

Computer Vision, and Digital Image Processing

Prepared by Dr. Mohamed Mostafa

AI 612

Lecture 3



Intensity Transformations and Spatial Filtering

Lecture 3

AI 612

Prepared by
Dr. Mohamed Mostafa

www.midocean.ae

Chapter

02

Intensity Transformations and Spatial Filtering



What is Spatial Domain?

- The term *spatial domain* refers to the image plane itself.
- Image processing methods in this category are based on direct manipulation of pixels in an image.

Two principal categories of spatial processing :

- **Intensity transformations** : operate on single pixels of an image for tasks such as contrast manipulation and image thresholding.
- **Spatial filtering**: performs operations on the neighborhood of every pixel in an image. include image smoothing and sharpening

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

$f(x, y)$: is an input image,

$g(x, y)$ is the output image,

T is an operator on f defined over a neighborhood of point (x, y) .



Number of Basic Intensity Transformation Functions?

- IMAGE NEGATIVES
- LOG TRANSFORMATIONS
- POWER-LAW (GAMMA) TRANSFORMATIONS

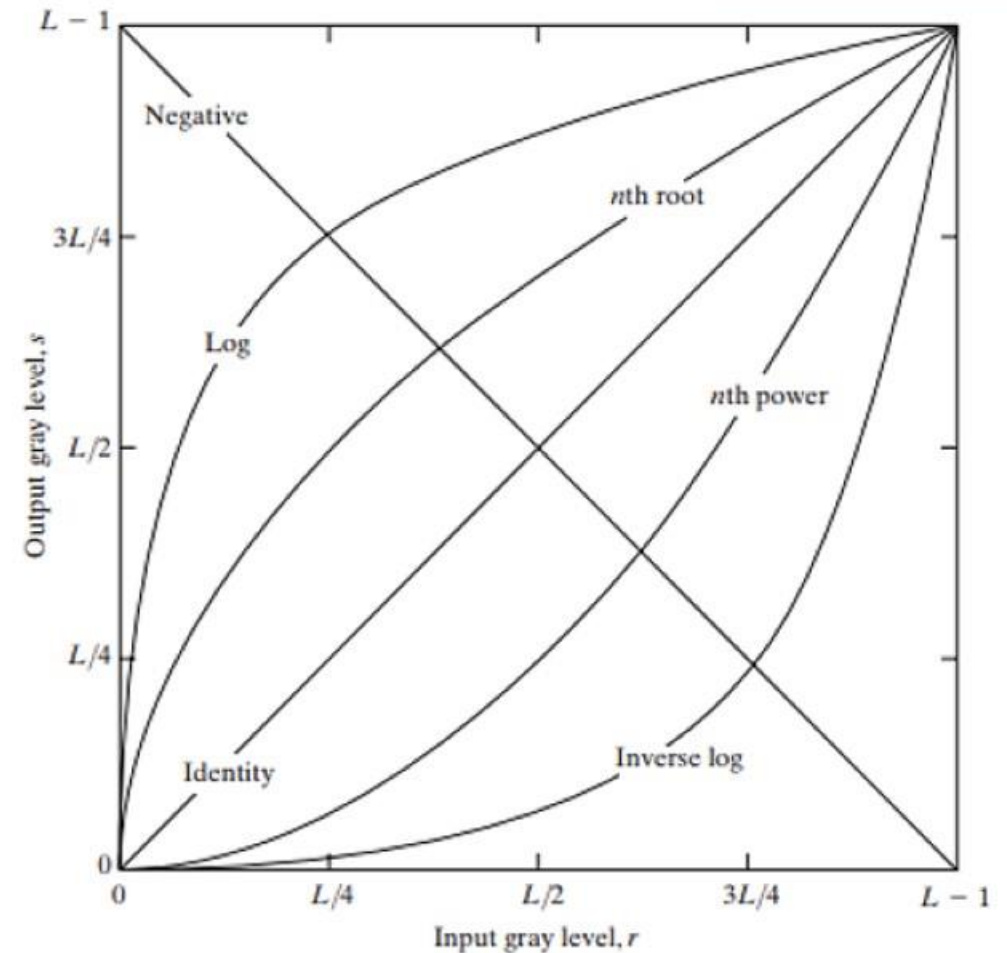


Image Negatives

The **negative of an image (monochrome image)** with intensity levels in the range $[0, L - 1]$ is obtained by using the negative transformation function

$$s = L - 1 - r$$

Let us take an 8-bit image with intensity range of 0–255.

