Image Processing

1. Computer Imaging

Can be defined as acquisition and processing of visual information by computer. Computer representation of an image requires the equivalent of many thousands of words of data, so the massive amount of data required for image is a primary reason for the development of many sub areas with field of computer imaging, such as image compression and segmentation.

Computer imaging can be separate into two primary categories:

A. Computer Vision.

B. Image Processing.

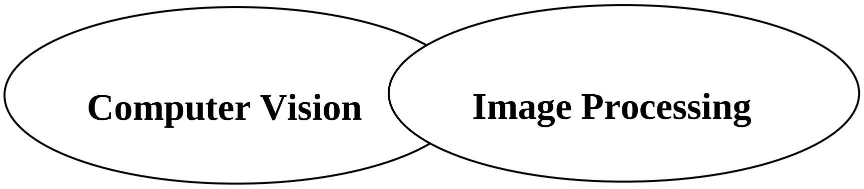


Figure 1: Computer Imaging.

2. Computer Vision

Computer vision computer imaging where the application doses not involve a human being in visual loop. One of the major topics within this field of computer vision is image analysis.

- Image Analysis: involves the examination of the image data to facilitate solving vision problem.

3. Image Processing

Image processing is computer imaging where application involves a human being in the visual loop. In other words the image are to be examined and a acted upon by people. The major topics within the field of image processing include:

1. Image restoration.
2. Image enhancement.
3. Image compression.

1.1 Image Restoration

Is the process of taking an image with some known, or estimated degradation, and restoring it to its original appearance. Image restoration is often used in the field of photography or publishing where an image was somehow degraded but needs to be improved before it can be printed(Figure 2).

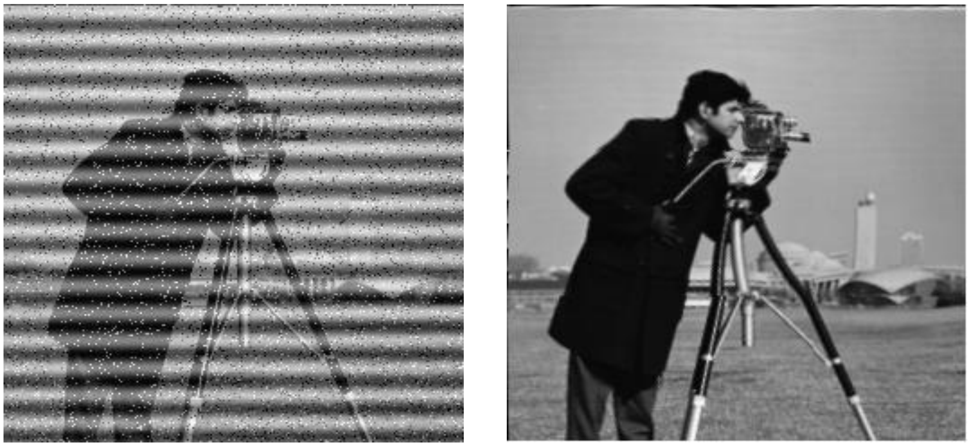
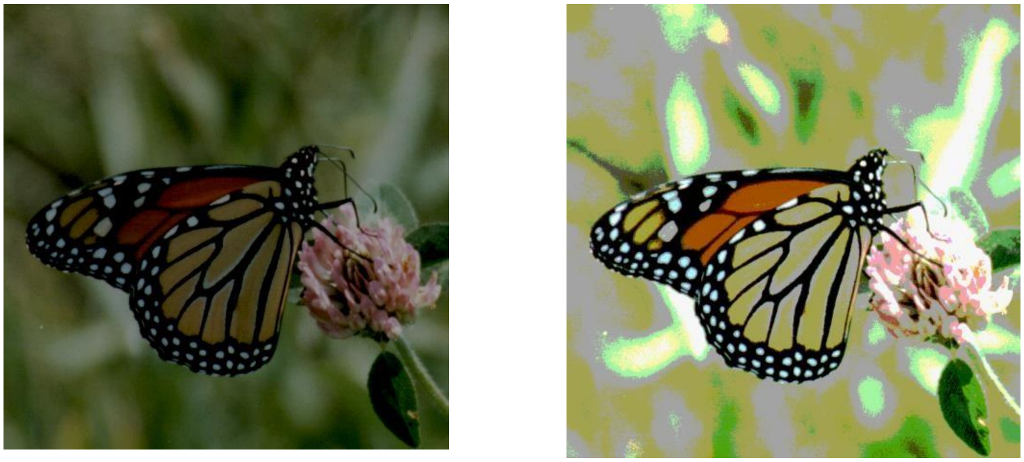


Figure 2: Image Restoration

1.2. Image Enhancement

Involves taking an image and improving it visually, typically by taking advantages of human Visual Systems responses. One of the simplest enhancement techniques is to simply stretch the contrast of an image. Enhancement methods tend to be problem specific. For example, a method that is used to enhance satellite images may not suitable for enhancing medical images. Although enhancement and restoration are similar in aim, to make an image look better. They differ in how they approach the problem. Restoration method attempt to model the distortion to the image and reverse the degradation, where enhancement methods use knowledge of the human visual systems responses to improve an image visually.



a. image with poor contrast **b. Image enhancement by contrast stretching**

Figure 3: Image Enhancement

3.Digitization

The value of the voltage at each instant is converted into a number that is stored, corresponding to the brightness of the image at that point. Note that **the image brightness of the image at that point** depends on both the intrinsic properties of the object and the lighting conditions in the scene.

The image can now be accessed as a two-dimension array of data, where each data point is referred to a pixel (picture element). for digital images we will use the following notation:

I(r,c) = The brightness of image at the point (r, c)

Where r= row and c= column.

"When we have the data in digital form, we can use the software to process the data".

$$Ng= 2^{m} \ldots\ldots\ldots. (1)$$

Where Ng denotes the number of gray levels m, where m is the no. of bits contains in digital image matrix.

Example: If we have (6 bit) in 128 X 128 image. Find the no. of gray levels to represent it ,then find the no. of bit in this image?

