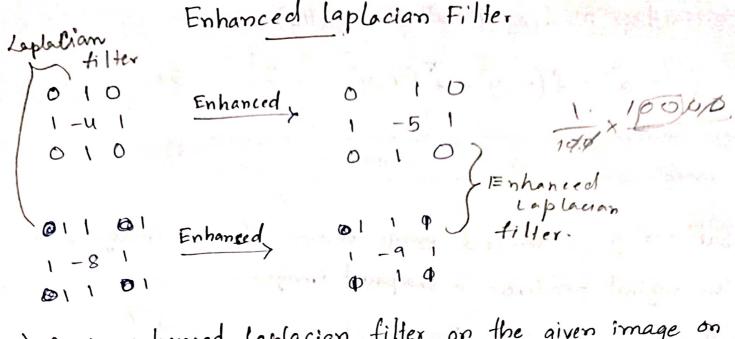
-4f(x,y)

Laplacian mask

٥

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

Apply laplacian filter on the given image on the center pixel.



2) Apply enhanced laplacian filter on the given image on

the center pixel. Laplacian filer

input img.
$$+$$
 $8 5 4$
 $0 6 2$

replace 1 3 7

center pixel with output to get output image.

Unsharp masking and Highboost Filtering.

- if 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 eunsharp marking consists of the following steps:-1) Blur the original image.
 - 2) subtract the blurred image from the original (the resulting difference is called the made).
 - 3) Add the mask to the original. Shorpend = Original + (original - blurred) x amount.

Unshap masking (mathemetically).

J(my)= J(m,y)-f(m, $f_s(x,y) = f(x,y) - f(x,y)$ Sharperdo , original _ blurred image image.

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

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

fnb (x,y) = Af (x,y) - f(x,y) = Af(x,y) - (f(x,y) - f(x,y))

fhb = A (A-1) f(x,y) +fs(x,y)

This is the generalized equin. of unsharp masking. where A>1,1, that is this equation is used for both highboost and unsharp masking.

If A=1, then it is the equin for unsharp masking and if A> 1 then it is the equin, for high boost filterning Thus, A specifies the amount of sharpening of the image. (but only to a certain point).

Now if we use laplacion filter to create the sharpened image (fc) with addition of the original image,

we get,

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

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

Highboost Mask :-

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

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

(91) Apply highboost filter on the image given below on the center pixel. Use the mark with A=1.7.

Solun 8-

: Thus the output image after masking will be

First Order Derivative Fiters.

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

21 22 23 24 25 26 27 28 29 2x2 masks. Image.

0 0 1

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 to a like the size of the siz on images as it doesn't contain any center pixel or doesn't have

symetry which make it difficult to present the spatial

2) No of calculations are more (due to the small size of) resolution of the image)

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

4) To solve these problems, we made the following Charages 3-

+ change in cize of the mark.

-> change in the no. of neighbouring pixels considered.

