# Digital Image Processing and Pattern Recognition

**E1528**

**Fall 2021-2022**



**Lecture 4**

# Histogram Equalization Techniques

## INSTRUCTOR

DR / AYMAN SOLIMAN

# Contents

- Objectives
- Basics of intensity transformations and spatial filtering
- Contrast Stretching
- Thresholding Function
- Intensity-Level Slicing
- Bit-Plane Slicing
- Histogram Processing
- Fixed Intensity Transformations
- Full-Scale Contrast Stretch
- Histogram Equalization

# ➢ Objectives

- ➢ Understand the meaning of spatial domain processing, and how it differs from transform domain processing.
- ➢ Be familiar with the principal techniques used for intensity transformations.
- ➢ Understand the physical meaning of image histograms and how they can be manipulated for image enhancement.
- ➢ Understand the mechanics of spatial filtering, and how spatial filters are formed.

# ➢ Objectives (cont.)

- ➢ Understand the principles of spatial convolution and correlation.
- ➢ Be familiar with the principal types of spatial filters, and how they are applied.
- ➢ Be aware of the relationships between spatial filters, and the fundamental role of lowpass filters.
- ➢ Understand how to use combinations of enhancement methods in cases where a single approach is insufficient.

# ➢ Background

- All the image processing techniques discussed in this lecture are implemented in the spatial domain, which is the plane containing the pixels of an image.

- Spatial domain techniques operate directly on the pixels of an image, as opposed, for example, to the frequency domain in which operations are performed on the Fourier transform of an image, rather than on the image itself.

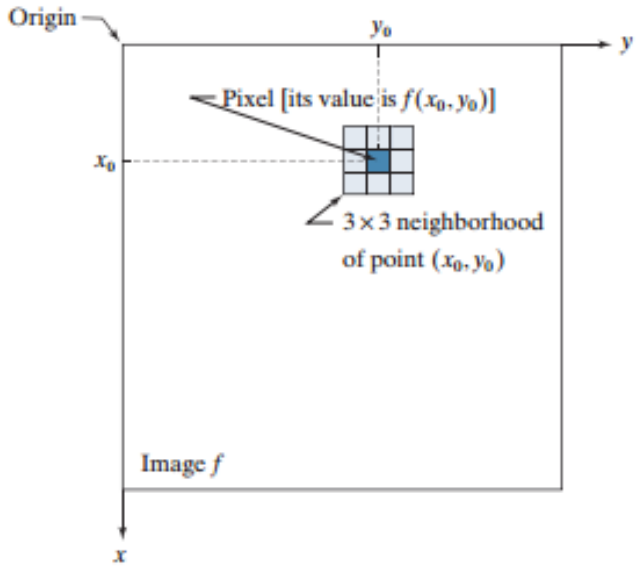
- As you will learn, some image processing tasks are easier or more meaningful to implement in the spatial domain, while others are best suited for other approaches.

# ➢ Basics of intensity transformations and spatial filtering

➢ The spatial domain processes we discuss are based on the expression

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

Where $f(x,y)$ is the input image, g(x, y) is the processed image, and T is an operator on f, defined over some neighborhood of (x, y).



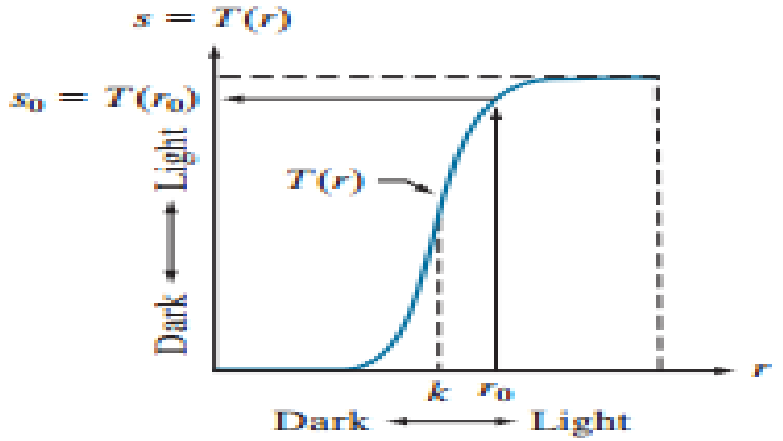
# ➢ Contrast Stretching

➢ The smallest possible neighborhood is of size $1 \times 1$ . In this case, g depends only on the value of f at a single point (x ,y ) and T becomes an intensity (also called a gray-level, or mapping) transformation function of the form

$$s = T(r)$$

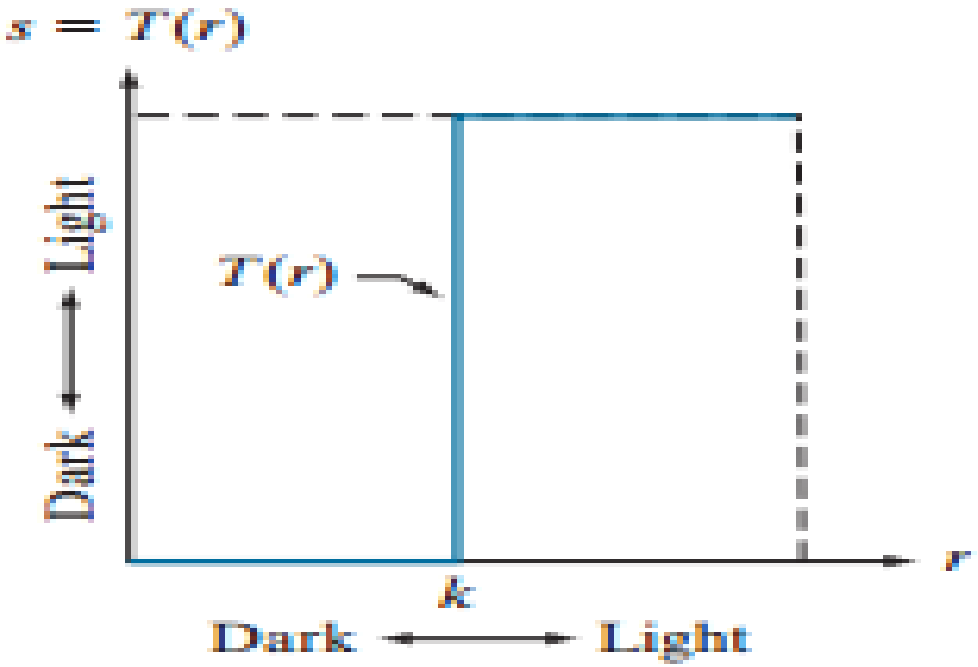
where, for simplicity in notation, we use *s* and *r* to denote, respectively, the intensity of *g* and *f* at any point (x , y).

By darkening the intensity levels below k and brightening the levels above k. In this technique, sometimes called contrast stretching



# Thresholding Function

- In the limiting case shown in Fig., T(r) produces a two-level (binary) image.
- A mapping of this form is called a thresholding function.
- Some simple yet powerful processing approaches can be formulated with intensity transformation functions.



# ➢ Intensity-Level Slicing

- There are applications in which it is of interest to highlight a specific range of intensities in an image.

- Some of these applications include enhancing features in satellite imagery, such as masses of water, and enhancing flaws in X-ray images.

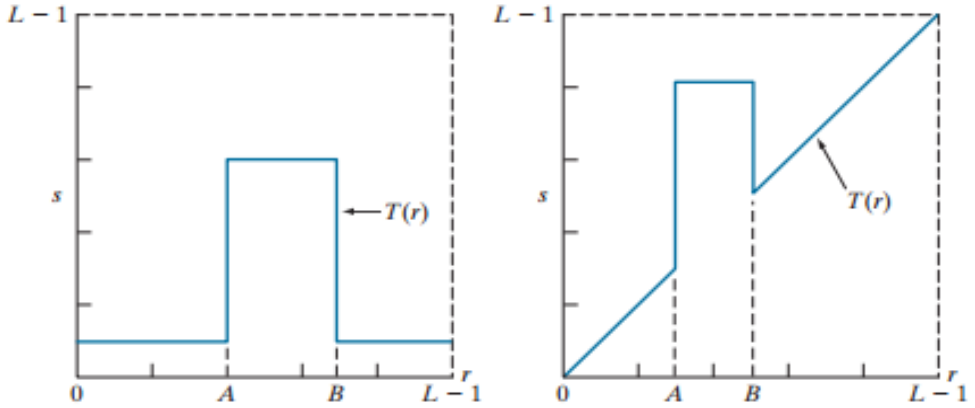
- The method, called intensity-level slicing, can be implemented in several ways, but most are variations of two basic themes.