Solution:

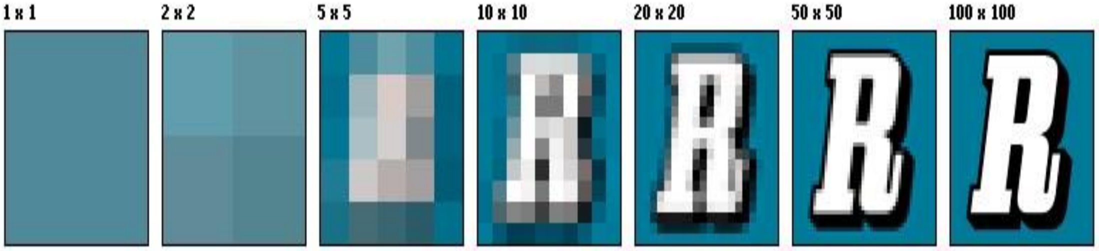
$N_g= 2^{6}=64$ Gray Level

N_b= 128 * 128* 6= 9.830 4 * 10^4 bit

4. Image Resolution

Pixels are the building blocks of every digital image. In computers, resolution is the number of pixels (individual points of color) contained on a display monitor, expressed in terms of the number of pixels on the horizontal axis and the number on the vertical axis.

- Resolution can also be used to refer to the total number of pixels in a digital camera image.
  For example, a camera that can create images of 1600x1200 pixels will sometimes be referred to as a 2-megapixel resolution camera since 1600 x 1200 = 1,920,000 pixels, or roughly 2 million pixels.
- Below is an illustration of how the same image might appear at different pixel resolutions, if the pixels were poorly rendered as sharp squares (normally, a smooth image reconstruction fr om pixels would be preferred, but for illustration of pixels, the sharp squares make the point better).



- An image that is 2048 pixels in width and 1536 pixels in height has a total of 2048×1536 = 3,145,728 pixels or 3.1 megapixels. One could refer to it as 2048 by 1536 or a 3.1-megapixel image.

5. Image Representation

We have seen that the human visual system (HVS) receives an input image as a collection of spatially distributed light energy; this is form is called an optical image. The image types we will consider are:

a. Binary Image

Binary images are the simplest type of images and can take on two values, typically black and white, or '0' and '1'. A binary image is referred to as a 1 bit/pixel image because it takes only 1 binary digit to represent each pixel.

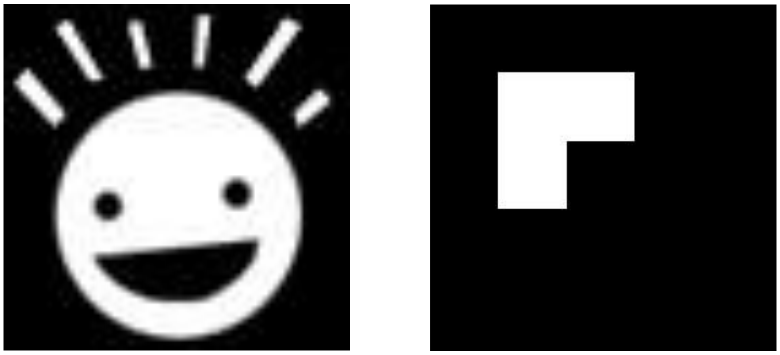


Figure 4: Binary image

b. Gray Scale Image

These types of images are most frequently in computer vision application where the only information required for the task is general shapes, or outlines information. For example, to position a robotics gripper to grasp an object or in optical character recognition (OCR).

Binary images are often created from gray-scale images via a threshold value is, those values above it are turned white ('1'), and those below it

Gray _scale images are referred to as monochrome, or one-color image. They contain brightness information only, no color information. The number of different brightness level available. The typical image contains 8 bit/ pixel (data, which allows us to have (0-255) different brightness (gray) levels. The 8-bit representation is typically due to the fact that the byte, which corresponds to 8-bit of data, is the standard small unit in the world of digital computer.

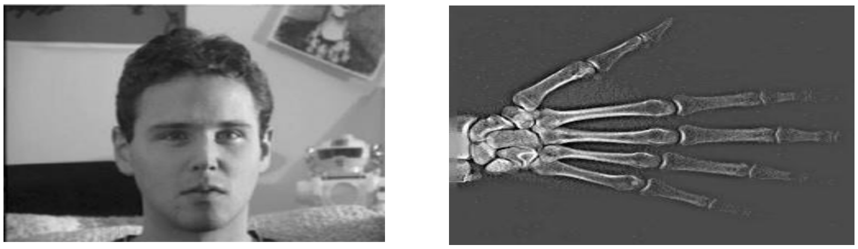


Figure 5: Gray Image

c. Color Image

Color image can be modeled as three band monochrome image data where each band of data corresponds to a different color.

Figure 6: Color Images.

The actual information stored in the digital image data is brightness information in each spectral band. When the image is displayed, the corresponding brightness information is displayed on the screen by picture elements that emit light energy corresponding to that particular color.

Typical color images are represented as red, green, and blue or RGB images. using the 8-bit monochrome standard as a model , the corresponding color image would have 24 bit/pixel – 8 bit for each color bands (red, green and blue). The following figure we see a representation of a typical RGB color image. The following figure illustrate that in addition to referring to arrow or column as a vector, we can refer to a single pixel red ,green, and blue values as a color pixel vector –(R,G,B).

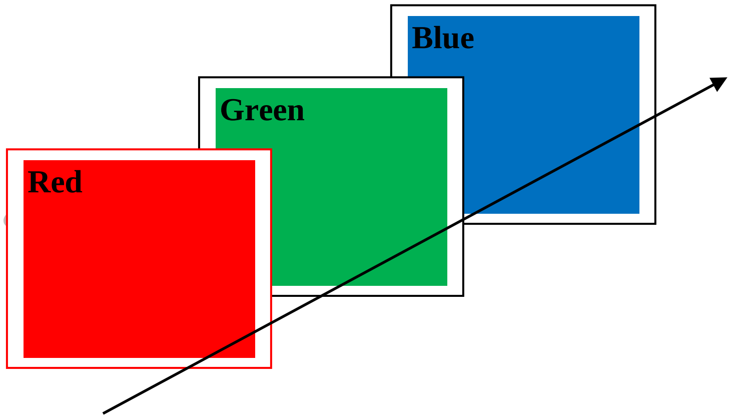


Figure 7: A color pixel vector consists of the red, green and blue pixel values (R, G, B) at one given row/column pixel coordinate(r , c) [1].

