

QUALITY CONTROL OF CERAMIC TILES USING INDUCTIVE LEARNING

by

Ali TÜRKYILMAZ

B. Sc. in I.E., Yıldız Technical University, 1996

**Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
The requirements for the degree of
Master of Science
in
Industrial Engineering**

12723

Fatih University

August 2000



Fatih Üniversitesi

Tarih: 07 / 08 / 2000

Fen Bilimleri Enstitüsü Müdürlüğü'ne

TUTANAK

ALİ TÜRK YILMAZ'a ait

"Quality Control of Ceramic Tiles Using Inductive Learning

adlı çalışma ...67... dk.'lık süre içinde savunulmuş ve jüri tarafından

Endüstri Mühendisliği Anabilim Dalında YÜKSEK LİSANS TEZİ olarak oy

birliğiyle / oy çokluğuyla kabul edilmiştir / ~~edilmemiştir~~.

..... yarıyıl ek süre verilmiştir.

Başkan ..Prof. Dr. Mazhar ÜNSAL

Üye ..Doc. Dr. M. Subih AKSOY

Üye ..Yrd. Doc. Dr. Nurullah ARSLAN

İ.C. ...
DOKÜMANLAMA MERKEZİ

ACKNOWLEDGEMENTS

First of all, I would like to express my deep gratitude to Assoc. Prof. Dr. M. Sabih AKSOY for his supervision, guidance, support and encouragement throughout this study.

I'm grateful to all the members of the Industrial Engineering Department especially Mehmet Sevkli and Assist. Prof. Dr. Nurullah ARSLAN for their help and friendship which will be remembered all the time.

I am deeply indebted to my mother and my father who did their best to foster and educate me.

Last, but not least, I would like to thank to my wife for her patience and support.



ABSTRACT

The ceramic tiles manufacturing process has now been completely automated with the exception of the final stage of production concerned with visual inspection. In this thesis a visual inspection system is described for the detection of ceramic tiles. The system employs RULE-3 inductive learning algorithm to classify if the tiles are defected or not. The results suggest that the performance is adequate to provide a basis for a viable commercial visual inspection system.



ÖZETÇE

Seramik endüstrisi son yıllarda bütün üretim safhalarında otamasyon kullanımına geçmektedir. Fakat bu geçiş ürünün kalite kontrolü safhasında henüz tam anlamıyla gerçekleştirilememiştir. Bu tezde, seramik kalite kontrolünde endüstriyel görüntü işleme metodu uygulaması ve RULES-3 endüktif öğrenme tekniği kullanarak hatalı ve hatasız ürünlerin tesbiti için alternatif bir teknik geliştirilmiştir. Sağlam ve hatalı seramikler sisteme öğretildikten sonra yeni ürünler sisteme sorulmuş ve sistemin etkinliği seramiğin türüne ve üzerindeki resmin detaylarına bağlı olarak %88-%96 arasında tesbit edilmiştir.



TABLE OF CONTENTS

ACKNOWLEDGEMENTS	III
ABSTRACT.....	IV
ÖZETÇE.....	V
TABLE OF CONTENTS	VI
LIST OF TABLES	IX
LIST OF FIGURES.....	X
1 INTRODUCTION	1
1.1 Problem Statement	1
1.2 Outline of the Thesis	2
2 MACHINE VISION.....	3
2.1 Relationship to Other Fields	3
2.2 Role of Knowledge	4
2.3 Vision System Applications.....	4
2.3.1 Feasibility Criteria	4
2.3.1.1 Inspection.....	6
2.3.1.2 Industries.....	6
2.4 Why Use Vision?	7
2.4.1 Safety and Reliability.....	7
2.4.2 No alternative:.....	7
2.4.3 Quality	8
2.4.4 Automation Aid	8

3	IMAGE PROCESSING	9
3.1	What is Image Processing?	9
3.2	Overview of Digital Image Processing	9
3.3	Digital Image Fundamentals	10
3.3.1	Image Acquisition.....	10
3.3.2	Image Storage	10
3.3.3	Image Processing	10
3.3.4	Image Display	10
3.3.5	Software	10
3.3.6	Image Model	10
3.3.7	Color	11
3.4	Image Processing Steps	11
3.4.1	Gray Scales	11
3.4.2	Color Information	11
3.4.3	Smoothing Noise.....	12
3.5	Binary Image Processing	12
3.5.1	Thresholding	13
3.6	Edge Detection	14
3.6.1	Edge Detection Operators.....	14
3.6.1.1	Roberts Operator.....	14
3.6.1.2	Sobel Operator	14
3.6.1.3	Prewitt operator.....	15
4	PATTERN CLASSIFICATION	16
4.1	Introduction	16
4.2	Artificial Neural Networks	16
4.2.1	Multilayer Perceptrons.....	16
4.2.2	Adaline Algorithm	16

4.3	Fuzzy Logic.....	16
4.4	Inductive Learning	17
4.4.1	RULES-3 Inductive Learning Algorithm	18
4.4.2	A simple Problem Illustrating the Operation of RULES-3.....	19
5	IMPLEMENTATION	23
5.1	How the System Works	25
5.2	Image Processing Steps	25
5.2.1	Setting Up The Inspection System	25
5.2.2	Gray Level	27
5.2.3	Increase the Contrast of Gray Level	28
5.2.4	Noise Smoothing.....	28
5.2.5	Increase the Contrast Level of Filtered Image.....	29
5.2.6	Convert to Binary Bmage	29
5.2.6.1	Thresholding Value.....	29
5.2.7	Edge Detection.....	30
6	CONCLUSION	42
7	APPENDIX	44
7.1	Files In the Disk.....	44
7.1.1	Main.m (Main Program).....	45
7.1.2	Fmask.m (Finding Mask Frequency).....	46
7.1.3	Cercsmall.m (Cutting Edges of Image)	48
7.2	Hardware Requirements.....	50
7.3	Software Requirements	50
	REFERENCES.....	51
	REFERENCES NOT CITED.....	54

LIST OF TABLES

TABLE 3-1 OVERVIEW OF IMAGE PROCESSING	9
TABLE 3-2 THE ADJACENT IMAGE SPECIFIES 256 DIFFERENT SHADES OF GRAY IN A 16 X 16 GRID..	11
TABLE 4-1 TRAINING SET FOR CAR ACCELERATION PROBLEM	20
TABLE 4-2 SUMMARY OF RULE INDUCTION SEQUENCE FOR CAR ACCELERATION PROBLEM.....	22
TABLE 5-1 PIXEL POSITION.....	30
TABLE 5-2 ALL POSSIBLE 3*3 MASKS	32
TABLE 5-3 MASKS USED FOR THIS TILE BECAUSE OF THEIR HIGH FREQUENCIES	33
TABLE 5-4 FREQUENCY OF MASKS FOR 'GOOD' TILE.....	33
TABLE 5-5 SET OF EXAMPLES FOR TRAINING.....	34
TABLE 5-6 EXTRACTED RULES (PART A)	35
TABLE 5-7 EXTRACTED RULES (PART B)	36
TABLE 5-8 AN UNSEEN EXAMPLE.....	38
TABLE 5-9 EXTRACTED RULES WHEN MINIMUM NUMBER OF CONDITIONS IS FIVE (5).....	37
TABLE 5-10 RECOGNIZED UNSEEN EXAMPLES.....	40

LIST OF FIGURES

FIGURE 1-1 A) GOOD TILE B) CRACK D) SPOT	1
FIGURE 4-1 AN EXAMPLE DECISION TREE	17
FIGURE 5-1 A) GOOD PRODUCT B) CRACK C) SPOT.....	23
FIGURE 5-2 SOME TILES USED FOR IMAGE PROCESSING AND LEARNING.....	24
FIGURE 5-3 HOW THE SYSTEM WORKS	25
FIGURE 5-4 VISUAL INSPECTION SYSTEM.....	26
FIGURE 5-5 ORIGINAL PICTURE OF A TILE.....	26
FIGURE 5-6 GRAY LEVEL IMAGE.....	27
FIGURE 5-7 AFTER INCREASING CONTRAST OF GRAY LEVEL IMAGE	28
FIGURE 5-8 MEDIAN FILTERED IMAGE.....	29
FIGURE 5-9 CONTRAST OF FILTERED IMAGE	29
FIGURE 5-10 BINARY IMAGE	30
FIGURE 5-11 EDGES OF THE IMAGES.....	31
FIGURE 5-12 A) ORIGINAL PICTURES B) GRAY LEVEL C) BINARY IMAGE D) EDGES OF IMAGE	41

1 INTRODUCTION

The increasing importance of quality control in recent years has led to increasing diversity of methods and widespread automation in quality control applications. Machine vision methods are used to give intelligence to robots that carry out the quality control operation.

1.1 Problem Statement

The ceramic tiles industrial sector has taken significant advantage of the advances in the world of automation in recent years. All production phases have been addressed through various technical innovations, with the exception of the final stage of the manufacturing process, namely the product inspection. This is still performed manually and is concerned with the sorting of tiles into distinct categories or the rejection of the tiles found with defects and pattern faults.

Tile production is a kind of mass production in which huge numbers of tiles are produced. Like the others, quality control is one of the most important operations for tile industry. In most tile industries inspection process of ceramic tiles are currently carried out by human, which decrease the speed of the process as well as the reliability. It's also too difficult for a worker to glaze at the products and pick up the defected ones among all for a long time. Nobody can continuously pay attention to products more than one hour. It effects the performance of quality control, and cause that defective goods reach the customer.

Control operation carried out by workers takes much time.

Because of production conditions cracks and spots could exist on the tiles. Our main problem is to inspect the tiles and find the defective ones.

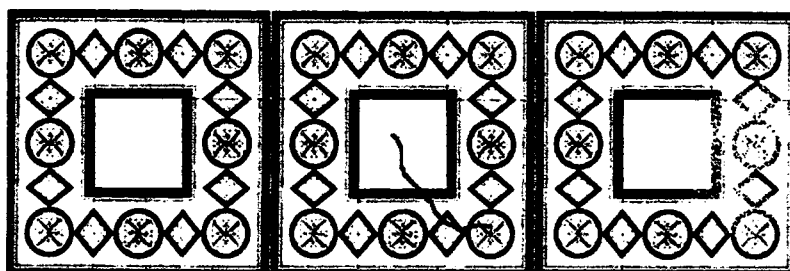


Figure 1-1 a) good tile b) crack d) spot

Instead of human vision, computer vision system is used to classify the tile. First of all a picture of the tile is taken and image-processing steps are applied to image. After getting the data, which consist of the frequencies of masks RULES-3 [1] algorithm is employed to determine the class of the tile. MATLAB and RULES-3 programs are used as the software for image processing and pattern recognition.

1.2 Outline of the Thesis

In this thesis, a computer visual inspection system for quality control of ceramic tiles is proposed. RULES-3 inductive learning algorithm is employed to extract the necessary set of rules to classify the tiles.

In chapter two, machine vision is explained. Some relations with the other fields and role of knowledge is briefly mentioned. Vision system applications and their advantages are explained.

In chapter three, image processing and the required steps, gray level, binary image, thresholding, edge detection, are examined. Digital image processing fundamentals are explained.

In chapter four, pattern classification methods are examined. Comparison, clustering, Bayesian classifiers, artificial neural network, Adaline algorithm, Fuzzy logic, Nearest neighbour and inductive learning method are explained. RULES-3 inductive learning algorithm is detailed because of its use in this study.

In chapter five, image processing steps are applied to tile and results are given. The necessary set of rules are extracted using RULES-3 inductive learning system. The recognition of new tiles is shown and performance of the system is given.

In chapter six conclusion takes place.

2 MACHINE VISION

The goal of a machine vision system is to create a model of the real world from images. A machine vision system recovers useful information about a scene from its two-dimensional projections. Since images are two-dimensional world, the information is not directly available and must be recovered. This recovery requires the inversion of a many-to-one mapping. To recover the information knowledge about the objects in the scene and projection geometry is required. [2]

2.1 Relationship to Other Fields

Many fields are related to machine vision. As we will see, techniques developed from many areas are used for recovering information from images.