# ➢ Intensity-Level Slicing (cont.)

- One approach is to display in one value (say, white) all the values in the range of interest and in another (say, black) all other intensities.
- This transformation, shown in Fig(a), produces a binary image.

a b

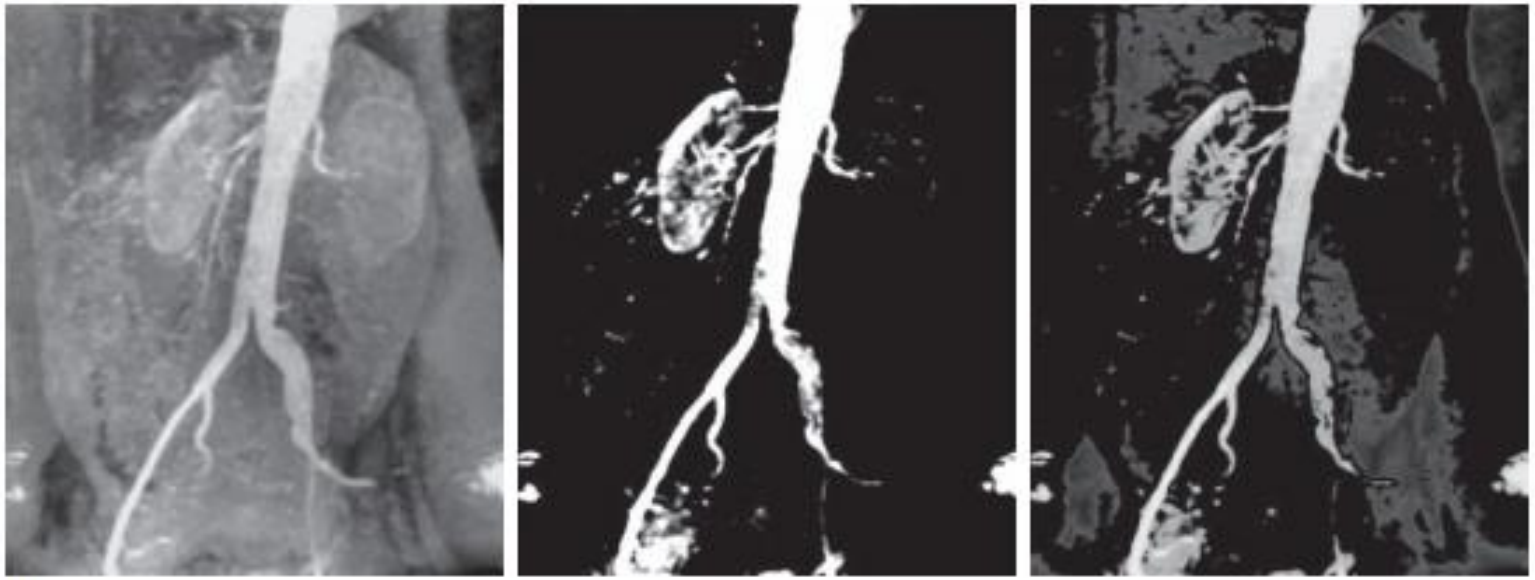
(a) This transformation function highlights range $[A, B]$ and reduces all other intensities to a lower level. (b) This function highlights range $[A, B]$ and leaves other intensities unchanged.




- The second approach, based on the transformation in Fig(b), brightens (or darkens) the desired range of intensities, but leaves all other intensity levels in the image unchanged.

# ➢ Intensity-Level Slicing Example

The objective of this example is to use intensity-level slicing to enhance the major blood vessels that appear lighter than the background, as a result of an injected contrast medium.



a b c

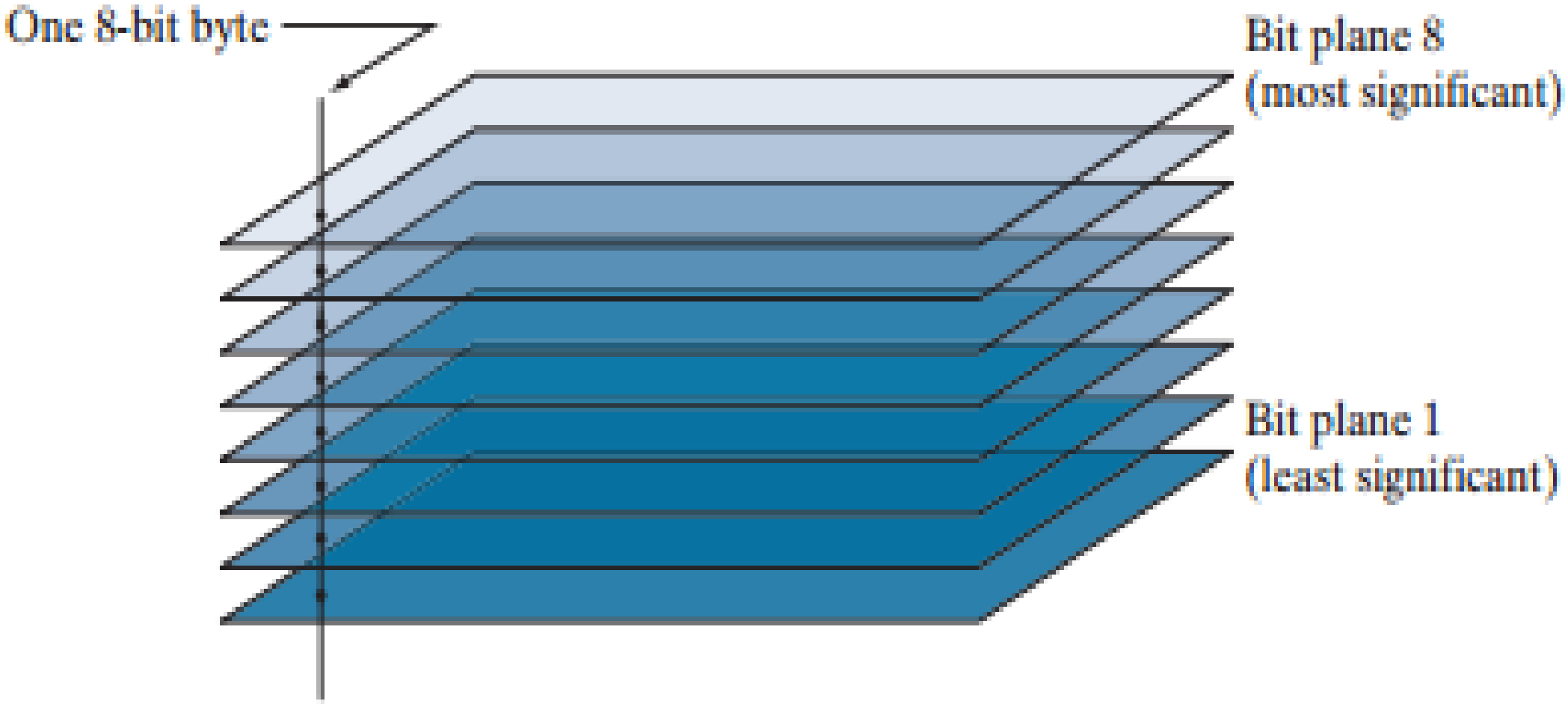
(a) Aortic angiogram. (b) Result of using a slicing transformation of the type illustrated in Fig.(a), with the range of intensities of interest selected in the upper end of the gray scale. (c) Result of using the transformation in Fig.(b), with the selected range set near black, so that the grays in the area of the blood vessels and kidneys were preserved.

