Computer Vision
01CE0612

Unit - 2
Image Processing and Enhancement

Dr. Anjali Diwan
Department of Computer Engineering

Content of Unit 2

- Digital Image Fundamentals,
- **Image Enhancement Techniques**
  - Histogram Equalization, Contrast Stretching, etc.
- **Image Filtering**
  - Spatial Filters
- **Noise Reduction Techniques**
  - **Smoothing Filters**
- **Image Sharpening**
  - Laplacian and Gradient-Based Techniques

Outline

- What is Image Enhancement?
- Image Enhancement Techniques
  - Spatial Domain Methods
    - Intensity (Gray-level) transformations functions
    - Histogram Processing
    - Spatial Filtering

Image Enhancement

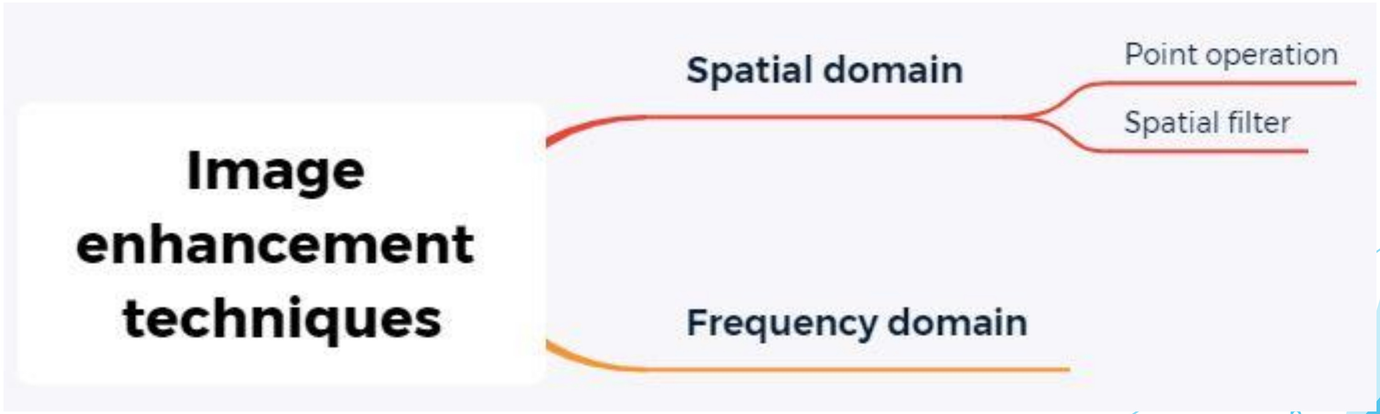
- Image enhancement refers to the process of highlighting certain information of an image, as well as weakening or removing any unnecessary information according to specific needs.
- For example, eliminating noise, revealing blurred details, and adjusting levels to highlight features of an image.

Image Enhancement (Cont.)

- It is to process an image so that the result is more suitable than the original image for a specific application.
- The idea behind enhancement techniques is to **bring out details that are hidden, or simple to highlight certain features of interest** in an image.

Image Enhancement Techniques

- Image enhancement techniques can be divided into two broad categories:
  - Spatial Domain
  - Frequency Domain



Spatial Domain

- It is an enhancement of the image space that divides an image into uniform pixels according to the spatial coordinates with a particular resolution.
- The spatial domain methods perform operations on pixels directly.

Frequency Domain

- It an enhancement achieved by applying the Fourier Transform to the spatial domain.
- In the frequency domain, pixels are operated in groups as well as indirectly.

Spatial Domain Technique

- **Intensity Transformation Techniques / Point Operation**
  - Point operations refer to running the same conversion operation for each pixel in a grayscale image.
  - The transformation is based on the original pixel and is independent of its location or neighboring pixels.
- **Spatial Filtering**
  - The output value depends on the values of f(x,y) and its neighborhood.

Intensity Transformation Techniques / Point Operation

- Point operations are often used to change the grayscale range and distribution.
- The concept of point operation is to map every pixel onto a new image with a predefined transformation function.

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

Where,

- $g(x, y)$ is the output image
- T is an operator of intensity transformation
- $f(x, y)$ is the input image

Intensity Transformation Techniques

Intensity Transformation Functions Fall Into 2 Approaches:

- Basic Intensity Transformation / Grey Level Transformation
- Piecewise Linear Transformation

Basic Intensity Transformations

- The simplest image enhancement method is to use a 1 x 1 neighborhood size. It is a point operation.
- In this case, the output pixel (‘s’) only depends on the input pixel (‘r’), and the point operation function can be simplified as follows:

$$s = T(r)$$

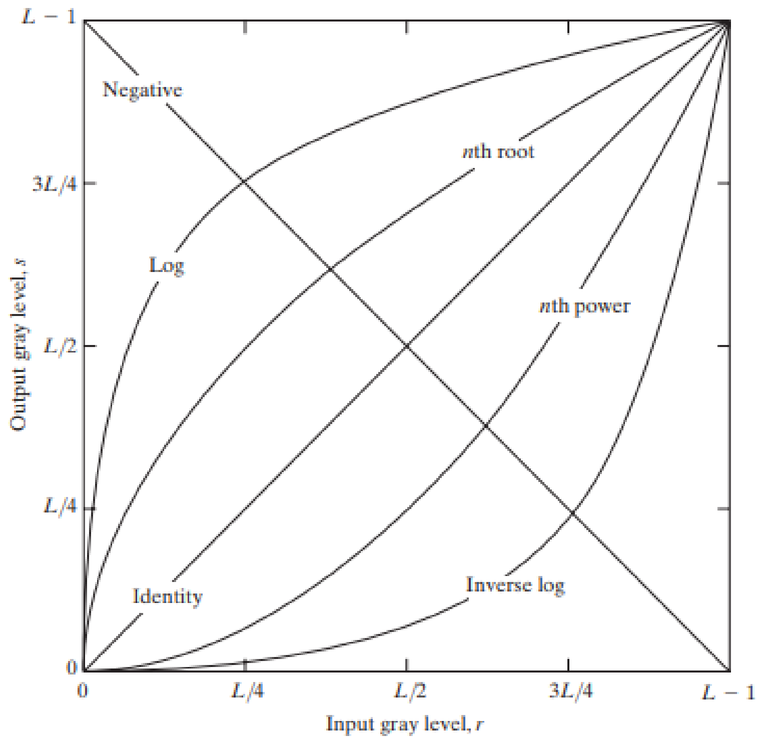
- Where
  - T is the point operator of a certain gray-level mapping relationship between the original image and the output image.
  - s,r: denote the gray level of the input pixel and the output pixel.

Basic Intensity Transformations

- **Linear Functions**
  - Identity Transformation
  - Negative Transformation
- **Logarithmic Functions**
  - Log Transformation
  - Inverse-log Transformation
- **Power-law / Gamma / Exponential Functions**
  - Nth Power Transformation
  - Nth Root Transformation

Basic Intensity Transformations

- Different transformation functions work for different scenarios.



Identity Transformation

- Output intensities are identical to input intensities
- This function doesn't have an effect on an image, it was included in the graph only for completeness
- In identity transformation, the input image is the same as the output image.

$$s = r$$

Negative Transformation

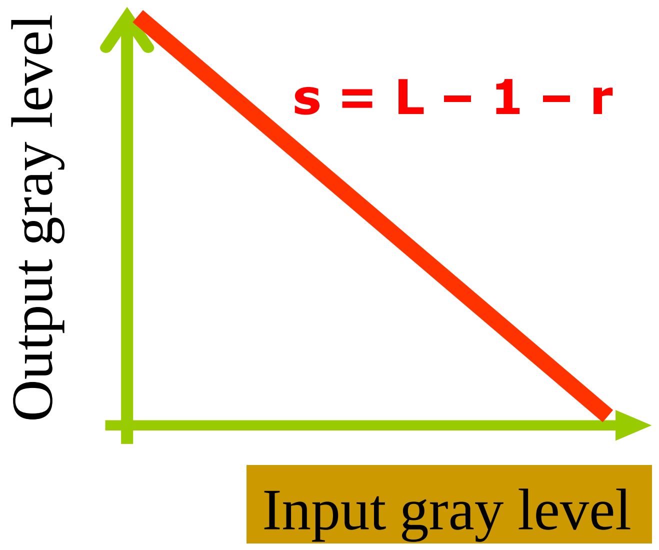
- The negative of an image with gray level in the range [0, L-1], where L = Largest value in an image, is obtained by using the negative transformation's expression:

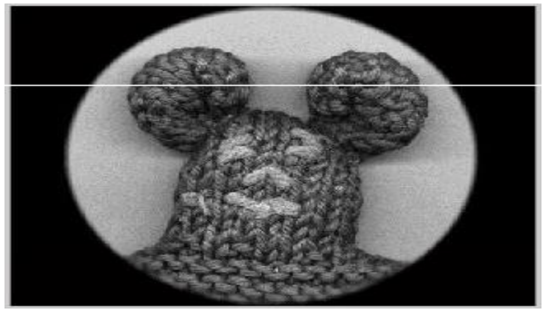
$$s = L - 1 - r$$

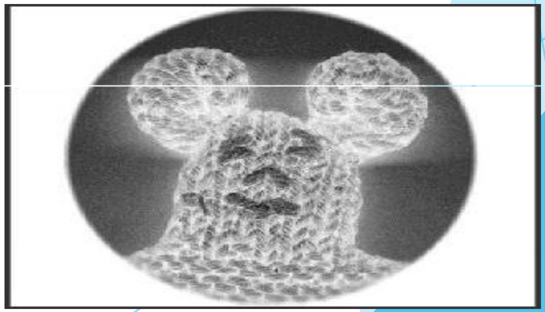
- Negative transformation reverses the intensity levels of an input image , in this manner produces the equivalent of a photographic negative.

Negative Transformation (Cont.)

- The negative transformation is suitable for enhancing white or gray detail embedded in dark regions of an image, especially when the black area are dominant in size

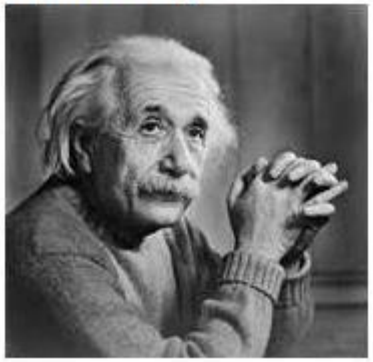




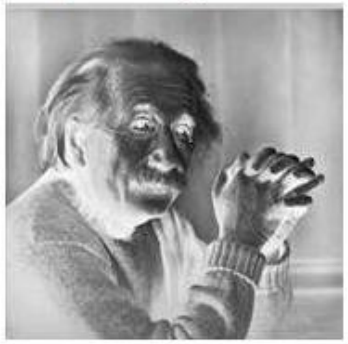


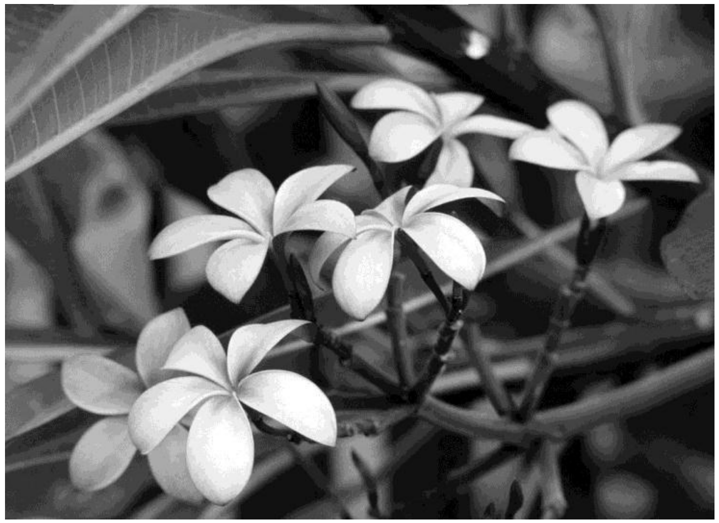
Negative Transformation (Cont.)

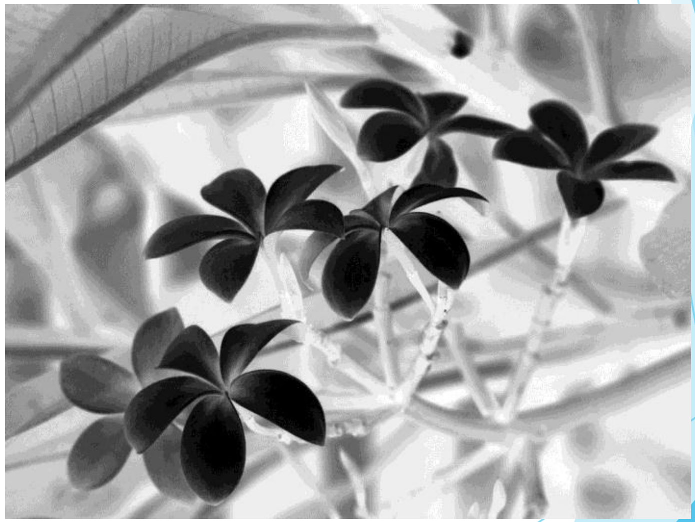
Input Image



Output Image







Advantages of negative :

✓ Produces an equivalent of a photographic negative.

✓ Enhances white or gray detail embedded in dark regions.

Negative Transformation (Cont.)

- Example 1: The following matrix represents the pixels values of an 8-bit image (r) , apply negative transform and find the resulting image pixel values.

Image (r)

100	110	90	95
98	140	145	135
89	90	88	85
102	105	99	115

Solution:

$L = 2^8 = 256$

$s = L-1-r$

$s = 255-r$

Apply this transform to each pixel to find the negative

Image (s)

155	145	165	160
157	115	110	120
166	165	167	170
153	150	156	140

Negative Transformation (Cont.)

- Example 2: the following matrix represents the pixels values of a 5-bit image (r) , apply negative transform and find the resulting image pixel values.

Image (r)

21	26	29	30
19	21	20	30
16	16	26	31
19	18	27	23

Solution:

$L = 2^5 = 32$

$s = L - 1 - r$

$s = 31 - r$

Apply this transform to each pixel to find the negative

Image (s)

10	5	2	1
12	10	11	1
15	15	5	0
12	13	4	9

Negative Transformation (Cont.)

- The negative of an image can be obtained also with Image Processing Toolkit function imcomplement:

g = imcomplement (f);

Log Transformation

- The equation of general log transformation is:

$$s = c * \log(1 + r)$$

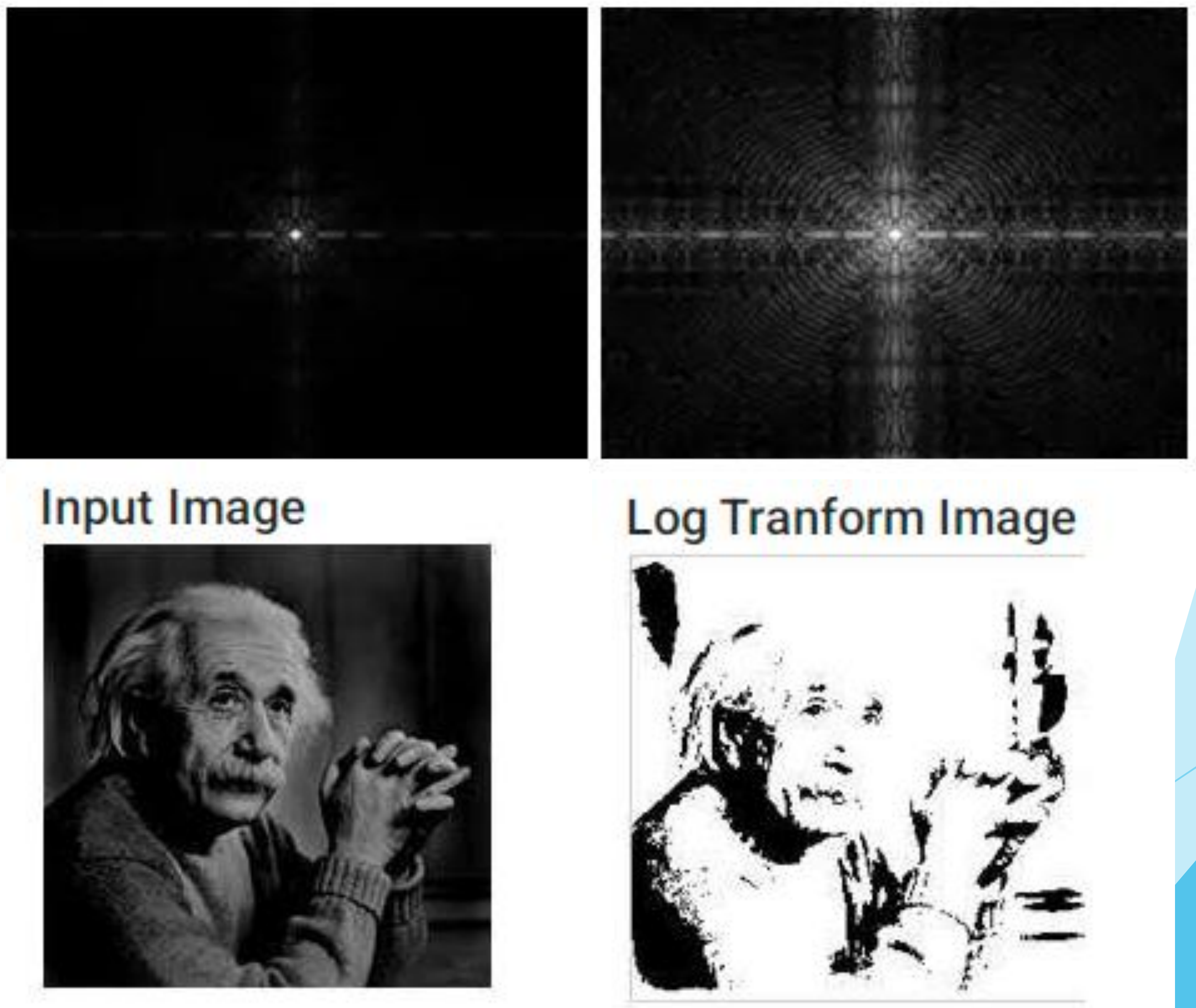
- Note:
  - s,r: denote the gray level of the input pixel and the output pixel.
  - ‘c’ is a constant; to map from [0,255] to [0,255],

    $c = 256/LOG(256)$
  - the base of a common logarithm is 10

Log Transformation (Cont.)

