

Digital Image Processing

المحاضرة السابعة
Image Enhancement
العمليات على البكسلات باستخدام
هستوغرام الصورة

د. عيسى الغنام
2025 الفصل الصيفي

- ▶ The shape of the histogram of an image does provide useful info about the possibility for contrast enhancement.

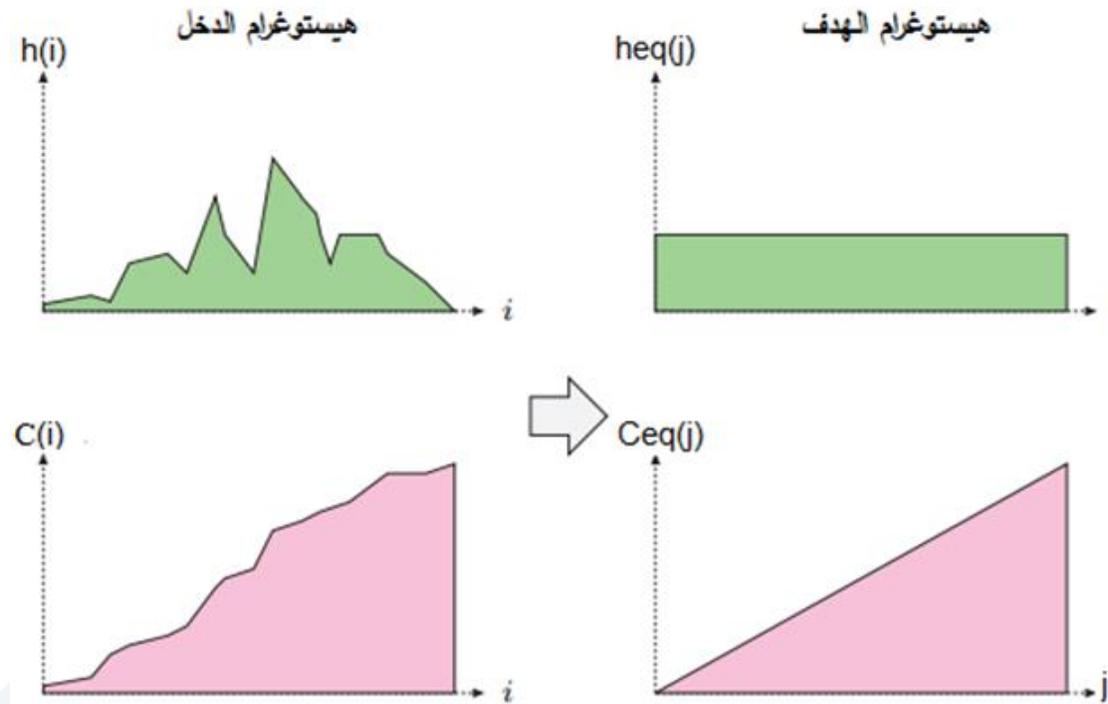
- فيما يلي بعض المواضيع التي تندرج تحت معالجة الرسم البياني:
 - مساواة الرسم البياني Histogram Equalization
 - مطابقة الرسم البياني Histogram Matching

تسوية الهستوغرام

□ تحويل التوزيع غير الخطي والفريد لبكسلات صورة الدخل إلى صورة خرج ذات هستوغرام ذي توزيع احتمالي موحد

□ تحتوي الصورة الناتجة عن عملية تسوية الهستوغرام في الحالة المثالية على عدد متساو من البكسلات التي تمثل كل سوية

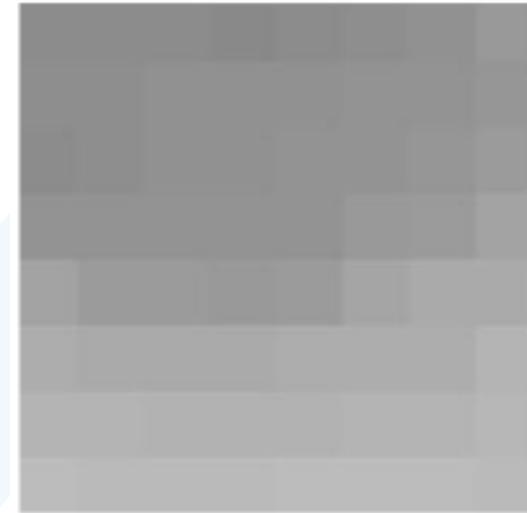
لونية في الصورة



مثال 1: عملية تسوية الهستوغرام

8 bits

140	140	140	137	140	142	145	153
142	142	145	145	145	147	147	150
140	142	145	145	147	147	150	155
147	147	147	147	147	153	155	163
163	155	155	153	155	165	170	170
173	170	170	170	173	173	173	181
181	181	178	178	178	181	181	183
188	186	186	186	188	188	188	186



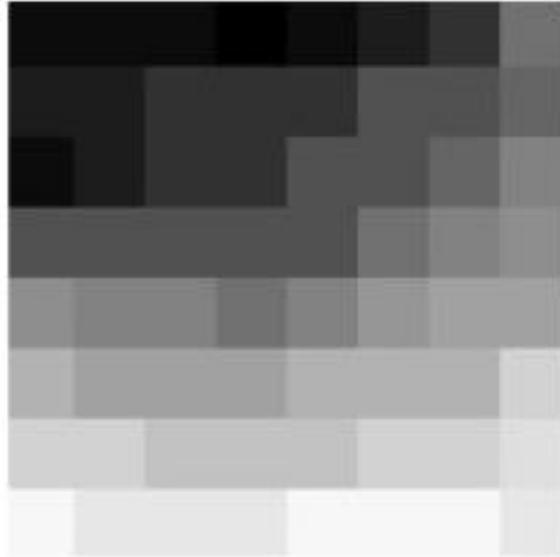
مصفوفة الصورة



مثال 1: عملية تسوية الهيستوغرام

القيمة	التكرار #	التوزيع التراكمي C	C/N	(L-1)*C/N floor	(L-1)*(C - Cmin)/(N - Cmin) floor
137	1	1	1/64	3	0
140	5	6	6/64	23	20
142	4	10	10/64	39	36
145	6	16	16/64	63	60
147	9	25	25/64	100	97
150	2	27	27/64	108	105
153	3	30	30/64	120	117
155	5	35	35/64	139	137
163	2	37	37/64	147	145
165	1	38	38/64	151	149
170	5	43	43/64	171	170
173	4	47	47/64	187	186
178	3	50	50/64	199	198
181	5	55	55/64	219	218
183	1	56	56/64	223	222
186	4	60	60/64	239	238
188	4	64	64/64	255	255

مثال 1: عملية تسوية الهيستوغرام



المصفوفة الناتجة عن تسوية الهيستوغرام



مصفوفة الصورة

تمرين

□ بفرض الصورة المبينة والمخزنة بعمق بكسل قدره 4

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

L=16

■ ما هي صورة الخرج الناتجة عن تسوية الهيستوغرام

i	#	C	C/N	(L-1)*C/N	floor	(L-1)*(C - Cmin)/(N - Cmin) floor
0	2	2	2/25	1.2	1	$15*(2-2)/(25-2)=0$
1	4	6	6/25	3.6	3	$15*(6-2)/(25-2)=2.6=2$
2	5	11	11/25	6.6	6	
3	2	13	13/25	7.8	7	
4	3	16	16/25	9.6	9	
5	3	19	19/25	11.4	11	
6	3	22	22/25	13.2	13	
7	3	25	25/25	15	15	