# ➢ Bit-Plane Slicing

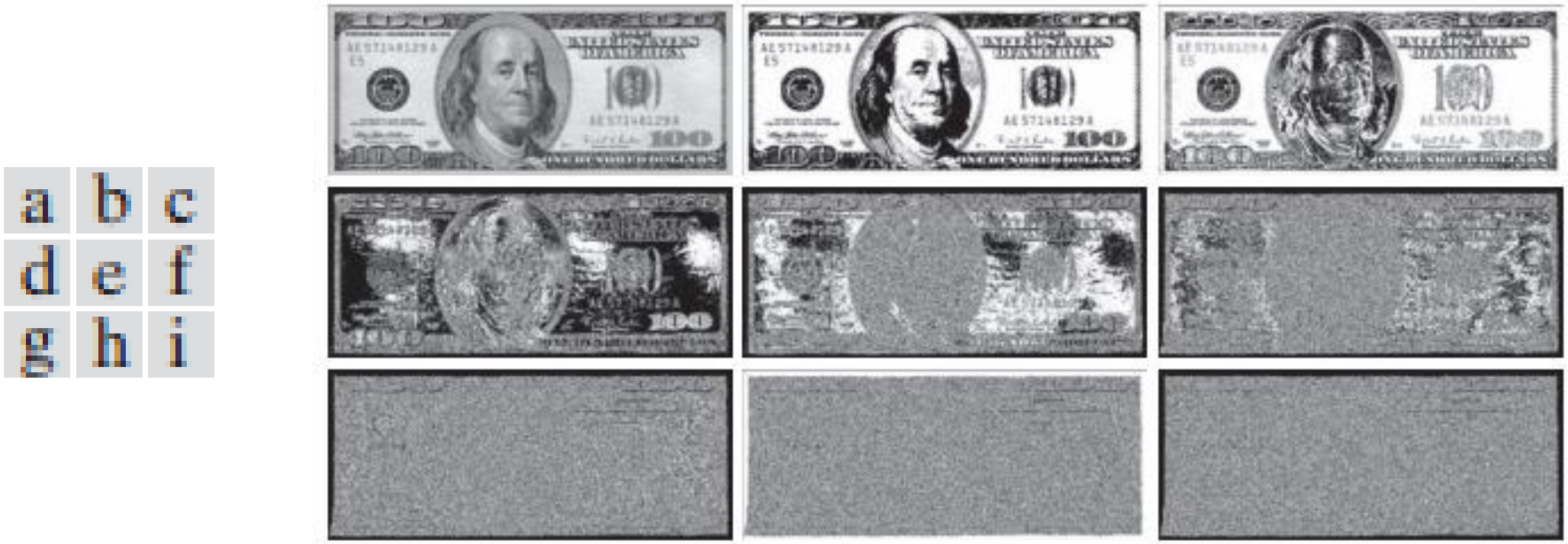
- Pixel values are integers composed of bits. For example, values in a 256-level grayscale image are composed of 8 bits (one byte).

- Instead of highlighting intensity-level ranges, we could highlight the contribution made to total image appearance by specific bits.
- As next Figure illustrates, an 8-bit image may be considered as being composed of eight one-bit planes, with plane 1 containing the lowest-order bit of all pixels in the image, and plane 8 all the highest-order bits.

# ➢ Bit-Plane Slicing (cont.)



# ➢ Bit-Plane Slicing Example




(a) An 8-bit gray-scale image of size 550 × 1192 pixels. (b) through (i) Bit planes 8 through 1, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image..

# ➢ Histogram Processing

➢ Let $r_k$, for k = 0 1 2 , , , , , … L -1 denote the intensities of an L-level digital image, f (x ,y ). The unnormalized histogram of f is defined as

$$h(r_k) = n_k \quad for\ k = 0, 1, 2, 3, \dots, L-1$$

Where $n_k$ is the number of pixels in *f* with intensity $r_k$, and the subdivisions of the intensity scale are called *histogram bins*. Similarly, the *normalized histogram* of *f* is defined as

$$p(r_k) = \frac{h(r_k)}{MN} = \frac{n_k}{MN}$$

# ➢ Histogram Processing (cont.)

$$p(r_k) = \frac{h(r_k)}{MN} = \frac{n_k}{MN}$$

where, as usual, M and N are the number of image rows and columns, respectively. Mostly, we work with normalized histograms, which we refer to simply as histograms or image histograms.

The sum of p ($r_k$) for all values of k is always 1.

The components of p($r_k$) are estimates of the probabilities of intensity levels occurring in an image.

# ➢ Histogram Processing (cont.)

- As you will learn in this section, histogram manipulation is a fundamental tool in image processing.

- Histograms are simple to compute and are also suitable for fast hardware implementations, thus making histogram-based techniques a popular tool for real-time image processing.

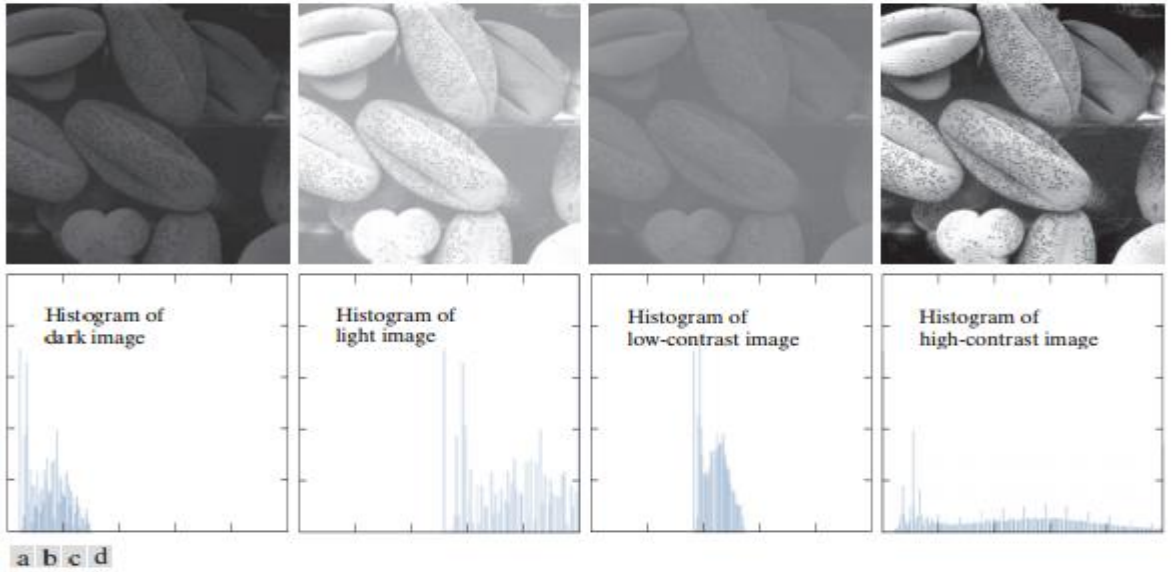
- Histogram shape is related to image appearance.

# ➢ Histogram Processing (cont.)

- For example, next figure shows images with four basic intensity characteristics: dark, light, low contrast, and high contrast; the image histograms are also shown.
- We note in the dark image that the most populated histogram bins are concentrated on the lower (dark) end of the intensity scale.
- Similarly, the most populated bins of the light image are biased toward the higher end of the scale. An image with low contrast has a narrow histogram located typically toward the middle of the intensity scale, as Fig.(c) shows.

# ➢ Histogram Processing (cont.)

- we see that the components of the histogram of the high-contrast image cover a wide range of the intensity scale, and the distribution of pixels is not too far from uniform, with few bins being much higher than the others.




Four image types and their corresponding histograms. (a) dark; (b) light; (c) low contrast; (d) high contrast. The horizontal axis of the histograms are values of $r_k$ and the vertical axis are values of $p(r_k)$.

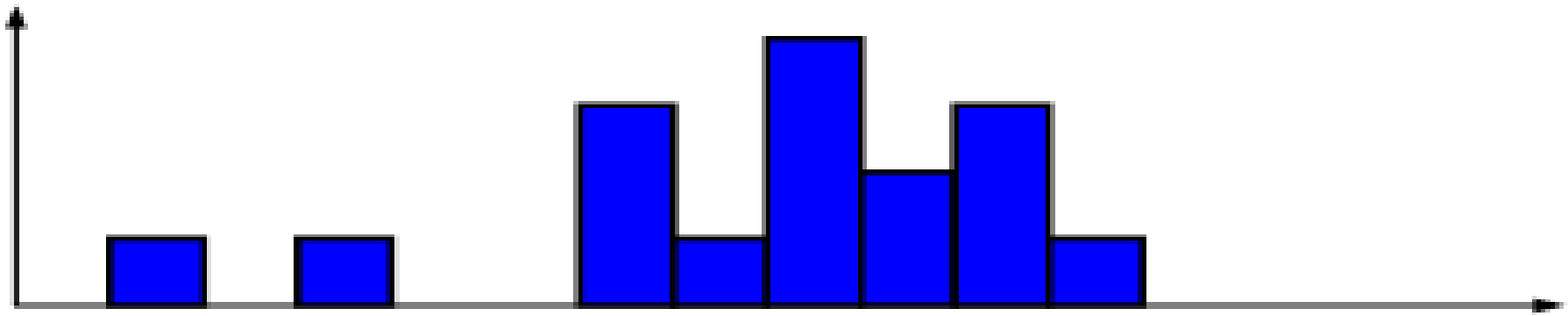
# ➢ Intensity Histogram

- **Example**

  a 4x4, 4bits/pixel image →

| | | | |
|---|---|---|---|
| 1 | 8 | 6 | 6 |
| 6 | 3 | 11 | 8 |
| 8 | 8 | 9 | 10 |
| 9 | 10 | 10 | 7 |

| k | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| H(k) | 0 | 1 | 0 | 1 | 0 | 0 | 3 | 1 | 4 | 2 | 3 | 1 | 0 | 0 | 0 | 0 |