then if the pixel is Black (0) to inverse it ($255 - 0 = 255$, which means White color)

The White to opposite ($255 - 255 = 0$ then goes to Black)



Input image (X-ray image)



Output image (negative)

Image Negatives

Find the negative of the current image with 255 gray levels

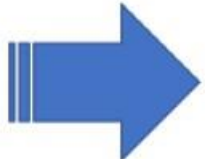
Gray_image = $\begin{bmatrix} 100 & 150 & 200 \\ 50 & 75 & 100 \\ 25 & 0 & 255 \end{bmatrix}$  Negative Image = $\begin{bmatrix} 155 & 105 & 55 \\ 205 & 180 & 155 \\ 230 & 255 & 0 \end{bmatrix}$

Image Negatives

For color image, we can convert it into its negative form by simply subtracting the RGB intensity value from 255.

Mathematically,

$$R = 255 - R$$

$$G = 255 - G$$

$$B = 255 - B$$



Image Negatives

Problem:

Consider a colored image with RGB channels represented as matrices. The pixel values range from 0 to 255, where 0 represents black and 255 represents white.

Calculate the negative values for a pixel with **RGB values (100, 150, 200)**.

Image Negatives

Answer:

Given RGB values: **R = 100, G = 150, B = 200**

Negative_R = $255 - R = 255 - 100 = 155$

Negative_G = $255 - G = 255 - 150 = 105$

Negative_B = $255 - B = 255 - 200 = 55$

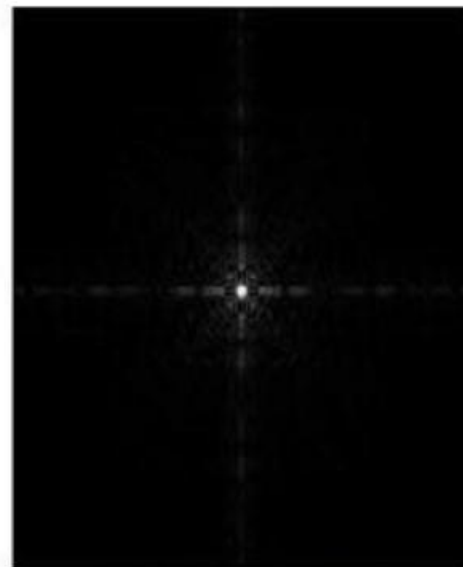
So, the negative values for the pixel are **(155, 105, 55)**.

Log Transformations

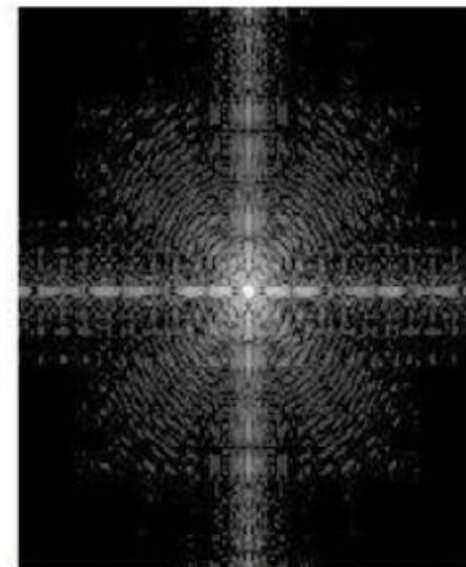
$$s = c \cdot \log(1 + r)$$

where:

- s is the output pixel value,
- r is the input pixel value,
- c is a scaling factor.



