جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

قسم الانظمة الطبية الذكية

المرحلة الثالثة

Subject : Medical image processing

University of Information Technology and Communications
College of Medical Informatics
Intelligent Medical Systems Department

Class 7 : Medical image preprocessing algorithms

By : Dr. Ansam Ali Abdulhussein

Contrast Enhancement

3. Histogram modeling

- Redistributes pixel intensity values to enhance contrast. It is useful for images with poor visibility.

Histogram equalization (HEs):

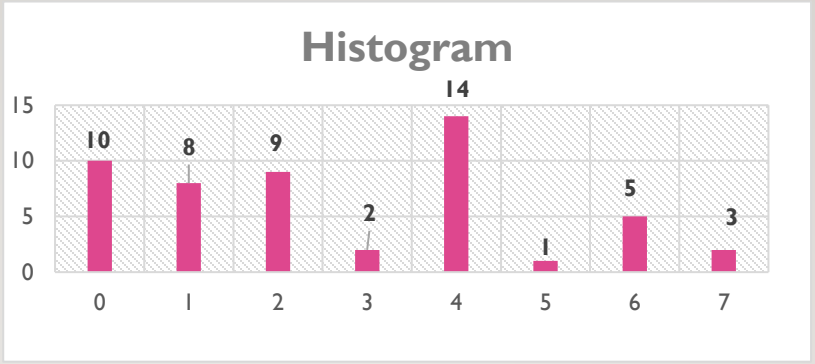
is a technique where the histogram of the resultant image is as flat as possible (with histogram stretching the overall shape of the histogram remains the same). It Is a popular technique used for improving the appearance of a poor image. It's a function is similar to that of a histogram stretch but often provides more visually pleasing results across a wide range of images

The histogram equalization process for digital images consists of four steps:

1. **Compute the Histogram**: Calculate the histogram of the input image, which shows the frequency of each intensity level (e.g., 0–255 for an 8-bit grayscale image).
2. **Compute the Cumulative Distribution Function (CDF):** Calculate the cumulative distribution function (CDF) of the histogram. This function accumulates the histogram values and is used to map the original intensity values to new values.
3. **Normalize the CDF**:
   Normalize the CDF so that its values range from 0 to the maximum intensity level (e.g., 255). This creates a mapping function that redistributes the intensity values.
4. **Map Original Intensities to Equalized Intensities**:
   Use the normalized CDF as a transformation function to map each pixel's original intensity to a new intensity. This results in an image with a more uniform histogram and improved contrast.

Dr. Ansam Ali Abdulhussein

Example :



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

k	0	1	2	3	4	5	6	7
1) nk	10	8	9	2	14	1	5	3
2) Sum(kn)	10	18	27	29	43	44	49	53
3)CDF (Sum(kn)/n=53)	10/531=0.188	18/53=0.679	27/53=0.509	29/53= 0.547	43/53=0.8	44/53=0.83	49/53=0.92	53/53=1
Normalize= CDF *L-1=7	1.316	4.753	3.563	3.892	5.6	5.81	6.44	7
Round equ.	1	5	4	4	6	6	7	7

4- **Adaptive Histogram Equalization (AHE):**

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

(AHE) is an advanced version of standard histogram equalization. Unlike global histogram equalization (which works on the entire image), AHE improves local contrast by applying histogram equalization to small regions of the image.

Here's how Adaptive Histogram Equalization works:

1. Divide the image into overlapping regions or tiles (segment).
2. Compute the histogram for each region.
3. Calculate the transformation function for each point using neighboring pixels
4. The result is then merged to create an image with enhanced contrast in localized areas.

Advantages:

•Improves contrast in specific regions of the image.
•Useful in images with varying lighting conditions or low contrast.

Disadvantages:

•Can lead to increased noise or artifacts in some regions of the image, especially in areas with homogeneous content.

Key Difference from Standard Histogram Equalization:

Feature	Histogram Equalization	Adaptive Histogram Equalization
Works on whole image	✅ Yes	❌ No (works on small regions)
Enhances local contrast	❌ No	✅ Yes
Risk of over-enhancing noise	❌ Less	✅ More (especially without CLAHE)
Used in medical, aerial images	❌ Less common	✅ More common

Two pixels with intensity 80 in the central region.