- In the log transformation, the low-intensity values are mapped into higher intensity values.
- It maps a narrow range of low gray levels to a much wider range.
- The log transformation works the best for dark images.
- It compresses the dynamic range of images with large variations in pixel values.
- Log functions are particularly useful when the input grey level values may have an extremely large range of values

Log Transformation (Cont.)



Log Transformation (Cont.)

- Example 1: The following matrix represents the pixels values of an 8-bit image (r) , apply Log transform and find the resulting image pixel values.

(i) C = 1;

(ii) C = L / Log(1+L)

110	120	90
91	94	98
90	91	99

Marwadi University

Log Transformation (Cont.)

- Logarithmic transformations are implemented using expression:

  g = c * log (1 + double (f))

- But this function changes the data class of the image to double, so another sentence to return it back to uint8 should be done:

- implemented expression:

  gs = im2uint8 (mat2gray(g));

- Use of mat2gray brings the values to the range [0 1] and im2uint8 brings them to the range [0 255]

Inverse-log Transformation

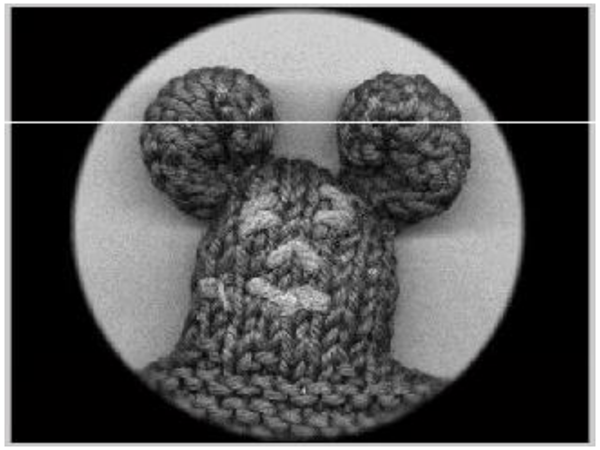
- The inverse log transform is opposite to log transform.
- The inverse log transform expands the values of light-level pixels while compressing the darker-level values.

s = power(10, r * c)-1

- Note:
  - s,r: denote the gray level of the input pixel and the output pixel.
  - ‘c’ is a constant; to map from [0,255] to [0,255], c =LOG(256)/255

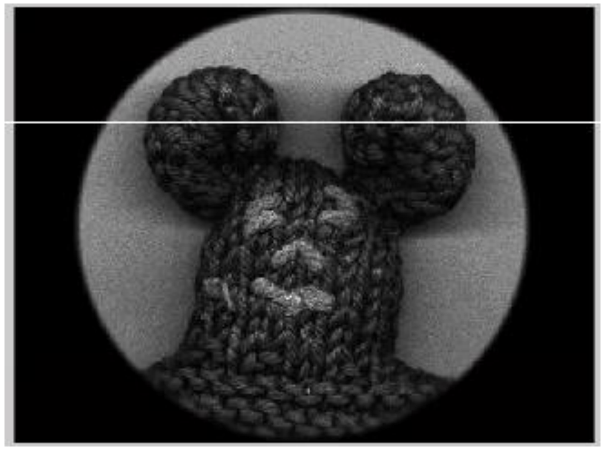
Inverse-log Transformation (Cont.)

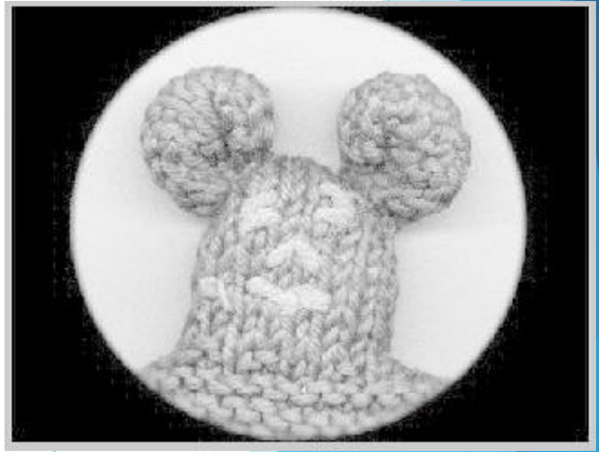
- Used to expand the values of high pixels in an image while compressing the darker-level values.
- It maps a narrow range of high gray levels to a much wider range.



InvLog

Log





Power-law / Gamma / Exponential Transformation

- Power-law(Gamma) transformations have the basic form of:

$$s = c.r^{\gamma}$$

Where c and γ are positive constants

- Variation in the value of γ varies the enhancement of the images. Different display devices / monitors have their own gamma correction, that's why they display their image at different intensity.

Power-law / Gamma / Exponential Transformation (Cont.)

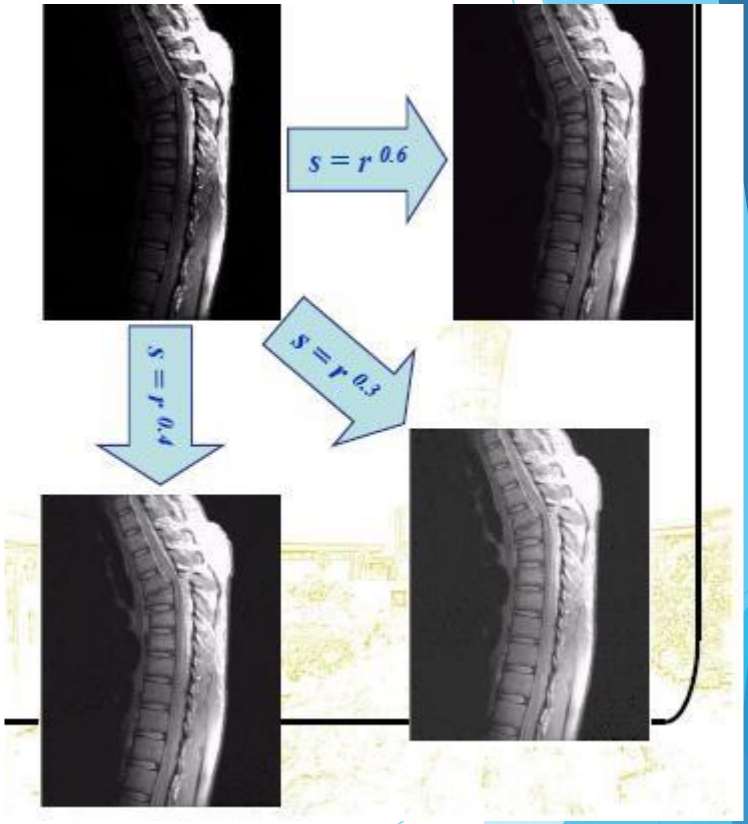
- This type of transformation is used for enhancing images for different type of display devices.
- Map a narrow range of dark input values into a wider range of output values or vice versa
- The gamma of different display devices is different.
- For example Gamma of CRT lies in between of 1.8 to 2.5, that means the image displayed on CRT is dark.

Power-law / Gamma / Exponential Transformation (Cont.)

- This type of transformation is used for enhancing images for different type of display devices.
- The gamma of different display devices is different.
- For example Gamma of CRT lies in between of 1.8 to 2.5, that means the image displayed on CRT is dark.

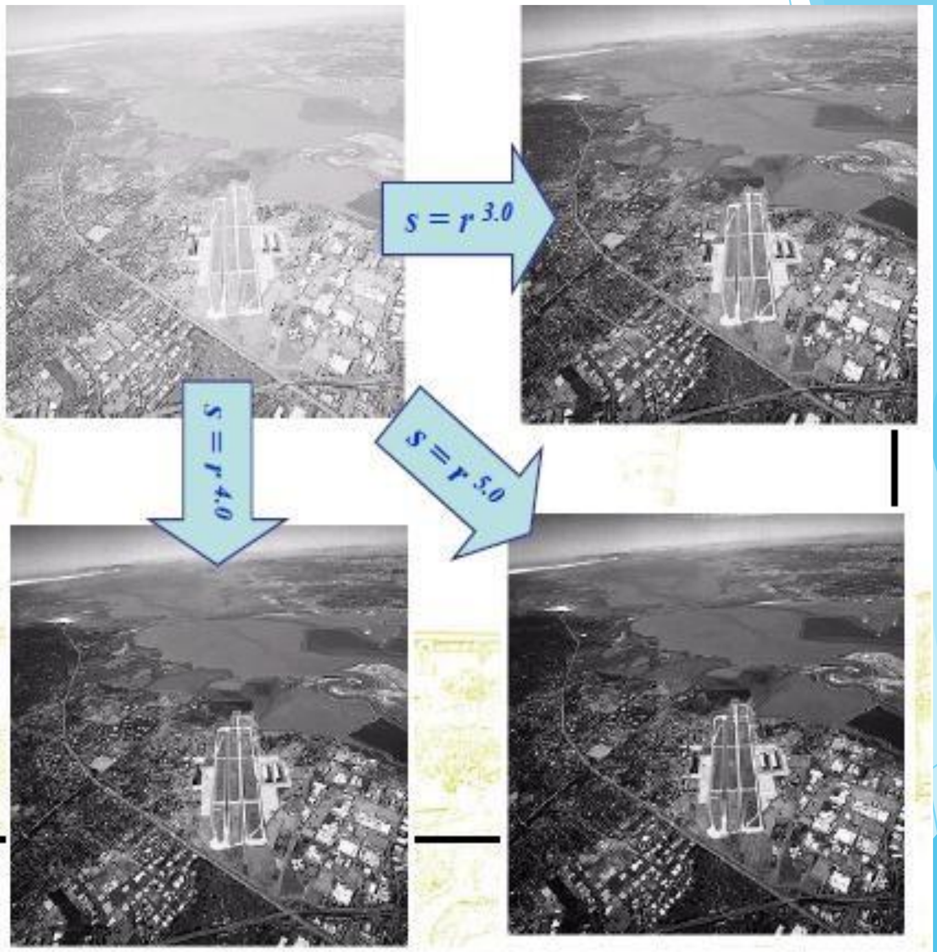
Power-law / Gamma / Exponential Transformation (Cont.)

- The images to the right show a magnetic resonance (MR) image of a fractured human spine
- Different curves highlight different detail



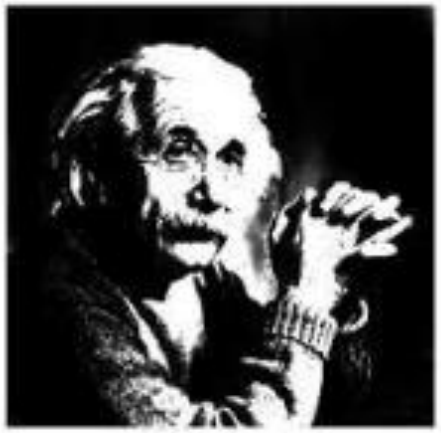
Power-law / Gamma / Exponential Transformation (Cont.)

- An aerial photo of a runway is shown
- This time power law transforms are used to darken the image
- Different curves highlight different detail

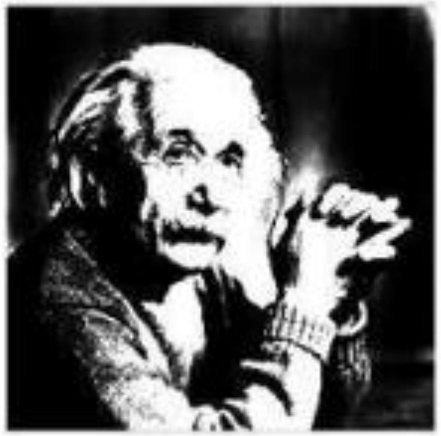


Power-law / Gamma / Exponential Transformation (Cont.)

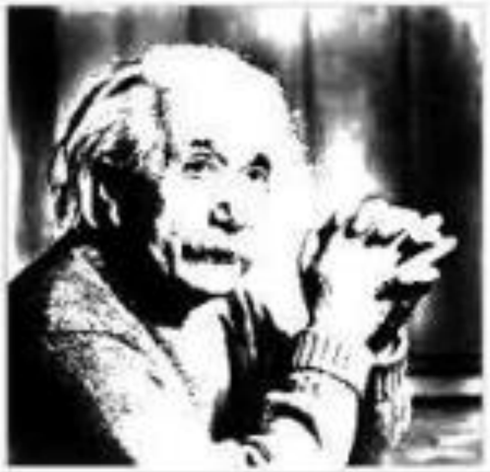
Gamma = 10



Gamma = 8



Gamma = 6



Power-law / Gamma / Exponential Transformation (Cont.)

- Different transformation curves are obtained by varying γ (gamma)

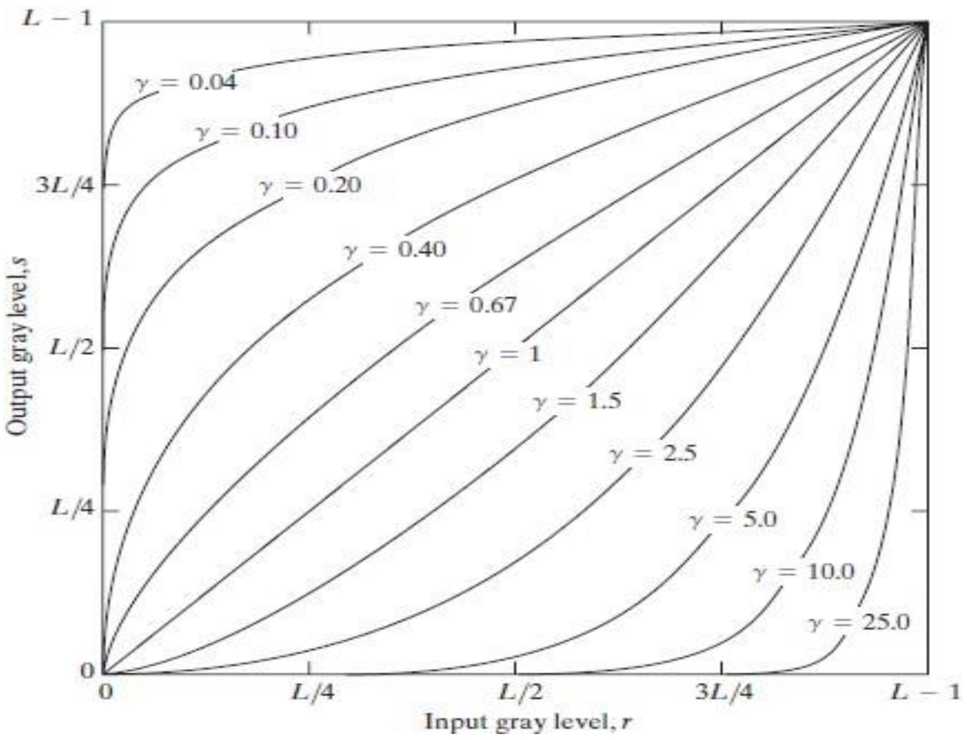


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

Power-law / Gamma / Exponential Transformation (Cont.)