(a) Original image



(b) Result of Log transform with $c = 1$

Power-law (Gamma) Transformations

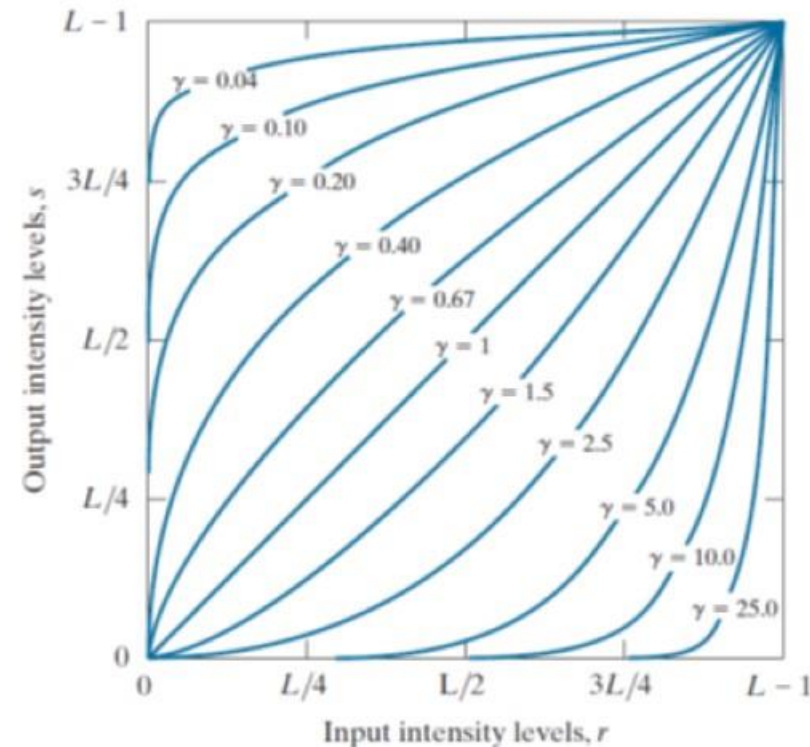
The brightness of an image can be adjusted with a gamma correction transformation, by applying a power-law function to its pixel values. The general form of a power-law transformation is given by:

$$s = c \cdot r^\gamma$$

where:

- s is the output pixel value,
- r is the input pixel value,
- c is a scaling factor,
- γ is the gamma parameter.

If $\gamma > 1.0$, then the result is logarithmic curves that darken an image. RGB monitors have gamma values of 1.4 to 2.8.



Power-law (Gamma) Transformations



Power-law (Gamma) Transformations

Problem:

Suppose we have a grayscale image with 256 gray levels and we want to perform a power-law (gamma) transformation on it.

Let's consider the following parameters for the transformation:

- Input intensity level (r) = 100
- Constant scaling factor (c) = 1.5
- Gamma value (γ) = 0.5

Power-law (Gamma) Transformations

Answer:

Given intensity level (r) = 100

Then using the transformation Function $s = c \cdot r^\gamma$

$$s = 1.5 \cdot 100^{0.5}$$

$$s = 1.5 \cdot \sqrt{100}$$

$$s = 1.5 \cdot 10$$

$$s = 15$$

So, the new value for the pixel is **(15)**.

Power-law (Gamma) Transformations



$\gamma = 0.6$



$\gamma = 0.4$



$\gamma = 0.3$

Piecewise Transformations

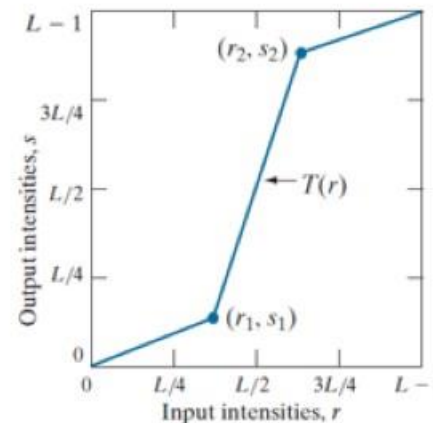
Instead of applying a single transformation function to the entire image, different regions of the image are subjected to different transformations. This approach allows for more flexibility in adjusting pixel values and enhancing specific features in different parts of the image.

Types:

Contrast stretching,
Gray level slicing and
Bit-Plane slicing

Contrast Stretching Transformations

Contrast stretching, also known as intensity stretching or normalization, is an image processing technique used to enhance the contrast of an image by expanding the range of pixel values. The goal is to improve the visibility of details and features within the image.



Contrast Stretching Transformations

- **Problem:**

Consider a gray image with **256 gray-levels** represented by pixel values ranging from 0 to 255. Apply contrast stretching transformation to enhance the contrast of the image with the following parameters:

- Original pixel values: [**100, 150, 200, 50, 75, 225, 175, 125**]
- New pixel values after contrast stretching transformation: ?

Contrast Stretching Transformations

- Answer:

Determine the minimum and maximum pixel values in the original image:

Minimum pixel value (Min): 50

Maximum pixel value (Max): 225

Compute the range of pixel values in the original image:

Range = Max - Min = 225 - 50 = 175

Define the desired minimum and maximum pixel values for the stretched image (commonly 0 and 255, respectively).

