3. Image Analysis Techniques

Image analysis is the important step in digital image processing. Image analysis is nothing but the extraction of information or data from an image by automatic or semiautomatic method [1]. The goal of all image processing applications is some kind of analysis of the image data. Humans are visual creatures. Image analysis is a process of discovering, identifying and understanding patterns that are relevant to the performance of an image – based task. The principal goal of an image analysis task by computer is to endow a machine with the capability to approximate in some sense with. For example, we describe beauty in visual terms. We categorize visual features of importance to us. For example, we compare an object's size with the size of a standard. We also compare the shape of an object with other familiar shapes. We can extract information like area, perimeter, edge, boundary from an image and analyze it and diagnose it. As an example we say that in a medical image, it is a tumor of 3mm in size. It is character ‘A’ from a printed document. Image analysis attempts to formalize this type of process.

An image analyzer comprises an image acquisition device, a means for converting the image to a digital form, and software/hardware to process the data in order to extract the desired information from it. Image analysis is distinguished from other types of image processing as in other cases like in image enhancement, image transform, image restoration; the output is the another image while in image analysis the output is seen on the same image. [2] As in image analysis we extract image data or information from the image and we can describe the image and interpret it.

Image analysis is the extraction of useful information from images mainly from digital images by means of digital image processing techniques. For the analysis of an image, an image gets captured by any camera or other medical instrument it gets converted to its digital form for further processing. The image gets digitized and then it gets analyzed. The applications of digital image analysis are continuously expanding through all areas of science and industry, including medicine, microscopy, remote sensing, defense, robotics, document processing.
3.1 Levels of image analysis:

There are three levels of image analysis

1. **Low Level Processing:** It deals with functions that may be viewed as automatic reactions, requiring no intelligence on the part of the image analysis system. For example, Image acquisition and pre-processing. This classification encompasses activities from image formation process itself to compensations such as noise reduction, de-blurring, sharpening and contrast.

2. **Intermediate Processing:** It deals with the task of extracting and characterizing components (e.g. Regions) in an image resulting from a low level process. Intermediate level process encompasses segmentation and description.

3. **High-Level Processing:** It involves recognition and interpretation. We can recognize an object and interpret it what it is. For example,: it is cancer tumor? Or it is character ‘A’?

3.2 Approaches in Image Analysis:

According to pattern recognition, image analysis is accomplished by number of ways with different approaches. Low-Level Features may be extracted from the raw gray-level image data and this information is processed sequentially at increasing higher levels. This is known as **bottom-up or data directed approach.** The highest level scene characteristics and then proceed sequentially towards lower-levels, extracting other supporting entities, until the raw image gray levels have been reached. This is called as **top-down approach.** Both of this have practical algorithmic approach and based upon these levels and approaches. [3]

Depending upon these approaches and levels, we see that image analysis is a process consisting of image pre-processing, segmentation, classification and recognition, and interpretation. The steps involved in it are described through its block diagram.
3.3 Block Diagram of Image Analysis

The main block diagram of image analysis through a computer vision is:

![Block Diagram of Image Analysis](image)

3.4 Steps In Image Analysis:

Image Analysis is divided into 4 steps:

1. Pre-processing
2. Feature Extraction
3. Segmentation
4. Classification

From above figure we can say that, the input image taken with the help of camera or any visual system is first pre-processed which involves image enhancement and image restoration processes for noise removal, contrast and brightness adjustment, resolution, etc. Then certain features are extracted through a process of segmentation with extracting boundaries, edge etc. Then the next step of the segmentation in which segmented image is fed to a classifier. This will separate the image into different regions or objects and gives a label to it. Image understanding system will determine the relationship between different objects in a given image as per its description. This process is repeated till we get results up to the mark of satisfaction.
1. **Pre-processing:** Pre-processing functions involve those operations that are normally required prior to the main data analysis and extraction of information, and are generally grouped as radiometric or geometric corrections. Radiometric corrections include correcting the data for sensor irregularities and unwanted sensor or atmospheric noise, and converting the data so they accurately represent the reflected or emitted radiation measured by the sensor. For example,: noise in ultrasound image, contrast adjustment in x-rays. Geometric corrections include correcting for geometric distortions due to sensor geometry variations, and conversion of the data to real world coordinates on the Earth's surface. [4] The objective of the second group of image processing functions grouped under the term of image enhancement is solely to improve the appearance of the imagery to assist in visual interpretation and analysis. Examples of enhancement functions include contrast stretching to increase the tonal distinction between various features in a scene, and spatial filtering to enhance (or suppress) specific spatial patterns in an image. Image transformations are used for image enhancement.

2. **Feature Extraction:** An image feature is a distinguishing primitive characteristic or attribute of an image. Some features are natural, it means they are directly described in an image, For example, Luminance of an image and textural region; while others are artificial which are being displayed during manipulation, For example,: image histogram, spatial frequency spectra. Image Features are used for isolating region from an image and identify or label. This is the most important step. An image feature is a distinguishing primitive characteristics or attribute of an image. Features can be detected in normal gray level or in color images. Feature extraction is the identification of segments, which have particular relevance to the task in hand [5]. Feature Extraction techniques are spatial features, histogram, edge detection, boundary detection, texture, shape features, etc. In Medical field the presence, size, location of an object is the important thing in feature extraction. Bio-Medical images consists of pixel intensities only, we can use this only to calculate edge, boundary, texture, and shape. These techniques are based on fuzzy set theory and on neural network. Various algorithms are there to extract features. The main goal of feature extraction in medical field is the diagnosis of the medical condition; that is whether the tumor is present or not? Or whether it is a cancer or not? If we already know the
features, location, i.e. shape then with the help of robot we can identify that object, it is also a type of feature extraction.

3. **Segmentation:** It is nothing but the subdivision of the image into its constituent parts or objects. The division of the image into regions of similar attributes [6]. The most basic attribute for segmentation is the image amplitude, luminance for a monochrome image while color coordinates for a color image. The segmentation only divides the object and it does not recognize the object and their relationship with one another. The segmentation algorithms are based on two properties of gray levels; discontinuity and similarity.

**Representation and Description:** As representation is nothing but only representation of raw data. This data is represented as per description and is also called as feature selection for quantitative information of an image. In medical field this is a clinical report.

4. **Classification:** Image classification and analysis operations are used to digitally identify and classify pixels in the data. Classification is usually performed on multi-channel data sets (A) and this process assigns each pixel in an image to a particular class or theme (B) based on statistical characteristics of the pixel brightness values. There are a variety of approaches taken to perform digital classification. We will briefly describe the two generic approaches which are used most often, namely supervised and unsupervised classification. Common classification procedures can be broken down into two broad subdivisions based on the method used - supervised classification and unsupervised classification. In a **supervised classification**, the analyst identifies in the imagery homogeneous representative samples of the different surface cover types (information classes) of interest. These samples are referred to as training areas. The selection of appropriate training areas is based on the analyst's familiarity with area and the knowledge of the actual surface cover types present in the image. Thus, the analyst is "supervising" the categorization of a set of specific classes. The numerical information in all spectral bands for the pixels comprising these areas is used to "train" the computer to recognize spectrally similar areas for each class. The computer uses a special program or algorithm (of which there are several variations), to determine the numerical "signatures" for each training class.
Once the computer has determined the signatures for each class, each pixel in the image is compared to these signatures and labeled as the class it most closely "resembles" digitally. Thus, in a supervised classification we are first identifying the information classes which are then used to determine the spectral classes which represent them.

**Unsupervised classification** in essence reverses the supervised classification process. Spectral classes are grouped first, based solely on the numerical information in the data, and are then matched by the analyst to information classes (if possible). Programs, called clustering algorithms, are used to determine the natural (statistical) groupings or structures in the data. Usually, the analyst specifies how many groups or clusters are to be looked for in the data. In addition to specifying the desired number of classes, the analyst may also specify parameters related to the separation distance among the clusters and the variation within each cluster. The final result of this iterative clustering process may result in some clusters that the analyst will want to subsequently combine, or clusters that should be broken down further - each of these requiring a further application of the clustering algorithm. Thus, unsupervised classification is not completely without human intervention. However, it does not start with a pre-determined set of classes as in a supervised classification. In the following sections we will describe each of these four categories of digital image processing functions in more detail. As analysis of sensing imagery involves the identification of various targets in an image and those targets may be environmental or artificial features which consist of points, lines or areas. Targets may be defined in terms of the way they reflect or emit radiation. This radiation is measured and recorded by a sensor, and ultimately is depicted as an image product.

**Interpretation** benefits greatly in many applications when images are viewed in stereo, as visualization (and therefore, recognition) of targets is enhanced dramatically. Viewing objects from directly above also provides a very different perspective than what we are familiar with. Combining an unfamiliar perspective with a very different scale and lack of recognizable detail can make even the most familiar object unrecognizable in an image. Finally, we are used to seeing only the visible wavelengths, and the imaging of wavelengths outside of this window is more difficult for us to comprehend. Recognizing targets is the key to interpretation and information extraction. Observing the differences between targets and their backgrounds involves comparing different targets based on any or all of the visual elements of tone, shape,
size, pattern, texture, shadow, and association. Identifying targets in remotely sensed images based on these visual elements allows us to further interpret and analyze. For interpretation of elements, certain projections are also introduced on image for further image construction [7] and then the concern person gives the diagnosis on these two bases.

