CLASSIFICATION OF TEXTILE IMAGES

by

Rıfat AŞLIYAN

April, 2002

İZMİR

CLASSIFICATION OF TEXTILE IMAGES

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of
Dokuz Eylül University
In Partial Fulfillment of the Requirements for
the Degree of Master of Science in Computer Engineering

by
Rıfat AŞLIYAN

April, 2002
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We certify that we have read the thesis, entitled **"Classification of Textile Images"** completed by **Rıfat AŞLIYAN** under supervision of **Asst. Prof. Dr. Adil ALPKOÇAK** and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Adil ALPKOÇAK

Supervisor

Prof. Dr. Tatyana Yakhno

Committee Member

Y. Doç. Dr. Reşat YILMAZ

Committee Member

Approved by the
Graduate School of Natural and Applied Sciences

Prof.Dr. Cahit HELVACI
Director

ACKNOWLEDGMENTS

I would like to express my deepest gratitude towards my supervisor, Adil ALPKOÇAK, who has patiently guided me through every stages of this thesis. I am truely grateful for his continuous source of ideas, his friendly support and help on editing this thesis.

I am also grateful to my supervisor and Erbuğ ÇELEBİ for obtaining textile images. I would like to thank Şakir DOLGUN, Ferkan KAPLANSEREN, and Korhan GÜNEL for their help on this thesis.

Finally, I thank to my parents, my sisters and my brother for their great support, endless encouragement and efforts to finish my study in the university.

Rıfat AŞLIYAN

To my parents, Fatma and Mehmet AŞLIYAN.

ABSTRACT

In this study, textile images are classified into striped, plaided and speckled image classes. Striped images are also categorized into the classes of horizontal, vertical, 45 and 135 degree lines. Line and circle features are used for the classification. To detect line and circle features, it is applied to line detection operator and Hough transform method respectively. With the line feature, striped and plaided images are classified, but speckled images are classified with circle feature. For the classification of the textile images, it is used some classification methods as Manual Thresholding, Nearest Neighbor, *k*-Nearest Neighbor, Minimum Distance. After the evaluation of the system, we have found that the best methods to classify striped images are "Nearest Neighbor with Manual Thresholding" and "*k*-Nearest Neighbor with Manual Thresholding and *k*=5, *k*=7, *k*=11". The best methods to classify plaided images are "Nearest Neighbor" and "Nearest Neighbor with Manual Thresholding". The best method to classify speckled images is "Manual Thresholding with Line and Circle Threshold > 0". The results of the classification are quite successful.

ÖZET

Bu tezde, tekstil resimleri çizgili, ekoseli ve puantiyeli resim kategorilerine sınıflandırılmıştır. Çizgili resimler de yatay, dikey, 45 dereceli ve 135 dereceli çizgili resimler gruplarına sınıflandırılmıştır. Bu sınıflandırmada, doğru ve çember özellikleri kullanılmıştır. Doğru özelliklerini tespit etmek için "Line Detection" operatörü, çember özelliğini tespit etmek için "Hough Transform" metodu uygulanmıştır. Çizgi özelliğiyle çizgili ve ekoseli resimler, çember özelliğiyle puantiyeli resimler sınıflandırıldı. Resimlerin sınıflandırılmasında "Manual Thresholding", "Nearest Neighbor", "*k*-Nearest Neighbor", "Minimum Distance" ve "Manual Thresholding" ile yukardaki diğer sınıflandırma metotlarının birleştirilmesiyle oluşturulan sınıflandırma metotları kullanıldı. Sınıflandırmayı yaptıktan sonra sistemi değerlendirdik ve oldukça iyi sonuçlar elde ettik. Çizgili resimlerin sınıflandırılmasında "Nearest Neighbor with Manual Thresholding" ve "k-Nearest Neighbor with Manual Thresholding and k=5, k=7, k=11" en iyi metotlardır. Ekoseli resimlerin sınıflandırılmasında "Nearest Neighbor" ve "Nearest Neighbor with Manual Thresholding" en iyi sonuç aldığımız metotlardır. Puantiyeli resimlerin sınıflandırılmasında ise "Manual Thresholding with Line and Circle Threshold > 0" metodu en başarılı olan metot oldu.

CONTENTS

Chapter Three
FEATURE EXTRACTION

Chapter Four
CLASSIFICATION METHODS

Chapter Five
IMPLEMENTATION

Chapter Six
EXPERIMENTS OF THE SYSTEM

Chapter Seven
CONCLUSION

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

The information, which is in the form of image, video, text, or audio, can be reached by a number of people at any time. In most cases, the information is accessible in a computer readable form. There are huge images available on-line, on the contrary the number of people who process them is limited. The task included with sorting, indexing and classifying images by human operators are very expensive since the number of images are increasing rapidly. Searching a huge database of unlabeled images for an image containing a certain pattern is a very difficult task to automate, and the number of research groups around the world and private companies investing this kind of efforts in research topics are mounting.

In this thesis, it is planned to classify textile images into striped, plaided and speckled image classes. In other words, the aim of this study is, for a new image that is not included in the image database, to decide whether the image is in striped, plaided or speckled class. For making the purpose a reality, we have used approximately 3500 textile color images with different size. All of the images in the database are in jpeg format. To classify these images into the categories, it is obvious that color feature is not critical. To find striped and plaided images, line features are the best choice and to find speckled images, the best choice is circle features. Therefore, for the feature extraction we used line detection operator and Hough transform. After constructing the line and circle features of a chosen image, by using some image classification methods as Manual Thresholding, Nearest Neighbor, *k*-Nearest Neighbor, Minimum Distance, the image is classified into the most similar image class.

The program has been written with the programming language "Delphi 3.0" on the Windows 95 operating system. It can be executed on the operating systems as Windows 95, Windows 98, Windows Me, Windows NT, Windows 2000 and Windows XP.

To date, in the area of image classification many projects are built up. In general, these projects are related to the technologies of computer vision. Some systems are developed to diagnose patients such as tumor recognition. In military area, some computer vision software is developed to scan environment, to find the position of the weapons. In addition, autopilot automobile and helper systems of drivers are in the stage of experiment whether or not it works well. All of these studies are related to image classification since these systems are based on calculation of statistical features of images. There are also some researches about indoor-outdoor, photographical-synthetical, portraits-non-portraits, face, blue sky, sea, and animal images classification.

This study is quite beneficial to textile engineers, designers and customers. Designers will access the images very fast, and are not bored and very productive. The production of the images that is exactly similar each other can be prevented by easily accessing of designers to the images. Sellers and customers will also profit from image classification at shopping. For example, the customers who want to buy a speckled cloth do not have to search the entire image database. It will be enough to search only in the speckled image class.

In this thesis, a new approach is developed to determine striped and plaided images. The number of lines and its direction are computed by using line detection operator. This features shows quite successful results to detect striped and plaided images. And furthermore this approach is quite fast at line feature extraction.

1.1 Automated image classification

Image classification process is to categorize all of the digital images into the most suitable classes using some features as color, texture, and shape. So, before the classification of images, it is necessary to extract the features of images. In this study, shape features are used since shape features are the best shoice for textile image classification. More detailed information about image features is explained in Chapter 3.

Figure 1.1 shows general image classification procedure.

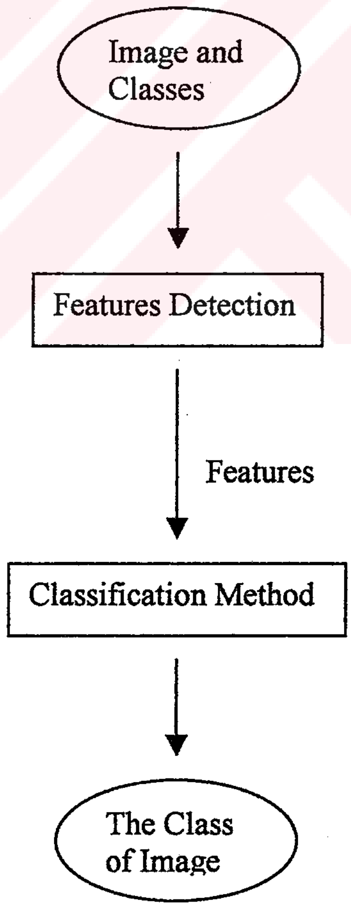


Figure 1.1 The general structure of image classification

Traditionally automated image classification has been divided into supervised and unsupervised classification. The main difference between these two classifications is the amount of information the operator has about the images. In supervised approach, there is some training data. That means it is known what the features on the image represent. But, unsupervised approach is used to classify unknown pixels and place them into classes based on natural groupings. And it can be said that unsupervised classification does not include training data and features about the image.

After choosing and extracting the images' features, it is necessary that one or more image classification method must be selected for classification of the images. As it is mentioned above, image classification is divided in two main categories: Supervised and Unsupervised classification. Some methods such as Nearest Neighbor, *k*-Nearest Neighbor, Minimum Distance, Maximum Likelihood, Parallel Piped methods are in Supervised classification. However the others such as K-Means Classifier, Fuzzy K-Means and ISODATA classifier are in Unsupervised classification. Nearest Neighbor, *k*-Nearest Neighbor and Minimum Distance, which are used in this thesis, are explained in Chapter 4 in detail.

1.2 Thesis organization

This thesis consists of seven chapters. In chapter 2, some important studies about image classification are summarized. In chapter 3, line and circle feature extraction methods are explained. In chapter 4, we described the image classification methods used in the study. In chapter 5, the implementation of the program and the algorithms are discussed in detail. In chapter 6, the experimental results of the classifiers used in this study are presented. We drew the conclusions and presented the future work in chapter 7.

CHAPTER TWO
A SURVEY OF IMAGE CLASSIFICATION

In this chapter, nine important image classification studies, which has been made in recent years, are summarized to be acquired a general idea of image classification. Each study is explained in different subsections as follows.

2.1 Image Sorting-Image Classification: A Global Approach

The study of Clavier et al. focused on the structure of the classifier called the rejection unit and particularly on the methods used to determine the rejection threshold. (Clavier et al., 1998)

Sorting system must respect different constraints. First, the sorting unit can model only a small set of forms among the whole set of possible documents. Moreover, only a small set of examples may be available when the system models a new form. Document modelization must be simple. Then the system has to perform the followings: Document modelization with few examples, efficient reject of unknown patterns, and automatic determination of the different parameters.

Sorting unit can be thought as a classifier. Labeling task performance will be valued with a set of known classes. An efficient labeling module must have a confusion rate as low as possible. The efficiency of rejection mechanism can be evaluated with two criteria: rejection rate and recall rate. Sorting unit was developed via based on primitives. These primitives are densities of black pixels extracted by cutting up the image successively in 64 and 256 regular blocks. The labeling unit proposes a label class, which is not confirmed by the rejection unit. An overlap threshold is defined in order to get margins during the validity step. A low threshold entails a decrease of the rejection rate.

Sorting unit is characterized by the determination of templates representing the different classes. After an overlap rate is defined, overlap margins can be calculated. To set the threshold value, it can be checked different methods. The best reject results can be taken by methods, which considers the extreme values of a known class as unknown patterns.

2.2 Integrating Image Matching and Classification for Multimedia Retrieval on the Web

Classification of images provides users to search large image data more easily and efficiently. Using classification eliminates unrelated images and reduces the search space. A method for classification of image based on shape, color and composition using the primary objects is presented in the study of Hirata et al. (Hirata et al.,1998) After the experiment, this approach can maintain 73 % of recall by searching 24 % of all data set. In addition, the classification results are used to help the user's interaction.

Because of a number of images, it is difficult to search for images of interest. Many search engines' effectiveness is prevented by these two reasons: Too many images on the web to be searched in a short time and a large number of matching images due to different visual factors such as color, shape, etc.

An effective classification model, which is called primary objects, is suggested in this study. These primary objects are a group of atomic units for image objects. These objects describe the contents of image objects. All the images are categorized in the primary objects according to the similarity of the primary objects, and they are represented as primary object groups.

The choice of primary objects is application-specific. In the study, three types of visual primary objects are extracted. First, primary objects for major color of the objects contained in the image. Second, primary objects for the object shape contained in the image. Third, primary objects for object composition considering the object location in addition to shape. This system firstly extracts the feature of the image and then decides the similarity to the primary objects. The images containing a similar object as for shape and color are placed into the same category.

According to this classification method, this system constructs a new query interface. It can be chosen by users a set of primary objects and their importance and the candidates from the category are retrieved. These retrieved results are classified into primary objects and presented to the users in the hierarchical way.

To conclude, an image retrieval approach integrating image matching and classification techniques is presented in the study, and images based on shape, color and composition with the primary objects that can be thought as visual representation are classified automatically. This technique is quite efficient to access the images. The search space is reduced by 76 %.

2.3 A Classification Method of Images Based on Composition and its Application to Image Retrieval

The situation of visual query techniques that is used with concrete images as images is pointed out in the study of Maeda et al. (Maeda et al., 1998) This comes from the communication between the user and the system. There exists the ambiguity at the system interpretation of queries. For overcoming these tradeoffs, some methods in prototype system are developed. From image database, some composition types are derived to show the condition of the query methods working in the system.

The tools for image retrieval are absolutely necessary because of efficiency of retrieval system. In the early retrieval systems, there exist some problems such that users cannot find suitable visual features of the target images. It uses only linguistic query methods. In these times, some keywords or text that are tied with the images are used. To cope with this problem, many innovations have been introduced. Drawings, sketches or similar images as visual query methods are applied for retrieval systems. These techniques make possible the user to express enough visual properties of the target images. But these query methods don't ensure. To cope with this tradeoff between expressiveness and definiteness of queries, some visual query methods in terms of image communication between the system and its user can be thought. Firstly what the user and the system do in image retrieval can be examined, secondly the problem in communication of visual properties between them is pointed out. As a solution to the problem, a query method is proposed, in which the user selects 'composition type' of the target image from the listing of typical composition patterns derived from the image database in advance, and implemented the query method in an image retrieval system and showed how our query method works in image retrieval. To use an image retrieval system, it is necessary for the user to communicate properties of the target image with the system. Thus, image retrieval must be considered as image communication between the user and the system.

The image communication's difficulty originates the ambiguity of image media. In other words, various image properties at various abstract levels are derived from the same pattern of an image. So, image properties are resolved into image factors to simplify and clarify the image communication. Each image factor, it is named as IF, indicates an image property and designates the set of images containing the property. A concrete image of a clock can be resolved into several IFs: two straight lines, a circle, and their locations. Each straight line can also be resolved into more abstract IFs such as the class of line, the size and the shape. In addition, the clock image also includes the abstract IF, the class of clock. Thus, all image properties including images themselves can be resolved into definite IFs. IFs can be classified into two classes based on whether or not they correspond to the patterns on concrete images directly. The word 'clock' is not directly connected with specific image patterns while the other IFs are closely related to.

As a result, the problem in image retrieval by visual contents from the viewpoint of image communication between the retrieval system and its user is explained. And a visual query method is proposed for composition as a solution, in which the user can depict the composition of the target image by selecting one of the composition types presented by the system. In addition, it is proposed as an image classification method based on composition and a visualization method of composition types derived by the classification. The methods are implemented in a prototype image retrieval system and an experiment was performed with the system to derive composition types from the image database. The results of the experiment show that the methods are appropriate for obtaining composition types. The query method for retrieval by composition must be used together with other query methods suited to convey other image properties than composition.

2.4 Texture Image Retrieval by Universal Classification for Wavelet Transform Coefficients

A method that used texture image database retrieval is developed in the study of Yue & Guo. (Yue & Guo, 1997) According to the classification, the retrieval system used the universal classification theory. Firstly, the wavelet transform is taken for input images and then wavelet transform coefficients in each sub band are processed to take the type based discrimination measure between these images. The type based empirical sequence classification rule is asymptotically optimal. After the simulations, the type based retrieval scheme is capable of yielding very good texture retrieval performance. If it is compared with the conventional sub band energy based distance measure for wavelets coefficients of images, this method yields on average approximately 6 % to 8 % higher texture retrieval rate for an image database of 97 textures. Texture image database retrieval is very good applications in situations such as querying medical image database or getting image-related information through Internet. Because traditional text-based indexing schemes are hardly sufficient for large image databases, the more exible and detective means of retrieval using similarity measures of images become more important. A similarity measure is just as important in the image retrieval task as in image classification, where the test image must be compared with the training images. For closeness of two images a superior quantitative measure is vital to succeed good image classification and retrieval performance. Texture analysis algorithms using wavelet transform have been shown to perform very well and a detailed performance comparison of different kinds of wavelet transforms in texture based pattern retrieval was presented in. Using wavelet transform coefficients of each sub band to characterize the spatial-frequency contents of images, by taking the distance measure of two sequences of coefficients it can be measured the dissimilarity between two images.

The distance measure that is referred as the type-based statistic is developed. After training sequences from different classes of unknown models, universal classification has been shown to be asymptotically optimal in test sequence classification. Because it is not possible to have access to the real models of the wavelet transform coefficients of images, this problem is treated as an empirical sequence classification problem and, to obtain the distance, applied the type based detection theory. The universal classification theory shows that no other classification rules based on the same wavelet coefficients can perform better asymptotically. It is also shown that the type-based measure by image retrieval from an image database of 97 textures is very effective. The type-based statistic is compared with the conventional sub band energy-based distance measure.

In conclusion, a method for texture image database retrieval is used by Yue & Guo. The method uses the type-based statistic from information theory as the distance measure. The distance of the quantized wavelet transform coefficients of image sub bands is taken as the distance of the images. After the simulations, it is shown that the new distance measure is very effective. It is also worthwhile to point out that the proposed distance measure operating on wavelet coefficients has great importance for the image classification system.

2.5 Classification of Images on Internet by Visual and Textual Information

Gevers et al. have proposed (Gevers et al., 2000) the computational models and techniques to add textual and image features for images classification. An index structure is developed to index images on the basis of textual, pictorial and composite information as textual-pictorial information. The *k*-Nearest Neighbor classifier is used to organize images into semantically meaningful groups of images, which is downloaded from Internet. Firstly Internet images are classified into photographical and synthetical images. And after classifying images into photographical and synthetical images, photographical images are then classified into portraits and non-portraits. Then, synthetical images are classified into button and non-button images. Their study has been conducted on a large set of images evaluating the accuracy of combining textual and pictorial information for classification. From the results of the experiment, it is shown that for classifying images into photographic-synthetic classes, the contribution of image and textual features is equally important. Finally, high discriminative classification power is obtained based on composite information. After classifying images into portraits-non-portraits, it is clear that pictorial information is more important then textual information. This result comes from the inconsistent textual image descriptions, such as surnames. Namely, for classifying into portrait and non-portrait classes, only marginal improvement in performance is achieved by using composite information.

Recently, content-based image retrieval systems have been developed retrieving images on the basis of multiple image features such as color, shape and texture. Nowadays, a lot of systems are available for retrieving images from Internet on the basis of textual or visual information. And new research is directed towards the use of both textual and pictorial information to retrieve images from Internet. Most of the content-based search systems are based on the so-called query by example paradigm. The basic idea to image retrieval by image example is to extract characteristic features from images in the database which are stored and indexed. In this technique, it is done off-line. These features are typically derived from shape, texture and color feature. The on-line image retrieval process consists of a query example image from which image features are extracted. These image features are used to find the images in the database, which are most similar to the query image. Although significant results have been achieved, low-level image features, extracted from the images, are often too restricted to describe images on a conceptual or semantic level. This semantic gap is a well-known problem in content-based image retrieval. Thus, image classification has been proposed to group images in the database into semantically meaningful classes to enhance the performance of content-based retrieval system. The advantage of these classification schemes is that simple low-level image features can be used to express semantically meaningful classes. In this way, the gap is bridged between low-level image features and high-level concepts. Images classification can be based on unsupervised learning techniques such as clustering, Self-Organization Maps and Markov models. Further, supervised grouping can be applied. For instance, vacation images have been classified based on a Bayesian framework into city-landscape by supervised learning. Landscape images are further classified into sunset, forest, and mountain classes. But, very little attention has been paid on using both textual and pictorial information for classifying images on Internet.

Now the goal is to get to a framework allowing to classify images on internet by means of composite pictorial and textual information into semantically meaningful groups and to evaluate its added value, in other words, to what extent the use of composite information will increase the classification rate as opposed classification based only on visual or textual information. Thus, computational models and techniques are studied to combine textual and image features to classify images on Internet. A framework is presented to index images on textual, pictorial and composite information. The scheme makes use of weighted document terms and color invariant image features to obtain a high-dimensional similarity descriptor to be used as an index. The indexing scheme is used to study the added value of using both information sources. So, the classification scheme will be applied on image, text and composite features. To achieve this, web-robots down loaded over 100.000 images of the gif and jpeg formats. And then, with these 100.000 images, textual descriptions have been down loaded located near the images such as URL and html tags. These textual descriptions are parsed and stemmed yielding a robust set of discriminative words. Next, images and their textual attachments are represented in multidimensional feature space. The classification method is based on supervised learning based on the *k*-Nearest Neighbor classifier. Images are classified into photographical and synthetical images. After classifying images into photographical and synthetical images, photographical images are further classified into portraits. Finally synthetical images are classified into button or non-button images.