Apply the contrast stretching transformation to each pixel in the original image using the formula:

$$\text{NewPixelValue} = (\text{OriginalPixelValue} - \text{Min}) * (\text{NewMax} / \text{Range})$$

Substitute the values into the formula:

NewMin = 0 (desired minimum pixel value)

NewMax = 255 (desired maximum pixel value)

Contrast Stretching Transformations

- Answer:

For each original pixel value:

For pixel value 50: $\text{NewPixelValue} = (50 - 50) * (255 / 175) = 0 * 1.457 = 0$

For pixel value 75: $\text{NewPixelValue} = (75 - 50) * (255 / 175) = 25 * 1.457 = 36.425 \approx 36$

For pixel value 100: $\text{NewPixelValue} = (100 - 50) * (255 / 175) = 50 * 1.457 = 72.85 \approx 73$

For pixel value 125: $\text{NewPixelValue} = (125 - 50) * (255 / 175) = 75 * 1.457 = 109.275 \approx 109$

For pixel value 150: $\text{NewPixelValue} = (150 - 50) * (255 / 175) = 100 * 1.457 = 145.7 \approx 146$

For pixel value 175: $\text{NewPixelValue} = (175 - 50) * (255 / 175) = 125 * 1.457 = 182.125 \approx 182$

For pixel value 200: $\text{NewPixelValue} = (200 - 50) * (255 / 175) = 150 * 1.457 = 218.55 \approx 219$

For pixel value 225: $\text{NewPixelValue} = (225 - 50) * (255 / 175) = 175 * 1.457 = 255.475 \approx 255$

The new pixel values after contrast stretching transformation are approximately: [0, 36, 73, 109, 146, 182, 219, 255]

Intensity Level Slicing Transformations

Intensity level slicing, also known as thresholding or binary slicing, is an image processing technique that involves dividing an image into regions based on the intensity levels of its pixels. The goal of intensity level slicing is often to highlight specific features or regions within an image by isolating pixels with intensity values falling within a certain range.

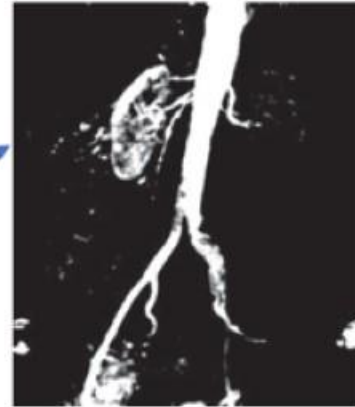
1. Thresholding:

Intensity level slicing is essentially a form of thresholding where a threshold or range of intensity values is defined, and pixels with intensities within that range are treated differently from those outside the range.

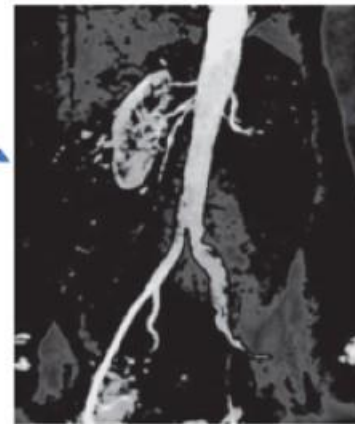
2. Binary Image Creation:

The result of intensity level slicing is often a binary image, where pixels within the specified intensity range are assigned one value (usually white), and pixels outside the range are assigned another value (usually black). This creates a high-contrast representation of the selected features.

Intensity Level Slicing Transformations



Binary Image Creation



Thresholding

Intensity Level Slicing Transformations

Thresholding Example:

Let's consider a grayscale image represented by the following pixel values (ranging from 0 to 255)

[25, 50, 75, 100, 125, 150, 175, 200, 225, 250]

Now, let's apply intensity level slicing transformation using thresholding to highlight pixels with intensity values between 100 and 200.

Define the threshold range:

- Lower threshold (T1): **100**
- Upper threshold (T2): **200**

Apply thresholding to create a binary mask:

For pixels with intensity values between T1 and T2, set the corresponding pixel value in the mask to 1 (white).

For pixels with intensity values outside the threshold range, set the corresponding pixel value in the mask to 0 (black).

