

**DOKUZ EYLÜL UNIVERSITY**  
**GRADUATE SCHOOL OF NATURAL AND APPLIED**  
**SCIENCES**

**CLASSIFICATION OF MARBLE TEXTURES**  
**USING NEURAL NETWORKS AND IMAGE**  
**PROCESSING METHODS**

by  
**Emre ARDALI**

**August, 2008**  
**İZMİR**

**CLASSIFICATION OF MARBLE TEXTURES  
USING NEURAL NETWORKS AND IMAGE  
PROCESSING METHODS**

**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 Electrical and Electronics Engineering**

**by  
Emre ARDALI**

**August, 2008  
İZMİR**

## M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “**CLASSIFICATION OF MARBLE TEXTURES USING NEURAL NETWORKS AND IMAGE PROCESSING METHODS**” completed by **EMRE ARDALI** under supervision of **Asst. Prof. Dr. OLCAY AKAY** and we certify 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. Olcay AKAY  
Supervisor

.....  
\_\_\_\_\_  
Asst. Prof. Dr. Güleser Kalaycı Demir  
(Jury Member)

.....  
\_\_\_\_\_  
Assoc. Prof. Dr. Aydoğan Savran  
(Jury Member)

\_\_\_\_\_  
Prof. Dr. Cahit HELVACI  
Director  
Graduate School of Natural and Applied Sciences

## **ACKNOWLEDGMENTS**

I would like to thank my advisor Asst. Prof. Dr. Olcay AKAY for his patient support and valuable guidance during this thesis study. I must express that his scholarship and attention to details have set an example for me and added considerably to my graduate experience. I also would like to thank Research Asst. M. Alper SELVER for his support and encouragement. His ideas and hard work always motivated me. I want to express my appreciation to Prof. Dr. Cüneyt GÜZELİŞ, research students Aykut KOCAOĞLU and Edip BİNER for their contributions on some topics. Lastly, I thank my family for their never ending support and motivation.

Emre ARDALI

# **CLASSIFICATION OF MARBLE TEXTURES USING NEURAL NETWORKS AND IMAGE PROCESSING METHODS**

## **ABSTRACT**

Marbles are used commonly in daily life for different purposes (building block, decorative material etc.). Classification of marble slabs according to usage purpose and quality is an important procedure. Generally human experts perform the classification procedure which is time consuming, costly and error prone. Therefore, automatic and computerized methods of classification are needed for stable and low cost procedure. In this thesis, an automatic classification method for marble slabs using image processing and artificial neural network methods, is studied under the scope of TÜBİTAK MAG 104M358 research project. Different image processing and neural networks strategies are investigated to achieve high classification performance and their performances are compared based on simulation results.

**Keywords:** Classification of marble slabs, feature extraction, artificial neural network classifier, cascaded classifier networks, sum and difference histograms, perceptron pocket learning algorithm.

# DOĞAL MERMER KAYAÇ ÖRNEKLERİNİN YAPAY SİNİR AĞLARI VE GÖRÜNTÜ İŞLEME YÖNTEMLERİ İLE SINIFLANDIRILMASI

## ÖZ

Mermer blokları günlük yaşamda çeşitli amaçlarla yaygın olarak kullanılmaktadır (yapı elemanı, dekorasyon malzemesi vb.). Mermer bloklarının kullanım amacına ve kalitesine göre sınıflanması oldukça önemli bir süreçtir. Genel olarak zaman alıcı, maliyetli ve hataya açık bu işlem uzmanlar tarafından gerçekleştirilmektedir. Bu nedenle, kararlı ve düşük maliyetli bir süreç için otomatik ve sayısallaştırılmış bir yöntem ihtiyacı duyulmaktadır. Bu tezde, TÜBİTAK MAG 104M358 araştırma projesi kapsamında mermer bloklarının imge işleme ve yapay sinir ağları yöntemleri kullanılarak otomatik sınıflanması üzerine çalışılmıştır. Farklı imge işleme ve sinir ağı teknikleri yüksek sınıflama başarımı elde etmek için incelenmiş ve benzetim sonuçlarına göre karşılaştırmaları yapılmıştır.

**Anahtar sözcükler:** Mermer bloklarının sınıflanması, öznitelik çıkarımı, yapay sinir ağı sınıflayıcı, çok katlı sınıflayıcı ağ, toplam ve fark histogramları, algılayıcı cep öğrenme algoritması.