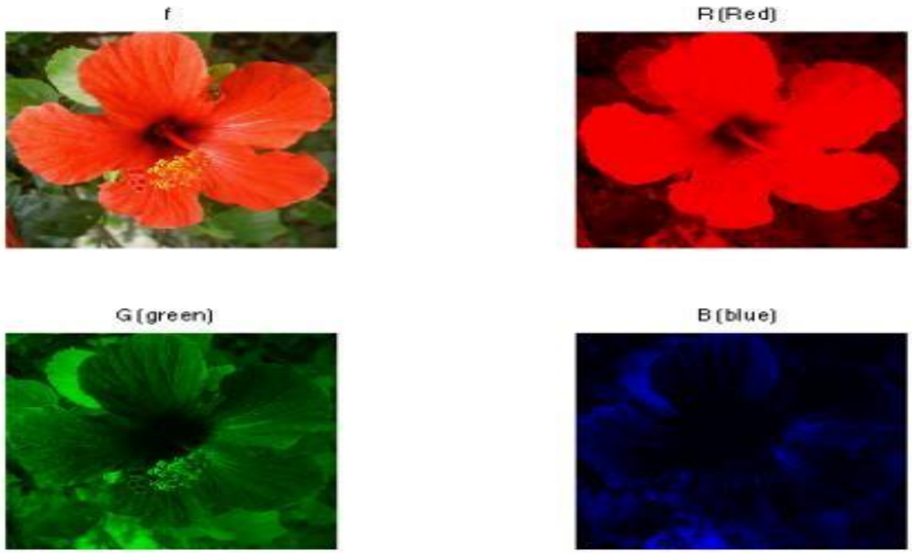
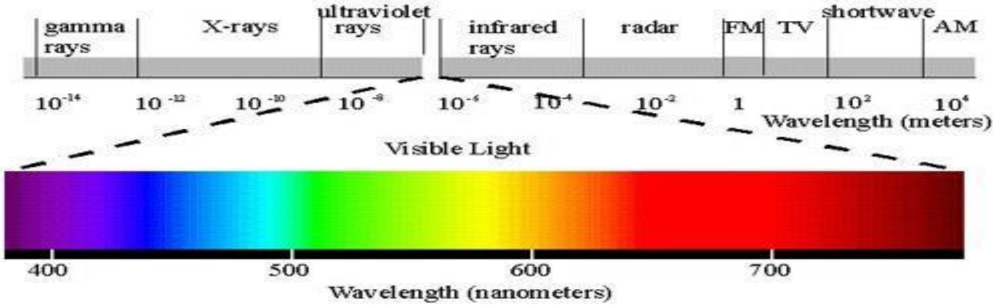


Figure 8: split color image into RGB A color pixel vector consists of the red, green and blue

The hue/saturation /lightness (HSL) color transform allows us to describe colors in terms that we can more readily understand. The lightness is the brightness of the color, and the hue is what we normally think of as "color" and the hue (ex: green, blue, red, and orange). The saturation is a measure of how much white is in the color (ex: Pink is red with more white, so it is less saturated than a pure red).

d. Multispectral Images

A multispectral image is one that captures image data at specific frequencies across the electromagnetic spectrum. Multispectral images typically contain information outside the normal human perceptual range. This may include infrared (تحت الحمراء), ultraviolet (فوق البنفسجيه), X-ray, acoustic (سمعي) or radar data. Source of these types of image include satellite systems, underwater sonar systems and medical diagnostics imaging systems.



Electromagnetic Spectrum

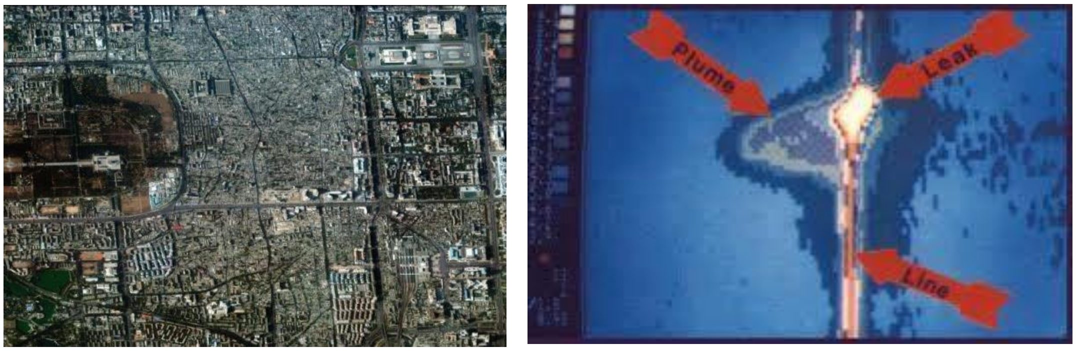


Figure 9: Multispectral images

7. Computer Graphics :

Computer graphics is a specialized field within that refers to the computer science realm that refers to the reproduction of visual data through the use of computer. In computer graphics, types of image data are divided into two primarily categories:

1. **Bitmap image (or raster image):** can represented by our image model I(r, c), where we have pixel data and corresponding brightness values stored in file format.
2. **Vector images**: refer to the methods of representing lines, curves shapes by storing only the key points. These key points are sufficient to define the shapes, and the process of turing theses into an image is called rending after the image has been rendered, it can be thought of as being in bit map format where each pixel has specific values associated with it.

What is the difference between vector and raster graphics?

- Raster graphics are composed of pixels, while vector graphics are composed of paths.
- A raster graphic, such as a gif or jpeg, is an array of pixels of various colors, which together form an image. A vector graphic, such as an .eps file or Adobe Illustrator file, is composed of paths, or lines, that are either straight or curved.
- Because vector graphics are not made of pixels, the images can be scaled to be very large without losing quality. Raster graphics, on the other hand, become "blocky," since each pixel increases in size as the image is made larger.

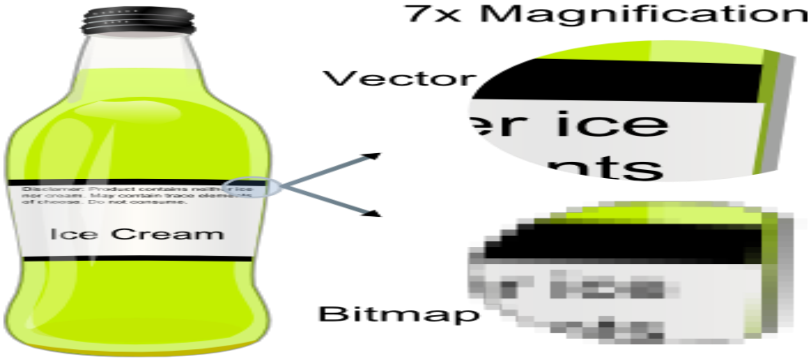


Figure 10: vector and bitmap image.