- If gamma <1 :the mapping is weighted toward brighter output values.
- If gamma =1 (default):the mapping is linear.
- If gamma >1 :the mapping is weighted toward darker output values.

Power-law / Gamma / Exponential Transformation (Cont.)

- Example 1: The following matrix represents the pixels values of an 8-bit image (r) , apply Power – Law transform and find the resulting image pixel values.

(i) C = 1; Gamma = 0.2

110	120	90
91	94	98
90	91	99

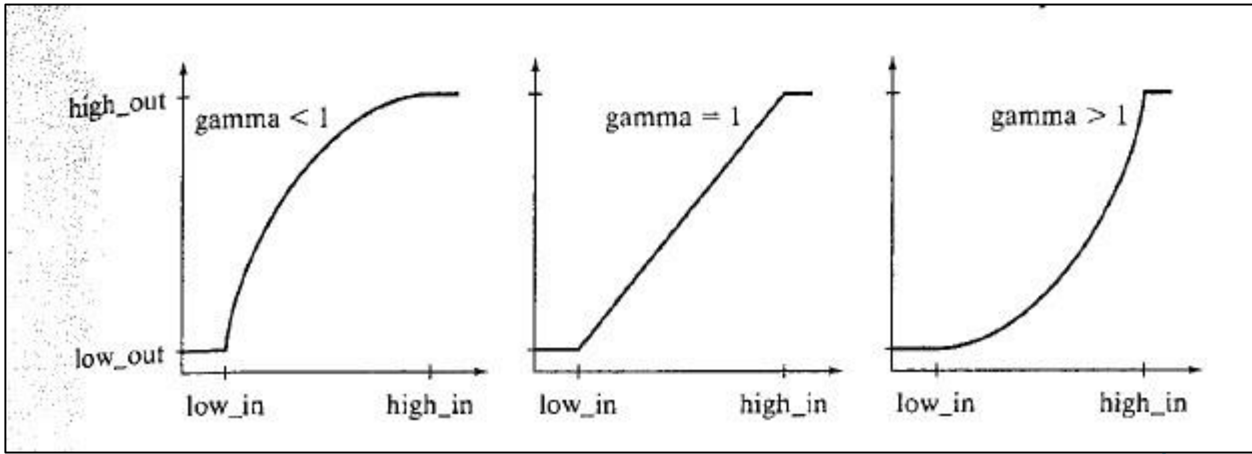
Power-law / Gamma / Exponential Transformation (Cont.)

- Function **imadjust** is the basic IPT tool for intensity transformations of gray-scale images. It has the syntax:

g = imadjust (f, [low_in high_in], [low_out high_out], gamma)

Power-law / Gamma / Exponential Transformation (Cont.)

- As illustrated in figure, this function maps the intensity values in image f to new values in g, such that values between low_in and high_in map to values between low_out and high_out.
- Values below low_in and above high_in are clipped; that is values below low_in map to low_out, and those above high_in map to high_out.



a b c

FIGURE 3.2 The various mappings available in function `imadjust`.

Piecewise Linear Transformation

- Principle Advantage: Some important transformations can be formulated only as a piecewise function.
- Principle Disadvantage: Their specification requires more user input that previous transformations
- In mathematics, a piecewise-defined function is a function defined by multiple sub-functions, where each sub-function applies to a different interval in the domain.

Piecewise Linear Transformation (Cont.)

Types of Piecewise Linear Transformation Function:

- Contrast Stretching
- Thresholding / Grayscale Threshold Transform / Binarization
- Gray-level Slicing
- Bit-plane Slicing

Contrast Stretching

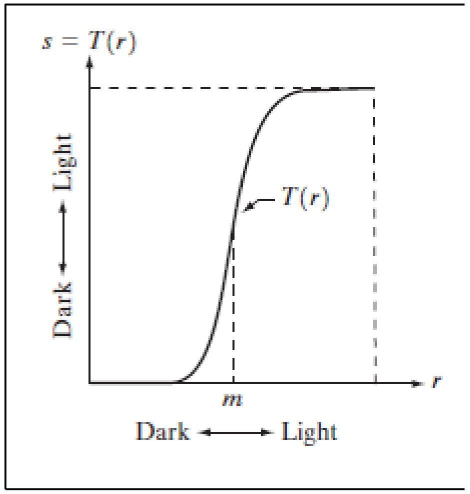
- One of the simplest piecewise linear functions is a contrast-stretching transformation, which is used to enhance the low contrast images.
- Low contrast images may result from:
  - Poor illumination
  - Wrong setting of lens aperture during image acquisition.

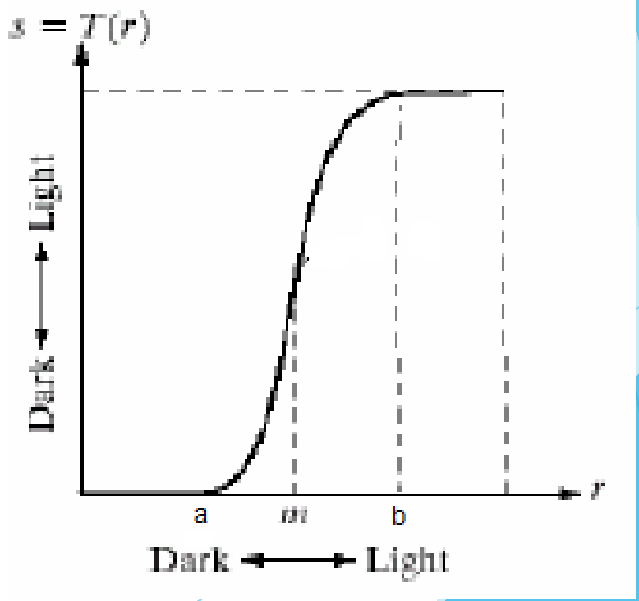
Contrast Stretching (Cont.)

- If T(r) has the form as shown in the figure below, the effect of applying the transformation to every pixel of f to generate the corresponding pixels in g would:
- Produce higher contrast than the original image, by:
  - Darkening the levels below m in the original image
  - Brightening the levels above m in the original image

Contrast Stretching (Cont.)

- So, Contrast Stretching: is a simple image enhancement technique that improves the contrast in an image by 'stretching' the range of intensity values it contains desired range of values.





Contrast Stretching (Cont.)

☝ Remember that:

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

Or

$s = T(r)$

✍ Example: in the graph, suppose we have the following intensities : a=90, b=180, m=100

✓if r is above 180 ,it becomes 255 in s.

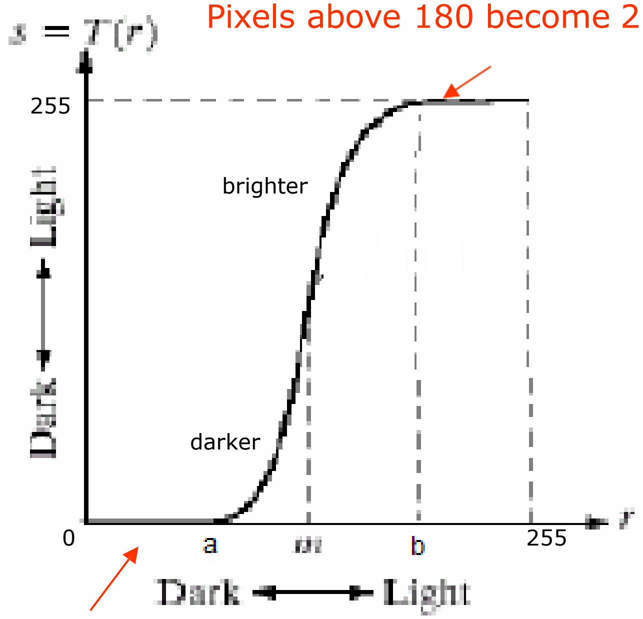
✓If r is below 90 , it becomes 0,

✓If r is between 90, 180 , T applies as follows:

when r < 100 , s closes وق to zero (darker)

when r>100 , s closes to 255 (brighter)

Pixels above 180 become 255



Pixels less than 90 become 0

$$T = \begin{cases} \text{If } r > 180;\ s = 255 \\ \text{If } r < 180 \text{ and } r < 90;\ s = T(r) \\ \text{If } r < 90;\ s = 0 \end{cases}$$

This is called contrast stretching, which means that the bright pixels in the image will become brighter and the dark pixels will become darker, this means : Higher Contrast Image

Contrast Stretching (Cont.)

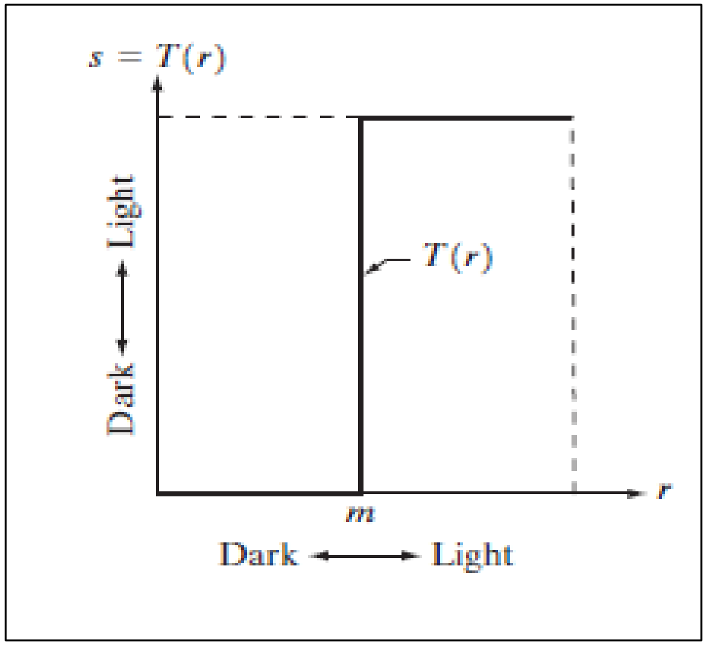
- This equation is implemented in MATLAB for the entire image as

```
g = 1./(1 + (m./(double(f) + eps)).^E)
```

- Note the use of eps to prevent overflow if f has any 0 values.

Thresholding / Grayscale Threshold Transform / Binarization

- Is a limited case of contrast stretching, it produces a two- level (binary) image.



Thresholding / Grayscale Threshold Transform / Binarization (Cont.)

- Thresholding Transform converts a grayscale image into a black and white binary image.
- The user specifies a value that acts as a dividing line.
- If the gray value of a pixel is smaller than the dividing, the intensity of the pixel is set to 0, otherwise it’s set to 255.
- The value of the dividing line is called the threshold.
- The grayscale threshold transform is often referred to as thresholding, or binarization.

Thresholding / Grayscale Threshold Transform / Binarization (Cont.)

Remember that:

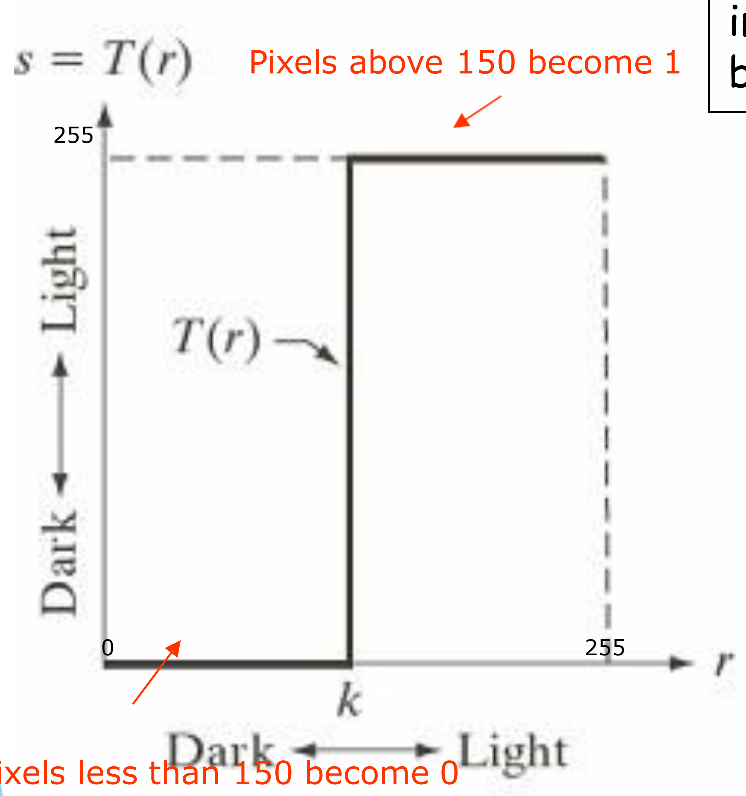
$$g(x,y) = T[f(x,y)]$$

Or

$$s = T(r)$$

Example: suppose m= 150 (called threshold),

if r (or pixel intensity in image f (الصورة الأصلية)) is above this threshold it becomes 1 in s (or pixel intensity in image g (الصورة بعد [illegible])), otherwise it becomes zero.



$$T = \begin{cases} \text{If } f(x,y)>150;\ g(x,y)=1 \\ \text{If } f(x,y)<150;\ g(x,y)=0 \end{cases}$$

Or simply…

$$T = \begin{cases} \text{If } r>150;\ s=1 \\ \text{If } r<150;\ s=0 \end{cases}$$

This is called thresholding, and it produces a binary image!

Thresholding / Grayscale Threshold Transform / Binarization (Cont.)

- Example: The following matrix represents the pixels values of a 8-bit image (r) , apply thresholding transform assuming that the threshold m=95, find the resulting image pixel values.

110	120	90	130
91	94	98	200
90	91	99	100
82	96	85	90

Thresholding / Grayscale Threshold Transform / Binarization (Cont.)

Logical Code: Or im2bw() function

```
S=95
y=x;
[m n]=size(x);
for i=1:m
    for j=1:n
        if x(i,j)>= s
            y(i,j)=255;
        else
            y(i,j)=0;
        end
    end
end
figure, imshow(x);  figure, imshow(y);
```

Marwadi University

Gray-level Slicing

- This technique is used to highlight a specific range of gray levels in a given image.
- Similar to thresholding Other levels can be suppressed or maintained
- Useful for highlighting features in an image
- It can be implemented in several ways, but the

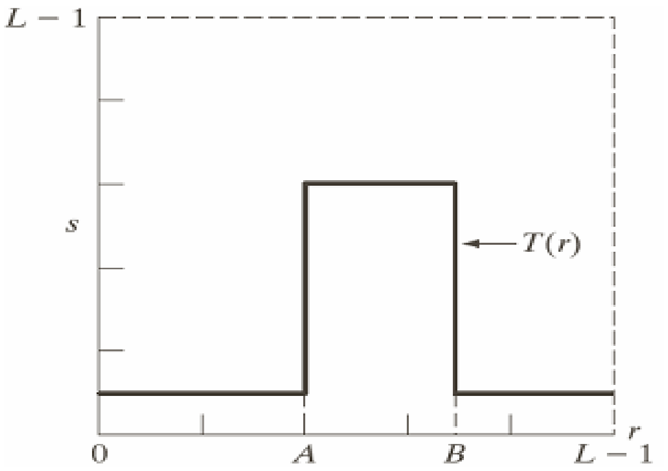
Gray-level Slicing (Cont.)

- Two basic Approach are:
  - One approach is to display a high value for all gray levels in the range of interest and a low value for all other gray levels.
  - The second approach, based on the transformation brightens the desired range of gray

Gray-level Slicing (Cont.)

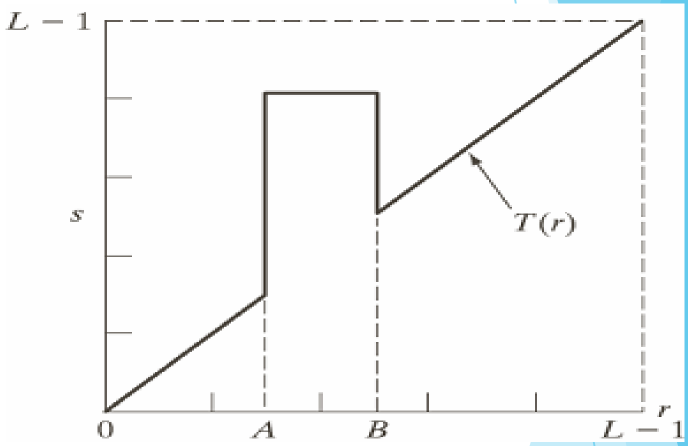
Highlighting a specific range of intensities in an image.