## CONTENTS

	<b>Page</b>
THESIS EXAMINATION RESULT FORM .....	ii
ACKNOWLEDGMENTS .....	iii
ABSTRACT .....	iv
ÖZ .....	v
<b>CHAPTER ONE - INTRODUCTION .....</b>	<b>1</b>
1.1 Introduction .....	1
<b>CHAPTER TWO - BACKGROUND.....</b>	<b>3</b>
2.1 Sum and Difference Histograms (SDH) .....	3
2.1.1 Statistical Features Extracted From SDH .....	6
2.2 Wavelet Analysis .....	7
2.3 Principal Component Analysis (PCA) .....	8
2.3.1 Dimensionality Reduction Using PCA .....	10
2.4 Neural Networks .....	11
2.4.1 Multi Layer Perceptrons (MLP).....	12
2.4.2 Radial Basis Function Networks (RBFN).....	13
2.4.3 Probabilistic Neural Networks (PNN) .....	14
<b>CHAPTER THREE - DATASET .....</b>	<b>15</b>
3.1 Acquisition of Marble Images.....	15
3.2 Image Database and Quality Classes .....	17
<b>CHAPTER FOUR - SIMULATIONS AND RESULTS .....</b>	<b>19</b>
4.1 Classification Using Sum and Difference Histograms (SDH) Method .....	20
4.1.1 Multi Layer Perceptron (MLP) Classifier Evaluation.....	24
4.1.2 Radial Basis Function Networks (RBFN) Classifier Evaluation .....	25
4.1.3 Probabilistic Neural Networks (PNN) Classifier Evaluation.....	26

4.2 Classification Using Wavelet Analysis Method .....	27
4.3 Classification Using Combination of SDH and Wavelet Analysis Methods...	28
4.4 Classification Using Two Stage (Cascaded) Network .....	29
4.4.1 First Stage, Pre-classifier .....	30
4.4.2 Second Stage, Post-Classifier .....	33
4.4.3 A Modified Pre-Classifier for Two Stage Network .....	34
<b>CHAPTER FIVE - CONCLUSION .....</b>	<b>36</b>
<b>REFERENCES.....</b>	<b>39</b>
<b>APPENDIX A .....</b>	<b>41</b>

# **CHAPTER ONE**

## **INTRODUCTION**

### **1.1 Introduction**

Marbles are used in daily life as decorative materials and building blocks. Hence classification of marbles, in terms of quality, arises as an important problem. Quality classification of marbles is an important procedure and traditionally performed by human experts. On the other hand this method has some drawbacks; it may be time consuming, include subjective decisions (depending on human expert or illumination conditions), tend to be faulty because of visual fatigue and could be a costly and time consuming method. In this thesis, an automatic and computational method is developed using digital image processing and neural network methods so that savings of time and cost and reduction of human related mistakes are possible.

There are various researches in the literature on the classification of marble slab images (Alajarin, et. al., 2005), (Sousa & Pinto, 2004), (Hernandez et. al., 1995). This thesis study considers (Alajarin, et. al., 2005) as a starting point, due to its superior and challenging results, and aims to adopt the procedures described in (Alajarin, et. al., 2005) for our own database with improved performance. In this thesis, based on (Alajarin, et. al., 2005), different textural feature extraction methods on different color spaces are investigated and different types of classifiers and neural network architectures are researched. A better classification performance is aimed to be obtained by extending the methods used in (Alajarin, et. al., 2005) and using some other approaches. An automatic method is attempted to be developed for the classification of marble samples.

In Chapter 2, some background information is given on some fundamental methods used in this thesis. Textural feature extraction methods, Sum and Difference Histograms and wavelet analysis methods are explained briefly. A well known transformation method, Principal Component Analysis, is described with a basic

derivation. A short explanation of different neural network methods and architectures is given.

Chapter 3 introduces marble samples used in the study, content of the database which consists of marble images. Typical quality groups and their characteristic features are described.

Chapter 4 consists of detailed explanation of the computational methods and details of the applications. Simulation results of different feature set/neural network combinations are tabulated with some performance metrics.

Finally, conclusions are given in Chapter 5. Brief comments on applied methods and results are summarized.

## **CHAPTER TWO**

### **BACKGROUND**

In this chapter, theoretical background of used methods and tools are covered in a brief manner. In the first section, a textural information extraction method, Sum and Difference Histograms, is described briefly and statistical features defined by them are given. Second section comprises basic review of another feature extraction method, wavelet analysis, which is also used in marble classification literature. Third section covers a nonparametric transformation method, Principal Component Analysis (PCA). The last section includes a basic discussion on neural network architectures which are used within this thesis.