8. Image Enlarge (Zooming)

The zoom process can be done in numerous ways:

1. **Zero-Order Hold.**
2. **First _Order Hold.**
3. **Convolution.**

1. **Zero-Order hold:** is performed by ~~r~~epeating previous pixel values, thus creating a blocky effect as in the following figure:

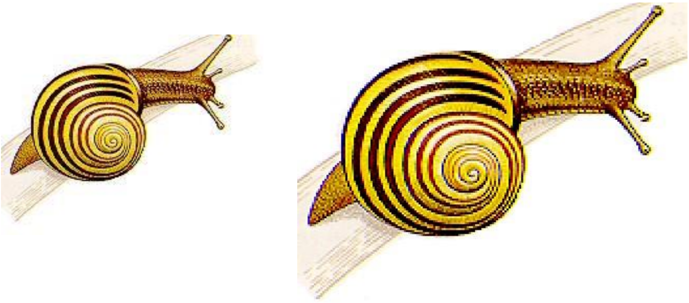


Figure 11: Zero _Order Hold Method

2. **First _Order Hold:** is performed by finding linear interpolation between a adjacent pixels, i.e., finding the average value between two pixels and use that as the pixel value between those two, we can do this for the rows first as follows:

Original Image Array

$$\begin{pmatrix} 8 & 4 & 8 \\ 4 & 8 & 4 \\ 8 & 2 & 8 \end{pmatrix}$$

Image with Rows Expanded

$$\begin{pmatrix} 8 & 6 & 4 & 6 & 8 \\ 4 & 6 & 8 & 6 & 4 \\ 8 & 5 & 2 & 5 & 8 \end{pmatrix}$$

The first two pixels in the first row are averaged (8+4)/2=6, and this number is inserted between those two pixels. This is done for every pixel pair in each row. Next, take result and expanded the columns in the same way as follows:

Image with rows and columns expanded

$$\begin{pmatrix} 8 & 6 & 4 & 6 & 8 \\ 6 & 6 & 6 & 6 & 6 \\ 4 & 6 & 8 & 6 & 4 \\ 6 & 5.5 & 5 & 5.5 & 6 \\ 8 & 5 & 2 & 5 & 8 \end{pmatrix}$$

This method allows us to enlarge an N×N sized image to a size of (2N-1) × (2N-1) and be repeated as desired.

3- Convolution: this process requires a mathematical process to enlarge an image. This method required two steps:

1. Extend the image by adding rows and columns of zeros between the existing rows and columns.
2. Perform the convolution.

The image is extended as follows**:**

Original Image Array

3	5	7
2	7	6
3	4	9

Image extended with zeros

0	0	0	0	0	0	0
0	3	0	5	0	7	0
0	0	0	0	0	0	0
0	2	0	7	0	6	0
0	0	0	0	0	0	0
0	3	0	4	0	9	0
0	0	0	0	0	0	0

Next, we use a convolution mask, which slides a cross the extended image, and perform the simple arithmetic operation at each pixel location

Convolution mask for first–order hold

¼	**½**	**¼**
½	**1**	**½**
¼	**½**	**¼**

The convolution process requires us to overlay the mask on the image, multiply the coincident (متقابله) values and sum all these results. This is equivalent to finding the inner vector product of the mask with the underlying sub-image. So, first, the inner vector product is found by overlaying the mask on the sub-image. Then, multiply coincident terms and sum the resulting products.

For example, if we put the mask over the upper-left corner of the image, we obtain (from right to left and top to bottom):

1/4(0) +1/2(0) +1/4(0) +1/2(0) +1(3) +1/2(0) + 1/4(0) +1/2(0) +1/4(0) =3
Note that the current image values do not change. The next step is to slide the mask over by one pixel and repeat the process as follows:
1/4(0) +1/2(0) +1/4(0) +1/2(3) +1(0) +1/2(5) + 1/4(0) +1/2(0) +1/4(0) =4
Note this is the average of the two existing neighbors. This process continues until we get to the end of the row, placing the result of the operation in the location corresponding to the center of the mask. When the end of the row is reached, the mask is moved down one row, and the process is repeated row by row. This procedure has been performed on the entire image; the process of **sliding, multiplying and summing** is called convolution.

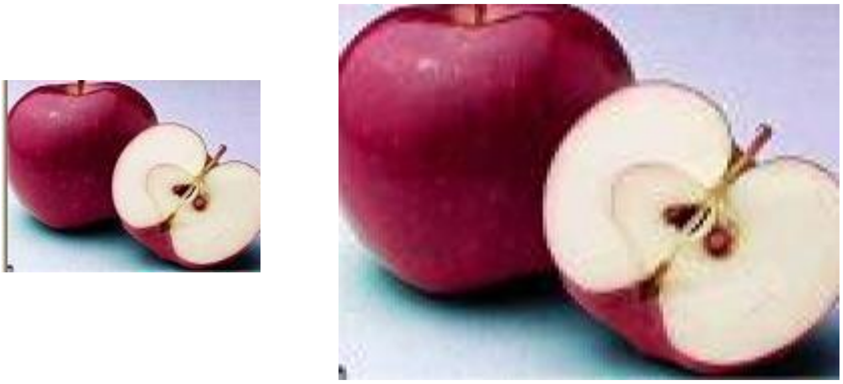


Figure 12: First _Order Hold Method

Note that the output image must be put in a separate image array called a buffer so that the existing values are not overwritten during the convolution process.