Approach 1



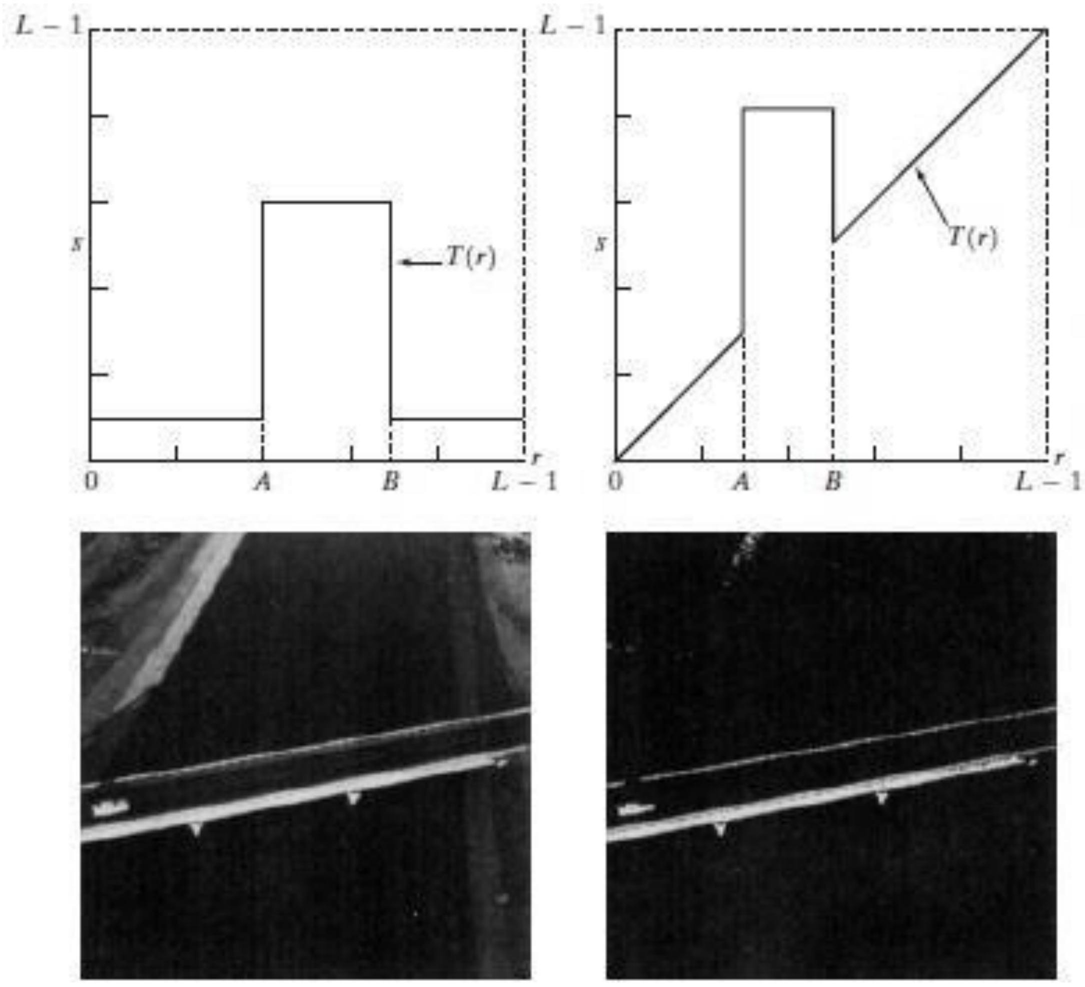
display in one value(e.g white) all the values in the range of interest , and in another (e.g black) all other intensities

Approach 2



Brightens or darkens the desired range of intensities but leaves all other intensity levels in the image unchanged

Gray-level Slicing (Cont.)



a b
c d

FIGURE 3.11 (a) This transformation highlights range $[A, B]$ of gray levels and reduces all others to a constant level. (b) This transformation highlights range $[A, B]$ but preserves all other levels. (c) An image. (d) Result of using the transformation in (a).

Gray-level Slicing (Cont.)

- Example 1 : Apply intensity level slicing in below image ,
  - Approach 1: then If the pixel intensity in the old image is between (100 - 200) convert it in the new image into 255 (white). Otherwise convert it to 0 (black).
  - Approach 2: then If the pixel intensity in the old image is between (100 - 200) convert it in the new image into 255 (white). Otherwise it leaves it the same.

110	120	90	130
91	94	98	200
90	91	99	100
82	96	85	90

Gray-level Slicing (Cont.)

- Example for Approach 1: apply intensity level slicing in Matlab to read cameraman image , then If the pixel intensity in the old image is between (100 - 200) convert it in the new image into 255 (white). Otherwise convert it to 0 (black).

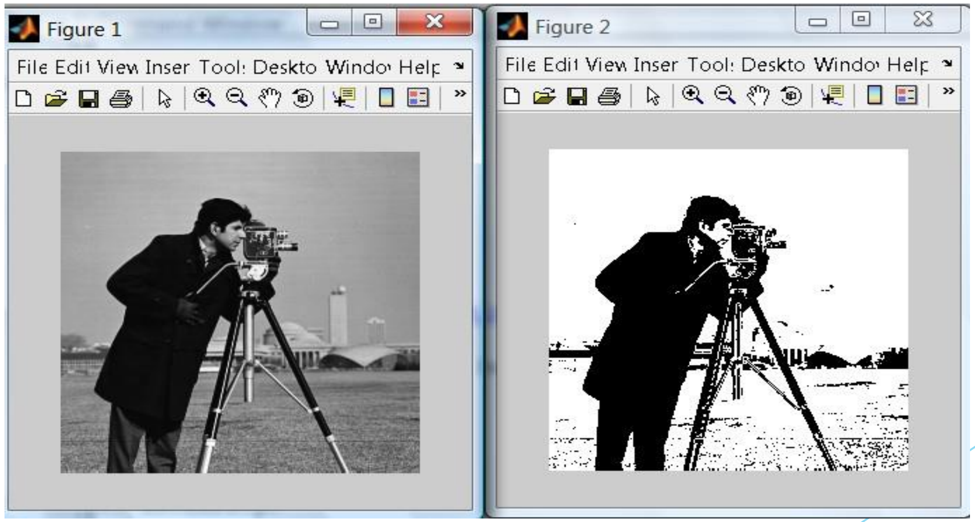


Figure 1
File Edit View Insert Tools Desktop Window Help
Figure 2
File Edit View Insert Tools Desktop Window Help

Gray-level Slicing (Cont.)

- Example for Approach 2 : apply intensity level slicing in Matlab to read cameraman image , then If the pixel intensity in the old image is between (100 - 200) convert it in the new image into 255 (white). Otherwise it leaves it the same.

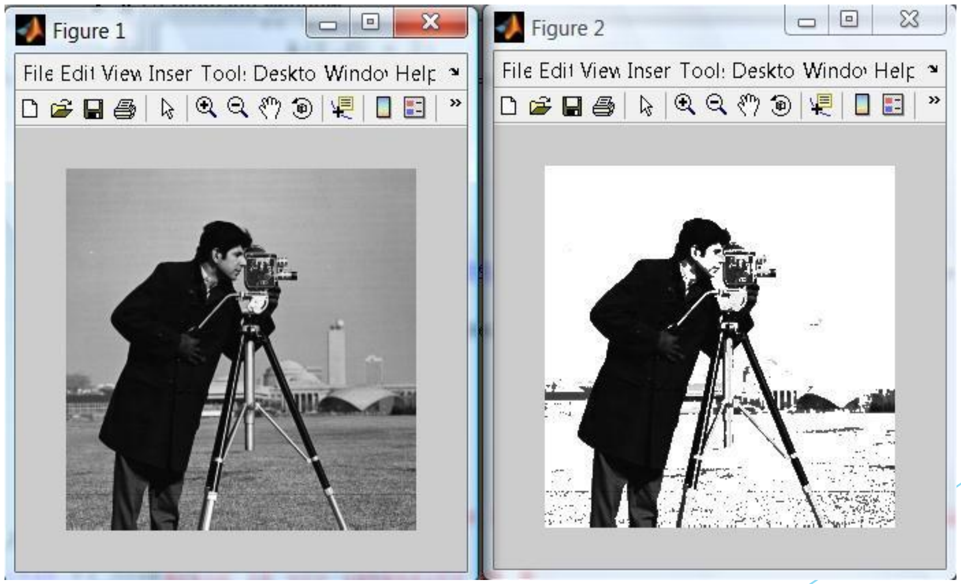


Figure 1
File Edit View Insert Tools Desktop Window Help
Figure 2
File Edit View Insert Tools Desktop Window Help

Bit-Plane Slicing

- Pixels are digital numbers, each one composed of bits. Instead of highlighting gray-level range, we could highlight the contribution made by each bit.
- This method is useful and used in image compression.
- Most significant bits contain the majority of visually significant data

Bit-Plane Slicing (Cont.)

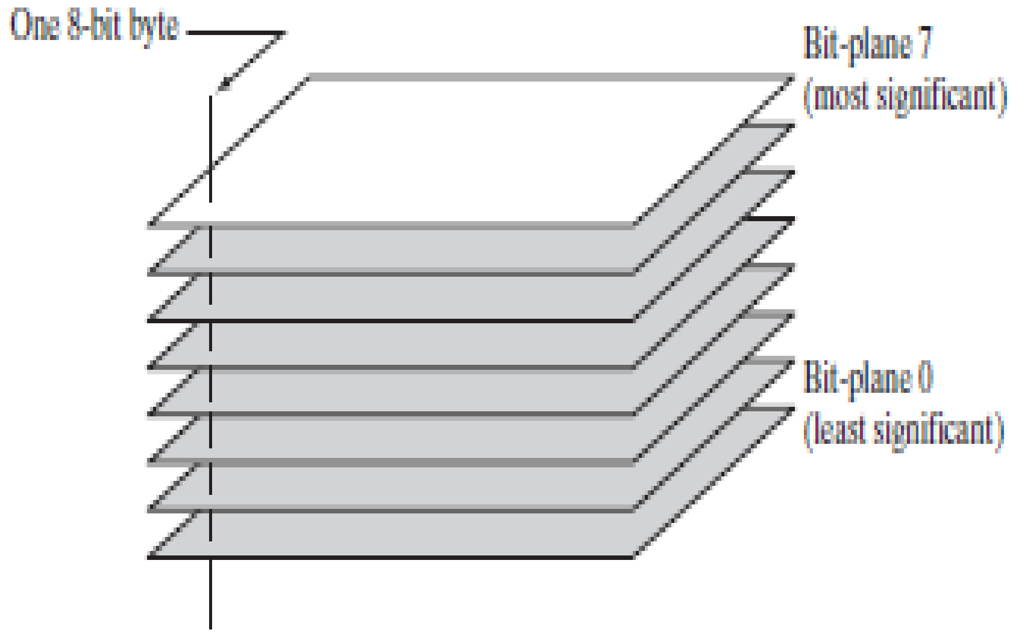


FIGURE 3.12 Bit-plane representation of an 8-bit image.

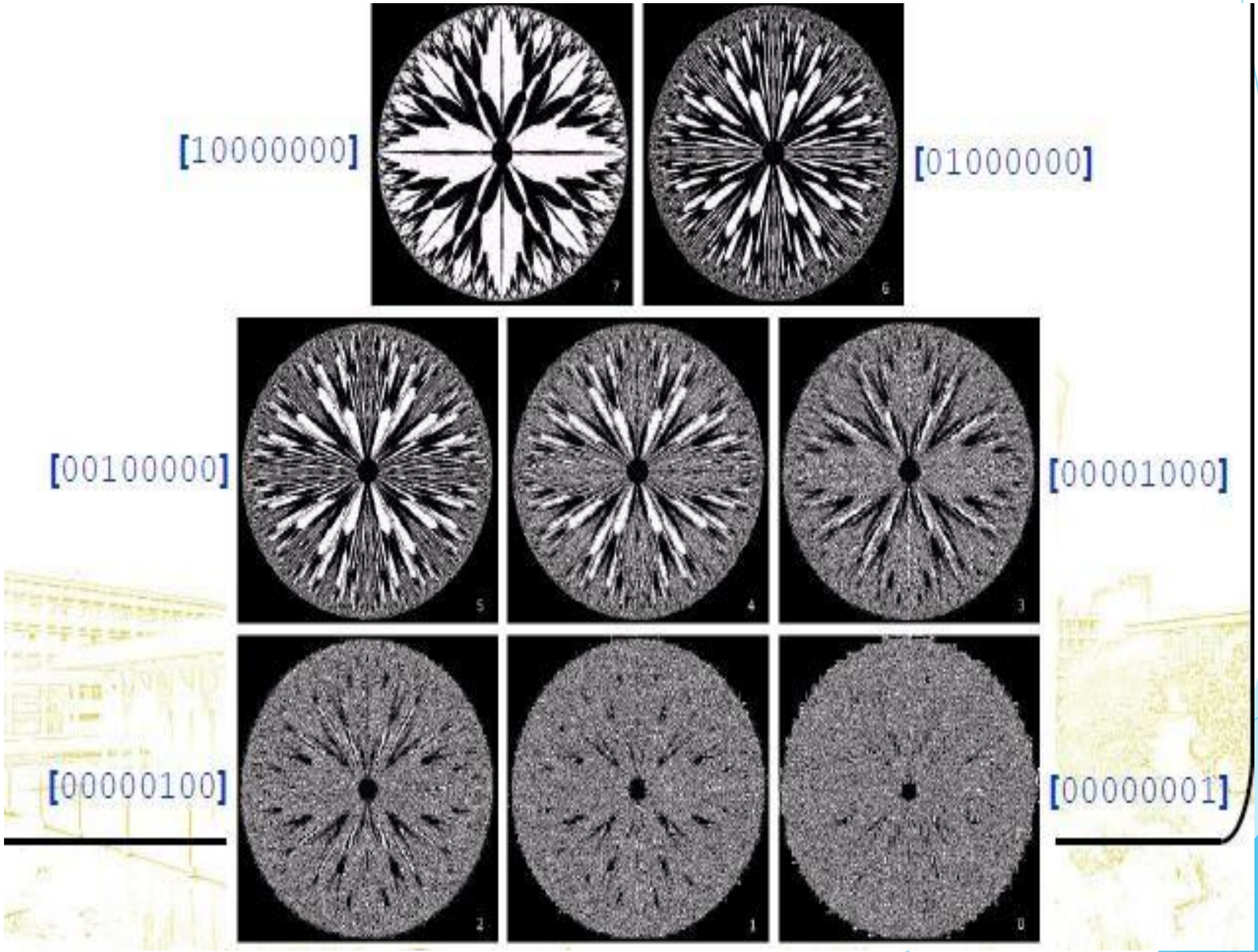
Remember that pixels are digital numbers composed of bits.

8-bit Image composed of 8 1-bit planes

Bit-Plane Slicing (Cont.)

- Often by isolating particular bits of the pixel values in an image we can highlight interesting aspects of that image
  - Higher-order bits usually contain most of the significant visual information
  - Lower-order bits contain subtle details

Bit-Plane Slicing (Cont.)



Bit-Plane Slicing (Cont.)