Image processing is a well-developed field. Image processing techniques usually transform images into other images; the task of information recovery is left to a human user. This field includes topics such as image enhancement, image compression, and correcting blurred or out-of-focus images. On the other hand, machine vision algorithms take images as inputs but produce other type of outputs, such as representations for the object contours in an image. Thus, emphasis in machine vision is on recovering information automatically, with minimal interaction with a human. Image processing algorithms are useful in early stages of a machine vision system. They are usually used to enhance particular information and suppress noise.[3] [4]

Pattern recognition classifies numerical and symbolic data. Many statistical and syntactical techniques have been developed for classification of patterns. Techniques from pattern recognition play an important role in machine vision for recognizing objects. Object recognition in machine vision usually requires many other techniques.[5]

Artificial intelligence is concerned with designing systems that are intelligent and with studying computational aspects of intelligence. Artificial intelligence is used to analyze scenes by computing a symbolic representation of the scene contents after the images have been processed to obtain features. Artificial intelligence may be viewed as having three stages: perception, cognition, and action. Perception translates signals from the world into signals that effect changes in the world. Many techniques from artificial

intelligence play important roles in all aspects of computer vision. In fact, computer vision is often considered a subfield of artificial intelligence.[6]

2.2 Role of Knowledge

Decision-making always requires knowledge of the application or goal. At every stage in machine vision decisions must be made by the system. Emphasis in machine vision systems is on maximizing automatic operations at each stage, and these systems should use knowledge accomplish this. The knowledge used by the system includes models of features, image formation, models of object, and relationship among objects. Without explicit use of knowledge, machine vision systems can be designed to work only in a very constrained environment for very limited applications. To provide more flexibility and robustness, knowledge is represented explicitly and used by the system. The efficiency of a system is usually governed by the quality of the knowledge used by the system. Difficult problems are often solvable only by identifying the proper source of knowledge and appropriate mechanisms to use it in the system.[7]

2.3 Vision System Applications

Machine vision systems can be considered for use in most manufacturing applications in which human vision is now required. Human vision required for applications where non-contact feedback is used to provide information about a production process or a part. For example, a human welder or machinist uses visual feedback to insure that the correct spatial relationship is maintained between the tools and the work piece. Human assemblers visually analyze the position of parts so that other parts can be correctly aligned for insertion or some other form of mating. Quality control inspectors visually check products or parts to insure that there are no defects, such as missing parts, damage, or incorrect location of various features. [4]

2.3.1 Feasibility Criteria

To be feasible for use in a particular manufacturing application, a machine vision system must be able to correctly interpret an image with a satisfactory degree of reliability. Assuming this fundamental criteria can be satisfied, and then the system must be compared

with the use of human vision to determine if machine vision can perform a task with higher accuracy or lower cost than its human counterpart.

The evaluation of a machine vision system's applicability to a manufacturing operation requires that three basic criteria be considered. First, the fundamental system capabilities must be evaluated. Capabilities are the specific functions, which the system can perform. These include recognition of an object and measurement of various object features, such as position or shape.

There are several fundamental performance criteria, which can be used to evaluate a vision system, including operating speed, accuracy, resolution, and discrimination.

In addition, a measure of operating cost or payback should be evaluated when considering a specific application.

Another consideration is the requirements of the application. Each application requires that a vision system have certain capabilities and be able to achieve certain levels of performance. For example, an inspection operation in which objects with subtle differences in surface markings are to be differentiated may require a vision system with gray scale surface interpretation capabilities. Or a certain quality control application may require that a vision system be able to correctly read product labels with 99% accuracy.

An automated vision system will be suitable for a specific application only if it will perform a task better than a human operator. In general, a machine vision system will be more suitable for an application than a human in one of three situations:

When machine vision has capabilities that humans do not.

When machine vision offers better performance than humans, such as higher inspection speeds or lower error rate.

When a visual, inspection or feedback task is unpleasant or hazardous for humans to perform.

2.3.1.1 Inspection

The ability of an automated vision system to recognize well-defined patterns and determine if these patterns match those stored in the system's CPU memory makes ideal for the inspection of parts, assemblies, containers, and labels. Two types of inspection can be performed by vision systems: quantitative and qualitative. Quantitative inspection is the verification that measurable quantities fall within desired ranges of tolerance, such as dimensional measurements or numbers of holes. Qualitative inspection is the verification that certain components or properties are present and in a certain position, such as defects, missing parts, extraneous components, or misaligned parts. The output from a simple machine vision inspection task is a pass/reject evaluation for the object being inspected.[8]

Visual inspection represents one of the last manufacturing areas in which automation techniques have been employed. Most visual inspection is performed manually, either by simple observation or using a measurement tool, such as an optical comparator. One major advantage of using machine vision systems in place of human inspectors is that they can be used to perform 100% on-line inspection of parts at high speeds, possibly in working environments that are unpleasant for humans. Human workers are subject to fatigue, boredom, and the need for breaks, while machine vision systems can operate with consistent results over long periods of time. As a result, while human inspectors can only expect to achieve an 85-90% rate of accuracy in many situations machine vision systems, can achieve close to a 100% accuracy rate. [9]

2.3.1.2 Industries

Machine vision systems can potentially be used in nearly all industries. even at this early stage in the development of the field, vision systems are used by a variety of different industries. Although early users tend to be large sophisticated companies, the technology has been proven to be effective enough so that several medium-sized companies also use vision systems. Aerospace, automotive, and other heavy equipment manufacturers tend to be users of sophisticated systems used in conjunction with industrial robots or controlling processes or sorting manufactured parts. These companies also use machine vision systems for performing high speed inspection of moving parts, such as parts on a conveyor, the most sophisticated systems, such as the three-dimensional.

inspection system of Robotic Vision Systems, are being used by such large companies as Cummins Engine or Lockheed. [10]

In general, it can be assumed that machine vision systems are likely to be used by any industry which is involved in the application of flexible manufacturing systems or components. This includes nearly all manufacturing industries which deal with discrete parts. [11]

2.4 Why Use Vision?

Some idea of the present opportunities for using machine vision can be obtained from the types of industries and applications where it is already in operation. Because it's still a very new technology and users are still learning how to get the best out of it, it's not surprising that most applications so far have been in situations where there is no reasonable alternative way of reaching the desired goal.

2.4.1 Safety and Reliability

Safety and reliability requirements are probably the most important pressures on many manufacturers driving them to find better methods of inspecting their products. In pharmaceuticals, automobile engineering, aerospace, defence and some other industries where human lives can be at stake there are very stringent requirements for the reliability of products, and the trend is always towards higher and more difficult standards

In such circumstances one might think that nothing less than 100% inspection by skilled personnel could be tolerated. Experience has shown, however, that even repeated inspection, three or four times, on large batches of work can fail to remove every defective product. In fact the better the general level of quality, the more difficult it's to spot the occasional defect.

2.4.2 No alternative:

There are some situations where vision has provided a solution to a problem has been the means of making a new production process possible. In such a case the cost of the vision system is not the main consideration.

2.4.3 Quality

Below the level of safety criticality the cost of maintaining quality is less easily balanced against the cost of failures in quality. One of the things that may be possible with machine vision is to change from sample inspection to 100% inspection. When samples of outgoing goods are inspected and the customer uses sample inspection of goods inward, a single defect found by the customer may lead to the return of an entire batch, which must then be sorted completely before it is returned. The level of inspection of finished products is generally chosen to take account of this possibility. Machine vision holds out the possibility of fast enough operation to allow complete inspection of every product, reducing the cost of sample inspection for both vendor and customer and making the vendor a more attractive supplier.

2.4.4 Automation Aid

As we have already seen, if machine vision is instrumental in the introduction of flexible automation, then it can claim an important share in many tangible and intangible benefits, like reduced works in progress, shorter door-to-door time, possibly a change from manufacturing against orders, shorter lead time from design to production, and capital savings through the purchase of adaptable plant and tooling with a longer life than dedicated equipment. [12]

3 IMAGE PROCESSING

3.1 What is Image Processing?

Digital Image Processing is a term that covers a lot of different areas. A common misconception is that this is a *new* field. Actually, digital images have been created, processed and viewed since very early in this century. But its *practical* usefulness didn't reach very far before reasonably powerful computers became available in the '70s and '80s.[13]

3.2 Overview of Digital Image Processing

		Output	
		Image	Description
Input	Image	image processing	Computer vision <i>image analysis</i>
	Description	computer graphics <i>image synthesis</i>	information processing

Table 3-1 Overview of image processing

Computer vision goes by many aliases: machine vision, image analysis, image understanding, and computational vision. Each has its own connotation and is used by different parts of the community. Traditionally *computer vision* and *machine vision* imply endowing a computer or other machine with visual capabilities, suggesting robotics and other autonomous computer-guided devices. *Image analysis* and *image understanding* generally refer to figuring out the content of images, a part of computer vision but without the implication of the computer having immediate access to live video (*seeing*). *Computational vision* is starting to gain favor as a term that refers to modeling visual processes by computational means. [14]

3.3 Digital Image Fundamentals

3.3.1 Image Acquisition

Scanner, video camera, digital still camera, Photo CD

3.3.2 Image Storage

File formats (TIF, TGA, BMP, GIF, JPG, IMG), HTML

3.3.3 Image Processing

Manipulation, transformation, analysis

3.3.4 Image Display

Output devices (video, print), figures in reports and theses, limitations

3.3.5 Software

XV (Unix)

Lview (Windows)

Matlab

C routines for image processing

Image processing software packages

3.3.6 Image Model

Sampling and Quantization

Types of pixels

Relationships between pixels

Imaging Geometry

3.3.7 Color

Achromatic images

Color models

Color coordinate systems [15]

3.4 Image Processing Steps

3.4.1 Gray Scales



Table 3-2 The adjacent image specifies 256 different shades of gray in a 16 x 16 grid. It is unlikely that you will see all 256 shades unless your browser runs on a 24-bit color display system. You should have no difficulty seeing the changes in gray level that do appear because of a visual effect known as Mach bands.

Grayscale images are images without color, or **achromatic** images. The levels of a grayscale range from 0 (black) to 1 (white).

Each pixel value in a grayscale image corresponds to an amount or quantity of light. In the realm of physics, the terms **intensity** or **luminance** are used to describe the energy density of light. The term **brightness** is often used to describe perceived intensity in the psychological sense of visual perception.[13]

3.4.2 Color Information

In automated image analysis, color is a powerful descriptor that is often simplifies identification.

The purpose of color model is to facilitate the specification of color in some standard.[16] In essence, a color model is a specification of a 3-D coordinate system and a subspace within that system where each color is represented by a single point.

RGB color space is formed to mimic the human color processing. [17] Eye has two different kinds of light receptors, called rods and cones. Rods do not respond to color information. cones are located in the part of eye, called the fovea. According to the color range they respond, cones are divided into three subgroups called: cones responding to blue, cones responding to green, cones responding to red.[18]

Another color space commonly used is Hue, Saturation, Intensity (HIS). The characteristics generally used to distinguish one color from another are, hue, saturation and intensity. Hue is an attribute associated with the dominant wavelength in a mixture of light waves. The Hue represents dominant color as perceived by an observer, when we call an object red, orange or yellow we are specifying its Hue. Saturation refers to relative purity or the amount of white light mixed a hue.[19]

3.4.3 Smoothing Noise

The idea with noise smoothing is to reduce various spurious effects of a local nature in the image, caused perhaps by

- noise in the image acquisition system,
- arising as a result of transmission of the image, for example from a space probe utilizing a low-power transmitter. [20]

There are many noise reduction operators such as ideall low pass filter, neighbourhood averaging, median filter etc. to be used for noise removal.