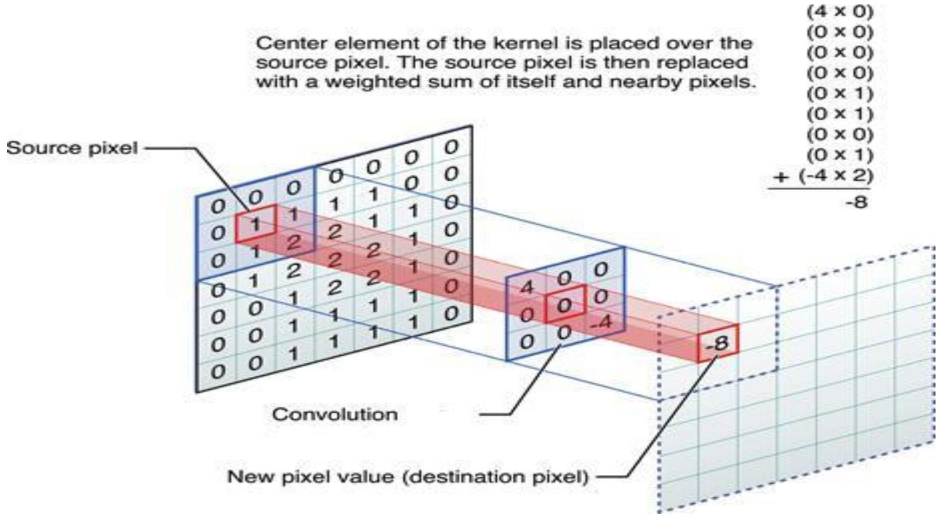
10. The convolution process

Convolution is the process of adding each element of the image to its local neighbors, weighted by the kernel. This is related to a form of mathematical convolution. The matrix operation being performed—convolution—is not traditional matrix multiplication, despite being similarly denoted by

For example, if we have two three-by-three matrices, the first a kernel, and the second an image piece, convolution is the process of flipping both the rows and columns of the kernel and multiplying locally similar entries and summing. The element at coordinates [2, 2] (that is, the central element) of the resulting image would be a weighted combination of all the entries of the image matrix, with weights given by the kernel:

$$\left(\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} * \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right)[2,2] = (i\cdot 1)+(h\cdot 2)+(g\cdot 3)+(f\cdot 4)+(e\cdot 5)+(d\cdot 6)+(c\cdot 7)+(b\cdot 8)+(a\cdot 9).$$

The other entries would be similarly weighted, where we position the center of the kernel on each of the boundary points of the image, and compute a weighted sum.


Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.
(4 x 0)
(0 x 0)
(0 x 0)
(0 x 0)
(0 x 1)
(0 x 1)
(0 x 0)
(0 x 1)
+ (-4 x 2)
-8
Source pixel
Convolution
New pixel value (destination pixel)

c. Move the mask down one row and repeat the process until the mask is convolved with the entire image. We lose the outer row(s) and columns(s).

Why do we use this convolution method when it requires much more calculation than the basic averaging of the neighbours method? The answer is that many computer boards can perform convolution in hardware, which is generally very fast, typically much faster than applying a faster algorithm in software. Moreover, not only first-order hold be performed via convolution, but zero-order hold can also be achieved by extending the image with zeros and using the following convolution mask.

Zero-order hold convolution mask

$$\begin{pmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} \end{pmatrix}$$

Note that for this mask, we will need to put the result in the pixel location corresponding to the lower-right corner because there is no center pixel.

These methods will only allow us to enlarge an image by a factor of (2N-1), but what if we want to enlarge an image by something other than a factor of (2N-1)?

To do this, we need to apply a more general method. We take two adjacent values and linearly interpolate more than one value between them. This is done by defining an enlargement number k and then following this process:

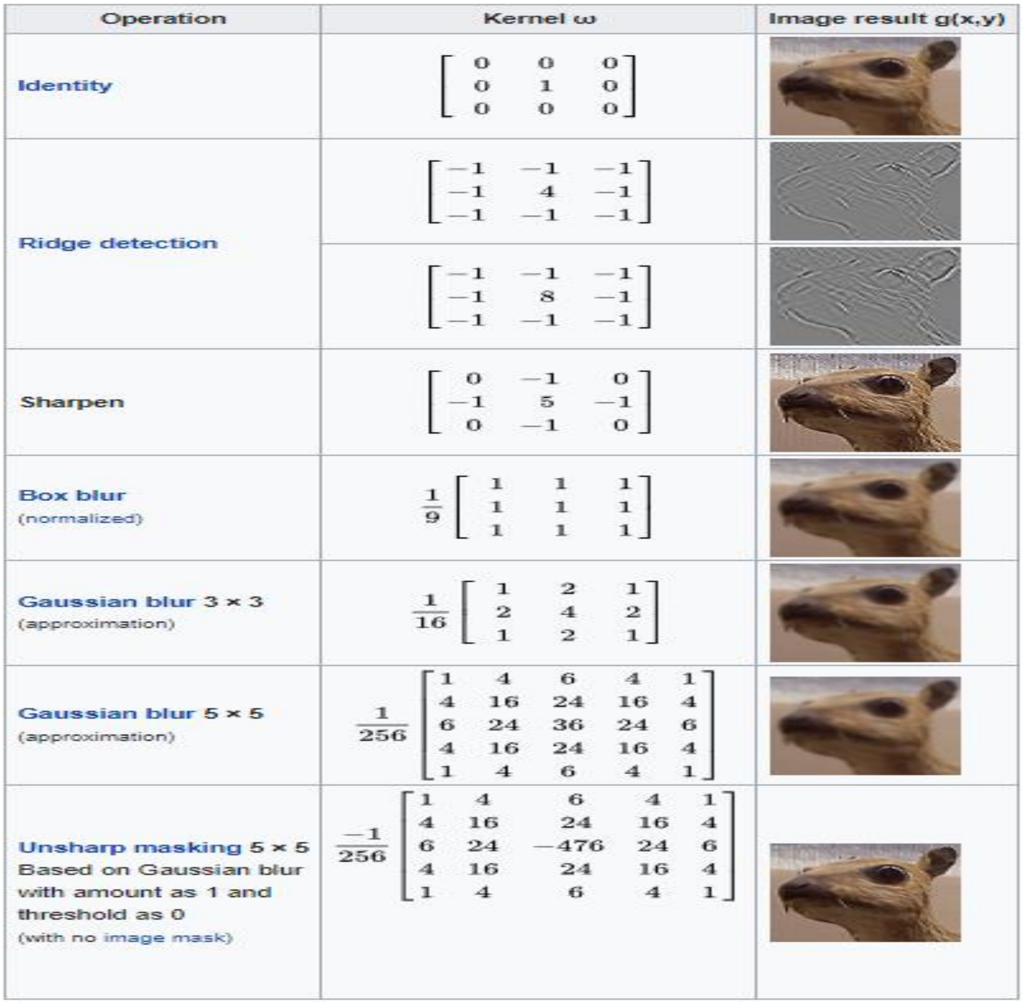
1. Subtract the result by k.
2. Divide the result by k.
3. Add the result to the smaller value, and keep adding the result from the second step in a running total until all (k-1) intermediate pixel locations are filled.