				0
			2	2
	0			2
			2	

صورة الخرج

Intensity	# pixels	c
0	20	20
1	5	25
2	25	50
3	10	60
4	15	75
5	5	80
6	10	90
7	10	100
Total	100	

Accumulative Sum of $P_r=C/N$
$20/100 = 0.2$
$(20+5)/100 = 0.25$
$(20+5+25)/100 = 0.5$
$(20+5+25+10)/100 = 0.6$
$(20+5+25+10+15)/100 = 0.75$
$(20+5+25+10+15+5)/100 = 0.8$
$(20+5+25+10+15+5+10)/100 = 0.9$
$(20+5+25+10+15+5+10+10)/100 = 1.0$
1.0



جامعة
المنارة

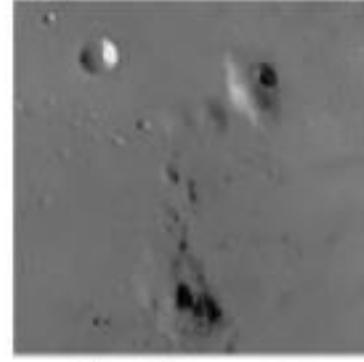
تمرين

Intensity (r)	No. of Pixels (n_j)	Acc Sum of $C/N=P_r$	Output value: $(L-1)C/N$	Quantized Output (s) round	$(L-1)*(C - C_{min})/(N - C_{min})$
0	20	0.2	$0.2 \times 7 = 1.4$	1	
1	5	0.25	$0.25 * 7 = 1.75$	2	
2	25	0.5	$0.5 * 7 = 3.5$	4	
3	10	0.6	$0.6 * 7 = 4.2$	4	
4	15	0.75	$0.75 * 7 = 5.25$	5	
5	5	0.8	$0.8 * 7 = 5.6$	6	
6	10	0.9	$0.9 * 7 = 6.3$	6	
7	10	1.0	$1.0 \times 7 = 7$	7	
Total	100				

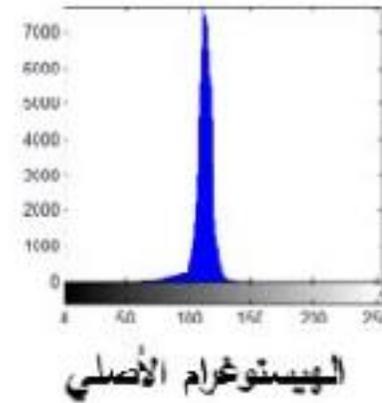
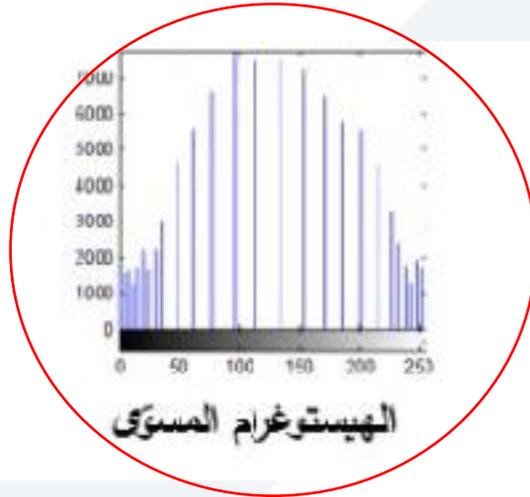
تطبيق عملية تسوية الهستوغرام



الصورة الناتجة عن تسوية الهستوغرام

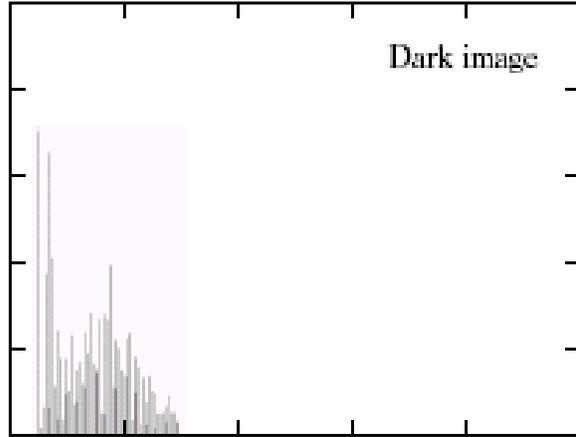
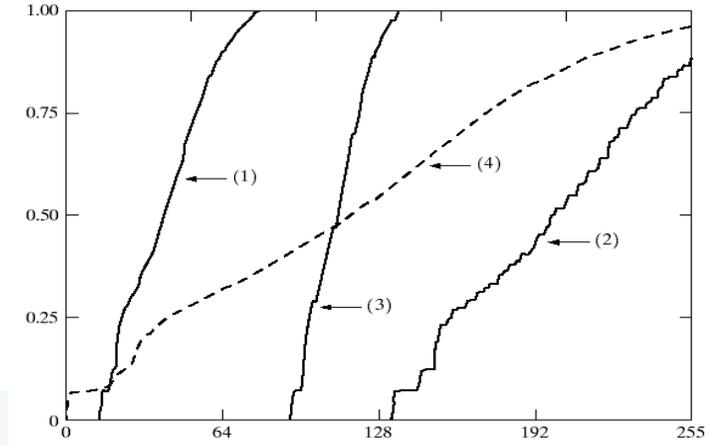
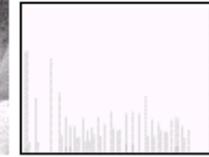
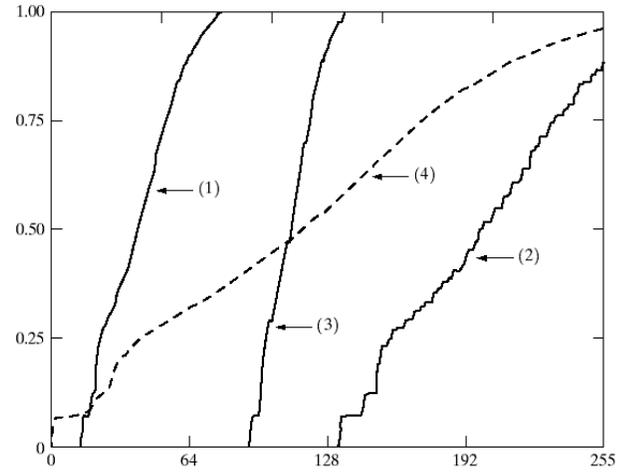


