

**NATIONAL INSTITUTE OF TECHNOLOGY,**  
**ROURKELA**



**Submission of project report for the evaluation of the final year  
project**

Titled

**Recognition of Electrical & Electronics Components**

Under the guidance of:

Prof. S. Meher

Dept. of Electronics & Communication Engineering  
NIT, Rourkela.

Submitted by:  
Amiteshwar Kumar Abhijeet  
10307022  
B.Tech IV  
Dept. of Electronics & Communication  
Engg.  
NIT, Rourkela

Submitted by:  
Pradeep Kumar Chachan  
10307030  
B.Tech IV  
Dept. of Electronics & Communication  
Engg.  
NIT, Rourkela

**NATIONAL INSTITUTE OF TECHNOLOGY,**  
**ROURKELA**



**Submission of project report for the evaluation of the final year  
project**

Titled

**Recognition of Electrical & Electronics Components**

Under the guidance of:

Prof. S. Meher

Dept. of Electronics & Communication Engineering  
NIT, Rourkela.

Submitted by:

Amiteshwar Kumar Abhijeet

10307022

B.Tech IV

Dept. of Electronics & Communication

Engg.

NIT, Rourkela

Submitted by:

Pradeep Kumar Chachan

10307030

B.Tech IV

Dept. of Electronics & Communication

Engg.

NIT, Rourkela



**National Institute of Technology  
Rourkela**

**CERTIFICATE**

This is to certify that the thesis entitled, “RECOGNITION OF ELECTRICAL & ELECTRONICS COMPONENTS” submitted by Sri AMITESHWAR KUMAR ABHIJEET and Sri PRADEEP KUMAR CHACHAN in partial fulfillments for the requirements for the award of Bachelor of Technology Degree in Electronics & Instrumentation Engg. at National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Date:

**Dr. S. Meher**  
Dept. of Electronics &  
Communication Engg.  
NIT, Rourkela.

## **ACKNOWLEDGEMENT**

I wish to express my deep sense of gratitude and indebtedness to Dr.S.Meher, Department of Electronics & Communication Engg. , N.I.T Rourkela for introducing the present topic and for his inspiring guidance, constructive criticism and valuable suggestion throughout this project work.

I would like to express my gratitude to Dr. G. Panda (Head of the Department) for his constant support and encouragement. I am also thankful to all staff members of Department of Electronics & Communication Engg. NIT Rourkela.

04<sup>TH</sup> May 2007

Amiteshwar Kumar Abhijeet

Roll: 10307022

Pradeep Kumar Chachan

Roll: 10307030

# CONTENTS

Page No

<b>Chapter 1</b>	<b>GENERAL INTRODUCTION</b>	
1.1	Introduction	1-2
<b>Chapter 2</b>	<b>FUNDAMENTALS OF IMAGE PROCESSING</b>	
2.1	Introduction	3
2.2	Image Definition	3-4
2.3	Image Types	4-9
	2.3.1 Intensity Image	
	2.3.2 Binary Image	
	2.3.3 Indexed Image	
	2.3.4 RGB Image	
2.4	Image Segmentation	9-10
2.5	Thinning	10-12
2.6	Morphological Image Processing	12-15
	2.6.1 Hit & Miss Transform	
<b>Chapter 3</b>	<b>OBJECT RECOGNITION TECHNIQUES</b>	
3.1	Introduction	16
3.2	Pattern & Pattern Classes	16-17
3.3	Recognition based on Decision-Theoretic Methods	17-20
	3.3.1 Forming Pattern Vector	
	3.3.2 Matching	
	3.3.3 Pattern Matching Using Minimum Distance Classifiers	
	3.3.4 Matching by Correlation	
3.4	Structural Recognition	20
	3.4.1 String Matching	
<b>Chapter 4</b>	<b>ELECTRICAL &amp; ELECTRONICS COMPONENTS</b>	
4.1	Capacitors	21-22
4.2	Diodes	22-23
4.3	Resistors	23-24
4.4	Transistors	25-28
<b>Chapter 5</b>	<b>EXPERIMENTS &amp; RESULTS</b>	
5.1	Quantitative Approach	29-34
5.2	By finding Roundness of Objects	34-36
5.3	Matching by Correlation	36-38
<b>Chapter 6</b>	<b>CONCLUSION</b>	39
	<b>REFERENCES</b>	40

# Chapter 1

**GENERAL INTRODUCTION**

## **1.1 INTRODUCTION**

Recognition or more specifically Pattern or Object recognition is a typical characteristic of human beings and other living organisms. The term pattern or object means something that is set as an idea to be imitated. For example, in our childhood a shape 'A' is shown to us and we are asked to imitate that. So the shape is the ideal one. On the other hand, if what we produce or draw obeying that instruction is close to that shape, our teacher identifies that as 'A'. This identification is called recognition and the shapes we draw (that is object we made) may be termed as patterns. Thus, the pattern recognition means identification of the real object. Recognition should, therefore, be preceded by the development of the concept of the ideal or model or prototype. This process is called Learning. In most real life problems no ideal example is available. In that case, the concept of ideal is abstracted from many near perfect examples. Under this notion learning is of two types : supervised learning if appropriate label is attached to each of these examples ; and unsupervised learning if no labeling is available. It is obvious that for recognizing an object we must receive or sense some information or features from that object. Based on these features we assign object being considered to one of the following classes each of which represents a pattern. Hence classification is the actual task to be done and we call it recognition. If the classes are labelled with particular patterns. Sometimes, learning and recognition works together; outcome of recognition process modifies knowledge about the pattern classes. Unsupervised methods usually fall in this category.

Usually recognition process deals with physical items. Thus, it depends on features from the items. Such process is called sensory recognition. There is another kind of recognition process which deals with abstract items such as an idea, theory, a solution to a problem, a philosophical question etc. this may be termed as conceptual recognition.

Since objects are assigned to classes based on invariant features associated with them, it is obvious that objects of same class possess similar features and those which belong to different classes possess different features. Therefore, the set of features that distinguishes objects of different classes and is common to objects of the same classes is the key for classification and recognition. Identifying such a minimal feature set is an important step in the process of recognition. The process is called feature selection. Another major step is designing the decision process. The decision procedure should be optimum in a sense that the classification error must be minimum. This is developed usually through learning. That means given a set of training data, a set of decision rules is to be devised so that the training data are separated into given sets of classes in an optimum way. Note that each training data is feature vector along with the class level which the object actually belongs to. This is an example of supervised learning. Unsupervised learning may be exemplified by clustering where feature vectors are

supplied along with some indirect information about the class. This may include number of classes, interclass distance and interclass distance among feature vectors. Decision methodology adopt one of two major approaches : mathematical and heuristic. Mathematical approaches are based on the given set of models and decision rules are devised satisfying optimal criteria of classification. On the other hand when no such model is available decision rules are designed using human intuition and experience for a specific problem. The mathematical approaches includes deterministic, statistical, fuzzy set theoretic and syntactic recognition; while heuristic methods include graph matching, tree searching etc.

Object recognition is used in many computer vision applications. It is particularly useful for industrial inspection tasks, where often an image of an object must be aligned with a model of the object. In most cases, the model of the object is generated from an image of the object. A large number of object recognition strategies exist. In most 2D matching object recognition implementations the search is usually done in a coarse-to-fine manner. The simplest class of object recognition methods is based on the gray values of the model and image itself and uses normalized cross correlation. A more complex class of object recognition methods does not use the gray values of the model or object itself, but uses the object's edges for matching.

Central idea to the theme of recognition is the concept of learning from sample patterns. We call objects as patterns. Approaches to computerized pattern recognition may be divided into two principle areas: decision-theoretic and structural. The first category deals with patterns described using quantitative descriptors, such as length, area, texture etc. The second category deals with patterns best represented by symbolic information, such as strings under structural recognition techniques.

# Chapter 2

# Fundamentals of Image Processing

## **2.1 Introduction**

Modern digital technology has made it possible to manipulate multi-dimensional signals with systems that range from simple digital circuits to advanced parallel computers. The goal of this manipulation can be divided into three categories:

- Image Processing
- Image Analysis
- Image Understanding

We will focus on the fundamental concepts of image processing. Space does not permit us to make more than a few introductory remarks about image analysis. We will restrict ourselves

to two-dimensional (2D) image processing although most of the concepts and techniques that are to be described can be extended easily to three or more dimensions.

An image defined in the “real world” is considered to be a function of two real variables, for example,  $a(x,y)$  with  $a$  as the amplitude (e.g. brightness) of the image at the real coordinate position  $(x,y)$ . An image may be considered to contain sub-images sometimes referred to as regions-of-interest, ROIs, or simply regions. This concept reflects the fact that images frequently contain collections of objects each of which can be the basis for a region. In a sophisticated image processing system it should be possible to apply specific image processing operations to selected regions. Thus one part of an image (region) might be processed to suppress motion blur while another part might be processed to improve color rendition. The amplitudes of a given image will almost always be either real numbers or integer numbers. The latter is usually a result of a quantization process that converts a continuous range (say, between 0 and 100%) to a discrete number of levels. In certain image-forming processes, however, the signal may involve photon counting which implies that the amplitude would be inherently quantized. In other image forming procedures, such as magnetic resonance imaging, the direct physical measurement yields a complex number in the form of a real magnitude and a real phase.

## **2.2 Digital Image Definition**

A digital image  $a[m,n]$  described in a 2D discrete space is derived from an analog image  $a(x,y)$  in a 2D continuous space through a sampling process that is frequently referred to as digitization. For now we will look at some basic definitions associated with the digital image. The effect of digitization is shown in Figure.