Example: we want to enlarge an image to three times its original size, and we have two adjacent pixel values, 125 and 140.

1. Find the difference between the two values, 140-125 =15.
2. The desired enlargement is k=3, so we get 15/3=5.
3. Next, determine how many intermediate pixel values. We need K-1=3-1=2. The two-pixel values between 125 and 140 are 125+5=130 and 125+2*5 = 135.

- We do this for every pair of adjacent pixels. First along the rows and then along the columns.
- This will allow us to enlarge the image by any factor of K (N-1) +1, where K is an integer and N×N is the image size.
- To process the opposite of enlarging an image is shrinking. This process is done by reducing the amount of data that needs to be processed.
- Two other operations of interest in image geometry are Translation and Rotation. These processes may be performed for application-specific reasons,

for example, to align an image with a known template in the pattern-matching process or make specific image details easier to see.



Operation	Kernel ω	Image result g(x,y)
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Ridge detection	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 4 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur 3 × 3 (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	
Gaussian blur 5 × 5 (approximation)	$\frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	
Unsharp masking 5 × 5 Based on Gaussian blur with amount as 1 and threshold as 0 (with no image mask)	$\frac{-1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & -476 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	

10. Image Algebra

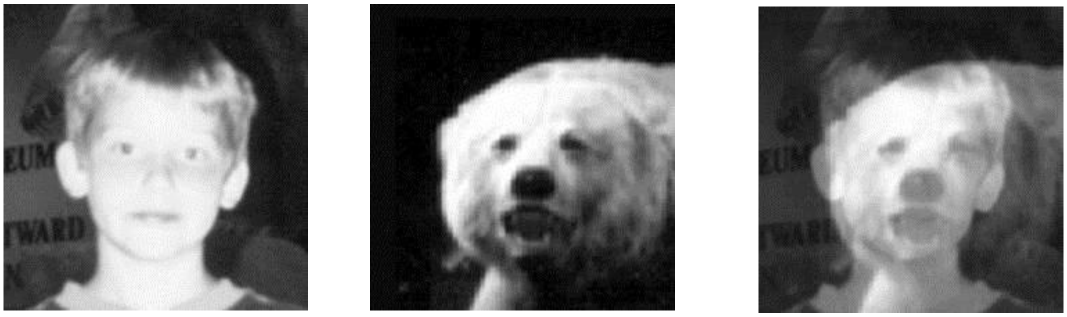
There are two primary categories of algebraic operations applied to the image:

1. Arithmetic operations.
2. Logic operations.

Addition, subtraction, Division and multiplications comprise the arithmetic operations, while AND, OR and NOT make up the logic operations. These operations require only one image and are done on a pixel-by-pixel basis.
We operate on corresponding pixel values to apply the arithmetic operations to two images. For example, to add images I_1 and I_2 to create I_3:

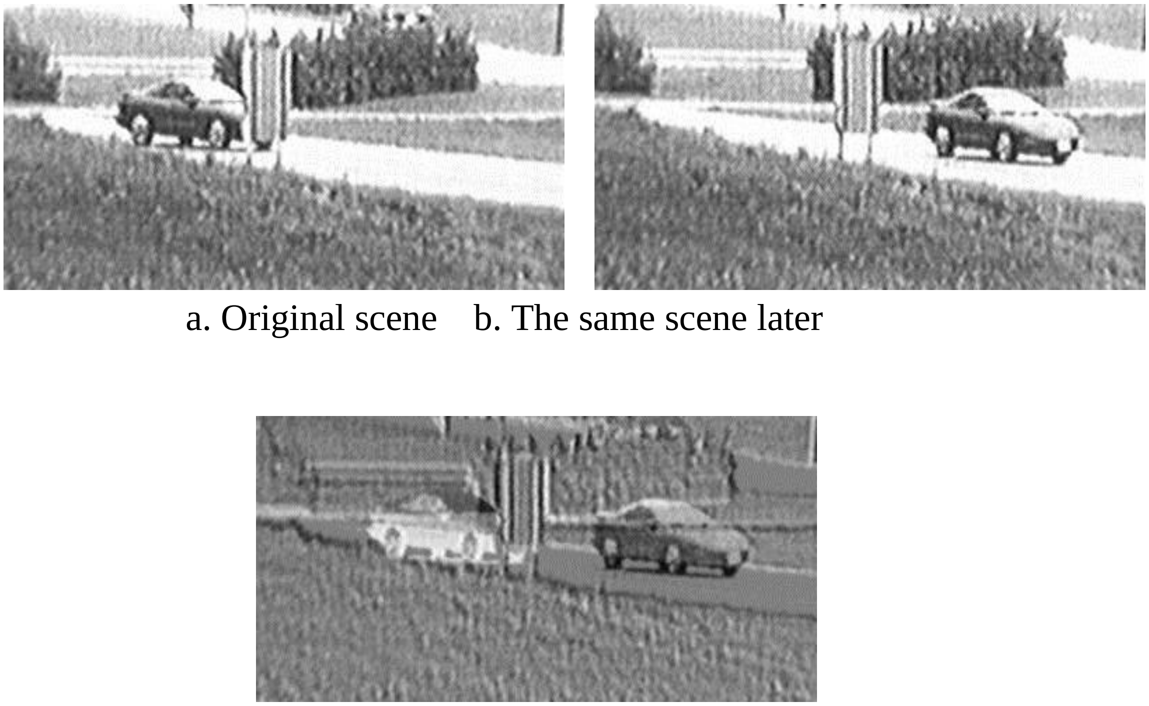
- The addition is used to combine the information in two images. Applications include the development of image restoration algorithms for molding additive noise and special effects, such as image morphing in motion pictures.
- Subtraction of two images is often used to detect motion; consider the case where nothing has changed in a sense; the image resulting from the subtraction of two sequential images is filled with a zero-a-black image. If something has moved in the scene, subtraction produces a nonzero result at the movement location. Applications include Object tracking, Medical imaging, Law enforcement and Military applications
- Multiplication and Division are used to adjust the brightness of an image. One image typically consists of a constant number greater than one. Multiplicating the pixel values by a number greater than one will darken the image (Brightness adjustment is often used as a processing step in image enhancement).