Intensity Level Slicing Transformations

Thresholding Example:

Original Image: [25, 50, 75, 100, 125, 150, 175, 200, 225, 250]

Mask: [0, 0, 0, 1, 1, 1, 1, 1, 0, 0]

Resulting Image: [0, 0, 0, 100, 125, 150, 175, 200, 0, 0]



Intensity Level Slicing Transformations

Binary Image Creation Example:

Let's consider a grayscale image represented by the following pixel values (ranging from 0 to 255)

[25, 50, 75, 100, 125, 150, 175, 200, 225, 250]

Now, let's apply intensity level slicing transformation using thresholding to highlight pixels with intensity values between 100 and 200.

Define the threshold range:

- Lower threshold (T1): **100**
- Upper threshold (T2): **200**

Apply thresholding to create a binary mask:

For pixels with intensity values between T1 and T2, set the corresponding pixel value in the mask to 1 (white).

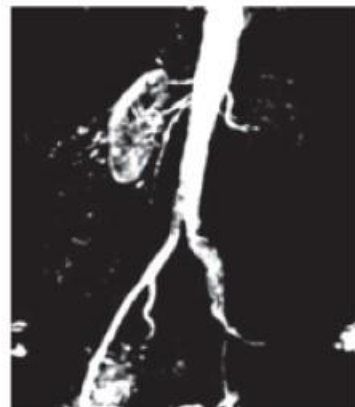
For pixels with intensity values outside the threshold range, set the corresponding pixel value in the mask to 0 (black).

Intensity Level Slicing Transformations

Binary Image Creation Example :

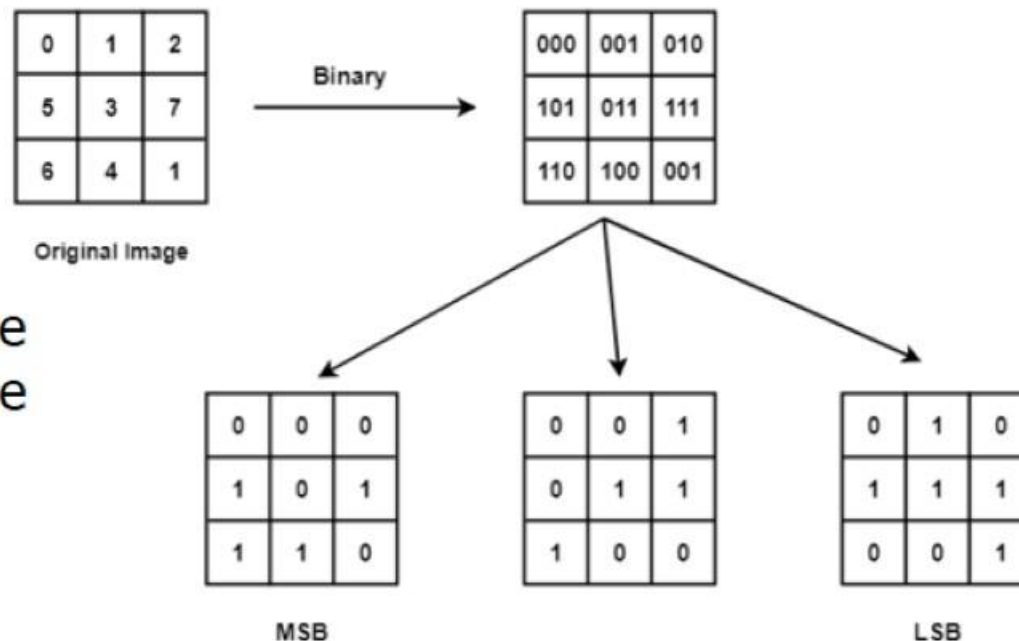
Original Image: [25, 50, 75, 100, 125, 150, 175, 200, 225, 250]

Binary Image : [0, 0, 0, 1, 1, 1, 1, 1, 0, 0]