•**GridSize**: Divide the 8×8 image into 5×5 grids, resulting in small sub-grids

Example :

Two pixels with intensity 80 in the central region.
•**GridSize**: Divide the 8×8 image into 5×5 grids, resulting in small sub-grids

20	20	20	20	20
20	80	80	80	81
20	80	80	80	80
20	80	80	80	81
20	80	81	82	80

20	20	20	20	20	20	20	0
20	20	20	20	20	20	20	20
20	20	80	80	80	81	80	81
20	20	80	80	80	80	82	80
20	20	80	80	80	81	82	82
20	20	80	81	82	80	80	82
20	20	80	82	81	81	80	80
20	20	80	82	81	81	80	200

Step 1: Calculate Local Histograms
For simplicity, assume the histogram of the grid containing the two 80 intensity pixels is:

Intensity	count
20	9
80	12
81	3
82	1

Step 2: Compute the Cumulative Distribution Function (CDF)**:** Divide the counts by the total number of pixels in the grid (4):

Intensity	Frequency	CDF
20	9	9/25=2.25
80	12	12/25=3
81	3	3/25= 0.75
82	1	1/25= 0.25
	sum:= 25	

Step 3 : Normalize CDF where L-1=255

91.8	91.8	91.8	91.8	91.8
91.8	214.2	214.2	214.2	244.8
91.8	214.2	214.2	214.2	214.2
91.8	214.2	214.2	214.2	244.8
91.8	214.2	244.8	255	214.2

Intensity	Frequency	CDF	Normalize= CDF *L-1	CDF*L -1=
20	9	9/25=0.36	0.36	91.8
80	12	12/25=0.48	0.84	214.2
81	3	3/25=0.12	0.96	244.8
82	1	1/25= 0.04	1	255

AHE can amplify noise in uniform areas. To prevent this, Contrast Limited Adaptive Histogram Equalization (CLAHE) is often used. It clips the histogram at a threshold before computing the CDF, which limits contrast enhancement in flat regions.

5. Contrast Limited Adaptive Histogram Equalization (CLAHE) : Contrast Limiting to prevent over-enhancement, the contrast is limited by clipping histogram values and redistributing them.

- **Apply Contrast Limiting (Clip Histogram) CLAHE Parameters**

- **Define a Clip Limit**

1- Case 1 : Suppose we set a **ClipLimit=100**

- this limit is large, so no clipping is applied.
- The histogram remains unchanged

Intensity	count	Excess to Clip	New count
20	9	9-5=4	5
80	12	12-5=7	5
81	3	0	5
82	1	0	5

•2- Case 2 ClipLimit is 5 in terms of raw frequency
•All bins exceeding 5 are clipped, and the excess is redistributed.

Intensity	Frequency	Excess to Clip	New Frequency
20	9	9-5=4	5
80	12	12-5=7	5
81	3	0	5
82	1	0	5
		Sum=11	

Total excess=7+4+0+0=11
Redistribute excess=11/5=2.2

Intensity	Frequency after redistribution
20	5+2.2=7.2
80	5+2.2=7.2
81	5+2.2=7.2
82	5+2.2=7.2

New CDF (ClipLimit = 5):L-1=255

Intensity	New Frequency	CDF	CDF	CDF*L-1
20	7.2	7.2/28.8=0.25	0.25	63.75
80	7.2	7.2/28.8=0.25	0.5	127.5
81	7.2	7.2/28.8=0.25	0.75	191.25
82	7.2	7.2/28=0.25	1	255
	Sum =28.8			

After CLAHE

Before CLAHE

CDF*L-1=
91.8
214.2
244.8
255

3- case 3: ClipLimit = 0 (Standard AHE)

•No clipping is applied; the **initial CDF** is used.

Sol :The pixel intensity 80

Intensity	New Value for Intensity 80
100	214.2
5	127.5
0	214.2

Filtering methods

Filtering is a **fundamental step in medical image processing**. It helps in enhancing image quality, reducing noise, and extracting meaningful features for diagnosis or further analysis (like segmentation or classification).

1- Linear Filters in Medical Image Processing

Linear filters are commonly used in medical image processing for tasks such as noise reduction, edge detection, and image enhancement. These filters operate by applying a linear transformation to the pixel values in an image.

Types of Linear Filters:

- **Gaussian Filter:**

•Used for smoothing and noise reduction.
•Applies a Gaussian-weighted sum of surrounding pixel values.

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

where:
•(x,y) are the spatial coordinates,
•σ is the standard deviation controlling the spread of the Gaussian function.

A 3x3 Gaussian kernel with σ=1 (approximate values):

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

$$I=\begin{bmatrix}52 & 55 & 61\\ 60 & 59 & 55\\ 61 & 58 & 55\end{bmatrix}$$

$\frac{1}{16}\times(1\cdot52+2\cdot55+1\cdot61+2\cdot60+4\cdot59+2\cdot55+1\cdot61+2\cdot58+1\cdot55) = \frac{1}{16}\times(52+110+61+120+236+110+61+116+55)=$
$\frac{1}{16}\times921=57.56=58$