3.5 Binary Image Processing

In the early days of machine vision, the memory and computing power available was very limited and expensive. These limitations encouraged designers of vision applications to focus their efforts on binary vision systems. A binary image contains only two gray levels. [21]

In addition, designers noted that people have no difficulty in understanding line drawings, silhouettes, and other images formed using only two gray levels. Encouraged by this human capability, they used binary images in many applications.[22]

Even though computers have become much more powerful, binary vision systems are still useful. First of all, the algorithms for computing properties of binary images are well understood. They also tend to be less expensive and faster than vision systems that operate on gray level or color images. This is due to the significantly smaller memory and processing requirements of binary vision. The memory requirements of a gray level system working with 256 gray levels will be eight times that of a system working with a binary image of the same size. The processing time requirements are lower because many operations on binary images may be performed as logical operations instead of integer arithmetic operations.

Smaller memory requirements and faster execution times are not the only reasons for studying binary vision systems. Many techniques developed for these systems are also applicable to vision systems which use gray scale images. A convenient way to represent an object, in a gray level or color image is to use its mask. The mask of an object is a binary picture in which the object points are 1 and other points are 0.

3.5.1 Thresholding

One of the most important problems in a vision system is to identify the subimages that represent objects. This operation, which is so natural and so easy for people, is surprisingly difficult for computers. The partitioning of an image into regions is called segmentation. Ideally, a partition represents an object or part of an object.

Thresholding is a method to convert a gray scale image into a binary image so that objects of interest are separated from the background. For thresholding to be effective in object-background separation, it is necessary that the objects and background have sufficient contrast and that we know the intensity levels of either the objects or the background. In a fixed thresholding scheme, these intensity characteristics determine the value of the threshold.

Automatic thresholding of images is often the first step in the analysis of images in machine vision systems. Many techniques have been developed for utilizing the intensity distribution in an image and the knowledge about the objects of interest for selecting a proper threshold value automatically.

3.6 Edge Detection

The early stages of vision processing identify features in images that are relevant to estimating the structure and properties of objects in a scene. Edges are one such feature. Edges are significant local changes in the image and are important features for analyzing images. Edges typically occur on the boundary between two different regions in an image. Edge detection is frequently the first step in recovering information from images. Due to its importance, edge detection continues to be an active research area. [23]

There are many edge detection operators. Some of them are explained below.

3.6.1 Edge Detection Operators

3.6.1.1 Roberts Operator

The Roberts cross operator provides a simple approximation to the gradient magnitude: [24]

$$G[f[i, j]] = |f[i, j] - f[i + 1, j + 1]| + |f[i + 1, j] - f[i, j + 1]| \quad (1)$$

Using convolution masks; this becomes

$$G[f[i, j]] = |G_x| + |G_y| \quad (2)$$

where G_x and G_y are calculated using the following masks:

G_x	G_y								
<table border="1" style="border-collapse: collapse; width: 40px; height: 40px;"> <tr><td style="padding: 5px;">1</td><td style="padding: 5px;">0</td></tr> <tr><td style="padding: 5px;">0</td><td style="padding: 5px;">-1</td></tr> </table>	1	0	0	-1	<table border="1" style="border-collapse: collapse; width: 40px; height: 40px;"> <tr><td style="padding: 5px;">0</td><td style="padding: 5px;">-1</td></tr> <tr><td style="padding: 5px;">1</td><td style="padding: 5px;">0</td></tr> </table>	0	-1	1	0
1	0								
0	-1								
0	-1								
1	0								

3.6.1.2 Sobel Operator

Like the other gradient operators, s_x and s_y can be implemented using convolution masks:

Sx

-1	0	1
-2	0	2
-1	0	1

Sy

1	2	1
0	0	0
-1	-2	-1

Note that this operator places an emphasis on pixels that are closer to the center of the mask.

The Sobel operator is the magnitude of the gradient computed by

$$M = \sqrt{s_x^2 + s_y^2} \quad (3)$$

In this study this operator was used for edge detection.

3.6.1.3 Prewitt operator

The Prewitt Operator is as follow:

$$M = \sqrt{s_x^2 + s_y^2}$$

where Sx and Sy are:

Sx

-1	0	1
-1	0	1
-1	0	1

Sy

1	1	1
0	0	0
-1	-1	-1

4 PATTERN CLASSIFICATION

4.1 Introduction

To identify patterns in images various techniques can be used to make judicious inferences from the available information. In this chapter some of the statistical techniques for pattern classification are briefly reviewed. [25]

4.2 Artificial Neural Networks

It is well known that neural classifiers suffer from small sample size problems. Training a classifier on too few samples will generally result in classifiers having a poor generalization capability.[26] Such classifiers have adapted too much to the noise in the dataset and are therefore expected to have an increased nonlinearity.

4.2.1 Multilayer Perceptrons

One useful ANN architecture is the so-called multi-layer perceptron, or feed-forward neural network. The network consists of a number of hidden layers, one input layer and one output layer. In supervised learning, often a **back-propagation** algorithm is used to change the weights.[27]

4.2.2 Adaline Algorithm

Widrow[28] and Hoff suggested Adaline as a supervised neural network algorithm. It is trained by a gradient descent rule, called Least Mean Squares (LMS), which is very similar to the perceptron-learning algorithm. [29]

In many real life applications, classes are not well separated. Applying single perceptron as a classifier gives poor results. The reason is that the error fed back takes binary values.

4.3 Fuzzy Logic

Fuzzy logic is a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth - truth-values between "completely true" and "completely false". It was introduced by Dr. Lotfi Zadeh of U.C. Berkeley in the 1960's.

There is a strong relationship between Boolean logic and the concept of a subset. There is a similar strong relationship between fuzzy logic and fuzzy subset theory.[30]

4.4 Inductive Learning

In recent years, there has been a growing amount of research on inductive learning.[31] In its broadest sense, induction (or inductive inference) *is a method of moving from the particular to the general - from specific examples to general rules.* [32]

Induction can be considered as the process of generalizing a procedural description from presented or observed examples [33].

The purpose of inductive learning is to perform a synthesis of new knowledge, and this is independent of the form given to the input information [34].

The outcome of an induction algorithm is either a decision tree or a set of rules. Production rules can easily be extracted from decision trees [35] [36].

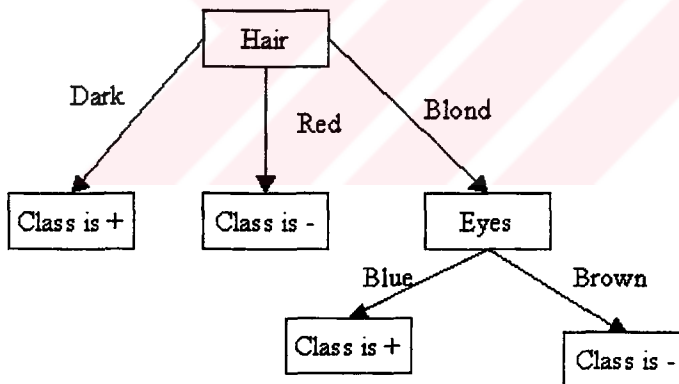


Figure 4-1 An example Decision tree

Each path of a decision tree can be regarded as an IF-THEN production rule. For example, the following production rules can be obtained from the decision tree shown in Figure 4.1

IF hair IS dark THEN Class IS +

IF hair IS red THEN Class IS -

IF hair IS blond AND eyes ARE blue THEN Class IS +

IF hair IS blond AND eyes ARE brown THEN Class IS -

4.4.1 RULES-3 Inductive Learning Algorithm

RULES-3 [1] is a simple algorithm for extracting a set of classification rules from a collection of examples for objects belonging to one of a number of known classes. An object must be described in terms of a fixed set of attributes, each with its own range of possible values, which could be nominal or numerical. For example, attribute "length" might have nominal values {short, medium, long} or numerical values in the range {-10, 10}.

An attribute-value pair constitutes a condition in a rule. If the number of attributes is N_a , a rule may contain between one and N_a conditions. Only conjunction of conditions is permitted in a rule and therefore the attributes must be all different if the rule comprises more than one condition.[1]

RULES-3 extracts rules by considering one example at a time. It forms an array consisting of all attribute-value pairs associated with the object in that example, the total number of elements in the array being equal to the number of attributes of the object. The rule farming procedure may require at most N_a iterations per example. In the first iteration, rules may be produced with one element from the array. Each element is examined in turn to see if, for the complete example collection, it appears only in objects belonging to one class. If so, a candidate rule is obtained with that element as the condition. In either case, the next element is taken and the examination repeated until all elements in the array have been considered. At this stage, if no rules have been formed, the second iteration begins with two elements of the array being examined at a time. Rules formed in the second iteration therefore have two conditions. The procedure continues until iteration when one or more candidate rules can be extracted or the maximum number of iterations for the example is reached. In the latter case, the example itself is adopted as the rule. If more than one candidate rule is formed for an example, the rule that classifies the highest number of examples is selected and used to classify objects in the collection of examples. Examples of which objects are classified by the selected rule are removed from the collection. The next example remaining in the collection is then taken and rule extraction is carried out for

that example. This procedure continues until there are no examples left in the collection and all objects have been classified. This algorithm can be summarized as follows:

Step 1. Define ranges for the attributes, which have numerical values and assign labels to those ranges

Step 2. Set the minimum number of conditions (N_{cmin}) for each rule

Step 3. Take an unclassified example

Step 4. $N_c = N_{cmin} - 1$

Step 5. If $N_c < N_a$ then $N_c = N_{c+1}$

Step 6. Take all values or labels contained in the example

Step 7. Form objects which are combinations of N_c values or labels taken from the values or labels obtained in Step 6

Step 8. If at least one of the objects belongs to a unique class then form rules with those objects; ELSE go to Step 5

Step 9. Select the rule, which classifies the highest number of examples

Step 10. Remove examples classified by the selected rule

Step 11. If there are no more unclassified examples then STOP; ELSE

go to Step 3

Here N_c is the number of condition(s) for each rule and N_a is the number of attributes for each example.

4.4.2 A simple Problem Illustrating the Operation of RULES-3

The following problem will be used to illustrate the operation of RULES-3 step by step. The problem, adapted from , involves three nominal attributes: Car-Size, Fuel and Max-Speed. Car-Size has three values, large, compact and medium. Fuel has three values, propane, petrol and diesel. Max-Speed has three values; high, average and low. There are

three classes of acceleration, good, excellent and poor. The example set for the problem is given in Table 4.1

No	Car-size	Fuel	Max.Speed	Acceleration
1	Large	Diesel	High	Good
2	Large	Propane	High	Good
3	Compact	Petrol	High	Excellent
4	Large	Petrol	High	Excellent
5	Medium	Diesel	Low	Good
6	Compact	Petrol	Low	Good
7	Medium	Petrol	Average	Excellent
8	Medium	Diesel	Average	Poor

Table 4-1 Training set for car acceleration problem

In Step 1, because in the problem none of the attributes have numerical values, ranges are not defined.

In Step 2, the minimum number of conditions is set to 1.

In Steps 3-8, two rules are extracted from example 1. These are:

IF Car-Size IS large AND Fuel IS diesel THEN Acceleration IS good.

IF Fuel IS diesel AND Max-Speed IS high THEN Acceleration IS good.

In Step 9, because each of these rules can classify only one example (example 1), one of them can be chosen. If the first rule is chosen, in Step 10, example 1 is classified using this rule. In step 11, the example is removed from the list of unclassified examples. The test in Step 12 for examples remaining to be classified is positive. Therefore, the procedure returns to Step 3 for a new iteration.

In the second iteration, there are seven unclassified examples. One rule can be extracted from example 2, namely,

IF Fuel IS propane THEN Acceleration IS good. Only example 2 can be classified using this rule and so it is removed from the list of unclassified examples.

In the third iteration, two rules are extracted from example 3. They are:

IF Car-Size IS compact AND Max-Speed IS high THEN Acceleration IS excellent.

IF Fuel IS petrol AND Max-Speed IS high THEN Acceleration IS excellent.

The first rule classifies only one example (example 3). The second rule can classify two examples (examples 3 and 4). Therefore, the second rule is selected. Using this rule, examples 3 and 4 can be classified and so they are removed from the list of unclassified examples.

In the fourth iteration, one rule can be extracted from example 5, namely,

IF Max-Speed IS low THEN Acceleration IS good. Using this rule, examples 5 and 6 can be classified and they are removed from the list of unclassified examples.