#### **2.1 Sum and Difference Histograms (SDH)**

In this thesis, a useful textural information extraction method, which is called Sum and Difference Histograms (SDH), is used. SDH are first introduced in 1986 by M. Unser (Unser, 1986). SDH are an alternative method to co-occurrence matrix (COM) (which was first introduced by Haralick (Haralick et al. 1973)). COM is based on the spatial gray level dependence and gives an approximation of joint probability distribution of gray levels. Since each channel of the color image presents 256 gray levels, the requirements of memory storage and time consumption with COM are extremely large (processing of three 256x256 matrices). SDH offer, in this sense, a very good alternative to traditional COM used for texture analysis. Experimental results show that SDH are as powerful as COM for texture discrimination with the advantages of decreased computation time and memory storage space (Selver, et. al., 2007). The number of elements to analyze grows as a quadratic function with the number of gray levels for COM, while it grows linearly for SDH. Below, the SDH algorithm is briefly summarized.

Consider a  $K \times L$  image denoted by  $\{y_{k,l}\}$ ,  $k = 1, 2, \dots, K$ ,  $l = 1, 2, \dots, L$  with grey levels  $G = \{0, 1, \dots, N_G - 1\}$ . Two picture elements  $y_{k,l}$  and  $y_{k+d1, l+d2}$  are separated by  $(d1, d2) \in D$ , where  $D$  is the subset of indexes specifying the texture region to be analyzed. Then, for a relative displacement  $(d1, d2)$ , the sum and difference are defined as

$$\begin{aligned} s_{k,l} &= y_{k,l} + y_{k+d1, l+d2} \\ d_{k,l} &= y_{k,l} - y_{k+d1, l+d2} \end{aligned} \quad \text{Eq. 2.1}$$

$1 \times (2N_G - 1)$  dimensional normalized sum and difference histogram vectors can be defined as;

$$\begin{aligned} P_s(i) &= h_s(i) / N; \quad i = 0, 1, \dots, 2N_G - 2 \\ P_d(j) &= h_d(j) / N; \quad j = -N_G + 1, \dots, 0, 1, \dots, N_G - 1 \end{aligned} \quad \text{Eq. 2.2}$$

where

$$\begin{aligned} h_s(i) &= \text{Card}\{(k,l) \in D, s_{k,l} = i\} \\ h_d(j) &= \text{Card}\{(k,l) \in D, d_{k,l} = j\} \\ N &= \text{Card}\{D\} = \sum_i h_s(i) = \sum_j h_d(j). \end{aligned} \quad \text{Eq. 2.3}$$

Referring cardinality, *Card* function gives the number of elements in the set that is its argument, which corresponds to unnormalized histograms of  $s_{k,l}$  and  $d_{k,l}$  for all  $k, l$ . To obtain the SDH, distance set is selected as 8 neighborhoods in this study. The graphical representation of pixels is given in Figure 2.1.

(-1, 1)	(0, 1)	(1, 1)
(-1, 0)	* (0, 0)	(1, 0)
(-1, -1)	(0, -1)	(1, -1)

Figure 2.1 8 pixel neighborhoods for SDH.

A simple example is given below for a better understanding of the SDH method. In Figure 2.2 there are two binary images whose normal (global) histograms are the same, but SDH are different. Since images contain two gray levels, sum histogram is within the range  $[0, 2]$  while the difference histogram is within the range of  $[-1, 1]$ . Sum and difference of the neighbor elements for the center pixels are given in Table 2.1.

<b>1</b>	<b>0</b>	<b>1</b>
<b>0</b>	<b>1</b>	<b>0</b>
<b>1</b>	<b>0</b>	<b>1</b>

**[ 1 ]**

<b>1</b>	<b>1</b>	<b>1</b>
<b>1</b>	<b>0</b>	<b>1</b>
<b>0</b>	<b>0</b>	<b>0</b>

**[ 2 ]**

Figure 2.2 Simple binary images for SDH example.

Table 2.1 Sum and difference values of neighbor pixels and the center pixel.

	$n_{(-1,1)}$	$n_{(0,1)}$	$n_{(1,1)}$	$n_{(-1,0)}$	$n_{(1,0)}$	$n_{(-1,-1)}$	$n_{(0,-1)}$	$n_{(1,-1)}$
$s[1]_{0,0}$	2	1	2	1	1	2	1	2
$d[1]_{0,0}$	0	1	0	1	1	0	1	0
$s[2]_{0,0}$	1	1	1	1	1	0	0	0
$d[2]_{0,0}$	-1	-1	-1	-1	-1	0	0	0

This simple procedure is normally applied to all pixel elements on larger images.