Now days, digital image plays an important role in diagnosis of a disease in biomedical field. First the image is captured and then we can analyze it and interpret the results. “The image can tell thousands of words to us.” In biomedical imaging, an image is obtained. Then we diagnose the disease and the disease is seen clinically and plans for treatment and monitoring of the disease for further treatment like therapy or surgery. When the data consists of picture or portion of visual images there is need to first measure various features of the images or of the objects contained in the images before they can be classified.

To extract various features from an image, the pre-processing of images is essential to reduce irrelevant information or noise to enhance the image properties which makes the feature measurement easier and more reliable. In more complicated cases, there will be cycles of image processing and image analysis and decision making to obtain the best results from an image.

### 3.4.1 Pre-processing Techniques

There are various techniques for pre-processing like image enhancement, image restoration and geometric image modifications. These techniques are used for the improvement of an image. The main aim of image enhancement is to improve the interpretability or perception of information in images for human viewers, or to provide 'better' input for other automated image processing techniques. This is suited for the machine. Image modification is nothing but the improvement of observed image. Image geometric modification includes magnification, modification, rotation, non-linear spatial wrapping. [8] It means image enhancement is subjective while image restoration is objective. Image enhancement techniques can be divided into two broad categories:

1. Spatial domain methods, which operate directly on pixels, and
2. Frequency domain methods, which operate on the Fourier transform of an image.
Unfortunately, there is no general theory for determining what ‘good’ image enhancement is when it comes to human perception. If it looks good, it is good! However, when image enhancement techniques are used as pre-processing tools for other image processing techniques, then quantitative measures can determine which techniques are most appropriate. Hence, it is a subjective process while in image restoration depending upon the mathematical formulation the image gets processed. It means depending upon the model, image gets degraded and then it is restored. The common process for image restoration is to add a noise to input image and produce a degraded image. For spatial domain

\[ g(x, y) = h(x, y) * f(x, y) + \eta(x, y) \]  --- 3.1

Where \( h(x, y) * f(x, y) = H \) which is linear, \( \eta(x, y) \) is the additive noise, \( g(x, y) \) is the degraded image and * indicates spatial convolution.

And in Frequency domain it is given by

\[ G(u,v) = H(u,v)F(u,v) + N(u,v) \]  --- 3.2

This is corresponding Fourier Transform.

3.4.1.1 Noise Reduction

Noise is nothing but the errors occurring in the image acquisition process that results in pixel values that do not reflect the true intensities of the real scene. It is due to electronic equipments. Noise reduction is the process of removing noise from a picture. An image can be obtained by any device and various types of noise is formed in images like electrical sensor noise, photographic grain noise. There are several ways that noise can be introduced into an image, depending on how the image is created.

- If the image is scanned from a photograph made film, the film grain is a source of noise. Noise can also be formed with the help of damage film or scanner itself.
- If the image is obtained with the help of a digital (CCD) camera, the gathering of data introduces a noise.
• Electronic interchange of data introduces a noise.

Different kinds of techniques are available for noise reduction. Images taken with both digital cameras and conventional film cameras will pick up noise from a variety of sources. Many further uses of these images require that the noise will be (partially) removed. Spatial noise descriptor is there which is concerned with the gray-level values which are random variables characterized by a PDF (probability density function). [4]

Most common PDF’s are -

• In salt and pepper (also known as random noise or independent noise), pixels in the image are vastly different in color from their surrounding pixels. The defining characteristic is that the color of a noisy pixel bears no relation to the color of surrounding pixels. Generally this type of noise will only affect a small number of image pixels. When viewed, the image contains dark and white dots, hence the term salt and pepper noise. Typical sources include flecks of dust on the lens or inside the camera, or with digital cameras, faulty CCD elements.

• In Gaussian noise (dependent noise), an amount of noise is added to every part of the picture. Each pixel in the image will be changed from its original value by a (usually) small amount. Taking a plot of the amount of distortion of a pixel against the frequency with which it occurs produces a Gaussian distribution of noise. It is also called as white noise in which the intensity does not decrease with increasing frequency.

**Additive Noise: -** When an image is transmitted through some channel, noise is independent degradation it is called as additive noise. It is given by

\[
 f(x, y) = g(x, y) + v(x, y)
\]

--- 3.3

Where ‘v’ is the noise and input image ‘g’ are independent variables. With the help of this we can calculate zero mean Gaussian noise and signal-to-noise ratio (SNR) is given by the square value of the noise contribution.

\[
 E = \xi(x, y) = v^2(x, y)
\]

--- 3.4
And the total value of the observed signal is
\[ F = \xi(x, y) = f^2(x, y) \]  --- 3.5

The SNR is given by
\[ SNR = \frac{F}{E} \]  --- 3.6

SNR is nothing but the mean observation with mean error. It is a measure of an image quality. If it is high, image is good. Depending upon the ability to stimulate the behavior and effects of noise, two types of noise models are there for image restoration:

1. **Spatial Domain described by PDF**
2. **Frequency Domain by Fourier properties**

In Matlab ‘imnoise’ function is used as
\[ g = \text{imnoise}(f, \text{type, parameters}) \]

Where ‘f’ is the input image which converts input image to double in the range [0,1] before adding noise to it. And then give type and parameters like Gaussian, salt-&-pepper, speckle, Poisson, mean and variance

**Removing image noise**

Some algorithms are discussed below for removal of noise. These algorithms are applied in order to reduce noise and/or to prepare images for further processing such as segmentation. We distinguish between linear and non-linear algorithms where the former are analysis in the Fourier domain and the latter are not. We also distinguish between implementations based on a rectangular support for the filter and implementations based on a circular support for the filter. These are used for smoothing of an image. As in our study we consider ultrasound images, speckle and additive noise can be present in it which will be discussed in next chapter.

**3.4.1.2 Filters**

It is the technique used for modifying or enhancing the image. Filters can be used for extracting or removing features. Filtering is a neighborhood operation in which the output image is determined by applying some algorithm to the values of the
pixels in the neighborhood of the corresponding input pixel. [9] A pixel’s neighborhood is some set of pixels defined by their locations relative to that pixel.

There are two types

1. Linear Filters
2. Non-linear Filters

If computation is done on pixels neighborhood then it is linear filtering otherwise it is a non-linear filtering.

**Linear Filters:** - In these the operations are directly on pixels. Common linear filters are Gaussian, uniform, triangular.

**Non-Linear Filters:** - A variety of smoothing filters have been developed that are not linear. While they cannot, in general, be submitted to Fourier analysis, their properties and domains of application have been studied extensively. Common non-linear filters are median, speckle. Some most common filters are:-

Noise added to an image generally has a higher–spatial-frequency spectrum than the normal image components because of spatial de-correlatedness . Simple low-pass filtering can be effective for noise cleaning.

**Spatial Domain Processing:**
- Linear Filtering
- Median Filtering
- Adaptive Filtering

**Linear Filtering:** - This technique is used to remove noise in which Gaussian filters or image averaging technique is used. For removing grain noise, image averaging technique is used .Here each pixel gets set to the average value of the pixel in its neighborhood.

**Median Filtering:** - This is just like image averaging filter but in this each pixel is set to an average of the pixel values in the neighborhood of the corresponding input pixel. The value of the output pixel is determined by the median of the neighborhood pixels rather than mean. These are used to remove outliers without reducing the sharpness of the image and also less blurring of edges. This filtering is also called as rank filtering.
Adaptive Filtering: - This is a type of linear filter which applies to an image adaptively tailoring itself to local variance. When variance is small, it performs more smoothing and vice versa.

Gaussian filters: - One method to remove noise is by convolving the original image with a mask. The Gaussian mask comprises elements determined by a Gaussian function. It gives the image a blurred appearance if the standard deviation of the mask is high, and has the effect of smearing out the value of a single pixel over an area of the image. This brings the value of each pixel into closer harmony with the value of its neighbors. Gaussian filtering works relatively well, but the blurring of edges can cause problems, particularly if the output is being fed into edge detection algorithms for other applications.

Averaging: -

Averaging is a degenerate case of Gaussian filtering, where the function defining the mask values has an infinite standard deviation. An infinite number of filters, both linear and non-linear, are possible for image processing. It is therefore impossible to describe more than the basic types. It is important to use a small consistent set of test images that are relevant to the application area to understand the effect of a given filter or class of filters. The effect of filters on images can be frequently understood by the use of images that have pronounced regions of varying sizes to visualize the effect on edges or by the use of test patterns such as sinusoidal sweeps [11] to visualize the effects in the frequency domain. The filters used in these are low pass, high pass and band pass.