The logic operations AND, OR and NOT form a complete set, meaning that any other logic operation (XOR, NOR, NAND) can be created by combining these essential elements. They operate in a bit-wise fashion on pixel data.



a. First Original image b. Second Original c. Addition of two images

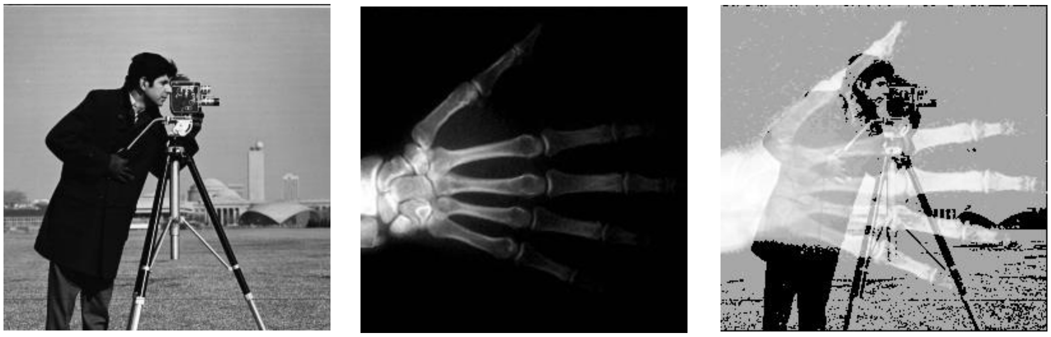
Figure 13: Image Addition.



a. Original scene b. The same scene later

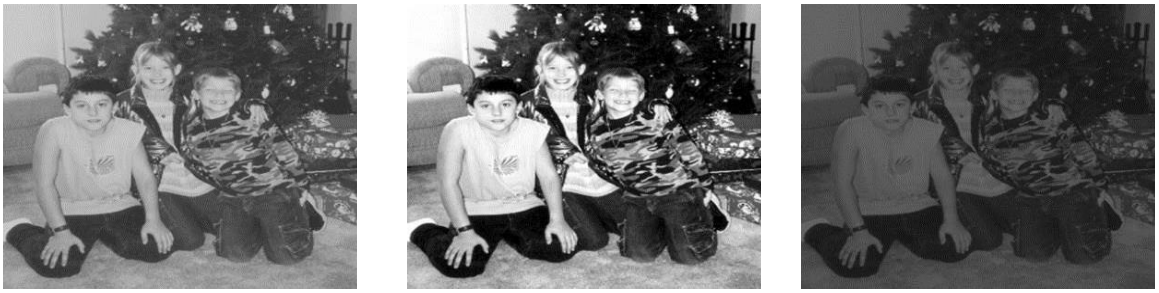
c. Subtraction of scene a from scene b

Figure 14: image subtraction



a. Cameraman image **b.** X-ray image of hand c. Multiplication of two images

Figure 15: Image Multiplication.



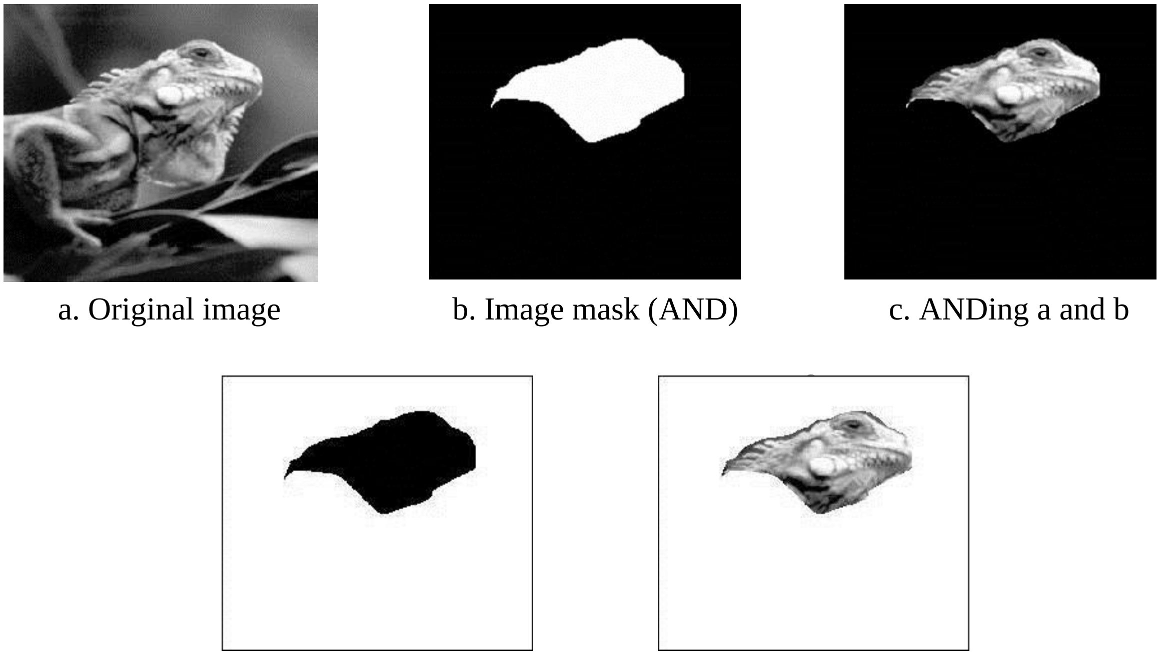
a. Original image Image divided by value<1 c. Image divided by value>1

Figure 16 Image Division.

2.Logic Operation

The logic operations AND and OR combine the information in two images. They may be done for special effects, but a more practical application for image analysis is to perform **a masking operation**. Use AND and OR as simple methods to extract a Region of Interest from an image if more sophisticated graphical methods are unavailable.