# Fixed Intensity Transformations

Negative

$s = L - 1 - r$

Log

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

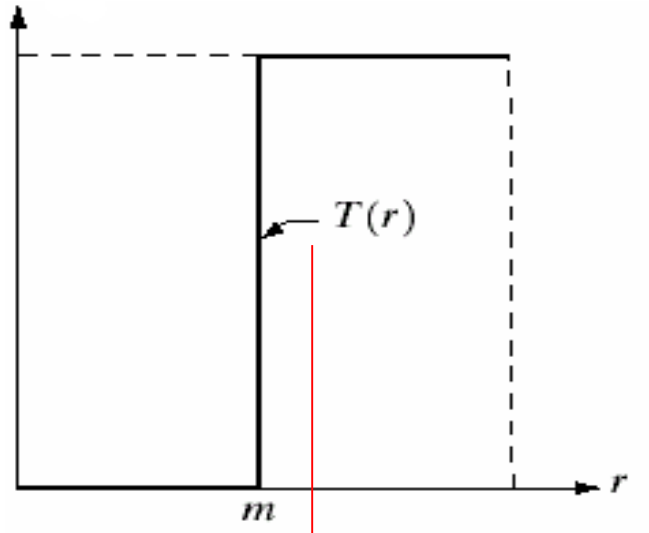
Inverse Log

$s = e^{cr} - 1$

Power-law

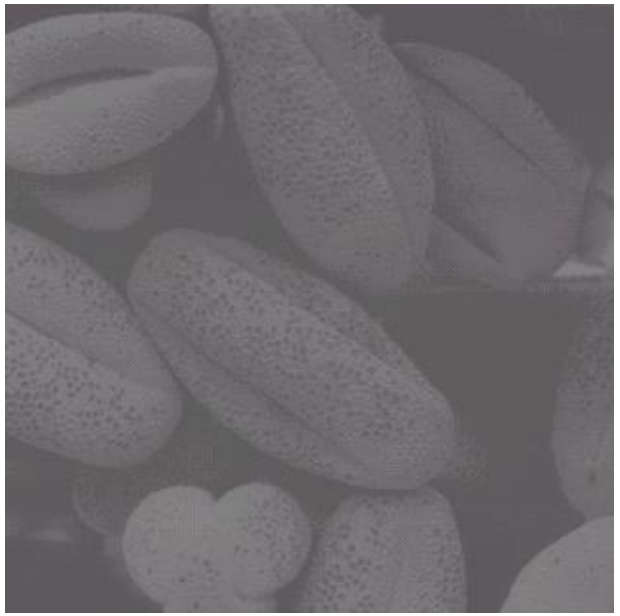
$s = cr^{\gamma}$

# ➢ Thresholding




$$s = \begin{cases} 0 & if\, r \le m \\ c & if\, r > m \end{cases}$$

$m$ : threshold



# ➢ Example: Fixed Intensity Transformation

- A 4x4, 4bits/pixel image

| | | | |
|---|---|---|---|
| 1 | 8 | 6 | 6 |
| 6 | 3 | 11 | 8 |
| 8 | 8 | 9 | 10 |
| 9 | 10 | 10 | 7 |

passes through an intensity transformation $s = T(r) = round\left(\frac{1}{15}r^2\right)$

1 → round(0.0667) = 0;

3 → round(0.6) = 1;

6 → round(2.4) = 2;

7 → round(3.2667) = 3;

8 → round(4.2667) = 4;

9 → round(5.4) =5;

10 → round(6.6667) = 7;

11 → round(8.0667) = 8;

The resulting image is:

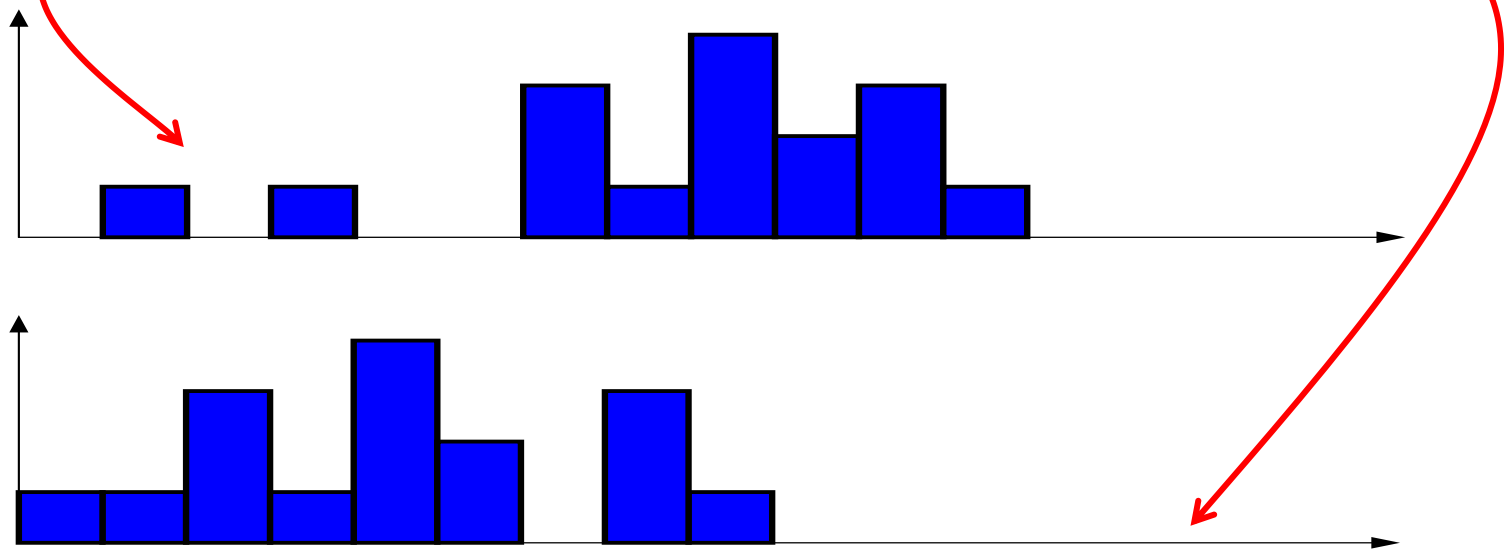
| | | | |
|---|---|---|---|
| 0 | 4 | 2 | 2 |
| 2 | 1 | 8 | 4 |
| 4 | 4 | 5 | 7 |
| 5 | 7 | 7 | 3 |

# ➢ Example: Histogram Change

| | | | |
|---|---|---|---|
| 1 | 8 | 6 | 6 |
| 6 | 3 | 11 | 8 |
| 8 | 8 | 9 | 10 |
| 9 | 10 | 10 | 7 |

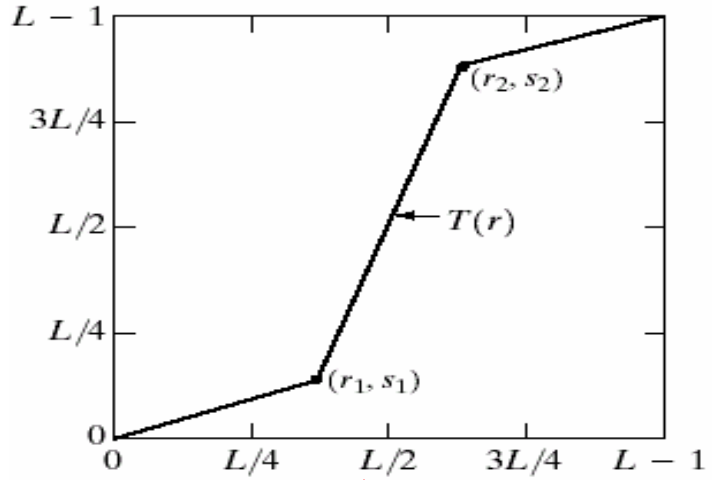
→

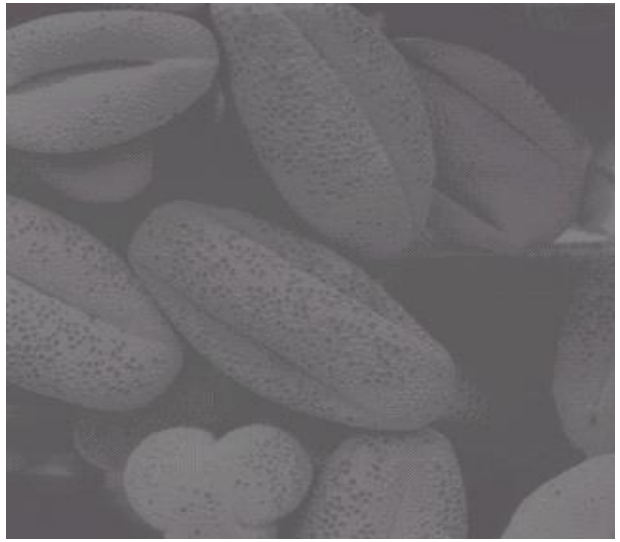
| | | | |
|---|---|---|---|
| 0 | 4 | 2 | 2 |
| 2 | 1 | 8 | 4 |
| 4 | 4 | 5 | 7 |
| 5 | 7 | 7 | 3 |