الصورة الأصلية

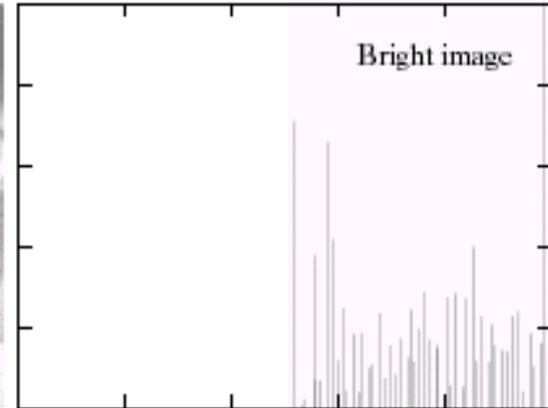




جامعة
النَّارِ
NARA UNIVERSITY

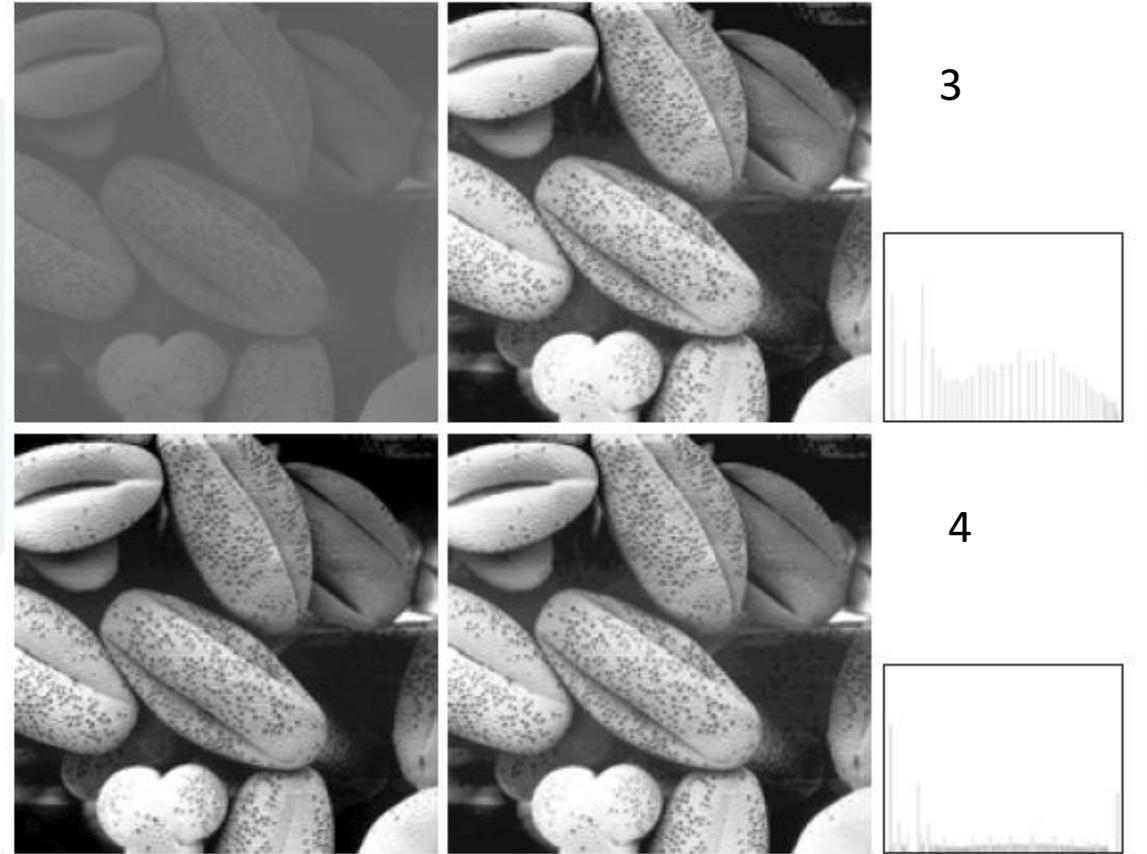
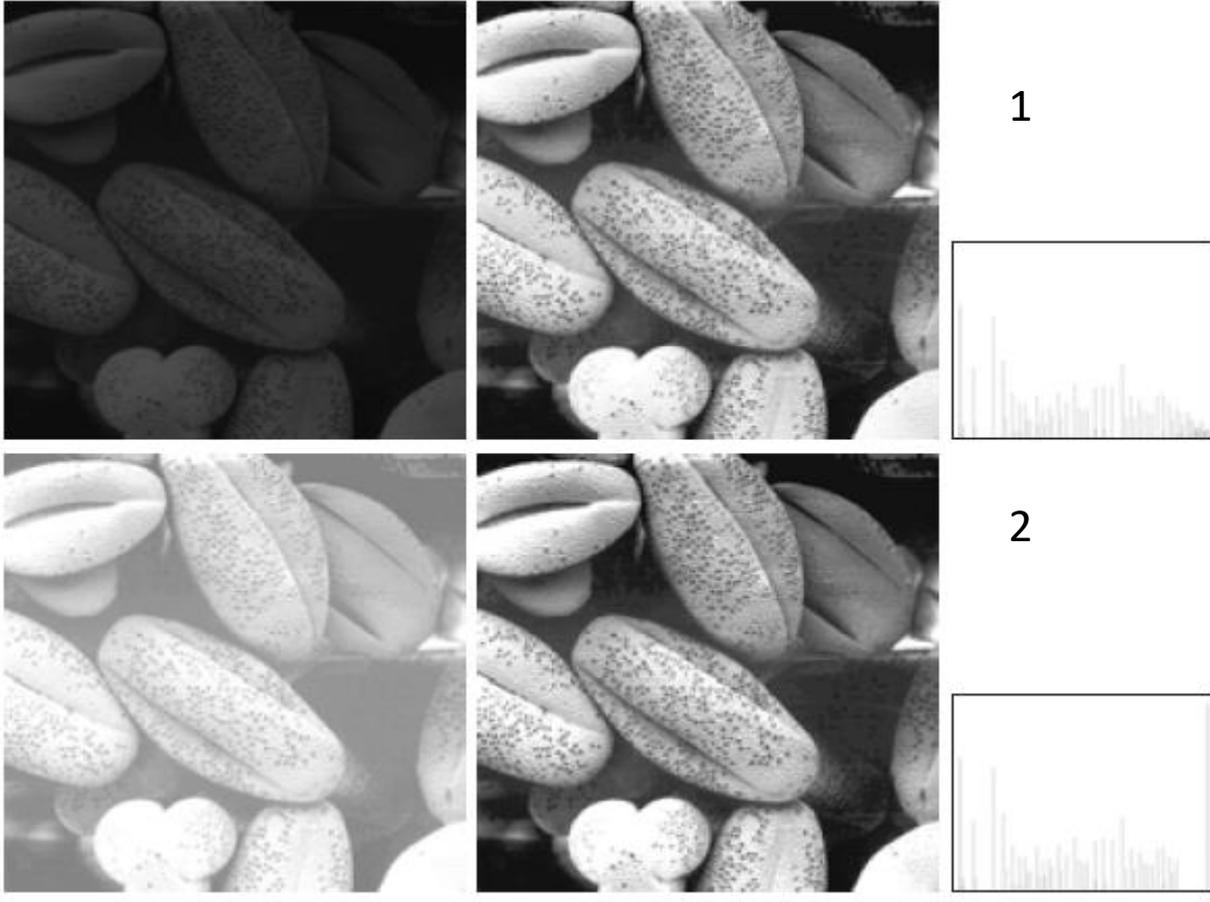