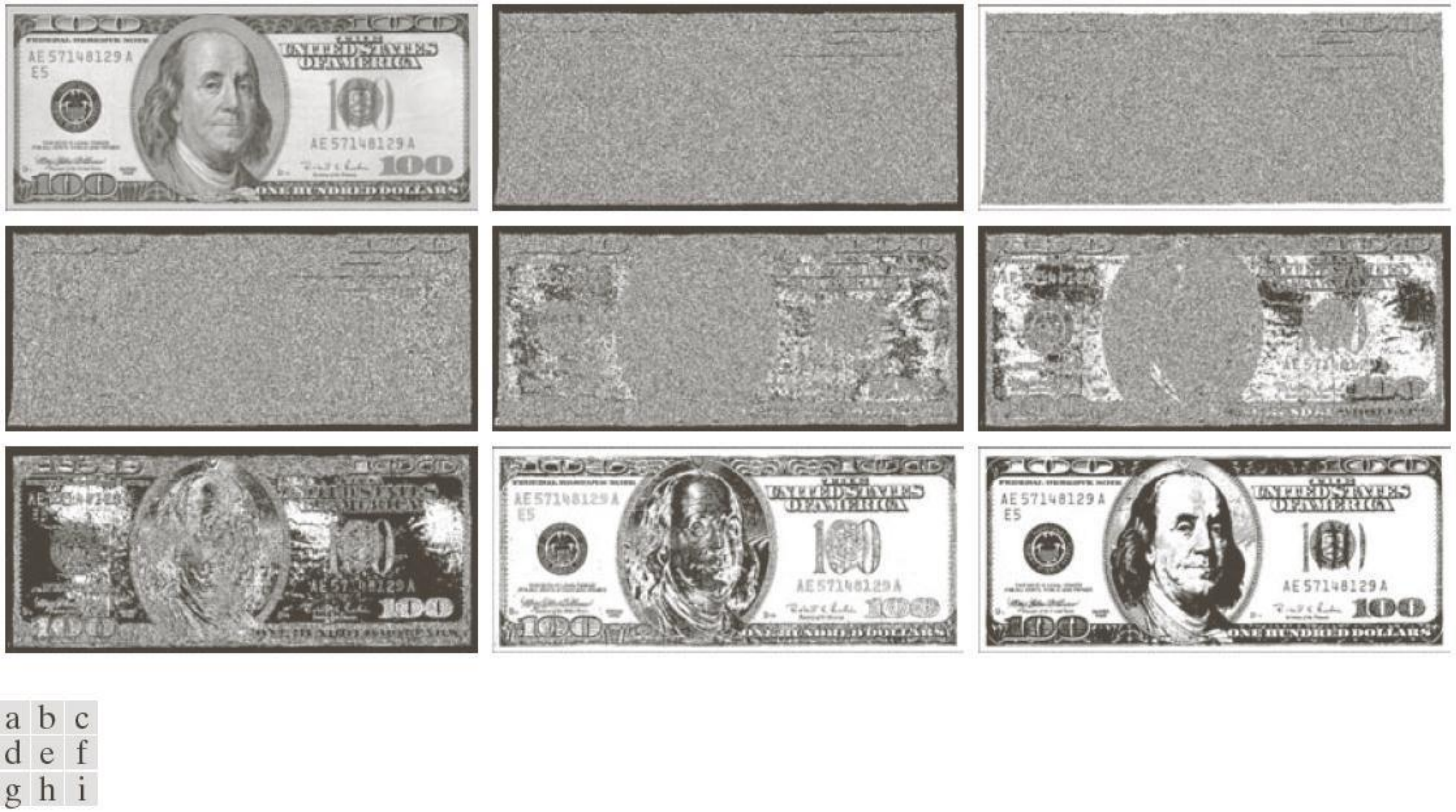
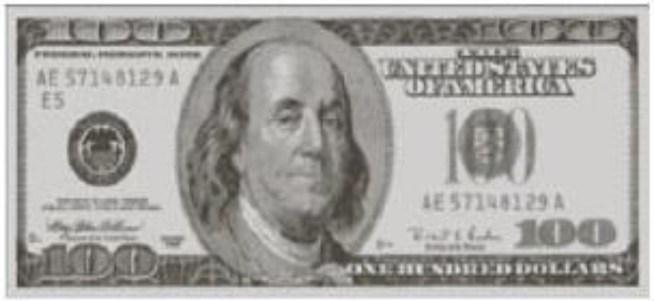
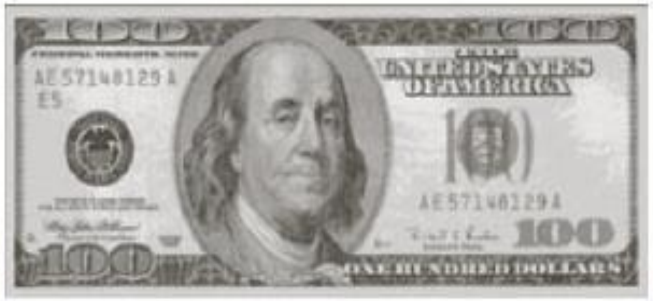


FIGURE 3.14 (a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

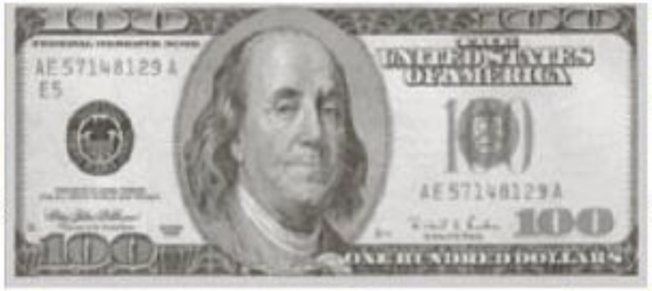
Bit-Plane Slicing (Cont.)



Reconstructed image using only bit planes 8 and 7



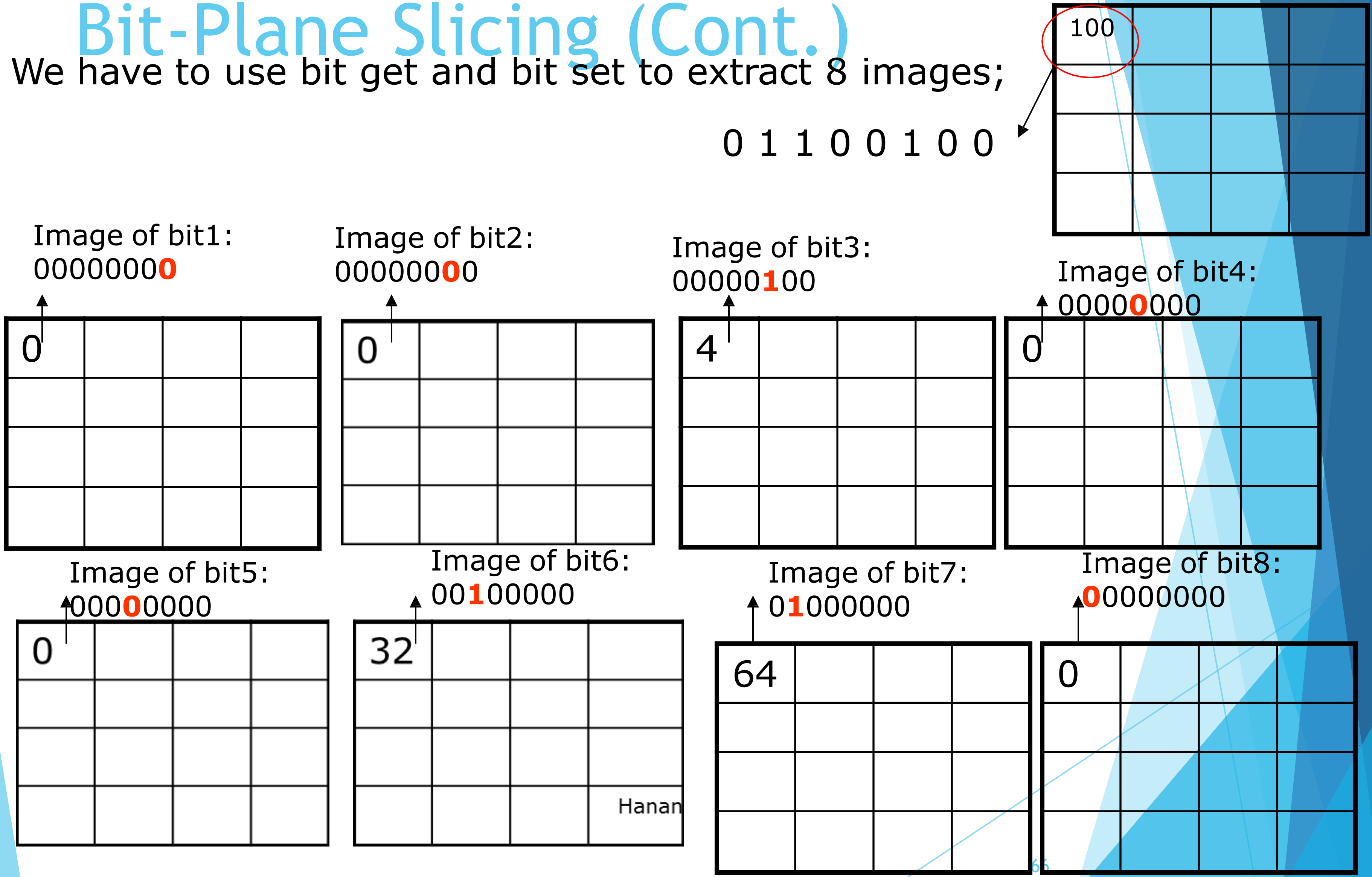
Reconstructed image using only bit planes 8, 7 and 6



Reconstructed image using only bit planes 7, 6 and 5

Bit-Plane Slicing (Cont.)

We have to use bit get and bit set to extract 8 images;



Bit-Plane Slicing (Cont.)

- Function to implement Bit-Plan Slicing:

```
b=bitget(x(i,j),6);
y(i,j)=bitset(y(i,j),6,b);
```

Histogram

- A histogram is a graph.
- A graph that shows frequency of anything.
- Usually histogram have bars that represent frequency of occurring of data in the whole data set

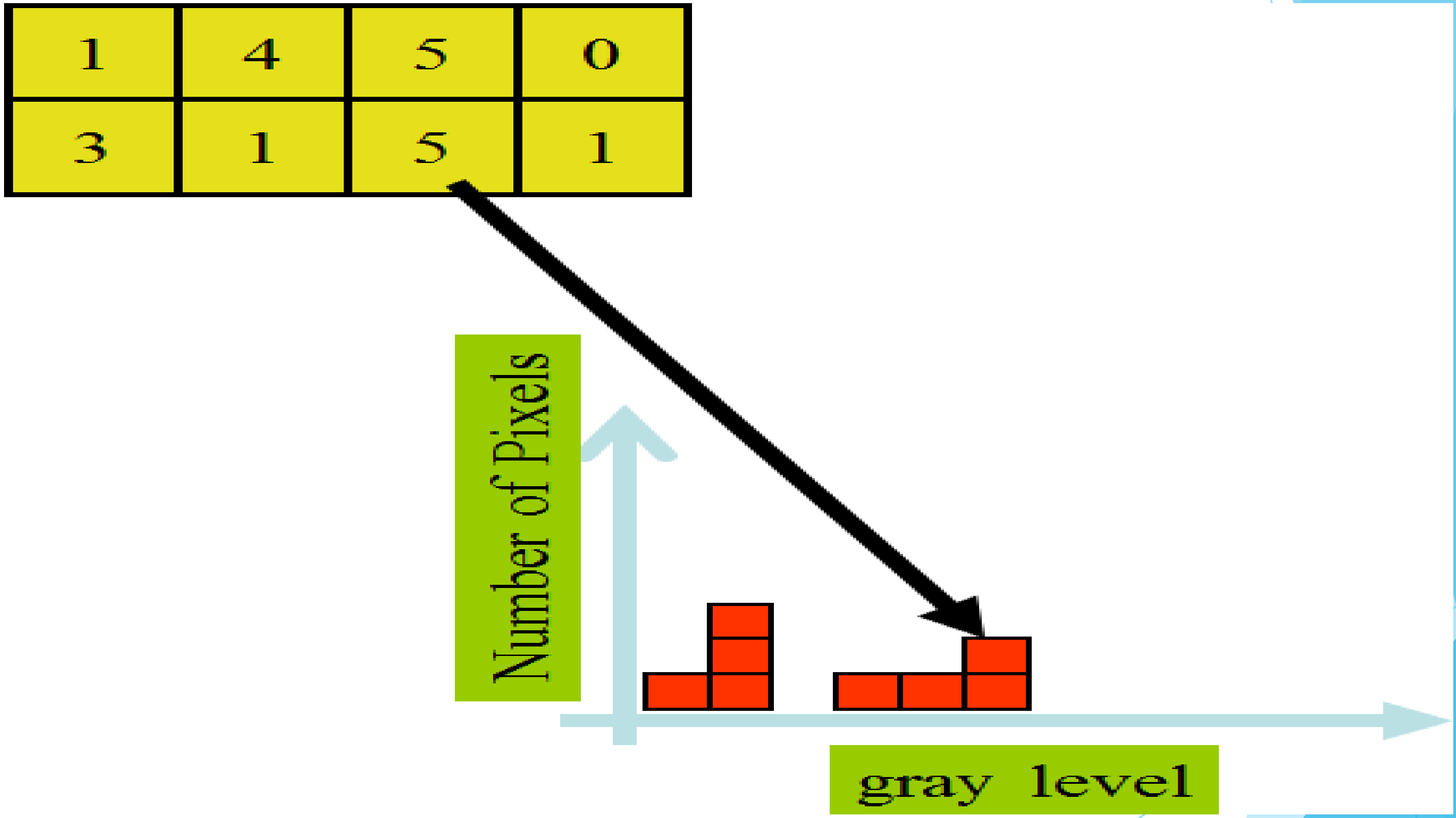
Histogram (Cont.)

- In Statistics, Histogram is a graphical representation showing a visual impression of the distribution of data.
- An Image Histogram is a type of histogram that acts as a graphical representation of the lightness/color distribution in a digital image.
- It plots the number of pixels for each value.

Histogram (Cont.)

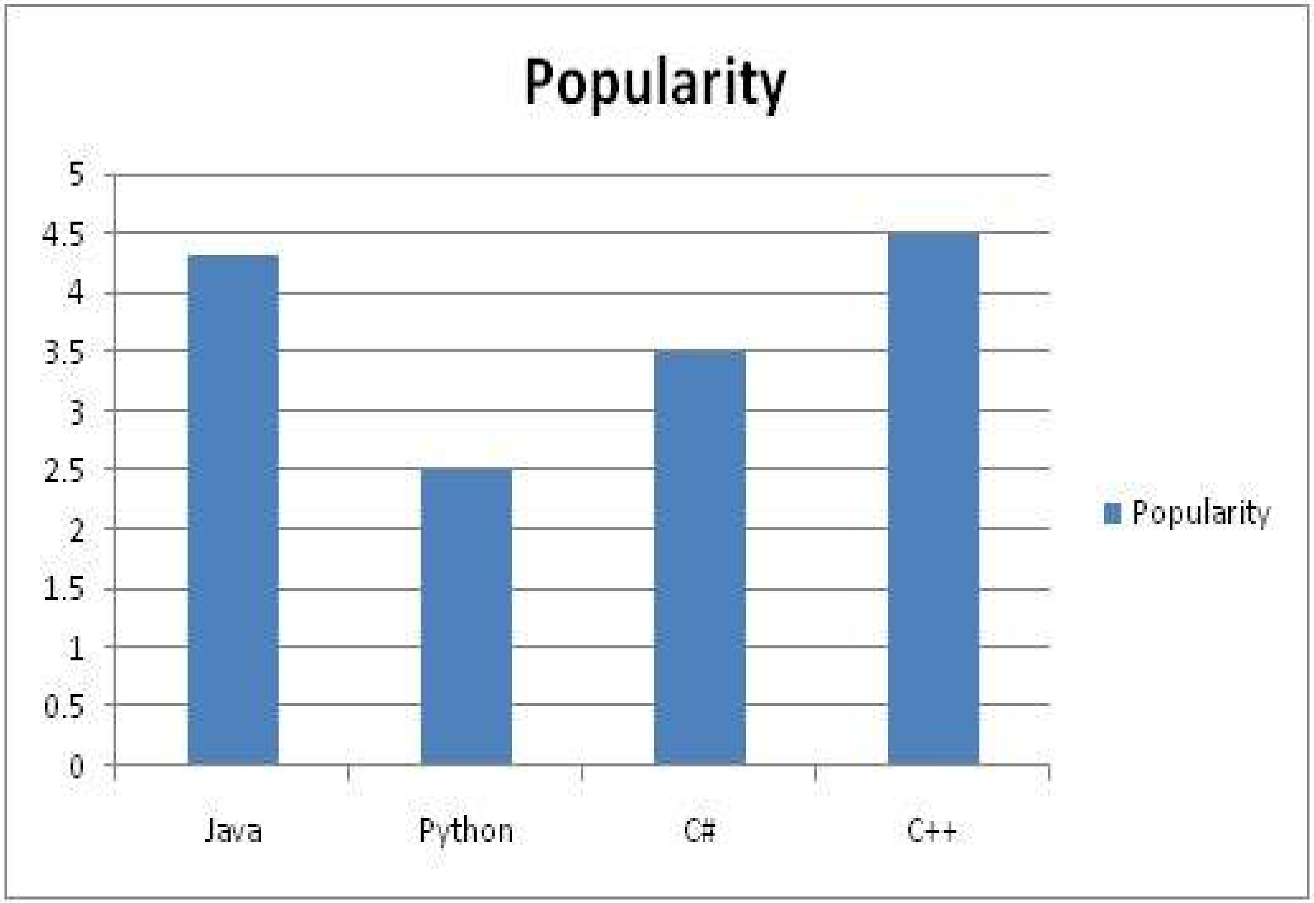
- A Histogram has two axis the x axis and the y axis.
- The x axis contains event whose frequency you have to count.
- The y axis contains frequency.
- The different heights of bar shows different frequency of occurrence of data.