Figure 3.2: (a) Low pass filter (b) Band pass filter (c) High pass filter
Various convolution algorithms applied to sinusoidal image the most common filters are as shown in following table:-

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Filtering Mode</strong></td>
<td></td>
</tr>
<tr>
<td>'corr'</td>
<td>Filtering is done using correlation. This is the default.</td>
</tr>
<tr>
<td>'conv'</td>
<td>Filtering is done using convolution.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Boundary Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'p'</td>
<td>The boundaries of the input image are extended by padding with a value, p. this is the default, with value 0.</td>
</tr>
<tr>
<td>'replicate'</td>
<td>The size of the image is extended by replicating the values in its outer border.</td>
</tr>
<tr>
<td>'symmetric'</td>
<td>The size of the image is extended by mirror-reflecting it across its border.</td>
</tr>
<tr>
<td>'circular'</td>
<td>The size of the image is extended by treating the image as one period a 2-D periodic function.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Size option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'full'</td>
<td>The output is of the same size as the extended (padded) image.</td>
</tr>
<tr>
<td>'same'</td>
<td>The output is of the same size as the input. This is achieved by limiting the excursions of the center of the filter mask to points contained in the original image. This is the default.</td>
</tr>
</tbody>
</table>

**Options for function imfilter.**

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Method</strong></td>
<td></td>
</tr>
<tr>
<td>'symmetric'</td>
<td>The size of the image is extended by mirror-reflecting it across its border.</td>
</tr>
<tr>
<td>'replicate'</td>
<td>The size of the image is extended by replicating the values in its outer border.</td>
</tr>
<tr>
<td>Type</td>
<td>Syntax and Parameters</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>‘circular’</td>
<td>The size of the image is extended by treating the image as one period a 2-D periodic function.</td>
</tr>
</tbody>
</table>

**Direction**

<table>
<thead>
<tr>
<th>Direction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘pre’</td>
<td>Pad for the first element of each dimension.</td>
</tr>
<tr>
<td>‘post’</td>
<td>Pad for the last element of each dimension.</td>
</tr>
<tr>
<td>‘both’</td>
<td>Pad before the first element and after the last element of each dimension. This is default.</td>
</tr>
</tbody>
</table>

**Options for function padarray**

<table>
<thead>
<tr>
<th>Type</th>
<th>Syntax and Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘average’</td>
<td>fspecial(‘average’, [r c ]). A rectangular averaging filter of size r * c. The default is 3*3. A single number instead of [r c] specifies the square filter.</td>
</tr>
<tr>
<td>‘disk’</td>
<td>fspecial(‘disk’, r). A circular averaging filter (within a square of size 2r + 1) with radius r. The default radius is 5.</td>
</tr>
<tr>
<td>‘gaussian’</td>
<td>fspecial(‘gaussian’, [r c], sig). A Gaussian low pass filter of size r * c and standard deviation sig (positive). The defaults are 3 *3 and 0.5. A single number instead of [r c] specifies a square filter.</td>
</tr>
<tr>
<td>‘laplacian’</td>
<td>fspecial(‘laplacian’, alpha). A 3 * 3 laplacian filter whose shape is specified by alpha, a number in the range [0, 1]. The default value for alpha is 0.5.</td>
</tr>
<tr>
<td>‘log’</td>
<td>fspecial(‘log’, [r c], sig). Laplacian of a guassian (LoG) filter of size r * c and standard deviation sig (positive). The defaults are 5 * 5 and 0.5. A single number instead of [r c] specifies a square filter.</td>
</tr>
<tr>
<td>‘motion’</td>
<td>fspecial(‘motion’, len, theta). Outputs a filter that, when convolved with an image, approximates linear motion (of a camera with respect to the image) of len pixels. The direction of motion is theta, measured in degrees, counterclockwise from the horizontal. The defaults are 9 and 0, which represents a motion of 9 pixels in the horizontal direction.</td>
</tr>
</tbody>
</table>
Spatial filters supported by function fspecial

These entire filters have been used but still noise should be present in ultrasound images. This noise is of type additive, due to its head of the probe and speckle noise should be present in an image. To suppress this noise called as speckle noise (additive noise due to head of the probe) with combination of two median filters with different masks called as de speckle filters or Homogenous Region Growing Mean Filter (HRGMF) [6] which preserves the edges. This new filter is used in our study to distinguish objects from its background which will be discussed in next chapter.

3.4.1.3. Edge Detection: - Edge is nothing but the abrupt change in gray level. The edges of an image hold much information in that image. [3, 15] The edges tell where objects are, their shape and size, and something about their texture. An edge is where the intensity of an image moves from a low value to a high value or vice versa. There are numerous applications for edge detection, which is often used for various special effects. Digital artists use it to create dazzling image outlines. The output of an edge detector can be added back to an original image to enhance the edges.

Edge detection is often the first step in image segmentation. Image segmentation, a field of image analysis, is used to group pixels into regions to determine an image's composition. Edge detection is also used in image registration. Image registration aligns two images that may have been acquired at separate times or from different sensors. Some common edge profiles are

| ‘prewitt’ | fspecial (‘prewitt’). Outputs a 3 * 3 prewitt mask, wv, that approximates a vertical gradient. A mask for the horizontal gradient is obtained by transposing the result: wh = wv’. |
| ‘sobel’ | fspecial (‘sobel’). Outputs a 3 * 3 sobel mask, sv, that approximates a vertical gradient. A mask for the horizontal gradient is obtained by transposing the result: sh = sv’. |
| ‘unsharp’ | fspecial (‘unsharp’, alpha). Outputs a 3 * 3 unsharp filters. Parameter alpha controls the shape; it must be greater than 0 and less than or equal to 1.0; the default is 0.2. |
There are an infinite number of edge orientations, widths and shapes, some edges are straight while others are curved with varying radii. There are many edge detection techniques. Each having its own strengths. Some edge detectors may work well in one application and perform poorly in others. Sometimes it takes experimentation to determine what the best edge detection technique for an application is. The simplest and quickest edge detectors determine the maximum value from a series of pixel subtractions. The homogeneity operator subtracts each 8 surrounding pixels from the center pixel of a 3 x 3 window as in following figure [16] Output of the operator is the maximum of the absolute value of each difference.

Similar to the homogeneity operator is the difference edge detector. It operates more quickly because it requires four subtractions per pixel as opposed to the eight needed by the homogeneity operator. The subtractions are upper left – lower right, middle left – middle right, lower left – upper right, and top middle – bottom middle as in following figure.
Figure 3.4: First order derivative for edge detection

If we are looking for any horizontal edges it would seem sensible to calculate the difference between one pixel value and the next pixel value, either up or down from the first (called the crack difference), i.e. assuming top left origin

\[ H_c = y\_difference \ (x, y) = \text{value} \ (x, y) - \text{value} \ (x, y +1) \]  \quad ---3.7

In effect this is equivalent to convolving the image with a 2 x 1 template

\[
\begin{pmatrix}
1 \\
-1
\end{pmatrix}
\]  \quad ---3.8

Likewise

\[ H_r = x\_difference \ (x, y) = \text{value} \ (x, y) - \text{value} \ (x - 1, y) \]  \quad ---3.9

uses the template

\[
\begin{pmatrix}
-1 & 1
\end{pmatrix}
\]  \quad --- 3.10

\( H_c \) and \( H_r \) are column and row detectors. Occasionally it is useful to plot both \( X\_difference \) and \( Y\_difference \), combining them to create the gradient magnitude (i.e. the strength of the edge). Combining them by simply adding them could mean two
edges canceling each other out (one positive, one negative); it is better to sum absolute values (ignoring the sign) or sum the squares of them and then, possibly, take the square root of the result.