The 2D continuous image  $a(x,y)$  is divided into  $N$  rows and  $M$  columns. The intersection of a row and a column is termed a pixel. The value assigned to the integer coordinates  $[m,n]$  with  $\{m=0,1,2,\dots,M-1\}$  and  $\{n=0,1,2,\dots,N-1\}$  is  $a[m,n]$ . In fact, in most cases  $a(x,y)$ —which we might consider to be the physical signal that impinges on the face of a 2D sensor—is actually a function of many variables including depth ( $z$ ), color ( $l$ ), and time ( $t$ ).

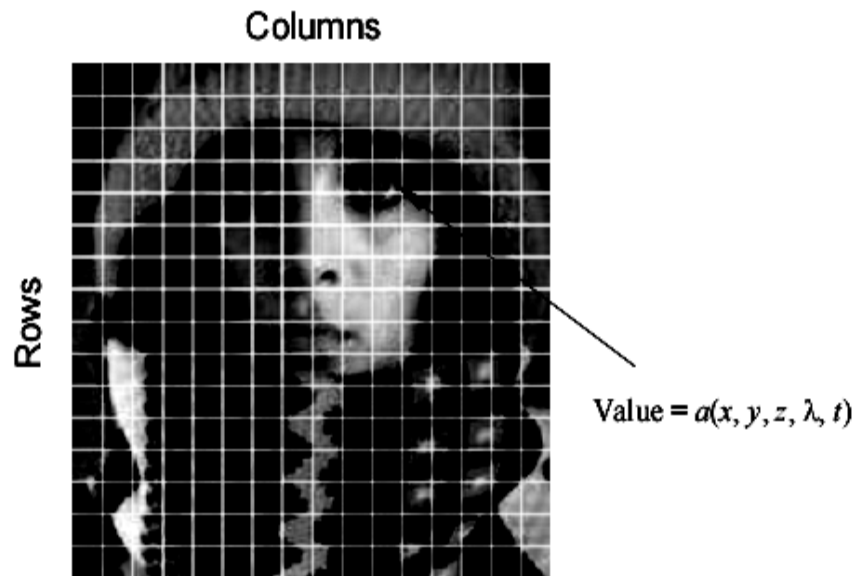


Figure represents Digitization of a continuous image. The pixel at coordinates  $[m=10, n=3]$  has the integer brightness value 110.

The image shown in Figure above has been divided into  $N = 16$  rows and  $M = 16$  columns. The value assigned to every pixel is the average brightness in the pixel rounded to the nearest integer value. The process of representing the amplitude of the 2D signal at a given coordinate as an integer value with  $L$  different gray levels is usually referred to as amplitude quantization or simply quantization.

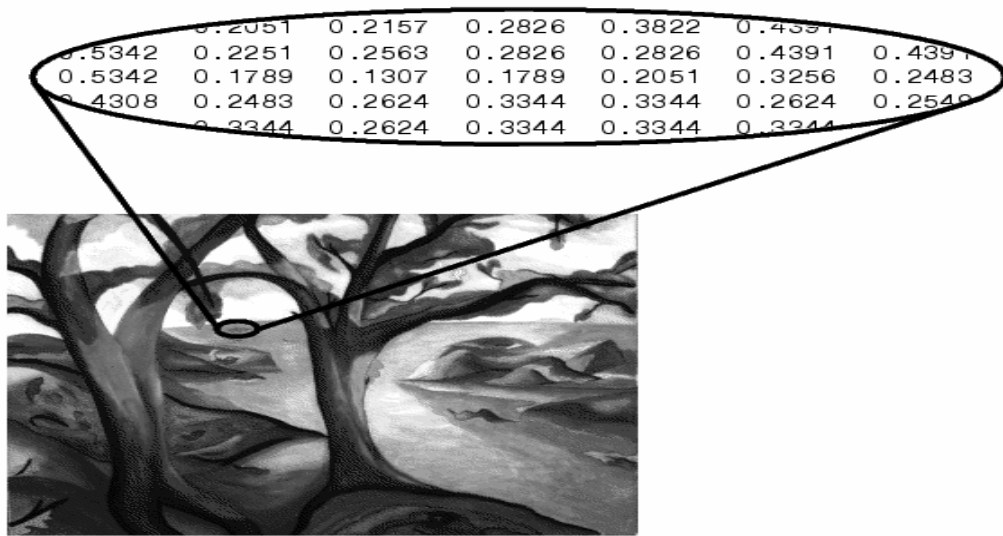
## 2.3 Image Types

The toolbox supports four types of images:

- Intensity images
- Binary images
- Indexed images
- RGB images

**2.3.1 Intensity Images :** An intensity image is a data matrix whose values have been scaled to represent intensities. When the elements of an intensity image are of class uint8, or class uint16, they have integer values in the range  $[0, 255]$  and  $[65535]$ , respectively. If the image is of class double, the values are floating-point numbers. Values of scaled, class double intensity images are in the range  $[0,1]$  by convention.

This figure depicts an intensity image of class double.

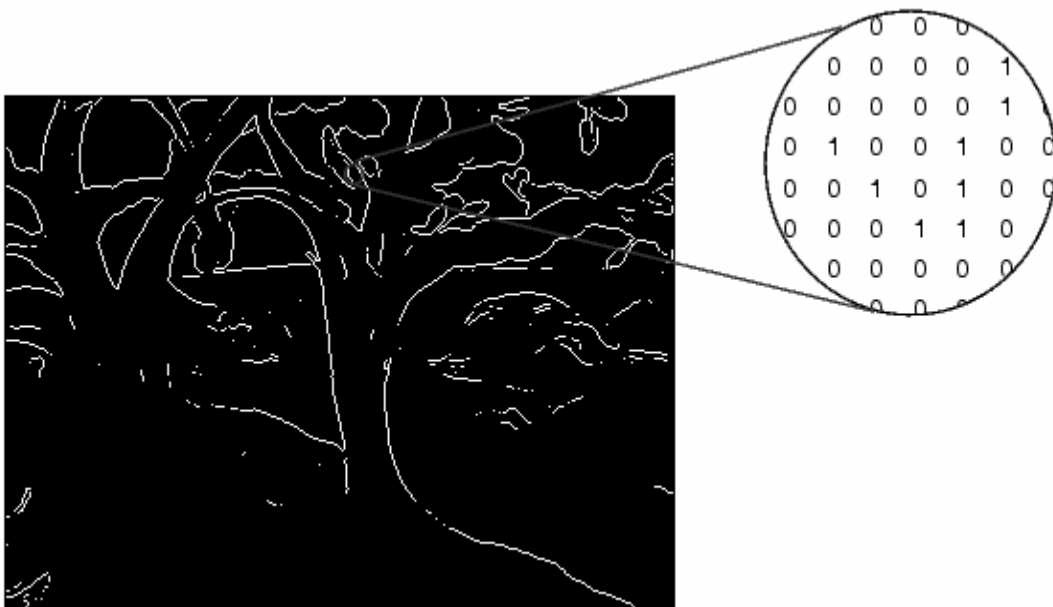


**2.3.2 Binary Images :** In a binary image, each pixel assumes one of only two discrete values. Essentially, these two values correspond to on and off. A binary image is stored as a logical array of 0's (off pixels) and 1's (on pixels). Pixels in a Binary Image Have Two Possible Values: 0 or 1. An array of 0s and 1s whose values are of data class, say, uint8, is not considered a binary image in MATLAB. A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, a logical array B can be created using the statement

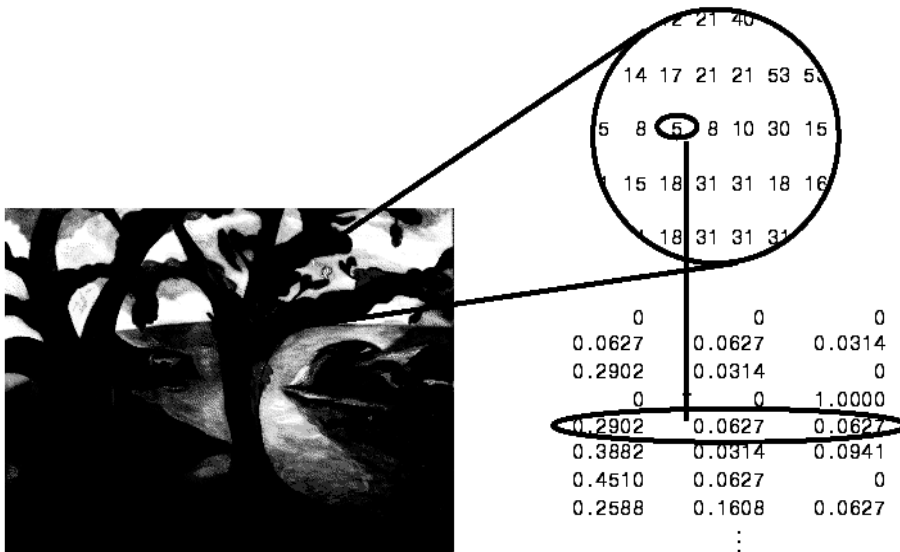
$$B = \text{logical}(A)$$

If A contains elements other than 0s and 1s, use of logical function converts all nonzero quantities to logical 1s and all entries with with value 0 to logical 0s.using relational and logical operators also creates logical arrays.