To conclude, a framework has been developed. It is for classifying images on the basis of textual, pictorial and composite features. The scheme makes use of weighted document terms and color invariant image features. The goal is to obtain a high-dimensional similarity descriptor to be used as an index. For the classification, the *k*-Nearest Neighbor classifier is used to organize images into semantically meaningful classes: photographical and synthetical images and portraits. The results of the experiment show that for classifying images into photographic-synthetic the contribution of image and text features is equally important. Therefore, high discriminative classification power is obtained based on composite information. Classifying images into portraits-non-portraits shows that pictorial information is more important then textual information.

2.6 Integrating Multiple Classifiers in Visual Object Detectors Learned from User Input

Jaimes & Chang have been studied on integrating multiple classifiers in visual object detectors. (Jaimes & Chang, 2000) Many efforts in content-based retrieval are performed for automatic classification of images-visual objects. Nowadays, most approaches have based on using individual classifiers. For Visual Object Detectors, the classifier is proposed as a hybrid classifier combination approach, in which decisions of individual classifiers are combined in the following three ways:

a) Classifier fusion

b) Classifier cooperation

c) Hierarchical combination

It is presented as the Visual Apprentice framework, in which a user defines visual object models via a multiple-level object-definition hierarchy such as region, perceptual-area, object part, and object. When the user provides examples from images, visual features are extracted and multiple classifiers are learned for each node of the hierarchy. It is also discussed the benefits of hybrid classifier combination in the Visual Apprentice framework, and showed some experimental results in classifier fusion. These results suggest some possible improvements in classification accuracy, particularly of detectors reported earlier for images with handshakes, baseball video and images with skies.

Some important studies are made in a few years and some new techniques have been proposed to automatically index visual content. It can be said that similarity or query-by sketch approaches is one of these techniques. For example, an image that resembles a drawing looks like another one. Some recent work has based on the automatic extraction of higher-level descriptions of the visual content, through classification. In this stage, the image or video is automatically placed into a semantic category. Examples of this approach contain the classification of images according to scenes. For example: indoor-outdoor, city, landscape, and objects such as naked people and horses. But, one of the problems with those approaches is that they are static. Namely, many of their components are built by hand and cannot be changed. For building general detection systems successfully, it is imperative to have dynamic frameworks in which different components adapt to the task at hand and where such components interact to exploit the structure of elements to be detected. The first step in that direction is to use machine-learning techniques to build classifiers that can automatically label content. The second step is to exploit the structure present in the objects to be detected, and to allow the interaction of different hypotheses during the detection process. Similarly, hypotheses are formed by means of classification. Generally an instance x as an image, video can be represented by a feature vector $v = f_1 ... fn$. The feature vector serves as input to a classifier function f(v), which outputs a label l that determines x's classification. Classification is often performed using an individual classifier: The decision regarding the class of x is made by a single expert as a classification function. But, it may helpful to decide on the class of x, based on a combination of distinct classification functions. This can cause better accuracy and more robustness. Then, the final decision relays on the outputs of the individual classifiers and on the strategy used to combine those outputs. In building detectors for objects or scenes though, there are many other possible ways to combine classification functions. Some of these strategies have their origin in the machine learning community, in statistics, and in studies of how the human visual system works. Multiple classifiers can be combined when building Visual Object Detectors. And a hybrid approach that combines several distinct classifiers using the framework of the Visual Apprentice is developed. It means that a user defines visual object models according to his interests via a multiple-level object-definition hierarchy such as region, perceptual-area, object part, and object. When the user provides examples from images or video, visual features are extracted and multiple classifiers are learned for each node of the hierarchy. In this new hybrid combination approach to detect visual objects, classifiers interact in different ways at different levels. Classifiers are combined in these three ways:

1. Classifier fusion: In this way, each classifier takes same input and their goal is to achieve a form of consensus.

2. Classifier cooperation: Each classifier is affected by other classifiers, and these classifiers use different inputs.

3. Hierarchical classification: An output of a classifier is the input of another classifier.

The approach of Jaimes & Chang can be separated from earlier studies in some ways: Automatic classification is performed using an individual classifier. The body plans approach performs classification based on a multiple stage process in which each decision is made by a single classifier function. In the indoor-outdoor approach, individual blocks are classified by different classifiers, whose outputs are combined through stacking. For earlier works, only hierarchical classification takes place.

In conclusion, some ways which many classifiers can be combined in when constructing object detectors is presented. In the study, a hybrid classifier combination approach is also explained in detail. In that approach, decisions of each classifier are combined in three ways: Hierarchical Classification, Classifier Cooperation, and Classifier Fusion.

2.7 A Conceptual Framework for Indexing Visual Information at Multiple Levels

A conceptual framework for indexing visual information is developed in the study of Jaimes & Chang. (Jaimes & Chang, 2000) This indexing is made in different ways. The developed framework combines many concepts of the literature, library sciences, cognitive psychology, art, and content-based retrieval. For visual and non-visual information, the multiple level structures are also developed. This multiple level visual structure gives a systematic way of indexing images. This image indexing focuses on syntax and semantics. Here, syntax represents color, texture, and semantics represents events, objects, and so forth. Different types of relation are given at different levels of the visual structure. A semantic information table is constructed. This provides a summarization of important aspects of an image. The purpose of the study is to unify the issues that should be considered when building a digital image library. This analysis stresses the limitations of state of the art content-based retrieval systems.

In recent years, there is a growth of digital images and video and this provides new opportunities to end-users because they can have a large amount of image and video resources. A lot of visual information can be acquired on diverse topics and from many different sources. However, users cannot review large quantities of data when searching such content. So, It is necessary to allow users to efficiently browse content and perform queries based on their specific needs.

So, it is very important to understand the data, and index it appropriately. This indexing must be structured and it must be based on how users will want to access such information. For some approaches, textual annotations are used for indexing as a cataloguer manually assigns a set of keywords or expressions to describe an image. Users can then perform text-based queries or browse through manually assigned categories. Apposite of text-based approaches, recent techniques in content-based retrieval have based on indexing images based on their visual content. Users can perform queries by example such as images that look like this one or user-sketch such as image that looks like this sketch. More recent efforts attempt automatic classification of images based on their content: a system classifies each image, and assigns it a label such as indoor, outdoor, contains a face. In both paradigms there are classification issues, which are often overlooked, particularly in the content-based retrieval community. There are some problems to index visual information: First, there is a large amount of information present in a single image and it can be said what to index. Second, different levels of description are possible and it can be said how to index. For example, think in mind a man wearing a suit in an image. You can label the image with the terms suit or man. The term man, in turn, could carry information at multiple levels: conceptual and visual. The study focuses on the problem of multiple levels of description for indexing visual information. The study presents a conceptual framework, which unifies concepts from the literature in diverse fields. It has also made some distinctions between visual and non-visual information and provided the appropriate structures. The ten-level visual structure presented provides a systematic way of indexing images based on syntax and semantics. Different types of relations such as syntactic, semantic relations are also defined. Using structures is beneficial in terms of understanding the users and their interests and in characterizing the content-based retrieval problem.

In conclusion, they have been developed a conceptual framework for indexing visual information at multiple levels. The structures that are presented are good for syntactic-semantic and perceptual-conceptual difference. It have been separated general concepts and visual concepts, and presented a structure as the semantic information table that means semantic information from visual and non-visual data. These structures provide indexing of visual information at multiple levels.

2.8 Indoor-Outdoor Image Classification

Szummer & Picard tried to classify indoor-outdoor images. (Szummer & Picard, 1998) The used features: (a) Histograms in the Ohta color space. (b) Multiresolution, simultaneous autoregressive model parameters. (c) Coefficients of a shift-invariant DCT. Each image is divided into sub block, and then these sub blocks are classified. 75-86 % performance is obtained with using single-feature methods. However, the new method provides 90.3 % correct classification.

On the scene classification, very little study has been done since this kind of studies is very hard.

On the image database, there are 1343 images and nineteen out of these are labeled as unclear. So they are removed from the database.

The images are collected and labeled by Kodak. This database consists of sea, snow, bright sun, sunset, night, and silhouette images. These images are labeled with aiding of two people by hand. 694 images are labeled as outdoor, and 630 images are labeled as indoor. 19 images are labeled as unambiguous. All of the images have landscape orientation.

In this classification study, three types of features are used: (a) Color feature. (b) Texture feature. (c) Frequency information. These features are calculated for 4x4 sub blocks of each image. The color feature is constructed by a 32 bins color histogram. This histogram is gotten from Ohta color space, which can be computed from RGB color space:

$$I_1 = R + G + B$$
$$I_2 = R - B$$
$$I_3 = R - 2G + B$$

The decorrelating of color channels is the main benefit of Ohta color space. That's why Ohta color space provides 73.2 % better performance than RGB space. The histogram intersection norm is used for the distance metric. It is defined as:

$$\text{Dist}(h_1, h_2) = \sum_{i=1}^{N} \left(h_{i,1} - \min\left(h_{i,1}, h_{i,2}\right)\right)$$

Texture features are calculated by using the multiresolution, simultaneous autoregressive model (MSAR) (Mao & Jain, 1992). This provides best texture features on the Brodatz album (Picard et al., 1993). For this feature the distance metric is Mahalanobis norm. The performance of the MSAR classification is 82.2 % and 77.7 % correct at half and quarter resolution respectively.

The frequency features are computed by using the 2D DFT and the 2D DCT. The distance metric is Mahalanobis norm for that. In this classification there are two approaches. The first approach is based on the features that are computed on the whole image. The second one is based on the features of 4x4 sub blocks of an image. The features of sub blocks are calculated separately. The second approach gives better result than the first one.

Table 2.1 Whole image classification results, using *k*-Nearest Neighbor. The best result in which row is marked with (*). (Szummer & Picard, 1998)

Feature	k=1	k=3	k=5	k=9	k=13
RGB Histogram	69.5	72.0	72.4	73.9	74.0 (*)
RGB Histogram Intersection	69.1	71.3	72.4	73.5 (*)	72.9
Ohta HistogramEuclidean	73.2	75.0	75.3	75.2	75.4 (*)
Ohta Histogram Intersection	74.2	74.0	75.1	75.3	75.6 (*)
MSAR Quarter Resolution	77.7	80.2	81.9	82.3	83.0 (*)
MSAR Half Resolution	82.2	85.2	84.9	86.2 (*)	86.1
DCT Half Resolution	80.4	81.0	81.3	81.3	81.9 (*)

There are three methods for unifying features: (a) Simple majority classifier selects the label for the image that has the biggest number of class label. (b) One-layer neural net. (c) A Mixture of Experts. The first method gives good results. But the other two methods provide slightly better results. The best classification results are generally obtained by combining color features with texture features.

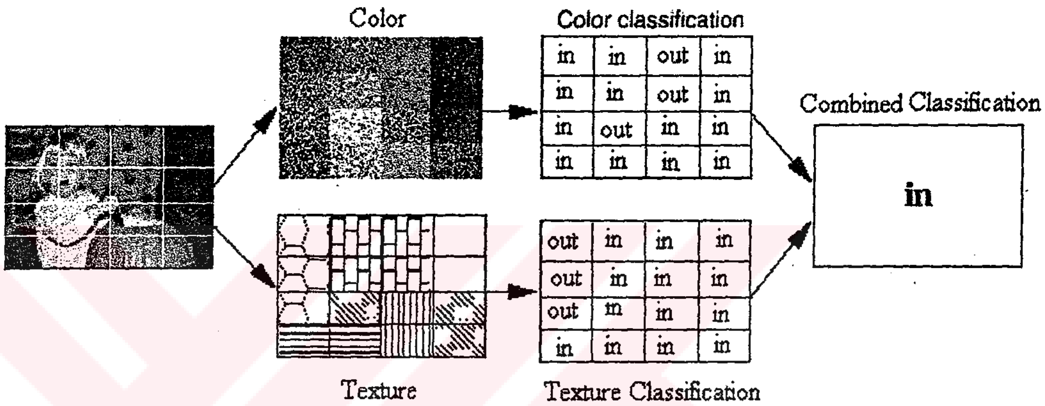


Figure 2.1 Two-stage classification combining color and texture (Szummer & Picard, 1998)

High-level image properties are extracted from low-level image features. This indoor-outdoor classification task is only one example of a high-level image property. It can be said that the performance of a feature or features' combination is very hard to determine; with *k*-Nearest Neighbor, combining two weaker features resulted in more robust result than a single good feature. Furthermore, a relative simple classifier (*k*-Nearest Neighbor) shows better than the more sophisticated neural network and "mixture of experts" classifier.

Table 2.2 Majority classifier based on *k*-Nearest Neighbor (Szummer & Picard, 1998)

Feature	k=1	k=3	k=5	k=9	k=13
Ohta Histogram Intersection	78.2	80.2	81.0 (*)	81.0 (*)	80.5
MSAR Hal Resolution	82.0	81.1	81.5 (*)	81.5	81.0
MSAR Quarter Resolution	80.0	83.0	81.9	82.6	84.0 (*)
DCT Quarter Resolution	82.0	81.1	85.0 (*)	81.5	81.0

2.9 Multi-stage Classification of Images from Features and Related Text

In the study of Smith & Chang, a new approach for classifying images is developed by image features and related texts. (Smith & Chang, 1997) For that classification, a multi-stage classification system is defined. This system's each stage restricts the class of every image. The classification system includes a hierarchy of classifiers, which are trained from the image on the World-Wide Web. The image's related text is used for training the classifiers.

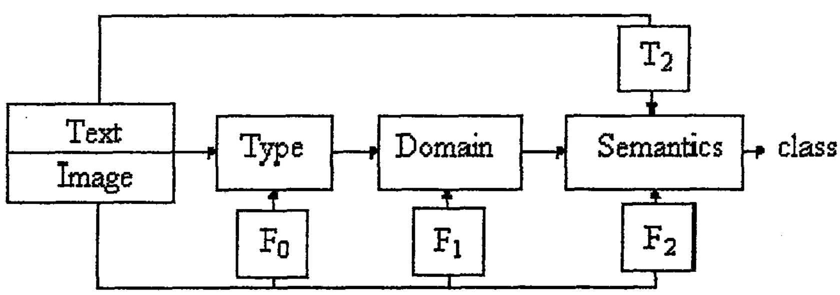


Figure 2.2 Multi-stage image classification system uses image feature sets: F_1, F_2, F_3 and related text T_2 (Smith & Chang, 1997)

The multi-stage classifier system shown in Figure 2.2 is divided into three stages. In the first stage, images are classified into image type classes. In the second stage, the images are classified into more restricted image composition classes. In the third stage, the images are classified into semantic classes such as buildings, nature, sunsets, beaches, and so forth by training of images using related text.

In the first stage, image type classes are used for classification system. This image type classes are color photographs, complex color graphics, simple color graphics, gray photos, gray graphics, b/w photographs, and b/w graphics. This system extracts the image features from HSV color space. The features are related the number of black, white, gray, colors which are fully saturated, colors which are half saturated, number of colors, number of grays, number of hues, and number of saturation present from the 166-color quantized HSV color space. After computing the image color features from training, the subset of features is constructed for a decision point. After generating a multi-dimensional space for each decision point, it is partitioned to training classes. The frequencies of training classes determine the decision criteria. With this process, an image can be categorized into the most similar class.

In the second classification stage, images are classified into silhouettes, center-surround images, scenes, and textures. The separation of the center and surround of image decides this image composition. For that, there are two methods: most prominent color and pooled color histogram. In the first method, the most prominent color is computed and back-projected onto the image to extract the surround region of the image. In the second method, after generating a pooled color histogram $hs = hA + hB + hC + hD$ from A, B, C, D in Figure 2.3, hs is back-projected onto the image. In this stage's features are size($C1$), size($C2$), distance($C1$,$S1$), distance($C2$,$S2$). After generating the decision space from training images the four-dimensional features space, the classification is made by extracting the center-surround features and finding the partition that corresponds to the feature values.

In the third stage, images are categorized into semantic classes as buildings, beaches, crabs, horses, divers, nature, tigers, sunsets.

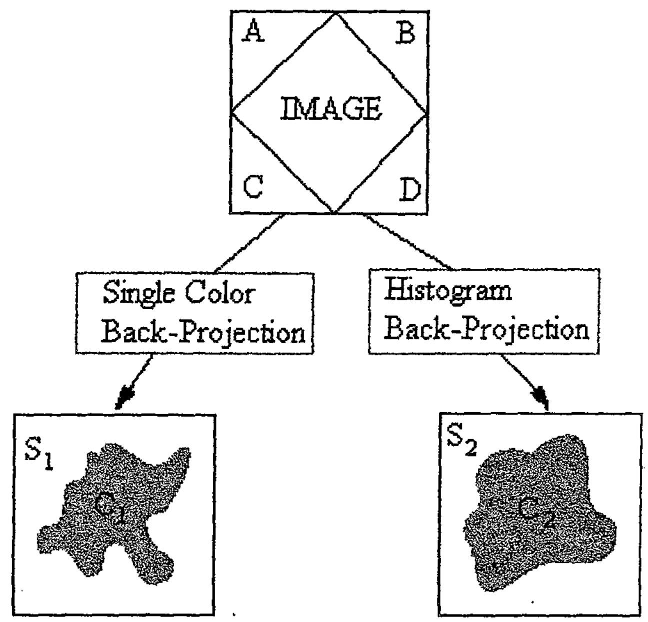


Figure 2.3 Image center-surround separation process for image composition classification extracts center regions and surround regions (Smith & Chang, 1997)

Composite region templates (CRTs) are extracted from training images of semantic classes. In this stage of the system, after extracting color regions of an image, for all semantic classes, region strings are constructed. Then, region strings are consolidated to the sets of CRTs for all classes. To generate the region strings is necessary to scan the image with five vertical scan from bottom to up. A symbol value such as 'A', 'B', and so fort is given to each scan and put the symbol on the image.

Definition 1. Region String: *A region string S is a series of symbols* $S = S_0S_1 \ldots S_{n-1}$, *which is separated from the regions of an image where* S_n *is the symbol value of the* n^{th} *successive region in a top-to-bottom scan.*

Definition 2. CRT: *A composite region template T is an ordering of M symbols,* $T = t_0t_1 \ldots t_{m-1}$. (Smith & Chang, 1997)

The region strings are consolidated with frequencies of CRT. For instance, assume that $T = t_0 t_1 t_2$ in the region string S, then

$$I(T,S)=\begin{Bmatrix} 1 \text{ if } S_1 = t_0 \text{ and } S_m = t_1 \text{ and } S_n = t_2 \; 1 \le m \le n \\ 0 \text{ otherwise} \end{Bmatrix}$$

The frequency of each CRT T_i of region strings set $\{ S_j \}$ is $P(T_i) = \sum_j I(T_i, S_j)$.

The frequency of each CRT T_i of region strings set $\{S_j\}_k$ is

$$P(T_i \mid C_k) = \sum_{\forall_i S_j \in C_k} I(T_i, S_j).$$

The CRTs of training images generate CRT library.

Definition 3. CRT Library: *A composite region template library is given by a set of (K+2)-tuples:* $\{T_i, P(T_i), P(T_i|C_0), \ldots, P(T_i|C_{K-1})\}$ *where K is the number of semantic classes.* (Smith & Chang, 1997)

To semantically classify an unknown image, decoding image semantics is used as follows:

1. Region strings are constructed and consolidated for the images.

2. For all unknown images $P(C_k \mid T_i) = \frac{P(T_i \mid C_k)}{P(T_i)} P(C_k)$ is computed.

3. $\forall i \neq k \; \sum_i P(C_k \mid T_i) > \sum_i P(C_l \mid T_i)$ then classify the image into classes.

Table 2.3 Image semantics classification experiment results using 71 training images and 190 test images (Smith & Chang, 1997)

	Overall	Beaches	Building	Crabs	Divers	Horses	Nature	Sunsets	Tigers
The number of Total	261	14	56	9	33	26	46	46	31
The number of Train	71	7	10	4	10	10	10	10	10
The number of Test	190	7	46	5	23	16	36	36	21
The number of Correct	149	6	30	5	23	14	20	31	21
% Correct	78.4	85.7	65.2	100	100	87.5	55.6	86.1	100

In the study, a new classification system is developed. In the system, there are three stages. Each stage classifies the images into image types, image composition and semantic classes respectively. In Table 2.3, the results of the system are shown from eight semantics classes.

CHAPTER THREE
FEATURE EXTRACTION

Feature extraction is one of the most important operations in image classification. Before classifying the images, the features, which will be used for the classification, must be determined. If unsuitable features are chosen, the classification process can be unsuccessful. After the selection of the features, the method of feature extraction must be selected. Feature extraction method is crucial for image classification since this method affects the performance of the classification and specifies the conclusion of the classification. For this reason, in this chapter, firstly image features are summarized. Then, line and circle features of shape features and their extraction are explained in detail. Line feature detector as a line feature extraction method is mentioned to show how to work and how to extract line features. After that, circle feature extraction is presented to acquire circle features. To derive circle features from images, Hough transformation method is chosen for the application of the thesis and is stated in this chapter.