It is also to divide the $Y_{\text{difference}}$ by the $X_{\text{difference}}$ and identify a gradient direction (the angle of the edge between the regions)

$$\text{gradient\_direction} = \tan^{-1} \frac{Y_{\text{difference}}(x, y)}{X_{\text{difference}}(x, y)}$$ \hfill --- 3.11

The amplitude can be determine by computing the sum vector of $H_c$ and $H_r$.

$$H(x, y) = \sqrt{H_r^2(x, y) + H_c^2(x, y)}$$ \hfill --- 3.12

Sometimes for computational simplicity, the magnitude is computed as

$$H(x, y) = |H_r(x, y)| + |H_c(x, y)|$$ \hfill --- 3.13

The edge orientation can be found by

$$\theta = \tan^{-1} \left[ \frac{H_c(x, y)}{H_r(x, y)} \right]$$ \hfill --- 3.14

In real image, the lines are rarely so well defined; more often the change between regions is gradual and noisy. [10]

The common first order derivatives used for edge detection are
Sobel edge detection

The Sobel operator is more sensitive to diagonal edges than vertical and horizontal edges. The Sobel 3 x 3 templates are normally given as

\[
x - direction \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}
\]

\[
y - direction \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}
\]

Other first order operation:

The Roberts operator has a smaller effective area than the other mask, making it more susceptible to noise.

\[
h_r = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad h_c = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}
\]

The Prewitt operator is more sensitive to vertical and horizontal edges than diagonal edges.

\[
h_r = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad h_c = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}
\]

The Frei-Chen mask operator

\[
h_r = \begin{bmatrix} 0 & 0 & -1 \\ \sqrt{2} & 0 & \sqrt{2} \\ 0 & 0 & -1 \end{bmatrix} \quad h_c = \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & -1 \end{bmatrix}
\]
Second Order Detection

In many applications, edge width is not a concern. In others, such as machine vision, it is a great concern. The gradient operators discussed above produce a large response across an area where an edge is present. This is especially true for slowly ramping edges. Ideally, an edge detector should indicate any edges at the center of an edge. This is referred to as localization. If an edge detector creates an image map with edges several pixels wide, it is difficult to locate the centers of the edges. It becomes necessary to employ a process called thinning to reduce the edge width to one pixel. Second order derivative edge detectors provide better edge localization. Another advantage of second order derivative operators is that the edge contours detected are closed curves. This is very important in image segmentation. Also, there is no response to areas of smooth linear variations in intensity. Once we have gradient, if the gradient is then differentiated and the result is zero, it shows that the original line was straight. [11] The Laplacian is a good example of a second order derivative operator. It is distinguished from the other operators because it is omni directional. It will highlight edges in all directions. The Laplacian operator will produce sharper edges than most other techniques. These highlights include both positive and negative intensity slopes. The edge Laplacian of an image can be found by convolving with masks such as

\[
\begin{pmatrix}
0 & -1 & 0 \\
-1 & 4 & -1 \\
0 & -1 & 0 \\
\end{pmatrix}
\]

or

\[
\begin{pmatrix}
-1 & -1 & -1 \\
-1 & 8 & -1 \\
-1 & -1 & -1 \\
\end{pmatrix}
\]

The Laplacian set of operators is widely used. Since it effectively removes the general gradient of lighting or coloring from an image it only discovers and enhances much more discrete changes than, for example, the Sobel operator. It does not produce any information on direction which is seen as a function of gradual change. It enhances noise, though larger Laplacian operators and similar families of operators tend to ignore noise. Kirsch’s operator is used to detect the edges .It is a compass operator. It is given
Determining zero crossings

The method of determining zero crossings with some desired threshold is to pass a 3 x 3 window across the image determining the maximum and minimum values within that window. If the difference between the maximum and minimum value exceed the predetermined threshold, an edge is present. Notice the larger number of edges with the smaller threshold. Also notice that the width of all the edges is one pixel wide.

A second order derivative edge detector that is less susceptible to noise is the Laplacian of Gaussian (LoG). The LoG edge detector performs Gaussian smoothing before application of the Laplacian. Both operations can be performed by convolving with a mask of the form.

\[
\text{LoG}(x, y) = \frac{1}{\pi \sigma^4} \left[ 1 - \frac{(x^2 + y^2)}{2\sigma^2} \right]^2 e^{-\frac{(x^2 + y^2)}{2\sigma^2}} \quad \text{---3.15}
\]

Where \(x, y\) presents row and column of an image, \(s\) is a value of dispersion that controls the effective spread. Due to its shape, the function is also called the Mexican hat filter. Fig. 3.5 shows the cross the LoG edge operator with different values of mean. The wider the function, the wider the edge that will be detected. A narrow function will detect sharp edges and more detail.

![Figure 3.5: LoG Edge Operator with Different Mean Values](image-url)
The greater the value of mean, the wider the convolution mask necessary. The first zero crossing of the LoG function is at $\sqrt{2\sigma}$. The width of the positive center lobe is twice that. To have a convolution mask that contains the nonzero values of the LoG function requires a width three times the width of the positive center lobe. Edge detection based on the Gaussian smoothing function reduces the noise in an image. That will reduce the number of false edges detected and also detects wider edges. Most edge detector masks are seldom greater than $7 \times 7$. Due to the shape of the LoG operator, it requires much larger mask sizes. The initial work in developing the LoG operator was done with a mask size of $35 \times 35$.

### 3.4.2. Segmentation Techniques:

One of the major steps in digital image analysis is the image segmentation. Since from last 3 decades, the research is going on and lots of algorithms are proposed. [13] The main aim of segmentation is to divide an image into regions or parts which have same arguments or parameters such as gray level, texture, color. Various kinds of segmentation techniques are found in literature. Image segmentation techniques are based on the approaches of the algorithm. These include edge, region, Knowledge, template matching, watershed, etc. Each of it has own merits and demerits. With the help of suitability and cost & time estimation, the algorithms are applied. In the analysis of the objects in images it is an essential step that we can distinguish between the objects of interest and "the rest." This latter group is also referred to as the background. The techniques that are used to find the objects of interest are usually referred to as segmentation techniques - segmenting the foreground from background. In this section we will two of the most common techniques—thresholding and edge finding—and we will present techniques for improving the quality of the segmentation result. It is important to understand that:

1. There is no universally applicable segmentation technique that will work for all images.

2. No segmentation technique is perfect.

The general formula of segmentation technique is to divide an image into homogenous segments; it means spatially connected groups of pixels. To separate these neighborhood segments without errors is a skillful job. Image segmentation process is found in intermediate level and has many practical applications in medical imaging. [12] Traditional approach in segmentation is the localization of region, edge
or boundary. For this purpose various algorithms are developed based on the properties of gray level that are discontinuity and similarity. In the first category the approach is to partition an image based on changes in gray level or image lamination for a monochrome image and for a color image, color components are seen. Means here there is a detection of isolated point, lines, and edges. [6, 9]

Following figure is used for segmentation process:-

As per the above figure, input image is given to pre-processing phase, where image gets enhanced which is carried out either by spatial domain or by frequency domain methods. The result of pre-processing is given to segmentation phase where the image gets isolated in to homogenous parts of same attribute and the output is the segmented region which is nothing but the region of interest. The key of success of image segmentation is the image enhancement because if the image improvement is good, then quality of segmented region is also good. In image analysis, image segmentation is an important step; it has lot of applications within various disciplines. In medical image analysis image segmentation is used for diagnosis of disease and interpretation of it.

Segmentation process group pixels which have similar attributes. Segmentation process bridges the gap between low-level and high-level processing means enhance and analyze the group of pixels which will be used for detection, recognition and measurement of object of interest. Segmentation process can be classified into two types

**Contextual:** It exploits the relationship between image features. It means it group pixels of similar gray levels and is close to each other.
Non-contextual: It ignores the relationship exist between features in an image. Pixels are grouped on global attribute ex. texture, gray level Segmentation is done as follows:

![Image of Size M×N and Segmented Image]

Segmentation can be broadly differentiated in 3 groups:

1. **Global Knowledge Based:** About an image or its parts depending upon the image features which are calculated with the help of histograms.

2. **Edge Based:** Which is based on edge detection operator.

3. **Region Based:** Which are based on closed edge detection or boundary.

1. **Global Knowledge Base:** It is about an image or its parts, the knowledge is represented by a histogram. Threshold techniques, which make decisions based on local pixel information, are effective when the intensity levels of the objects fall squarely outside the range of levels in the background. Because spatial information is ignored, however, blurred region boundaries can create havoc. Means the object is separated from background. Histogram is considered as a PDF (Probability Density Function) of a Gaussian mixture, thus the segmentation is done on pixel classification. [15] This method is good only for small noise reduction and for smoothing of spatial coordinates. Various threshold values are applied for good result. Thresholding technique is based upon a simple concept. A parameter $\theta$ called the **brightness threshold** is chosen and applied to the image $f[x, y]$ as follows:

   If $F(x, y) >= \theta \rightarrow f(x, y) = object = 1$

   Else $f(x, y) = background = 0$ --- 3.16

   This version of the algorithm assumes that we are interested in light objects on a dark background and vice versa. The output is the label "object" or "background" which, due to its dichotomous nature, can be represented as a Boolean variable "1" or "0". For selection of threshold there is no such theory applicable to all images.
Various algorithms are developed for threshold selection like isodata, background symmetry and triangle algorithm.

2. Edge-Based Segmentation:

Edge-based methods center around contour detection: their weakness in connecting together broken contour lines make them, too, prone to failure in the presence of blurring. Edge-based segmentation represents a large group of methods based on information about edges in the image; it is one of the earliest segmentation approaches and still remains very important. Edge-based segmentations rely on edges found in an image by edge detecting operators -- these edges mark image locations of discontinuities in gray level, color, texture, etc. But the image resulting from edge detection cannot be used as a segmentation result. Supplementary processing steps must follow to combine edges into edge chains that correspond better with borders in the image.[18,19] The final aim is to reach at least a partial segmentation -- that is, to group local edges into an image where only edge chains with a correspondence to existing objects or image parts are present. [17] We will discuss several edge-based segmentation methods which differ in strategies leading to final border construction, and also differ in the amount of prior information that can be incorporated into the method. The more prior information that is available to the segmentation process, the better the segmentation results that can be obtained. The most common problems of edge-based segmentation, caused by image noise or unsuitable information in an image, are an edge presence in locations where there is no border, and no edge presence where a real border exists. First we will discuss simple edge-based methods requiring minimum prior information and the necessity for prior knowledge will increase during the section.