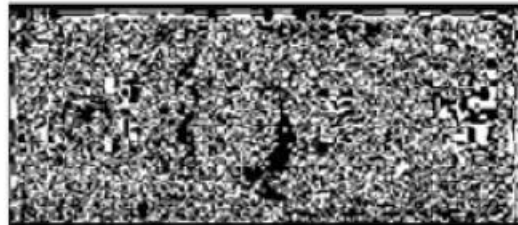
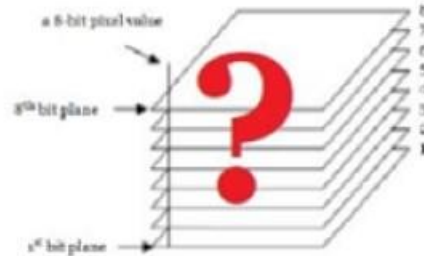
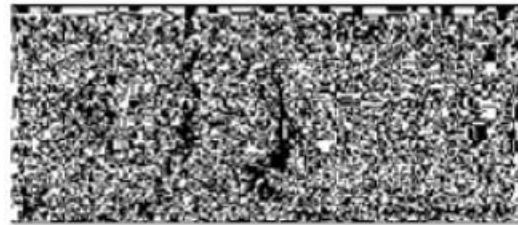
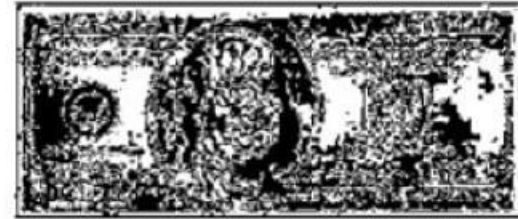
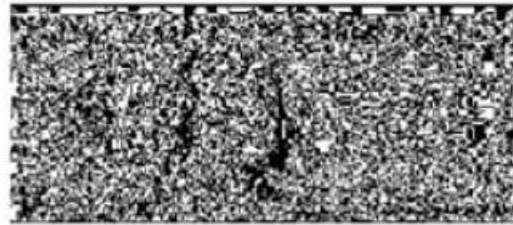
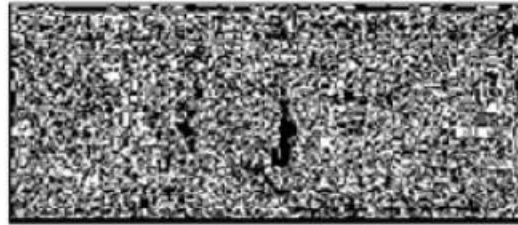
Bit-Plane Slicing Transformations

In Bit-plane slicing, we divide the image into bit planes. This is done by first converting the pixel values in the binary form and then dividing it into bit planes.



The higher-order bits (especially the top four) contain the majority of the visually significant data.

Bit-Plane Slicing Transformations



Bit-Plane Slicing Transformations

Practice:

Consider a grayscale image represented by the following gray values (each value is an 8-bit binary number):

4	5	7
0	2	3
2	3	1

Visualize the bit-planes for the given image

Bit-Plane Slicing Transformations

Practice:

Consider a grayscale image represented by the following gray values (each value is an 8-bit binary number):

100	101	111
000	010	011
010	011	001

Step 1: Converted to Binary digits

Bit-Plane Slicing Transformations

Practice:

Consider a grayscale image represented by the following gray values (each value is an 8-bit binary number):

100	101	111
000	010	011
010	011	001

MSB

Mid

LSB

1	1	1
0	0	0
0	0	0

0	0	1
0	1	1
1	1	0

0	1	1
0	0	1
0	1	1

Step 2: Bit-wise slicing

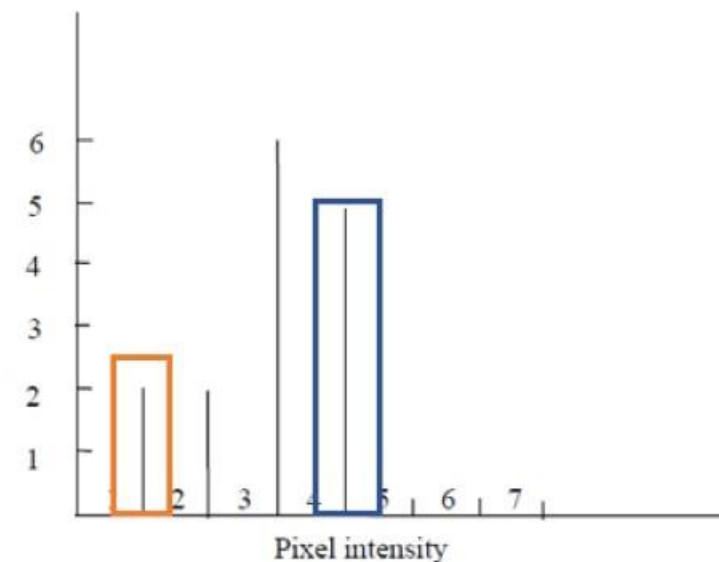
Histogram Processing

The image histogram is a valuable tool used to view the intensity profile of an image.