Example: A white square ANDed with an image will allow only the portion of the image coincident with the square to appear in the output image with the background turned black, and a black square ORd with an image will allow only the part of the image corresponding to the black square to appear in the output image but will turn the rest of the image white. This process is called image masking. The NOT operation creates a negative of the original image by inverting each bit within each pixel value.



a. Original image b. Image mask (AND) c. ANDing a and b

d. Image mask (OR) e. ORing a and d

Figure 17: Image masking.

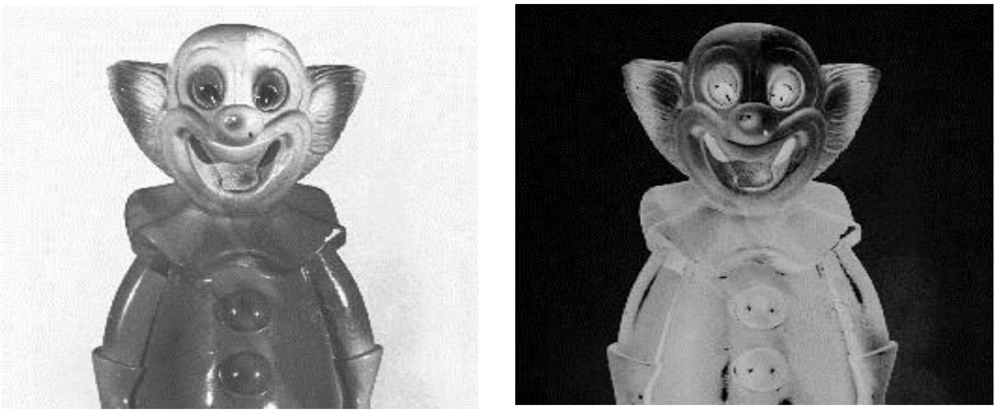


Figure 18: Complement Image (Not Operation)

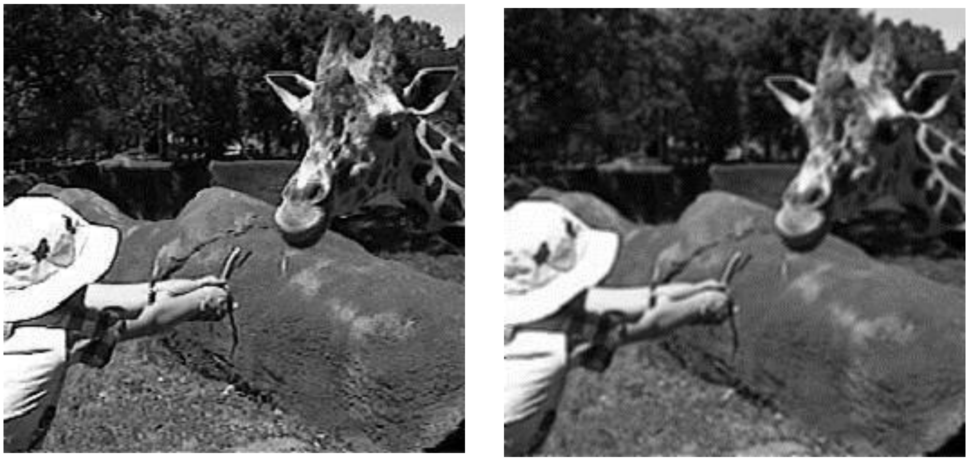
2.5 Image Restoration: Image restoration methods are used to improve an image's appearance by applying a restoration process that uses a mathematical model for image degradation.

.

Arithmetic means filter smoothing or low-pass filter.

$$\begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix}$$

Note that the coefficient of this mask sum to one, so the image brightness will be retained, and the coefficients are all positive, so it will tend to blur the image. This means the filter smoothes out local variations within an image, making it a low-pass filter. **So a low filter can be used to attenuate image noise composed primarily of high-frequency components.**



a. Original image b. Mean filtered image

Figure 19: Mean Filter.

2.8 Histogram

An image's histogram is a plot of the gray _levels values versus the number of pixels at that value. A histogram appears as a graph with "brightness" on the horizontal axis from 0 to 255 (for an 8-bit) intensity scale) and "number of pixels "on the vertical axis. For each colored image, three histograms are computed, one for each component (RGB, HSL). The histogram gives us a convenient -easy -to -read representation of the concentration of pixels versus the brightness of an image; using this graph, we can see immediately:

1 Whether an image is a dark or light and high or low contrast.

2 Give us our first clues about what contrast enhancement would

3 be appropriately applied to make the image more subjectively pleasing to an observer or easier to interpret by succeeding image analysis operations.

So the shape of histogram provide us with information about nature of the image or sub image if we considering an object within the image.

For example:

1 Very narrow histogram implies a low-contrast image

2 Histogram skewed to word the high end implies a bright image

3 Histogram with two major peaks , called bimodal, implies an object that is in contrast with the background

Examples of the different types of histograms are shown in figure (18).

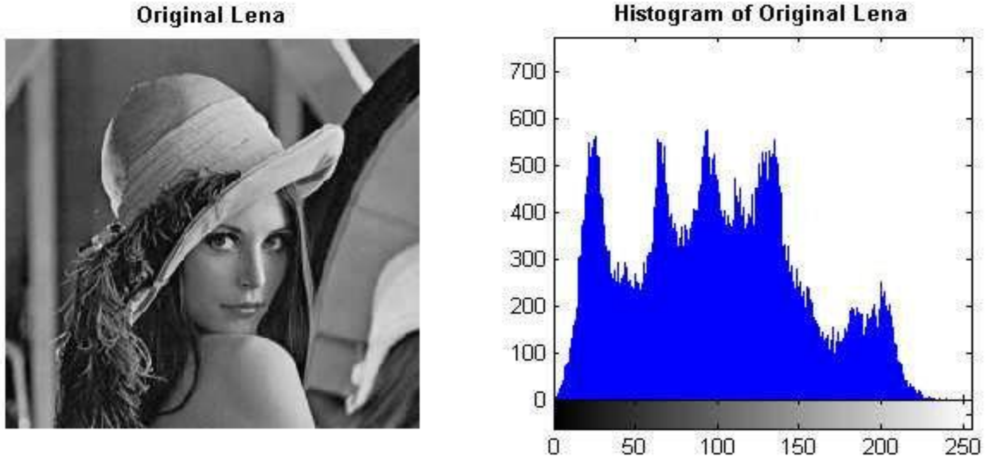


Figure 20: Lenna Image Histogram