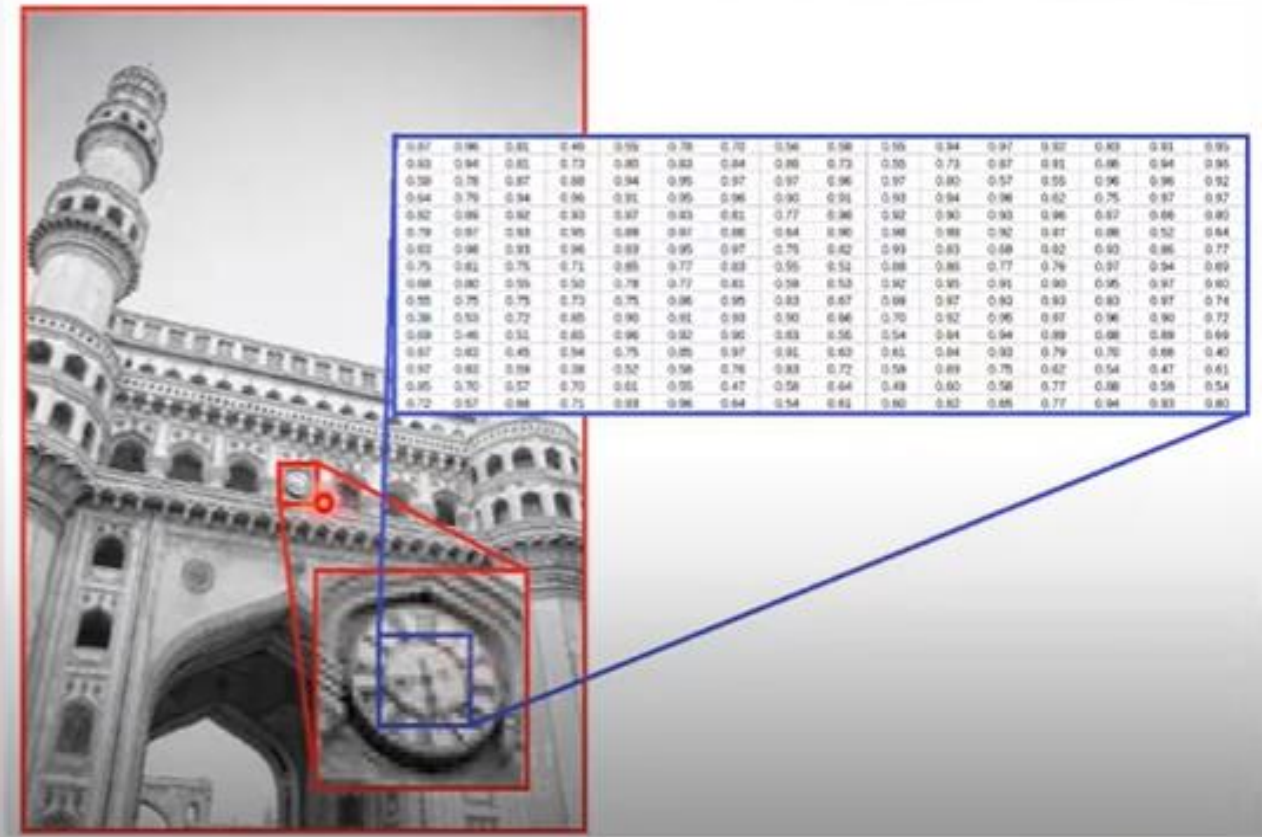
Image Representation

Image Representation

- After getting an image, it is important to devise ways to represent the image. There are various ways by which an image can be represented. Let's look at the most common ways to represent an image.

Image Representation (Cont.)

- **Image as a matrix**
  - The simplest way to represent the image is in the form of a matrix.
  - In fig. 6, we can see that a part of the image, i.e., the clock, has been represented as a matrix. A similar matrix will represent the rest of the image too.



0.97	0.96	0.81	0.46	0.55	0.78	0.70	0.56	0.58	0.55	0.94	0.97	0.92	0.83	0.91	0.95
0.93	0.94	0.81	0.73	0.80	0.83	0.84	0.86	0.73	0.95	0.73	0.87	0.91	0.86	0.94	0.96
0.58	0.78	0.87	0.88	0.94	0.95	0.97	0.97	0.96	0.97	0.80	0.57	0.55	0.96	0.96	0.92
0.64	0.79	0.94	0.96	0.91	0.95	0.96	0.90	0.91	0.93	0.94	0.98	0.62	0.75	0.97	0.97
0.82	0.89	0.92	0.93	0.97	0.93	0.81	0.77	0.98	0.92	0.90	0.93	0.96	0.67	0.66	0.80
0.79	0.97	0.93	0.95	0.89	0.97	0.86	0.64	0.90	0.98	0.98	0.92	0.97	0.88	0.52	0.64
0.83	0.98	0.93	0.96	0.93	0.95	0.97	0.75	0.82	0.93	0.83	0.69	0.92	0.93	0.86	0.77
0.75	0.81	0.75	0.71	0.85	0.77	0.83	0.55	0.51	0.88	0.86	0.77	0.78	0.97	0.94	0.89
0.68	0.80	0.55	0.50	0.78	0.77	0.81	0.59	0.53	0.92	0.95	0.91	0.90	0.95	0.97	0.80
0.55	0.75	0.75	0.73	0.75	0.86	0.95	0.83	0.67	0.89	0.97	0.93	0.93	0.93	0.97	0.74
0.38	0.53	0.72	0.85	0.90	0.91	0.93	0.90	0.86	0.70	0.92	0.95	0.97	0.98	0.90	0.72
0.69	0.46	0.51	0.85	0.96	0.92	0.90	0.83	0.55	0.54	0.84	0.94	0.89	0.88	0.89	0.69
0.87	0.83	0.45	0.54	0.75	0.85	0.97	0.91	0.63	0.61	0.84	0.93	0.79	0.70	0.66	0.40
0.97	0.83	0.59	0.38	0.52	0.58	0.76	0.83	0.72	0.59	0.68	0.75	0.62	0.54	0.47	0.61
0.85	0.70	0.57	0.70	0.63	0.55	0.47	0.58	0.64	0.49	0.60	0.58	0.77	0.88	0.59	0.54
0.72	0.57	0.66	0.71	0.93	0.96	0.68	0.54	0.61	0.60	0.82	0.65	0.77	0.84	0.93	0.80

Image Representation (Cont.)

