What is an image? An image is a visual representation of an object or something. It is the most important sense of human perception.Mathematically, an image may be defined as a two dimensional function, f(x,y), where x and y are special co-ordinates and the amplitude of f at any coordinate point (x,y) is called the intensity or gray label at that point.

What is a digital image image? When, x,y, and the intensity values of f all are finite, discrete quantities we call the image a digital image.

Digital image processing: Digital image processing (DIP) refers to the processing of digital images by means of the computer. Here the input and output both are digital images.

Scope of Digital image processing

* Humans are limited to the visual band of the electromagnetic (EM) spectrum.
* Imaging machines cover almost entire EM spectrum.

Applications of DIP

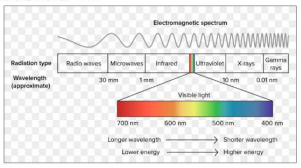
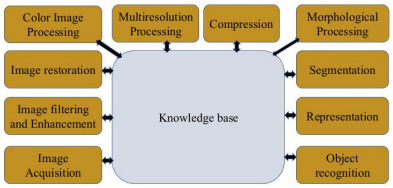


Fig: Electromagnetic spectrum

Applications of DIP : *Nuclear medicine: Inject patient radioactive isotope that emits gamma rays as it decays. Images are produced from the emissions collected by gamma-ray detectors. (Infection and Tumor detection) *Astronomical observation. *Medical diagnosis: Uses a vacuum tube with anode and cathode to generate x-ray. X-ray is passed through the object(patient) and the other side contains a film sensitive to x-ray. *Angiography: Angiography is a type of X-ray used to check blood vessels. Blood vessels do not show clearly on a normal X-ray, so a special dye called a contrast agent needs to be injected into your blood first. *Environmental monitoring, earth-resource mapping, and military systems *SAR imagery must be acquired in inclement weather and all-day-all-night. *SAR produces relatively fine azimuth resolution that differentiates it from other radars.

Fundamental steps of DIP



MATLAB: MATLAB, an abbreviation for 'matrix laboratory,' is a platform for solving mathematical and scientific problems. It is a proprietary programming language developed by MathWorks, allowing matrix manipulations, functions and data plotting, algorithm implementation, user interface creation and interfacing with programs written in programming languages like C, C++, Java and so on.

*In MATLAB, the IPT is a collection of functions that extends the capability of the MATLAB numeric computing environment. It provides a comprehensive set of reference-standard algorithms and workflow applications for image processing, analysis, visualisation and algorithm development.

*It can be used to perform image segmentation, image enhancement, noise reduction, geometric transformations, image registration and 3D image processing operations. Many of the IPT functions support C/C++ code generation for desktop prototyping and embedded vision system deployment.

What is a digital image? A digital image may be defined as a two-dimensional function f(x,y), where 'x' and 'y' are spatial coordinates and the amplitude of 'f' at any pair of coordinates is called the intensity of the image at that point. When 'x,' 'y' and amplitude values of 'f' are all finite discrete quantities, the image is referred to as a digital image. Digitising the coordinate values is referred to as 'sampling,' while digitising the amplitude values is called 'quantisation.' The result of sampling and quantisation is a matrix of real numbers.

A digitised image is represented as:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & .... & f(0,N-1) \\ f(1,0) & f(1,1) & .... & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & .... & f(M-1,N-1) \end{bmatrix}$$

Each element in the array is referred to as a pixel or an image element.

Basic image processing commands in MATLAB: In MATLAB a digital image is represented as:

$$f(x,y) = \begin{bmatrix} f(1,1) & f(1,2) & .... & f(1,N) \\ f(2,1) & f(2,2) & .... & f(2,N) \\ \vdots & \vdots & \vdots & \vdots \\ f(M,1) & f(M,2) & .... & f(M,N) \end{bmatrix}$$

In this representation, you can notice the shift in origin.

Reading images : Images are read in MATLAB environment using the function 'imread.' Syntax of imread is:
imread('filename');
where 'filename' is a string having the complete name of the image, including its extension.
For example, >>F = imread(Penguins_grey.jpg); >>G = imread(Penguins_RGB.jpg);
Please note that when no path information is included in 'filename,' 'imread' reads the file from the current directory. When an image from another directory has to be read, the path of the image has to be specified. *Semicolon (;) at the end of a statement is used to suppress the output. If it is not included, MATLAB displays on the screen the result of the operation specified in that line.
'>>' indicates the beginning of a command line as it appears in the MATLAB command window.

Image processing: Image processing is the technique to convert an image into digital format and perform operations on it to get an enhanced image or extract some useful information from it. Changes that take place in images are usually performed automatically and rely on carefully designed algorithms.

*Image processing is a multidisciplinary field, with contributions from different branches of science including mathematics, physics, optical and electrical engineering. Moreover, it overlaps with other areas such as pattern recognition, machine learning, artificial intelligence and human vision research. Different steps involved in image processing include importing the image with an optical scanner or from a digital camera, analysing and manipulating the image (data compression, image enhancement and filtering), and generating the desired output image.

*The need to extract information from images and interpret their content has been the driving factor in the development of image processing. Image processing finds use in numerous sectors, including medicine, industry, military, consumer electronics and so on.