- The histogram provides information about the contrast and overall intensity distribution of an image.
- The image histogram is simply a bar graph of the pixel intensities. The pixel intensities are plotted along the x-axis and the number of occurrences for each intensity represents the y-axis.

4	4	3	3
4	4	3	3
4	1	2	3
0	1	2	3

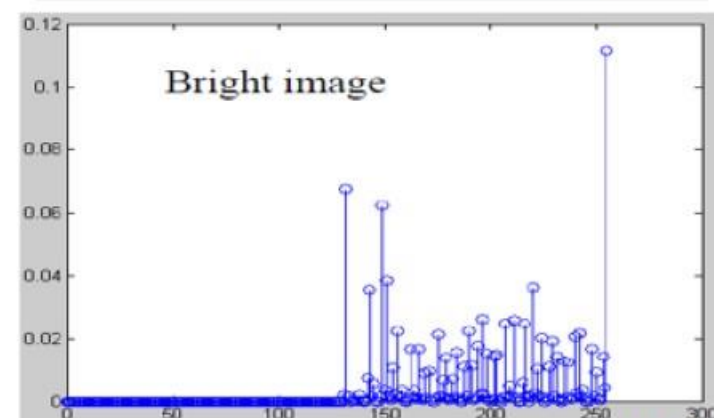
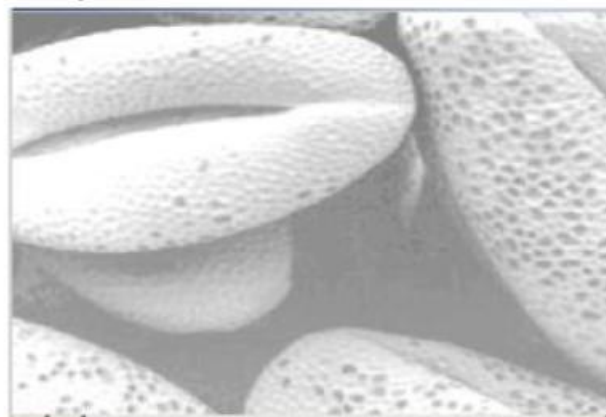
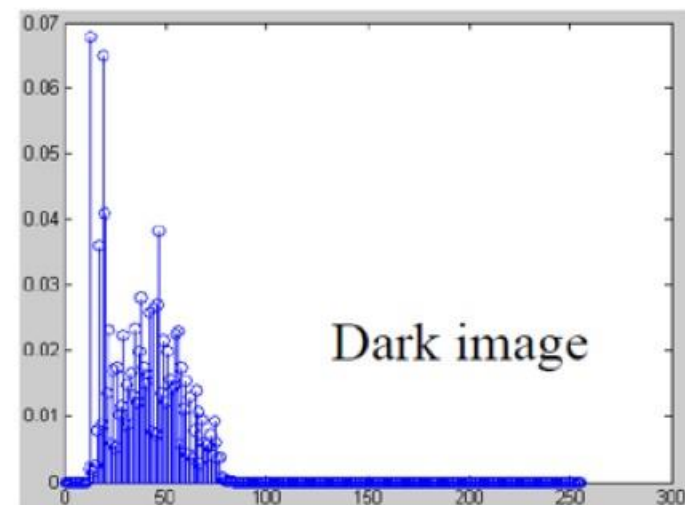
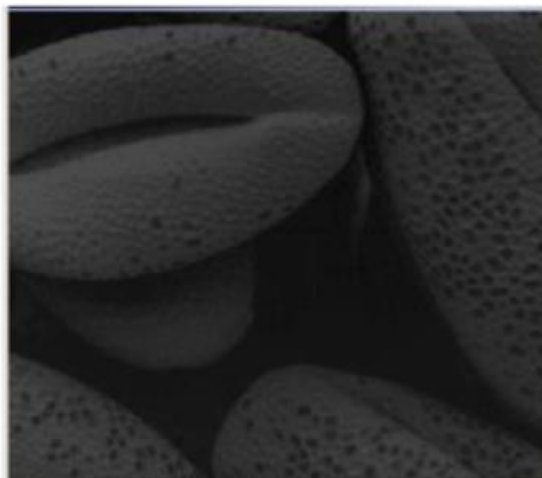
Image



Let's try:

<https://www.dcode.fr/image-histogram>

Histogram Processing



Histogram Equalization

– To improve the contrast of an image

– To transform an image in such a way that the transformed image has a nearly uniform distribution of pixel values

The intensity levels in an image may be viewed as random variables in the interval $[0, L-1]$.