**Edge Image Thresholding:**

Almost no zero-value pixels are present in an edge image, but small edge values correspond to non-significant gray level changes resulting from, e.g., quantization noise, small lighting irregularities, etc. Simple thresholding of an edge image can be applied to remove these small values. The approach is based on an image of edge magnitude processed by an appropriate threshold. Selection of an appropriate global threshold is often difficult and sometimes impossible; p-tile thresholding can be
applied to define a threshold. Alternatively, non-maximal suppression and hysteresis thresholding can be used. [14, 19]

**Hough Transform**: In this technique we see how to group isolated edge points into image structures. As edge linking, edge relaxation, and optimal edge determination using graph searching. All of these require some continuous path of edge pixels, near edge pixels, or edge-cost information. This technique is called as the *Hough Transform*, which doesn't require connected or even nearby edge points. [20]

Here, finding edges through straight lines, consider a single isolated edge point \((x, y)\). There could be an infinite number of lines that could pass through this point.

Each of these lines can be characterized as the solution to some particular equation. The simplest form in which to express a line is the *slope-intercept* form:

\[
y = mx + c
\]

Where \(m\) is the slope of the line and \(b\) is the \(y\)-intercept (the \(y\) value of the line when it crosses the \(y\) axis). Any line can be characterized by these two parameters \(m\) and \(b\).

\[
b = y - mx
\]

We can characterize each of the possible lines that pass through point \((x, y)\) as having coordinates \((m, b)\) in some slope-intercept space. In fact, for all the lines that pass through a given point, there is a unique value of \(b\) for \(m\): The set of \((m, b)\) values corresponding to the lines passing through point \((x, y)\) form a line in \((m, b)\) space. Every point in image space \((x, y)\) corresponds to a line in parameter space \((m, b)\) and each point in \((m, b)\) space corresponds to a line in image space \((x, y)\).

**Algorithm:**

1. Find all of the desired feature points in the image.
2. For each feature point
3. For each pixel \(I\) on the target boundary
4. Get relative position from the reference point ‘i’
5. Add this offset position from ‘i’
6. Increment the position in the accumulator
7. Find local maxima in the accumulator
8. If desired map each maxima in the accumulator back to image space using the
target boundary table.

This is used to find small shape in the image. With the help of Hough transform we
can find isolated pixel, line and edge also.

3. Region-Based Segmentation It depends upon the boundary or closed edge. The
main objective is to partition the image in to homogenous regions. The basic
formulation is

\( i = 1 \bigcup^n R_i = R \)

b) \( R_i = \) is a connected region; \( i=1,2, \ldots n \)

c) \( R_i \cap R_j = 0 \) for all \( i \) and \( j; i \neq j \)

d) \( P(R_i) = \) True for \( i=1,2, \ldots n \)

e) \( P(R_i \cup R_j) = \) False, for \( i \neq j \) ---3.17

Here, \( P(R_i) \) is the logical predicate defined over the points in the set Ri and \( \emptyset \)
is the null set. The condition (a) indicates that the segmentation must be complete i.e.
every pixel must be in a region. Condition (b) indicates that points must be connected
in some sense. Equation c indicates that the regions must be disjoints. condition (d)
indicates that this deals with the properties that must be satisfied by pixel in a
segmented region i.e. \( P(R_i) = \) TRUE if all pixels in \( R_i \) have same gray level,
condition (e) indicates that the region \( R_i \) & \( R_j \) are different in the sense of predicate
\( P \). [15]

Region Growing Segmentation: As its name indicates that region growing group’s
pixels or sub regions into large regions based on some predefined criteria. The basic
approach is to give the “seed” points and from these grow regions by appending to
each seed points with its neighboring pixels that have properties similar to each seed.
[16] The selection of similarity criteria depends not only upon the neighboring pixel
but on available image data also. The use of connectivity is the solution for this
problem, in this method we can also define multiple seed points or select seed point for finding more than one region simultaneously. It means by recursive procedure the problem gets solved and the cost and time gets reduced in very less effort.

**Region Merging and Splitting :-** This is the method in which an image is divided into set of, arbitrary disjointed regions and the merge or split the regions in an attempt to satisfy the condition as per above equation. It works iteratively for finding similar attributes for merging and splitting. [17] Let R represent the entire image regions & select predicate P. For segmenting R into smaller & smaller quadrant regions so that for any region Rᵢ, P (Rᵢ) = TRUE. We start the entire region if P (R) = FALSE, we divide the image in to quadrants. If P is FALSE for any quadrant we again divide into sub quadrants. This technique is called as Quad tree. If only splitting were used, the final partition likely would contain adjacent regions with identical properties.

**Algorithm :**

1. Split into 4 disjoint quadrants any regions Rᵢ for which P(Rᵢ) = FALSE
2. Merge any adjacent regions Rⱼ & Rₖ for which P(Rⱼ U Rₖ) = TRUE
3. Stop when no further merging & splitting is possible.

Merging should be stopped when condition 2 is FALSE but regions are of different sizes.

**Clustering:** Clustering can be considered the most important *unsupervised learning* problem; it deals with finding a *structure* in a collection of unlabeled data. A loose definition of clustering could be “the process of organizing objects into groups whose members are similar in some way”. A *cluster* is therefore a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters.[17] In this case we easily identify the 4 clusters into which the data can be divided; the similarity criterion is *distance*: two or more objects belong to the same cluster if they are “close” according to a given distance (in this case geometrical distance or Euclidian distance). This is called *distance-based clustering.*[21] Another kind of clustering is *conceptual clustering*: two or more objects belong to the same cluster if this one defines a concept *common* to all that objects. In other words, objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures.
**MinkowskiMetric**

For higher dimensional data, a popular measure is the Minkowski metric,

\[
d_p(x_i, x_j) = \left[ \sum_{k=1}^{d} |x_{ik} - x_{jk}|^p \right]^{\frac{1}{p}}
\]

--- 3.18

Where \(d\) is the dimensionality of the data. The *Euclidean* distance is a special case where \(p=2\), while *Manhattan* metric has \(p=1\). However, there are no general theoretical guidelines for selecting a measure for any given application. It is often the case that the components of the data feature vectors are not immediately comparable. It can be that the components are not continuous variables, like length, but nominal categories, such as the days of the week. In these cases again, domain knowledge must be used to formulate an appropriate measure.

**Template Matching:** Matching is another basic approach to segmentation that can be used to locate known objects in an image, to search for specific patterns, etc.[22]

The best match is based on some criterion of optimality which depends on object properties and object relations. Matched patterns can be very small, or they can represent whole objects of interest. While matching is often based on directly comparing gray-level properties of image sub-regions, it can be equally well performed using image-derived features or higher-level image descriptors. In such cases, the matching may become invariant to image transforms. Criteria of optimality can compute anything from simple correlations up to complex approaches of graph matching. The matching criterion is: Exact copy of the pattern of interest cannot be expected in the processed image - some part of the pattern is usually corrupted in real images by noise, geometric distortion, occlusion, etc. Search for locations of maximum match is appropriate.
**Algorithm:**

Evaluate a match criterion for each location and rotation of the pattern in the image.

1. Local maxima of this criterion exceeding a preset threshold represent pattern locations in the image.

Matching criteria can be defined in many ways; in particular, correlation between a pattern and the searched image data is a general matching criterion. Let $f$ be a processed image, $h$ be a pattern for which to search, and $V$ be the set of all image pixels in the processed image. Possible matching optimality criteria describing a match between $f$ and $h$ located at a position $(u, v)$:

\[
C_1(u, v) = \frac{1}{\max_{(i,j) \in V} |f(i+u+\nu, j+\nu) - h(i, j)|} \quad \text{--- 3.19}
\]

\[
C_2(u, v) = \frac{1}{\sum_{(i,j) \in V} |f(i+u+\nu, j+\nu) - h(i, j)|} \quad \text{--- 3.20}
\]

\[
C_3(u, v) = \frac{1}{\sum_{(i,j) \in V} \left( f(i+u+\nu, j+\nu) - h(i, j) \right)^2} \quad \text{--- 3.21}
\]

Since absolute match must be considered, adding "1" to all denominators is practical.

If a fast, effective Fourier transform algorithm is available, the convolution theorem can be used to evaluate matching. The correlation between a pattern $h$ and image $f$ can be determined by first taking the product of the Fourier transform $F$ of the image $f$ and the complex conjugate of the Fourier transform $H^\#$ of the pattern $h$ and then applying the inverse transform. To compute the product of Fourier transforms, $F$ and $H^\#$ must be of the same size; if a pattern size is smaller, zero-valued lines and columns can be added to inflate it to the appropriate size. Sometimes, it may be better to add non-zero numbers, For example, the average gray level of processed images can serve the purpose well.
**Morphological Segmentation:** It is based on form and structure. It will operate on binary images & a structuring element is there as an input and combine it with operators such as intersection, complement, and union. The process object in input image is based on its shape as like in mathematical morphology. Usually, it is 3×3 and has starting form center pixel. It is shifted over the image and each of the pixels is compared with the underlying pixel. If two pixels must satisfy the condition defined by the set of operator then assign a value 0 or 1 for binary images.