- It is commonly seen that people use up to a byte to represent every pixel of the image. This means that values between 0 to 255 represent the intensity for each pixel in the image where 0 is black and 255 is white. For every color channel in the image, one such matrix is generated.

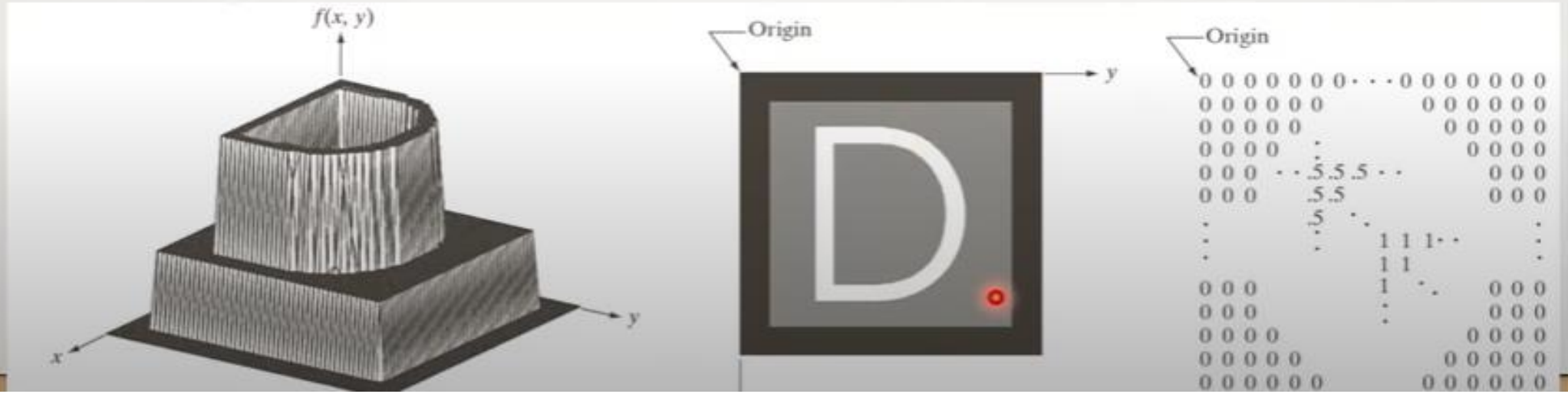


Image Representation (Cont.)

- Image as a function
  - An image can also be represented as a function. An image (grayscale) can be thought of as a function that takes in a pixel coordinate and gives the intensity at that pixel.
  - It can be written as function $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ that outputs the intensity at any input point (x,y). The value of intensity can be between 0 to 255 or 0 to 1 if values are normalized

Image Processing Operations

Image Processing Operations

- Point Operations
  - Output value at (m_0, n_0) is dependent only on the input value at the same coordinate
  - *Complexity/pixel:* Constant
- Local Operations
  - Output value at (m_0, n_0) is dependent on input values in a $p \times p$ neighborhood of that same coordinate
  - *Complexity/pixel:* p^2
- Global Operations
  - Output value at (m_0, n_0) is dependent on on all the values in the input $N \times N$ image
  - *Complexity/pixel:* N^2

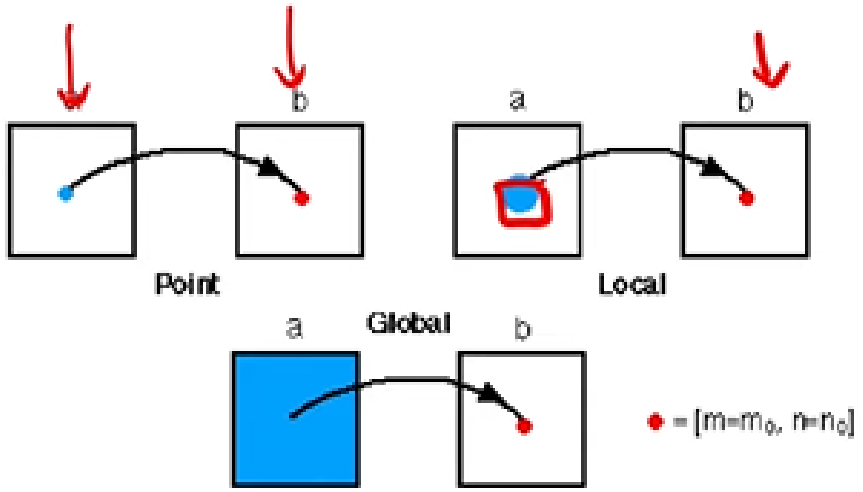


Image Contrast

CONTRAST STRETCHING

- Contrast stretching is a technique used to enhance the visual quality of an image by adjusting its intensity levels.
- This results in a more visually appealing image, with enhanced contrast and improved visibility of details.

WHY CONTRAST STRETCHING

- Corrects non-uniform lighting conditions and improves the visibility of features in an image.

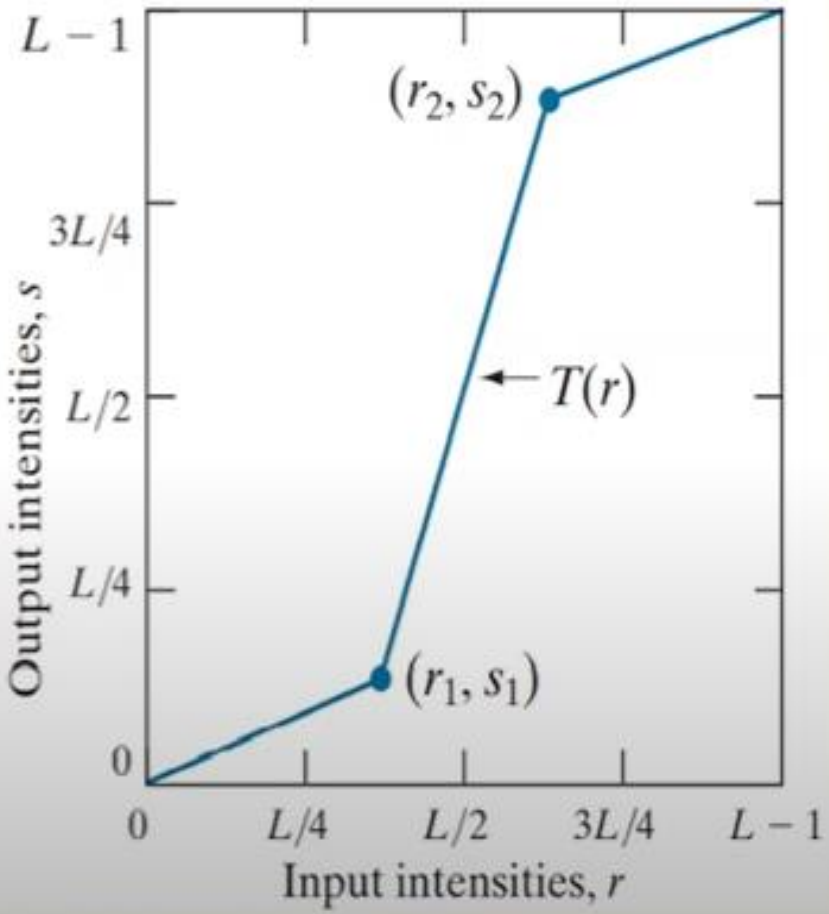
- By applying contrast stretching, we can expand the dynamic range of the image and make it more visually appealing.