1



2

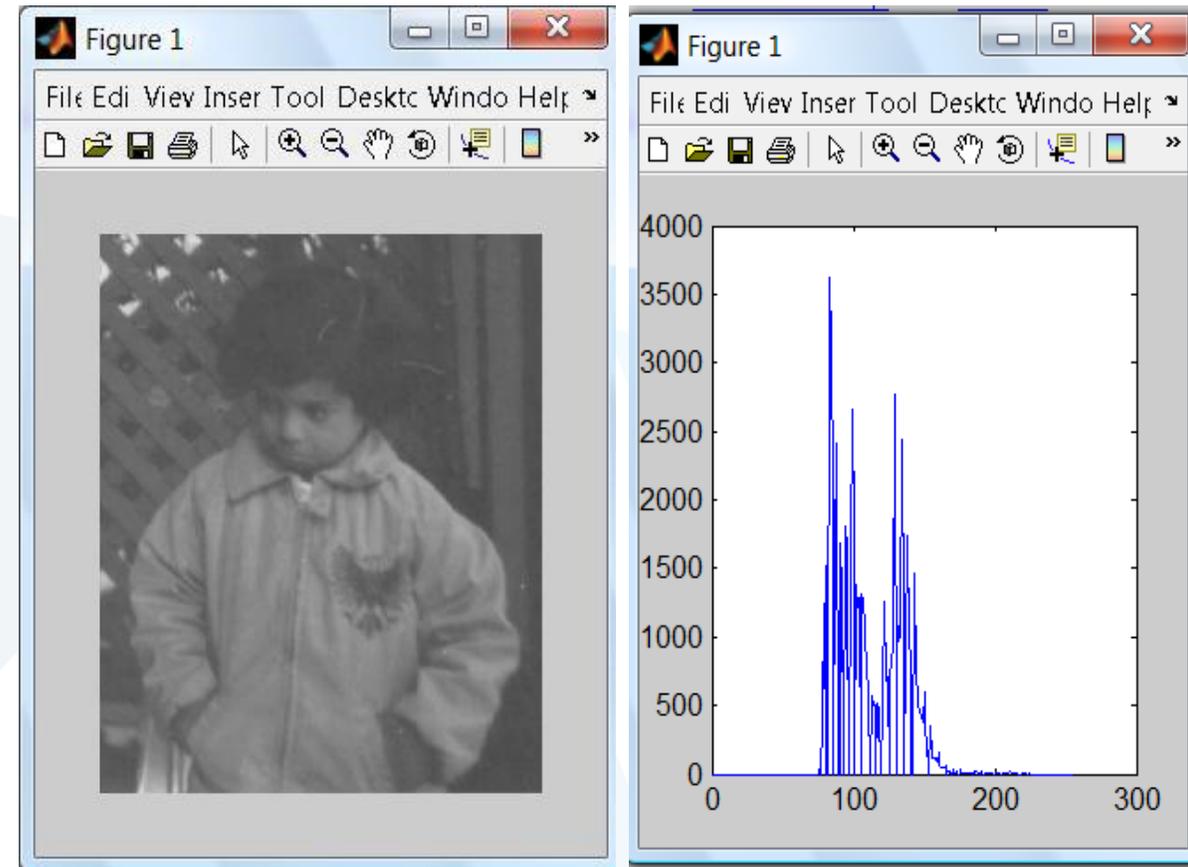
Histogram Equalization مساواة الرسم البياني



the image after histogram equalization becomes a low contrast image.

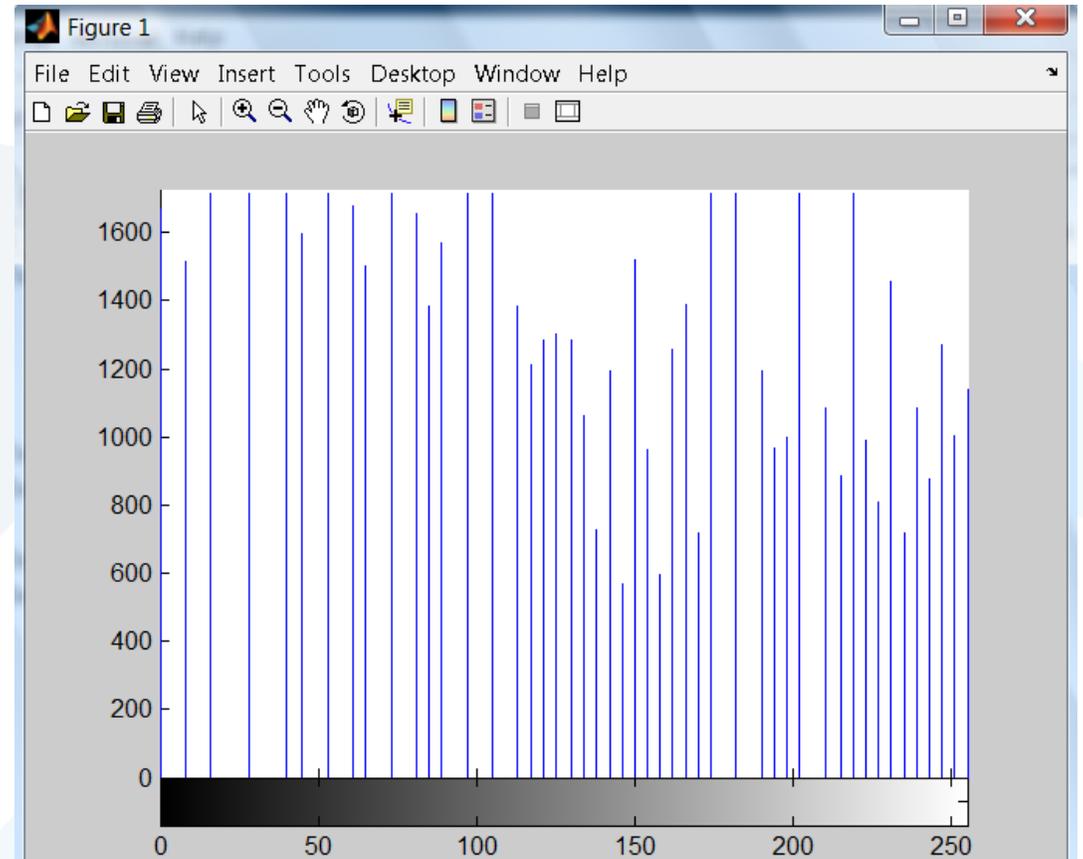
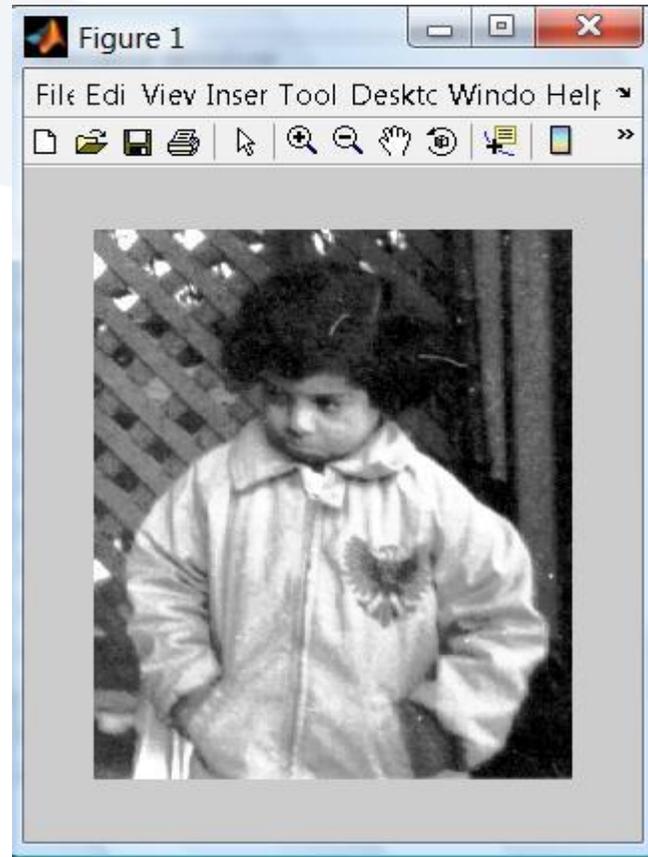
Histogram equalization of the image in MATLAB