Histogram (Cont.)

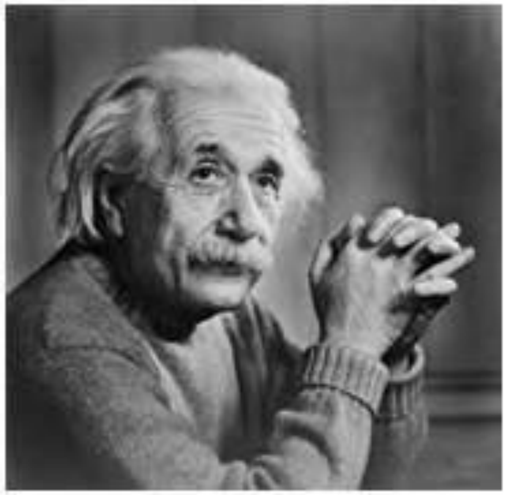


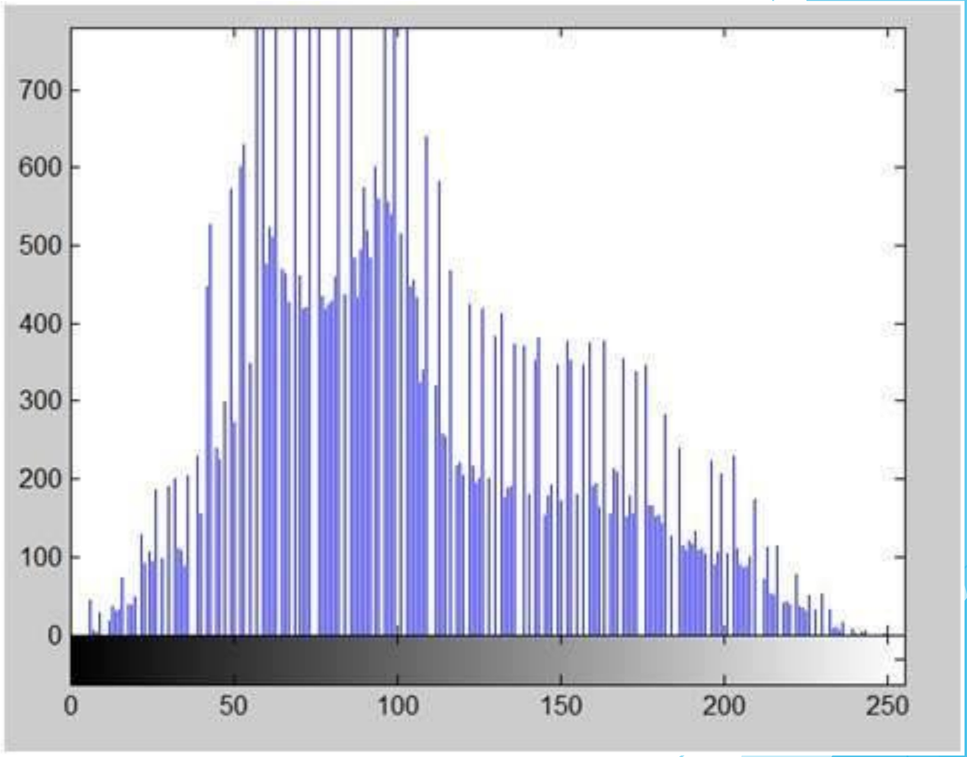
Histogram (Cont.)

- Usually a histogram looks like this.



Histogram (Cont.)





Histogram (Cont.)

- The range of x axis starts from 0 and end at 255 with a gap of 50. Whereas on the y axis, is the count of these intensities.
- Most of the bars that have high frequency lies in the first half portion which is the darker portion.
- That means that the image we have got is darker. And this can be proved from the image too.

Histogram (Cont.)

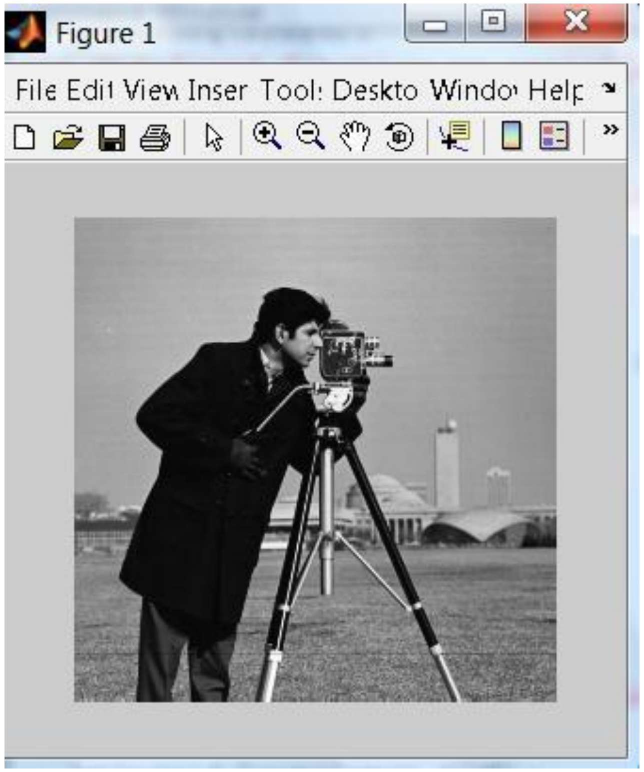
- The histogram of a digital image with gray levels in the range [0, L-1] is a discrete function:

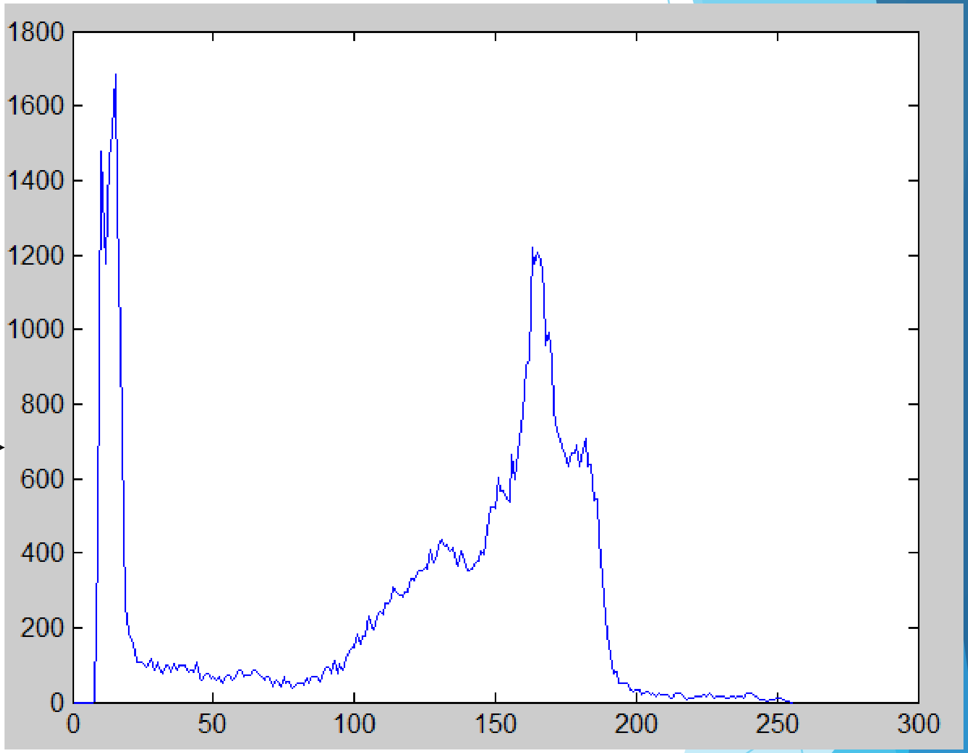
$$h(r_k) = n_k$$

Where:

- r_k : kth gray level
- n_k : No of pixels with having gray level r_k

Histogram (Cont.)

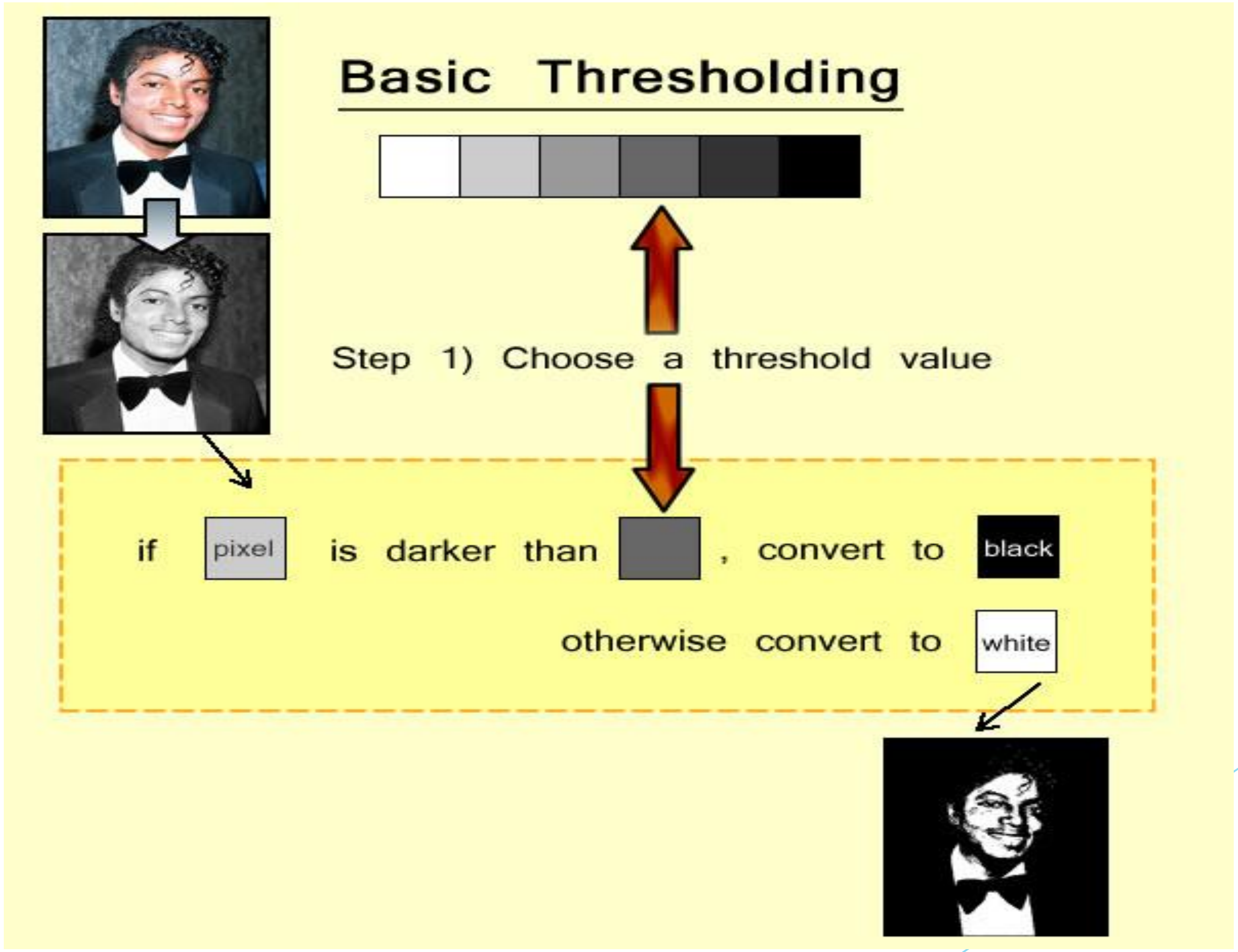




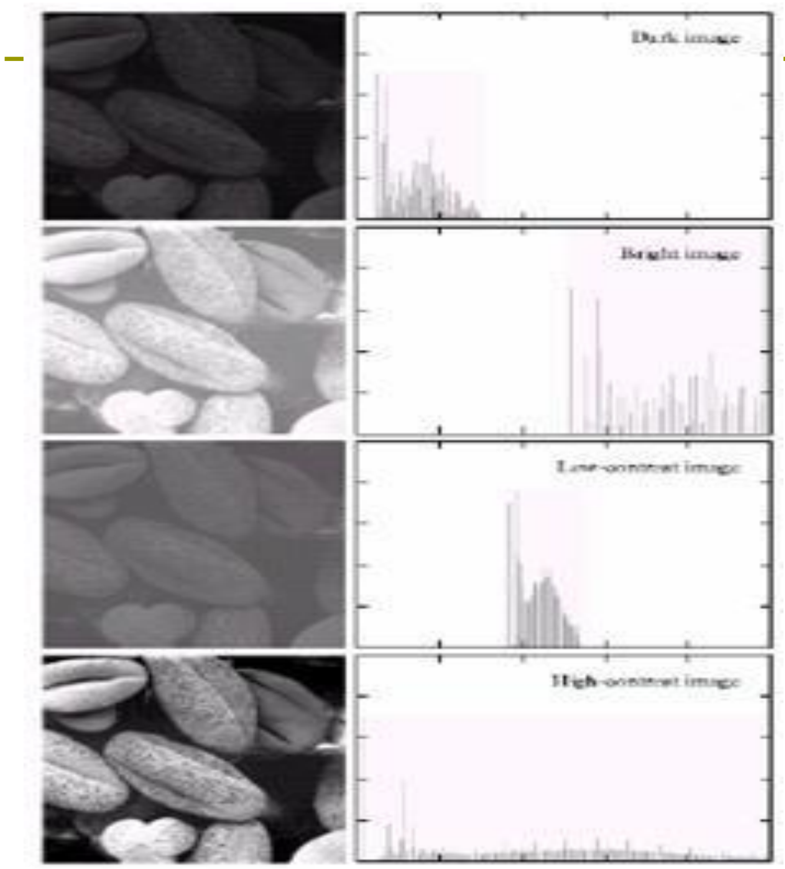
Importance of Histogram

- Histograms are the basis for numerous spatial domain processing techniques
- Histogram manipulation can be used effectively for image enhancement
- Histograms can be used to provide useful image statistics
- Information derived from histograms are quite useful in other image processing applications, such as image compression

Importance of Histogram (Cont.)


Basic Thresholding
Step 1) Choose a threshold value
if pixel is darker than , convert to black
otherwise convert to white

Importance of Histogram



→ low contrast image

→ low contrast image

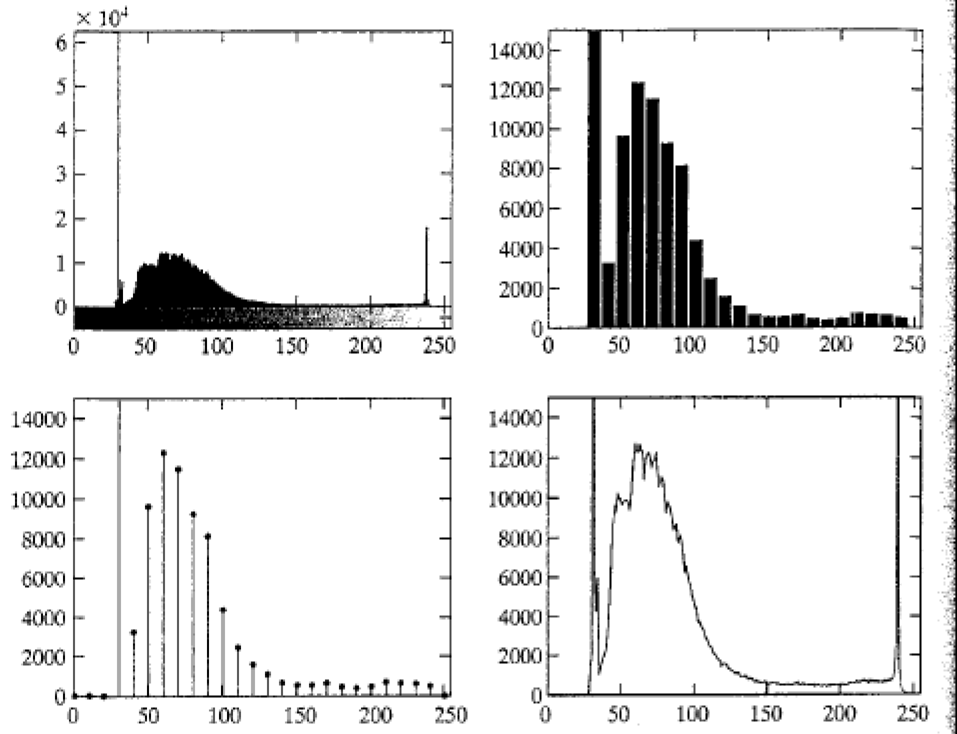
→ low contrast image

→ high-contrast image

An image whose pixels tend to occupy the entire range of possible gray levels and, in addition, tend to be distributed uniformly, will have an appearance of high contrast and will exhibit a large variety of gray tones.