Case 2: Whole Image with Padding

Let's **pad** the image by 1 pixel (zero-padding):

$$I_{padded} = \begin{bmatrix}0 & 0 & 0 & 0 & 0\\ 0 & 52 & 55 & 61 & 0\\ 0 & 60 & 59 & 55 & 0\\ 0 & 61 & 58 & 55 & 0\\ 0 & 0 & 0 & 0 & 0\end{bmatrix}$$

$$I=\begin{bmatrix}0 & 0 & 0\\ 0 & 52 & 55\\ 0 & 60 & 59\end{bmatrix}$$

Apply the kernel:

$$\frac{1}{16}\times(1\cdot0+2\cdot0+1\cdot0+2\cdot0+4\cdot52+2\cdot55+1\cdot0+2\cdot60+1\cdot59=\frac{1}{16}\times(0+0+0+0+208+110+0+120+59)=\frac{1}{16}\times497=31.06=$$

$$I=\begin{bmatrix}0 & 0 & 0\\ 52 & 55 & 61\\ 60 & 59 & 55\end{bmatrix}$$

Sum = 0+0+0+104+220+122+60+118+55=6790 + 0 + 0 + 104 + 220 + 122 + 60 + 118 + 55 = 6790+0+0+104+220+122+60+118+55=679

Result = 679/16=42.44≈42

Continue….

➢ Mean (Averaging) Filter:

•Replaces each pixel with the average of its neighboring pixels.

•Useful for reducing random noise.

•A simple **3×3 mean filter** applies the following kernel to an image:

$$H = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

•The filtered pixel value $I'(x,y)$ at location (x,y) is computed as:

$$I'(x,y) = \sum_{i=-1}^{1}\sum_{j=-1}^{1} H(i,j)\cdot I(x+i,y+j)$$

Let's apply a **3×3 mean filter** to a small **grayscale image** patch.

Step 1: Given Image Patch (3×3 Matrix)

Consider a **grayscale** image patch where pixel values range from **0 to 255**

$$I=\begin{matrix}120 & 125 & 130\\ 135 & 140 & 145\\ 150 & 155 & 160\end{matrix}$$

Step 2: Apply the Mean Filter Kernel

A **3×3 mean filter** uses the following kernel:

$$H=\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

Substituting the pixel values:

$$I'(2,2)=\frac{1}{9}(120+125+130+135+140+145+150+155+160)$$

$$I'(2,2)=\frac{1}{9}(1260)=140$$

So, the new value of the **center pixel (140) remains unchanged**, but neighboring pixels will also be smoothed.

Step 4: Apply to the Whole Image

Filtered Image (After Applying the Mean Filter)

$$I' = \begin{bmatrix} 52 & 60 & 70 & 80 & 88 \\ 92 & 100 & 110 & 120 & 128 \\ 132 & 140 & 150 & 160 & 168 \\ 172 & 180 & 190 & 200 & 208 \\ 192 & 200 & 210 & 220 & 228 \end{bmatrix}$$

2- Non-Linear Filters

Unlike **linear filters** (like Gaussian and Mean), which use matrix convolution and work with weighted averages, **non-linear filters** don't follow a linear operation. Instead, they apply rules that depend on the **rank or order of the pixel values**.

1. Median Filter

•Replaces the center pixel with the **median** of the neighborhood.

•Excellent for removing **salt-and-pepper noise**

2. Min Filter

•Replaces the center pixel with the **minimum** value in the window.

•Good for removing **bright spots** (pepper noise).

3. Max Filter

•Replaces the center pixel with the **maximum** value in the window.

•Good for removing **dark spots** (salt noise).

Let's take this small grayscale matrix:

2	**80**	**6**
3	**100**	**7**
4	**5**	**8**

Apply the filter **to the center pixel** (100), we:

1.Take all values in the 3x3 window:
2.Neighborhood=[2,80,6,3,100,7,4,5,8]\
3.Sort the values:
4.Sorted=[2,3,4,5,6,7,8,80,100]\
5.Find the **median** (middle value):
6.Median=6

So, the center pixel (which was 100) becomes **6** after median filtering.

assuming we **pad with 0** (zero-padding). Here's a 3x3 image again:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 80 & 6 & 0 \\ 0 & 3 & 100 & 7 & 0 \\ 0 & 4 & 5 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Sorted: [0, 0, 0, 0, 2, 3, 80, 100]
Median = **2**

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 80 \\ 0 & 3 & 100 \end{bmatrix}$$

Sorted: [0, 0, 0, 2, 3, 6, 7, 80, 100]
Median = **6**

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 80 & 6 \\ 3 & 100 & 7 \end{bmatrix}$$

Continue…..

Linear vs Non-Linear

Feature	Linear Filter	Non-Linear Filter
Noise Handling	Handles Gaussian	Handles impulsive noise
Edge Preservation	Poor	Good
Operation	Weighted sum	Ranking, rules, selection

Thank you