The figure below depicts a binary image :



**2.3.3 Indexed Images :** An indexed image has two components: a data matrix of integers,  $X$ , and a color map matrix,  $map$ . Matrix  $map$  is an  $m \times 3$  array of class double containing floating point values in the range  $[0,1]$ . The length  $m$  of the map is equal to the number of colors it defines. Each row of  $map$  specifies the red, green and blue components of a single color. An indexed image uses “direct mapping” of pixel intensity values to color map values. The color of each image pixel is determined by using the corresponding value of  $X$  as an index into  $map$ . The value 1 points to the first row in  $map$ , the value 2 points to the second row, and so on. A colormap is often stored with an indexed image and is automatically loaded with the image when you use the `imread` function. The color of each pixel is determined by using the corresponding value of integer matrix  $X$  as pointer into  $map$ . If  $X$  is of class double, then all of its components with values less than or equal to 1 point to the first row in  $map$ , all components with value 2 point to the second row, and so on. If  $X$  is of class `uint8` or `uint16`, then all components with value 0 point to the first row in the map, all components with value 1 point to the second row, and so on. The offset is also used in graphics file formats, to maximize the number of colors that can be supported. In the image above, the image matrix is of class double. Because there is no offset, the value 5 points to the fifth row of the colormap. The figure below illustrates the structure of an indexed image. The pixels in the image are represented by integers, which are pointers (indices) to color values stored in the colormap.

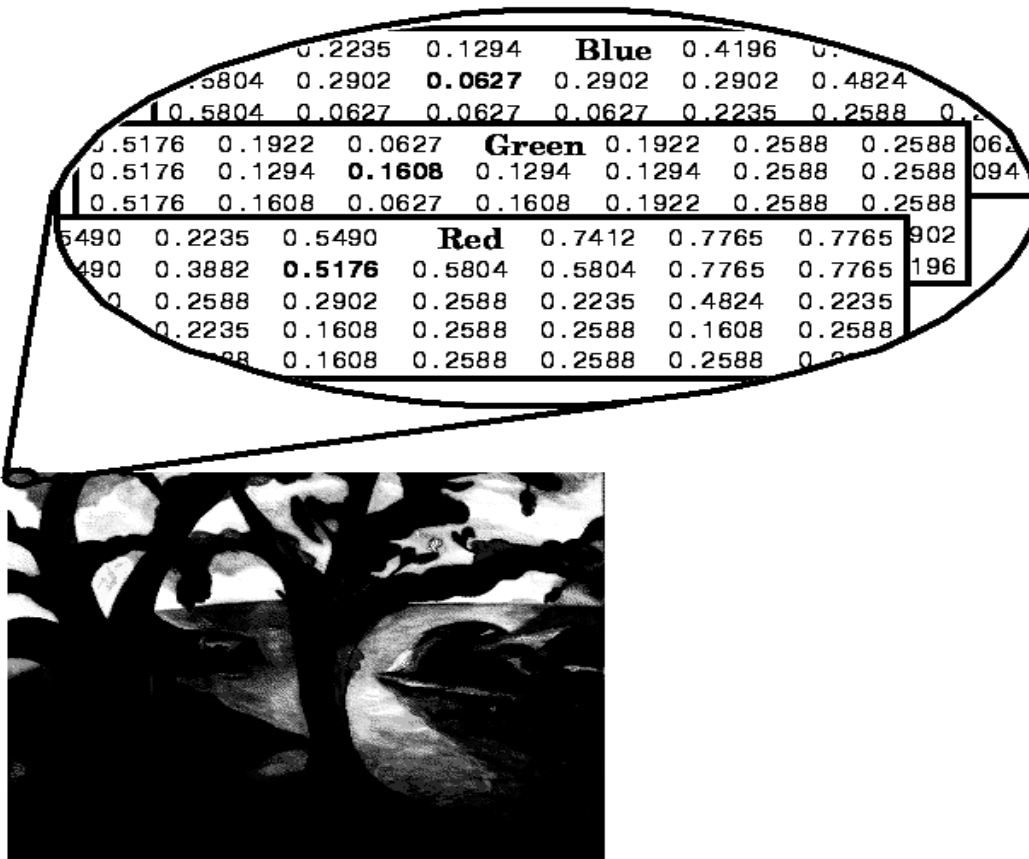


**2.3.4 RGB Images :** An RGB image, sometimes referred to as a "truecolor" image. RGB images do not use a palette. It is an  $M \times N \times 3$  array of color pixels, where each color pixel is a triplet corresponding to the red, green, and blue components of an RGB image at a specific spatial location. An RGB image may be viewed as a “stack” of three grayscale images that,

when fed into the red, green and blue inputs of a color monitor, produce a color image on the screen. By convention, the three images forming an RGB color image are referred to as the red, green and blue component images. The data class of the component images determines their range of values. If an RGB image of class double, the range of values is [0, 1]. Similarly, the range of values is [0, 255] or [0, 65535] for RGB images of class uint8 or uint16 respectively. The number of bits used to represent the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8-bit image, the corresponding RGB image is said to be 24 bits deep. Generally the number of bits in all the component images is the same. In this case the number of possible colors in an RGB image is  $(2^b)^3$ , where b is the number of bits in each component images. For the 8-bit case, the number is 1677216 colors.

A pixel whose color components are (0,0,0) is displayed as black, and a pixel whose color components are (1,1,1) is displayed as white. The three color components for each pixel are stored along the third dimension of the data array. For example, the red, green, and blue color components of the pixel (10,5) are stored in RGB(10,5,1), RGB(10,5,2), and RGB(10,5,3), respectively.

The figure below shows an RGB image of class double :



## 2.4 Image Segmentation

Image segmentation refers to the decomposition of an image into its components. It is a key step in image analysis. Segmentation accuracy determines the eventual success or failure of computerized analysis procedures. For this reason, considerable care should be taken to improve the probability of rugged segmentation. Segmentation is an important part of practically any automated image recognition system, because it is at this moment that one extracts the interesting objects, for further processing such as description or recognition. Segmentation of an image is in practice the classification of each image pixel to one of the image parts. If the goal is to recognize black characters, on a grey background, pixels can be classified as belonging to the background or as belonging to the characters: the image is composed of regions which are in only two distinct greyvalue ranges, dark text on lighter background. The greylevel histogram, viz. the probability distribution of the greyvalues, has two separated peaks, i.e. is clearly bimodal. In such a case, the segmentation, i.e. the choice of a greylevel threshold to separate the peaks, is trivial. The same technique could be used if there were more than two clearly separated peaks. A variety of techniques for automatic threshold selection exists. A relatively successful method for certain applications where it is suggested that a modified histogram is employed by using only pixels with a small gradient magnitude, i.e. pixels which are not in the region of the boundaries between object and background.

In many cases, segmentation on the basis of the greyvalue alone is not efficient. Other features like colour, texture, gradient magnitude or orientation, measure of a template match etc., can be put to use. This produces a mapping of a pixel into a point in an  $n$ -dimensional feature space, defined by the vector of its feature values. The problem is then reduced to partitioning the feature space into separate clusters, a general pattern recognition problem that is discussed in the literature.

The two halves of the image labelled ``original" contain peaks of random height, but of different shape: in the bottom half, the peaks are steeper than in the top half. The greylevel histogram of the original image is clearly not bimodal. We create two different morphological features, and show them in the images labelled ``feature1" and ``feature2". We now enter, for all pixels, the greyvalue of feature1 against that of feature2, into a two-dimensional histogram (``feature space"); in this representation it is easy to distinguish the two clusters.

## **2.5 Thinning**

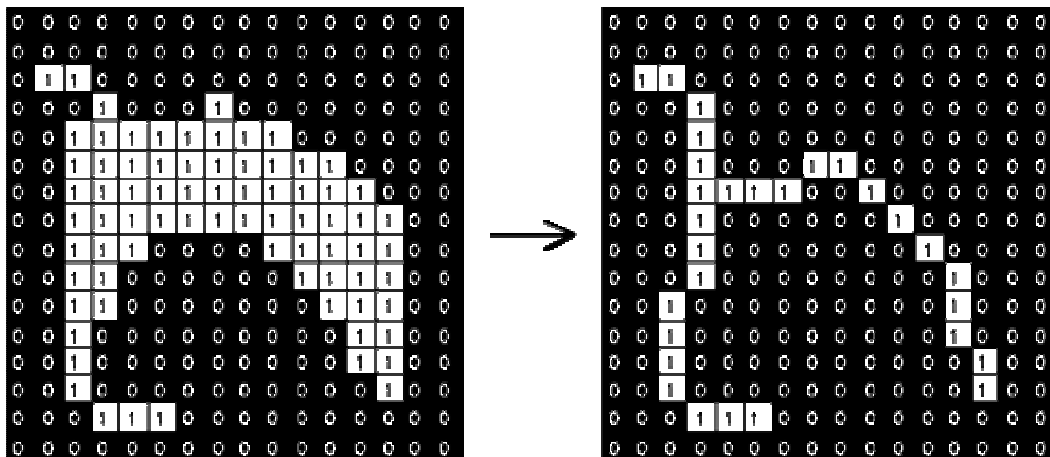
Thinning is a morphological operation that is used to remove selected foreground pixels from binary images, somewhat like erosion or opening. It can be used for several applications, but is particularly useful for skeletonization. In this mode it is commonly used to tidy up the output of

edge detectors by reducing all lines to single pixel thickness. Thinning is normally only applied to binary images, and produces another binary image as output. The thinning operation is related to the hit-and-miss transform, and so it is helpful to have an understanding of that operator before reading on. Like other morphological operators, the behavior of the thinning operation is determined by a structuring element. The binary structuring elements used for thinning are of the extended type described under the hit-and-miss transform (*i.e.* they can contain both ones and zeros).