It is nothing but to produce a skeleton object using skeletization and thinning. Mathematical morphology refers to this which is based on topology, set theory, lattice which is powerful to extract features in an image. Most commonly used operations in this are 1) opening and closing 2) Erosion and dilation. Both of them required a structure element which is a small grid pixel either “1” or “0”. It is applied to change the structure of the image content. It is of size 3×3, 5×5 and so on depending upon the application. For 3×3 it is given by using labels

\[
\begin{pmatrix}
  x_3 & x_2 & x_1 \\
  x_4 & X & x_0 \\
  x_5 & x_6 & x_7
\end{pmatrix}
\]

Means every pixel has 8 neighbors. Different structures like ‘disk’, ‘rectangle’, ‘line’, ‘diamond’ etc can be found. and the results are obtained as per the application.

**Erosion and dilation:** It is used for more complex morphological operations. The two basic morphological set transformations are *erosion* and *dilation*; these transformations involve the interaction between an image \( A \) (the object of interest) and a structuring set \( B \), called the *structuring element*. Typically the structuring element \( B \) is a circular disc in the plane, but it can be any shape. The image and structuring element sets need not be restricted to sets in the 2D plane, but could be defined in 1, 2, 3 (or higher) dimensions. Let \( A \) and \( B \) be subsets of \( \mathbb{Z}^2 \). The *translation* of \( A \) by \( x \) is denoted \( A_x \) and is defined as

\[
A_x = \{ c : c = a + x, \text{ for } a \in A \}
\]
The reflection of $B$, denoted $\hat{B}$, is defined as

$$\hat{B} = \{x: x = -b, \text{ for } b \in B\}.$$ 

The complement of $A$ is denoted $A^c$ and the difference of two sets $A$ and $B$ are denoted by $A - B$.

**Dilation**

Dilation of the object $A$ by the structuring element $B$ is given by

$$A \oplus B = \{I : B_\hat{i} \cap A \neq \emptyset\}$$

The result is a new set made up of all points generated by obtaining the reflection of $B$ about its origin and then shifting this reflection by $x$. Consider the example where $A$ is a rectangle and $B$ is a disc centered on the origin. (Note that if $B$ is not centered on the origin we will get a translation of the object as well.) Since $B$ is symmetric, $\hat{B} = B$. This definition becomes very intuitive when the structuring element $B$ is viewed as a convolution mask.

**Erosion**

Erosion of the object $A$ by a structuring element $B$ is given by

$$A \ominus B = \{I : B_\hat{i} \subseteq A\}$$

$$A \ominus B = \{x : B_x \subseteq A\}.$$ 

Dilation and erosion are duals of each other with respect to set complementation and reflection. That is,

$$(A \ominus B)^c = A^c \oplus \hat{B}.$$ 

To see this, consider first the left hand side:

$$(A \ominus B)^c = \{x : B_x \subseteq A\}^c.$$
Now, if $B_x$ is contained in $A$, then $B_x \cap A^c = \emptyset$ and so

$$(A \ominus B)^c = \{x: B_x \cap A^c = \emptyset\}^c$$

But the complement of the set of all $x$’s that satisfy $B_x \cap A^c = \emptyset$ is just the set of all $x$’s such that $B_x \cap A^c \neq \emptyset$

Thus $$(A \ominus B)^c = \{x: B_x \cap A^c \neq \emptyset\} = A^c \ominus \hat{B}.$$ 

Opening and closing:

Two very important transformations are opening and closing. Now intuitively, dilation expands an image object and erosion shrinks it. Opening generally smoothes a contour in an image, breaking narrow isthmuses and eliminating thin protrusions. Closing tends to narrow smooth sections of contours, fusing narrow breaks and long thin guls, eliminating small holes, and filling gaps in contours.

The opening of $A$ by $B$, denoted by $A \circ B$, is given by the erosion by $B$, followed by the dilation by $B$, that is

$$A \circ B = (A \ominus B) \oplus B.$$ 

Texture segmentation: It is nothing but partitions an image into a set of disjoint regions based on texture properties, so that each region is homogeneous with respect to certain texture characteristics. Results of segmentation can be applied to further image processing and analysis, for instance, to object recognition. Similar to classification, segmentation of texture also involves extracting features and deriving metrics to segregate textures. However, segmentation is generally more difficult than classification, since boundaries that separate different texture regions have to be detected in addition to recognizing texture in each region.

Texture segmentation could also be supervised or unsupervised depending on if prior knowledge about the image or texture class is available. [24] Supervised texture segmentation identifies and separates one or more regions that match texture properties shown in the training textures. Unsupervised segmentation has to first recover different texture classes from an image before separating them into regions.
Compared to the supervised case, the unsupervised segmentation is more flexible for real world applications despite that it is generally more computationally expensive. Partitioning an image into homogeneous regions is very useful in a variety of applications of pattern recognition and machine leaning.

3.2.3 Interpretation:

Image Interpretation is defined as the extraction of qualitative and quantitative information in the form of a map, about the shape, location, structure, function, quality, condition, relationship of and between objects, etc. by using human knowledge or experience. [26] This is the important step in image analysis in which the homogenous regions of an image gets constructed. Recognition of this region is through the way of understanding image data and describe it with suitable for a classifier. This description would generate a numeric feature or non-numeric syntactic description word which is shape of the region. [18]. In medical imaging, we have to see the object of interest, its shape and size of an object and compare it with the information available for a correct diagnosis for further treatment planning or guideline for surgery.

3.2.3.1. Region Identification: It is used for region description. One method for region identification is to label each region or each boundary with a unique integer. This identification is called as labeling or coloring or connected component labeling. The largest integer label gives the number of regions in an image. Region identification is necessary for region description. One of the many methods for region identification is to label each region (or each boundary) with a unique (integer) number; such identification is called labeling or coloring, also connected component labeling. Goal of segmentation was to achieve complete segmentation, now, the regions must be labeled.

\[
R_b^c = \bigcup_{i=1, i \neq b}^{m} R_i
\]

Different algorithms are developed for this purpose.
4-neighborhood and 8-neighborhood region identification:-

1. **First Pass**: Search the entire image R row by row and assign a non-zero value \( v \) to each non-zero pixel \( R(i, j) \). The value \( v \) is chosen according to the labels of the pixel’s neighbors.
   - If all the neighbors are background pixels \( R(i, j) \) is assigned a new unused label.
   - If there is just one neighboring pixel with a non-zero label, assign this label to the pixel \( R(i, j) \).
   - If there is more than one non-zero pixel among the neighbors, assign label to any one of labeled pixel. If collision occurs for same label then store these two labels in equivalent two tables.

2. **Second Pass**: All of the region pixels were labeled during the first pass but some regions have pixels with different labels due to label collision. The whole image is scanned again and pixels are re-labeled using the equivalence table information.

   But the disadvantage of this algorithm is for label collision. It is difficult to detect U-shaped object and all labels should be unique in step 2.

**Algorithm 2: Region identification in run length encoded data**

1. **First Pass**: Use a new label for each continuous run in first image row that is not part of background.

2. **Second Pass**: For the second and subsequent rows, compare partitions of runs.
   - If run length does not neighbor (in 4 or 8 sense) any run previous row, assign a new label.
   - If a run neighbor precisely one run in the previous row, assign its new label run.
   - If the new run neighbors more than one run in the previous row a label collision has occurred.

3 **Second Pass**: Search the image row by row and re-label the image according to the equivalence table.
Algorithm 3: Quad tree region identification:-

1. **First pass:** Search quad tree nodes in a given order i.e. Root and in NW, NE, SW, SE directions. Whenever an unordered leaf node is entered, a new label is assign to it. Then search for neighboring leaf nodes in the E and S directions plus SE in 8-connectivity If those leaves are non-zero and have not yet been labeled, assign a label of the node from which the search started. If the neighboring leaf node has already been labeled, store the collision information in an equivalence table.

2. Repeat step 1 until the whole tree has been searched.

3. **Second Pass:** Re-label the leaf nodes of the quad tree according to the equivalence table.

3.2.3.2. Shape Representation:

Defining the shape of an object can prove to be very difficult. Shape is usually represented verbally or in figures [25]. Shape description methods can be characterized from different points of view-

- Input representation form: Object description can be based on boundaries or on more complex knowledge of whole regions
- Object reconstruction ability: That is, whether an object's shape can or cannot be reconstructed from the description.
- Incomplete shape recognition ability: That is, to what extent an object's shape can be recognized from the description if objects are occluded and only partial shape information is available.
- Local/global description character: Global descriptors can only be used if complete object data are available for analysis. Local descriptors describe local object properties using partial information about the objects. Thus, local descriptors can be used for description of occluded objects.
- Mathematical and heuristic techniques: A typical mathematical technique is shape description based on the Fourier transform. A representative heuristic method may be elongatedness.
- Statistical or syntactic object description.
• A robustness of description to translation, rotation, and scale transformations: Shape description properties in different resolutions.