3.1 Image features

It is vital to extract the features for the image classification. Image's features generally consist of text, color, texture and shape features. The selection of features, which will be used for true classification, is very critical. Namely, choosing of image's features is application-dependent. For example, while we categorize the speckled images, it is not advantageous to be taken into account color features. On the contrary, for that shape feature is the best choice.

- **Text Features:** Text features are generally acquired from names of the images and HTML codes, which the images are placed. It is possible that these features can be acquired from the World-Wide Web.

- **Color Features:** Each image consists of certain color pixels. The number of certain color pixels of an image can discriminate one image class from the others. In some classifications, color features are extremely vital. For instance, in the classification of sea and sky pictures, it is quite reasonable to use the density of blue color on the top of the images.

- **Texture Features:** Texture feature is one of the important characteristics of an image, and described by the spatial distribution of gray levels in a neighborhood. Texture demonstrates its characteristics by both each pixel and pixel values. There are many approaches using texture for classification. However, the gray-level co-occurrence matrix seems to be a well-known statistical technique for texture feature extraction

- **Shape Features:** The basic shapes of an image are line, circle and ellipse. That is to say, these fundamental shapes can construct almost all shapes of images. For getting the shape features, some edge detection operators such as Sobel or Prewit edge detector are applied on gray-level or monochrome images.

3.2 Line feature detection

Firstly, some basic definitions and structures are shown before line detection is explained. They will be useful to be understood the system of line and circle detection.

3.2.1 Kernel

Sobel kernel :

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

Prewit kernel :

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

Figure 3.1 Two convolution kernels of Sobel and Prewit operator with 3×3 matrix

Kernel is a matrix of numbers as it is shown in Figure 3.1. It is used in image convolution. The kernels that include different patterns of numbers and have different size give different results with convolution. For example, Figure 3.1 shows a 3*3 Sobel kernel Prewit kernel that implements edge detection.

3.2.2 Structuring elements

For some image processing operations, there are two input data as image and structuring element. Structuring element decides the effect of the operations, which are done, on the image.

0	1	0
1	1	1
0	1	0

0	0	0
1	0	0
1	1	0

		1	1	1		
	1	1	1	1	1	
1	1	1	1	1	1	1
	1	1	1	1	1	
		1	1	1		

Figure 3.2 Some examples of structuring elements

The structuring element is sometimes called the kernel. But, the structuring element has patterns specified as the coordinates of numbers according to an origin. It is represented as a small image, which has Cartesian coordinates. In Figure 3.2, two different structuring elements are shown. The origin does not have to be on the center. It can be on any other position of the structuring element. The centers are shown as the underlined on the Figure 3.2.

For carrying out an operation the structuring element is overlapped onto the image, and then the value of the overlapped cell on the structuring element is compared with the value of the overlapped cell on the image. The results of the comparison are related with the processing operations.

3.2.3 Convolution

It can be said that convolution is the basic operation of image processing operators. Convolution constructs a third array that is represented as an output image after taking two arrays as an input image and a kernel with different sizes. The operators whose output pixel values are some linear combinations of input pixel values are implemented by convolution.

For the convolution operation, we need two input arrays. The first array is generally graylevel image. The second array is known as kernel. These two arrays are illustrated in Figure 3.3.

I_{11}	I_{12}	I_{13}	I_{14}	I_{15}	I_{16}	I_{17}	I_{18}
I_{21}	I_{22}	I_{23}	I_{24}	I_{25}	I_{26}	I_{27}	I_{28}
I_{31}	I_{32}	I_{33}	I_{34}	I_{35}	I_{36}	I_{37}	I_{38}
I_{41}	I_{42}	I_{43}	I_{44}	I_{45}	I_{46}	I_{47}	I_{48}
I_{51}	I_{52}	I_{53}	I_{54}	I_{55}	I_{56}	I_{57}	I_{58}

K_{11}	K_{12}	K_{13}
K_{21}	K_{22}	K_{23}
K_{31}	K_{32}	K_{33}

Figure 3.3 An example image (up) and kernel (down) to show convolution. The labels within each grid square are used to identify each square

The convolution operation is made with shifting the kernel over the image. First the kernel is placed at the top left corner. Then the kernel is shifted one pixel to the right until a pixel of the kernel is over the end pixel of the image row. After that, the kernel is moved to the pixel of next row and first column of the image. Before shifting operation, firstly some values are computed by multiplying the kernel pixel value with underlying image pixel value. Then the output value of the kernel is calculated by adding of these values.

Assume that the above kernel is over the up left pixel of the image. The first output value will be given as the following:

$$R_{22} = I_{11}K_{11} + I_{12}K_{12} + I_{13}K_{13} + I_{21}K_{21} + I_{22}K_{22} + I_{23}K_{23} + I_{31}K_{31} + I_{32}K_{32} + I_{33}K_{33}$$

After one right shift operation, the result output operation is:

$$R_{23} = I_{12}K_{11} + I_{13}K_{12} + I_{14}K_{13} + I_{22}K_{21} + I_{23}K_{22} + I_{24}K_{23} + I_{32}K_{31} + I_{33}K_{32} + I_{34}K_{33}$$

These output results are generally put at the center pixel of the output image, which is under the kernel. For example, first calculated value R_{22} is assigned to the value of I_{22} of the output image, but the second value R_{23} is assigned to the value of I_{23} of the output image.

The convolution is written as the following: $O_{i,j} = \sum_{k=1}^{m}\sum_{s=1}^{n} I_{i+k-1,\, j+s-1}\, K_{k,s}$

where i is a value from 1 to $R_I - R_K + 1$ and j is a value from 1 to $C_I - C_K + 1$. R_I, R_K, C_I, C_K are image row, kernel row, image column and kernel column respectively.

3.2.4 Skeletonization

Skeletonization is a method to lessen foreground regions in a binary image. Skeletonization does not destroy the main structure and connectivity of the image, but it decreases the original foreground pixels to the single pixel thickness. We say that this end structure of the image is the skeleton of the image.

There are two fundamental methods for generating the skeleton. One is to use morphological thinning. In this method, the boundary of the image is reduced with iteration of the process. The process is stopped until no more thinning is possible. The second method is related with the distance transform of the image. After the computing the distance transform, the singularities of the transform specify the skeleton. We can say that there are many skeletonization algorithms because of many distinct distance transforms, but the results are quite similar.

The skeleton gives a general structure of a shape, which is similar to the shape of the original image. The topological and size properties are not generally destroyed. Skeletons provide the length of the shape and the end points of the shapes. It is also useful to compare the shapes of different images.

3.2.5 Thinning

Thinning is an operation that is applied to binary images, and it produces another binary image as output. It destroys some foreground pixels of the image. Thinning is useful for some applications such as skeletonization. It is generally used to regulate after the application of some edge detectors. It uses a structuring element.

For the thinning operation, it is firstly necessary to translate the origin of the structuring element to each possible pixel position in the image, and to compare it with the underlying image pixels at each position. 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. Otherwise nothing is done.

It is also very important to select the structuring element because it decides the application of the thinning operation. In other words, it decides the foreground and background pixels of the image. For this operation, single pass cannot be enough, so this operation can continue again and again until no change exists on the image.

Thresholding is one of the most common uses of thinning. It lessens the output of an edge detector to lines of a single pixel thickness, but it does not reduce the length of the lines in the image.

For morphological operation, the eight structuring elements that are shown in Figure 3.4 are necessary. These structuring elements are repeatedly used to construct final shape of the image. In this process, the skeleton of a binary shape is determined. After skeletonization process, a connected skeleton is produced.

1	1	1
	1	
0	0	0

	1	
0	1	1
0	0	

0		1
0	1	1
0		1

0	0	
0	1	1
	1	

0	0	0
	1	
1	1	1

	0	0
1	1	0
	1	

1		0
1	1	0
1		0

	1	
1	1	0
	0	0

Figure 3.4 Eight structuring elements of skeletonization with thinning. The center pixel is the origin of the structuring element

3.2.6 Thresholding

Thresholding is an important operation to set apart the objects on the image and the background of the image, and it is also vital to execute the segmentation on color intensities of the background and the foreground of the image.

For thresholding, there are input and output images. The input can be grayscale or color images. The output is changed by simple and complicated implementations. In simple implementation, the output is binary image, however the output can be color or grayscale image in complicated implementation. In simple implementation, segmentation is decided by single threshold. According to this threshold, every pixel is compared with it. The pixel's color will be white if the pixel's intensity is bigger than the threshold. Otherwise the pixel's color will be black. For the complicated implementations, more than one threshold can be used. The pixel's value can be set to white for the values between some specified thresholds. Otherwise it is set to black. For color images, distinct thresholds can be used for different channels. It is also possible to say that the pixel values of background region can be set to black, and the other pixel values are not changed.

3.2.7 Line detection

In this study, convolution-based technique is used for line detection. This technique constructs an output image with thin lines in the input image. The other technique is Hough transform for line detection. Unlike convolution-based technique, this technique produces a parametric description of the lines. Hough transform is explained in Section 3.2.2 for circle detection.

In convolution-based technique, the line detection operator that includes a convolution kernel to detect the lines and their orientations is used. In Figure 3.5, eight convolution kernels of this technique are shown, and every kernel represents one direction of the line and one pixel width.

To collect the light lines against dark background, the operator gives a high response to dark lines against a light background. The kernel values must be multiplied with -1 to detect dark lines against light background. To detect any kind of line, the sign of the operator result is not important, so the convolution output must be taken with absolute value.

Assume that O_n is output of the kernel n. For any pixel of the image $O_m < O_n \; \forall n \neq m$, then we can say that the point is more likely to include a line whose width and direction correspond to those of the kernel n. To eliminate the lines, it is possible to use some thresholds.

-1	-1	-1
2	2	2
-1	-1	-1

a)

-1	2	-1
-1	2	-1
-1	2	-1

b)

-1	-1	2
-1	2	-1
2	-1	-1

c)

2	-1	-1
-1	2	-1
-1	-1	2

d)

Figure 3.5 Four line detection kernels: a) horizontal kernel b) vertical kernel c) 45 degree kernel d) 135 degree kernel

3.3 Circle Feature Detection

3.3.1 Sobel Edge Detector

Sobel operator is one of the most effective edge detection operators. In general, the operator has two 3*3 convolution kernels. The second kernel is obtained by rotation of ninety degrees of the first kernel. In Figure 3.6, these kernels are shown as G_x and G_y. Absolute gradient magnitude and orientation can be computed by using these kernels on the image. Therefore, the edges of the images can be found. Each kernel can be used separately on the image, and they construct their gradient magnitudes, $|G_x|$ and $|G_y|$. Using $|G_x|$ and $|G_y|$, the absolute magnitude and gradient orientation θ can be calculated as the followings:

The gradient magnitude is given by $|G| = \sqrt{((G_x)^2 + (G_y)^2)}$. An alternative magnitude can be calculated by $|G| = |G_x| + |G_y|$. The second magnitude is faster than the first. The angle of orientation of the edge giving rise to the spatial gradient is given by $\theta = \arctan(G_x / G_y)$.

The angle of the gradient is anticlockwise from zero degree, which is the direction from left to right.

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

G_x

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

G_y

Figure 3.6 Sobel convolution kernels: G_x is the first kernel. G_y is the second kernel

P_1	P_2	P_3
P_4	P_5	P_6
P_7	P_8	P_9

Figure 3.7 Pseudo-convolution kernels for measuring approximate gradient magnitude

With the above kernel, the magnitude can be computed by

$$| G | = | (P_1+2xP_2+P_3) - (P_7+2xP_8+P_9) | + | (P_3+2xP_6+P_9) - (P_1+2xP_4+P_7) |$$

Sobel operator is quite similar to Robert cross operator, but sobel operator is slower to calculate than Robert cross operator. Comparing the results of the Robert cross, Sobel operator is more successful.

3.3.2 Hough transform

Hough transform is applied differently for different shapes on the images. It is a method to specify general characteristics of the shapes on the images. There are two versions of Hough transforms. One is the classical Hough transform, which is most commonly used for the detection of regular curves as ellipses, circles, and lines. The other one is the generalized Hough transform, which is employed in application where a simple analytic description of features is not possible. In this thesis, classical Hough transform is used for circle detection.

Hough transform is very effective because it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise and it computed a global description of a features where the number of solution classes need not be known as priori. Every input measurement of line detection with Hough transform computes its contribution to a globally consistent solution.

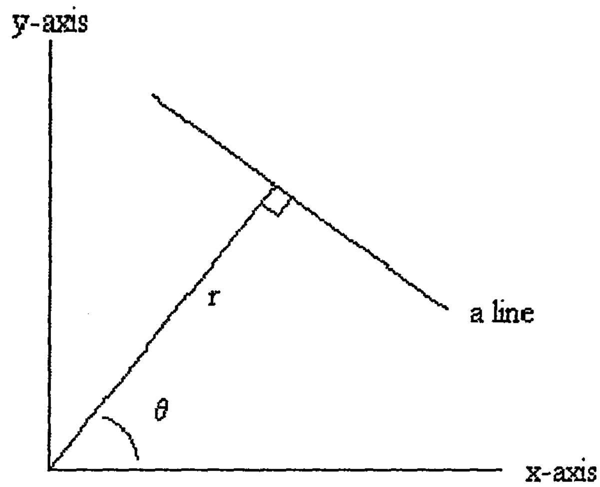


Figure 3.8 Polar coordinates for a line

A line can be described in some forms. A line can be represented with the equation: $r = x \cos\theta + y \sin\theta$ where r is the length of a normal from the origin to this line and θ is the orientation of r with respect to the x-axis. In above formula, r and θ are constant for any x and y values.

The coordinates of the points of edge segments (x_i, y_j) in the image are known and therefore serve as constants in the parametric line equation whereas r and θ are unknown variables. If every (x_i, y_j) is represented by (r, θ) value which is shown in Figure 3.8, the points of (x_i, y_j) Cartesian space of image maps to curves of (r, θ) polar Hough parameter space.

For the lines, Hough transformation is point to curve transformation. Points of Hough transformations, which are collinear in the Cartesian space, become readily apparent as they yield curves, which intersect at a common point.

Hough parameter space is quantized into the accumulator cells. For any i and j, (x_i, y_j) is transformed into (r, θ) polar Hough parameter space. Accumulator cells are incremented if accumulator cells lie along the (r, θ) curve. The maximum points of the accumulator cells represent straight line.

For circle detection, we can use the procedure that is applied for line detection. The circle equation can be represented as $r^2 = (x - c_1)^2 + (y - c_2)^2$ where c_1 and c_2 are the coordinates of the center of the circle. r is the radius of the circle. Because there are three coordinates in parameter space, we need a three dimensional accumulator arrays.

The Hough transform is applicable to find the parameters of a curve that is suitable to a set of given edge point. The edge point can be computed Sobel, Canny edge and Robert Cross detector. When the Hough transform is applied, we can find the location of the shape, the number of the shape. The boundary of images does not tell us anything about the identity of features in this boundary description. In this condition, Hough transform can be used to detect the shapes.

The maximum intersection points (r, θ) in the Hough transform space characterize the line segments in the image. For extracting the local maxima from the accumulator array, there are many techniques. Some of them uses thresholding to find the shapes. Relative threshold is used to extract the unique points that represent the straight lines in the image.

It is possible to verify the result of the Hough transform, which is applied to detect the shapes of the image. Hough transform space is mapped into Cartesian space. This gives a set of line features in the image. This constructed version of the original image provides us how much correct result is found with Hough transformation.

CHAPTER FOUR
CLASSIFICATION METHODS

It can be said that image classification is the process of assigning the pixels (images) to discrete categories (classes). In other words, the purpose of image classification process is to categorize all of the pixels (images) in a digital image into one or more classes. Traditionally, automated image classification has been divided into Supervised and Unsupervised methods.

The primary difference between these two main methods is the amount of information that is known about the images. In a Supervised classification, the user generally has some verification or training data. This means that the user generally knows what the features on the image represent.

An Unsupervised classification is a method used to classify unknown pixels and place them into classes based on natural groupings. This does not include user-specified training data.

Image classification is the most important part of digital image analysis. If images are classified some suitable categories, it provides high performance for image retrieval systems.

4.1 Supervised classification

With Supervised classification, we identify examples of the Information classes of interest in the image. These are called "training sites". The image processing software system is then used to develop a statistical characterization of the reflectance for each information class. This stage is often called "signature analysis" and may involve developing a characterization as simple as the mean or the rage of reflectance on each bands, or as complex as detailed analyses of the mean, variances and covariance over all bands. Once a statistical characterization has been achieved for each information class, the image is then classified by examining the reflectance for each pixel and making a decision about which of the signatures it resembles most.

4.1.1 Nearest Neighbor

A resampling method in which the output data file value is equal to the input pixel whose coordinates are closest to the retransformed coordinates of the output pixel. Nearest Neighbor rule is one of the oldest and simplest classification techniques. This decision rule assigns to an unclassified sample point the class label of the nearest point in a set of previously classified points. This rule is based on the assumption that observation, which is close together in some appropriate metric, will have the same classification, or at least will have almost the same posterior probability distributions on their respective classifications. The Nearest Neighbor rule is known to be sub-optimal, in the sense that its use will always lead to an error rate that is greater than the minimum possible, that is the Bayes rate. However, very well known results showed that when the number of samples is unlimited, the error rate is never worse than twice the Bayes rate. In fact, if P_n is the error rate on n samples, and if we define:

$$P = \lim_{n \to \infty} P_n$$

then it can be shown that, in the case of binary classification,

$$P^* \leq P \leq P^* (2 - P^*)$$

Where P^* is the Bayes error rate.

This result is only asymptotic, and should be taken with extreme care. However, the Nearest Neighbor decision rule has been proven to effective in a number of cases, and the pattern recognition community has taken renovated interest in techniques of this type. Moreover, the Nearest Neighbor method can be used as a general indicator of the performances of other, more sophisticated techniques.

A crucial choice in the Nearest Neighbor technique is the choice of the distance function. There are some metrics that are generally used in classification methods. These are Euclidean distance, Manhattan distance (or taxicab, or city block, or Hamming distance) and Sup distance.

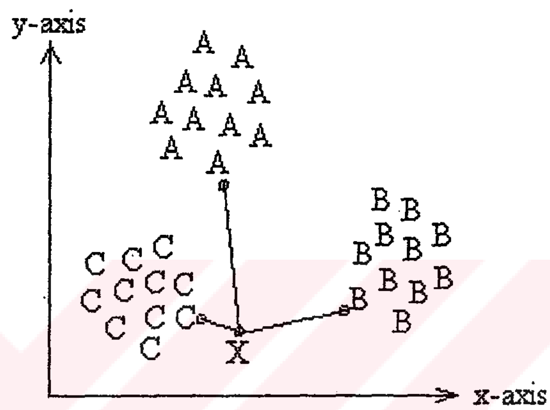


Figure 4.1 Nearest Neighbor classification method

Let $[x_{ij}]$ be a pattern matrix, where x_{ij} is the j^{th} feature for the i^{th} pattern. The i^{th} pattern, which is the i^{th} row of the pattern matrix, is denoted by the column vector x_i, $x_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{id})^T$, $I = 1,2,3,\ldots,n$ where d is the number of features, n is the number of patterns, and T denotes vector transpose.

Euclidean distance: $d(i,k) = \sum_{j=1}^{d} \left((x_{ij} - x_{kj})^2 \right)^{1/2}$

Manhattan distance: $d(i,k) = \sum_{j=1}^{d} | x_{ij} - x_{kj} |$

Sup distance: $d(i,k) = \max_{1 \le j \le d} | x_{ij} - x_{kj} |$

4.1.2 *k*-Nearest Neighbor

The Nearest Neighbor rule (Ricci & Avesani, 1999) can often be unstable and sensitive to outliers. A simple extension is the *k*-Nearest Neighbor rule, which consists in assigning to an unclassified point the class label that is most heavily represented among its *k* Nearest Neighbor. (is odd to avoid ties). In the limit of a large number of data points, it can be shown that a large value of *k* will lead to consistent improvements over the standard Nearest Neighbor rule. However, finite sample size, there is a trade off between the optimal value of *k* and the size of the data set. For small values of *k* an improvement can be usually obtained over the case $k = 1$, but when *k* is very large the size of the neighbor spanned by the *k* closest point becomes comparable to the size of the data set, and therefore the *k*-Nearest Neighbor rule will tend to classify points with the label of the most represented class in the data set, loosing its significance.

k–Nearest Neighbor classifier (Akkus & Guvenir, 1996) is an instance based learning method. It computes the similarity between the test instance and the training instances and considering the *k* top ranking nearest instances, find out the category that is most similar. There are two methods for finding the most similar instance: *majority voting* and *similarity score summing.*

In majority voting, a category gets one vote for each instance of that category in the set of *k* top–ranking nearest neighbors. Then the most similar category is the one that gets the highest amount of votes. In similarity score summing, each category gets a score equal to the sum of the similarity scores of the instance of that category in the *k* top-ranking neighbors. The most similar category is the one with the highest similarity score sum.