- Transformation function is represented as

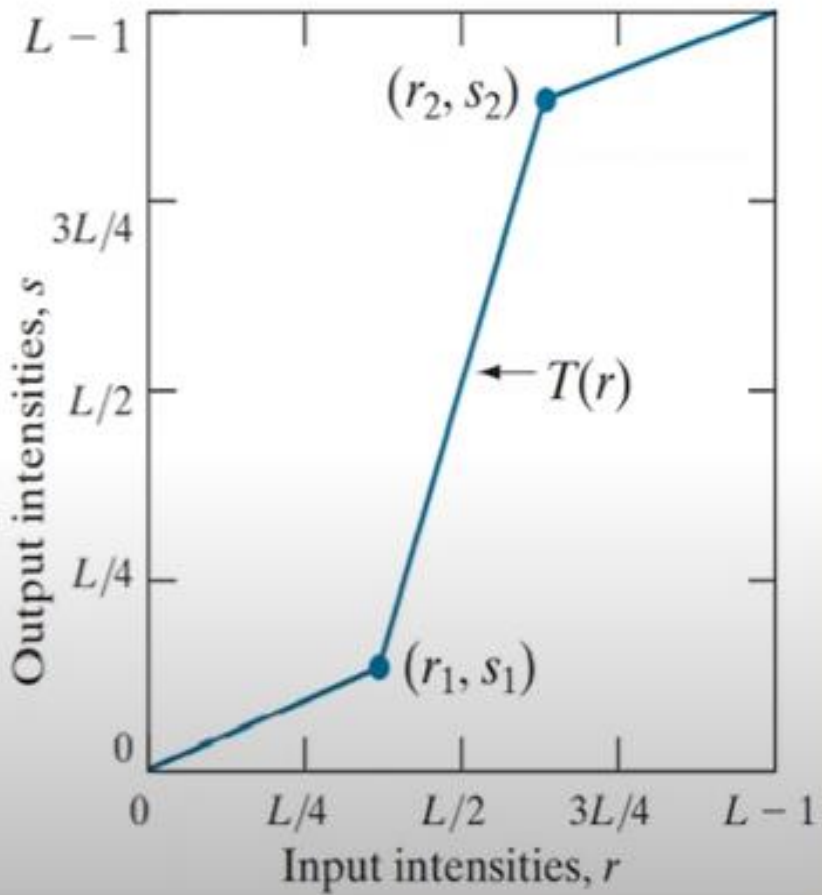
$$S=T(r)$$

- where r is the intensity level of the input pixel and s is the intensity level of the output pixel after transformation and T is the transformation function

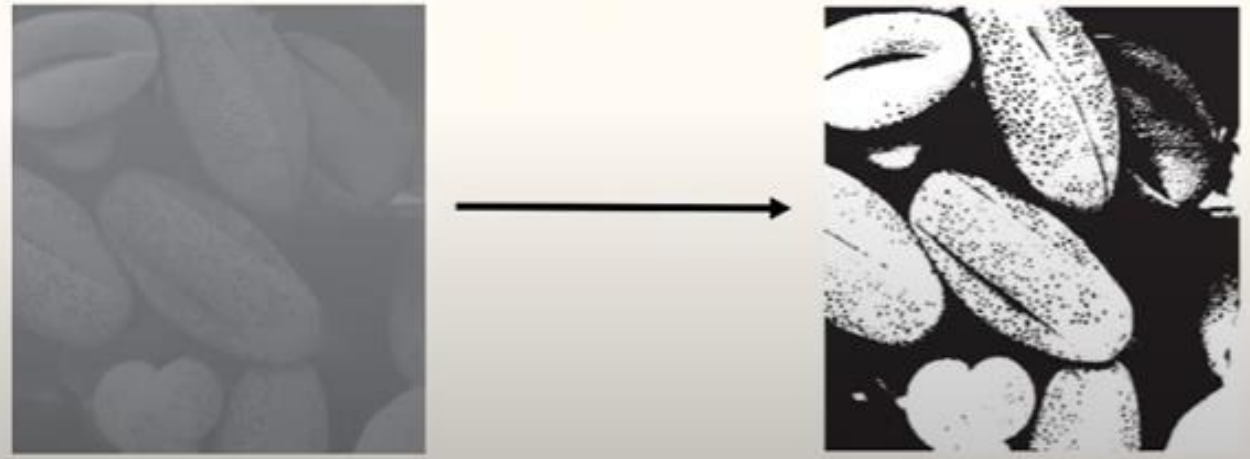
- The locations of points (r1,s1) and (r2,s2) control the shape of the transformation function.