It can be represented in many ways like counter based and region based. In counter based region borders must be in some mathematical form. The rectangular function X(n) is defined as a function in terms of rectangular, polar & tangential the most common ways for counter based representation [26, 27] are

1. **Chain codes**: It describes an object by a sequence of unit size line segments with a given orientation. A commonly used chain codes employ 8 directions which can be coded by 3 bit code words. [13, 27] The chain code contains the start pixel address followed by a string of code words. Such code can be generalized by increasing the number of allowed direction vectors between successively boundary pixels. A limiting case is to encode the curvature of the contour as a function of contour length.

**Algorithm:-**

1. Start at any boundary pixel A
2. Find the nearest edge pixel and code its orientation. In case of a tie, choose with largest code value.
3. Continue until there are no boundary pixels.

2. **Simple Geometric Border Representation**

The following descriptors are mostly based on geometric properties [28] of described regions. Because of the discrete character of digital images, all of them are sensitive to image resolution.

• *Boundary length*

• *Curvature*
Bending energy

$$BE = \frac{1}{L} \sum_{k=1}^{L} c^2(k)$$  \[3.22\]
- **Signature**

![Signature Diagram](image)

**Figure 3.9: Signature (a) Construction (b) Signature for a circle and a triangle**

- **Chord**: Chord is a line joining any two points of the region boundary is a chord.

Let $b(x, y) = 1$ represents the contour points and $b(x, y) = 0$ represent all other points.

\[
 h(\Delta_x, \Delta_y) = \int \int b(I_x, y) b(I + \Delta_x, + \Delta_y) 
 d_x d_y \\
\]

--- 3.23

\[
 h(\Delta_x, \Delta_y) = \Sigma_i \Sigma_j b(i, j) b(i + \Delta_x, j + \Delta_y) \\
\]

--- 3.24

- **Rotation-independent radial distribution**

\[
 h_r(r) = \int \int h(\Delta_x, \Delta_y) r 
 d_x d_y \\
\]

--- 3.25

- The angular distribution $h_a(\theta)$ is independent of scale, while rotation causes a proportional offset

\[
 h_a(\theta) = \int \int h(\Delta_x, \Delta_y) 
 d_x d_y \\
\]

--- 3.26
• **Fourier transforms of boundaries**

Suppose C is a closed curve (boundary) in the complex plane. Traveling anti-clockwise along this curve keeping constant speed, a complex function \( z(t) \) is obtained, where \( t \) is a time variable.

\[
z(t) = \sum_n T_n e^{i nt}
\]

The coefficients \( T_n \) of the series are called the **Fourier descriptors** of the curve C. It is more useful to consider the curve distance \( s \) in comparison to time

\[
t = 2\pi s / L
\]

where \( L \) is the curve length. The Fourier descriptors \( T_n \) are given by

\[
T_n = \frac{1}{L} \int_0^L z(s) e^{-i(2\pi/L)ns} ds
\]

The descriptors are influenced by the curve shape and by the initial point of the curve. Working with digital image data, boundary co-ordinates are discrete and the function \( z(s) \) is not continuous. Assume that \( z(k) \) is a discrete version of \( z(s) \), where 4-connectivity is used to get a constant sampling interval; the descriptors \( T_n \) can be computed from the discrete Fourier transform.

\[
z(k) \leftarrow DFT \rightarrow T_n
\]
The Fourier descriptors can be invariant to translation and rotation if the coordinate system is appropriately chosen.

\[
a_n = \frac{1}{L-1} \sum_{m=1}^{L-1} I_m e^{-i(2\pi/(L-1))mn} \quad \text{--- 3.31}
\]

\[
b_n = \frac{1}{L-1} \sum_{m=1}^{L-1} y_m e^{-i(2\pi/(L-1))mn} \quad \text{--- 3.32}
\]

The coefficients \(a_n, b_n\) are not invariant, but after the transform

\[
r_n = \left( |a_n|^2 + |b_n|^2 \right)^{1/2} \quad \text{--- 3.33}
\]

\(r_n\) are translation and rotation invariant.

To achieve magnification invariance the descriptors \(w_n\) are used:

\[
w_n = r_n / r_1 \quad \text{--- 3.34}
\]

The first 10 -- 15 descriptors \(w_n\) are found to be sufficient for character description.

Figure 3.11: Fourier Descriptors of boundaries (a) Descriptors \(T_n\) (b) Descriptor \(S_n\)
A closed boundary can be represented as a function of angle tangents versus the distance between the boundary points from which the angles were determined

\[ a(l_k) = \varphi_k + u_k \quad \text{--- 3.35} \]

\[ u_k = 2\pi l_k / L \quad \text{--- 3.36} \]

The descriptor set is then

\[ S_n = \frac{1}{2\pi} \int_0^{2\pi} a(u)e^{-inu} du \quad \text{--- 3.37} \]

The high quality boundary shape representation obtained using only a few lower order coefficients is a favorable property common to Fourier descriptors.

- **Boundary description using segment sequences**

If the **segment type** is known for all segments, the boundary can be described as a chain of segment types, a code-word consisting of representatives of a type alphabet. A **polygonal representation** approximates a region by a polygon, the region being represented using its vertices. Polygonal representations are obtained as a result of simple boundary segmentation. Another method for determining the boundary vertices is a **tolerance interval approach** based on setting a maximum allowed difference \( e \).

![Figure 3.12: Tolerance Interval](image)
• Recursive boundary splitting

![Recursive Boundary splitting](image)

**Figure 3.13: Recursive Boundary splitting**

Boundary segmentation into segments of constant curvature or curve segmentation into circular arcs and straight lines is used. Segments are considered as primitives for syntactic shape recognition procedures.

![Structural description of chromosomes by a chain of boundary segments](image)

**Figure 3.14: Structural description of chromosomes by a chain of boundary segments**
code word: d, b, a, c, b, a, b, d, b, a, b, c, b, a, b

Sensitivity of shape descriptors to scale is also important if a curve is to be divided into segments -- a **scale-space approach** to curve segmentation. Only new segmentation points can appear at higher resolutions, and no existing segmentation points can disappear. Fine details of the curve disappear in pairs with increasing size.
of the Gaussian smoothing kernel, and two segmentation points always merge to form a closed contour showing that any segmentation point existing in coarse resolution must also exist in finer resolution. Moreover, the position of a segmentation point is most accurate in finest resolution and this position can be traced from coarse to fine resolution using the scale-space image.

A multi-scale curve description can be represented by an interval tree.

A multi-scale curve description can be represented by an interval tree.

Figure 3.15: Scale-space image: (a) Varying number and location of curve segmentation points as a function of scale, (b) curve representation by an interval tree

- **B-spline representation**

  Representation of curves using piecewise polynomial interpolation to obtain smooth curves is widely used in computer graphics. B-splines are piecewise polynomial curves whose shape is closely related to their control polygon - a chain of vertices giving a polygonal representation of a curve. B-splines of the third-order are most common because this is the lowest order which includes the change of curvature.[29,30] Splines have very good representation properties and are easy to compute:
- Firstly, they change their shape less than their control polygon, and do not oscillate between sampling points as many other representations do. A spline curve is always positioned inside a convex \(n+1\)-polygon for a B-spline of the \(n\)-th order.

- Secondly, the interpolation is local in character. If a control polygon vertex changes its position, a resulting change of the spline curve will occur only in a small neighborhood of that vertex.

- Thirdly, methods of matching region boundaries represented by splines to image data are based on a direct search of original image data.

Each part of a cubic B-spline curve is a third-order polynomial, meaning that it and its first and second derivatives are continuous. B-splines are given by

\[
X(s) = \sum_{i=0}^{n+1} v_i B_i(s)
\]

--- 3.38
\[ V_0 = 2v_1 - v_2 \quad --- 3.39 \]

\[ V_{n+1} = 2v_n - v_{n-1} \quad --- 3.40 \]

\[ C_0(t) = \frac{t^3}{6} \quad --- 3.41 \]

\[ C_1(t) = \frac{-3t^3 + 3t^2 + 3t + 1}{6} \quad --- 3.42 \]

\[ C_2(t) = \frac{3t^3 - 6t^2 + 4}{6} \quad --- 3.43 \]

\[ C_3(t) = \frac{-t^3 + 3t^2 - 3t + 1}{6} \quad --- 3.44 \]

\[ x(s) = c_{i-1,3}(s)v_{i-1} + c_{i,2}(s)v_i + c_{i+1,1}(s)v_{i+1} + c_{i+2,0}(s)v_{i+2} + c_{i,j}(s) = c_{j}(s-i) \quad --- 3.45 \]

\[ c_{i,j}(s) = c_{j}(s-i) \quad i = 0, \ldots, n+1; \quad j = 0, 1, 2, 3 \quad --- 3.46 \]

\[ x(s) = c_{3}(s-i)v_{i-1} + c_{2}(s-i)v_i + c_{1}(s-i)v_{i+1} + c_0v_{i+2} \quad --- 3.47 \]

\[ x(5) = C_3(0)v_4 + C_2(0)v_5 + C_1(0)v_6 = \frac{1}{6}v_4 + \frac{4}{6}v_5 + \frac{1}{6}v_6 \quad --- 3.48 \]

\[ x(5) = C_3(0.7)v_6 + C_2(0.7)v_7 + C_1(0.7)v_8 + C_0(0.7)v_9 \quad --- 3.49 \]

**Other contour-based shape description approaches**

Many other methods and approaches can be used to describe two-dimensional curves and contours. Some are
• The **Hough transform** has excellent shape description abilities depending upon peak, line or circle detection.