*In medicine, it is used for diagnostic imaging modalities such as digital radiography, positron emission tomography (PET), computerised axial tomography (CAT), magnetic resonance imaging (MRI) and functional magnetic resonance imaging (fMRI). Industrial applications include manufacturing systems such as safety systems, quality control and automated guided vehicle control.

*Complex image processing algorithms are used in applications ranging from detection of soldiers or vehicles, to missile guidance and object recognition and reconnaissance. Biometric techniques including fingerprinting, face, iris and hand recognition are being used extensively in law enforcement and security.

*Digital cameras and camcorders, high-definition TVs, monitors, DVD players, personal video recorders and cell phones are popular consumer electronics items using image processing.

Types of images: The MATLAB tool box supports four types of images, namely, grey-level images, binary images, indexed images and RGB images. Brief description of these image types is given below.

Grey-level image: Also referred to as monochrome images, these use 8 bits per pixel, where a pixel value of 0 corresponds to 'black,' a pixel value of 255 corresponds to 'white' and intermediate values indicate varying shades of grey. These are also encoded as a 2D array of pixels, with each pixel having 8 bits.

Binary images: These images use 1 bit per pixel, where a 0 usually means 'black' and a 1 means 'white.' These are represented as a 2D array. Small size is the main advantage of binary images.

Indexed images: These images are a matrix of integers (X), where each integer refers to a particular row of RGB values in a secondary matrix (map) known as a colour map.

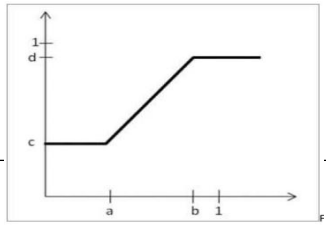
RGB image : In an RGB image, each colour pixel is represented as a triple containing the values of its R, G and B components. In MATLAB, an RGB colour image corresponds to a 3D array of dimensions M×N×3. Here 'M' and 'N' are the image's height and width, respectively, and 3 is the number of colour components. For RGB images of class double, the range of values is [0.0, 1.0], and for classes uint8 and uint16, the ranges are [0, 255] and [0, 65535], respectively.

*Most monochrome image processing algorithms are carried out using binary or greyscale images.

Image quality: Image quality is defined in terms of spatial resolution and quantisation. Spatial resolution is the pixel density over the image. The greater the spatial resolution, the more are the pixels used to display the image. Spatial resolution is expressed qualitatively as dots per inch (dpi). *The image resolution can be changed using the imresize function. The command imresize(x,1/2) halves the image size. This is done by taking a matrix from the original matrix having elements whose row and column indices are even imresize(x,2) means all the pixels are repeated to produce an image of the same size as original, but with half the resolution in each direction.

Histogram: Histogram of a greyscale image represents the frequency of its grey levels occurrence. It is a graph indicating the number of times each grey level occurs in the image. In a dark image, grey levels (and hence the histogram) are cluttered at the lower end. In a uniformly bright image, grey levels (and hence the histogram) are cluttered at the upper end. In a well contrasted image, grey levels (and hence the histogram) would be well spread out over much of the range.

Histogram stretching: This technique, also known as input cropping, consists of a linear transformation that stretches part of the original histogram so that its non-zero intensity range occupies the full dynamic grey scale. *If the histogram of the image is cluttered at the centre, it can be stretched using imadjust function. The following command stretches the histogram as shown in Fig. imadjust (F, [a,b], [c,d])
The values of a, b, c and d must be between 0 and 1.
Command imadjust (F, [], [1,0]) inverts the grey value of the image, to produce a result similar to photographic negative.



Thresholding : Thresholding is used to remove unnecessary details from an image and concentrate on essentials. It is also used to bring out hidden details, in case the object of interest and background have similar grey levels. Thresholding can be further classified as single thresholding and double thresholding. In MATLAB, single as well as double image thresholding can be done.

Single thresholding: A greyscale image is turned into a binary image (black and white) by first choosing a grey level 'T' in the original image, and then turning every pixel black or white depending on whether its grey value is greater than or less than 'T'. Thresholding is a vital part of image segmentation, where users wish to isolate objects from the background. To onvert an image F into black-and-white image G with threshold of 100, the command in MATLAB is G=F>100.
The following example reads image Penguins_grey.jpg and displays both the original image and the image generated after thresholding using a factor of 70: >>A = imread('Penguins_grey.jpg'); >>imshow(A),figure, imshow(A>70)

Double thresholding: In this case, there are two values T1 and T2, and the thresholding operation is performed as the pixel becomes white if the grey level is between T1 and T2. Also, the pixel becomes black if the grey level is outside these threshold values.

Image sharpening: Image sharpening is a powerful tool for emphasising texture and drawing viewer focus. It can improve image quality, even more than what is achieved through upgrading to a high-end camera lens. *Most image sharpening software tools work by applying something called an 'unsharp mask,' which actually acts to sharpen an image. The tool works by exaggerating the brightness difference along the edges within an image. Note that the sharpening process is not able to reconstruct the ideal image, but it creates the appearance of a more pronounced edge. *The command used for sharpening an image in MATLAB is: B = imsharpen(A). *It returns an enhanced version of the greyscale or the true-colour (RGB) input image A, where image features such as edges have been sharpened using the unsharp masking method.
B = imsharpen(A, Name, Value,....) sharpens the image using name-value pairs to control aspects of unsharp masking.
Let us see the use of imsharpen function: >> a=imread('Image_sharpen.jpg'); >>imshow(a)