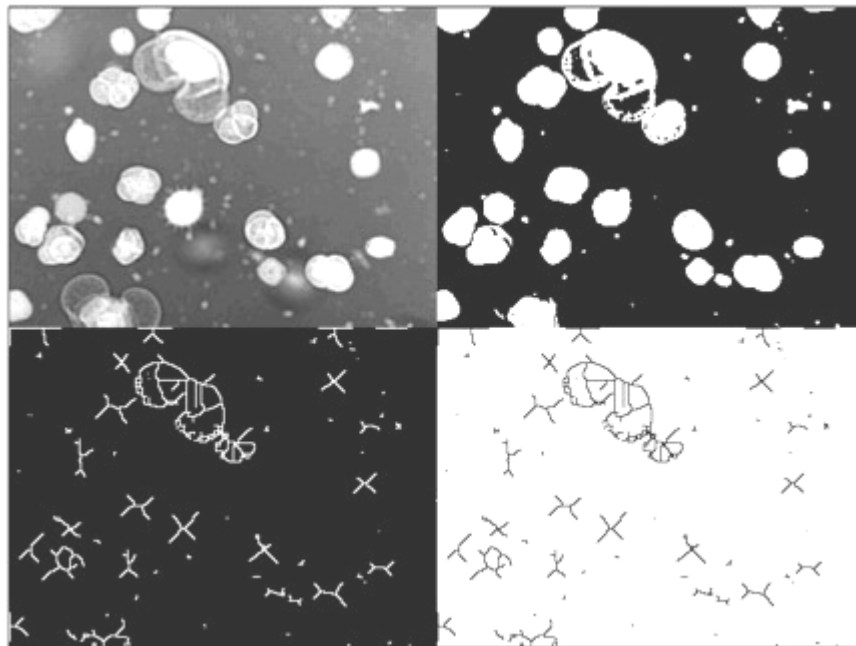
In everyday terms, the thinning operation is calculated by translating the origin of the structuring element to each possible pixel position in the image, and at each such position comparing it with the underlying image pixels. If the foreground and background pixels in the structuring element *exactly match* foreground and background pixels in the image, then the image pixel underneath the origin of the structuring element is set to background (zero). Otherwise it is left unchanged. Note that the structuring element must always have a one or a blank at its origin if it is to have any effect. The choice of structuring element determines under what situations a foreground pixel will be set to background, and hence it determines the application for the thinning operation.

We have described the effects of a single pass of a thinning operation over the image. In fact, the operator is normally applied repeatedly until it causes no further changes to the image (*i.e.* until *convergence*). Alternatively, in some applications, *e.g.* *pruning*, the operations may only be applied for a limited number of iterations. Thinning is the dual of thickening, *i.e.* thickening the foreground is equivalent to thinning the background.

Figure below shows the result of this thinning operation on a simple binary image.



The following figure displays the results of the thinning operation, reducing the original objects to a single pixel wide lines. It represents Original Image (top left), Binary Image (top right), Thinned Image (bottom left) and Inverse Thinned Image (bottom right) :



Each successive thinning iteration removed pixels marked by the results of the hit-or-miss operation as long as the removal of the pixels would not destroy the connectivity of the line.

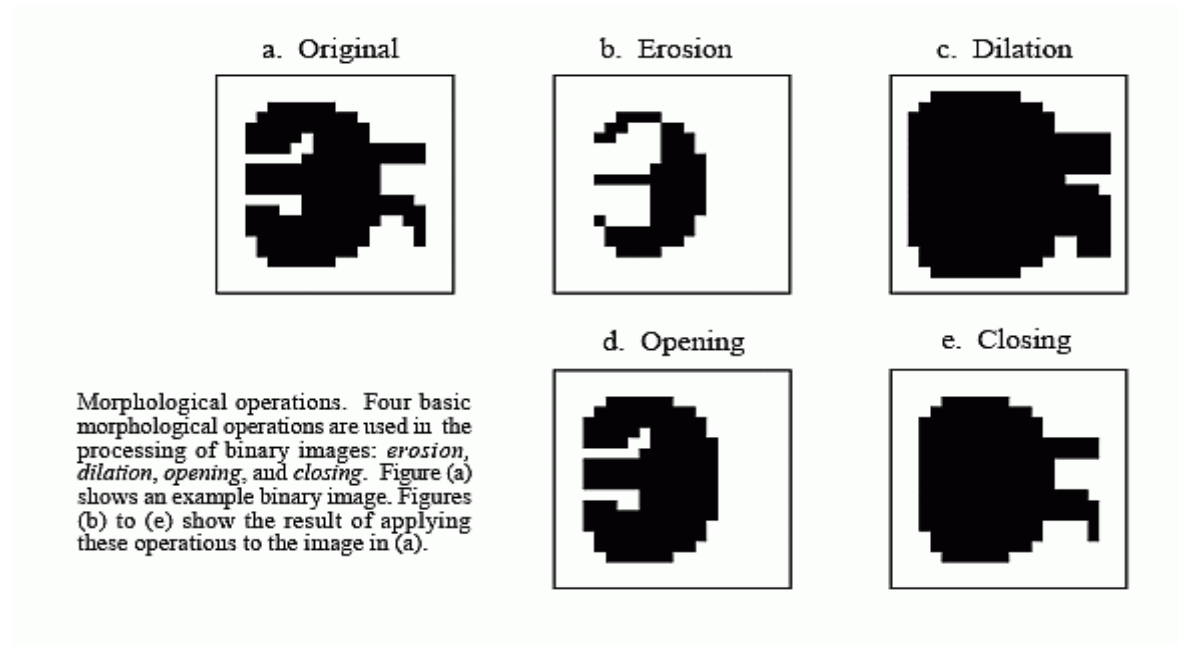
## **2.6 Morphological Image Processing**

Morphological image processing is a collection of techniques for digital image processing based on mathematical morphology. Since these techniques rely only on the relative ordering of pixel values, not on their numerical values, they are especially suited to the processing of binary images and grayscale images whose light transfer function is not known.

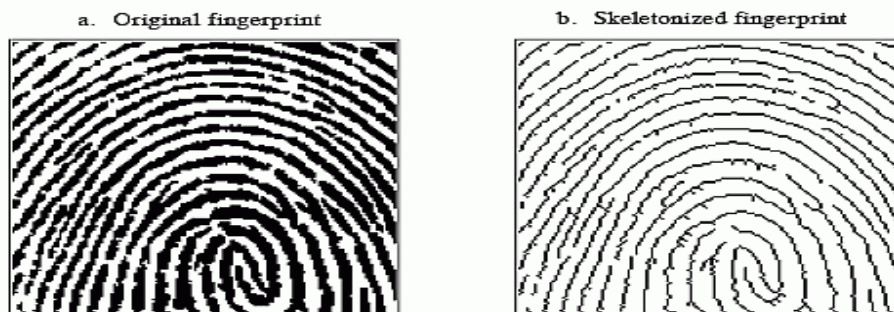
Four most common morphological operations used in image processing are : **erosion, dilation, opening & closing**. In erosion, every object pixel that is touching a background pixel is changed into a background pixel. In dilation, every background pixel that is touching an object pixel is changed into an object pixel. Erosion makes the objects smaller, and can break a single object into multiple objects. Dilation makes the objects larger, and can merge multiple objects into one.

**opening** is defined as an erosion followed by a dilation. The opposite operation of **closing**, defined as a dilation followed by an erosion. As illustrated by these examples, *opening* removes small islands and thin filaments of *object pixels*. Likewise, *closing* removes islands and thin

filaments of *background pixels*. These techniques are useful for handling noisy images where some pixels have the wrong binary value. For instance, it might be known that an object cannot contain a "hole", or that the object's border must be smooth.



The figure below shows an example of morphological processing. Figure (a) is the binary image of a fingerprint. Algorithms have been developed to analyze these patterns, allowing individual fingerprints to be matched with those in a database. A common step in these algorithms is shown in (b), an operation called **skeletonization**. This simplifies the image by *removing* redundant pixels; that is, changing appropriate pixels from black to white. This results in each ridge being turned into a line only a single pixel wide.



Binary skeletonization. The binary image of a fingerprint, (a), contains ridges that are many pixels wide. The skeletonized version, (b), contains ridges only a single pixel wide.

## 2.6.1 Hit-and-Miss Transform

The hit-and-miss transform is a general binary morphological operation that can be used to look for particular patterns of foreground and background pixels in an image. It is actually the basic operation of binary morphology since almost all the other binary morphological operators can be derived from it. As with other binary morphological operators it takes as input a binary image and a structuring element, and produces another binary image as output.

The structuring element used in the hit-and-miss is a slight extension to the type that has been introduced for erosion and dilation, in that it can contain both foreground and background pixels, rather than just foreground pixels, *i.e.* both ones and zeros. Note that the simpler type of structuring element used with erosion and dilation is often *depicted* containing both ones and zeros as well, but in that case the zeros really stand for `don't care's', and are just used to fill out the structuring element to a convenient shaped kernel, usually a square. In all our illustrations, these `don't care's' are shown as blanks in the kernel in order to avoid confusion. An example of the extended kind of structuring element is shown in Figure below. As usual we denote foreground pixels using ones, and background pixels using zeros.

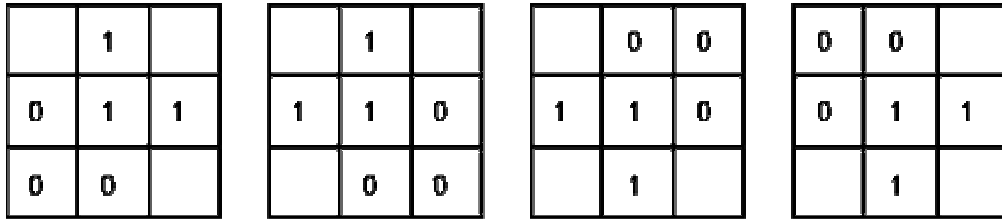
	1	
0	1	1
0	0	

**Figure :** Example of the extended type of structuring element used in hit-and-miss operations. This particular element can be used to find corner points, as explained below.