In the fifth iteration, two rules are extracted from example 7. These are:

IF Car-size IS Medium AND Fuel IS Petrol THEN Acceleration IS Excellent.

If Fuel IS Petrol AND Max-Speed IS Average THEN Acceleration IS Excellent.

As each of these rules can classify only one example (example 7), either of them can be chosen. If the first rule is chosen, example 7 can be classified It is deleted from the list of unclassified examples.

In the sixth iteration, only one rule can be extracted from example 8, namely,

IF Fuel IS Diesel AND Max-Speed IS Average THEN Acceleration IS Poor

Since there are no more unclassified examples, the procedure ends. The above rule induction sequence is summarized in Table 4.2

Iteration	Example Considered	Number of Extracted Rules	Classified Examples	Unclassified Examples
1	1	2	1	2,3,4,5,6,7,8
2	2	1	2	3,4,5,6,7,8
3	3	2	3,4	5,6,7,8
4	5	1	5,6	7,8
5	7	2	7	8
6	8	1	8	-

Table 4-2 Summary of rule induction sequence for car acceleration problem

5 IMPLEMENTATION

In quality control process of ceramic tiles the aim is to figure out if the product is good or defective. In order to get detailed control information defected products are defined as spot and crack. If there are some spots or cracks on the product the system learns it as spot or crack, otherwise the product is good. That is:

If the product is acceptable then the class is : 'GOOD'

If there is some spots on the tile then the class is : 'SPOT'

If there is some cracks on the tile then the class is : 'CRACK'

Some example of good, spot and crack tiles are given in figure 5.1.

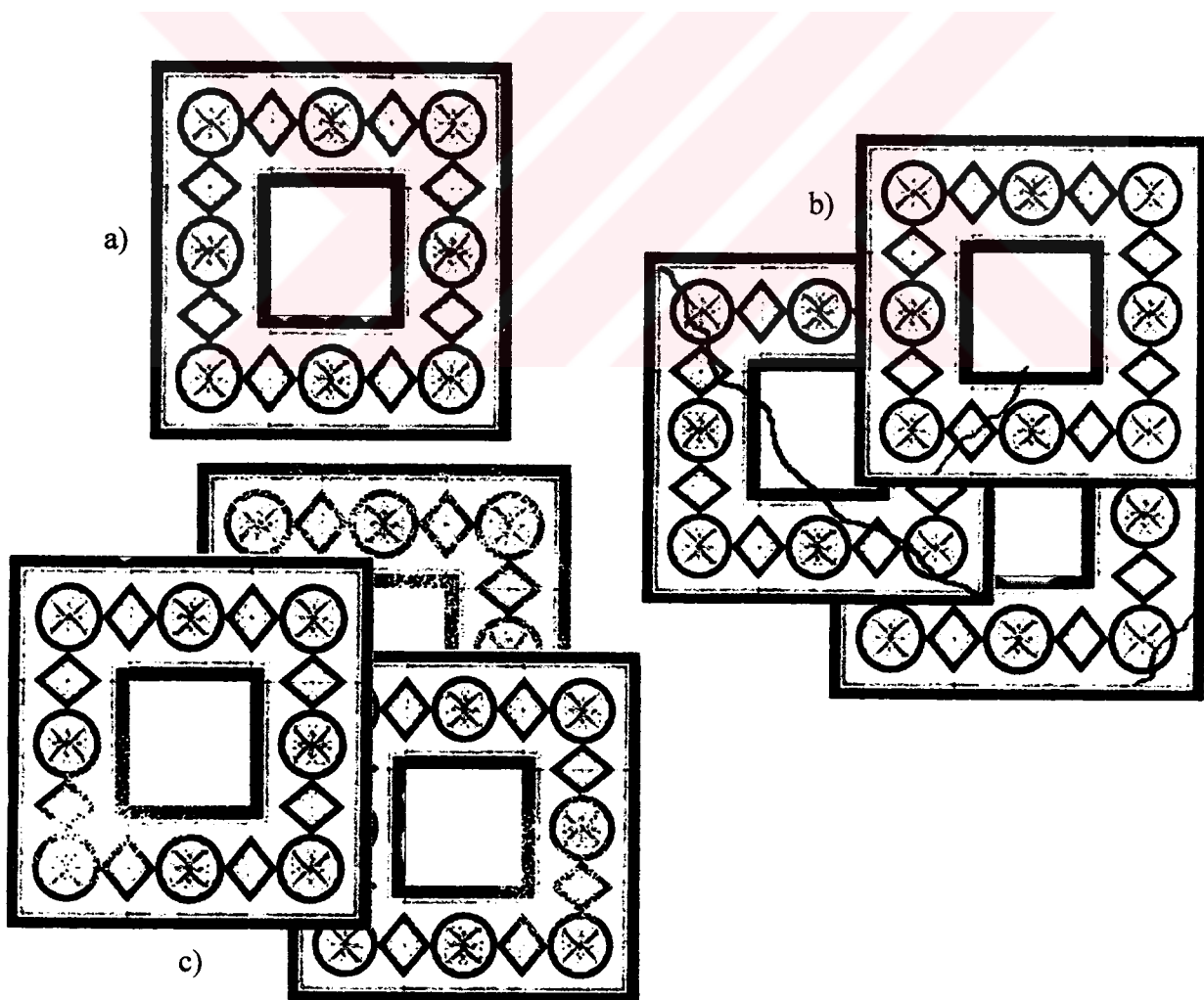


Figure 5-1 a) good product b) crack c) spot

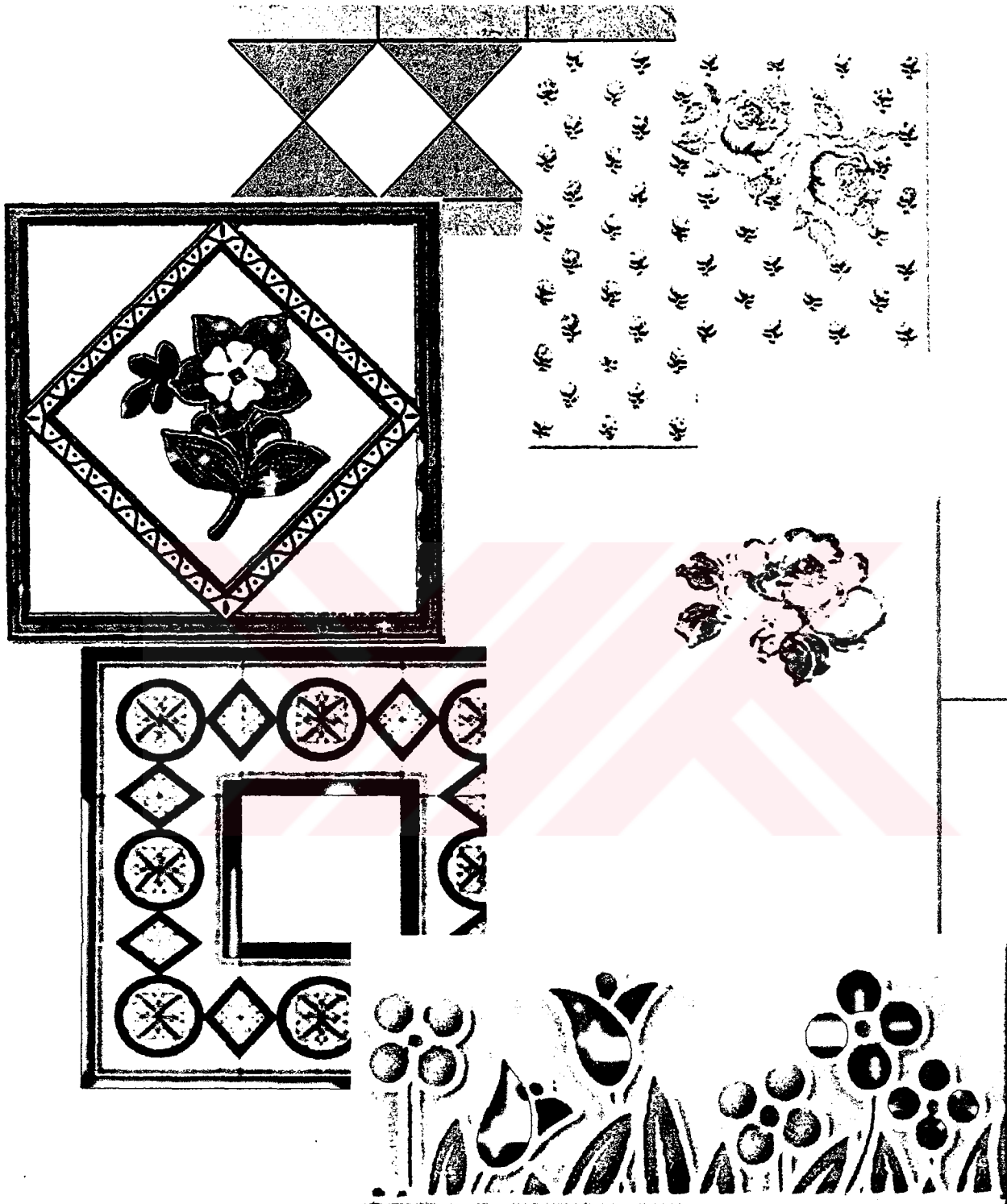


Figure 5-2 Some tiles used for image processing and learning.

5.1 How the System Works

The steps shown below are applied to recognize the tile. The first eight steps are about image processing, and the rest is about the recognition of the product.

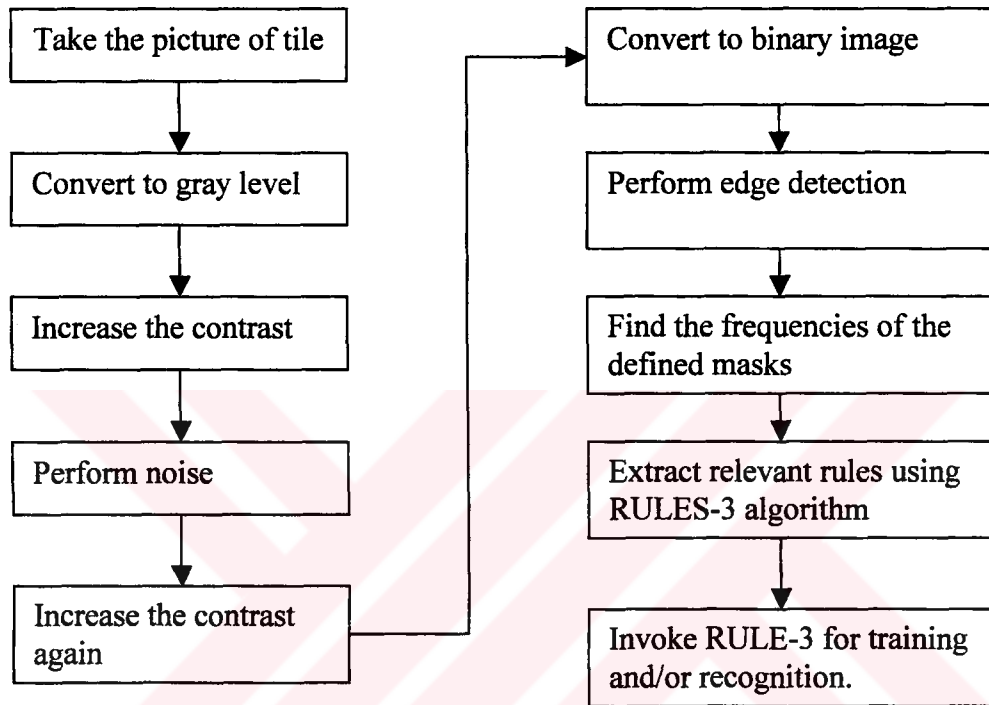


Figure 5-3 How the system works

5.2 Image Processing Steps

5.2.1 Setting Up The Inspection System

First of all the system snaps the picture of tile. A tv camera is suitable for this task. In this system pallets are also other necessary equipment to bring the products in an order. The place where pictures are snapped must have suitable situation with lighting system. Camera must be connected to a computer.