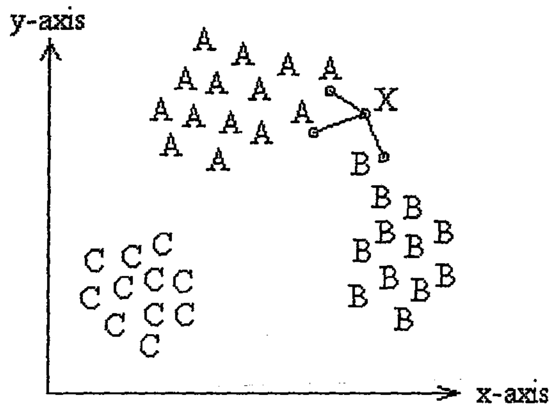


Figure 4.2 *k*-Nearest Neighbor classification method with *k*=3

The similarity value between two instances is the distance between them based on a distance metric. Generally Euclidean distance metric is used.

4.1.3 Minimum Distance classification

Minimum Distance classifies image data on a database file using a set of 256 possible class signature segments as specified by signature parameter. Each segment specified in signature, for example, stores signature data pertaining to a particular class. Only the mean vector in each class signature segment is used. Other data, such as standard deviations and covariance matrices, are ignored (though the maximum likelihood classifier uses this).

The result of the classification is a theme map directed to a specified database image channel. A theme map encodes each class with a unique gray level. The gray-level value used to encode a class is specified when the class signature is created. If the theme map is later transferred to the display, then a pseudo-color table should be loaded so that each class is represented by a different color.

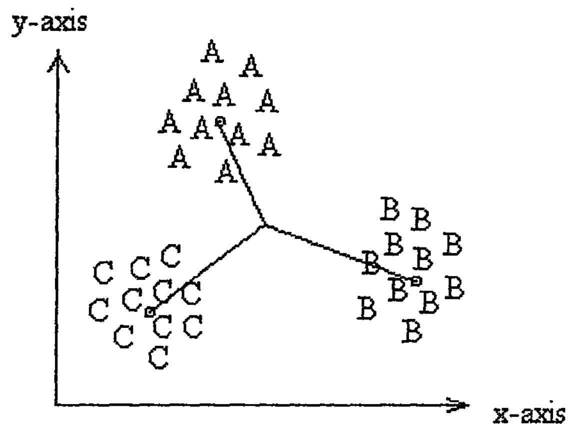


Figure 4.3 Minimum Distance classification method

This is a classification technique that assigns raw data to the class whose mean falls the shortest Euclidean distance (or any their distance metric) from it. This decision does not take into account variability or dispersion.

The distances between the pixel to be classified and each class center are compared. The pixel is assigned to the class whose center is the closest to the pixel.

4.2 Unsupervised classification

Unsupervised classification is a method that examines a large number of unknown pixels and divides into a number of clusters based on natural groupings presented in the image values. Unlike Supervised classification, Unsupervised classification does not require analyst-specified training data. The basic premise is that values within a given cover type should be close together in the measurement space (i.e. have similar gray levels), whereas data in different classes should be comparatively well separated. Figure 4.4 shows an example of Unsupervised classification.

The classes that result from Unsupervised classification are spectral classed which based on natural groupings of the image values, the identity of the spectral class will not be initially known, must compare classified data to some from of reference data (such as larger scale imagery, maps, or site visits) to determine the identity and informational values of the spectral classes. Thus, in the Supervised approach, to define useful information categories and then examine their spectral separability; in the Unsupervised approach the computer determines spectrally separable class, and then define their information value.

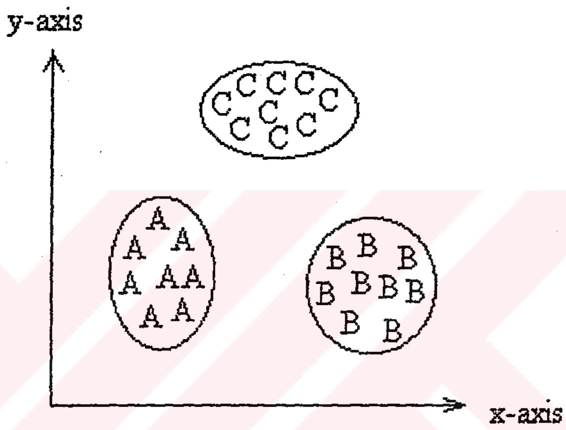


Figure 4.4 Clustering or Unsupervised classification

Unsupervised classification is becoming increasingly popular in agencies involved in long-term GIS (Geograhical Information System) database maintenance. The reason is that there are now systems that use clustering procedures that are extremely fast and require little in the nature of operational parameters. Thus it is becoming possible to train GIS analysis with only a general familiarity with remote sensing to undertake classifications that meet typical map accuracy standards. With suitable ground truth accuracy assessment procedures, this tool can provide a remarkably rapid means of producing quality land cover data on a continuing basis.

4.3 Manual Thresholding

Manual thresholding is a classification method, which uses some threshold values to categorize the input images. This method is an application-dependent, so each application can use some different threshold values.

In Figure 4.5, an example is given for this classification method. In this example, there are two classes with a 2-dimensional space. In Figure 4.5, the threshold value is given as 10, therefore it is possible to separate the two different classes. Here, the x-values are not used to categorize the two classes, however for the y-values which are greater than 10, we can say that the images are in *A* class. Otherwise, the images are in the *B* class.

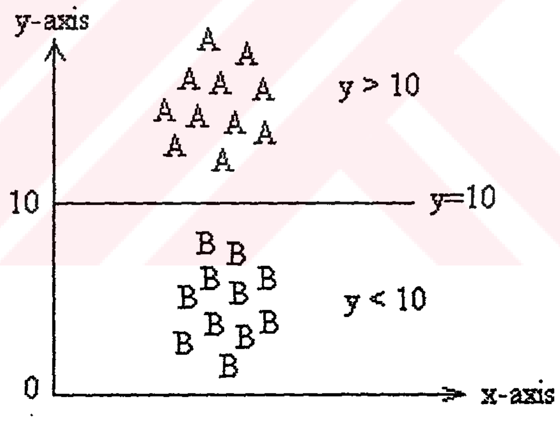


Figure 4.5 Separation of two classes using manual thresholding

CHAPTER FIVE
IMPLEMENTATION

5.1 Introduction

There are approximately 3500 textile images to classify into striped, plaided and speckled classes in this study. Line and circle features are fundamental features for this classification since striped and plaided images consist of line features, and speckled images include circle features. Line detection operator is used for line feature extraction, but for circle feature extraction, Hough transform is suitable.

After the line and circle feature extraction, the classification methods as Manual Thresholding, Nearest Neighbor, *k*-Nearest Neighbor and Minimum Distance methods decide the images' classes. Namely, the image can be categorized into striped, plaided and speckled class with the above classification methods.

In this chapter, the user-interface of the program 'Image Lab' is shown. Line and circle detection algorithms, which are used in this study, are explained in detail. The algorithms of the classification methods are also included.

Some classification results with images are added in Appendix B, and the flow diagrams of the algorithms are given in Appendix A.

5.2 The user-interface of the executable program "Image Lab"

The executable program "Image Lab" which is written the programming language "Delphi 3.0" on Windows 95 operating system can be executed on Windows 95, Windows 98, Windows ME, Windows 2000 and Windows NT. After the installation of this program, the executable file and other program files are copied to the main installation directory. And a shortcut of the executable program is constructed on the desktop of the computer.

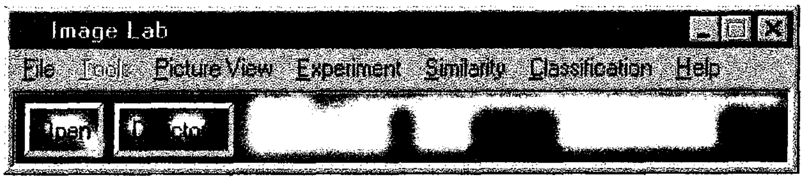


Figure 5.1 General appearance of the executable program "Image Lab"

When the program "Image Lab" is executed, a main window is shown as in Figure 5.1. As it is mainly expected, this program can make the classification of textile images, which is categorized into striped, plaided and speckled image classes. It also categorizes striped images into the four classes that have horizontal, vertical, 45 degree and 135 degree line. The program can construct the color, line and circle histograms with using the window in Figure 5.2.

At the same time, this program can shows the images that have jpeg format and process different sized images with some operators such as Sobel, Prewit, etc. According to color feature, the program can list the images that are similar to the image selected by the user.

After the program starts, "Working directory" and "Image directory" must be selected by pushing the "Directory " button. "Working directory" is the directory that includes all the files related with the programs. "Image directory" is the directory that the image database is placed.

Before the classification of the images, line and circle histograms must be constructed with the window, which is shown in Figure 5.2. These histograms increase the performance of the classification since the features are extracted only once with the histograms, and they can be used many times at the classification stage.

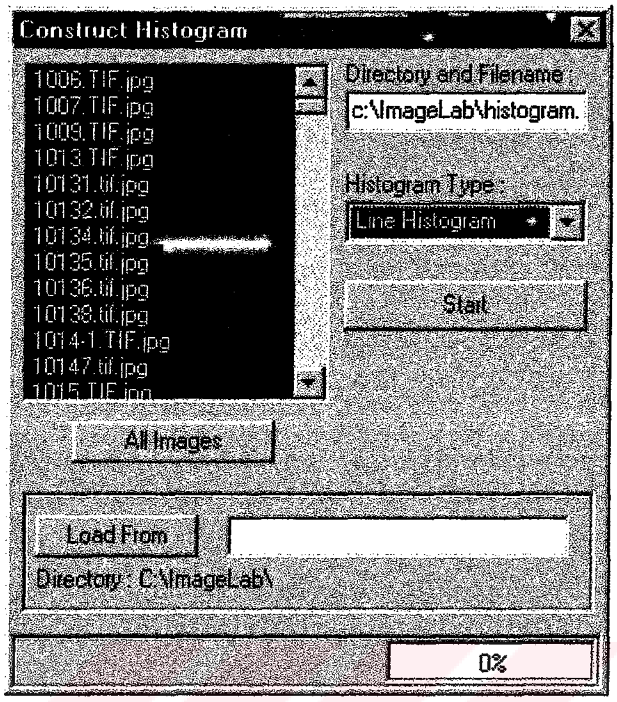


Figure 5.2 Histogram construction part of the executable program "Image Lab"

The window in Figure 5.2 is accessed from the "Experiment" menu in the main window. After loading images from "All Images" in the image directory or "Load From" image files in the working directory, the histogram name is entered and chosen the histogram type. Feature extraction begins with "Start" button. The histogram file extension is "hst".

Figure 5.3 shows circle features of some images. The features represent the number of circles in the image. There is an image file in the first line and the number of circles of the image in the second line.

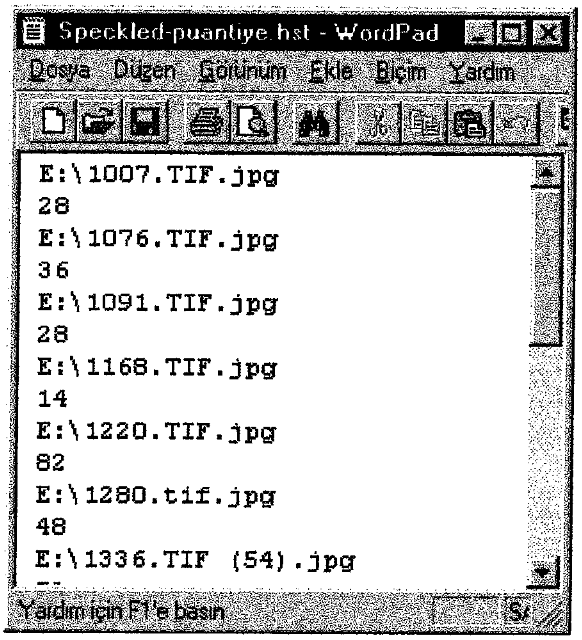


Figure 5.3 Contents of "Speckled-puantiye.hst" file

Figure 5.4 shows line features of some images. The features represent the number of lines in the image. There is an image file in the first line and the number of lines of the image in the second line. The first number in the second line represents the number of pixels of the image. The second number represents the number of pixels, which have the horizontal line property. The third number represents the number of horizontal lines in the image. The fourth number represents the number of pixels, which have the vertical line property. The fifth number represents the number of vertical lines in the image. This kind of features continues for 45 degree and 135 degree lines.

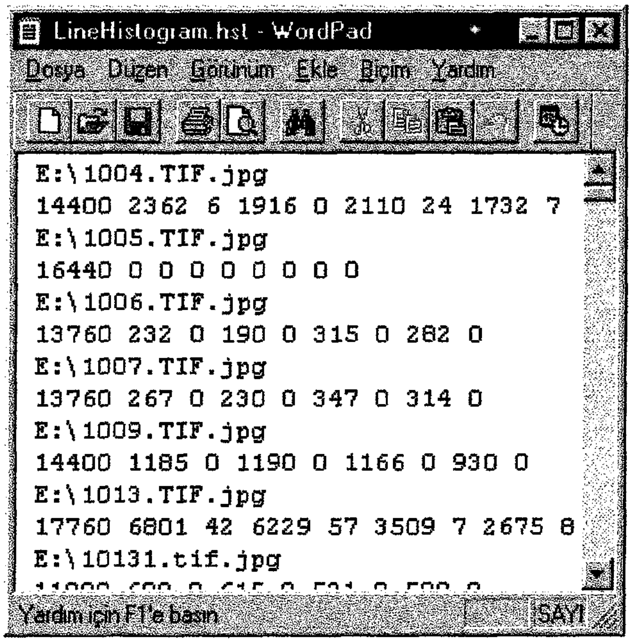


Figure 5.4 Contents of "LineHistogram.hst" file

Figure 5.5 shows "Class Definition" window, which defines the main classes and sub classes. The main classes, which have "def" file extension, consist of sub classes. Sub classes include the training classes, which the classification methods use for the classification. These classes have "trn" file extension. When a sub class is generated, an empty file, which has "cls" file extension and same file name with the "trn" file, is created since the classified images are placed into this "cls" files after the classification process.

In this window, Add, delete and rename operations can be done for the main and sub classes. Training sets can be constructed with adding the images from the image set to training set. The images in the image set can be loaded from an image directory or a file, which has image file names.

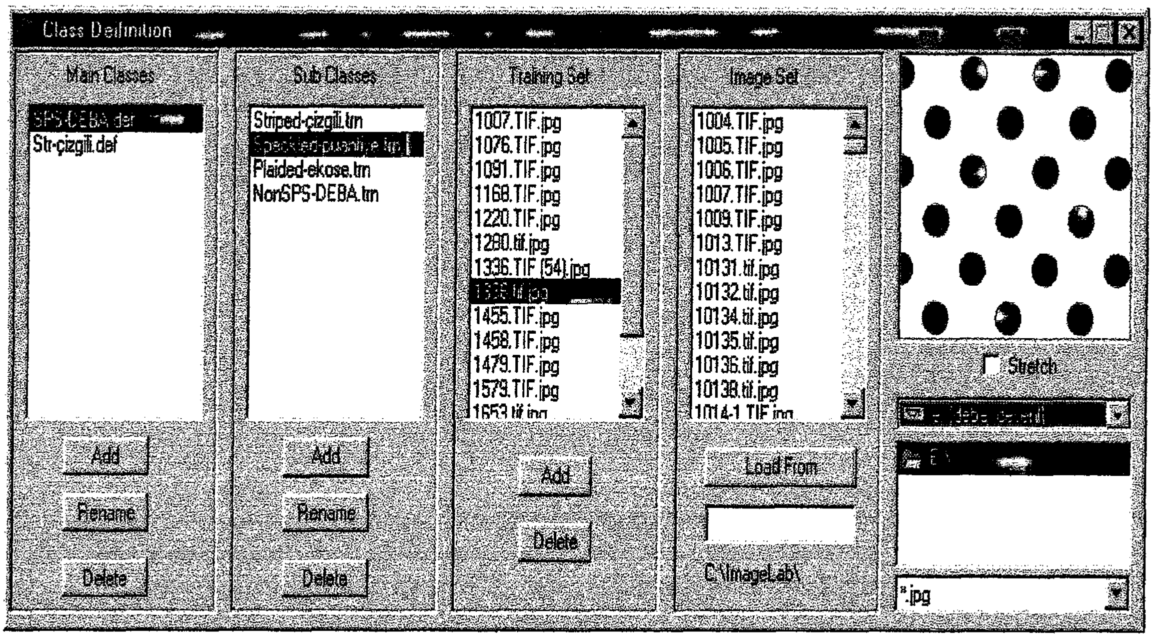


Figure 5.5 The "Class Definition" part of the executable program "Image Lab"

The window in Figure 5.6 classifies the selected images in the "Image Collection" list box with the suitable methods. The methods as Manual Thresholding, Nearest Neighbor, *k*-Nearest Neighbor and Minimum Distance can be chosen from the combo box. Each method has its own parameter so these parameters must be entered after pushing the "Start" button to begin the classification. Main class, the images that will be classified, and the classification methods must be selected to start the classification. Otherwise an error message will be seen.

The images in the training classes and classified classes can be listed on the window. All classified images can be cleared, and the classification process can be done again and again. When clicking the images, they can be seen on the right upper side of the window. Clicking "Show All Images" and "Show Images" buttons can show twelve images in one window.

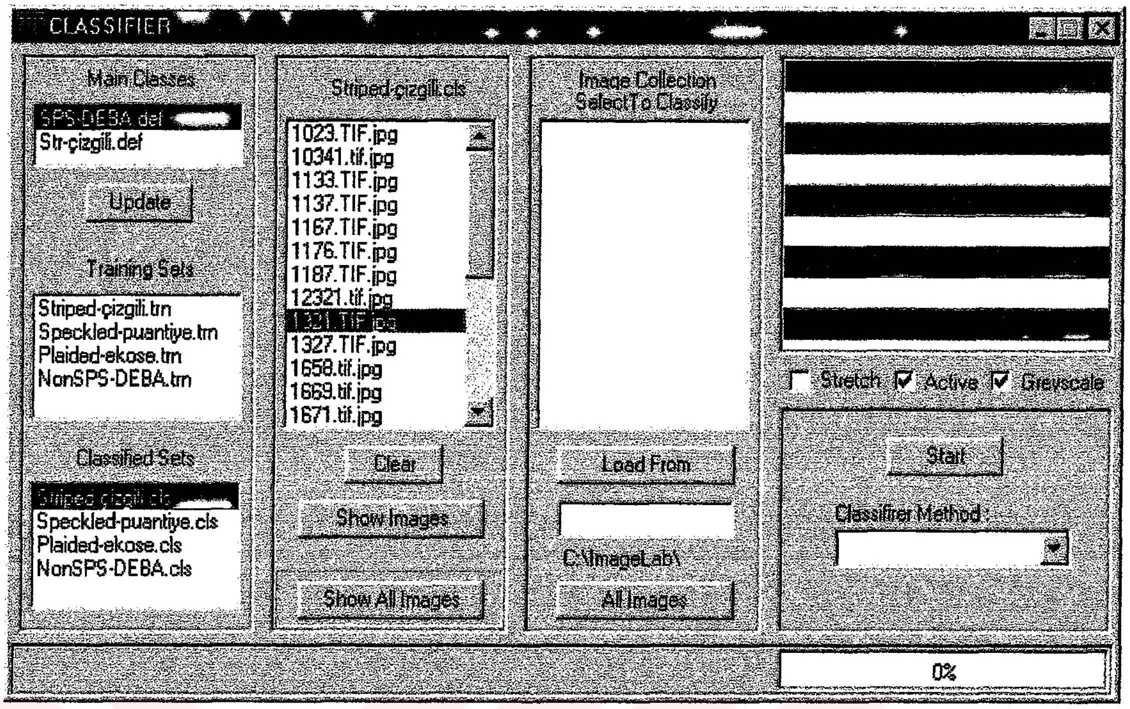


Figure 5.6 The "Classifier" part of the executable program "Image Lab"

5.3 The algorithm of line-feature extraction

We have used horizontal, vertical, 45-degree and 135-degree line detection algorithms for line detection. But, we mention about horizontal line algorithm since all four line detection algorithms are very similar except their kernels.

5.3.1 Horizontal line detection algorithm

Horizontal line detection uses horizontal kernel in Figure 3.5. After deciding line threshold and converting the color image to grayscale image, we can start the detection operation. In this algorithm, two-dimensional array represents the images. The operations from 11 to 15 in Algorithm I are applied for all pixels in the image. Then, the convolution results of the pixels are calculated. If the convolution result is greater than the threshold, we say that there is a horizontal line property in that pixel, and the pixel's color must be white. Otherwise, the pixel's color is black.