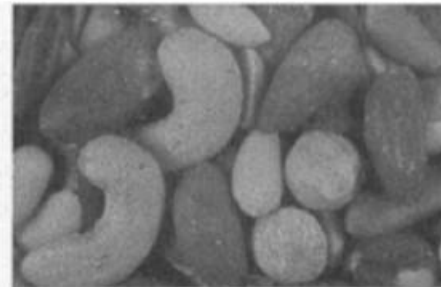
The probability of occurrence of intensity level r_k

$$p_r(r_k) = \frac{n_k}{MN}$$

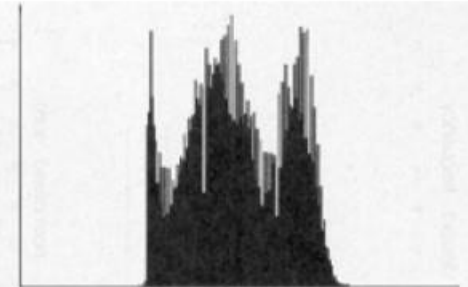
Normalized image histogram

$$s_k = T(r_k) = (L-1) \sum_{j=0}^k p_r(r_j)$$

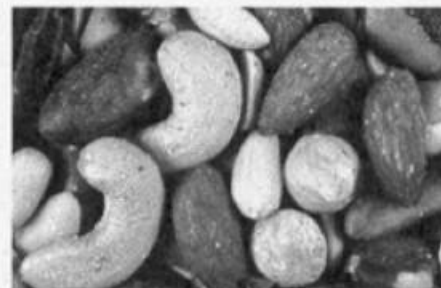
$$k = 0, 1, 2, \dots, L-1$$



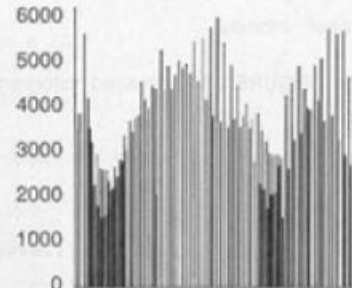
(a)



(b)



(c)



(d)

Histogram Equalization

The process involves the following steps:

Compute the Histogram: Calculate the histogram of the input image. The histogram is a plot or graph that shows the frequency of each pixel intensity value in the image. It provides a visual representation of the distribution of pixel values.

Normalize the Histogram: Normalize the histogram so that it represents the probability density function (PDF) of the image. This is done by dividing each bin (intensity value) in the histogram by the total number of pixels in the image.

Cumulative Distribution Function (CDF): Compute the cumulative distribution function (CDF) of the normalized histogram. The CDF represents the cumulative sum of the probabilities of each intensity value and ranges from 0 to 1.

Histogram Equalization

The process involves the following steps:

Histogram Equalization: For each pixel in the original image, replace its intensity value with a new value obtained by mapping the original intensity to its corresponding CDF value. This mapping can be done using the following formula:

$$\text{new_intensity} = \text{CDF}(\text{original_intensity}) * (L-1)$$

Create the Equalized Image: Replace each pixel's intensity in the original image with the new intensity calculated in the previous step.

Histogram Equalization

4	1	3	2
3	1	1	1
0	1	5	2
1	1	2	2

input image

5	3	4	4
4	3	3	3
1	3	5	4
3	3	4	4

output image

Gray Level	freq	PDF	CDF	CDF * (Levels-1)	New Gray Level Value
0	1	$1/16=0.06$	0.06	$0.06*5= 0.3$	1
1	7	$7/16=0.44$	0.5	$0.5*5= 2.5$	3
2	4	$4/16=0.25$	0.75	$0.75*5= 3.75$	4
3	2	$2/16=0.13$	0.88	$0.88*5= 4.4$	4
4	1	$1/16=0.06$	0.94	$0.94*5= 4.7$	5
5	1	$1/16=0.06$	1	$1*5= 5$	5
Total	16				

Histogram Equalization

Assignment

10	2	11	13	14
11	13	14	13	13
10	11	13	11	14
13	11	14	12	10
10	14	12	14	14

Normalize the following gray level image



Thank You