Then, normalized sum and difference histograms can be written using Table 2.1,

$$P[1]_s(i) = [0.0, 0.5, 0.5]$$

$$P[1]_d(j) = [0.0, 0.5, 0.5]$$

$$P[2]_s(i) = [0.375, 0.625, 0]$$

$$P[2]_d(j) = [0.625, 0.375, 0]$$

where  $i = 0, 1, 2$  and  $j = -1, 0, 1$ . Since SDH are different for those two images, any textural feature extracted from them helps to distinguish these images. In this thesis, some statistical features which are explained in (Unser, 1986) are extracted from the SDH and they are used as textural descriptors. They are discussed in the next section.

### 2.1.1 Statistical Features Extracted From SDH

There are seven fundamental statistical features extracted from SDH. These features are mean, variance (measure of dispersion of the gray level values around the mean), energy, correlation (dependency of gray level of neighbor pixels), entropy (complexity of texture, complex textures should yield high entropy), contrast (amount of local variations, calculated value for uniform images is zero, as the gray level variations increase calculated value should also increase) and homogeneity (local uniformity of texture, calculated value is high if the image has good homogeneity, calculated value would be low if there are many gray level transitions) (Acharya & Ray, 2005) which are defined in Table 2.2 mathematically.

Table 2.2 Statistical features defined by SDH.

Parameter	Expression
Mean	$\frac{1}{2} \sum_i iP_s(i)$
Variance	$\frac{1}{2} \left( \sum_i (i - 2\mu)^2 P_s(i) + \sum_j j^2 P_d(j) \right)$
Energy	$\sum_i P_s(i)^2 \sum_j P_d(j)^2$
Correlation	$\frac{1}{2} \left( \sum_i (i - 2\mu)^2 P_s(i) - \sum_j j^2 P_d(j) \right)$
Entropy	$-\sum_i P_s(i) \log(P_s(i)) - \sum_j P_d(j) \log(P_d(j))$
Contrast	$\sum_j j^2 P_d(j)$
Homogeneity	$\sum_j \frac{1}{1 + j^2} P_d(j)$

## 2.2 Wavelet Analysis

Wavelets are mathematical functions which are used to represent signals in time-frequency domain (versus amplitude, 3 dimensional representation) while Fourier transform represents signals in frequency domain (versus amplitude, 2 dimensional representation). Wavelets are similar to short-time Fourier analysis. Basis functions, unlike to Fourier which has sinusoidal basis functions, are small waves called wavelets of varying frequency and limited duration (Gonzales & Woods, 2002). Wavelets are generated from a basic wavelet (mother wavelet) by scaling and translations. While temporal analysis is performed with a shrunk (high frequency) version of mother wavelet, frequency analysis is performed with a dilated (low frequency) version of the mother wavelet (Graps, 1995). Two principal features are; mother wavelet should have finite energy and wavelets family function should form an orthonormal basis.

Wavelets may be applied to 2 dimensional signals (i.e. images) and specifically application on marble images is available in the literature (Delgado et. al., 2003). Proposed methods consider that each marble quality group has it is own information levels in the different frequency channels. Algorithm used in this application is the Discrete Wavelet Transform (DWT). DWT is equivalent to a series of low and high pass filtering of the original signal followed by down sampling. In case of two dimensional signals (marble images in our application), high pass filtering is applied at vertical, horizontal and diagonal directions and gives the details at the applied direction. A low resolution version of the original image is obtained with the low pass filtering and down sampling (Figure 2.3).

3 levels of DWT decomposition is applied in our study using MATLAB wavelet toolbox software. Then mean, median and variance values of each level of decomposition are computed as textural descriptors.

low res. image	level 3	hor. details level 2	horizontal details level 1
level 3	level 3		
ver. details level 2		diag. details level 2	
vertical details level 1			diagonal details level 1