Algorithm I: Horizontal line detection

```
1. Define a 3*3 kernel matrix for horizontal line detection.
2. Decide the threshold for line detection.
3. Create a 2-D jpg image array, which represents RGB color image.
4. Convert the color image array to grayscale image array.
5. Create a 2-D temporary image array.
6. Initialize the temporary image array to 0 (black color).
7. y, which is the row position of the array, is assigned to 0.
8. While y is smaller or equal to (height - 3) of image array do
9. Begin
10. x, which is the column position of the array,is assigned to 0.
11.    While x is smaller or equal to (width-3) of image array do
12.    begin
13.       Obtain the result of the convolution: Result = 2 *
          (H4 + H5 + H6)-(H1 + H2 + H3  + H7 + H8 + H9)
14.       if Result is greater than the line threshold then
15.          The temporary array [x+1,y+1] = 255
             (white color) //center pixel under the kernel.
16.       x =x+1 //Shift the kernel to the right
17. End  "while" loop 11
18. y = y+1 //Shift the kernel to the first column and 1 row down
19. End of "while" loop 8
//After the loop operation, temp image array represents the
  horizontal line features with white color.
//Convert the temporary image array to image format.
```

5.4 Circle detection algorithm

Circle detection algorithm is based on Hough transform in this study. Circle detection uses Sobel operators in Line 1 of Algorithm II. Sobel operator is useful to find the gradient magnitude and orientation of the pixels of the images. First, the threshold values specified in Line 4 of Algorithm II are decided. After converting the color image to grayscale image, we can start the circle detection operation. The operations from Line 10 to Line 19 in Algorithm II are applied for each pixel in the image to explore the gradient magnitude and orientation of the pixels. The skeletonization process from Line 23 to Line 32 in Algorithm II is applied if it is allowed. The operations from Line 33 to Line 44 in Algorithm II compute the centers of the circles. This is the most important part of Hough transformation.

Finally, we divide the accumulation center array, which intersections of gradient lines are placed, into small rectangle parts, which have equal number of pixels, and find the maximum intersection of gradient lines of each rectangle parts. This maximum value represents the center of a circle. If the maximum intersection pixels of the rectangles are greater than a threshold value, which is given by the user, then the pixels are the centers of some circles. To ensure that the maximum intersection is the center of a circle for some radius intervals, a circumference is generated, and then the circumference is overlapped on the grayscale image. The number of intersections after overlapping is calculated. So, we can decide the radius of circle with the threshold value.

Algorithm II: Circle detection

```
1. Define two Sobel operators. Two 3*3 matrix:
```

$$Gx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad Gy = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

```
2. Create a 2-dimensional jpg image array, which represents a RGB
   color image.
3. Convert the color image array to gray scale image array.
4. Specify some threshold values such as Sobel threshold,
   accumulation center threshold, minimum radius, maximum radius,
   and iteration for skeletonization.
5. Create temporary image array, which represents the values of
   magnitude that is greater than the threshold after applying
   Sobel edge detection.
6. Create orientation array that represents the orientation of the
   edges.
7. Initialize temporary image array to 0 and orientation array to
   -1
8. y which is the row position of the array is assigned to 0.
```

9. While y is smaller than or equal to ($height$-3) of image array do

10. Begin

11. x, which is the column position of the array, is assigned to 0.

12. While x is smaller or equal to ($width$-3) of image array do

13. Begin

14. Calculate the first gradient Gx and the second gradient Gy.

The image matrix $\begin{bmatrix} x,y & x+1,y & x+2,y \\ x,y+1 & x+1,y+1 & x+2,y+1 \\ x,y+2 & x+1,y+2 & x+2,y+2 \end{bmatrix}$

is under the kernel.

$$|Gx| = (-1)(\text{image}[x,y]+2\ \text{image}[x,y+1]+\text{image}[x,y+2])$$

$$+(\text{image}[x+2,y+2]+2\ \text{image}[x+2,y+1]+\text{image}[x+2,y])$$

$$|Gy| = (-1)(\text{image}[x,y+2]+2\ \text{image}[x+1,y+2]+\text{image}[x+2,y+2])$$

$$+(\text{image}[x+2,y]+2\ \text{image}[x+1,y]+\text{image}[x,y])$$

15. Total gradient magnitude is addition of the absolute value of Gx and the absolute value of Gy.

16. If total gradient is greater than Sobel threshold then

17. temporary image array [x+1,y+1]=255 (white color)

18. orientation array [x+1,y+1] = arc tan(Gy / Gx)

//The angle of the gradient.

19. x=x+1 // Shift the kernels to the right

20. End of "while" loop 11

21. y=y+1 //Shift the kernel to the first column and one row down.

22. Transfer the values of temporary image array to image array.

23. If skeletonization is allowed then

24. Begin

25. Repeat the number of iteration chosen do

26. Begin

27. For all of the eight skeletonization kernels as in Figure 3.4.

28. Begin

29. Make convolution operation for all pixels of the image.

30. If all cell values under the kernel and kernel cell values equal then

```
31.           Image array's center value = 0 and
              its orientation array value = -1
32.      End of "Repeat" loop
  // Until now, we have constructed a binary edge image
     and each white point's orientation.
  // By Sobel operator, each edge's orientation is set
     to orientation array.
  // The cells of the orientation array represent a
     point of a line in the image.
  // To construct accumulation array
33. For all cells of orientation array [x, y] do
34.   Begin
35.   If orientation array [x, y] does not equal -1
       (There is an edge) then
36.      Begin
37.      For Tx = 0   to the width of the
              accumulation center array do
38.      Begin
39.      Ty = round(((Tx-x) tan(orientation array [x, y])) + y
40.      If (Ty is greater than or equal to 0) and (Ty is smaller
          than height of accumulation array) then
41.           Add one to accumulation center array [Tx, Ty]
42.      End of "for" loop 40
43.   End of "if" condition 38
44.  End of "for" loop 36
  // Center points are obtained.
45. Divide the accumulation center array into some rectangle
     parts, which have equal number of pixels.
46. Find the maximum intersection of gradient lines of each
     rectangle parts.
   // This maximum value represents the center of a circle.
47. if the maximum intersection pixel of the rectangles is greater
     than a threshold value which is given by the user then the
     pixel is the center of some circles.
48. To ensure that the maximum intersection is the center of a
     circle for some radius intervals, a circumference is
     generated, and then the circumference is overlapped on the
     grayscale image, and the number of intersection after
     overlapping is calculated. So we can decide the radius of
```

```
circle with the threshold value.
```

5.5 The algorithm of Manual Thresholding method

Algorithm III.a shows the algorithm of Manual Thresholding method for horizontal line class. We have a line feature histogram, which consists of number of horizontal, vertical, 45-degree and 135-degree lines for each image. So there are four line features for all images. If the number of horizontal lines of the image is greater than the line threshold, and the number of the other three features equal to zero, then the image is assigned to "Horizontal" class. Otherwise, the image does not assigned to the "Horizontal" class. The same procedure is applied to "Vertical", "45-degree" and "135-Degree" classes.

Algorithm III.b shows Manual Thresholding method for "Striped", "Plaided" and "Speckled" classes. We have four line features and one circle feature for the classification of "Striped", "Plaided" and "Speckled" classes. In circle histogram we have the number of circles for each images. We determine the line threshold for "Striped" and "Plaided" classes and the circle threshold for "Speckled" class. If the number of horizontal lines and the number of vertical lines greater than the line threshold, and the number of 45-degree and 135-degree lines equal to zero, then the image is assigned to the "Plaided" class. If one of the four line features is greater than zero, and the other three line features equal to zero, then the image is assigned to the "Striped" class. If the number of circles is greater than the circle threshold, then the image is assigned to the "Speckled" class.

Algorithm III.a: Manual Thresholding method for horizontal line class

```
1. Decide the line threshold value.
2. For each image which will be classified
3. Begin
4.    If the number of horizontal lines of the image is greater
      than the line threshold and the number of the other three
      features equal to zero then
5.        The image is assigned to "Horizontal" class
```

```
6.    Else
7.          The image is assigned to "None" class.
8. End of "for" loop
```

Algorithm III.b: Manual Thresholding method for "Striped", "Plaided" and "Speckled" classes

```
1. Decide the line threshold for "Striped" and "Plaided" classes.
2. Decide the circle threshold for "Speckled" class.
3. For each image that will be classified do
4. Begin
5.   If the number of horizontal lines and the number of vertical
     lines greater than the line threshold and the number of
     45-degree and 135-degree lines equal to zero then
6.       Assign the image to  "Plaided" class.
7.   Else if one of the four line features is greater than zero
     and the other three line features equal to zero then
8.        Assign the image to  "Striped" class.
9.   Else if the number of circle is greater than the circle
     threshold then
10.   Assign the image to  "Speckled" class.
11.    Else
12.   Assign the image to  "None" class.
13.  End of "for" loop
```

5.6 The algorithm of Nearest Neighbor classification method

Algorithm IV.a shows the algorithm of Nearest Neighbor classification method for horizontal, vertical, 45-degree and 135-degree lines classes. We have four line features from line histogram, which consists of number of horizontal, vertical, 45-degree and 135-degree lines for each image, and we have five training classes which have been manually generated with images. To classify an image, its distance to the all images of the training classes is computed as it is shown between Line 1 and Line 12 in Algorithm IV.a. This classification uses Manhattan distance metric. The image, which will be classified, is classified to the class whose image is the nearest distance to the image.

Algorithm IV.b shows the algorithm of Nearest Neighbor classification method for "Striped", "Plaided", "Speckled" and "None" classes that are manually generated. We have four line features and one circle feature to classify images. The distances among the image which will be classified and the images in the training classes are calculated by Manhattan distance metric. There are two procedures for the classification. One is for "Striped", "Plaided" and "None" classes. If "None" class is chosen from Procedure I, then we use Procedure II that is used for "Speckled" and "None" classes.

Algorithm IV.a: Nearest Neighbor classification method for horizontal, vertical, 45-degree and 135-degree lines classes

```
1. For each image that will be classified do
2. Begin
3.     For each one of five classes as "Horizontal", "Vertical",
       "45-Degree", "135-Degree" and "None" classes  do
4.     Begin
5.        Capture the image's four line features.
6.        For each images of the five classes do
7.        Begin
8.          Find the distance between the image that will be
            classified and the images of the class by Manhattan
            Distance metric.
9.          Choose minimum distance for each class as horizontal,
            vertical,45,135 and none minimum.
10.       End of "for" loop 6
11.    End of "for" loop 3
12.  End of "for" loop 1
  // Until now, we have horizontal, vertical, 45, 135 and
   none minimum for five  manually constructed classes.
13. Calculate the smallest minimum distance of above
    minimum distances of the
14. Classes. Assume that it is horizontal minimum.
15. So, assign the image to "Horizontal" class.
```

Algorithm IV.b: Nearest Neighbor classification method for "Striped", "Plaided", "Speckled" classes

```
1. For each image that will be classified
2. Begin
3.  For each one of four classes as "Striped", "Plaided",
    "Speckled" and "None" classes do
4.  Begin
5.     Capture the image's four line and one circle features.
6.     For every image of the four classes
7.     Begin
         Find the distance between the image, which will be
         classified, and the images of the chosen class by
         Manhattan distance metric.
  //"Striped" and "Plaided" classes used four dimensional feature
    space.
  //"Speckled" class used one-dimensional feature space for the
    computing the distance.
  //"None" class used both feature spaces.
8.    End of "For" loop 6
9.  End of "For" loop 5
  //There are two procedures for the classification: one is for
    "Striped", "Plaided" and "None" classes.If "None" class is
    chosen from above, then we use the second procedure that is
    used for "Speckled" and "None" classes.
```

Procedure I:

```
1. Find the minimum values of Minimum Sriped, Minimum Plaided and
   Minimum Line None.
2. If the minimum value is for Minimum Sriped then
3.   Assign the image to "Striped" class.
4. Else if the minimum is Minimum Plaided then
5.   Assign the image to "Plaided" class.
6. Else if the minimum is Minimum Line None then
7.    Apply the Procedure II.
```

Procedure II:

```
1. Find the minimum values of Minimum Speckled and Minimum Circle
   None.
2. If the minimum is Minimum Speckled then
3.   Assign the image to "Speckled" class.
4. Else
5.   Assign the image to "None" class.
```

5.7 The algorithm of *k*-Nearest Neighbor classification method

This algorithm is quite similar to the algorithm of Nearest Neighbor classification. Algorithm V.a shows the algorithm of *k*-Nearest Neighbor classification method for horizontal, vertical, 45-degree and 135-degree lines classes. We have four line features from line histogram, which consists of number of horizontal, vertical, 45-degree and 135-degree lines for each image, and we have five training classes which have been manually generated with images. To classify an image, its distance to the all images of the training classes is computed by Manhattan distance metric. We choose the *k* minimum distances in each class, and find only one minimum distance by the addition of *k* minimum distances for each class. The image is classified to the class, which is the nearest to the image.

Algorithm V.b shows the algorithm of *k*-Nearest Neighbor classification method for "Striped", "Plaided", "Speckled" and "None" classes that are manually generated. We have four line features and one circle feature to classify images. The distances among the image, which will be classified, and the images in the training classes are calculated by Manhattan distance metric. We find a minimum distance by addition of *k* minimum distances for each class, and then we apply above procedures. There are two procedures for the classification. One is for "Striped", "Plaided" and "None" classes. If "None" class is chosen from Procedure I, then we use Procedure II that is used for "Speckled" and "None" classes.

Algorithm V.a: *k*-Nearest Neighbor classification method for horizontal, vertical, 45-degree and 135-degree lines classes

```
1. Decide the k value.
2. For each image that will be classified
3. Begin
4.    For each one of the five classes as horizontal, vertical,
      45-degree,135-degree and none classes do
5.    Begin
6.        Capture the image's four line features.
7.        For every image of the five classes
8.        Begin
9.          Find the distances between the image tha will be
            classified and the images if the classes by Manhattan
            distance metric.
10.         Choose the k minimum distances for each class:
            Minimum Horizontal[1],..., Minimum Horizontal[k],
            Minimum Vertical[1],..., Minimum Vertical[k],
            Minimum 45[1],...,Minimum 45[k],Minimum 135[1],...,
            Minimum 135[k], and Minimum None[1],...,Minimum
            None[k].
11.   End of "For" loop 4
12. End of "For" loop 2
13. For every class do
14. Begin
15. Generate only one distance with the addition of the
     distances from the index 1 to the index k. For example,
     Minimum Horizontal from Minimum Horizontal[1] ,...,  Minimum
     Horizontal[k]
16. End of "For" loop 13
17. Calculate the minimum distance of above minimum distances of
    the classes.
18. Assume that the minimum distance is Minimum Horizontal then
     assign the image to the "Horizontal" class.
```

Algorithm V.b: *k*-Nearest Neighbor classification method for "Striped", "Plaided", "Speckled" classes

```
1. Decide k value.
2. For each image which will be classified do
3. Begin
4.     For each one of four classes as "Striped", "Plaided",
       "Speckled" and "None" classes do
5.     Begin
6.         Find the k minimum distances between the image that will
           be classified and the  images in each one of four classes
           by Manhattan distance metric.
  // "Striped" and "Plaided" classes use 4-dimensional line
     feature space.
  // "Speckled" class uses 1-dimensional line feature space for
     the calculation of the minimum distance.
7.    End of "For" loop 4
8. End of "For" loop 2
9. For each one of  "Striped", "Plaided", "Speckled" and "None"
   classes do
10. Begin
11.   Generate only one distance with the addition of the
       distances from the index 1 to the index k.
  // Assume that Minimum Striped for Minimum Striped [1],..,
     Minimum Striped [k].
  // Assume that Minimum Striped for Minimum Plaided[1],..,
     Minimum Plaided [k].
  // Assume that Minimum Striped for Minimum Speckled [1],..,
     Minimum Speckled [k].
  // Assume that Minimum Striped for Minimum None [1],..,
     Minimum None [k].
12. End of "For" loop 9
  // Now, we have minimum values. After that, for the
     classification there are two procedures that are used in the
     Nearest Neighbor method.
13. Apply Procedure I and II in the Nearest Neighbor method.
```

5.8 The algorithm of Minimum Distance classification method

Minimum distance classification is based on the statistical values of the classes. Mean value of a class is important for this classification.

Algorithm VI.a shows the algorithm of Minimum Distance classification method for horizontal, vertical, 45-degree and 135-degree lines classes. We have four line features from line histogram, which consists of the number of horizontal, vertical, 45-degree and 135-degree lines for each image, and we have five classes, which have been manually generated. We have four- feature space. We have the mean of the class by calculating the means of four dimensions of each class. We compute the distance between the image, which will be classified, and the mean of the class by Euclidean distance metric. We assign the image to the class, which has the minimum distance.

Algorithm VI.b shows the algorithm of Minimum Distance classification method for "Striped", "Plaided", and "Speckled" classes. We have four line features, one circle feature and four classes, which are manually generated. After computing the mean of the classes, we first classify the image to the "Striped", "Plaided" and "None" using line mean values of these classes. If the image is in "Striped" or "Plaided" class, the classification process is finished. Otherwise, the process continues. We check the minimum distances of "Speckled" and "None" classes using circle mean values. The image is classified into "Speckled" class if it has the minimum distance.

Algorithm VI.a: Minimum Distance classification method for horizontal, vertical, 45-degree and 135-degree lines classes

```
1. For each one of five classes do
2. Begin
3.     For every image of the class do
4.     Begin
5.        Calculate the mean of the class. We have four-dimension
```

feature space. Calculate the mean for each dimension, then we have the mean of the class with four dimensions. These mean values are Horizontal Class Mean1, Horizontal Class Mean2, Horizontal Class Mean3 and Horizontal Class Mean4.

6. end of for loop 3
7. End of "For" loop 1
8. For each image that will be classified do
9. Begin
10. Capture the image's four line features as horizontal, vertical, 45 degree and 135 degree line features.
11. For every image of the five classes do
12. Begin
13. Calculate the distance between the image that will be classified and the mean of the class by Euclidean distance metric.
    // These areDistanceOfImageToHorizontal, DistanceOfImageToVertical, DistanceOfImageTo45Degree, DistanceOfImageTo135Degree and DistanceOfImageToNone.
14. End of "For" loop 11
15. End of "For" loop 9
16. Find the minimum of above five distances for classification.
17. Assign the image to the class, which has the minimum distance.

Algorithm VI.b: Minimum Distance classification method for "Striped", "Plaided", "Speckled" classes

1. For each one of four classes do
2. Begin
3. For every image of the class do
4. Begin
5. Calculate the mean of the class. For "Plaided" and "Striped" classes, there is four-dimensional line feature space. Calculate the mean of the class for each one of four dimensions. For "Speckled" class, there is one-dimensional circle feature space. Calculate the mean of the class for only this dimension. For "None" class, calculate two mean values for both four line features and one circle feature.
6. End of "For" loop 3

```
7. For each image that will be classified do
8. Begin
9.     Capture the images four line features and on circle feature.
10.     For each one of  "Striped", "Plaided" and "None" classes do
11.       Find the minimum distance between the image and the mean
          of  the class by Euclidea distance metric.
12.       If the minimum distance is not for the "None" class then
13.         Assign the image to the class, which has the minimum
            distance.
14.       Else
15.       Begin
16.       For each one of  "Speckled" and "None" classes do
17.       Begin
18.         Find the minimum distance between the image and the
            mean of the class by using Euclidean distance metric.
19.         If the minimum distance is for "Speckled" class then
20.           Assign the image to "Speckled" class.
21.         Else
22.           Assign the image to "None" class.
23.       End of "For" loop 16
24.       End of "Else" condition 14
25.   End of "For" loop 7
```

5.9 Combination of Manual Thresholding with other methods

In this study, Manual Thresholding is combined with Nearest Neighbor, k-Nearest Neighbor and Minimum Distance classifications. Line and circle threshold values bound the circle and line feature spaces by using Manual Thresholding, and then the other classification method decides the image's class.

CHAPTER SIX
EXPERIMENTS OF THE SYSTEM

Recall, precision and F-measure are well-known standard measures to evaluate a system in information retrieval. Recall is the ratio of relevant documents retrieved for a given query over the number of relevant documents for that query in the database, and precision is the ratio of the number of relevant documents retrieved over the total number of documents retrieved.(Frakes & Baeza-Yates, 1992) Recall is the ability to retrieve relevant documents, and precision is the ability to reject nonrelevant documents.(Alpkoçak, 2002)

We have adapted recall and precision with image classification terms. So, Recall is defined as the number of relevant images classified divided by the total number of relevant images in the image collection. For example, suppose there are 80 images relevant to striped images in the image collection. System X classifies 60 images into striped class, but 40 of 60 classified images are about striped images. Then X's recall is 40/80 = 50 %. Precision is defined as the number of relevant images classified divided by the total number of images classified. For example, suppose there are 80 images relevant to striped images in the image collection. System X classifies 60 images into striped class, but 40 of 60 classified images are about striped images. Then X's precision is 40/60 = 67 %. F-measure is to combine recall and precision into single measure of overall performance. Assume that C is the number of relevant images classified, T is the total number of relevant images in the image collection, and R is the total number of images classified. Figure 6.1 shows these sets with diagrams. Recall, precision and F-measure are defined as the followings.

$$\text{Recall} = C/_{T} \quad \text{Precision} = C/_{R} \quad \text{F - measure} = 2 \text{ x } \mathit{Recall} \text{ x } \mathit{Precision}/_{\mathit{Recall} + \mathit{Precisi}}$$

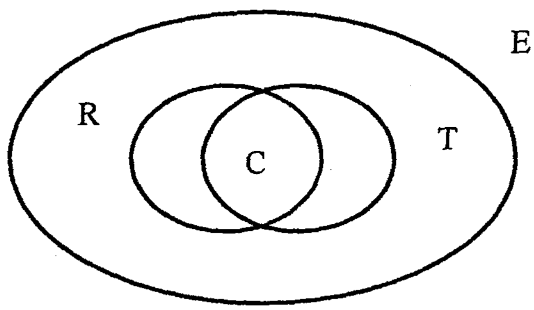


Figure 6.1 An example of recall and precision. E is the number of test images. R is the number of classified images. T is the number of relevant images. C is the number of relevant classified images.