# Contrast Stretch

# ➢ General Idea: Make Best Use of the Dynamic Range



# ➢ Contrast Stretch

General form:

$$s = \begin{cases} \dfrac{s_1}{r_1} \cdot r & 0 \le r < r_1 \\ \dfrac{s_2 - s_1}{r_2 - r_1} \cdot r + \dfrac{s_1 r_2 - s_2 r_1}{r_2 - r_1} & r_1 \le r \le r_2 \\ \dfrac{2^B - 1 - s_2}{2^B - 1 - r_2} \cdot r + (2^B - 1) \cdot \dfrac{s_2 - r_2}{2^B - 1 - r_2} & r_2 < r \le 2^B - 1 \end{cases}$$

Special case → Full-scale contrast stretch:

$r_1 = r_{\min}$ $\quad s_1 = 0$

$r_2 = r_{\max}$ $\quad s_2 = 2^B - 1$

$$\longrightarrow \quad s = (2^B - 1) \cdot \frac{r - r_{\min}}{r_{\max} - r_{\min}}$$

Typically used: $\quad s = round\left((2^B - 1) \cdot \dfrac{r - r_{\min}}{r_{\max} - r_{\min}}\right)$

# ➢ Example: Full-Scale Contrast Stretch

- Full-scale contrast stretch of a 4x4, 4bits/pixel image

| | | | |
|---|---|---|---|
| 4 | 8 | 6 | 6 |
| 6 | 4 | 11 | 8 |
| 8 | 8 | 9 | 10 |
| 8 | 11 | 10 | 7 |

- Find when $r_{\min}$=4 $r_{\max}$=11 $2^B - 1 = 15$

$$s = round\left((2^B - 1) \cdot \frac{r - r_{\min}}{r_{\max} - r_{\min}}\right) = round\left(15 \cdot \frac{r - 4}{11 - 4}\right) = round\left(\frac{15}{7}(r - 4)\right)$$

4 → round(0) = 0;
6 → round(4.29) = 4;
7 → round(6.43) = 6;
8 → round(8.57) = 9;
9 → round(10.71) = 11;
10 → round(12.86) = 13;
11 → round(15) = 15;

The resulting image is:

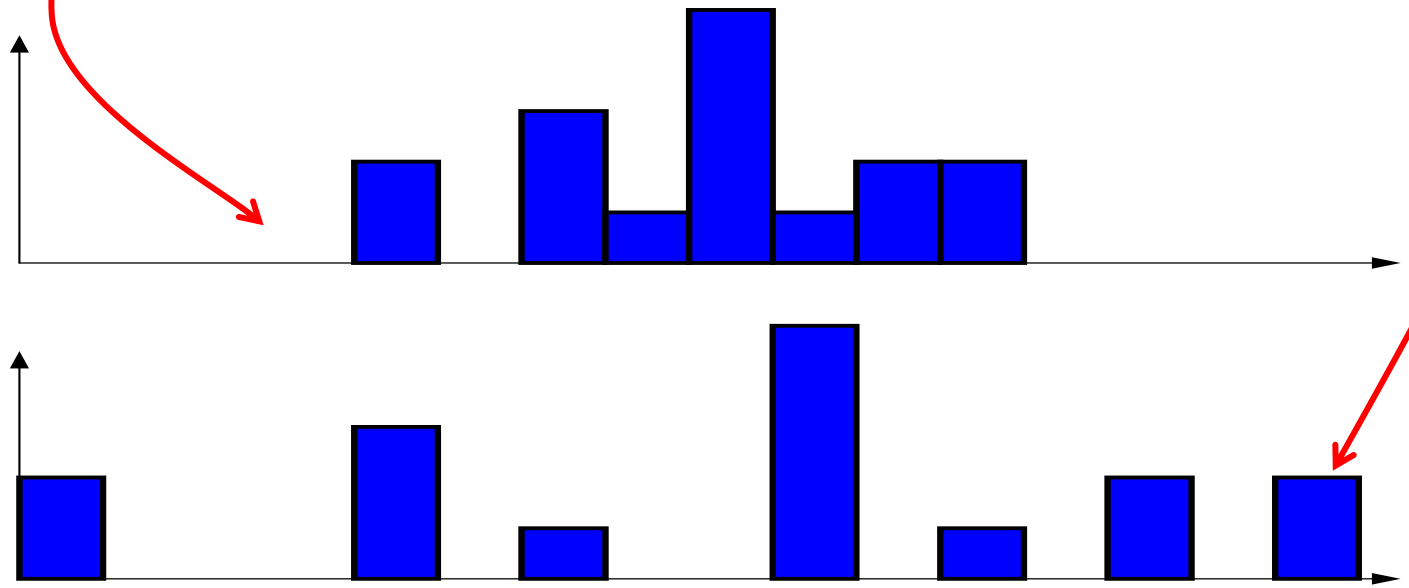
| | | | |
|---|---|---|---|
| 0 | 9 | 4 | 4 |
| 4 | 0 | 15 | 9 |
| 9 | 9 | 11 | 13 |
| 9 | 15 | 13 | 6 |

# ➢ Example: Histogram Change

| | | | |
|---|---|---|---|
| 4 | 8 | 6 | 6 |
| 6 | 4 | 11 | 8 |
| 8 | 8 | 9 | 10 |
| 8 | 11 | 10 | 7 |

→

| | | | |
|---|---|---|---|
| 0 | 9 | 4 | 4 |
| 4 | 0 | 15 | 9 |
| 9 | 9 | 11 | 13 |
| 9 | 15 | 13 | 6 |



# Histogram Equalization

# ➢ Introduction to Histogram Equalization

- Image pre-processing is the term for operations on the images at the lowest level of abstraction. These operations do not increase image information content, but they decrease it if entropy is an information measure.
- The aim of pre-processing is an improvement of the image data that suppresses undesired distortions or enhances some image features relevant for further processing and analysis tasks.

# ➢ Introduction to Histogram Equalization

- There are four different types of Image Pre-Processing techniques, and they are listed below.
  1) Pixel brightness transformations/ Brightness corrections
  2) Geometric Transformations
  3) Image Filtering and Segmentation
  4) Fourier transform and Image restoration
- Histogram equalization is one of the Pixel brightness transformations techniques. It is a well-known contrast enhancement technique due to its performance on almost all types of image.

# ➢ Histogram Equalization

- A histogram is a representation of frequency distribution. It is the basis for numerous spatial domain processing techniques. Histogram manipulation can be used for image enhancement.

- Contrast is defined as the difference in intensity between two objects in an image.

- If the contrast is too low, it is impossible to distinguish between two objects, and they are seen as a single object.

# ➢ Histogram Equalization (cont.)

- Histogram equalization is a widely used contrast-enhancement technique in image processing because of its high efficiency and simplicity.
- It is one of the sophisticated methods for modifying the dynamic range and contrast of an image by altering that image such that its intensity histogram has the desired shape.
- It can be classified into two branches as per the transformation function is used.
    1) Global histogram equalization (GHE)
    2) Local histogram equalization (LHE)

# ➢ Global histogram equalization (GHE)

- GHE is very simple and fast, but its contrast enhancement power is low.

- Here the histogram of the whole input image is used to compute the histogram transformation function.

- As a result, the dynamic range of the image histogram is flattened and stretched. The overall contrast is improved.