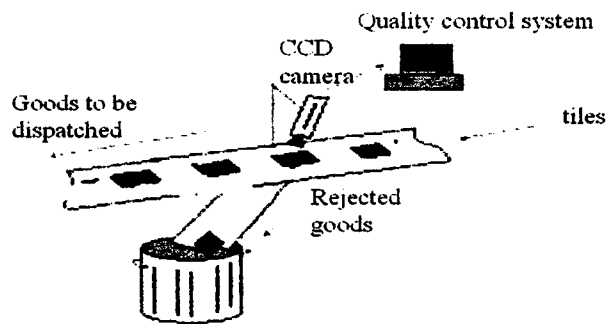


Figure 5-4 Visual inspection system

After the system is ready the picture of the product is snapped for training. For a good training a lot of image in three classes must be given to system. Here image-processing steps will be shown for only one good tile. The same steps are applied to all good and defective tiles for training and/or recognition.

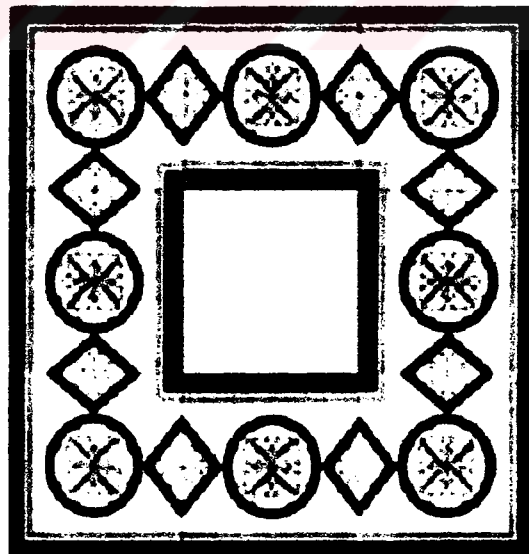


Figure 5-5 Original picture of a tile

Because of its high speed MATLAB software was chosen for all steps of image processing operations.

In MATLAB, *imread(name)* function is performed for snapping the image

5.2.2 Gray Level

First step of image processing operation is to convert picture to a gray level. Original image has different colors, which has three-dimensional color of information. These are red, blue and green colors. Each of them has 256 color levels. Our aim here is to decrease this three-dimensional color information to one. In order to find the gray level value the average of three color values is calculated using the formula given below:

$$\text{Gray level value} = \frac{(\text{red value} + \text{green value} + \text{blue value})}{3}$$

For example, if the pixel color values are 145 for red, 220 for blue and 150 for green the gray level value is: $(145+220+150) / 3 = 171,666 \approx 172$

This operation is applied to all pixel of the image.

The function for converting to gray level is: *rgb2gray(name)*

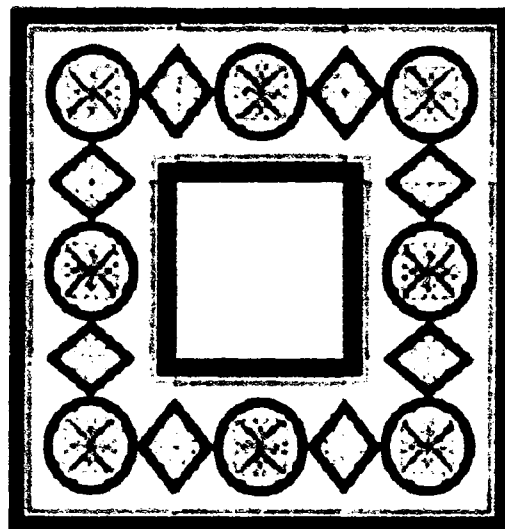


Figure 5-6 Gray level image

5.2.3 Increase the Contrast of Gray Level

While passing from gray level to binary image some information may be lost. In order to decrease the amount of lost information we should increase the contrast of image.

The function is : *imadjust(b, [0.3 0.5])*

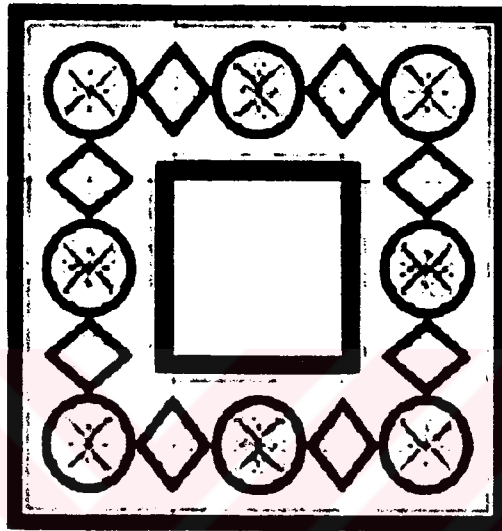


Figure 5-7 After increasing contrast of gray level image

5.2.4 Noise Smoothing

Noise smoothing is to reduce various effects of a local nature in the image, caused perhaps by

- noise in the image acquisition system,
- arising as a result of transmission of the image, for example from a space probe utilizing a low-power transmitter.

For noise smoothing Median filtering method was used because of its good efficiency.

The function for noise smoothing is: *fspecial('average',3)*

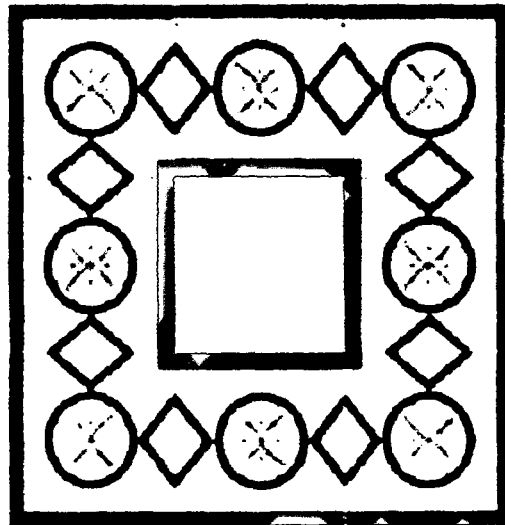


Figure 5-8 Median filtered image

5.2.5 Increase the Contrast Level of Filtered Image

For the next processes we have to get a binary (black and white) image. To do this we recontrast the image.

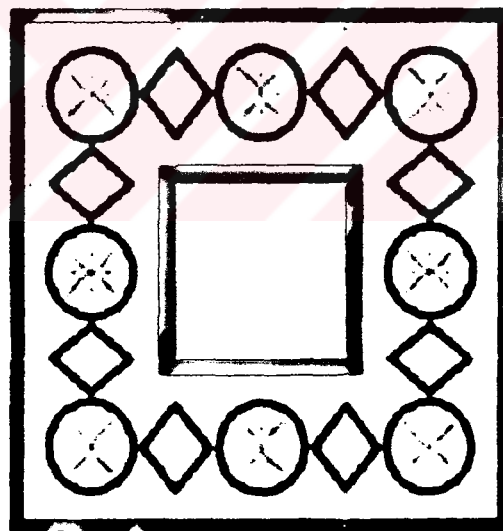


Figure 5-9 Contrast of filtered image

5.2.6 Convert to Binary Image

5.2.6.1 Thresholding Value.

As mentioned in Chapter-3 in order to get the binary image first we must determine the threshold value. Finding the threshold value is very important and difficult process in image processing. Threshold value is a value that is between 0 – 256 and used to

decide if the pixel will be black or white. If the pixels value is greater than threshold value pixel turns white color otherwise it turns black color. This operation is also applied to all pixels to find their color. Because each tile has different colors threshold value must be determined for each of them. After many experiments the threshold value was decided to be 230 for the tile being processed.

The function is: $im2bw(d, 9)$. (0,9 is threshold value)

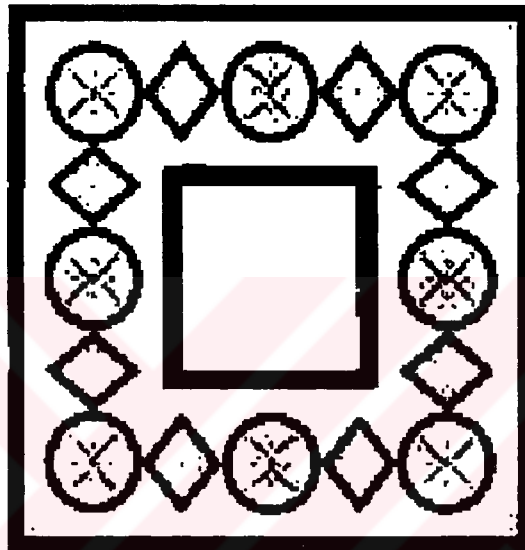


Figure 5-10 Binary image

5.2.7 Edge Detection

After getting binary image Edge detection operation is applied to the image. As mentioned in Chapter-3, an edge detection operator such as roberts, prewit, kirsch compass, robinson compass, canny and frei-chen operators etc must be chosen. In this study Sobel operator is used because of its good performance among the operators tested for this study.

$(i-1, j-1)$	$(i-1, j)$	$(i-1, j+1)$
$(i, j-1)$	(i, j)	$(i, j+1)$
$(i+1, j-1)$	$(i+1, j)$	$(i+1, j+1)$

Table 5-1 Pixel position

Here i and j are the coordinates of the pixel. Table 5-1 shows the pixels positions. $f(I, j)$ is the intensity of the pixel having (i, j) coordinates.

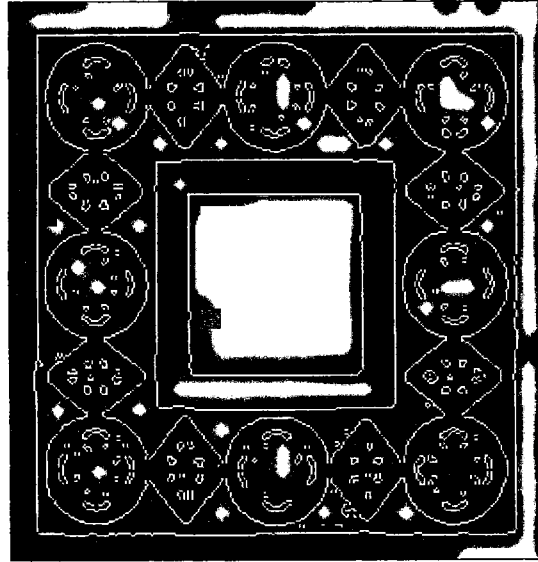


Figure 5-11 Edges of the images

After edge detection operation every pixels of the image are scanned from left to right and from top to bottom using the masks given in Table 5.2.

The function is: `edge(f,'sobel');`

Table 5-2 All possible 3*3 masks

In order to determine the edges, 3*3 masks given in Table 5.2 are used. The number of all possible masks is 28. The set of masks given in Table 5-3 have the highest frequency so the rest of masks are eliminated.

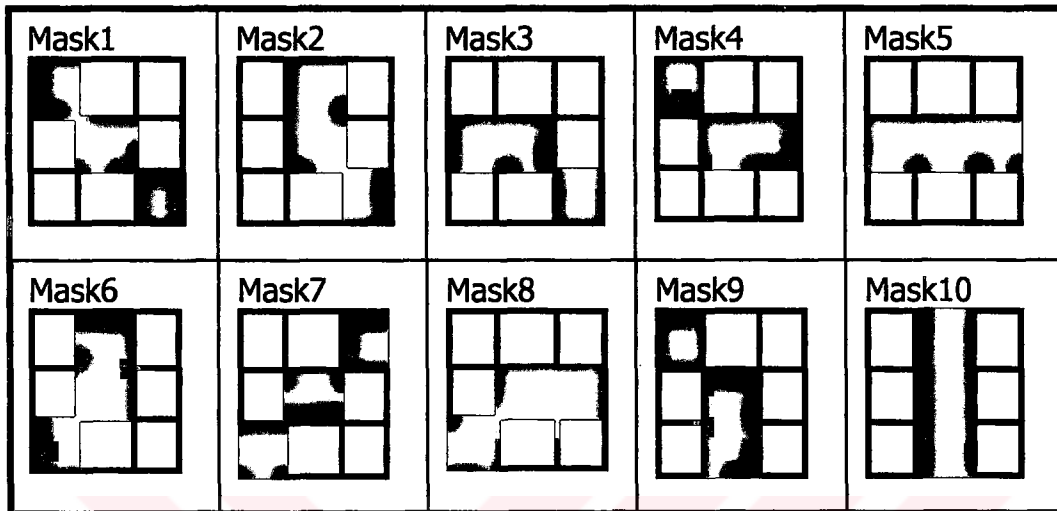


Table 5-3 Masks used for this tile because of their high frequencies

After applying these masks we got the results given in Table 5.4. This table shows the frequency of each mask. For example for the processed tile mask3 is matched 187 times. The use of frequencies of masks is also the meaning of the image.