• Region-based shape description using **statistical moments**.

• **Fractal** approach to shape is gaining attention in image shape description.

• **Mathematical morphology** can be used for shape description, typically in connection with region skeleton construction.

• **Neural networks** can be used to recognize shapes in raw boundary representations directly. Contour sequences of noiseless reference shapes are used for training, and noisy data are used in later training stages to increase robustness; effective representations of closed planar shapes result.

### 3.4.4 Object Recognition:

The classical problem in computer vision, image processing and machine vision is that of determining whether or not the image data contains some specific object, feature, or activity. This task can normally be solved robustly and without effort by a human, but is still not satisfactory solved in computer vision for the general case: arbitrary objects in arbitrary situations. The existing methods for dealing with this problem can at best solve it only for specific objects, such as simple geometric objects (e.g. polyhedrons), human faces, printed or hand-written characters, vehicles or tumors in specific situations, typically described in terms of well-defined illumination, background, and pose of the object relative to the camera. Several methods are discussed below and we apply as per our requirement this on ultrasound image for ROI detection. We will discuss this in next chapter. Different varieties of the recognition problem [29] are described below:

**Recognition:** one or several pre-specified or learned objects or object classes can be recognized, usually together with their 2D positions in the image or 3D poses in the scene.
Identification: An individual instance of an object is recognized. Examples: identification of a specific person face or fingerprint, or identification of a specific vehicle or identification of tumors.

Detection: the image data is scanned for a specific condition. Examples: detection of possible abnormal cells or tissues in medical images or detection of a vehicle in an automatic road toll system. Detection based on relatively simple and fast computations is sometimes used for finding smaller regions of interesting image data which can be further analyzed by more computationally demanding techniques to produce a correct interpretation.

Several specialized tasks based on recognition exist, such as:

Content-based image retrieval: finding all images in a larger set of images which have a specific content. The content can be specified in different ways, for example in terms of similarity relative a target image (give me all images similar to image X), or in terms of high-level search criteria given as text input

Pose estimation: estimating the position or orientation of a specific object relative to the camera. An example application for this technique would be assisting a robot arm in retrieving objects from a conveyor belt in an assembly line situation.

Optical character recognition (or OCR): identifying characters in images of printed or handwritten text, usually with a view to encoding the text in a format more amenable to editing or indexing (e.g. ASCII).

Object Detection Techniques

With this basic understanding of object detection we found some basic techniques that could be used for detecting objects in a scene. In words we will describe the process of finding an object using these techniques.

- Pattern Matching Using Correlation

The goal of the pattern matching technique is to find every instance of a specific object in the scene by applying a special template. [31, 32] The template is an image of the object of interest. This template is just a grouping of pixel values that correlate with the object of interest.
To accomplish this location task, the object mask is applied to the image in such a way that groupings of pixels that correlate with the template will be close to white while groups of pixels that do not correlate with the template will be close to black. The figure below shows how the template is applied to the image.

The image algebra to accomplish this template application is

\[ C := a \oplus t. \]  

In the equation above, \( c \) is the output image, \( a \) is the source image, and \( t \) is the template represented by pixel values \( p \). \( t \) is defined as:

\[
t(x, y)(u, v) = \begin{cases} 
0 & \text{if } -(m-1) \leq u - i \leq m-1 \\
(p(u-x, v-y)) & \text{otherwise } (n-1) \leq v - y \leq n-1
\end{cases}
\]

The locations of the objects are white. However, there are other objects in the scene that almost detect as ROI. While these objects do not have as strong a match as the tanks themselves (as can be seen by the relative whiteness of the pixels at those locations), they could be mistakenly identified as objects. To make that location
clearer, thresholding on the pixel values is applied. This type of threshold turns all pixels that are not white enough to black. The locations of these objects are now very clearly seen. This same type of process can be performed in the frequency domain.

Pattern matching does have some limitations. It requires an accurate image of the desired object as it is likely to appear in the image. Furthermore, changes in the object's orientation and size will adversely affect performance. Finally, if the template image was taken in different lighting than the source image, then the pixel values are less likely to line up, even for the object of interest.

**Using Distance Sets for Shape Recognition**

Using distance sets for shape recognition is a technique is called applying a distance set shape filter. [33] Like a band-pass filter only retains signal components of certain frequencies, the distance set shape filter will only retain groups of pixels with overall distance sets close to the prototype shape.

**Bayes classifier**

A naive Bayes classifier (also known as Idiot's Bayes) is a simple probabilistic classifier based on applying Bayes theorem with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be independent feature model.

Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without believing in Bayesian probability or using any Bayesian methods.

Abstractly, the probability model for a classifier is a conditional model

\[ p(c|F_1, \ldots, F_n) \]  

over a dependent class variable \( C \) with a small number of outcomes or classes, conditional on several feature variables \( F_1 \) through \( F_n \). The problem is that if the number of features \( n \) is large or when a feature can take on a large number of values,
then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable.

Using Bayes' theorem, we write

\[
p(c|F_1---, F_n) = \frac{p(c) p(F_1---, F_n \setminus c)}{p(F_1---, F_n \setminus c)}
\]

--- 3.53

In practice we are only interested in the numerator of that fraction, since the denominator does not depend on \( C \) and the values of the features \( F_i \) are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model.

\[
p(c, F_1---, F_n)
\]

--- 3.54

which can be rewritten as follows, using repeated applications of the definition of conditional probability

\[
p(c, F_1---, F_n)
= p(c) p(F_1---, F_n \setminus c)
= p(C) p(F_1 \setminus C), p(F_2---, F_n \setminus C, F_i)
= p(C) p(F_1 \setminus C), p(F_2 \setminus C, F_i) p(F_3---, F_n \setminus C, F_i, F_2)
= p(C) p(F_1 \setminus C), p(F_2 \setminus C, F_i) p(F_3 \setminus C, F_i, F_2) p(F_4---, F_n \setminus C, F_i, F_2)
\]

--- 3.55

and so forth. Now the "naive" conditional independence assumptions come into play: assume that each feature \( F_i \) is conditionally independent of every other feature \( F_j \) for \( j \neq i \). This means that

\[
p(F_i|C,F_j) = p(F_i|C)
\]

--- 3.56

And so the joint model can be expressed as
\[ p(C, F_1, \ldots, F_n) = p(C) p(F_1|C)(F_2|C)(F_3|C) \quad \text{--- 3.57} \]

\[ = p(C) \prod_{i=1}^{n} p(F_i|C) \quad \text{--- 3.58} \]

This means that under the above independence assumptions, the conditional distribution over the class variable \( C \) can be expressed like this:

\[ = p(C|F_1, \ldots, F_n) \frac{1}{Z} p(C) \prod_{i=1}^{n} p(F_i|C) \quad \text{--- 3.59} \]

Where \( Z \) is a scaling factor dependent only on \( F_1, \ldots, F_n \), i.e. a constant if the values of the feature variables are known.

Models of this form are much more manageable, since they factor into a so-called \textit{class prior} \( p(C) \) and independent probability distributions \( p(F_i|C) \). If there are \( k \) classes and if a model for \( p(F_i) \) can be expressed in terms of \( r \) parameters, then the corresponding naive Bayes model has \((k - 1) + n r k\) parameters. In practice, often \( k = 2 \) (binary classification) and \( r = 1 \) (Bernoulli variables as features) are common, and so the total number of parameters of the naive Bayes model is \( 2n + 1 \), where \( n \) is the number of binary features used for prediction.
References

23 J. Serra Image Analysis & Mathematical Morphology, Academic Press March 1982
27 Tou J.T. “Pattern Recognition Principles ” Addison Wesley Reading Mass
31 Amit, Yali; 2D Object Detection and Recognition Models, Algorithms, and Networks; The MIT Press, Cambridge Massachusetts, 2002
32 Ritter, Gerhard & Wilson, Joseph; Computer Vision Algorithms in Image Algebra; CRC Press LLC, Boca Ratton Florida, 2001