We have this image in matlab called pout.tif, when we plot its histogram it is showed like this:



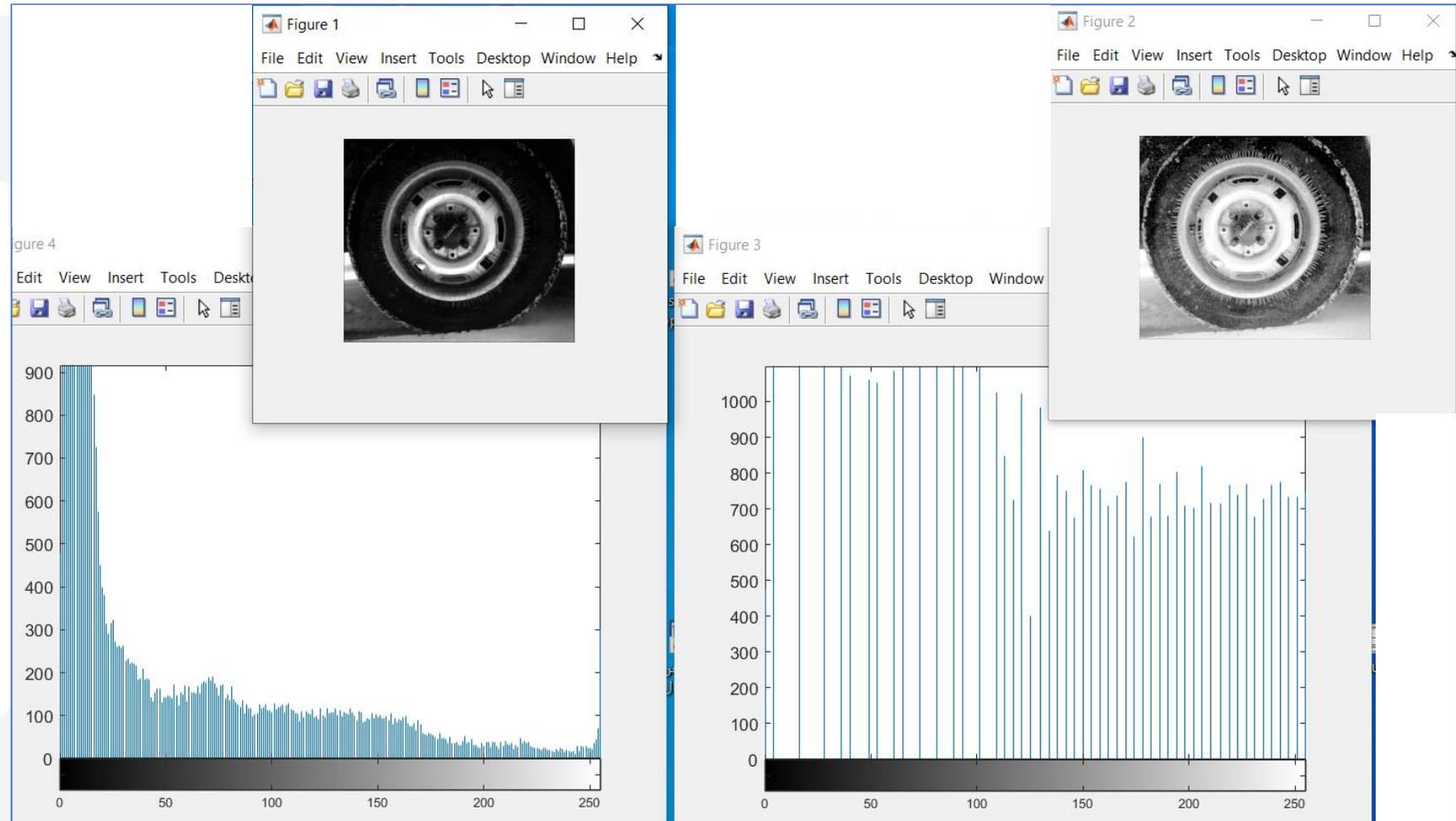
Notice that the pixels intensity values are concentrated on the middle (low contrast)

histogram equalization :
is the process of adjusting intensity values of pixels. The process which increases the dynamic range of the gray level in a low contrast image to cover full range of gray levels. In matlab : we use **histeq** function



Histogram produces pixels having values that are distributed throughout the range

```
clc  
clear  
I = imread('tire.tif');  
imshow(I)  
J = histeq(I);  
figure,imshow(J)  
figure, imhist(J)  
figure, imhist(I)
```



```
import cv2
import numpy as np
import matplotlib.pyplot as plt
image_path = 'noisy1.png'
img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
if img is None:
    print(f"Error: Could not load image from {image_path}. Please check the path and file existence.")
    exit()

# --- 3. Perform Histogram Equalization ---
# cv2.equalizeHist() takes a single-channel (grayscale) image and returns the equalized image.
equalized_img = cv2.equalizeHist(img)

# --- 4. (Optional) Calculate Histograms for comparison ---
hist_original = cv2.calcHist([img], [0], None, [256], [0, 256])
hist_equalized = cv2.calcHist([equalized_img], [0], None, [256], [0, 256])
```

```
# --- 5. Display the original and equalized images with  
their histograms ---
```

```
plt.figure(figsize=(12, 8)) # Adjust figure size
```

```
# Subplot 1: Original Grayscale Image
```

```
plt.subplot(2, 2, 1) # 2 rows, 2 columns, first plot
```

```
plt.imshow(img, cmap='gray')
```

```
plt.title('Original Grayscale Image')
```

```
plt.axis('off')
```

```
# Subplot 2: Equalized Grayscale Image
```

```
plt.subplot(2, 2, 2) # 2 rows, 2 columns, second plot
```

```
plt.imshow(equalized_img, cmap='gray')
```

```
plt.title('Equalized Grayscale Image')
```

```
plt.axis('off')
```

```
# Subplot 3: Original Histogram
```

```
plt.subplot(2, 2, 3) # 2 rows, 2 columns, third plot
```

```
plt.plot(hist_original, color='blue')
```

```
plt.title('Original Histogram')
```

```
plt.xlabel('Pixel Intensity')
```

```
plt.ylabel('Number of Pixels')
```

```
plt.xlim([0, 256])
```

```
# Subplot 4: Equalized Histogram
```

```
plt.subplot(2, 2, 4) # 2 rows, 2 columns, fourth plot
```

```
plt.plot(hist_equalized, color='red')
```

```
plt.title('Equalized Histogram')
```

```
plt.xlabel('Pixel Intensity')
```

```
plt.ylabel('Number of Pixels')
```

```
plt.xlim([0, 256])
```

```
plt.tight_layout() # Adjust layout to prevent overlapping
```

```
plt.show()
```