Mask1	Mask2	Mask3	Mask4	Mask5	Mask6	Mask7	Mask8	Mask9	Mask10	Class
187	105	109	110	784	107	165	124	109	787	good

Table 5-4 Frequency of masks for 'good' tile.

Since our first aim is to teach the computer if the product is good or not both good and defective products' images are processed. Teaching the system with much example increases the efficiency of learning. More than 20 example of each kind of products have been used to teach this product to the computer. Some results are given in Table 5.5.

Fmask program is used to find the frequency of the masks.

Mask1	Mask2	Mask3	Mask4	Mask5	Mask6	Mask7	Mask8	Mask9	Mask10	Class
46	67	75	67	437	63	43	60	65	442	Spot
68	62	80	65	603	72	63	59	70	639	Spot
92	74	87	81	723	73	84	83	79	656	Spot
87	79	82	78	660	73	80	80	81	642	Spot
58	68	75	73	495	69	63	76	68	538	Spot
59	64	69	61	601	70	66	72	59	556	Spot
48	75	61	57	398	72	55	66	76	413	Spot
66	77	82	82	611	68	68	78	87	566	Spot
70	35	36	47	164	23	57	46	70	271	Spot
60	62	43	39	31	25	83	57	79	128	Spot
61	57	53	55	342	67	62	73	58	398	Spot
45	62	64	60	459	63	46	65	55	373	Spot
54	38	27	35	22	33	50	30	28	36	Spot
34	29	24	25	42	20	38	21	27	15	Spot
43	79	77	66	437	62	59	62	74	392	Spot
17	13	22	23	32	21	40	29	14	19	Spot
121	90	106	96	673	112	99	95	95	799	Crack
134	110	104	95	694	90	93	89	113	720	Crack
117	86	102	96	717	90	98	91	89	751	Crack
122	93	103	92	753	93	106	98	96	730	Crack
115	101	101	94	708	87	96	87	98	760	Crack
125	89	101	107	764	100	105	90	97	731	Crack
134	81	105	99	677	83	96	101	103	791	Crack
122	91	98	96	710	90	107	90	94	757	Crack
123	92	105	96	699	105	108	99	96	792	Crack
126	92	102	94	725	96	125	102	94	728	Crack
120	86	106	100	741	82	97	92	88	738	Crack
126	91	105	94	693	109	129	99	97	780	Crack
125	108	105	94	677	86	97	91	109	750	Crack
123	92	102	94	718	91	100	88	91	757	Crack
118	100	103	95	705	88	94	91	100	754	Crack
115	86	94	91	715	89	90	93	88	776	Crack
111	91	98	89	706	95	100	98	95	735	Crack
116	94	103	96	711	90	95	87	95	759	Crack
123	93	105	97	715	96	97	89	98	722	Crack
165	101	107	118	795	102	170	122	98	794	Good
183	105	102	120	787	109	161	124	114	834	Good
187	105	109	110	784	107	165	124	109	787	Good
185	99	104	114	771	109	165	132	116	833	Good

Table 5-5 Set of examples for training

What we have done so far was about image processing. As mentioned in Chapter 6 we need the set of examples to extract the rules. In order to get the rules, RULES-3 inductive learning algorithm is employed. Here frequencies of the masks and the classes they belong to are used as the set of examples. (See Table 7-5) RULES-3 extract the rules to classify the tiles. The quantisation level for each masks was set to 7 and minimum number of conditions for each rule was decided to be 10 and 50 rules were extracted. Since minimum number of conditions was decided to be 10, all masks' frequencies were taken into an account. If it was decided to be for example 5, then minimum 5 masks would be used for extracting a rule. When minimum number of conditions was decided to be 5 we extracted 33 rules. Some of them are shown in Table 5.8

After applying RULES-3 algorithm we got the rules given in Table 5.6. and Table 5.7.

Rule: 1 IF 51=<m01<85 AND 53=<m02<73 AND 60=<m03<79 AND 59=<m04<77 AND 484=<m05<638 AND 60=<m06<80 AND 64=<m07<90 AND 63=<m08<84 AND 54=<m09<74 AND 537=<m10<711 THEN CLASS IS spot	Rule: 2 IF 153=<m01<187 AND 93=<m02<113 AND 98=<m03<117 AND 95=<m04<113 AND 638=<m05<792 AND 100=<m06<120 AND 142=<m07<168 AND 105=<m08<126 AND 94=<m09<114 AND 711=<m10<885 THEN CLASS IS good	Rule: 3 IF 85=<m01<119 AND 93=<m02<113 AND 98=<m03<117 AND 77=<m04<95 AND 638=<m05<792 AND 80=<m06<100 AND 90=<m07<116 AND 84=<m08<105 AND 94=<m09<114 AND 711=<m10<885 THEN CLASS IS crack	Rule: 4 IF 51=<m01<85 AND 53=<m02<73 AND 41=<m03<60 AND 41=<m04<59 AND 330=<m05<484 AND 60=<m06<80 AND 38=<m07<64 AND 63=<m08<84 AND 54=<m09<74 AND 363=<m10<537 THEN CLASS IS spot
Rule: 5 IF 85=<m01<119 AND 73=<m02<93 AND 79=<m03<98 AND 77=<m04<95 AND 638=<m05<792 AND 80=<m06<100 AND 90=<m07<116 AND 84=<m08<105 AND 74=<m09<94 AND 711=<m10<885 THEN CLASS IS crack	Rule: 6 IF 119=<m01<153 AND 73=<m02<93 AND 98=<m03<117 AND 77=<m04<95 AND 638=<m05<792 AND 100=<m06<120 AND 116=<m07<142 AND 84=<m08<105 AND 94=<m09<114 AND 711=<m10<885 THEN CLASS IS crack	Rule: 7 IF 119=<m01<153 AND 73=<m02<93 AND 98=<m03<117 AND 95=<m04<113 AND 638=<m05<792 AND 80=<m06<100 AND 90=<m07<116 AND 84=<m08<105 AND 74=<m09<94 AND 711=<m10<885 THEN CLASS IS crack	Rule: 8 IF 119=<m01<153 AND 73=<m02<93 AND 98=<m03<117 AND 77=<m04<95 AND 638=<m05<792 AND 80=<m06<100 AND 116=<m07<142 AND 84=<m08<105 AND 94=<m09<114 AND 711=<m10<885 THEN CLASS IS crack

Table 5-6 Extracted rules (part a)

T.C. MİLLÎ EĞİTİM BAKANLIĞI
 MÜHÜR
 MÜHÜR

<p>Rule: 9 IF 17=<m01<51 AND 53=<m02<73 AND 60=<m03<79 AND 59=<m04<77 AND 330=<m05<484 AND 60=<m06<80 AND 38=<m07<64 AND 42=<m08<63 AND 54=<m09<74 AND 363=<m10<537 THEN CLASS IS spot</p>	<p>Rule: 10 IF 51=<m01<85 AND 53=<m02<73 AND 79=<m03<98 AND 59=<m04<77 AND 484=<m05<638 AND 60=<m06<80 AND 38=<m07<64 AND 42=<m08<63 AND 54=<m09<74 AND 537=<m10<711 THEN CLASS IS spot</p>	<p>Rule: 11 IF 85=<m01<119 AND 73=<m02<93 AND 79=<m03<98 AND 77=<m04<95 AND 638=<m05<792 AND 60=<m06<80 AND 64=<m07<90 AND 63=<m08<84 AND 74=<m09<94 AND 537=<m10<711 THEN CLASS IS spot</p>	<p>Rule: 12 IF 51=<m01<85 AND 53=<m02<73 AND 60=<m03<79 AND 59=<m04<77 AND 484=<m05<638 AND 60=<m06<80 AND 38=<m07<64 AND 63=<m08<84 AND 54=<m09<74 AND 537=<m10<711 THEN CLASS IS spot</p>
<p>Rule: 13 IF 17=<m01<51 AND 73=<m02<93 AND 60=<m03<79 AND 41=<m04<59 AND 330=<m05<484 AND 60=<m06<80 AND 38=<m07<64 AND 63=<m08<84 AND 74=<m09<94 AND 363=<m10<537 THEN CLASS IS spot</p>	<p>Rule: 14 IF 51=<m01<85 AND 33=<m02<53 AND 22=<m03<41 AND 23=<m04<41 AND 22=<m05<176 AND 20=<m06<40 AND 38=<m07<64 AND 21=<m08<42 AND 14=<m09<34 AND 15=<m10<189 THEN CLASS IS spot</p>	<p>Rule: 15 IF 17=<m01<51 AND 13=<m02<33 AND 22=<m03<41 AND 23=<m04<41 AND 22=<m05<176 AND 20=<m06<40 AND 38=<m07<64 AND 21=<m08<42 AND 14=<m09<34 AND 15=<m10<189 THEN CLASS IS spot</p>	<p>Rule: 16 IF 51=<m01<85 AND 73=<m02<93 AND 79=<m03<98 AND 77=<m04<95 AND 484=<m05<638 AND 60=<m06<80 AND 64=<m07<90 AND 63=<m08<84 AND 74=<m09<94 AND 537=<m10<711 THEN CLASS IS spot</p>
<p>Rule: 17 IF 17=<m01<51 AND 53=<m02<73 AND 60=<m03<79 AND 59=<m04<77 AND 330=<m05<484 AND 60=<m06<80 AND 38=<m07<64 AND 63=<m08<84 AND 54=<m09<74 AND 363=<m10<537 THEN CLASS IS spot</p>	<p>Rule: 18 IF 17=<m01<51 AND 73=<m02<93 AND 60=<m03<79 AND 59=<m04<77 AND 330=<m05<484 AND 60=<m06<80 AND 38=<m07<64 AND 42=<m08<63 AND 74=<m09<94 AND 363=<m10<537 THEN CLASS IS spot</p>	<p>Rule: 19 IF 85=<m01<119 AND 73=<m02<93 AND 98=<m03<117 AND 95=<m04<113 AND 638=<m05<792 AND 80=<m06<100 AND 90=<m07<116 AND 84=<m08<105 AND 74=<m09<94 AND 711=<m10<885 THEN CLASS IS crack</p>	<p>Rule: 20 IF 119=<m01<153 AND 93=<m02<113 AND 98=<m03<117 AND 77=<m04<95 AND 638=<m05<792 AND 80=<m06<100 AND 90=<m07<116 AND 84=<m08<105 AND 94=<m09<114 AND 711=<m10<885 THEN CLASS IS crack</p>

Table 5-7 Extracted rules (part b)