Other ways to display Histograms

- A stem graph
- A bar graph
- A Plot graph



a b
c d

FIGURE 3.7 Various ways to plot an image histogram. (a) `imhist`, (b) `bar`, (c) `stem`, (d) `plot`.

Histogram Processing

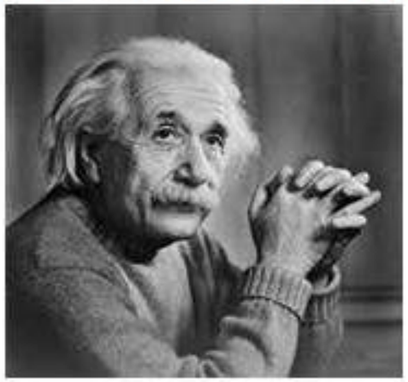
- Histogram Sliding / Histogram Stretching
- Histogram Normalization (PMF)
- Histogram Equalization (CDF)
- Histogram Matching / Histogram Specification

Histogram Sliding / Histogram Stretching

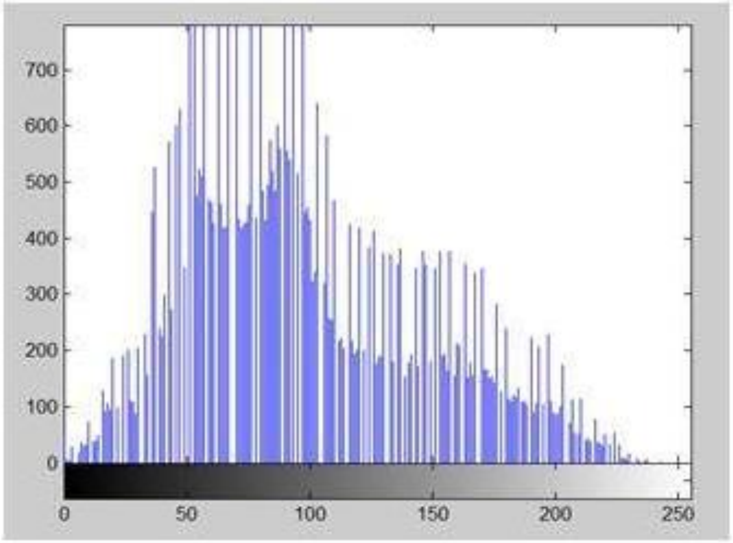
- In histogram sliding, we just simply shift a complete histogram rightwards or leftwards.
- Due to shifting or sliding of histogram towards right or left, a clear change can be seen in the image

Histogram Sliding / Histogram Stretching (Cont.)

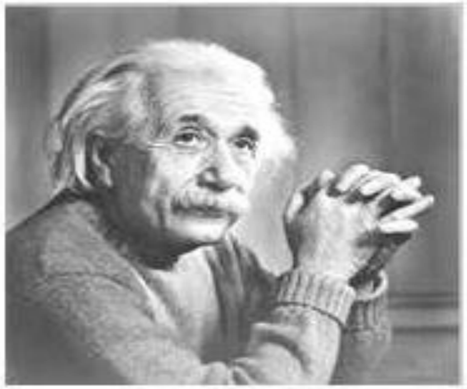
Increasing brightness using histogram sliding



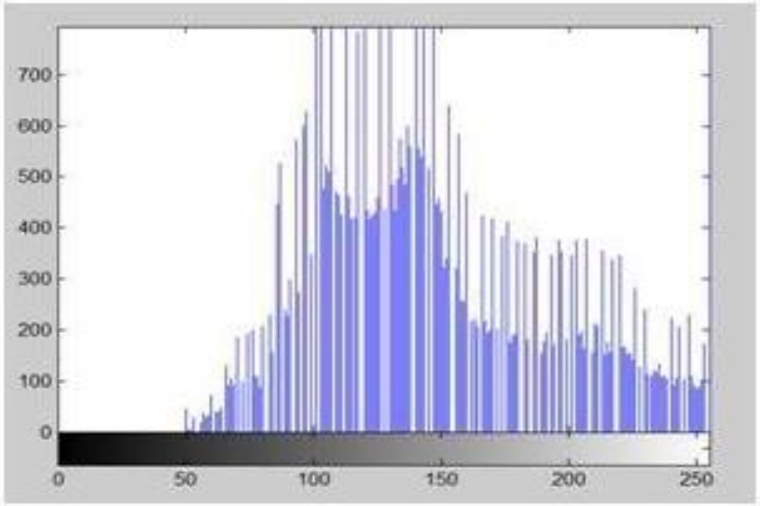
Histogram of this image has been shown below.



The image has been shown below.



And its histogram has been shown below.



Histogram Sliding / Histogram Stretching (Cont.)

- The formula for stretching the histogram of the image to increase the contrast is

$$g(x,y) = \frac{f(x,y)-fmin}{fmax-fmin} * 2^{bpp}$$

Histogram Sliding / Histogram Stretching (Cont.)

- In our case the image is 8bpp, so levels of gray are 256.
- The minimum value is 0 and the maximum value is 225. So the formula in our case is

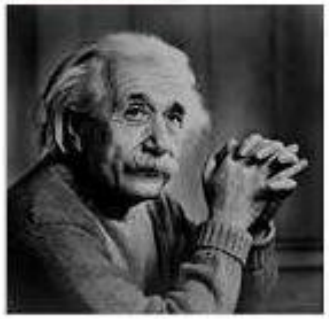
$$g(x,y) = \frac{f(x,y) - 0}{225 - 0} * 255$$

- where f(x,y) denotes the value of each pixel intensity. For each f(x,y) in an image , we will calculate this formula.
- After doing this, we will be able to enhance our contrast.

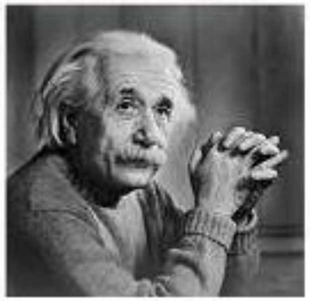
Histogram Sliding / Histogram Stretching (Cont.)

- Decreasing brightness using histogram sliding

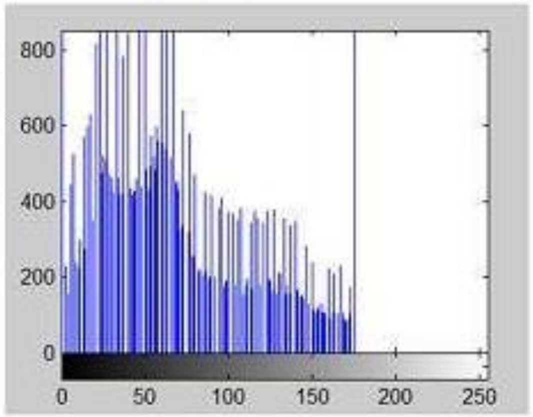
New image.



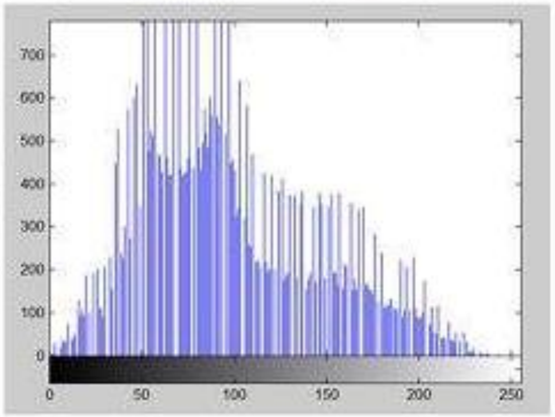
Original image.



New Histogram.



Original Histogram.

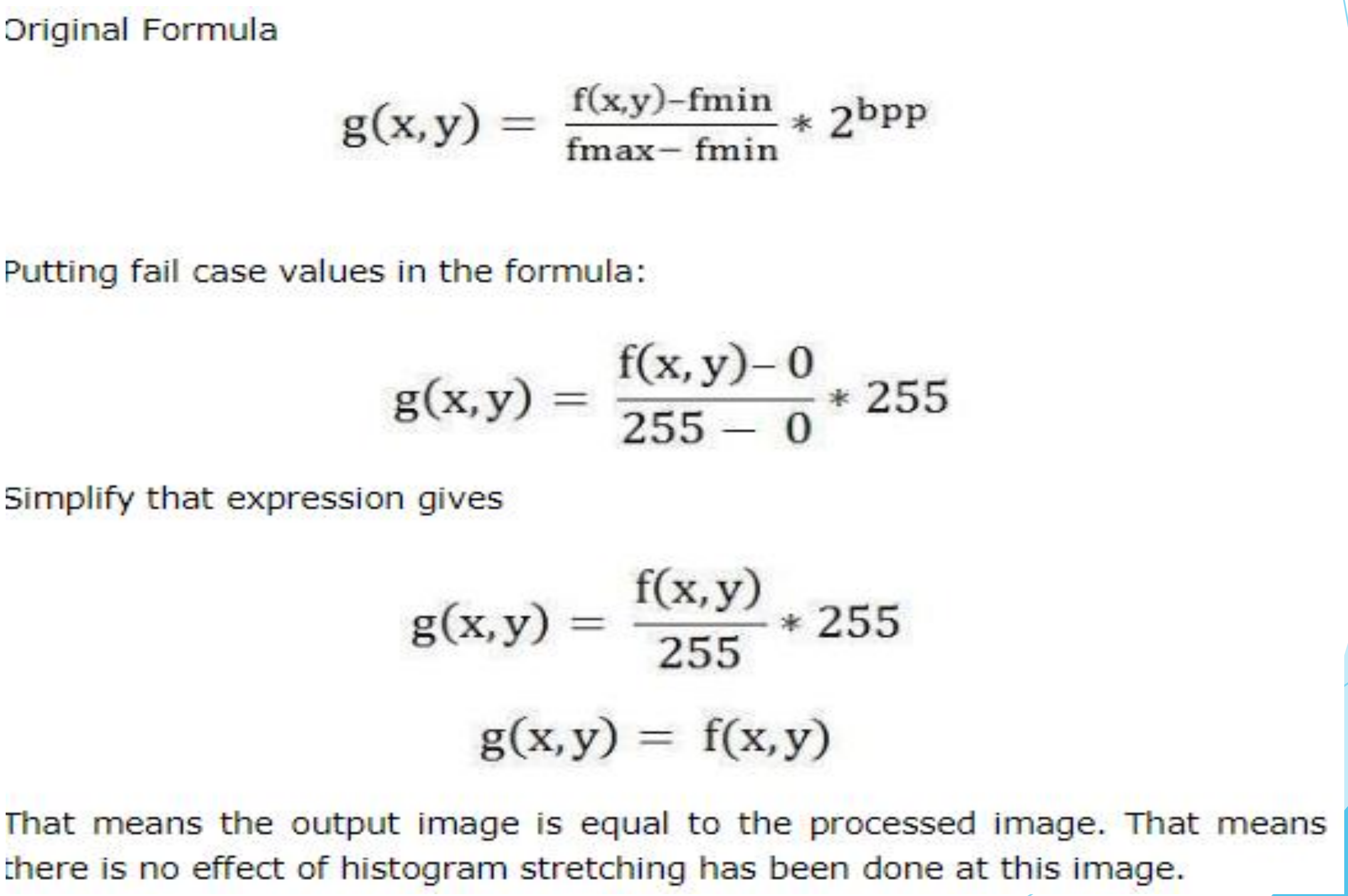


Failing of histogram stretching

- Those cases include images with when there is pixel intensity 0 and 255 are present in the image.

- Because when pixel intensities 0 and 255 are present in an image, then in that case they become the minimum and maximum pixel intensity which ruins the formula like this.

Failing of Histogram Stretching (Cont.)



Original Formula

$$g(x,y) = \frac{f(x,y) - fmin}{fmax - fmin} * 2^{bpp}$$

Putting fail case values in the formula:

$$g(x,y) = \frac{f(x,y) - 0}{255 - 0} * 255$$

Simplify that expression gives

$$g(x,y) = \frac{f(x,y)}{255} * 255$$

$$g(x,y) = f(x,y)$$

That means the output image is equal to the processed image. That means there is no effect of histogram stretching has been done at this image.

Histogram Normalization

- It is common practice to normalize a histogram by dividing each of its values by the total number of pixels in the image, denoted by n. Thus, a normalized histogram is given by

$$p(r_k) = n_k \,/\, n,$$

$$\text{for } k = 0, 1, \ldots, L-1.$$

- Thus, $p(r_k)$ gives an estimate of the probability of occurrence of gray level r_k.
- Note that the sum of all components of a normalized histogram is equal to 1.

Histogram Normalization in MATLAB

- We obtain the normalized histogram simply by using the expression.

  p = imhist (f, b) / numel(f)

- numel (f): a MATLAB function that gives the number of elements in array f (i.e. the number of pixels in an image).

PMF - Probability Mass Function

- PMF stands for probability mass function.
- As it name suggest, it gives the probability of each number in the data set or you can say that it basically gives the count or frequency of each element.

How PMF is calculated

Example: Calculate PMF for Below Image.

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

How PMF is calculated (Cont.)

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

Gray level / Intensity Level	No of pixels / Frequency n_K	PMF / $P_K = n_K/K$
0	2	2/25=0.08
1	4	4/25=0.16
2	3	3/25=0.12
3	3	3/25=0.12
4	2	2/25=0.08
5	4	4/25=0.16
6	3	3/25=0.12
7	4	4/25=0.16

Total No of Pixel K = 25

How PMF is calculated (Cont.)

- Calculate PMF for the Following image

Gray level / Intensity Level	0	1	2	3	4	5	6	7
No of pixels / Frequency n_K	9	8	11	4	10	15	4	3

How PMF is calculated (Cont.)

Gray level / Intensity Level	No of pixels / Frequency n_K	PMF / $P_K = n_K/K$
0	9	9/64 = 0.141
1	8	8/64 = 0.125
2	11	11/64 = 0.172
3	4	4/64 = 0.0625
4	10	10/64 = 0.156
5	15	15/64 = 0.234
6	4	4/64 = 0.0625
7	3	3/64 = 0.047

Total No of Pixel K = 64

How PMF is calculated (Cont.)

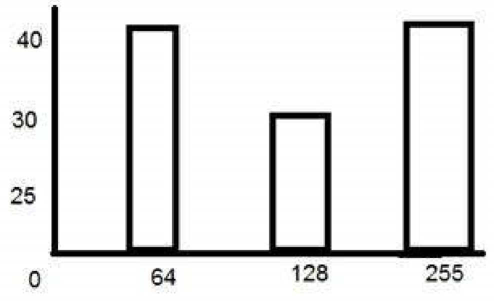
- The above histogram shows frequency of gray level values for an 8 bits per pixel image.
- Now if we have to calculate its PMF, we will simple look at the count of each bar from vertical axis and then divide it by total count.

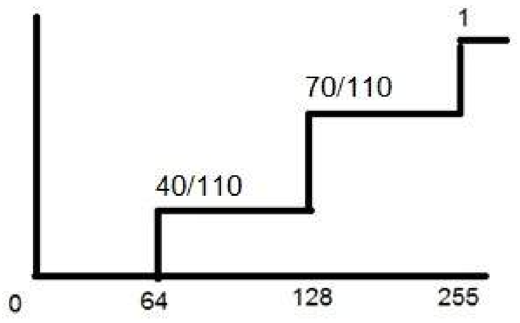
CDF (Cumulative Distributive Function)

- Another important thing to note in the above histogram is that it is not monotonically increasing. So in order to increase it monotonically, we will calculate its CDF
- CDF stands for cumulative distributive function. It is a function that calculates the cumulative sum of all the values that are calculated by PMF. It basically sums the previous one.