The hit-and-miss operation is performed in much the same way as other morphological operators, by translating the origin of the structuring element to all points in the image, and then comparing the structuring element with the underlying image pixels. If the foreground and background pixels in the structuring element *exactly match* foreground and background pixels in the image, then the pixel underneath the origin of the structuring element is set to the foreground color. If it doesn't match, then that pixel is set to the background color.

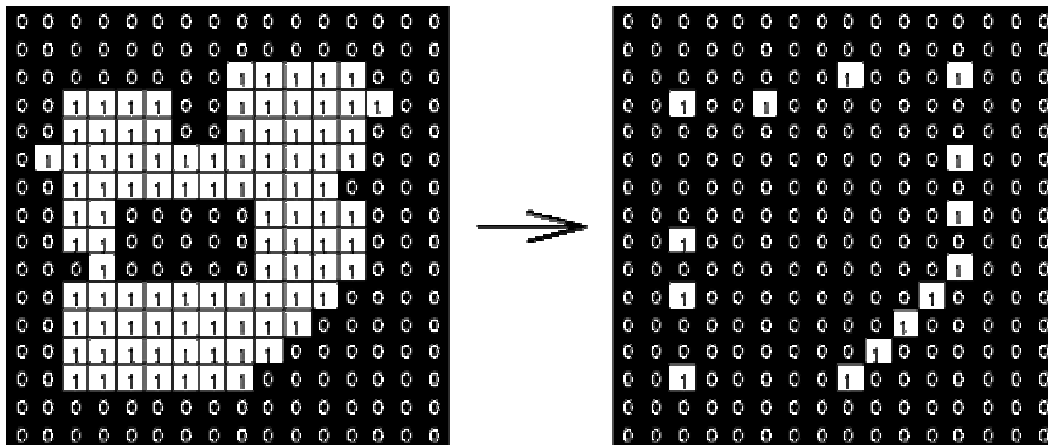
For instance, the structuring element shown in Figure below can be used to find right angle convex corner points in images. Notice that the pixels in the element form the shape of a bottom-left convex corner. We assume that the origin of the element is at the center of the 3×3 element. In order to find all the corners in a binary image we need to run the hit-and-miss transform four

times with four different elements representing the four kinds of right angle corners found in binary images. Figure below shows the four different elements used in this operation :



**Figure** represents Four structuring elements used for corner finding in binary images using the hit-and-miss transform. Note that they are really all the same element, but rotated by different amounts.

After obtaining the locations of corners in each orientation, We can then simply OR all these images together to get the final result showing the locations of all right angle convex corners in any orientation. Figure below shows the effect of this corner detection on a simple binary image.



Implementations vary as to how they handle the hit-and-miss transform at the edges of images where the structuring element overlaps the edge of the image. A simple solution is to simply assume that any structuring element that overlaps the image does not match underlying pixels, and hence the corresponding pixel in the output should be set to zero.

# Chapter 3

# Object Recognition Techniques

## 3.1 Introduction

Pattern recognition aims to classify data (patterns) based on either *a priori* knowledge or on statistical information extracted from the patterns. The patterns to be classified are usually groups of measurements or observations, defining points in an appropriate multidimensional space. An intriguing problem in pattern recognition yet to be solved is the relationship between the problem to be solved (data to be classified) and the performance of various pattern

recognition algorithms (classifiers). Holographic associative memory is another type of pattern matching scheme where a target small patterns can be searched from a large set of learned patterns based on cognitive meta-weight.

Typical applications are automatic speech recognition, classification of text into several categories (e.g. spam/non-spam email messages), the automatic recognition of handwritten postal codes on postal envelopes, or the automatic recognition of images of human faces. The last two examples form the subtopic image analysis of pattern recognition that deals with digital images as input to pattern recognition systems. Pattern recognition is more complex when templates are used to generate variants. For example, in English, sentences often follow the "N-VP" (noun - verb phrase) pattern, but some knowledge of the English language is required to detect the pattern. Pattern recognition is studied in many fields, including psychology, ethology, and computer science.

### **3.2 Pattern & Pattern Classes**

A pattern is an arrangement of descriptors that are useful when working with region boundaries. A pattern class is a family of patterns that share a set of common properties. Pattern classes are denoted by  $\omega_1, \omega_2, \dots, \omega_W$ , where  $W$  is the number of classes. Pattern recognition by machine involves techniques for assigning patterns to their respective classes – automatically and with as little human intervention as possible.

The nature of the components of a pattern vector depends on the approach used to describe the physical pattern itself. For example, consider the problem of automatically classifying alphanumeric characters. Descriptor suitable for a decision theoretic approach might include measures such 2-D moment invariants or a set of Fourier coefficients describing the outer boundary of the characters.

In some applications, pattern characteristics are best described by structural relationships. For example, fingerprint recognition is based on the interrelationships of print features called *minutiae*. Together with their relative sizes and locations, these features are primitive components that describe finger print ridge properties, such as abrupt endings, branching, merging and disconnected segments. Recognition problems of this type, in which not only quantitative measures about each feature but also the spatial relationships between the features determine class membership, generally are best solved by structural approaches. A complete pattern recognition system consists of a sensor that gathers the observations to be classified or described; a feature extraction mechanism that computes numeric or symbolic information

from the observations; and a classification or description scheme that does the actual job of classifying or describing observations, relying on the extracted features.

The classification or description scheme is usually based on the availability of a set of patterns that have already been classified or described. This set of patterns is termed the training set and the resulting learning strategy is characterized as supervised learning. Learning can also be unsupervised, in the sense that the system is not given an *a priori* labeling of patterns, instead it establishes the classes itself based on the statistical regularities of the patterns.

The classification or description scheme usually uses one of the following approaches: statistical (or decision theoretic), syntactic (or structural). Statistical pattern recognition is based on statistical characterizations of patterns, assuming that the patterns are generated by a probabilistic system. Structural pattern recognition is based on the structural interrelationships of features.

### **3.3 Recognition based on Decision-Theoretic methods**

Decision theoretic approaches to recognition are based on the use of decision or discriminant functions. Let  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$  represent an n dimensional pattern vector. For W pattern classes  $\omega_1, \omega_2, \dots, \omega_W$ , the basic problem in decision-theoretic pattern recognition is to find W decision functions  $d_1(x), d_2(x), \dots, d_W(x)$  with the property that, if a pattern x belongs to the class  $\omega_i$ , then

$$d_i(x) > d_j(x) \quad j=1,2,\dots,W; j \neq i$$

In other words, an unknown pattern x is said to belong to the ith pattern class if, upon substitution of x into all decision functions,  $d_i(x)$  yields the largest numerical value. Ties are resolved arbitrarily.

The decision boundary separating class  $\omega_i$  from  $\omega_j$  is given by values of x for which  $d_i(x) = d_j(x)$  or equivalently by values of x for which

$$d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0$$

Common practice is to identify the decision boundary between two classes by the single function  $d_{ij}(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0$ . Thus  $d_{ij}(\mathbf{x}) > 0$  for patterns of class  $\omega_i$  and  $d_{ij}(\mathbf{x}) < 0$  for patterns of class  $\omega_j$ .

#### **3.3.1 Forming Pattern Vectors**

Pattern vectors can be formed from quantitative descriptors. For example, suppose that we describe a boundary by using Fourier descriptors. The value of the ith descriptor becomes the value of  $x_i$ , the ith component of a pattern vector. In addition, we could append other components to pattern vector. Another approach used quite frequently when dealing with

(registered) multispectral images is to stack the images and then form vectors from corresponding pixels in the images. The images are stacked by using function `cat` :

$$S = \text{cat}(3, f_1, f_2, \dots, f_n)$$

Where  $S$  is the stack and  $f_1, f_2, \dots, f_n$  are the images from which the stack is formed. The vectors then are generated by using function `imstack2vectors`.

### **3.3.2 Matching**

Recognition techniques based on matching represent each class by a prototype pattern vector. An unknown pattern is assigned to the class to which it is closest in terms of a predefined metric. The simplest approach is the minimum distance classifier, which, as its name implies, computed the Euclidean distance between the unknown and each of the prototype vectors. It chooses the smallest distance to make a decision.

### **3.3.3 Pattern Matching Using Minimum Distance Classifiers**

Suppose that each pattern class,  $\omega_j$ , is characterized by a mean vector  $m_j$ . That is, we use the mean vector of each population of training vectors as being representative of that class of vectors:

$$m_j = (1/N_j) \sum_{x \in \omega_j} x \quad x \in \omega_j; j = 1, 2, \dots, W$$

where  $N_j$  is the number of training pattern vectors from class  $\omega_j$  and the summation is taken over these vectors. One way to determine the class membership of an unknown pattern vector  $x$  is to assign it to the class of its closest prototype. Using Euclidean distance as a measure of closeness reduces the problem to computing the distance measures :

$$D_j(x) = \|x - m_j\| \quad j = 1, 2, \dots, W$$

We then assign  $x$  to class  $\omega_i$  if  $D_i(x)$  is the smallest distance. That is, the smallest distance implies the best match in this formulation. Suppose that all the mean vectors are organized as rows of a matrix  $M$ . Then computing the distances from an arbitrary pattern  $x$  to all the mean vectors is accomplished.

It is not difficult to show that selecting the smallest distance is equivalent to evaluating the functions

$$D_j(x) = x^T m_j - (1/2) m_j^T m_j \quad j = 1, 2, \dots, W$$

And assigning  $x$  to class  $\omega_i$  if  $d_i(x)$  yields the largest numerical value. This formulation agrees with the concept of a decision function.