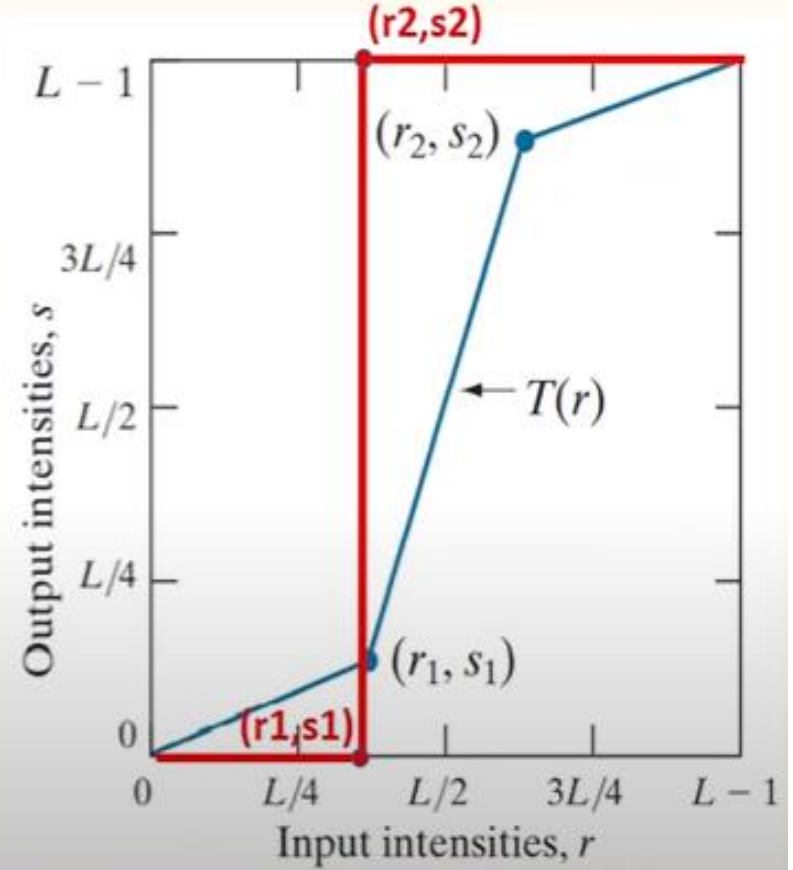


- Case 1: if r1 = s1 and r2=s2
- it means that there is no change in the intensity level after transformation
- the transformation is a linear function that produces no changes in intensity

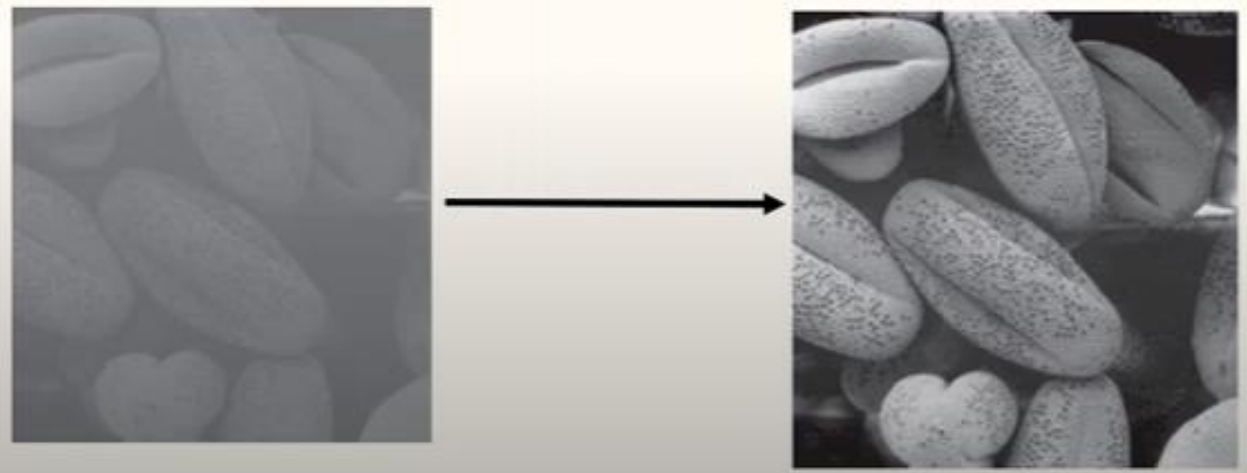


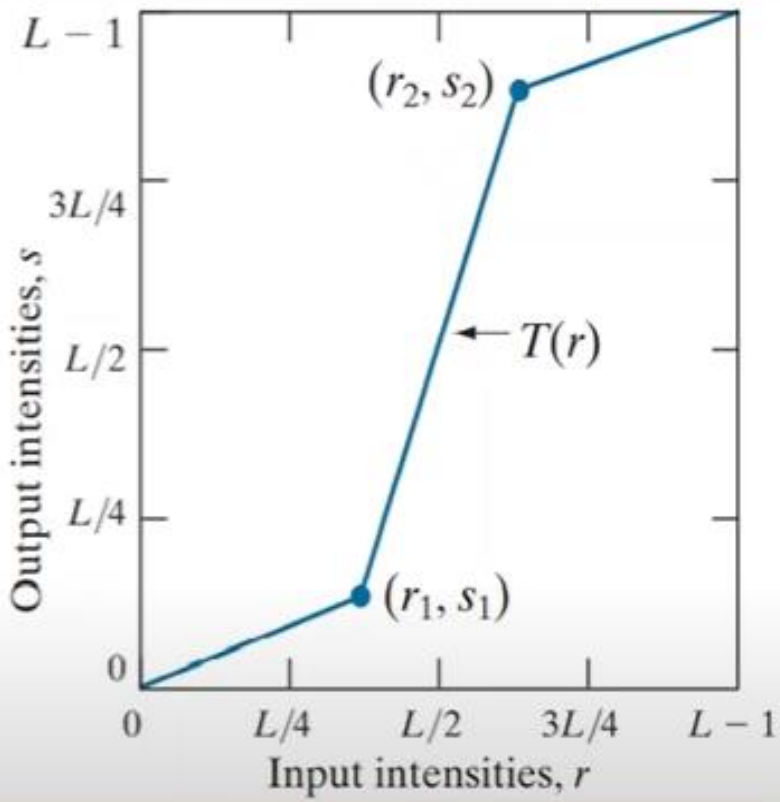
- Case 2: if r1 = r2 and s1=0, s2=L-1
- The transformation becomes a thresholding function that creates a binary image.
- Thresholding function can be created with (r1,s1) =(m,0) and (r2,s2)=(m,L-1) where m is the mean intensity level in the image





- Case 3: if (r1,s1) = (r_{min},0) and (r2,s2) = (r_{max},L-1)
- where r_{min} and r_{max} denote the minimum and maximum intensity levels in the input image
- The transformation results in contrast stretching
- The transformation stretched the intensity levels linearly to the full intensity range, [0,L-1]





LIMITATIONS

- Loss of information if the intensity levels are stretched too far.
- May not always produce the desired results and may require fine-tuning to achieve the best results.

HISTOGRAM SPECIFICATION/MATCHING

HISTOGRAM SPECIFICATION/MATCHING

- Equalize the levels of the original image.
- **Histogram matching** is the transformation of an image.
- The process of Histogram Matching takes in an input image and produces an output image that is based upon a specified histogram.
- The well-known **histogram** equalization method is a special case in which the specified **histogram** is uniformly distributed.