Figure 1

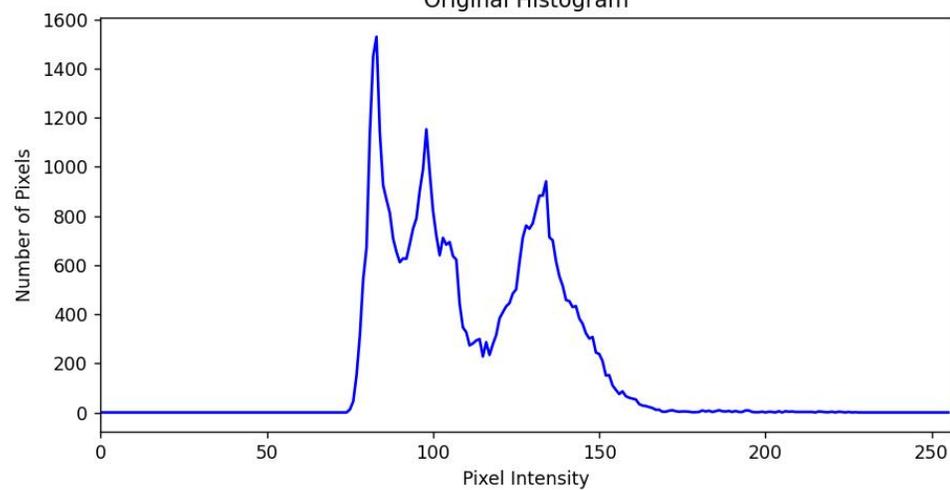
Original Grayscale Image



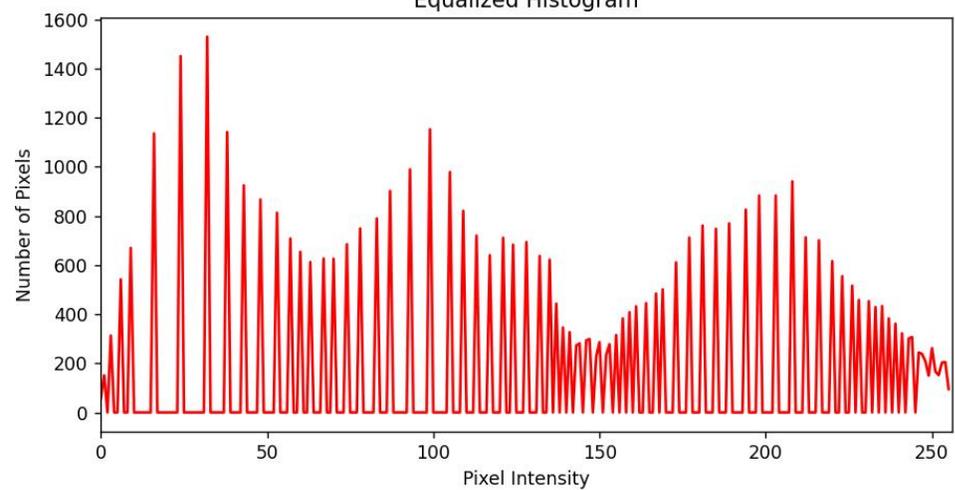
Equalized Grayscale Image



Original Histogram

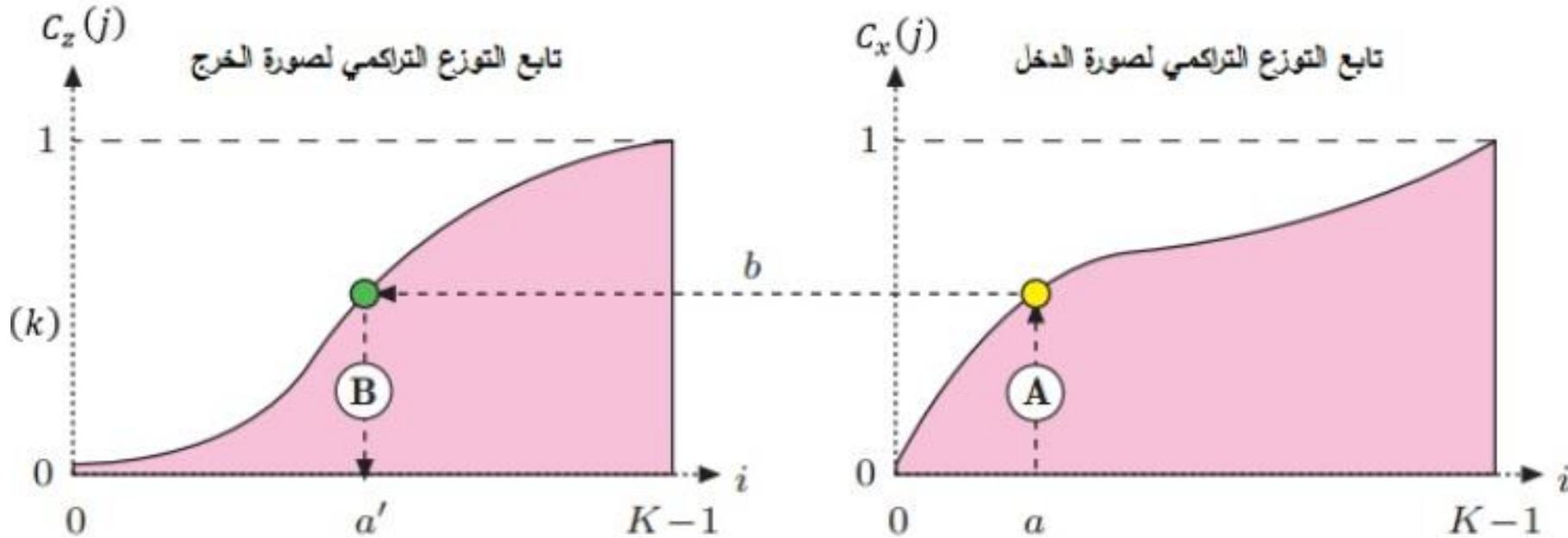


Equalized Histogram



Making the histogram look like a defined graph

□ هي عملية تعميم تسوية الهستوغرام وتتطلب تحديد شكل الهستوغرام المطلوب





مثال 2

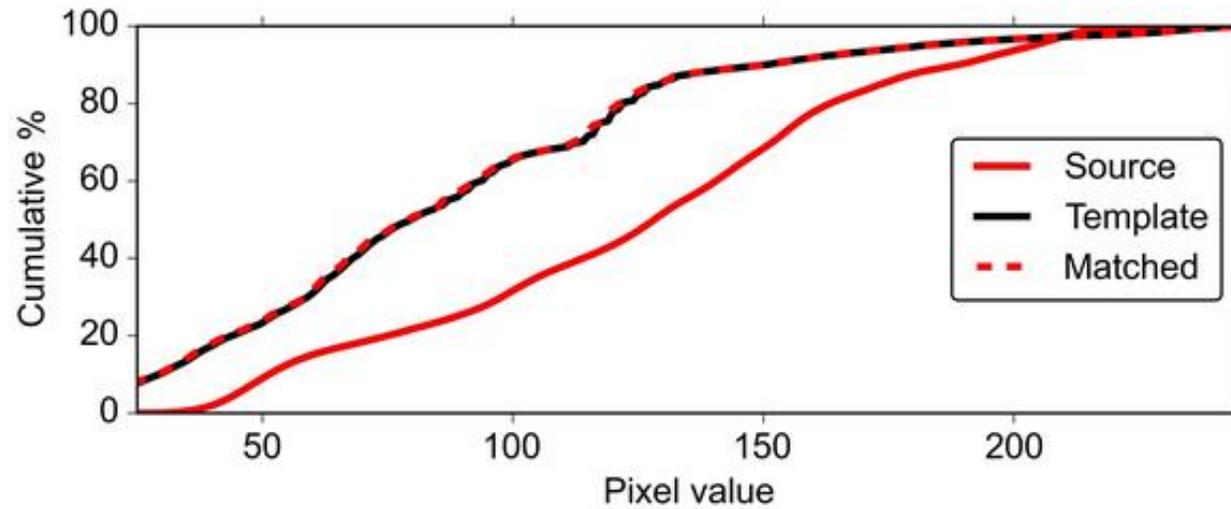
Source



Template

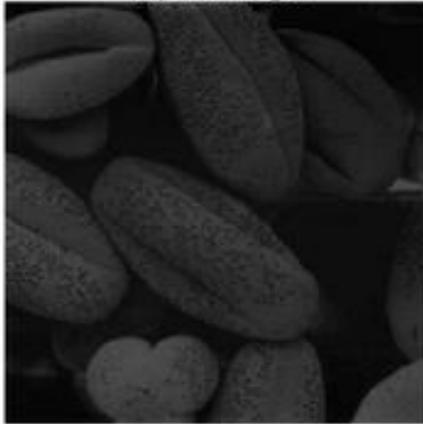


Matched



مثال 3

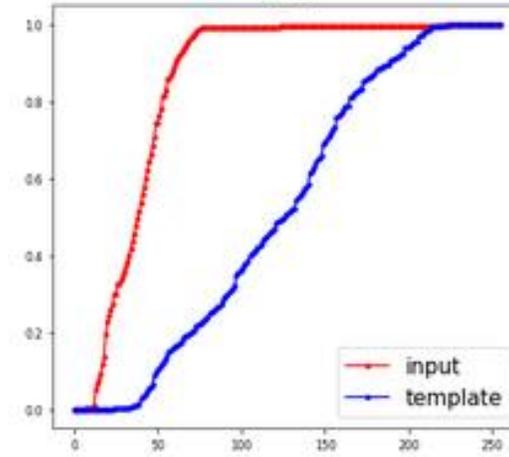
Input image



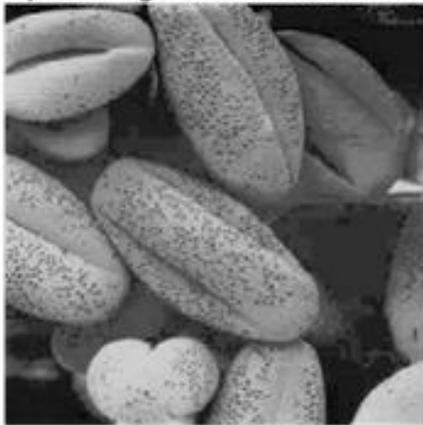
Template image



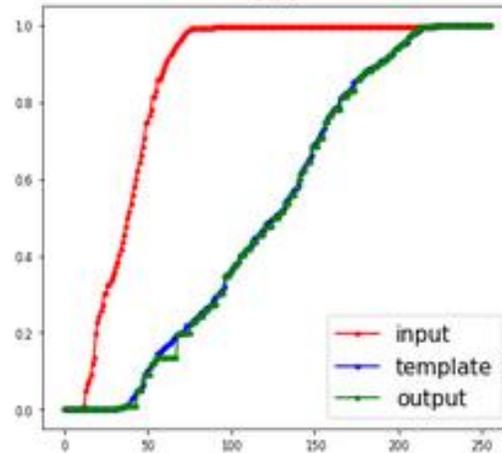
CDF



Output image with Hist. Matching



CDF



Histogram Matching Example



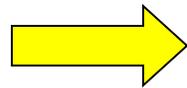
جامعة
المنارة
MANARA UNIVERSITY

Example
problem

The histogram
of the input
image

Want the histogram
of the output image to
be as follows

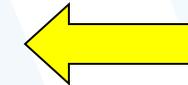
Initial data



Intensity (s)	# pixels
0	20
1	5
2	25
3	10
4	15
5	5
6	10
7	10
Total	100

Intensity (z)	# pixels
0	5
1	10
2	15
3	20
4	20
5	15
6	10
7	5
Total	100

User sets



Histogram Matching Example

1. Perform both histogram equalization tables

r	(n _j)	ΣP _r	s
0	20	0.2	1
1	5	0.25	2
2	25	0.5	4
3	10	0.6	4
4	15	0.75	5
5	5	0.8	6
6	10	0.9	6
7	10	1.0	7

$$s_k = T(r_k)$$

z	(n _j)	ΣP _z	v
0	5	0.05	0
1	10	0.15	1
2	15	0.3	2
3	20	0.5	4
4	20	0.7	5
5	15	0.85	6
6	10	0.95	7
7	5	1.0	7