The decision boundary between classes  $\omega_i$  and  $\omega_j$  for a minimum distance classifier is

$$\begin{aligned} d_{ij}(x) &= d_i(x) - d_j(x) \\ &= x^T (m_i - m_j) - (1/2)(m_i - m_j)^T (m_i + m_j) = 0 \end{aligned}$$

The surface given by this equation is the perpendicular bisector of the line segment joining  $m_i$  and  $m_j$ . For  $n = 2$ , the perpendicular bisector is a line, for  $n = 3$  it is a plane, and for  $n > 3$  it is called a hyperplane.

### **3.3.4 Matching by Correlation**

Correlation is quite simple in principle. Given an image  $f(x, y)$ , the correlation problem is to find all places in the image that match a given subimage (also called a mask or template)  $w(x, y)$ . Typically,  $w(x, y)$  is much smaller than  $f(x, y)$ . One approach for finding matches is to treat  $w(x, y)$  as a spatial filter and compute the sum of products (or normalized version of it) for each location of  $w$  in  $f$ . Then best match or matches of  $w(x, y)$  in  $f(x, y)$  is/are the location(s) of the maximum value(s) in the resulting correlation image. Unless  $w(x, y)$  is small, the approach just described generally becomes computationally intensive. For this reason, practical implementations of spatial correlation typically rely on hardware oriented solutions.

For prototyping, an alternative approach is to implement correlation in the frequency domain, making use of correlation theorem, which, alike the convolution theorem which relates spatial correlation to the product of the image transforms. Letting “ $\circ$ ” denote correlation and “ $*$ ” the complex conjugate, the correlation theorem states that

$$f(x, y) \circ w(x, y) \Leftrightarrow F(u, v)H^*(u, v)$$

In other words, spatial correlation can be obtained as the inverse Fourier transform of the product of the transform of one function times the conjugate of the transform of the other. Conversely it follows that

$$f(x, y)w^*(x, y) \Leftrightarrow F(u, v)\circ H(u, v)$$

This second aspect of the correlation theorem is included for its completeness.

## **3.4 Structural Recognition**

Structural recognition techniques are based generally on representing objects of interest as strings, trees or graphs and then defining descriptors and recognition rules based on those representations. The key difference between decision theoretic and structural methods is that the former uses quantitative descriptors expressed in the form of numerical vectors. Structural techniques, on the other hand, deal principally with symbolic informations. For instance, suppose that object boundaries in a given application are represented by minimum perimeter polygons. A decision theoretic approach might be based on forming vectors whose elements are the numeric values of the interior angles of the polygons, while a structural approach might be based on defining symbols for ranges of angle values and then forming a string of such symbols to describe the patterns.

### **3.4.1 String Matching**

Suppose that two region boundaries, a and b, are coded into strings  $a_1a_2\dots a_m$  and  $b_1b_2\dots b_n$ , respectively. Let  $\alpha$  denote the number of matches between these two strings, where a match is said to occur in the  $k$ th position if  $a_k=b_k$ . The number of symbols that do not match is

$$\beta = \max(|a|, |b|) - \alpha$$

Where  $|a|$  is the length (number of symbols) of the string in the argument. It can be shown that  $\beta = 0$  if and only if a and b are identical strings.

A simple measure of similarity between a and b is the ratio

$$R = \alpha / \beta = \alpha / (\max(|a|, |b|) - \alpha)$$

This measure, proposed by Sze and Yang, 1981 is infinite for a perfect match and 0 when none of the corresponding symbols in a and b match ( $\alpha$  is zero in this case).

# Chapter 4

In this chapter we are going to explain some well known facts about the electrical and electronics components that we used for recognition:

## **4.1 Capacitors**

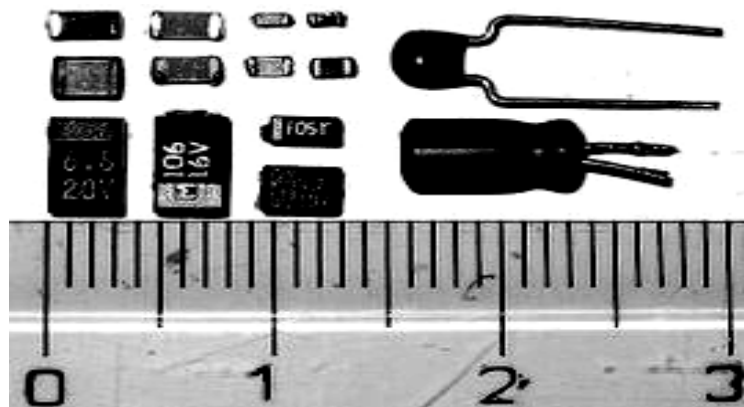
A **capacitor** is an electrical device that can store energy in the electric field between a pair of closely-spaced conductors (called 'plates'). When voltage is applied to the capacitor, electric charges of equal magnitude, but opposite polarity, build up on each plate. Capacitors are used in electrical circuits as energy-storage devices. They can also be used to differentiate between high-frequency and low-frequency signals and this makes them useful in electronic filters. Capacitors are occasionally referred to as **condensers**. This is now considered an antiquated term.

### **Capacitor types**

- **Metal film:** Made from high quality polymer film and metal foil (usually polycarbonate, polystyrene, polypropylene, polyester (Mylar), and for high quality capacitors polysulfone), with a layer of metal deposited on surface. They have good quality and stability, and are suitable for timer circuits. Suitable for high frequencies.
- **Mica:** Similar to metal film. Often high voltage. Suitable for high frequencies. Expensive.
- **Paper:** Used for high voltages.
- **Glass:** Used for high voltages. Expensive. Stable temperature coefficient in a wide range of temperatures.
- **Ceramic:** Chips of alternating layers of metal and ceramic. Depending on their dielectric, whether Class 1 or Class 2, their degree of temperature/capacity dependence varies. They

often have (especially the class 2) high dissipation factor, high frequency coefficient of dissipation, their capacity depends on applied voltage, and their capacity changes with aging. However they find massive use in common low-precision coupling and filtering applications. Suitable for high frequencies.

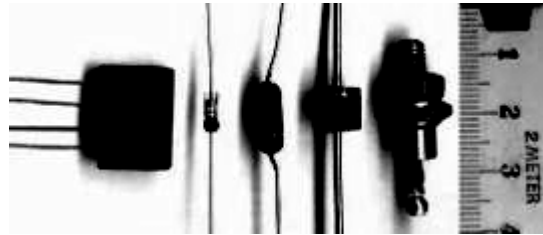
- **Electrolytic:** Polarized. Constructionally similar to metal film, but the electrodes are made of aluminium etched to acquire much higher surfaces, and the dielectric is soaked with liquid electrolyte. They suffer from poor tolerances, high instability, gradual loss of capacity especially when subjected to heat, and high leakage. Special types with low equivalent series resistance are available. Tend to lose capacity in low temperatures. Can achieve high capacities.
- **Tantalum:** Like electrolytic. Polarized. Better performance with higher frequencies. High dielectric absorption. High leakage. Has much better performance in low temperatures.
- **Supercapacitors:** Made from carbon aerogel, carbon nanotubes, or highly porous electrode materials. Extremely high capacity. Can be used in some applications instead of rechargeable batteries.











## 4.2 Diodes

In electronics, a **diode** is a component that restricts the direction of flow of charge carriers. Essentially, it allows an electric current to flow in one direction, but blocks it in the opposite direction. Thus, the diode can be thought of as an electronic version of a check valve. Circuits that require current flow in only one direction will typically include one or more diodes in the circuit design.

Early diodes included "cat's whisker" crystals and vacuum tube devices (called *thermionic valves* in British English Dialect). Today the most common diodes are made from semiconductor materials such as silicon or germanium.



## Types of semiconductor diode

			
Diode	Zener Diode	Schottky Diode	Tunnel Diode
			
Light-emitting diode	Photodiode	Varicap	SCR

Some diode symbols

## 4.3 Resistors

A **resistor** is a two-terminal electrical or electronic component that resists an electric current by producing a voltage drop between its terminals in accordance with Ohm's law:  $V=IR$ . The *electrical resistance* is equal to the voltage drop across the resistor divided by the current through the resistor. Resistors are used as part of electrical networks and electronic circuits.

### Identifying resistors

Most axial resistors use a pattern of colored stripes to indicate resistance. Surface-mount ones are marked numerically. Cases are usually brown, blue, or green, though other colors are occasionally found such as dark red or dark gray. One can use a multimeter or ohmmeter to test the values of a resistor.

## Four-band axial resistors

It has been suggested that this section be split into a new article entitled *Electronic color code*. Four-band identification is the most commonly used color coding scheme on all resistors. It consists of four colored bands that are painted around the body of the resistor. The scheme is simple: The first two numbers are the first two significant digits of the resistance value, the third is a multiplier, and the fourth is the tolerance of the value. Each color corresponds to a certain number, shown in the chart below. The tolerance for a 4-band resistor will be 2%, 5%, or 10%.

The Standard EIA Color Code Table per EIA-RS-279 is as follows:

**Color 1<sup>st</sup> band 2<sup>nd</sup> band 3<sup>rd</sup> band (multiplier) 4<sup>th</sup> band (tolerance) Temp. Coefficient**

Brown	1	1	$\times 10^1$	$\pm 1\%$ (F)	100 ppm
Red	2	2	$\times 10^2$	$\pm 2\%$ (G)	50 ppm
Orange	3	3	$\times 10^3$		15 ppm
Yellow	4	4	$\times 10^4$		25 ppm
Green	5	5	$\times 10^5$	$\pm 0.5\%$ (D)	
Blue	6	6	$\times 10^6$	$\pm 0.25\%$ (C)	
Violet	7	7	$\times 10^7$	$\pm 0.1\%$ (B)	
Grey	8	8	$\times 10^8$	$\pm 0.05\%$ (A)	
White	9	9	$\times 10^9$		
Gold			$\times 0.1$	$\pm 5\%$ (J)	
Silver			$\times 0.01$	$\pm 10\%$ (K)	
None				$\pm 20\%$ (M)	

*Note:* red to violet are the colors of the rainbow where red is low energy and violet is higher energy.