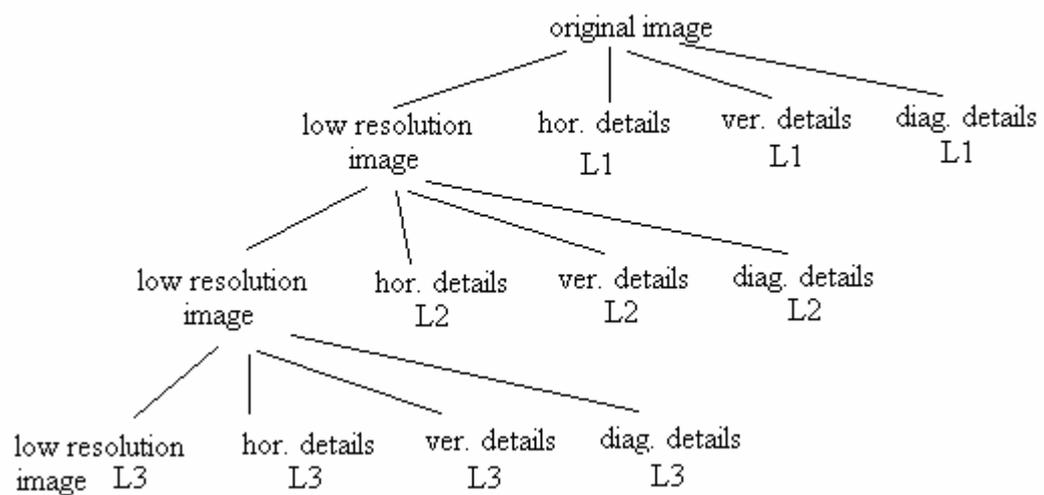


Figure 2.3 Wavelet decomposition scheme (3 levels).

### 2.3 Principal Component Analysis (PCA)

Principal component analysis (PCA) (also known as Karhunen-Loève transformation or Hotelling transformation) is a well known non-parametric transformation. This method is widely used to represent multi dimensional datasets. It is briefly explained below.

PCA is a statistical method which can find an optimal linear transformation  $\mathbf{Q} \in \mathfrak{R}^{n \times n}$  such that an input vector (i.e. feature set)  $\mathbf{x} \in \mathfrak{R}^{n \times 1}$  can be represented as uncorrelated orthogonal dataset  $\mathbf{y} \in \mathfrak{R}^{n \times 1}$ ;

$$\mathbf{y} = \mathbf{Q}^T \mathbf{x} \quad . \quad \text{Eq. 2.4}$$

Let  $\mathbf{x}$  denote an  $n$  dimensional random vector with zero mean (if  $\mathbf{x}$  has nonzero mean, first mean of  $\mathbf{x}$  should be subtracted, so that resulting vector has zero mean). Then  $\mathbf{y}$  can be written as;

$$\begin{aligned} y_1 &= \mathbf{q}_1^T \mathbf{x} \\ \dots & \\ y_i &= \mathbf{q}_i^T \mathbf{x} \\ \mathbf{y} &= [\mathbf{q}_1 \ \mathbf{q}_2 \ \dots \ \mathbf{q}_n]^T \mathbf{x} = \mathbf{Q}^T \mathbf{x} \end{aligned} \quad \text{Eq. 2.5}$$

where vectors  $\mathbf{q}_i$  should be orthonormal. That is;

$$\begin{aligned} \mathbf{q}_i^T \mathbf{q}_j &= \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \\ \mathbf{Q}^T \mathbf{Q} &= \mathbf{I}, \quad \mathbf{Q}^T = \mathbf{Q}^{-1} . \end{aligned} \quad \text{Eq. 2.6}$$

Mean of  $y_i$  is zero (since  $\mathbf{x}$  is zero mean) and variance of random variable  $y_i$  may be written as;

$$\begin{aligned} \text{Var}(y_i) &= E[y_i^2] - E[y_i]^2 \text{ where } E[\cdot] \text{ is expectation and } E[y_i] = 0. \\ \sigma_{y_i}^2 &= E[y_i^2] = \mathbf{E}[(\mathbf{q}_i^T \mathbf{x})(\mathbf{x}^T \mathbf{q}_i)] = \mathbf{q}_i^T \mathbf{E}[\mathbf{x} \mathbf{x}^T] \mathbf{q}_i \\ \sigma_{y_i}^2 &= \mathbf{q}_i^T \mathbf{R}_x \mathbf{q}_i \quad \mathbf{R}_x : \text{autocorrelation matrix of } \mathbf{x} \\ \sigma_{y_i}^2 &= \mathbf{q}_i^T \mathbf{C}_x \mathbf{q}_i \quad \mathbf{C}_x : \text{covariance matrix of } \mathbf{x}, \mathbf{C}_x = \mathbf{R}_x - \mathbf{E}[\mathbf{x}]^2, \mathbf{E}[\mathbf{x}] = 0 \end{aligned} \quad \text{Eq. 2.7}$$

Since it is desired that  $\mathbf{y}$  is an uncorrelated random vector,  $\mathbf{C}_y$  should be diagonal.