6.1 Manual Thresholding classification

For the evaluation of the system, we have used 200 test images, which are selected manually, from 3500 textile images. There are 20 striped images, 20 plaided images and 20 speckled images, which are checked manually, in 200 test images.

In Table 6.1, Table 6.2 and Table 6.3, the experiment results of Manual Thresholding classification have been shown for some threshold values. For the evaluation of the system with manual thresholding, according to 'Recall', when the Line and Circle Threshold is low, better results have been taken. For example, for Line Threshold > 0 and Circle Threshold > 0 values, the result is better for recall. According to 'Precision', when the thresholds increase, the precision results are better. Generally, the results are very good for Manual Thresholding method. Only the precision of speckled class is average. According to F-measure and the thresholds of Manual Thresholding, the best result for striped class is 89.3 % and the method is Manual Thresholding with Line Threshold > 5. The best result for plaided class is 92.0 % and the method is Manual Thresholding with Line Threshold > 5. The best result for speckled class is 72.6 % and the method is Manual Thresholding with Circle Threshold > 0.

Table 6.1 Manual Thresholding classification with Line Threshold > 0 and Circle Threshold > 0

	R	T	C	Recall(%)	Precision(%)	F-masure(%)
Striped Class	43	20	18	90	41	56.3
Plaided Class	28	20	20	100	71	83.0
Speckled Class	35	20	20	100	57	72.6

Table 6.2 Manual Thresholding classification with Line Threshold > 5 and Circle Threshold > 5

	R	T	C	Recall(%)	Precision(%)	F-masure(%)
Striped Class	18	20	17	85	94	89.3
Plaided Class	19	20	18	90	94	92.0
Speckled Class	32	20	18	90	56	69.0

Table 6.3 Manual Thresholding classification with Line Threshold > 10 and Circle Threshold > 10

	R	T	C	Recall(%)	Precision(%)	F-masure(%)
Striped Class	15	20	14	70	93	79.9
Plaided Class	15	20	14	70	93	79.9
Speckled Class	30	20	17	85	56	67.5

6.2 Nearest Neighbor classification

We have used some training classes for this classification. Striped class has 65 images; speckled class has 15 images; plaided class has 24 images, NonSPS class (This class does not include any striped, plaided and speckled images.) has 24 images for training. The images of these training classes are selected manually.

For the evaluation of the system, we have used 200 test images, which are selected manually, from 3500 textile images. There are 20 striped images, 20 plaided images and 20 speckled images, which are checked manually.

Table 6.4 Nearest Neighbor classification

	R	T	C	Recall(%)	Precision(%)	F-masure(%)
Striped Class	25	20	18	90	72	80.0
Plaided Class	20	20	19	95	95	95.0
Speckled Class	34	20	17	85	50	63.0

Table 6.5 Nearest Neighbor classification with Manual Thresholding classification

	R	T	C	Recall(%)	Precision(%)	F-masure(%)
Striped Class	18	20	18	90	100	94.7
Plaided Class	20	20	19	95	95	95.0
Speckled Class	34	20	17	85	50	63.0

In Table 6.4 and Table 6.5, the experiment results of Nearest Neighbor classification and Nearest Neighbor classification with Manual Thresholding have been shown. For the evaluation of the system with Nearest Neighbor, according to 'Recall', when Manual Thresholding is used, the results have not changed. According to 'Precision', when Manual Thresholding is used, the precision results are better especially for striped class. Generally, the results are very good for Nearest Neighbor method. Only the precision of speckled class is low.

6.3 *k*-Nearest Neighbor classification

We have used some training classes for this classification. Striped class has 65 images; speckled class has 15 images; plaided class has 24 images, NonSPS class (This class does not include any striped, plaided and speckled images.) has 24 images for training. The images of these training classes are selected manually.

For the evaluation of the system, we have used 200 test images, which are selected manually, from 3500 textile images. There are 20 striped images, 20 plaided images and 20 speckled images, which are checked manually.

Table 6.6 *k*-Nearest Neighbor classification and *k*=3

	R	T	C	Recall(%)	Precision(%)	F-masure(%)
Striped Class	23	20	17	85	74	79.1
Plaided Class	22	20	19	95	86	90.3
Speckled Class	35	20	19	95	54	68.9

Table 6.7 *k*-Nearest Neighbor classification and *k*=5

	R	T	C	Recall(%)	Precision(%)	F-masure(%)
Striped Class	24	20	18	90	75	81.8
Plaided Class	22	20	19	95	86	90.3
Speckled Class	33	20	19	95	57	71.2

Table 6.8 *k*-Nearest Neighbor classification and *k*=7

	R	T	C	Recall(%)	Precision(%)	F-masure(%)
Striped Class	25	20	18	90	72	80.0
Plaided Class	21	20	18	90	85	87.4
Speckled Class	35	20	19	95	54	68.9

Table 6.9 *k*-Nearest Neighbor classification and *k*=11

	R	T	C	Recall(%)	Precision(%)	F-masure(%)
Striped Class	25	20	18	90	72	80.0
Plaided Class	21	20	18	90	85	87.4
Speckled Class	35	20	19	95	54	68.9

Table 6.10 *k*-Nearest Neighbor classification with Manual Thresholding classification and *k*=3

	R	T	C	Recall(%)	Precision(%)	F-masure(%)
Striped Class	17	20	17	85	100	91.9
Plaided Class	21	20	18	90	85	87.4
Speckled Class	35	20	19	95	54	68.9

Table 6.11 *k*-Nearest Neighbor classification with Manual Thresholding classification and *k*=5

	R	T	C	Recall(%)	Precision(%)	F-masure(%)
Striped Class	18	20	18	90	100	94.7
Plaided Class	21	20	18	90	85	87.4
Speckled Class	35	20	19	95	54	68.9

Table 6.12 *k*-Nearest Neighbor classification with Manual Thresholding classification and *k*=7

	R	T	C	Recall(%)	Precision(%)	F-masure(%)
Striped Class	18	20	18	90	100	94.7
Plaided Class	20	20	18	90	90	90.0
Speckled Class	35	20	19	95	54	68.9

Table 6.13 *k*-Nearest Neighbor classification with Manual Thresholding classification and *k*=11

	R	T	C	Recall(%)	Precision(%)	F-masure(%)
Striped Class	18	20	18	90	100	94.7
Plaided Class	20	20	18	90	90	90.0
Speckled Class	35	20	19	95	54	68.9

In Table 6.6, Table 6.7, Table 6.8, Table 6.9, Table 6.10, Table 6.11, Table 6.12 and Table 6.13, the experiment results of *k*-Nearest Neighbor classification and *k*-Nearest Neighbor classification with Manual Thresholding have been shown for some *k* values. For the evaluation of the system with *k*-Nearest Neighbor, according to 'Recall', when Manual Thresholding is used, the results have not changed. When *k* values increase, the recall results are equal or better than lower *k* values.

According to 'Precision', when Manual Thresholding is used, the precision results are better especially for striped class. When *k* values increase, the recall results are equal or better than lower *k* values. Generally, the results are very good for *k*-Nearest Neighbor method. However, the precision of speckled class is average.

According to F-measure and *k* values of *k*-Nearest Neighbor, the best result for striped class is 94.7 % and the methods are *k*-Nearest Neighbor with Manual Thresholding and *k*=5, *k*=7 and *k*=11. The best result for plaided class is 90.3 % and the methods are *k*-Nearest Neighbor with *k*=3 and *k*=5. The best result for speckled class is 71.2 % and the method is *k*-Nearest Neighbor with *k*=5.

6.4 Minimum Distance classification

We used some training classes for this classification. Striped class has 65 images; speckled class has 15 images; plaided class has 24 images, NonSPS class (This class does not include any striped, plaided and speckled images.) has 24 images for training. The images of these training classes are selected manually.

For the evaluation of the system, we have used 200 test images, which are selected manually, from 3500 textile images. There are 20 striped images, 20 plaided images and 20 speckled images, which are checked manually.

Table 6.14 Minimum Distance classification

	R	T	C	Recall(%)	Precision(%)	F-masure(%)
Striped Class	48	20	17	85	35	49.6
Plaided Class	8	20	8	40	100	57.1
Speckled Class	21	20	14	70	67	68.5

Table 6.15 Minimum Distance classification with Manual Thresholding classification

	R	T	C	Recall(%)	Precision(%)	F-masure(%)
Striped Class	25	20	17	85	68	75.6
Plaided Class	8	20	8	40	100	57.1
Speckled Class	21	20	14	70	67	68.5

In Table 6.14 and Table 6.15, the experiment results of Minimum Distance Classification and Minimum Distance classification with Manual Thresholding have been shown. For the evaluation of the system with Minimum Distance method, according to 'Recall', when Manual Thresholding is used, the recall results have not changed. According to 'Precision', when Manual Thresholding is used, the precision result is better for striped class, but the other precision results have not changed. Generally it can be said that the results are quite good, but the recall result is bad for plaided class.

6.5 Evaluation of the system

F-measure is to combine recall and precision into single measure of overall performance. If we choose F-measure to compare all of the methods used in this study, we can find the best method for each class. The best result for striped class is 94.7 % and the methods are Nearest Neighbor with Manual Thresholding and *k*-Nearest Neighbor with Manual Thresholding and *k*=5, *k*=7 and *k*=11. The best result for plaided class is 95.0 % and the methods are Nearest Neighbor and Nearest Neighbor with Manual Thresholding. The best result for speckled class is 72.6 % and the method is Manual Thresholding with Line and Circle Threshold > 0.

The worst result for striped class is 49.6 % and the method is Minimum Distance. The worst result for plaided class is 57.1 % and the methods are Minimum Distance and Minimum Distance with Manual Thresholding. The worst result for speckled class is 63.0 % and the methods are Nearest Neighbor and Nearest Neighbor with Manual Thresholding.

The methods, which have the best and the worst recall values, and the best and the worst precision values according to the striped, plaided and speckled classes, are shown in this section.

The methods, which have the best recall value (90%) for striped class: Manual Thresholding with line threshold >0 and circle threshold >0, Nearest Neighbor, Nearest Neighbor with Manual Thresholding, *k*- Nearest Neighbor with k=5,7,11, *k*-Nearest Neighbor with Manual Thresholding and k=5,7,11.

The methods, which have the best recall value (100%) for plaided class: Manual Thresholding with line threshold >0 and circle threshold >0.

The methods, which have the best recall value (100%) for speckled class: Manual Thresholding with line threshold >0 and circle threshold >0.

The methods, which have the best precision value (100%) for striped class: Nearest Neighbor with Manual Thresholding, *k*-Nearest Neighbor with Manual Thresholding and *k*=3,5,7,11.

The methods, which have the best precision value (100%) for plaided class: Minimum Distance, Minimum Distance with Manual Thresholding.

The methods, which have the best precision value (67%) for speckled class: Minimum Distance, Minimum Distance with Manual Thresholding.

The methods, which have the worst recall value (70%) for striped class: Manual Thresholding with line threshold >10 and circle threshold >10

The methods, which have the worst recall value (40%) for plaided class: Minimum Distance, Minimum Distance with Manual Thresholding.

The methods, which have the worst recall value (70%) for speckled class: Minimum Distance, Minimum Distance with Manual Thresholding.

The methods, which have the worst precision value (35%) for striped class: Minimum Distance.

The methods, which have the worst precision value (71%) for plaided class: Manual Thresholding with line threshold >0 and circle threshold >0.

The methods, which have the worst precision value (50%) for speckled class: Nearest Neighbor, Nearest Neighbor with Manual Thresholding.

CHAPTER SIX
CONCLUSION

In this study, we have classified textile images into striped, plaided and speckled image classes. In addition, striped images are categorized into the classes of horizontal, vertical, 45-degree and 135-degree lines. Line detection operator and Hough transform is used for the line and circle feature extraction respectively. Manual Thresholding, Nearest Neighbor, *k*-Nearest Neighbor and Minimum Distance methods have been used for the textile image classification in this study.

The classification methods, which have the best and the worst recall values, and the best and the worst precision values according to the striped, plaided and speckled classes, are shown in Section 6.5. When evaluating the results in the Table 7.1 for recall, precision and F-measure, which combines recall and precision into single measure, the system is quite successful. This study has provided to access the striped, plaided and speckled images from textile images rapidly.

Table 7.1 The best and the worst values for the system

	Recall(%)		Precision(%)		F-measure(%)	
	The Best	The Worst	The Best	The Worst	The Best	The Worst
Striped Class	90	70	100	35	94.7	49.6
Plaided Class	100	40	100	71	95.0	57.1
Speckled Class	100	70	67	50	72.6	63.0

We have chosen F-measure to compare all of the methods used in this study because F-measure is to combine recall and precision into single measure of overall performance, then we can find the best method for each class. The best result for striped class is 94.7 % and the methods are Nearest Neighbor with Manual Thresholding and k-Nearest Neighbor with Manual Thresholding and k=5, k=7 and k=11. The best result for plaided class is 95.0 % and the methods are Nearest Neighbor and Nearest Neighbor with Manual Thresholding. The best result for speckled class is 72.6 % and the method is Manual Thresholding with Line and Circle Threshold > 0.

In textile image classification, other classification methods can be used for the future work. Other feature extraction methods as Extended Hough transform can be applied for line and circle features of the images. Other features as ellipses can be chosen for the image classification. This feature can be useful to classify flower images.

REFERENCES

Akkus, A., & Guvenir, H. A. (1996). K-nearest Neighbor Classification on Feature Projections. In Proceedings of ICMI'96 Lorenzo Saitta (ED.), Morgan Kaufmann, Bari, Italy, 12-19

Alpkoçak, A. (2002). Lecture Notes of Information Retrieval. Dokuz Eylül University, Computer Engineering, İzmir.

Clavier, E., Clavier, S., & Labiche, J. (1998). Image Sorting - Image Classification: A Global Approach. Proceedings of the Fifth International Conference on Document Analysis and Recognition, 1998 Institute of Electrical and Electronics Engineers, Session 5: Mo-3A: Poster Session I

Faloutsos, C. (1996). Searching Multimedia Databases by Content. Kluwer Academic Publishers; ISBN0-7923-9777-0.

Frakes, W. B., & Baeza-Yates, R. (1992). Information Retrieval: Data Structures and Algorithms. Prentice Hall. ISBN 0-13-463837-9

Gevers, T., Aldershoff, F., & Smeulders, A. W. M. (2000, January). Classification of Images on Internet by Visual and Textual Information. In SPIE, San Jose.

Greenshields, I. R., & Yang Z. (2000). A Multigrid Approach to the Gibbsian Classification of Mammograms. Proceedings of the 13th IEEE Symposium on Computer Based Medical Systems (CBMS'00), pp. 163-169.

Hirata, K., Mukherjea, S., Li,W., & Hara, Y. (1998). Integrating Image Matching and Classification for Multimedia Retrieval on the Web. Proceedings of the IEEE International Conference on Multimedia Computing and Systems, 1 , Institute of Electrical and Electronics Engineers, Inc., Oral Session D3: Web-Search of Single Images.

Hu, J., Kashi, R., & Wilfong, G. (1999, September 1-3). Document Classification Using Layout Analysis. IEEE Computer Society, pp.556-560.

Jaimes, A., & Chang, S.-F. (2000, January 8-11). Integrating Multiple Classifiers in Visual Object Detectors Learned from User Input. 4th Asian Conference on Computer Vision (ACCV 2000), Taipei, Taiwan. Invited paper, session on Image and Video Databases.

Jaimes, A., & Chang, S.-F. (2000, January). A Conceptual Framework for Indexing Visual Information at Multiple Levels. IS&T/SPIE Internet Imaging 2000, San Jose, CA.

Jain, A. K, & Dubes, R. C. (1998). Algorithms for Clustering Data. Prentice Hall, Inc. Englewood Cliffs, New Jersey 07632 pp. 14-15 ISBN 0-13-022278-X

Maeda, S., Mukunoki, M., & Ikeda, K. (1998). A Classification Method of Images Based on Composition and Its Application to Image Retrieval. Proceedings of the IEEE International Conference on Multimedia Computing and Systems, 2, Institute of Electrical and Electronics Engineers, Inc.; Poster Session III.

Mao, J., & Jain, A. K. (1992). Texture Classification and Segmentation Using Multiresolution Simultaneous Autoregressive Models. Pattern Recognition, 25(2) pp. 173-188.

Picard, R. W., Kabir, T., & Liu, F. (1993, June). Real-time Recognition With the Entire Brodatz Texture Database. In Proc. IEEE Conf. On Computer Vision and Pattern Recognition. pp. 638-639. New York. MIT Lab Perceptual Computing.

Prabhakaran, B. (1997). Multimedia Database Management Systems. Kluwer Academic Publishers. ISBN 0-7923-9784-3.

Puzicha, J., Buhmann, J. M., Rubner, Y., & Tomasi, C. (1998). Empirical Evaluation of Dissimilarity Measures for Color and Texture. Proceedings of the International Conference on Computer Vision, Institute of Electrical and Electronics Engineers, Inc. Matching, Recognition, and Indexing.

Rhee, I., & Joshi, S. R. (1999, June 7-11). FEC-Based Loss Recovery for Interactive Video Transmission. IEEE Computer Society, 1, 250-256.

Ricci, F., & Avesani, P. (1999, April). Data Compression and Local Metrics for Nearest Neighbor Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, No. 4.

Schauble, P. (1997). Multimedia Information Retrieval, Content Based Information Retrieval from Large Text and Audio Databases. Zurich : Switzerland, by Kluver Academic Publishers. ISBN 0-7923-9899-8.

Smith, J.R., & Chang, S.-F. (1997, August). Multi-stage Classification of Images From Features and Related Text. In 4th Europe EDLOS Workshop, San Miniato, Italy.

Szummer, M., & Picard, R. (1998). Indoor-Outdoor Image Classification. In Workshop in Content-based Access to Image and Video Databases, Bombay, India.

Yue, L. & Guo, H. (1997, Oktober). Texture Image Retrieval by Universal Classification for Wavelet Transform Coefficients. Proc.IEEE Int. Conf. on Image Processing.

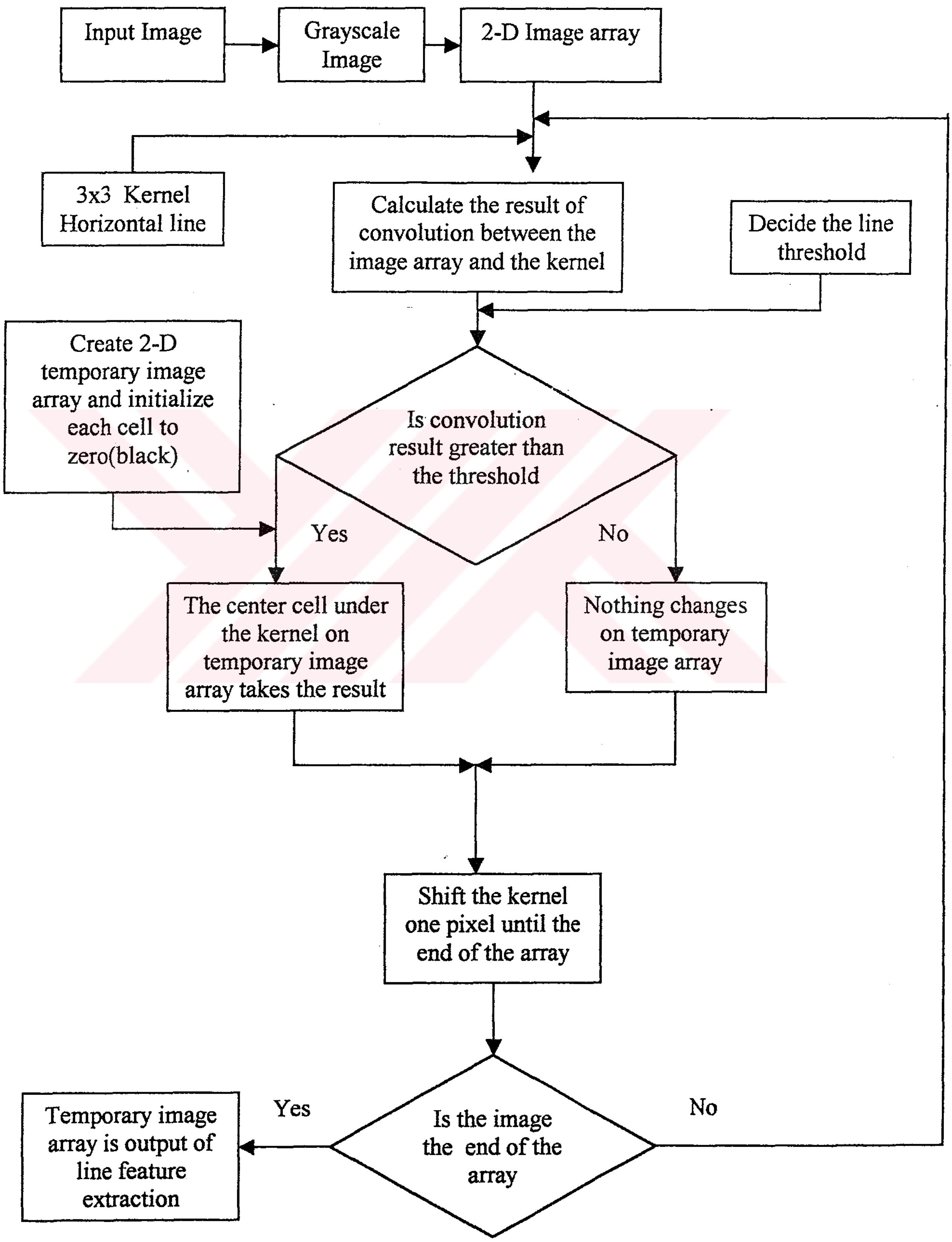
Yue, L., & Guo, H. (1997). Texture Image Retrieval by Universal Classification for Wavelet Transform Coefficients. Proceedings of the 1997 International Conference on Image Processing (ICIP '97). 1997 Institute of Electrical and Electronics Engineers, Inc.