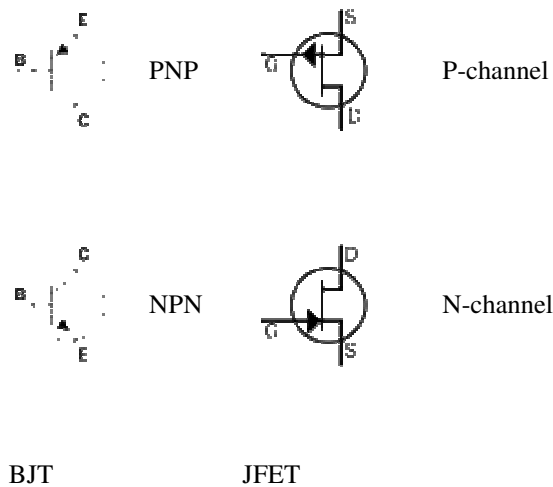
As an example, let us take a resistor which (read left to right) displays the colors *yellow, purple, yellow, brown*. We take the first two bands as the value, giving us 4, 7. Then the third band, another *yellow*, gives us the multiplier  $10^4$ . Our total value is then  $47 \times 10^4 \Omega$ , totalling  $470,000 \Omega$  or  $470 \text{ k}\Omega$ . Our brown is then a tolerance of  $\pm 1\%$ .

Resistors use specific values, which are determined by their tolerance. These values repeat for every exponent; 6.8, 68, 680, and so forth. This is useful because the digits, and hence the first



The transistor's low cost, flexibility and reliability have made it a universal device for non-mechanical tasks, such as digital computing. Transistorized circuits have replaced electromechanical devices for the control of appliances and machinery as well. It is often less expensive and more effective to use a standard microcontroller and write a computer program to carry out a control function than to design an equivalent mechanical control function.

## Types



BJT and JFET symbols

Transistors are categorized by:

- Semiconductor material: germanium, silicon, gallium arsenide, silicon carbide
- Structure: BJT, JFET, IGFET (MOSFET), IGBT, "other types"
- Polarity: NPN, PNP (BJTs); N-channel, P-channel (FETs)
- Maximum power rating: low, medium, high
- Maximum operating frequency: low, medium, high, radio frequency (RF), microwave (The maximum effective frequency of a transistor is denoted by the term  $f_T$ , an abbreviation for "frequency of transition". The frequency of transition is the frequency at which the transistor yields unity gain).
- Application: switch, general purpose, audio, high voltage, super-beta, matched pair
- Physical packaging: through hole metal, through hole plastic, surface mount, ball grid array, power modules

Thus, a particular transistor may be described as: *silicon, surface mount, BJT, NPN, low power, high frequency switch.*

## Other transistor types

- Heterojunction Bipolar Transistor
- Unijunction transistors can be used as simple pulse generators. They comprise a main body of either P-type or N-type semiconductor with ohmic contacts at each end (terminals *Base1* and *Base2*). A junction with the opposite semiconductor type is formed at a point along the length of the body for the third terminal (*Emitter*).
- **Dual gate FETs** have a single channel with two gates in cascode; a configuration that is optimized for **high frequency amplifiers, mixers**, and oscillators.
- **Transistor arrays** are used for general purpose applications, **function generation** and low-level, **low-noise amplifiers**. They include two or more transistors on a common **substrate** to ensure close parameter matching and thermal tracking, characteristics that are especially important for **long tailed pair** amplifiers.
- Darlington transistors comprise a medium power BJT connected to a power BJT. This provides a high current gain equal to the product of the current gains of the two transistors. Power diodes are often connected between certain terminals depending on specific use.
- Insulated gate bipolar transistors (IGBTs) use a medium power IGFET, similarly connected to a power BJT, to give a high input impedance. Power diodes are often connected between certain terminals depending on specific use. IGBTs are particularly suitable for heavy-duty industrial applications. The Asea Brown Boveri (ABB) **5SNA2400E170100** [4] illustrates just how far power semiconductor technology has advanced. Intended for three-phase power supplies, this device houses three NPN IGBTs in a case measuring 38 by 140 by 190 mm and weighing 1.5 kg. Each IGBT is rated at 1,700 volts and can handle 2,400 amperes.
- Single-electron transistors (SET) consist of a gate island between two tunnelling junctions. The tunnelling current is controlled by a voltage applied to the gate through a capacitor. [5][6]
- Nanofluidic Transistor Control the movement of ions through sub-microscopic, water-filled channels. Nanofluidic transistor, the basis of future chemical processors
- Trigate transistors (Prototype by Intel)
- Avalanche transistor
- Ballistic transistor Electrons bounce their way through maze.
- Spin transistor Magnetically-sensitive
- Thin film transistor Used in LCD display.
- Floating-gate transistor Used for non-volatile storage.

- Photo transistor React to light
- Inverted-T field effect transistor
- Ion sensitive field effect transistor To measure ion concentrations in solution.
- FinFET The source/drain region forms fins on the silicon surface.
- FREDFET Fast-Reverse Epitaxial Diode Field-Effect Transistor
- EOSFET Electrolyte-Oxide-Semiconductor Field Effect Transistor (Neurochip)

# Chapter 5

## **EXPERIMENTS & RESULTS**

**For the recognition of electrical and electronics components, we adopted basically three techniques as follows :**

- (i) Quantitative Approach**
- (ii) By finding Roundness of Objects**
- (iii) Matching by Correlation**

### **5.1 Quantitative Approach**

In this approach, the objects are being recognized based on its physical appearances viz length, breadth, number of legs of the electrical & electronics components. The algorithms followed in this approach is described below :

#### **5.1.1 Algorithm for recognition by Quantitative method :**

- (i) First of all, read input image which is an RGB image.

- (ii) Then, convert the RGB image into Binary image to make the processing operation easier.
- (iii) Find the centroid of the image so that we can find a point which should be inside the image.
- (iv) Find the length of the image. For finding the length, we keep y constant and move along x-direction. In 1<sup>st</sup> case we increase x position till we find boundary and count the number of pixels. Similarly in second case, this is done by decreasing x position.
- (v) Find the breadth of the object. As in step (iv) in the same manner we find the breadth of the image by increasing and decreasing y position keeping x constant.
- (vi) Find the number of legs of the image. For finding number of legs, we move towards boundary and upon finding it, our count is increased by a predetermined position by taking that point we move perpendicular to it. A leg can be indicated by changing of pixel from white to black and then the counter count the number of legs.
- (vii) Finally we decide what kind of object it may be depending on the length breadth ratio and the number of legs of the objects.

### 5.1.2 MATLAB CODE :

```

clc
clear all

%input image
i=imread('capa2.jpg');

%show input image
imview(i)

%converting input image to binary image
g=graythresh(i);
bw=im2bw(i,g);
% imview(bw);

%finding centroid of given image
[n m]=size(bw);
k=0;
j=0;
d=0;
for x=1:m
    for y=1:n
        if bw(y,x)==0
            k=k+x;
            j=j+y;
            d=d+1;
        end
    end
end
xcor=k/d;
x=round(xcor);
ycor=j/d;
y=round(ycor);
centroid=[x y]

%length of input image

```

```

Length=0; %initialising length
Legs1=0; %initialising legs of image
Legs2=0;
Legs3=0;
Legs4=0;

for k=x:m
    if bw(y,k)==0
        Length=Length+1;
    else
        k=k+100;
        for j=1:n-5
            if bw(j,k)==1 && bw(j+1,k)==0
                Legs1=Legs1+1;
            else
                continue;
            end
        end
        break;
    end
end

for k=x:-1:1
    if bw(y,k)==0
        Length=Length+1;
    else
        k=k-20;
        for j=1:n-5
            if bw(j,k)==1 && bw(j+1,k)==0
                Legs2=Legs2+1;
            else
                continue;
            end
        end
        break;
    end
end

if Length>0

    %Width of image
    Width=0; %initialising width
    for k=y:n
        if bw(k,x)==0
            Width=Width+1;
        else
            k=k+20;
            for j=1:m-5
                if bw(k,j)==1 && bw(k,j+1)==0
                    Legs3=Legs3+1;
                else
                    continue;
                end
            end
        end
    end
end

```

```

        end
        break;
    end
end

for k=y:-1:1
    if bw(k,x)==0
        Width=Width+1;
    else
        k=k-20;
        for j=1:m-5
            if bw(k,j)==1 && bw(k,j+1)==0
                Legs4=Legs4+1;
            else
                continue;
            end
        end
        break;
    end
end
end

```

```

%finding ratio of length & Width
Ratio=Length/Width
R=Ratio;

```

```

%output depend on condition
if R<.75 && Legs3==2
    imview('capacitor1.jpg')
elseif R<10 && Legs3==3
    Legs=Legs3;
    Legs
    imview('triode1.jpg')
elseif R>.75 && R<1.3 && Legs3==2
    imview('roundcap1.jpg')
elseif R<3
    imview('resistor1.jpg')
elseif R>3
    imview('diode1.jpg')
else
    imview('Blue hills.jpg')
end

```

```

else
    Length=0;
    Legs1=0;
    Legs2=0;
    Legs3=0;
    Legs4=0;
    for k=x:m
        if bw(y,k)==1
            Length=Length+1;
        else