$$\begin{aligned} \begin{bmatrix} \sigma_{y1}^2 & 0 & \dots & \dots & 0 \\ 0 & \sigma_{y2}^2 & \dots & \dots & 0 \\ 0 & 0 & \sigma_{y3}^2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \sigma_{yn}^2 \end{bmatrix} &= [\mathbf{q}_1 \ \mathbf{q}_2 \ \dots \ \mathbf{q}_n]^T \mathbf{C}_x [\mathbf{q}_1 \ \mathbf{q}_2 \ \dots \ \mathbf{q}_n] \\ \mathbf{C}_y &= \mathbf{Q}^T \mathbf{C}_x \mathbf{Q} \\ \mathbf{C}_y &= \mathbf{Q}^{-1} \mathbf{C}_x \mathbf{Q} \\ \mathbf{Q} \mathbf{C}_y &= \mathbf{C}_x \mathbf{Q} . \end{aligned} \quad \text{Eq. 2.8}$$

The above equation form is the well-known eigenvalue-eigenvector problem. Hence it is seen that the transformation matrix  $\mathbf{Q}$  is the matrix which is built by the eigenvectors of  $\mathbf{C}_x$  covariance matrix, and  $\mathbf{C}_y$  is the diagonal eigenvalue matrix  $diag[\lambda_1 \lambda_2 \dots \lambda_n] = diag[\sigma_{y1}^2 \sigma_{y2}^2 \dots \sigma_{ym}^2]$  of  $\mathbf{C}_x$  (Haykin, 1999), (Ham & Kostanic, 2001).

PCA method has many different application areas in signal processing; one of the most important application area is dimensionality reduction which is also the reason for using PCA in this thesis. Dimensionality reduction using PCA is explained in the next subsection.

### 2.3.1 Dimensionality Reduction Using PCA

Classification applications may suffer from multi dimensional feature sets. As the number of dimension in the feature set increases, it is possible to have reduction in the classification performance (curse of dimensionality). In that case PCA provides an effective method for dimensionality reduction. Number of features needed for effective representation of data may be reduced by discarding components that have small variances (contributions) and retain only the components that have large variances (Haykin, 1999), (Jolliffe, 2002).

Let eigenvectors of  $\mathbf{C}_x$  are ordered according to decreasing eigenvalues  $\lambda_1 > \lambda_2 > \dots > \lambda_n$  so that associated eigenvectors  $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n$  are stored in  $\mathbf{Q}$  transformation matrix in the given order  $\mathbf{Q} = [\mathbf{q}_1 \mathbf{q}_2 \dots \mathbf{q}_n]$ . It is possible to discard some columns (eigenvectors) of  $\mathbf{Q}$  which correspond to smallest eigenvalues, and have a new transformation matrix  $\mathbf{W} \in \mathcal{R}^{n \times m}$   $\mathbf{W} = [\mathbf{q}_1 \mathbf{q}_2 \dots \mathbf{q}_m]$  ( $m < n$ )

$$\mathbf{a} = \mathbf{W}^T \mathbf{x}, \quad \mathbf{a} \in \mathcal{R}^{m \times 1}. \quad \text{Eq. 2.9}$$

First  $m$  eigenvectors of  $\mathbf{C}_x$  are considered as principal eigenvectors. These are the directions where the input data have the greatest variance (predominant information content). The rest of the eigenvectors (discarded eigenvectors) are the directions where the input data have the minimum variance (irrelevant part of data, noise etc.). Thus, input data are represented in a reduced dimensional space ( $m$  dimensions). It

is obvious that there is some error introduced during the dimensionality reduction. Reconstructed data  $\hat{\mathbf{x}}$  can be written as;

$$\hat{\mathbf{x}} = \sum_{i=1}^m a_i \mathbf{q}_i = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \dots & \mathbf{q}_m \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \mathbf{W} \mathbf{a} \quad \text{Eq. 2.10}$$

and the error between the  $\mathbf{x}$  and  $\hat{\mathbf{x}}$  is given as

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}} = \sum_{i=1}^n a_i \mathbf{q}_i - \sum_{i=1}^m a_i \mathbf{q}_i = \sum_{i=m+1}^n a_i \mathbf{q}_i . \quad \text{Eq. 2.11}$$

Total variance of the  $m$  components of the data vector  $\mathbf{x}$  is

$$\sum_{i=1}^m \sigma_i^2 = \sum_{i=1}^m \lambda_i \quad \sigma_i^2 : \text{variance of the } i\text{th principal component } a_i \quad \text{Eq. 2.12}$$

Similarly total variance of the approximation vector  $\hat{\mathbf{x}}$  is

$$\sum_{i=1}^m \sigma_i^2 = \sum_{i=1}^m \lambda_i . \quad \text{Eq. 2.13}$$

Hence total variance of the  $(n - m)$  elements in the error vector  $\mathbf{x} - \hat{\mathbf{x}}$  is;

$$\sum_{i=m+1}^n \sigma_i^2 = \sum_{i=m+1}^n \lambda_i . \quad \text{Eq. 2.14}$$

The ratio  $\sum_{i=1}^m \lambda_i / \sum_{i=1}^n \lambda_i$  gives the amount of reserved data called, compression ratio (Haykin, 1999), (Ham & Kostanic, 2001). Experimental results show that considerable dimensionality reduction is possible using PCA, while the amount of data loss is quite small.