Note

- Histogram equalization has a disadvantage:
- It can generate only one type of output image.
- With **histogram specification** we can specify the shape of the histogram that we wish the output image to have.
- It doesn't have to be a uniform histogram.
- Histogram specification is a trial-and-error process.
- There are no rules for specifying histograms, and one must resort to analysis on a case-by-case basis for any given enhancement task.

Histogram Specification/Matching

Here we want to convert the image so that it has a particular histogram that can be arbitrarily specified. Such a mapping function can be found in three steps:

1. Equalize the histogram of the input image
2. Equalize the specified histogram
3. Relate the two equalized histograms

EXAMPLE: HISTOGRAM MATCHING

Suppose that a 3-bit image (L=8) of size 64 × 64 pixels (MN = 4096) has the intensity distribution shown in the following table (on the left). Get the histogram transformation function and make the output image with the specified histogram, listed in the table on the right.

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

	Specified
z_q	$p_z(z_q)$
$z_0 = 0$	0.00
$z_1 = 1$	0.00
$z_2 = 2$	0.00
$z_3 = 3$	0.15
$z_4 = 4$	0.20
$z_5 = 5$	0.30
$z_6 = 6$	0.20
$z_7 = 7$	0.15

CDF calculation for input histogram

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7 \times 0.19 = 1.33 \longleftrightarrow 1$$

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7 \times (0.19 + 0.25) = 3.08 \longleftrightarrow 3$$

$$s_2 = 4.55 \rightarrow 5 \qquad s_3 = 5.67 \rightarrow 6$$

$$s_4 = 6.23 \rightarrow 6 \qquad s_5 = 6.65 \rightarrow 7$$

$$s_6 = 6.86 \rightarrow 7 \qquad s_7 = 7.00 \rightarrow 7$$

$$s_0 = 1 \quad s_2 = 5 \quad s_4 = 6 \quad s_6 = 7$$

$$s_1 = 3 \quad s_3 = 6 \quad s_5 = 7 \quad s_7 = 7$$

CDF calculation for specified histogram

$$G(z_0) = 7\sum_{j=0}^{0} p_z(z_j) = 0.00$$

Similarly,

$$G(z_1) = 7\sum_{j=0}^{1} p_z(z_j) = 7\left[p(z_0) + p(z_1)\right] = 0.00$$

and

$$G(z_2) = 0.00 \quad G(z_4) = 2.45 \quad G(z_6) = 5.95$$

$$G(z_3) = 1.05 \quad G(z_5) = 4.55 \quad G(z_7) = 7.00$$

CDF calculation for specified histogram

$G(z_0) = 0.00 \rightarrow 0$

$G(z_1) = 0.00 \rightarrow 0$

$G(z_2) = 0.00 \rightarrow 0$

$G(z_3) = 1.05 \rightarrow 1$

$G(z_4) = 2.45 \rightarrow 2$

$G(z_5) = 4.55 \rightarrow 5$

$G(z_6) = 5.95 \rightarrow 6$

$G(z_7) = 7.00 \rightarrow 7$

z_q	$G(z_q)$
$z_0 = 0$	0
$z_1 = 1$	0
$z_2 = 2$	0
$z_3 = 3$	1
$z_4 = 4$	2
$z_5 = 5$	5
$z_6 = 6$	6
$z_7 = 7$	7

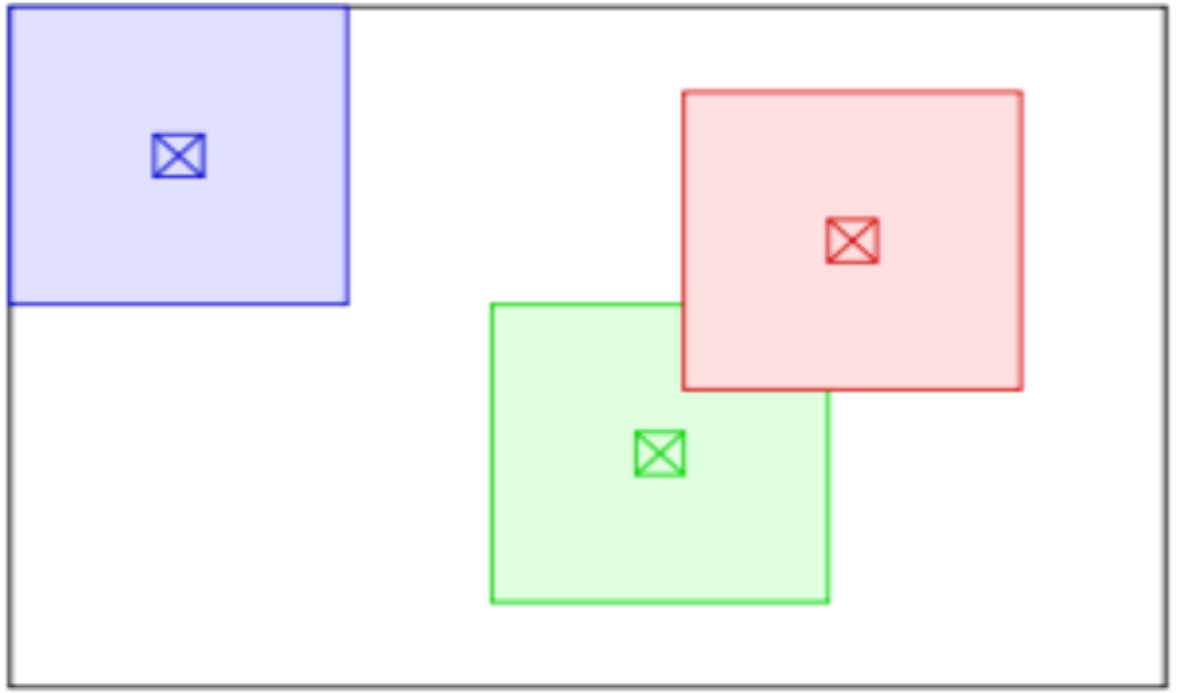
Histogram Matching / Specification

Gray Levels rk	Number of Pixels (Input Image) nk	Equalized Mapping of Input Image (S)	Equalized Mapping of Specified Image G(Zq)	New Mapping	Number of Pixels After Matching
0	790	1	0	3	790
1	1023	3	0	4	1029
2	850	5	0	5	850
3	656	6	1	6	985
4	329	6	2	6	
5	245	7	5	7	448
6	122	7	6	7	
7	81	7	7	7	

Adaptive histogram equalization

Adaptive histogram equalization

Adaptive histogram equalization (AHE) is a computer image processing technique used to improve contrast in images. It differs from ordinary histogram equalization in the respect that the adaptive method computes several histograms, each corresponding to a distinct section of the image, and uses them to redistribute the lightness values of the image. It is therefore suitable for improving the local contrast and enhancing the definitions of edges in each region of an image.