<p>Rule: 1 IF 17=<m01<51 AND 53=<m03<73 AND 60=<m03<79 AND 59=<m04<77 AND 330=<m05<484 THEN CLASS IS spot</p>	<p>Rule: 2 IF 51=<m01<85 AND 53=<m03<73 AND 79=<m03<98 AND 59=<m04<77 AND 484=<m05<638 THEN CLASS IS spot</p>	<p>Rule: 3 IF 17=<m01<51 AND 73=<m03<93 AND 60=<m03<79 AND 41=<m04<59 AND 330=<m05<484 THEN CLASS IS spot</p>
<p>Rule: 4 IF 51=<m01<85 AND 33=<m03<53 AND 22=<m03<41 AND 23=<m04<41 AND 22=<m05<176 THEN CLASS IS spot</p>	<p>Rule: 5 IF 17=<m01<51 AND 73=<m03<93 AND 60=<m03<79 AND 59=<m04<77 AND 330=<m05<484 THEN CLASS IS spot</p>	<p>Rule: 6 IF 85=<m01<119 AND 73=<m03<93 AND 98=<m03<117 AND 95=<m04<113 AND 638=<m05<792 THEN CLASS IS crack</p>
<p>Rule: 7 IF 119=<m01<153 AND 73=<m03<93 AND 98=<m03<117 AND 77=<m04<95 AND 116=<m07<142 THEN CLASS IS crack</p>	<p>Rule: 8 IF 119=<m01<153 AND 73=<m03<93 AND 98=<m03<117 AND 95=<m04<113 AND 638=<m05<792 THEN CLASS IS crack</p>	<p>Rule: 9 IF 85=<m01<119 AND 93=<m03<113 AND 98=<m03<117 AND 95=<m04<113 AND 638=<m05<792 AND 80=<m06<100 AND 90=<m07<116 AND 711=<m10<885 THEN CLASS IS crack</p>
<p>Rule: 10 IF 119=<m01<153 AND 93=<m03<113 AND 98=<m03<117 AND 95=<m04<113 AND 638=<m05<792 THEN CLASS IS crack</p>	<p>Rule: 11 IF 119=<m01<153 AND 73=<m03<93 AND 98=<m03<117 AND 77=<m04<95 AND 638=<m05<792 AND 80=<m06<100 AND 74=<m09<94 THEN CLASS IS crack</p>	<p>Rule: 12 IF 119=<m01<153 AND 73=<m03<93 AND 98=<m03<117 AND 77=<m04<95 AND 638=<m05<792 AND 80=<m06<100 AND 90=<m07<116 AND 711=<m10<885 THEN CLASS IS crack</p>
<p>Rule: 13 IF 85=<m01<119 AND 93=<m03<113 AND 98=<m03<117 AND 95=<m04<113 AND 638=<m05<792 AND 80=<m06<100 AND 90=<m07<116 AND 537=<m10<711 THEN CLASS IS spot</p>	<p>Rule: 14 IF 85=<m01<119 AND 73=<m03<93 AND 79=<m03<98 AND 77=<m04<95 AND 64=<m07<90 THEN CLASS IS spot</p>	<p>Rule: 15 IF 153=<m01<187 AND 93=<m03<113 AND 98=<m03<117 AND 95=<m04<113 AND 638=<m05<792 THEN CLASS IS good</p>

Table 5-8 Extracted rules when minimum number of conditions is five (5).

There are 20 rules for three kinds of tiles. After teaching the process the system is ready for test new products. Now we have to test the system whether it has learned the good and defective tiles well or not.

As an example a crack tile was asked to the system, and it recognised that the class was 'CRACK'.

Mask1	Mask2	Mask3	Mask4	Mask5	Mask6	Mask7	Mask8	Mask9	Mask10	Class
123	92	104	94	734	90	122	93	95	758	Crack

Table 5-9 An unseen example

The rule used for this example is given below:

Rule: 6

IF 119=<m01<153 AND

73=<m02<93 AND

98=<m03<117 AND

77=<m04<95 AND

638=<m05<792 AND

80=<m06<100 AND

116=<m07<142 AND

84=<m08<105 AND

94=<m09<114 AND

711=<m10<885

THEN

CLASS IS CRACK

In order to test to performance of the system 100 unseen examples in three classes were asked to computer to recognize them. Some of them are shown in Table 5.10. The efficiency of the system for correctly classifying unseen exsamples was up to 96 %. This is

a high performance. As it is said before if more examples are used for training the efficiency will be better.

Figure 5-12 shows the three different products' images after image processing steps are applied.



Mask1	Mask2	Mask3	Mask4	Mask5	Mask6	Mask7	Mask8	Mask9	Mask10	Class
29	74	51	61	189	43	68	55	68	485	Spot
130	83	106	99	720	89	124	89	106	701	Crack
164	106	103	120	763	107	168	133	108	788	Good
55	46	32	60	496	69	83	62	49	25	Spot
123	92	112	95	706	90	94	93	107	837	Crack
114	90	95	103	680	95	110	84	101	702	Crack
192	99	100	92	781	100	171	122	104	775	Good
38	52	41	71	259	59	50	67	56	603	Spot
47	29	46	62	208	22	73	32	16	299	Spot
136	94	97	104	741	100	84	83	91	760	Crack
123	105	108	98	738	109	90	98	94	787	Crack
25	72	35	23	44	62	60	77	52	75	Spot
126	94	103	88	769	119	109	88	103	733	Crack
126	105	103	94	663	104	97	82	101	832	Crack
79	60	49	35	618	62	79	35	29	328	Spot
128	84	107	93	676	81	103	91	98	726	Crack
39	63	29	49	171	55	46	58	73	498	Spot
67	74	61	45	559	70	42	45	42	470	Spot
139	102	96	108	772	104	98	92	110	869	Crack
168	97	96	102	795	109	161	132	104	821	Good
185	96	110	95	749	110	168	134	106	750	Good
194	97	117	119	757	100	170	123	104	781	Good
51	27	41	78	320	65	67	42	40	304	Spot
113	96	106	101	698	110	96	97	105	851	Crack
119	93	105	95	685	87	124	101	91	724	Crack
134	98	110	101	682	113	129	86	110	712	Crack
66	14	81	31	77	49	56	60	31	307	Spot
52	57	64	71	400	46	79	23	67	324	Spot
179	95	106	96	802	105	172	133	100	748	Good
128	91	101	95	679	83	119	86	98	822	Crack
36	19	47	36	260	31	54	44	25	68	Spot
197	102	114	112	745	108	168	127	117	739	Good
190	103	112	117	779	109	158	131	96	754	Good
121	94	101	103	694	112	116	103	99	853	Crack
171	103	101	110	778	110	172	117	106	823	Good
138	99	92	103	747	94	106	101	89	792	Crack
193	112	112	90	760	111	171	127	115	768	Good
121	110	86	91	759	100	124	84	92	728	Crack
75	40	75	65	424	69	70	41	43	529	Spot
135	91	81	102	662	106	88	97	109	758	Crack
131	96	85	94	722	80	120	88	97	765	Crack
162	102	116	118	770	107	173	125	116	805	Good
58	33	26	77	293	66	50	43	82	65	Spot
43	74	54	35	211	38	76	60	64	179	Spot
45	63	43	40	339	34	46	75	63	435	Spot

Table 5-10 Recognized unseen examples

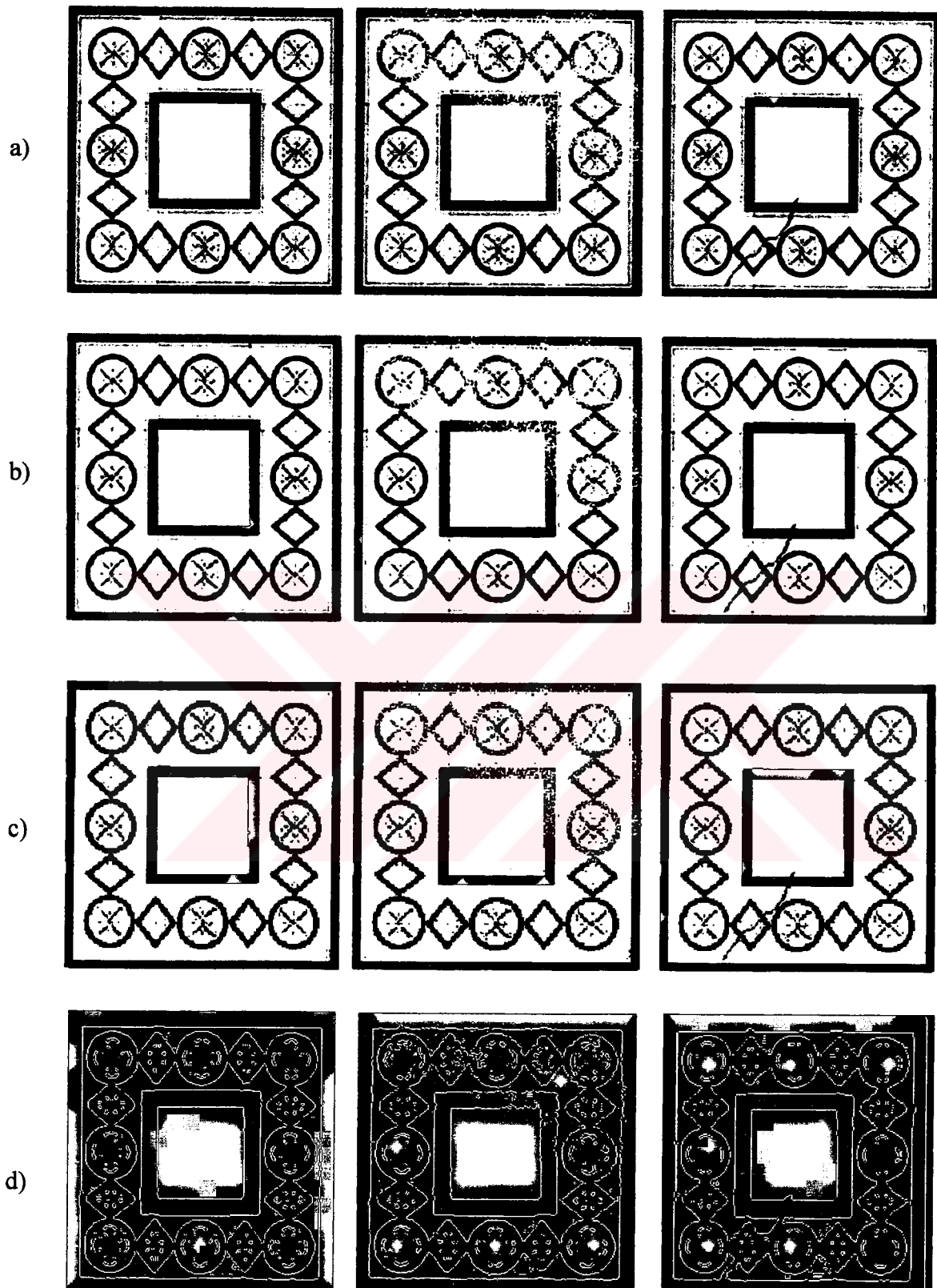


Figure 5-12 a) Original pictures b) gray level c) binary image d) edges of image

6 CONCLUSION

Due to conditions of the production environment, some cracks and spots may exist on the ceramic tiles. The main objective of inspection is to identify the defective tiles. Inspection process of ceramic tiles is currently carried out by workers. It takes too much time and the results are not satisfactory. It affects the performance of quality control process. It also may cause the defective products reaching the customer.

In this thesis an alternative technique for ceramic tile inspection was studied. More than 10 different ceramic tiles were used for training and/or recognition. First, image processing steps (gray level, binary image, thresholding and edge detection) are applied to tiles to determine the tiles' classes. In order to determine the tiles' classes ten(10), 3x3 masks were used. The use of frequencies of masks represents the class of the tile. For training and/or recognition RULES-3 inductive learning algorithm was employed as the pattern classifier. The frequencies of the masks were used as the set of examples to extract the required rules. Over than 100 good and defective tiles were given to the system for training. The efficiency of training depended on how many examples were given to the system. After training process, unseen defective and good tiles were asked to the system. The results showed that this system worked with a very high performance. Instead of currently used inspection system, this proposed visual inspection system can be employed.

The advantages of the system can be outlined as following:

- The production time with the proposed system is much shorter than the currently used inspection system..
- It requires simple tools.
- It does not require expensive and complicated Hardware and Software.
- It increases the inspection efficiency.
- It decreases the production cost.

- Production conditions are not suitable for human health. In the new system since the inspection process is operated by cameras and computers instead of workers there will not be such health problems.

As a final conclusion, the system indicates that RULES-3 inductive learning algorithm can achieve the identification of the defective tile automatically.

The future study will be based on improving the proposed system for tiles with defects, which are too difficult to inspect.



7 APPENDIX

The programs and codes used in this thesis are given below. Codes must be run in MATLAB Program.