# ➢ Local histogram equalization (LHE)

- LHE can enhance the overall contrast more effectively.

- One of the drawbacks of histogram equalization is that it can change the mean brightness of an image significantly because of histogram flattening and sometimes this is not a desirable property when preserving the original mean brightness of a given image is necessary. Bi-Histogram Equalization was proposed to overcome this problem.

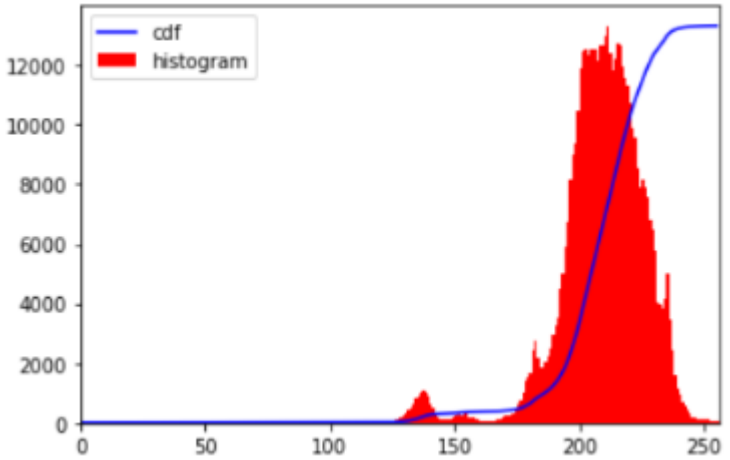
# ➢ Steps Involved

- ➢ Get the input image
- ➢ Generate the histogram for the image
- ➢ Find the local minima of the image
- ➢ Divide the histogram based on the local minima
- ➢ Have the specific gray levels for each partition of the histogram
- ➢ Apply the histogram equalization on each partition

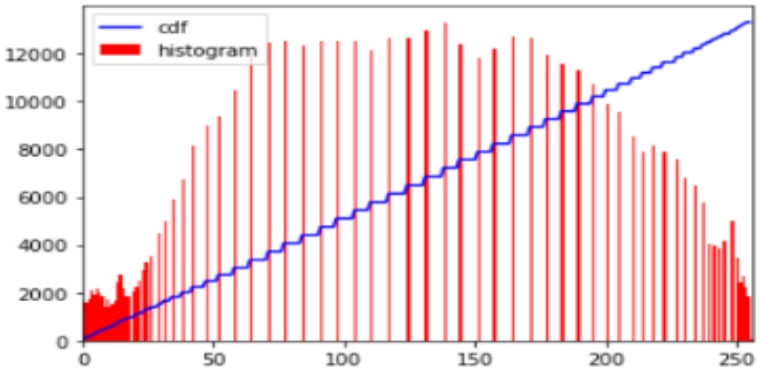
# ➢ Example

➢ The below example shows how Histogram equalization modifies the contrast of the Image.

Input Image




output Image



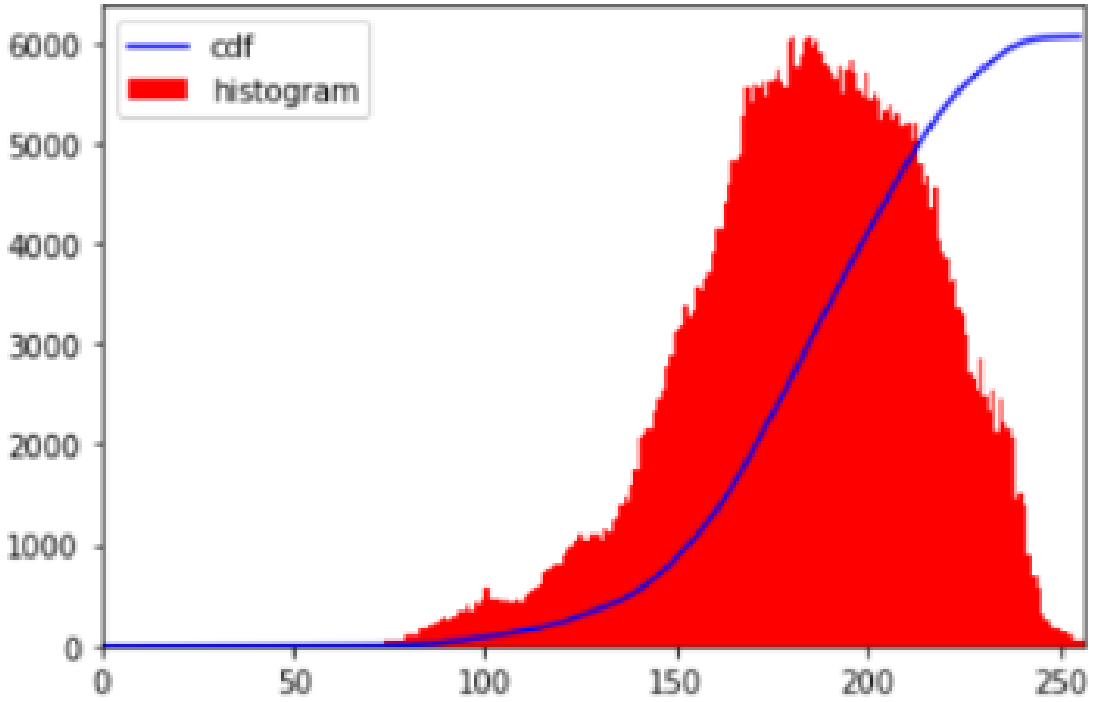
# ➢ CLAHE (Contrast Limited Adaptive Histogram Equalization)

- The above histogram equalization considers the global contrast of the image, and in many cases, it is not a good idea. The human pic in the original diagram is not shown correctly in the histogram equalization output. Even though the contrast of the image had been improved, we lost some of the information due to over brightness. The reason is that the histogram is not confined to the local region.

- Adaptive Histogram Equalization helps to solve this issue. In this method, the image is divided into small blocks, and each of these blocks is histogram equalized.

# ➢ CLAHE (Contrast Limited Adaptive Histogram Equalization)

➢ The same image has been converted, and below is the output of Adaptive Histogram Equalization.



# ➢ Example

- A 4x4, 4bits/pixel image

| | | | |
|---|---|---|---|
| 2 | 8 | 9 | 9 |
| 2 | 3 | 10 | 9 |
| 8 | 3 | 3 | 11 |
| 8 | 3 | 10 | 11 |

- First try full-scale contrast stretch $r_{\min}$=2 $r_{\max}$=11

$$s = round\left((2^B - 1) \cdot \frac{r - r_{\min}}{r_{\max} - r_{\min}}\right) = round\left(15 \cdot \frac{r-2}{11-2}\right) = round\left(\frac{5}{3}(r-2)\right)$$

2 → round(0) = 0;
3 → round(1.67) = 2;
8 → round(10.00) = 10;
9 → round(11.67) = 12;
10 → round(13.33) = 13;
11 → round(15) = 15;

The resulting image is:

| | | | |
|---|---|---|---|
| 0 | 10 | 12 | 12 |
| 0 | 2 | 13 | 12 |
| 10 | 2 | 2 | 15 |
| 10 | 2 | 13 | 15 |