$$v_k = G(z_k)$$

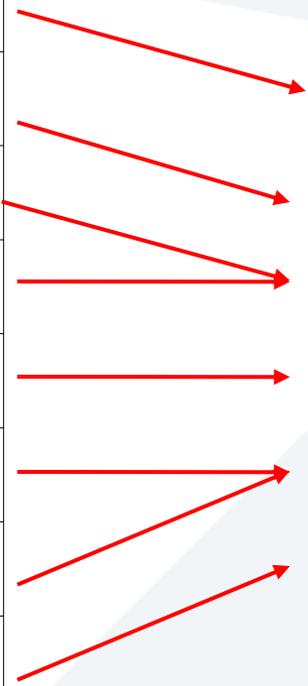
Histogram Matching Example

2. Get Table Map

r → s

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

s → v



v → z

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

Get to be

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

Actual Output Histogram

z	# Pixels
0	0
1	20
2	5
3	35
4	15
5	15
6	10
7	0

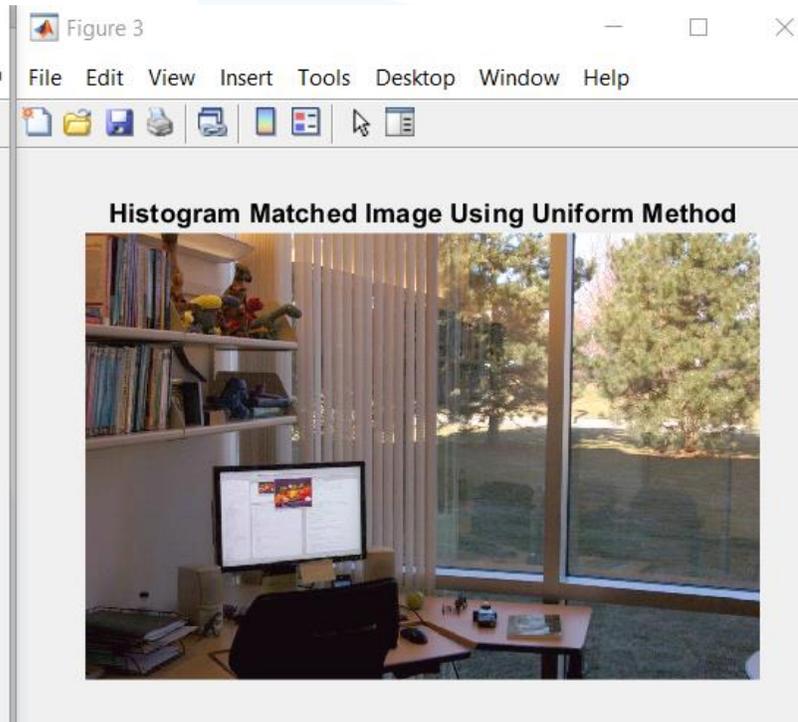
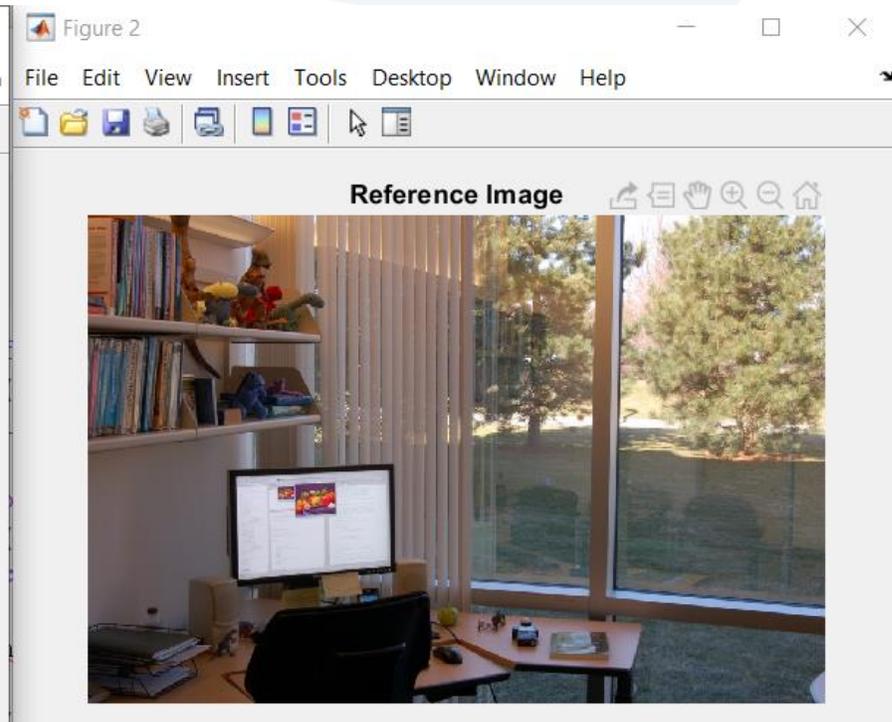
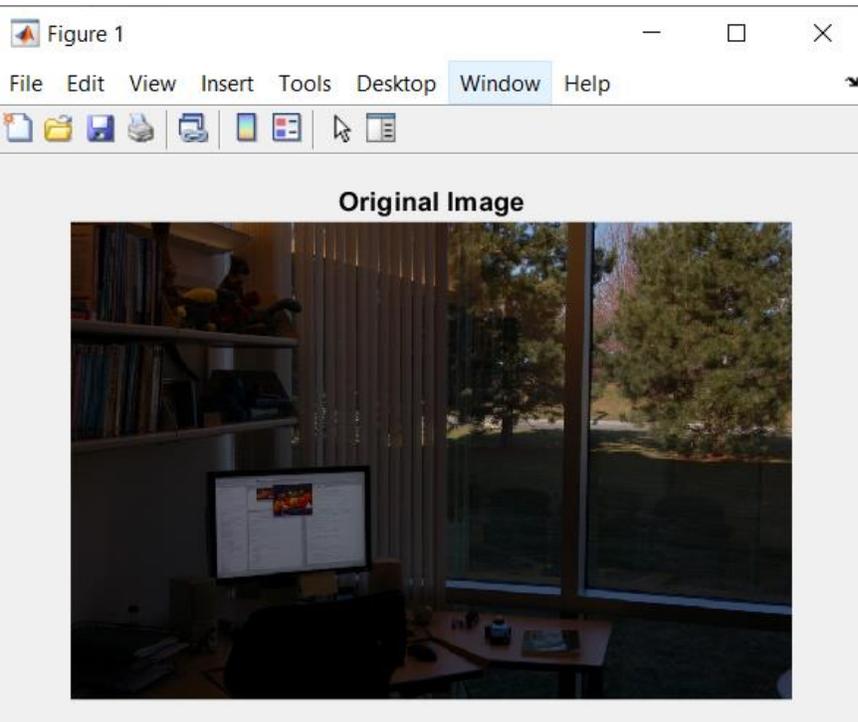
$$s_k = T(r_k)$$

$$z_k = G^{-1}(v_k)$$

```
close all;clc;clear  
A = imread('office_2.jpg');  
figure, imshow(A, []);  
title('Original Image');  
ref = imread('office_4.jpg');  
figure, imshow(ref, []);  
title('Reference Image');
```

```
B = imhistmatch(A, ref);
```

```
figure, imshow(B, []);  
title('Histogram Matched  
Image Using Uniform Method');
```





جامعة
المنارة
MANARA UNIVERSITY

```
import cv2
import matplotlib.pyplot as plt
from skimage.exposure import match_histograms
import numpy as np

# Read the source image
A = cv2.imread('1.png')
A = cv2.cvtColor(A, cv2.COLOR_BGR2RGB) # Convert from BGR to RGB

# Display the original image
plt.figure()
plt.imshow(A)
plt.title('original image')
plt.axis('off')

# Read the reference image
ref = cv2.imread('2.png')
ref = cv2.cvtColor(ref, cv2.COLOR_BGR2RGB) # Convert from BGR to RGB

# Display the reference image
plt.figure()
plt.imshow(ref)
plt.title('Reference Image')
plt.axis('off')
```

```
# Perform histogram matching
B = match_histograms(A, ref)
image_clipped = np.clip(B, 0, 255).astype(np.uint8)
```

```
#B = cv2.cvtColor(B, cv2.COLOR_BGR2RGB) # Convert from BGR to RGB
```

```
# Display the histogram matched image
plt.figure()
plt.imshow(image_clipped)
plt.title('Histogram Matched Image Using Uniform Method')
plt.axis('off')
```

```
# Show all figures
plt.show()
```

original image



Reference Image



Histogram Matched Image Using Uniform Method



نهاية المحاضرة