Zhao, W. (1999). Bayesian based Optimal Nearest Neighbor Metrics. CVPR'99, Center for Automation Research University of Marryland. College Park, MD 20742-3275.

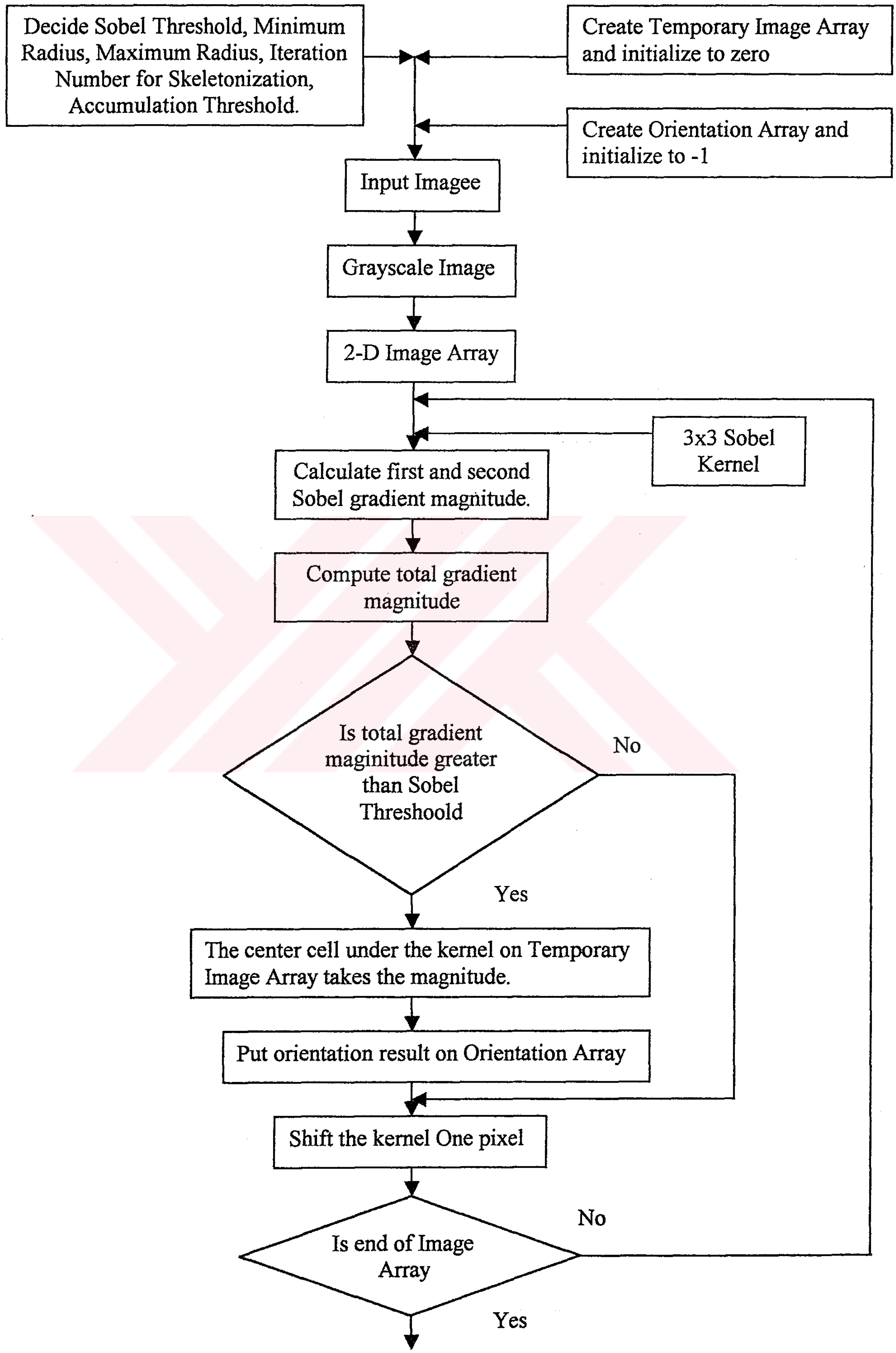
APPENDIX A

FLOW DIAGRAMS

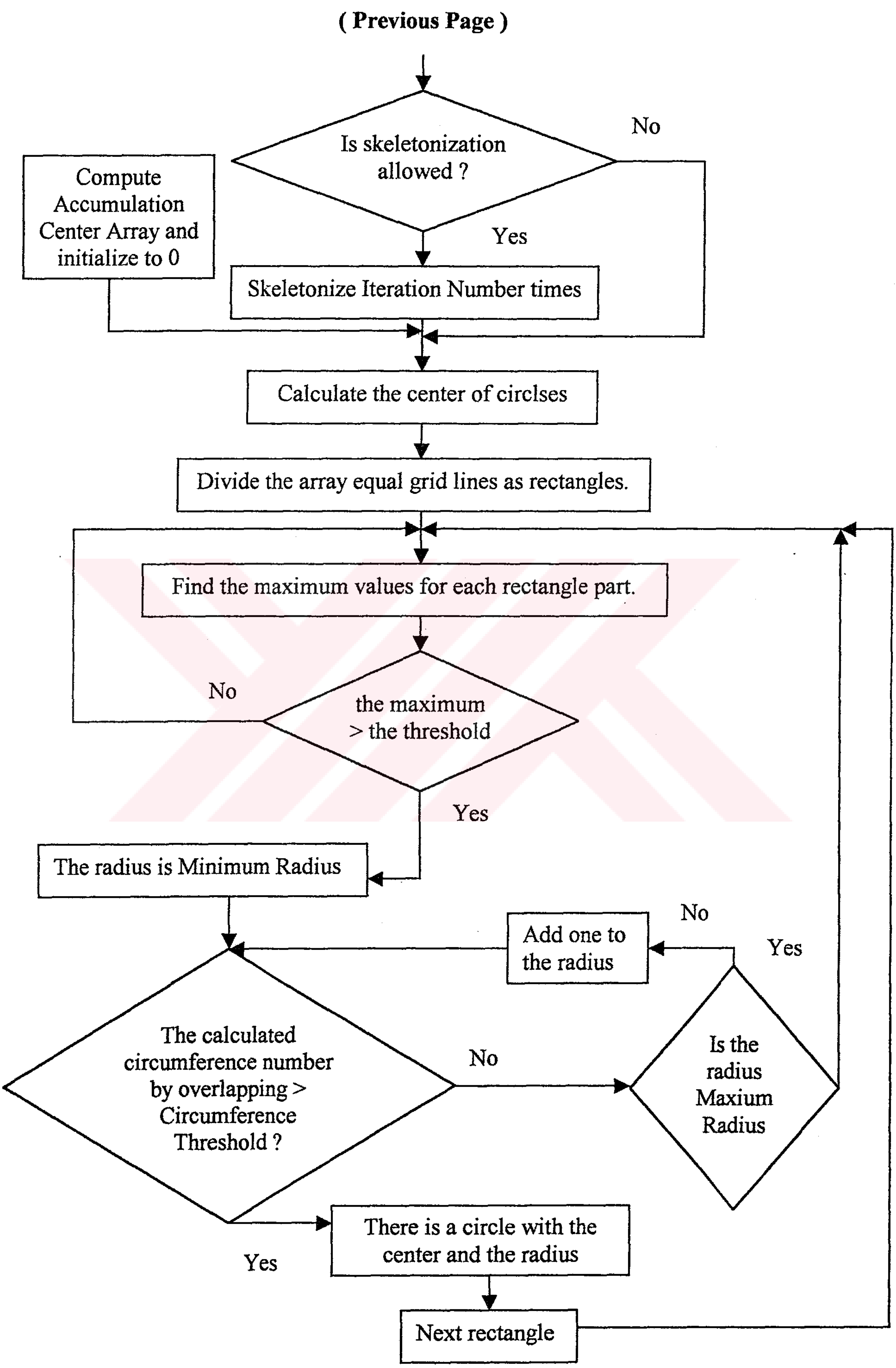
Horizontal Line Detection Flow Diagram



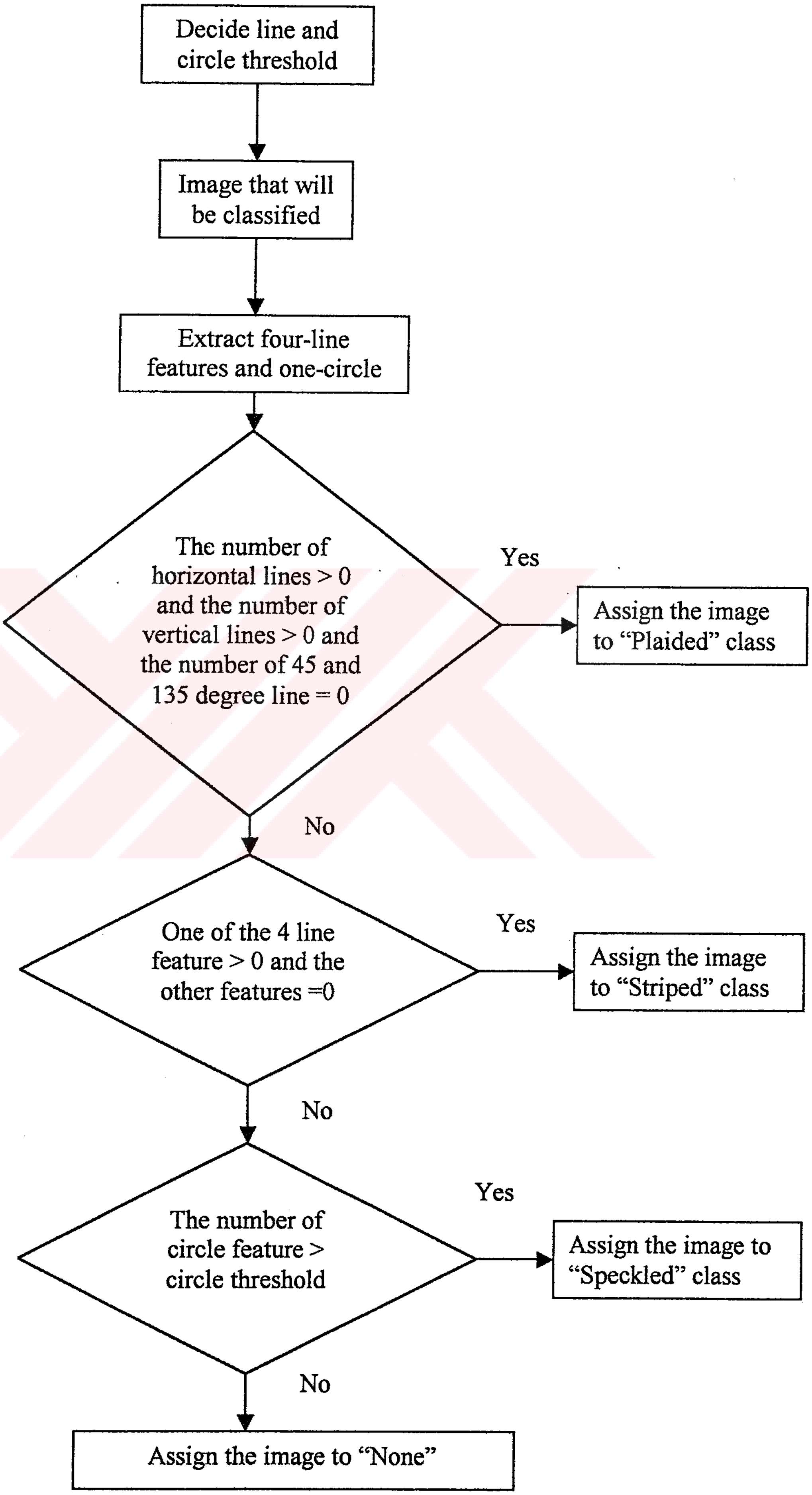
Circle Detection Flow Diagram



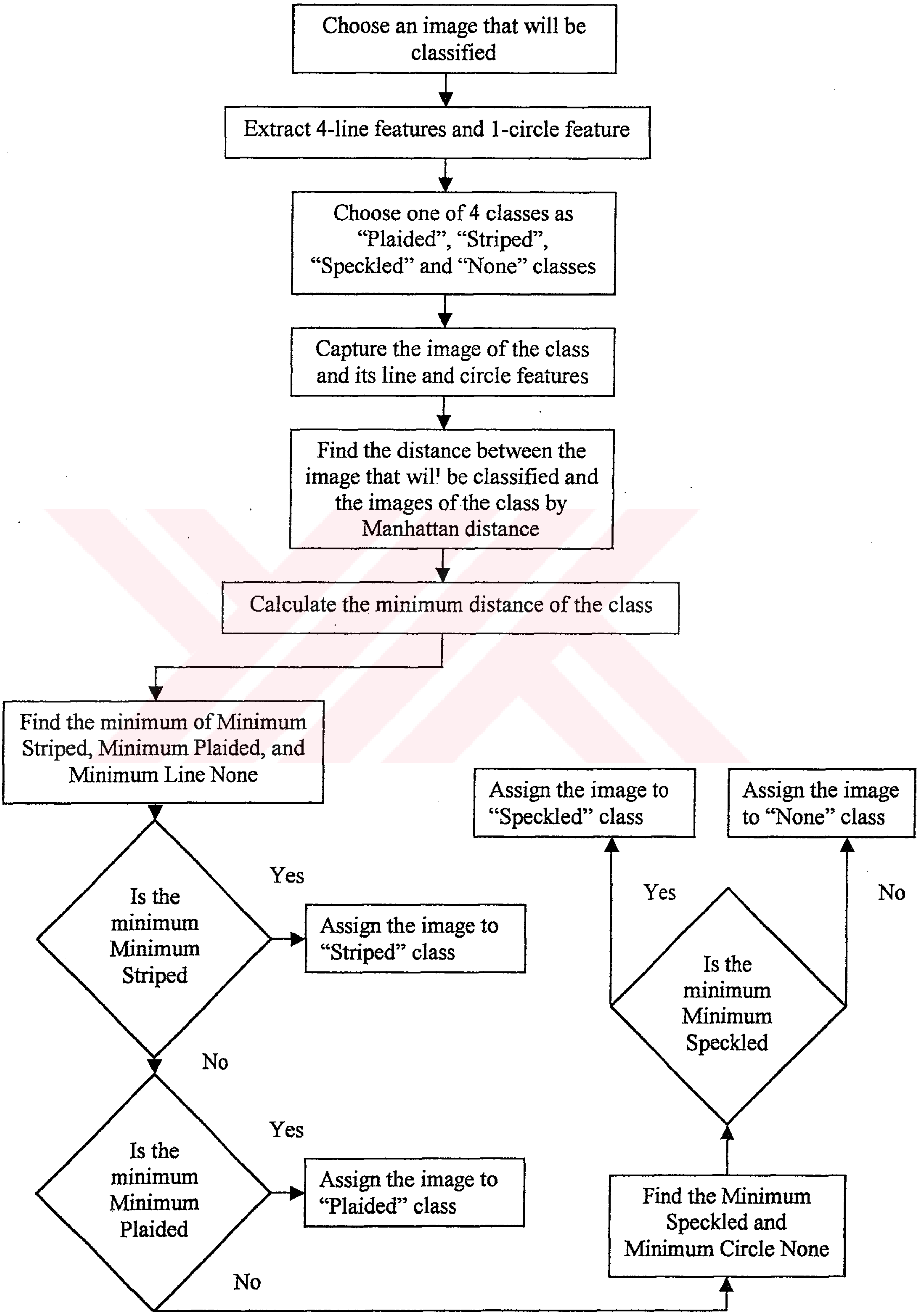
(Continue on Next Page)


(Previous Page)
Is skeletonization allowed ?
No
Yes
Compute Accumulation Center Array and initialize to 0
Skeletonize Iteration Number times
Calculate the center of circlses
Divide the array equal grid lines as rectangles.
Find the maximum values for each rectangle part.
No
the maximum > the threshold
Yes
The radius is Minimum Radius
Add one to the radius
No
Yes
The calculated circumference number by overlapping > Circumference Threshold ?
No
Is the radius Maxium Radius
Yes
There is a circle with the center and the radius
Next rectangle

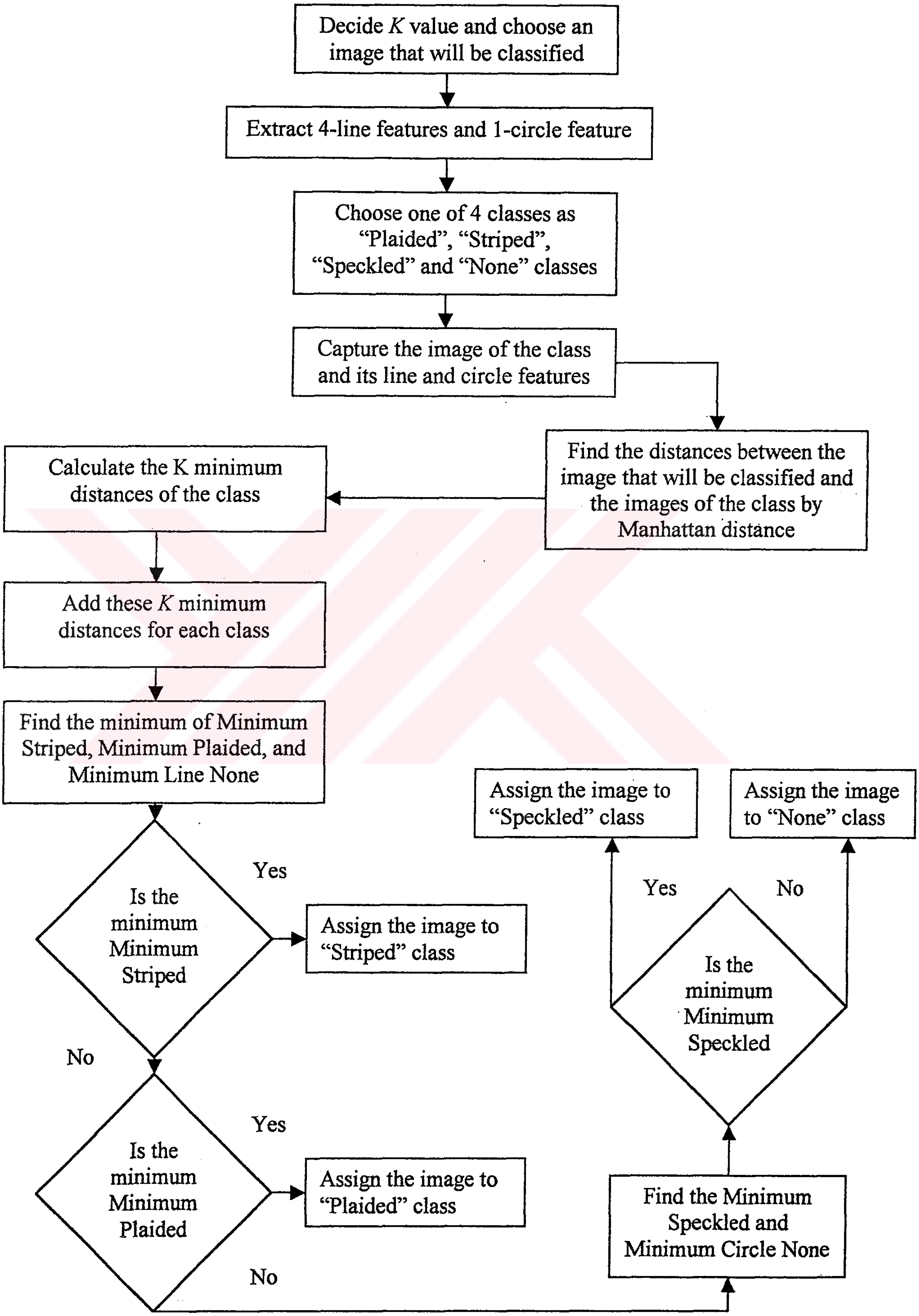
Manual Thresholding Classification Flow Diagram