```

```

k=k+20;
for j=1:n-5
    if bw(j,k)==1 && bw(j+1,k)==0
        Legs1=Legs1+1;
    else
        continue;
    end
end
break;
end
end

```

```

for k=x:-1:1
    if bw(y,k)==1
        Length=Length+1;
    else
        k=k-20;
        for j=1:n-5
            if bw(j,k)==1 && bw(j+1,k)==0
                Legs2=Legs2+1;
            else
                continue;
            end
        end
        break;
    end
end
end

```

% if Length>0

% Width of image

```

Width=0;
for k=y:n
    if bw(k,x)==1
        Width=Width+1;
    else
        k=k+100;
        for j=1:m-5
            if bw(k,j)==1 && bw(k,j+1)==0
                Legs3=Legs3+1;
            else
                continue;
            end
        end
        break;
    end
end
end

```

```

for k=y:-1:1
    if bw(k,x)==1
        Width=Width+1;
    else

```

```

k=k-20;
for j=1:m-5
    if bw(k,j)==1 && bw(k,j+1)==0
        Legs4=Legs4+1;
    else
        continue;
    end
end
break;
end
end

%finding ratio of length & Width

Ratio=Length/Width
R=Ratio;

%output depend on condition

if R<.75 && Legs3==2
    Legs=Legs3;
    Legs
    imview('capacitor1.jpg')
elseif R<10 && Legs3==3
    Legs=Legs3;
    Legs
    imview('triode1.jpg')
elseif R>.75 && R<1.3 && Legs3==2
    Legs=Legs3;
    Legs
    imview('roundcap1.jpg')
elseif R<3
    imview('resistor1.jpg')
elseif R>3
    imview('diode1.jpg')
else
    imview('Blue hills.jpg')
end
end

```

**5.2 By finding Roundness of Objects** : In this method, the objects are being recognized based on its roundness by using a specific MATLAB tool for it. The algorithms adopted in this method is described below.

### **5.2.1 Algorithms for Roundness method** :

- (i) First of all we need to create a database of standard objects or images with which we are going to compare the input images.

- (ii) Then, read the input object which is to be compared with the standard one in the database.
- (iii) We use '*bwboundaries*' function as present in MATLAB Image Processing toolbox to find the boundary of the objects.
- (iv) Find area & perimeter of the object.
- (v) We compute the roundness using the following formula :  

$$\text{metric} = 4 * \pi * \text{area} / \text{perimeter}^2$$
- (vi) Compare the roundness of the selected object with the standard image in the database
- (vii) Then, finally we make decision about the object.

### 5.2.2 MATLAB CODE :

```

clc;
clear all;

P=imread('roundcap.jpg');           %input image
imview(P)                           %show input image
for i=1:6
    if i==2
        RGB=imread('capacitor.jpg');
    elseif i==3
        RGB=imread('triode.jpg');     %database standard image
    elseif i==4
        RGB=imread('resistor.jpg');
    elseif i==5
        RGB=imread('diode.jpg');
    elseif i==6
        RGB=imread('roundcap.jpg');
    else
        RGB = P;
    end
    I = rgb2gray(RGB);                %convert rgb to grayscale image
    threshold = graythresh(I);        %finding threshold
    bw = im2bw(I,threshold);          %convert into binary image

    %concentrate only on the exterior boundary.option noholes will accelerate
    %the processing by preventing bwboundaries from searching ofinner contour

    [B,L] = bwboundaries(bw,'noholes');

    % hold on
    % loop over the boundaries
    for k = 1:length(B)
        boundary = B{k};
    end

    stats = regionprops(L,'Area','Centroid');

    % obtain (X,Y) boundary coordinates corresponding to label 'k', here k=1

```

```

boundary = B{1};

% compute a simple estimate of the object's perimeter
delta_sq = diff(boundary).^2;
perimeter = sum(sqrt(sum(delta_sq,2)));

% obtain the area calculation corresponding to label 'k'
area = stats(1).Area;

% compute the roundness metric
metric(i) = 4*pi*area/perimeter^2;

end

%find error vector
for i=2:6
    if(metric(1)-metric(i)>0)
        e(i)=metric(1)-metric(i);
    else
        e(i)=metric(i)-metric(1);
    end
end
end

err=min(abs(e));                %find min error

if e(2)==err
    imview('capacitor1.jpg')
elseif e(3)==err
    imview('triode1.jpg')
elseif e(4)==err
    imview('resistor1.jpg')
elseif e(5)==err
    imview('diode1.jpg')
elseif e(6)==err
    imview('roundcap1.jpg')
end

```

**5.3 Matching by Correlation** : In this method, we tried to correlate the input image with the standard image in the database. The algorithms adopted in this method is described below.

### **5.3.1 Algorithms for Matching by Correlation method** :

- (i) First of all create database of standard images.
- (ii) Read input image.
- (iii) Correlate each input image with database images.
- (iv) Choose a threshold value.
- (v) Determine no of pixels greater & less than threshold value.
- (vi) Decision based on maximum correlation.

### 5.3.2 MATLAB CODE :

```
clc;
clear all;
ip=imread('roundcap.jpg');           %input image
imview(ip)

%database images
cap=imread('capacitor.jpg');
rcap=imread('roundcap.jpg');
tri=imread('triode.jpg');

%converting all images into binary images
ip1=graythresh(ip);
cap1=graythresh(cap);
rcap1=graythresh(rcap);
tri1=graythresh(tri);
ip2=im2bw(ip,ip1);
cap2=im2bw(cap,cap1);
rcap2=im2bw(rcap,rcap1);
tri2=im2bw(tri,tri1);

%converting binary image matrix into numerical data matrix
ip3=im2double(ip2);
cap3=im2double(cap2);
rcap3=im2double(rcap2);
tri3=im2double(tri2);

%change resolution of image window
ip4=imresize(ip3,[64 64],'bilinear');
cap4=imresize(cap3,[64 64],'bilinear');
rcap4=imresize(rcap3,[64 64],'bilinear');
tri4=imresize(tri3,[64 64],'bilinear');

imview(ip4)           %input image after resizing
q=xcorr2(ip4);       %finding auto correlation of i/p image
q1=max(max(q));     %find max value of pixel in autocorrelation matrix

%find a threshold value
for x=1:64
    for y=1:64
        if q(y,x)==q1
            q2=(q(y,x)+q(y+1,x)+q(y-1,x)+q(y,x-1)+q(y,x+1)+q(y+1,x+1)+q(y+1,x-1)+q(y-1,x+1)+q(y-1,x-1))/9
        else
            continue;
        end
    end
end

%find no. of pixels grater than the threshold value after correlation
k1=xcorr2(ip4,tri4);
count1=0;
for t=1:64
```

```

for p=1:64
    if k1(p,t)>q2
        count1= count1+1;
    else
        continue;
    end
end
end
count1

k2=xcorr2(ip4,cap4);
count2=0;
for t=1:64
    for p=1:64
        if k2(p,t)>q2
            count2= count2+1;
        else
            continue;
        end
    end
end
count2

k3=xcorr2(ip4,rcap4);
count3=0;
for t=1:64
    for p=1:64
        if k3(p,t)>q2
            count3= count3+1;
        else
            continue;
        end
    end
end
count3

if count1>count2 && count1>count3
    imview('triode1.jpg')
elseif count3>count1 && count3>count2
    imview('roundcap1.jpg')
else
    imview('capacitor1.jpg')
end

```

# Chapter 6

## **CONCLUSION**

## **CONCLUSION:**

We have presented a new approach to shape matching. Our project in object recognition began a transition from processes whose outputs are images to processes whose outputs are attributes about those images. Although our step is introductory in nature and the thesis is fundamental to understanding the state of the art in object recognition. A key characteristic of our approach is the estimation of shape similarity and correspondences based on quantitative approach by analyzing the physical appearance of the objects (viz length, breadth, length-breadth ratio, no of legs etc) and matching using correlation method. Our approach is simple and easy to apply. It can be implemented for parts identification on assembly lines, defects and fault inspection of the electronic components and devices in industrial automation. In our experiments, we have converted an RGB image to Binary image so that the processing can be made easy as its matrix deals with binary values. If we combine region-based and contour-based segmentation techniques for object recognition at a level, where both pathways interpret the image data on the level of complete regions and contour groups. With this combination and a top-down strategy including a region memory handling different time scales the object recognition process is significantly improved in situations where projections of 3D surfaces are merged due to homogeneous on extensions of pixel values. In fact, recognition of individual objects is a logical place to conclude. Specifically our next step would be the development of image analysis methods whose proper development requires concepts from machine intelligence. As we know that machine intelligence and some areas in digital image processing depend on image analysis and computer vision. Solutions of image analysis problems today are characterized by heuristic approaches. While these approaches are indeed varied, most of them share a significant base of techniques.

## **REFERENCES :**

- ❖ Digital Image Processing by Rafael C. Gonzalez, Richard E. Woods

- ❖ Digital Image Processing Using MATLAB by Rafael C. Gonzalez, Richard E. Woods & Steven L. Eddins
- ❖ Digital Image Processing & Analysis by B. Chanda & D. Dutta Majumdar
- ❖ UMBAUGH, S. E. Computer Vision and Image Processing
- ❖ <http://www.wikipedia.org>
- ❖ <http://www.mathworks.com>
- ❖ [http://www.mathtools.net/MATLAB/Image\\_Processing/](http://www.mathtools.net/MATLAB/Image_Processing/)
- ❖ <http://www.dspexperts.com/>
- ❖ <http://www.imageprocessingplace.com>
- ❖ <http://www.ph.tn.tudelft.nl>