## 2.4 Neural Networks

Neural networks (NN) are widely used tools in classification applications. NN borrow their basic ideas from the human brain model. In NN, acquisition of knowledge from the environment is realized through a learning algorithm, storage of the acquired data is in the interconnection strengths (synaptic weights) and computing is in parallel manner (Haykin, 1999). NN may be mainly categorized according to learning methods;

- i) Learning with a teacher (Supervised learning),
- ii) Learning without a teacher (Unsupervised learning).

In supervised learning, the neural network is trained with a number of input-output example sets. Network parameters are tuned according to error signal which is difference between the desired response and the actual response of the network. In unsupervised learning, there is no external teacher. Network parameters are adjusted according to set of learning rules and task independent measures. After repeated application of the input patterns, network can extract meaningful relations (Haykin, 1999), (Ham & Kostanic, 2001). In this thesis, some of the supervised NN architectures are used; those are Multi Layer Perceptrons, Probabilistic Neural Networks and Radial Basis Function Networks, explained in the next section fundamentally. One may have more detailed explanations through (Haykin, 1999), (Ham & Kostanic, 2001) and (Duda, Hart & Stork, 2001).

#### ***2.4.1 Multi Layer Perceptrons (MLP)***

Multi Layer Perceptrons (MLP) architecture is an important class of neural network. Generally, it consists of three main layers which are called input layer, hidden layer and output layer (Figure 2.4). All the neural units in the network are fully connected by synaptic weights. Hidden layer may be one or more layers and gives the powerful classification and approximation properties to the network by nonlinear behavior (generally nonlinear sigmoid activation functions are used in hidden layer). MLP are trained by a popular algorithm, error backpropagation algorithm, which is based on error correction learning rule. Backpropagation consists of two passes; forward and backward pass. At the forward pass, synaptic weights are fixed and an output is evaluated. Then error signal is produced using the actual and desired output. This error signal is propagated backward through the all network layer and synaptic weights are adjusted accordingly. This kind of learning method is usually time consuming, since it needs many iterations on the whole training set (Haykin, 1999).

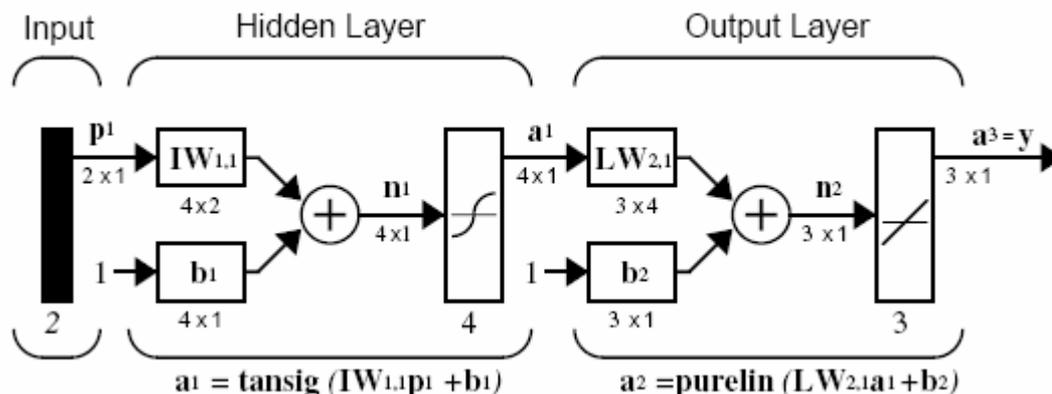


Figure 2.4 General structure of MLP (MATLAB NN Toolbox user guide v.5, p. 125).

Beside the described fundamental method, there are some modified versions of backpropagation algorithm. In this thesis, adaptive gradient descent algorithm is employed using MATLAB technical computational language NN toolbox. Simulation details are given in Chapter 4.

#### 2.4.2 Radial Basis Function Networks (RBFN)

Radial Basis Functions Networks (RBFN) may be considered as a special case of MLP such that there is only one hidden layer (generally high dimensional, i.e. number of hidden units is equal to number of training patterns) and the nonlinear activation function in the hidden layer is a Gaussian (bell shaped) function (Figure 2.5). Training time for this kind of network is quite short, storage of the network and response time to a test pattern are disadvantages of RBFN compared to MLP. Main consideration of the RBFN design is selection of mean and variance for each network unit. This type of network architecture is usually used for function approximation. In this thesis, performance of RBFN is evaluated as a classifier. Selection of parameters in our application, and simulation details are given in Chapter 4.