# ➢ Example: Histogram Change

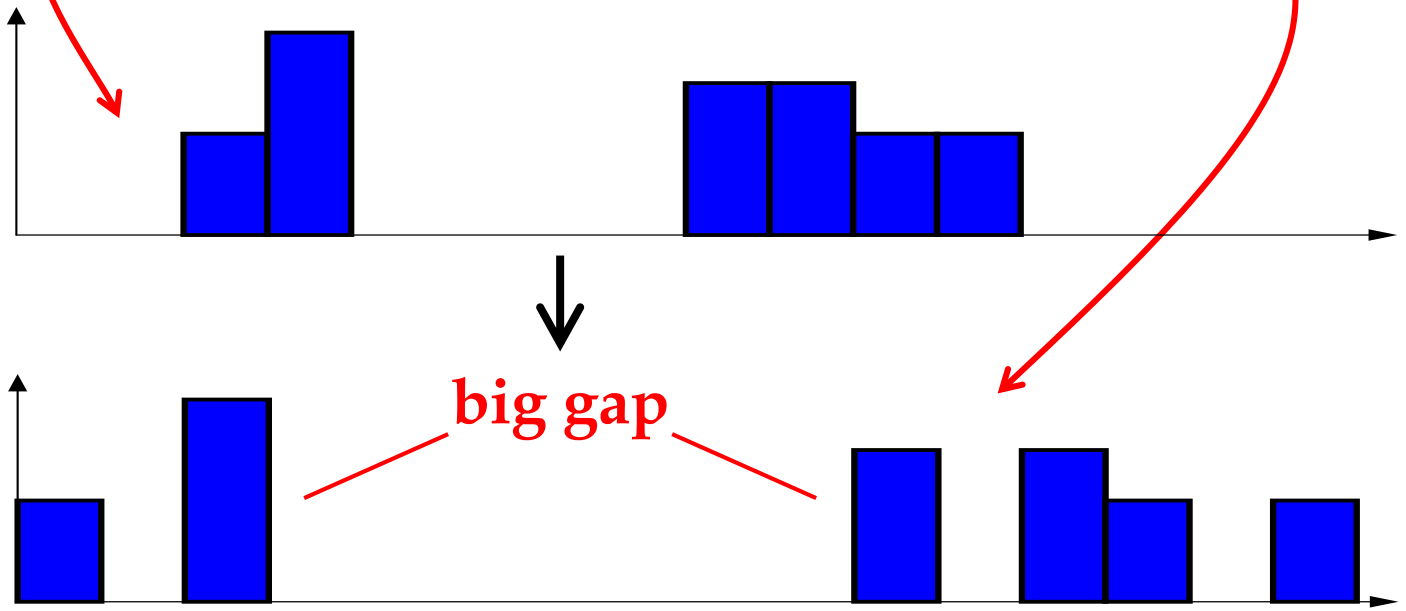
original

| | | | |
|---|---|---|---|
| 2 | 8 | 9 | 9 |
| 2 | 3 | 10 | 9 |
| 8 | 3 | 3 | 11 |
| 8 | 3 | 10 | 11 |

→

| | | | |
|---|---|---|---|
| 0 | 10 | 12 | 12 |
| 0 | 2 | 13 | 12 |
| 10 | 2 | 2 | 15 |
| 10 | 2 | 13 | 15 |

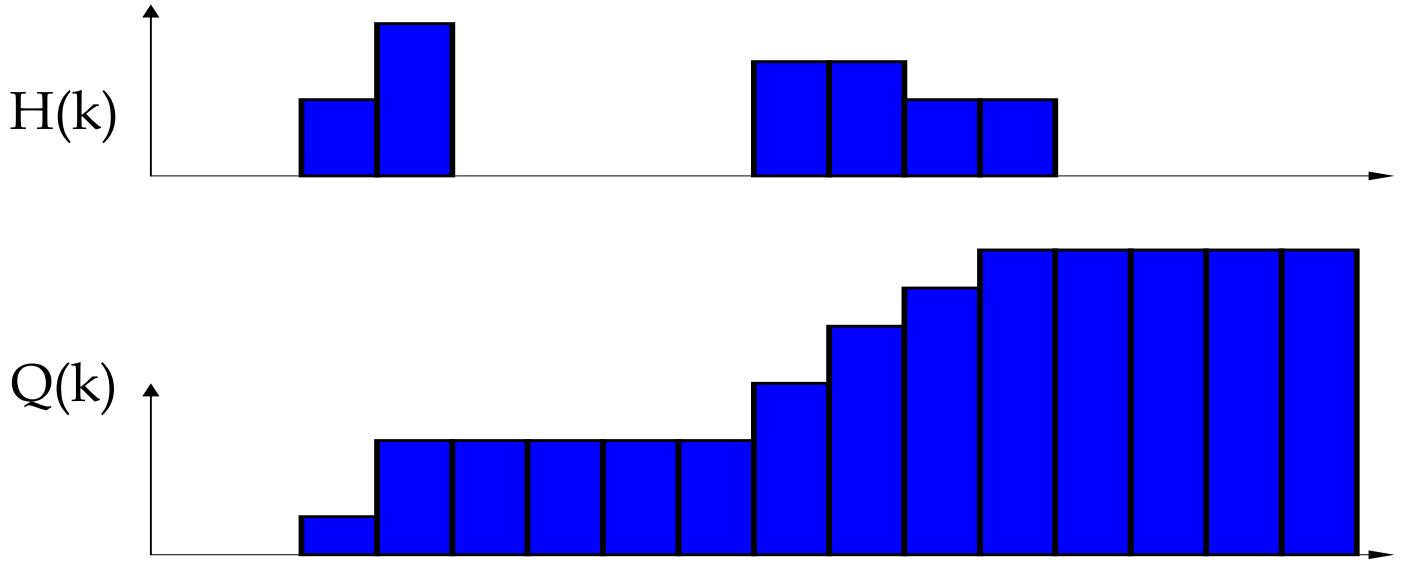
full-scale contrast stretch



# ➢ Cumulative Histogram

| | | | |
|---|---|---|---|
| 2 | 8 | 9 | 9 |
| 2 | 3 | 10 | 9 |
| 8 | 3 | 3 | 11 |
| 8 | 3 | 10 | 11 |

| k | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| H(k) | 0 | 0 | 2 | 4 | 0 | 0 | 0 | 0 | 3 | 3 | 2 | 2 | 0 | 0 | 0 | 0 |
| Q(k) | 0 | 0 | 2 | 6 | 6 | 6 | 6 | 6 | 9 | 12 | 14 | 16 | 16 | 16 | 16 | 16 |



# ➢ Intermediate Image

| k | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| H(k) | 0 | 0 | 2 | 4 | 0 | 0 | 0 | 0 | 3 | 3 | 2 | 2 | 0 | 0 | 0 | 0 |
| Q(k) | 0 | 0 | 2 | 6 | 6 | 6 | 6 | 6 | 9 | 12 | 14 | 16 | 16 | 16 | 16 | 16 |

original

| | | | |
|---|---|---|---|
| 2 | 8 | 9 | 9 |
| 2 | 3 | 10 | 9 |
| 8 | 3 | 3 | 11 |
| 8 | 3 | 10 | 11 |

→

| | | | |
|---|---|---|---|
| 2 | 9 | 12 | 12 |
| 2 | 6 | 14 | 12 |
| 9 | 6 | 6 | 16 |
| 9 | 6 | 14 | 16 |

intermediate image

# ➢ Full-Scale Contrast Stretch of Intermediate Image

intermediate image

| | | | |
|---|---|---|---|
| 2 | 9 | 12 | 12 |
| 2 | 6 | 14 | 12 |
| 9 | 6 | 6 | 16 |
| 9 | 6 | 14 | 16 |

$r_{\min}=2$ $r_{\max}=16$

$$s = round\left((2^B - 1)\cdot\frac{r - r_{\min}}{r_{max} - r_{\min}}\right) = \text{round}\left(15\cdot\frac{r-2}{16-2}\right) = \text{round}\left(\frac{15}{14}(r-2)\right)$$

2 → round(0) = 0;
6 → round(4.29) = 4;
9 → round(7.50) = 8;
12 → round(10.71) = 11;
14 → round(12.86) = 13;
16 → round(15) = 15;

result:
histogram equalized image

| | | | |
|---|---|---|---|
| 0 | 8 | 11 | 11 |
| 0 | 4 | 13 | 11 |
| 8 | 4 | 4 | 15 |
| 8 | 4 | 13 | 15 |

# ➢ Histogram Comparison

| | | | |
|---|---|---|---|
| 4 | 8 | 6 | 6 |
| 6 | 4 | 11 | 8 |
| 8 | 8 | 9 | 10 |
| 8 | 11 | 10 | 7 |