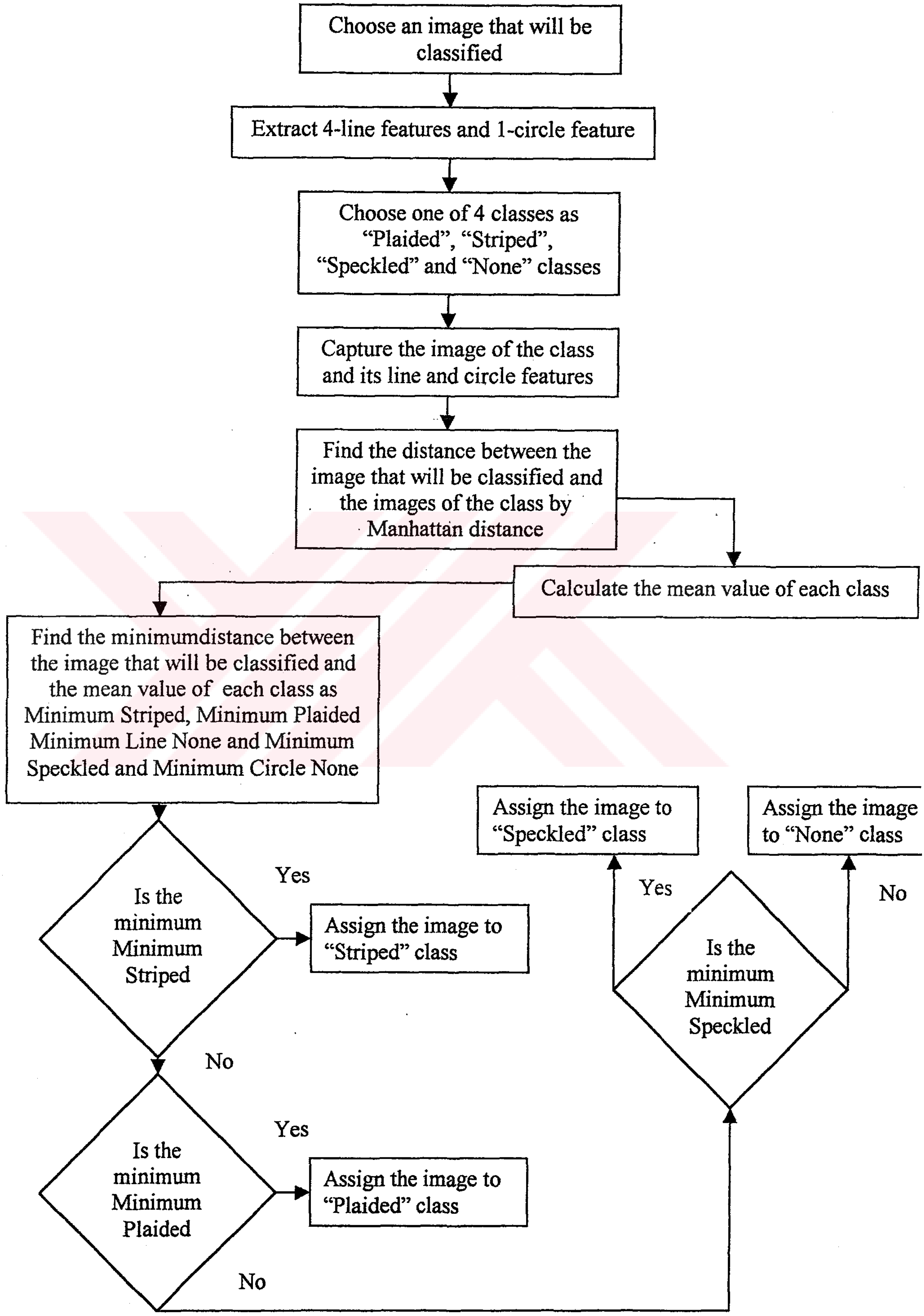
Nearest Neighbor Classification Flow Diagram



k-Nearest Neighbor Classification Flow Diagram



Minimum Distance Classification Flow Diagram

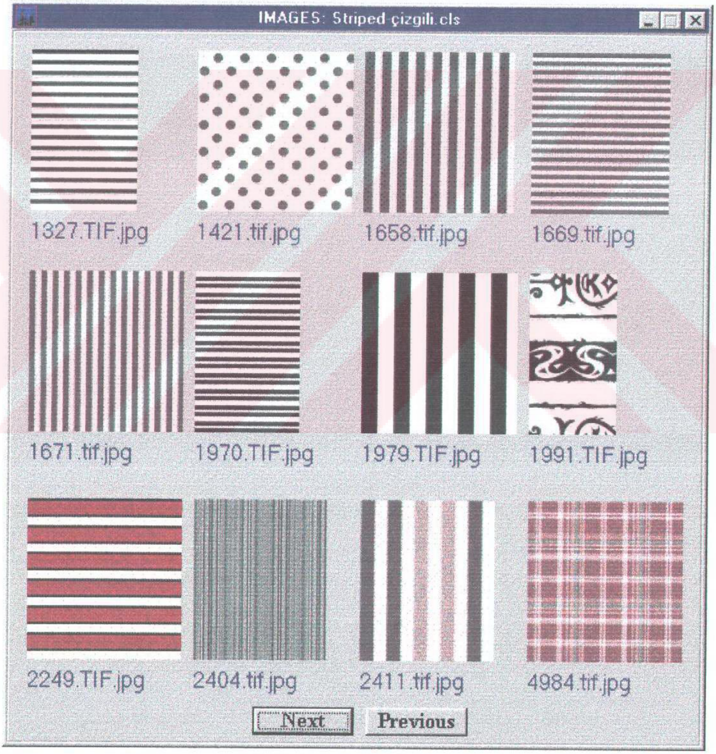


APPENDIX B

CLASSIFIED IMAGES

The Result Images of Manual Thresholding Classification with Line Threshold > 0 and Circle Threshold > 0

The Images of Striped Class

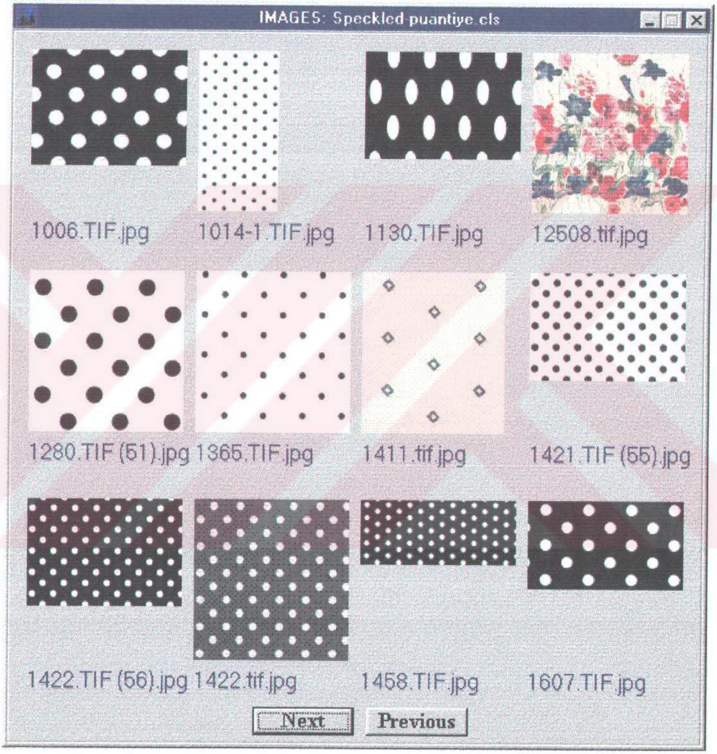


The Result Images of Manual Thresholding Classification with Line Threshold > 0 and Circle Threshold > 0

The Images of Plaided Class

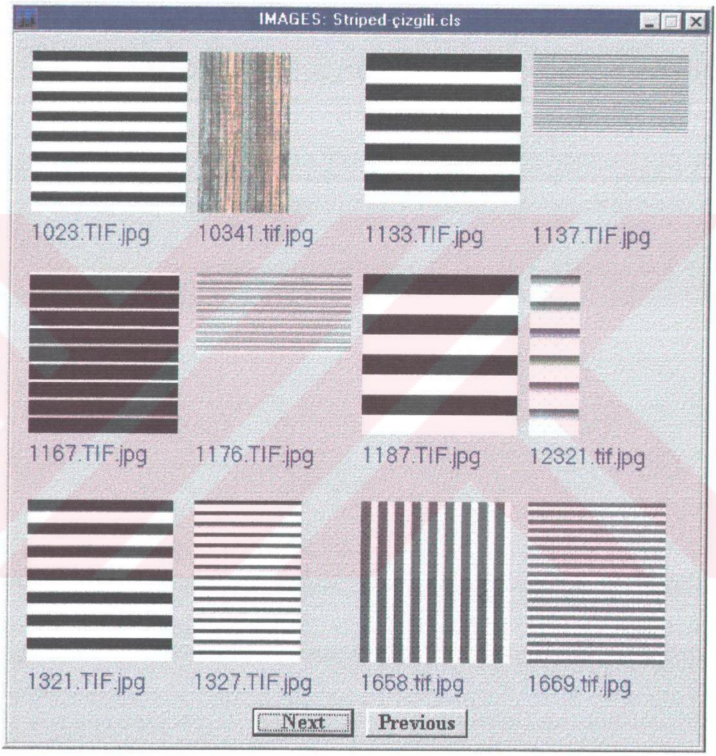
The Result Images of Manual Thresholding Classification with Line Threshold > 0 and Circle Threshold > 0

The Images of Speckled Class



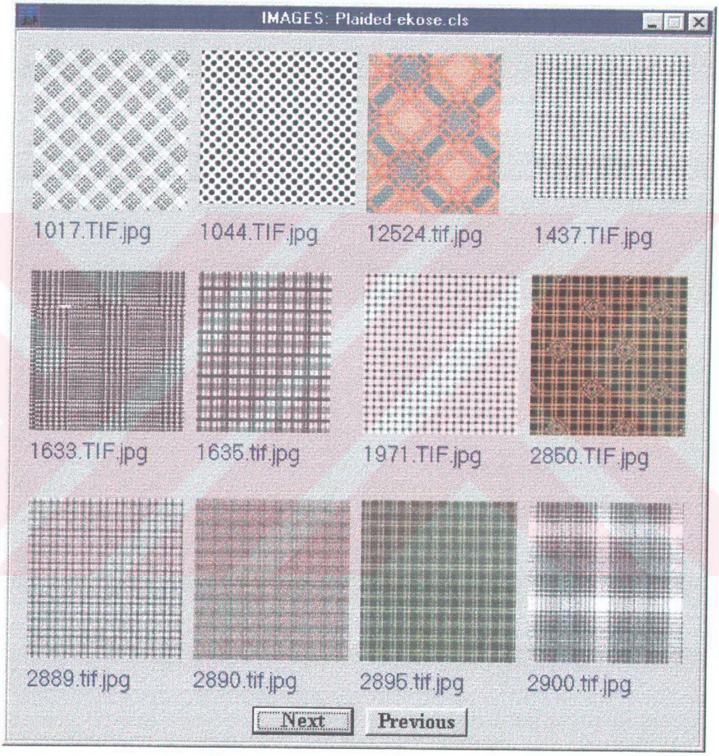
The Result Images of Nearest Neighbor classification

The Images of Striped Class



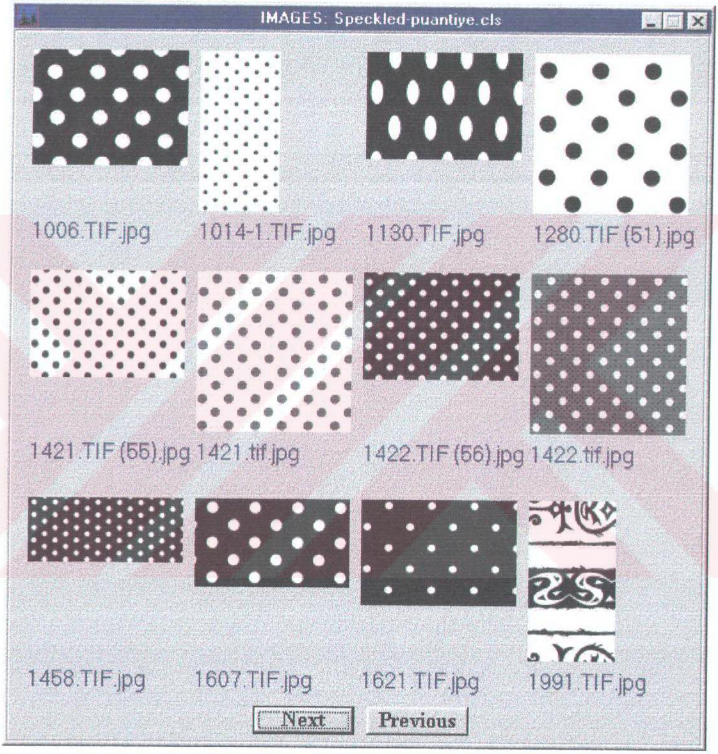
The Result Images of Nearest Neighbor classification

The Images of Plaided Class



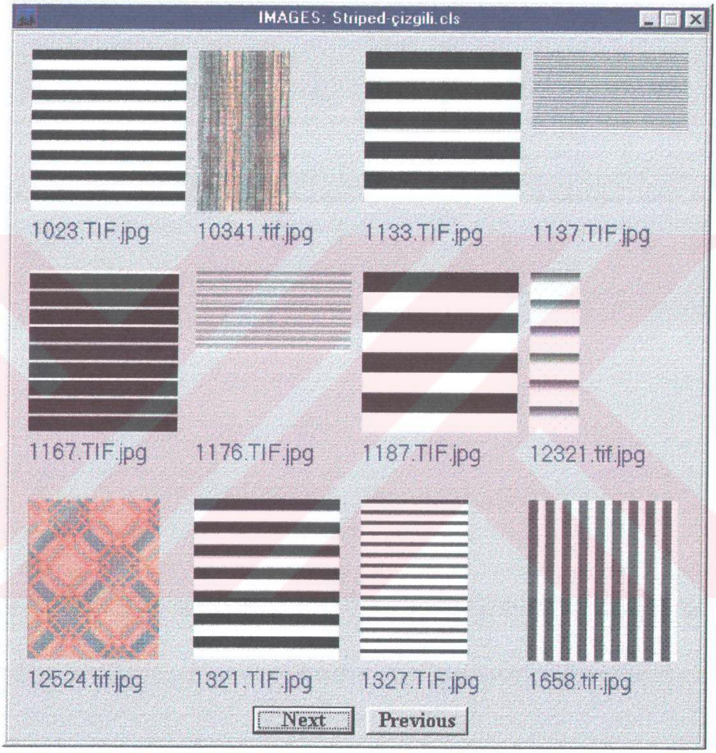
The Result Images of Nearest Neighbor classification

The Images of Speckled Class



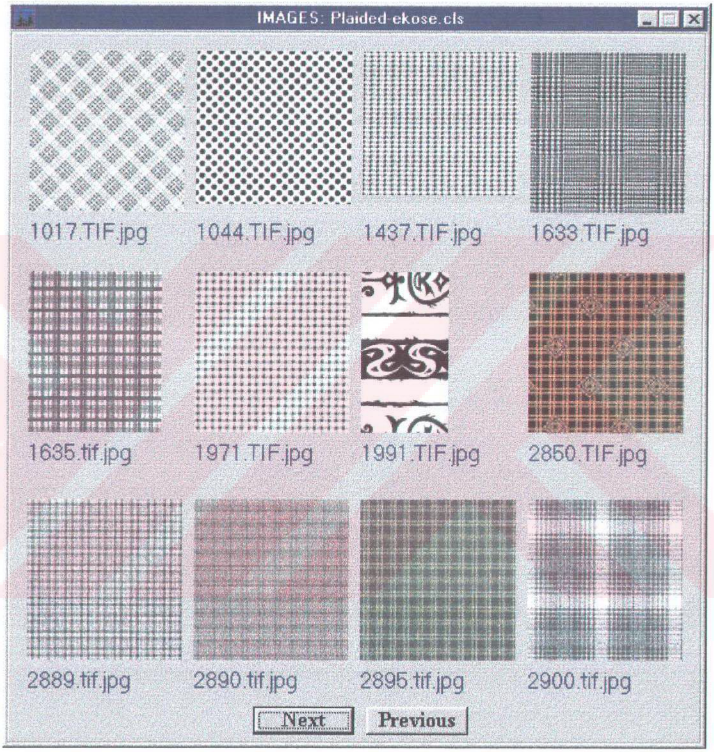
The Result Images of k-Nearest Neighbor Classification with *k*=3

The Images of Striped Class



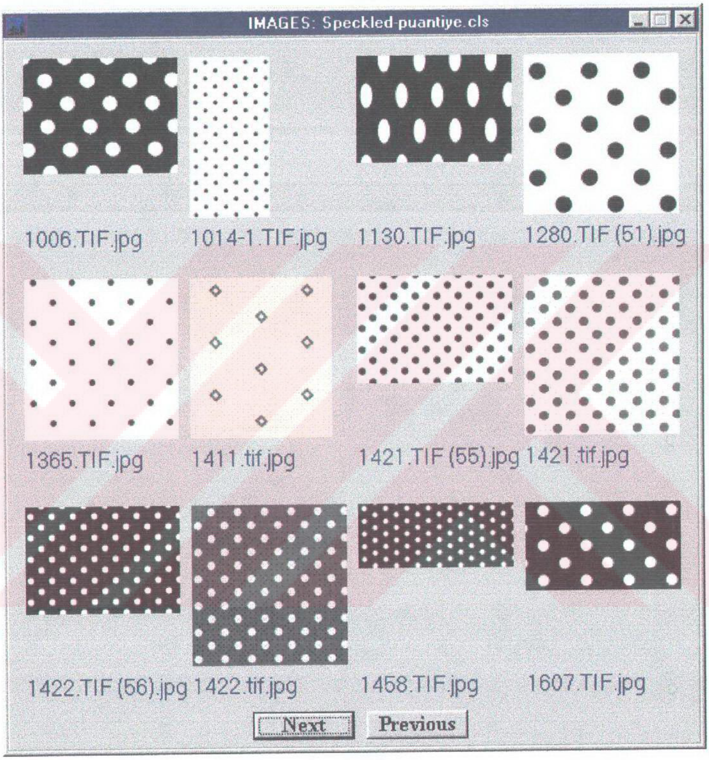
The Result Images of k-Nearest Neighbor Classification with *k*=3

The Images of Plaided Class



The Result Images of k-Nearest Neighbor Classification with *k*=3

The Images of Speckled Class

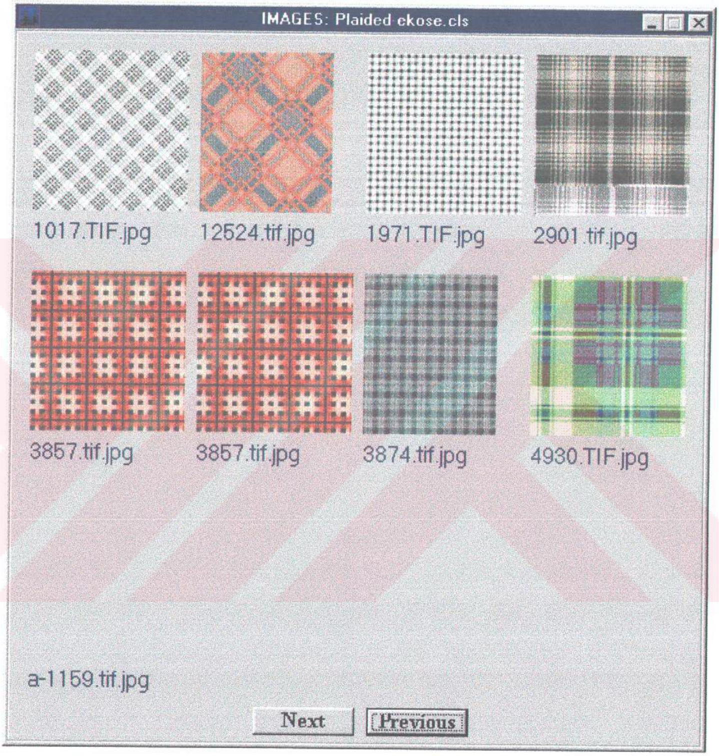


The Result Images of Minimum Distance Classification

The Images of Striped Class

The Result Images of Minimum Distance Classification

The Images of Plaided Class



The Result Images of Minimum Distance Classification

The Images of Speckled Class