7.1 Files In the Disk

rgb2gray.m: For turning gray level.

imadjust.m: For increasing level of contrast.

filter2.m: For applying linear filtering

im2bw.m: For converting binary image using thresholding value

edge.m: For finding edges operator such as sobel, roberts, laplacian and gaussian and so on.

cercsmall.m: For cutting pictures edges.

fmaskall.m: Finds frequencies of defined ten masks.

main.m: Main program. This program contains as follows;

Run the rgb2gray.m program

Run the imadjust.m program

Run the filter2.m program

Run the im2bw.m program

Run the edge.m program

Run the cercsmall.m program

Run the fmaskall.m

Write the frequencies of the mask on the tile.inp file to be recognized

tile.exm:Inludes set of examples to extracted rules

tile.atr:Inludes attributes values.

tile.cls: Inludes class values.

7.1.1 Main.m (Main Program)

```

name=input('Image Name.....','s');

uz='.inp';           // extension of input file

location='C:\rules3\';

isim=sprintf('%s%s',name,uz);

isim=sprintf('%s%s',location,isim);

fid = fopen(isim,'w'); // open input file

a=imread(name);      // read the image name that exists in the computer.

b=rgb2gray(a);       // contert it gray level

c=imadjust(b, [0.3 0.5]); // performing contrast.

h=fspecial('average',3); // linear filter mask

d=uint8(round(filter2(h,c))); // applying linear filter

f=im2bw(d,9);        // converting binary image

g=edge(f,'sobel');   // applying sobel operator

cercsmall;           // cutting edges of image

fmask;               // finding frequncy defined 10 masks.

mask=[cm01 cm02 cm03 cm04 cm05 cm06 cm07 cm08 cm09 cm10];

```

```
fprintf(fid,'%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d',mask); // writing mask values
on the input file
```

```
fprintf(fid,'%s\n',ad); // writing tiles' name on the input file
```

```
fclose(fid); // closing input file.
```

7.1.2 Fmask.m (Finding Mask Frequency)

```
m01=[1 0 0;0 1 0;0 0 1]; // defining masks
m02=[0 1 0;0 1 0;0 0 1];
m03=[0 0 0;1 1 0;0 0 1];
m04=[0 0 0;1 1 1;0 0 0];
m05=[0 1 0;0 1 0;1 0 0];
m06=[0 0 1;0 1 0;1 0 0];
m07=[0 0 0;0 1 1;1 0 0];
m08=[1 0 0;0 1 0;0 1 0];
m09=[0 1 0;0 1 0;0 1 0];
m10=[0 0 1;0 1 0;0 1 0];
cm01=0;cm02=0;cm03=0;cm04=0;cm05=0;
cm06=0;;cm07=0;cm08=0;cm09=0;cm10=0; // making equal zero
[q v]=size(g); // size of edges images
[p r]=find(g); // finding black pixels
w=[p r];
[u t]=size(w);
```

```
for m=1:u-1

z=w(m,:);

s=(g(z(1)-1:z(1)+1,z(2)-1:z(2)+1));           // finding 3x3 neighbours

if s==m01                                       // comparing defined masks

cm01=cm01+1;

    elseif s==m02

cm02=cm02+1;

    elseif s==m03

cm03=cm03+1;

    elseif s==m04

cm04=cm04+1;

    elseif s==m05

cm05=cm05+1;

    elseif s==m06

cm06=cm06+1;

    elseif s==m07

cm07=cm07+1;

    elseif s==m08

cm08=cm08+1;

    elseif s==m09

cm09=cm09+1;
```

```

elseif s==m10

    cm10=cm10+1;

end

end.

```

7.1.3 Cercsmall.m (Cutting Edges of Image)

```

[x y]=size(g);

for i=1:x-2

    for j=1:5

        k=g(i+1,j+1);

        if k==1

            g(i+1,j+1)=0;

        end

    end

end

end

end

```

```

for j=1:y-2

    for i=x-7:x-2

        k=g(i+1,j+1);

        if k==1

            g(i+1,j+1)=0;

        end

    end

```

```
end

end

for i=1:x-2

for j=y-7:y-2

k=g(i+1,j+1);

    if k==1

        g(i+1,j+1)=0;

    end

end

end

end

for j=1:y-2

for i=1:5

k=g(i+1,j+1);

    if k==1

        g(i+1,j+1)=0;

    end

end

end

end
```

7.2 Hardware Requirements

Minimum Hardware requirement,

486 Processor,

2 MB Graphics Card;

150 MB disk capacity for installing MATLAB 5.0 Software Program;

32 MB RAM capacities;

7.3 Software Requirements

The operating systems

Win95, Win 98, Win NT, Win 2000

Software

MATLAB 5.0 or MATLAB 5.2

REFERENCES

-
- 1 Aksoy M.S, *New algorithm for machine learning*, Ph.D. Thesis, University of Wales College of Cardiff, U.K., 1993
 - 2 T.M. Mitchell, *Machine learning*, Mc Graw-Hill, New York, 1997
 - 3 M.Levine. *Vision in Man and Machine*. McGraw-hill, New York, 1985.
 - 4 D. Vernon, *Machine vision - Automated visual inspection and robot vision*, Prentice Hall, New York, 1991.
 - 5 Y. Lu and R. C. Jain. *Behavior of edges in scala space*. IEEE Trans. Pattern Analysis and Intelligence, 11(3): 337-356, 1989.
 - 6 K. Kanatani and T. C. Chou. *Shape from texture: General principle*. Artificial intelligence, 38(1):1-48,1989.
 - 7 G.S. Sebestyen, *Decision-Making Processes in Pattern Recognition*, Macmillan, New York, 1962.
 - 8 B.K. P. Horn. *Robot Vision*. McGraw-Hill, New York, 1986.
 - 9 J.C. Simon, *From pixels to features*, North Holland, Amsterdam, 1989.
 - 10 N.J. Nilsson, "A Mobile automation: An Application of Artificial Intelligence Techniques," Proc. 1st Int'l Joint Conf. Artificial Intelligence, p.509-520, USA, May 1969
 - 11 Sanderson, J. Ronald, "Machine Vision Systems: A Summary and Forecast, Illinois, USA, 1983
 - 12 Braggins D., Hollingum J., "The Machine Vision Source Book", New York, Tokyo, 1986.
 - 13 J.C. Russ, *The Image Processing Handbook*, CRC Press, Inc., Boca Raton, Ann Arbor, London, Tokyo, 1992.
 - 14 <http://ud.udri.udayton.edu/ece563/introduction>
 - 15 L.J. Galbiati, Jr, *Machine vision and digital image processing fundamentals*, Prentice-Hall International, Inc, Englewood Cliffs, 1990.

-
- 16 Gonzales,R. C., Woods R.E. *Digital Image processing New York: Addison-Wesley, 1992*
- 17 Newton I., *New Theory about Light and Colors* Reprint Munchen: Werner-Fritsch 1965
- 18 Zeki S., *A Vision of the Brain*, London: Blackwell Science Ltd. 1993
- 19 Kauppinen H., Silven O. “ *Color Vision Approach for Grading Lumber*” Theory and Applications of Image Processing II, World Scientific, 1995
- 20 Atalay, A “ *Automated Defect Inspection of Textile Fabrics Using Machine Vision Techniques* “ M.S. Thesis, Bogazici University, 1995
- 21 Rafael C. Gonzalez and Richard E. Woods, *Digital Imaging Processing*, Addison-Wesley, Reading, Massachusetts, USA, 1992
- 22 V.S. Nalwa, *A guided tour of computer vision*, Addison-Wesley, 1993
- 23 Jan Teubner, *Digital Image Processing*, Prentice Hall, Copenhagen, 1989.
- 24 .K. Pratt, *Digital Image Processing (second edition)*, John Wiley & Sons, New York, 1991.
- 25 Devijver, P. A. and Kittler, J. V. *Pattern Recognition. A Statistical Approach*. Prentice Hall, Englewood Cliffs. 1982
- 26 Y.H. Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison Wesley, Reading, Massachusetts, 1989.
- 27 C.M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.
- 28 Widrow B, Winter G., Baxter R. A. “ *Layered Neural Nets for Pattern Recognition*” IEEE Trasn on ASSp vol. Assp, vol. Assp-36 no.7 pp.1109-1118, july 1998
- 29 S.M. Weiss and C.A. Kulikowski, *Computer Systems that Learn*, Morgan Kaufmann, San Mateo, California, 1991.
- 30 Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J., *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, Ca,1984.

-
- 31 Nakakuki Y., Koseki Y., Tanaka M., “*Inductive Learning in Probabilistic Domain*”, in proc. Eight National Conf. Boston, 1990
- 32 Forsyth R., “*Machine Learning Principles and techniques*”, London, 1989
- 33 Charniak E., McDermott D., “*Introduction the Artificial Intelligence*”, Colifornia, 1985
- 34 Kodratoff Y., “*Introduction the Machine Learning*”, London, 1988
- 35 Quinlan, J. R., “*Induction, Knowledge and Expert System*”, in Artificial Intelligence Developments and Applications, North Holland, 1988.
- 36 Al-Attar A., “*Rule Induction From Mythology to Methodology*”, London, 1991



REFERENCES NOT CITED

1. K.Y. Song, M. Petrou, and J. Kittler. *Defect Detection in Random Color Textures. Image and Vision Computing*. 1996
 2. J. Song, M. Petrou, and J. Kittler, R. Marik, M. Mirmehmedi. *Detection of defects in color texture surface*. IAPR Proc. Of Machine Vision Applications 94, pages 558-567, December 1994
 3. C. Boukouvalas, J. Kittler, R. Marik and Mpetrou. *Automated grading of ceramic tiles using machine vision*. In IEEE Int. Symposium on Industrial Electronics, pages 13-84, 1994
 4. R. Fioravanti, S. Fioravanti, D. Giusto, and Vernazza G. *Real-time Texture defect detection. In proceedings of Phare-accord Workshop on Efficient Texture analysis, Volume I, pages 57-66, 1994*
 5. Hawkins, J. K., " *Textural Properties for Pattern Recognition*" Picture Processing and Psychopictorics, B. Lipkin and A. Rosenfeld (editors), Academic Press, New York, 1969
 6. Castleman, K.R., *Digital Image Processing*. Prentice-Hall, Englewood Cliffs, NJ. 1983
 7. I. Pitas, *Digital image processing algorithms*, Prentice-Hall, Englewood Cliffs, 1993.
 8. Jain R., Kasturi R., Schunck G.B., *Machine Vision*, Singapore, 1995
 9. T.Y. Young and K.S. Fu, *Handbook of Pattern Recognition and Image Processing*, Academic Press, Orlando, Florida, 1986.
 10. Davies E.R., *Machine Vision: Theory, Algorithm, Practicalities*, USA, 1997
 11. T.M. Mitchell, *Machine learning*, Mc Graw-Hill, New York, 1997.
 12. L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and regression trees*, Wadsworth, California, 1984.
 13. Dubes, R. and Jain, A. K. *Clustering techniques: The user's dilemma*. Pattern Recognition, 8:247—260, 1976.
-

-
14. <http://faith.swan.ac.uk/~eederavi/sip/node4.html>
 15. Bledsoe, W. W. and Browning, I. *Pattern recognition and reading by machine*. In Proceedings of the Eastern Joint Computer Conference, pages 232--255, Boston. 1959.
 16. J. Sklanski and G.N. Wassel, *Pattern Classifiers and Trainable Machines*, Springer, New York, 1981.
 17. K. Fukunaga, *Introduction to Statistical Pattern Recognition (Second Edition)*, Academic Press, New York, 1990.
 18. B. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, 199
 19. Gagolowicz A., "Some application to the study of human vision" IEEE Trans on PAMI vol Pami-3 No.5 pp 520-533 September 1983.
 20. R.O. Duda and P.E. Hart, *Pattern classification and scene analysis*, John Wiley & Sons, New York, 1973.
 21. Keller J. M., Gray, M. R., Givens J. A. "A fuzzy K-nearest Neighbor Algorithm" IEEE Trans On SMC Vol Smc-15 no: 4 pp 580-585, October 1985