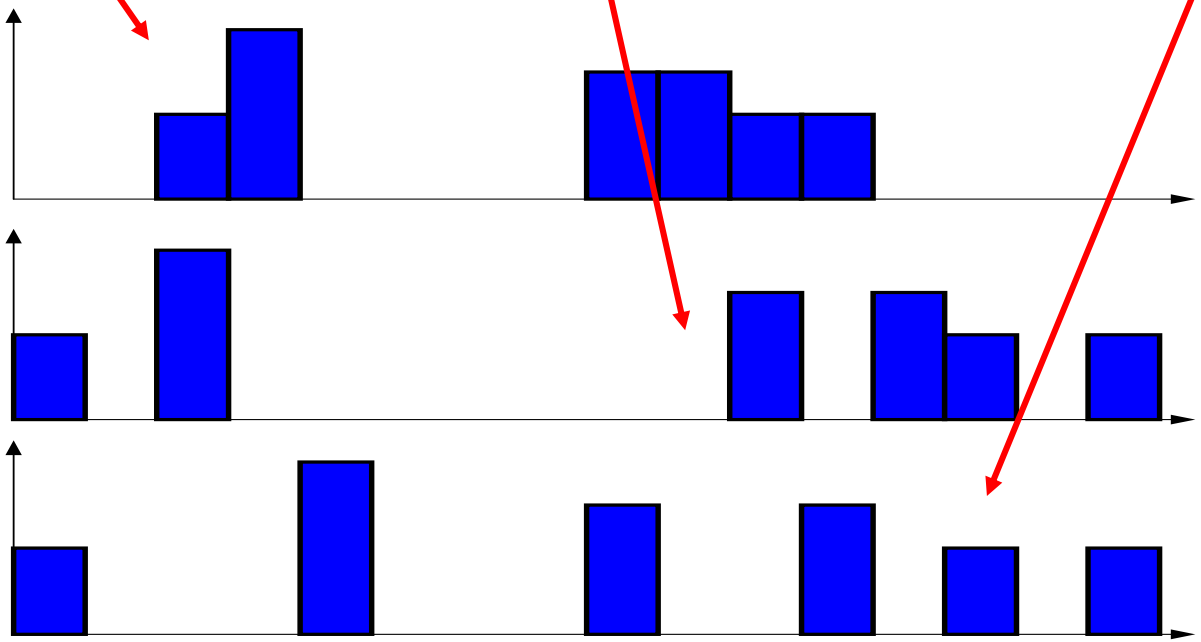
original

| | | | |
|---|---|---|---|
| 0 | 10 | 12 | 12 |
| 0 | 2 | 13 | 12 |
| 10 | 2 | 2 | 15 |
| 10 | 2 | 13 | 15 |

direct full-scale contrast stretch

| | | | |
|---|---|---|---|
| 0 | 8 | 11 | 11 |
| 0 | 4 | 13 | 11 |
| 8 | 4 | 4 | 15 |
| 8 | 4 | 13 | 15 |

histogram-equalized



more equalized

# ➢ Histogram Equalization wit MATLAB

- You can adjust the intensity values of image pixels automatically using histogram equalization.

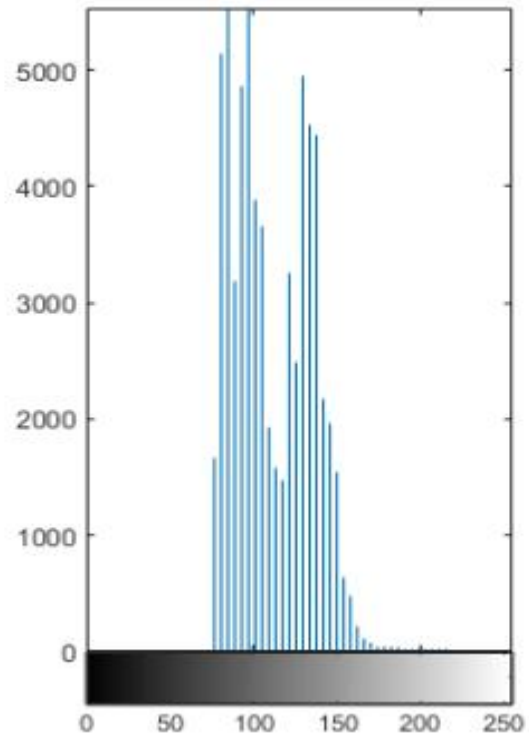
- Histogram equalization involves transforming the intensity values so that the histogram of the output image approximately matches a specified histogram. By default, the histogram equalization function, histeq, tries to match a flat histogram with 64 bins, but you can specify a different histogram instead.

# ➢ Histogram Equalization wit MATLAB

- This example shows how to use histogram equalization to adjust the contrast of a grayscale image. The original image has low contrast, with most pixel values in the middle of the intensity range. histeq produces an output image with pixel values evenly distributed throughout the range.

- Read an image into the workspace.

```
I = imread('pout.tif');
figure
subplot(1,2,1)
imshow(I)
subplot(1,2,2)
imhist(I,64)
```



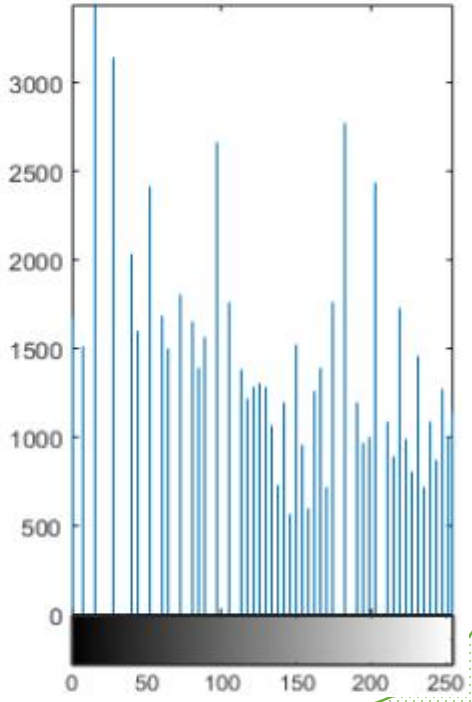
# ➢ Histogram Equalization wit MATLAB

- Adjust the contrast using histogram equalization. In this example, the histogram equalization function, histeq, tries to match a flat histogram with 64 bins, which is the default behavior. You can specify a different histogram instead.

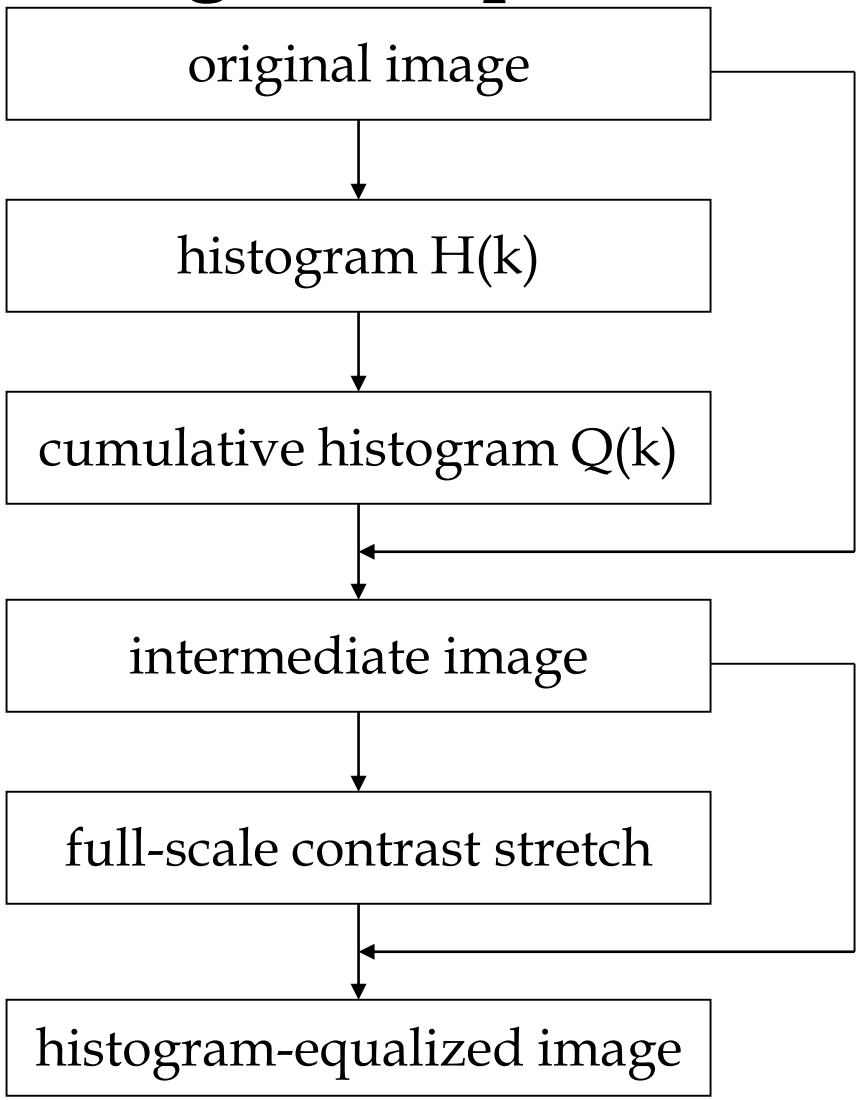
```
J = histeq(I);
```

- Display the contrast-adjusted image and its new histogram.

```
figure
subplot(1,2,1)
imshow(J)
subplot(1,2,2)
imhist(J,64)
```



# ➢ Summary of the Histogram Equalization Algorithm



# Thank you