CDF (Cont.)

- Consider the histogram which shows PMF. Since this histogram is not increasing monotonically, so will make it grow monotonically.
- For CDP, We will simply keep the first value as it is, and then in the 2nd value , we will add the first one and so on.





How CDF is calculated

- Example: Calculate CDF for the Following image

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

How CDF is calculated (Cont.)

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

Total No of Pixel K = 25

Gray level / Intensity Level	No of pixels / Frequency n_K	PMF / $P_K = n_K/K$	CDF / S_K
0	2	2/25=0.08	0.08
1	4	4/25=0.16	0.24
2	3	3/25=0.12	0.36
3	3	3/25=0.12	0.48
4	2	2/25=0.08	0.56
5	4	4/25=0.16	0.72
6	3	3/25=0.12	0.84
7	4	4/25=0.16	1

How CDF is calculated (Cont.)

Example: Calculate CDF for the Following image

Gray level / Intensity Level	0	1	2	3	4	5	6	7
No of pixels / Frequency n_K	9	8	11	4	10	15	4	3

How CDF is calculated (Cont.)

Gray level / Intensity Level	No of pixels / Frequency n_K	PMF / $P_K = n_K/K$	CDF / S_K
0	9	9/64 = 0.141	0.141
1	8	8/64 = 0.125	0.266
2	11	11/64 = 0.172	0.438
3	4	4/64 = 0.0625	0.5005
4	10	10/64 = 0.156	0.6565
5	15	15/64 = 0.234	0.8905
6	4	4/64 = 0.0625	0.953
7	3	3/64 = 0.047	1

Total No of Pixel K = 64

How CDF is calculated (Cont.)

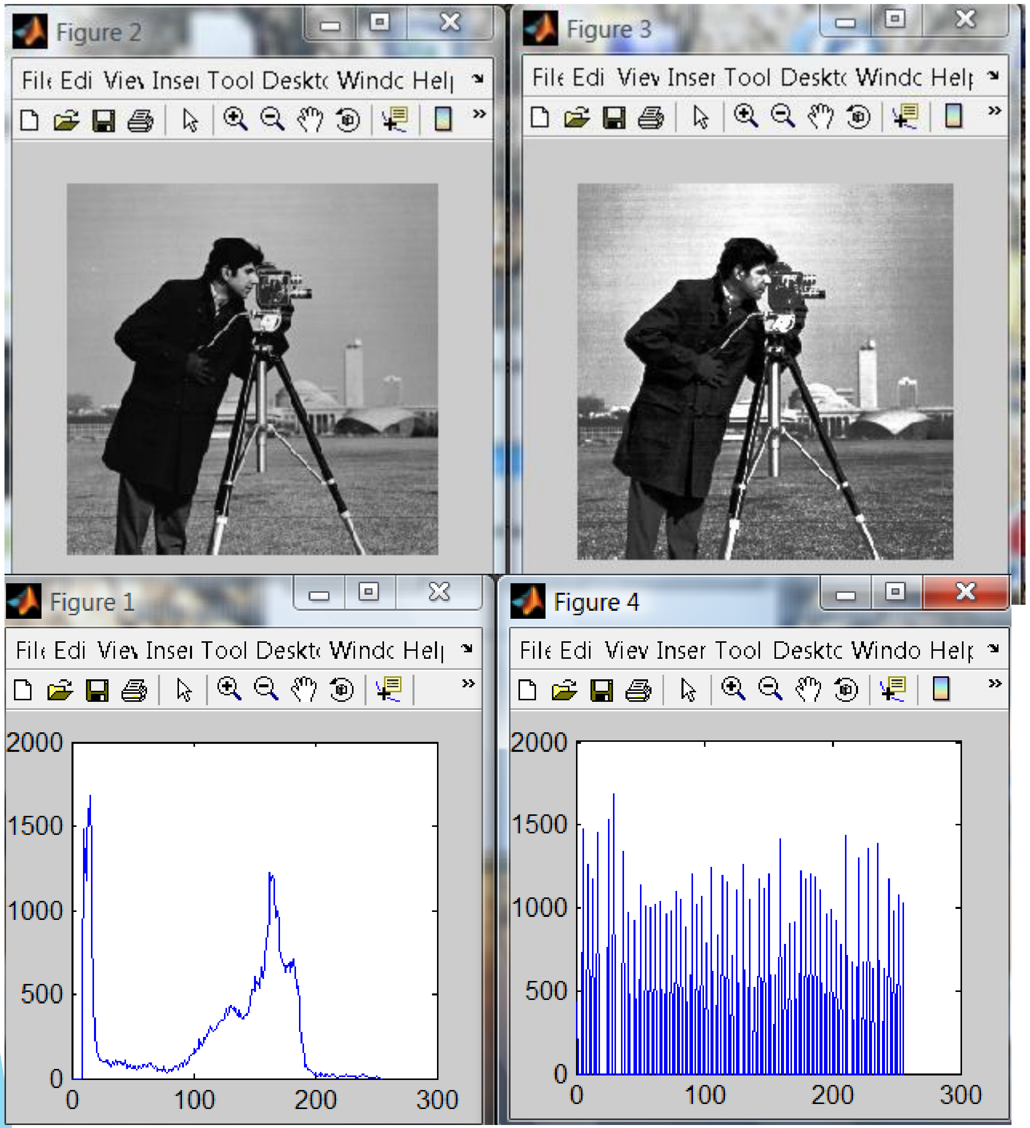
- Example: Calculate CDF for the Following image

100	110	90	95
98	140	145	135
89	90	88	85
102	105	99	115

Histogram Equalization

- Histogram Equalization is the process of adjusting intensity values of pixels.
- The process which increases the dynamic range of the gray level in a law contrast image to cover full range of gray levels.

Histogram Equalization (Cont.)



Notice that histogram equalization does not always produce a good result

Histogram Equalization (Cont.)

- Steps to Perform Histogram Equalization
  - **Calculate PMF** (PMF helps us calculating the probability of each pixel value in an image)
  - **Calculate CDF** (S_k) (CDF gives us the cumulative sum of PMF values)
  - **Calculate S_k*(L-1)** (CDF is multiplied by levels- 1)
  - **Find out Histogram Equalization Level** (find the new pixel intensities)
  - **Mapping** (mapping of Histogram Equalization Level to Frequency)

Histogram Equalization (Cont.)

- Example: Perform Histogram Equalization for the following Image.

Gray level / Intensity Level	0	1	2	3	4	5	6	7
No of pixels / Frequency n_K	9	8	11	4	10	15	4	3

Marwadi University

Histogram Equalization (Cont.)

Total No of Pixel K = 64 , L=8

Gray level / Intensity Level	No of pixels / Frequency n_K	PMF / $P_K = n_K/K$	CDF / S_K	S_K* (L-1) = S_K* (8-1) = S_K* 7	Histogram Equalization Level
0	9	9/64 = 0.141	0.141	0.987	1
1	8	8/64 = 0.125	0.266	1.862	2
2	11	11/64 = 0.172	0.438	3.066	3
3	4	4/64 = 0.0625	0.5005	3.5035	4
4	10	10/64 = 0.156	0.6565	4.5955	5
5	15	15/64 = 0.234	0.8905	6.2336	6
6	4	4/64 = 0.0625	0.953	6.671	7
7	3	3/64 = 0.047	1	7	7

Histogram Equalization (Cont.)

Histogram Equalization Level	No of pixels / Frequency n_K
1	9
2	8
3	11
4	4
5	10
6	15
7	4
7	3

New Gray level / Intensity Level	No of pixels / Frequency n_K
1	9
2	8
3	11
4	4
5	10
6	15
7	7

Histogram Equalization (Cont.)

- Example: Perform Histogram Equalization for the following Image.

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

Histogram Equalization (Cont.)

Total No of Pixel K = 25 , L=8

Gray level / Intensity Level	No of pixels / Frequency n_K	PMF / $P_K = n_K/K$	CDF / S_K	S_K* (L-1)	Histogram Equalization Level
0	2	2/25=0.08	0.08	0.56	1
1	4	4/25=0.16	0.24	1.68	2
2	3	3/25=0.12	0.36	2.52	3
3	3	3/25=0.12	0.48	3.36	3
4	2	2/25=0.08	0.56	3.92	4
5	4	4/25=0.16	0.72	5.04	5
6	3	3/25=0.12	0.84	5.88	6
7	4	4/25=0.16	1	7	7

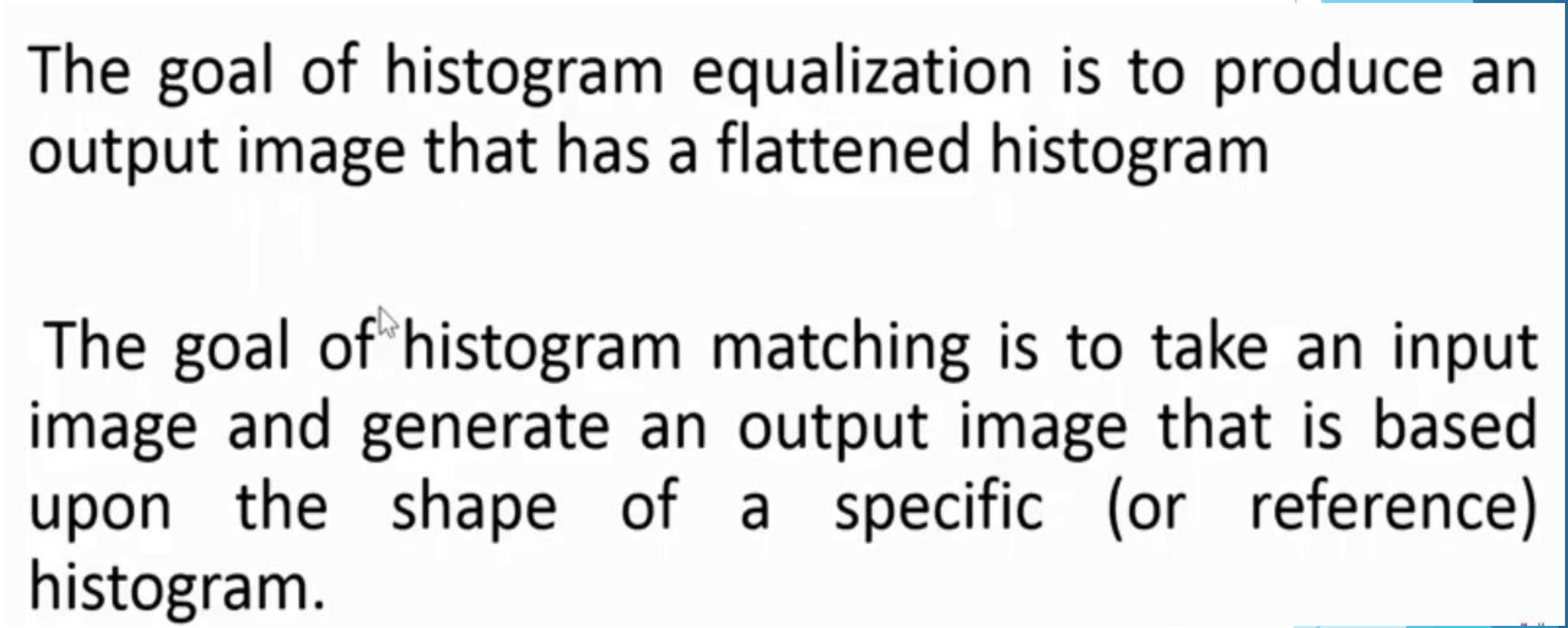
Histogram Equalization (Cont.)

Total No of Pixel K = 25 , L=8

Histogram Equalization Level	No of pixels / Frequency n_K
0	0
2	8
4	8
5	2
5	0
7	7
7	0
7	0

New Gray level / Intensity Level / Histogram Equalization Level	No of pixels / Frequency n_K
0	0
2	8
4	8
5	2
7	7

Histogram Equalization VS Histogram Matching

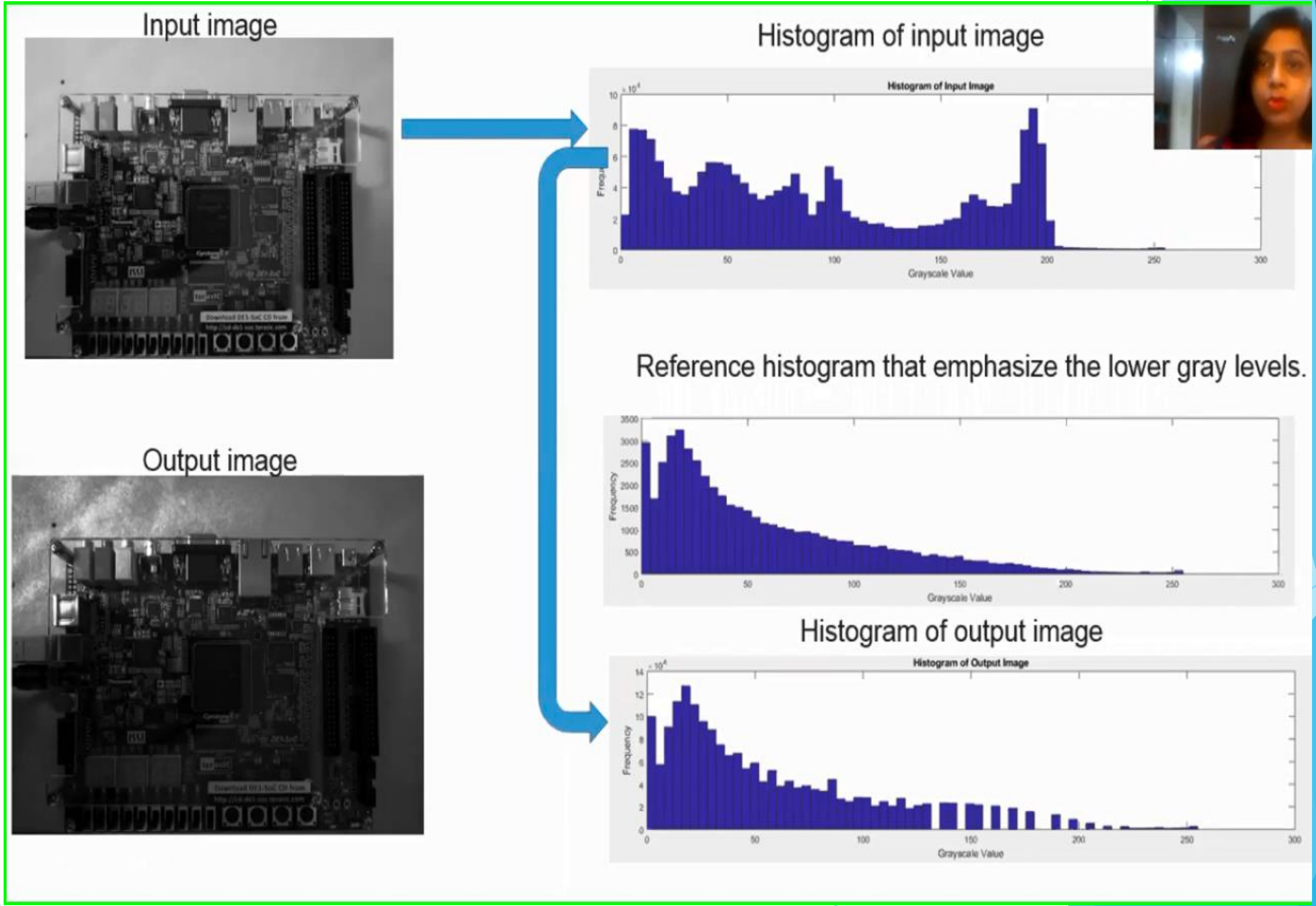

The goal of histogram equalization is to produce an output image that has a flattened histogram

The goal of histogram matching is to take an input image and generate an output image that is based upon the shape of a specific (or reference) histogram.

Histogram Matching / Histogram Specification

- Histogram matching is useful when we want to unify the contrast level of a group of images.

Histogram Matching / Histogram Specification (Cont.)


Input image
Histogram of input image
Histogram of Input Image
Grayscale Value
Frequency
Reference histogram that emphasize the lower gray levels.
Frequency
Grayscale Value
Output image
Histogram of output image
Histogram of Output Image
Frequency
Grayscale Value