Motivation and explanation of the method

Ordinary histogram equalization uses the same transformation derived from the image histogram to transform all pixels. This works well when the distribution of pixel values is similar throughout the image. However, when the image contains regions that are significantly lighter or darker than most of the image, the contrast in those regions will not be sufficiently enhanced.

Adaptive histogram equalization (AHE) improves on this by transforming each pixel with a transformation function derived from a neighborhood region. It was first developed for use in aircraft cockpit displays. In its simplest form, each pixel is transformed based on the histogram of a square surrounding the pixel, as in the figure below. The derivation of the transformation functions from the histograms is exactly the same as for ordinary histogram equalization: The transformation function is proportional to the cumulative distribution function (CDF) of pixel values in the neighborhood.

Properties of AHE

•The size of the neighbourhood region is a parameter of the method. It constitutes a characteristic length scale: contrast at smaller scales is enhanced, while contrast at larger scales is reduced.

•Due to the nature of histogram equalization, the result value of a pixel under AHE is proportional to its rank among the pixels in its neighbourhood. This allows an efficient implementation on specialist hardware that can compare the center pixel with all other pixels in the neighbourhood. An unnormalized result value can be computed by adding 2 for each pixel with a smaller value than the center pixel, and adding 1 for each pixel with equal value.

•When the image region containing a pixel's neighbourhood is fairly homogeneous regarding to intensities, its histogram will be strongly peaked, and the transformation function will map a narrow range of pixel values to the whole range of the result image. This causes AHE to overamplify small amounts of noise in largely homogeneous regions of the image.

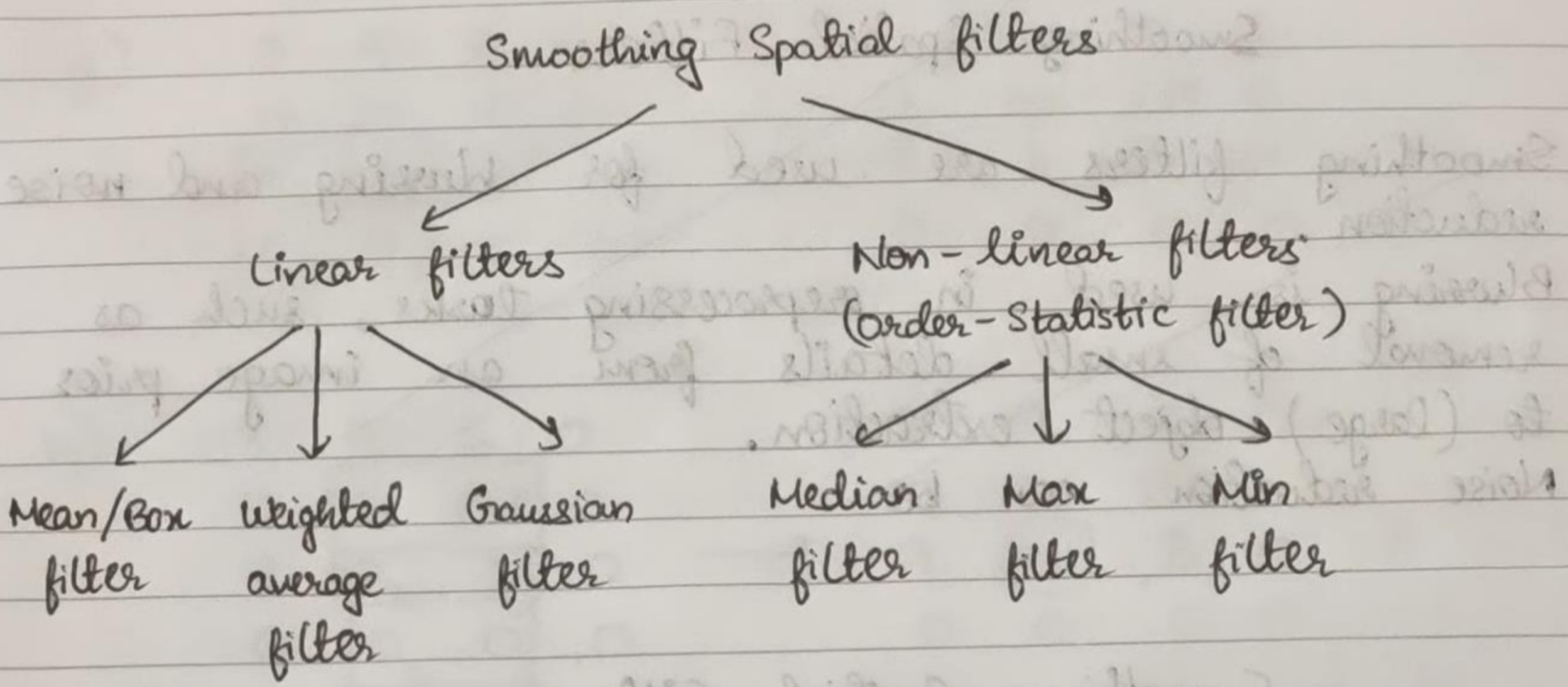
Smoothing Spatial Filters

Smoothing Spatial Filters

→ Smoothing filters are used for blurring and 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 accomplished by blurring with a linear filter and also by non-linear filtering.


Smoothing Spatial filters
Linear filters
Non-linear filters
(Order-Statistic filter)
Mean/Box filter
Weighted average filter
Gaussian filter
Median filter
Max filter
Min filter

* Smoothing linear filters

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

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

1) Box filter - all coefficients are equal.

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

2) Weighted average - 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:

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

(2D Gaussian function)

$\sigma \rightarrow$ Standard deviation

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_{3\times3} \rightarrow \text{Mask}$$

→ Used to blur edges and reduce contrast.
→ Similar to median filter but is faster.

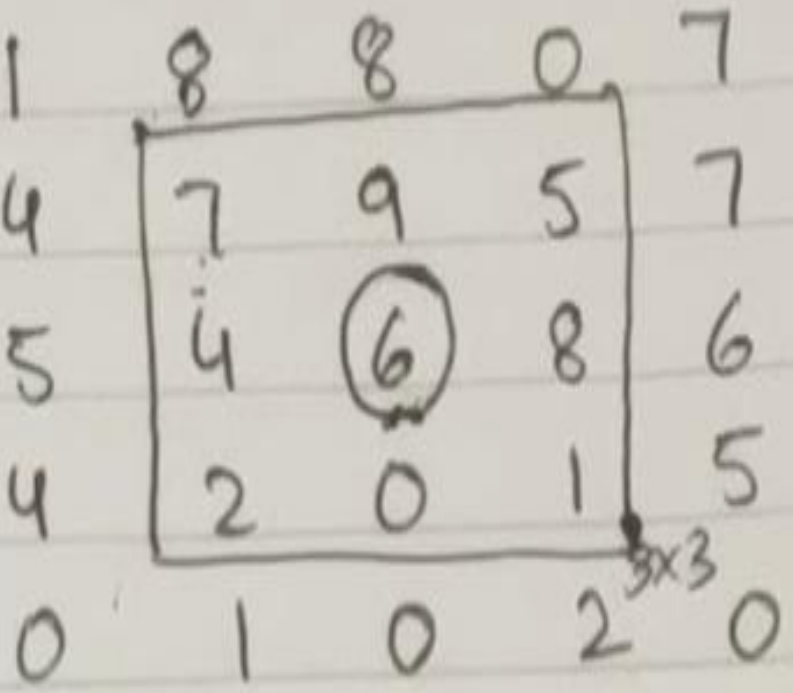
* Non-linear (Order-Statistic) filters

Their response is based on ordering (ranking) 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 filter - Find the median of all the pixel values.
2) Min filter - Find the minimum of all the pixel values.
3) Max filter - Find the maximum of all the pixel values.

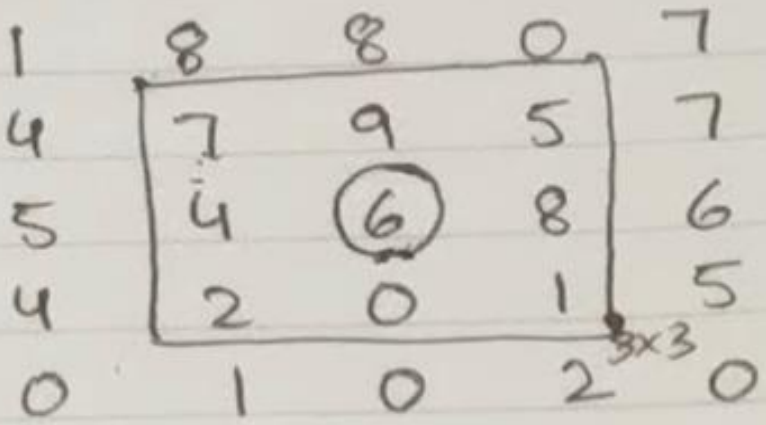
Q1. Consider the image below and calculate the output of the pixel (2,2) if smoothing is done using 3x3 neighbourhood using all the filters below:

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



Q1. Consider the image below and calculate the output of the pixel (2,2) if smoothing is done using 3×3 neighbourhood using all the filters below:

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



a) Box filter

$$= \frac{1}{9} \times [7+9+5+4+6+8+2+0+1] \quad \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \frac{1}{9} \times [42] = 4.66 \approx \underline{5}.$$

b) Weighted average filter

$$= \frac{1}{16}\Big[7\times1 + 9\times2 + 5\times1 + 4\times2 + 4\times6 + 8\times2 + 2\times1 + 0\times2 + 1\times1\Big]$$

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

$$= \frac{1}{16}[81] = 5.0625 \approx \underline{5}.$$

c) Median filter

0, 1, 2, 4, (5), 6, 7, 8, 9

Median = $\underline{5}$.

d) Min filter

= $\underline{0}$.

e) Max filter

= $\underline{9}$.

Q1. Why median filter is better than mean filter?

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 filter in preserving useful detail in an image.

Median filter has 2 main advantages:

1) The median is a more robust average than the mean and so a single very unrepresentative pixel in a neighborhood will not affect the median value significantly.

2) 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.

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.

https://www.youtube.com/watch?v=x6zoQ-a7A9U

Sharpening Spatial Filters

→ The principal objective of sharpening is to highlight transitions in intensity.

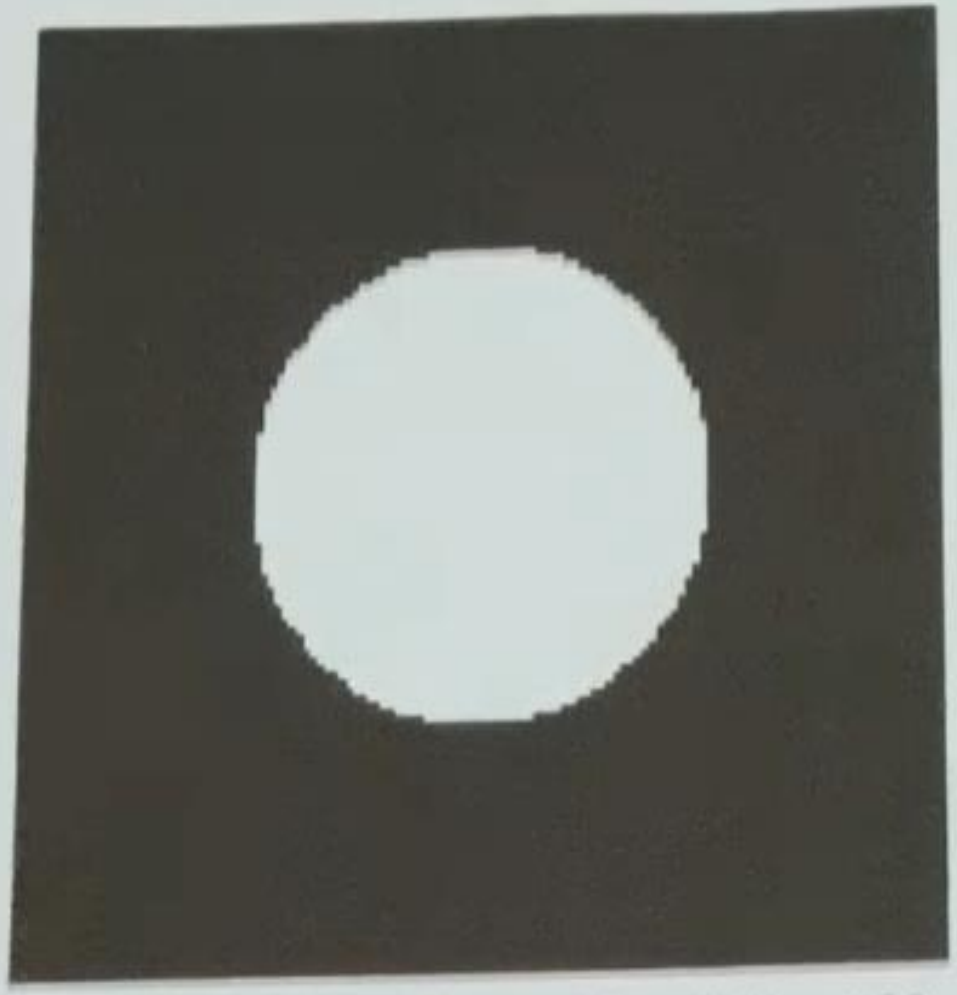
→ Applications of image sharpening include electronic printing, medical imaging, industrial inspection and autonomous guidance in military systems.

Blurring → Pixel averaging

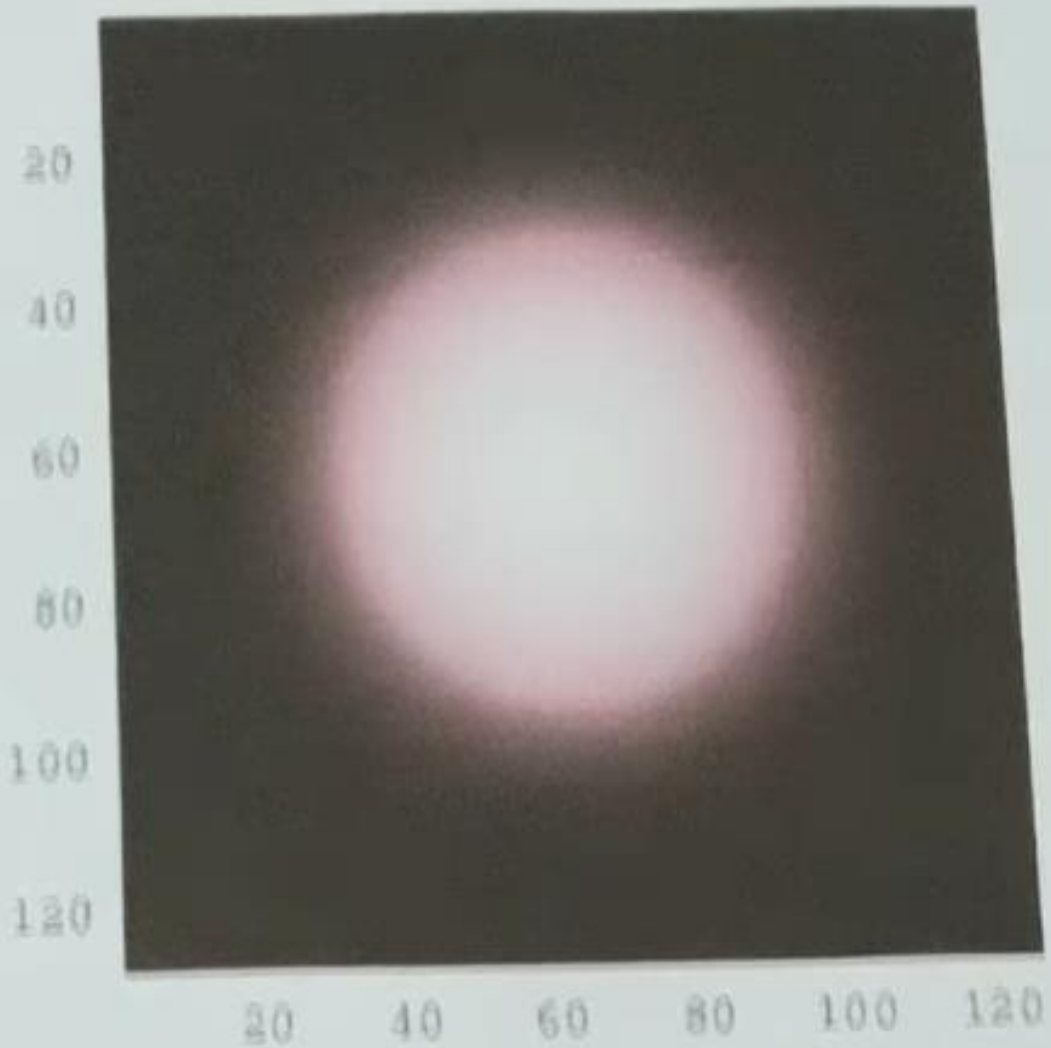
Sharpening → Spatial diffentiation

→ The strength of the response of a derivatino operator is proportional to the degree of intensity discontinuity of the image at the point at which the operator is applied.

original



after 4 iterations of 21x21 box fil



→ The strength of the response of a derivative operator is proportional to the degree of intensity discontinuity of the image at the point at which the operator is applied.

→ Therefore, image differentiation enhances edges and other discontinuities (such as noise) and deemphasizes areas which have slowly varying intensities.

Foundation of sharpening filters

1) First-order derivative of a one-dimensional function $f(x)$:

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

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

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

Laplacian filter

→ It highlights gray-level discontinuities in an image

→ It deemphasizes regions with slowly varying gray levels.

→ Formula: $\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$

$f(x)$

$f(x, y)$

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

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

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

* Laplacian mask

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

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

-1	-1	-1
-1	8	-1
-1	-1	-1

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

Input image

0	1	0
1	-4	1
0	1	0

Q1. Apply Laplacian filter on the given image on the center pixel.

8	5	4
0	(6)	2
1	3	7

Input image

$*$

0	1	0
1	-4	1
0	1	0

Mask

$= (8\times0) + (5\times1) + (4\times0)$
$+ (0\times1) + (6\times-4) + (2\times1)$
$+ (1\times0) + (3\times1) + (7\times0)$

$= 0 + 5 + 0 + 0 - 24 +$
$2 + 0 + 3 + 0$
$= 10 - 24 = -14.$

Enhanced Laplacian Filter

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

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

Q2. Apply enhanced laplacian filter on the given image on the center pixel.

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

$$= 8 + 5 + 4 + 0 - 54 + 2 + 1 + 3 + 7$$

$$= 30 - 54 = \underline{\underline{-24}}.$$