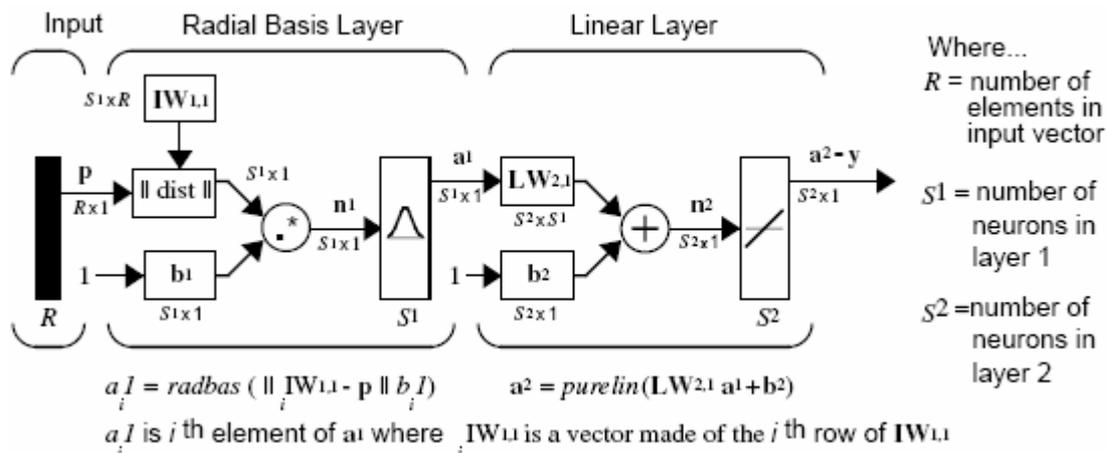


Figure 2.5 General structure of RBFN (MATLAB NN Toolbox user guide v.5, p. 260).

### 2.4.3 Probabilistic Neural Networks (PNN)

Probabilistic Neural Networks (PNN) aim to estimate of the probability density function and are based on Parzen window method. Their general structure is seen in Figure 2.6. Classification is performed by selecting the most probable class of the test pattern. Having one hidden layer and Gaussian type kernel function, it is similar to RBFN except that the output layer connections are only to radial units which belong to its class and there is no connection to units of other classes. In terms of training time, storage and response time, this kind of network has similar properties to RBFN.

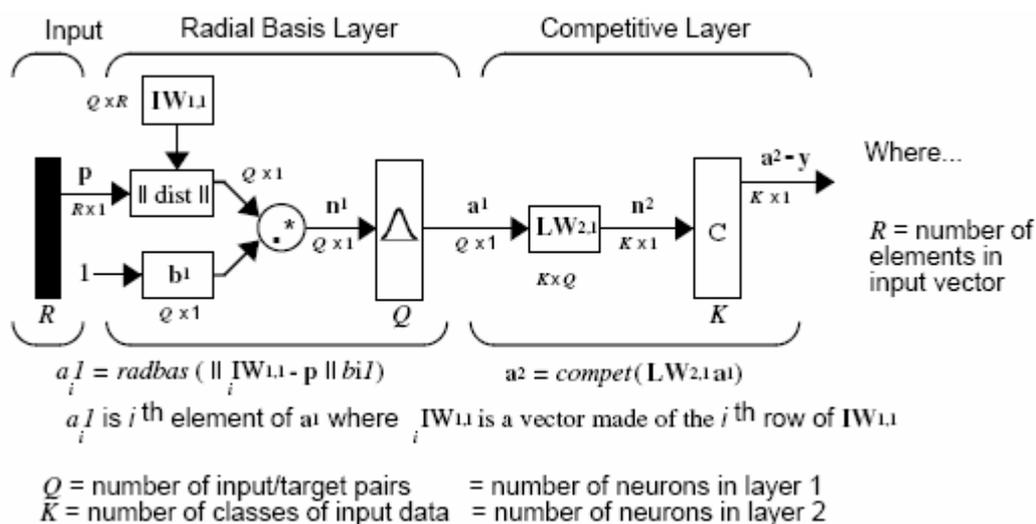


Figure 2.6 General structure of PNN (MATLAB NN Toolbox user guide v.5, p. 265).

## CHAPTER THREE

### DATASET

In this chapter, the dataset which is used in this thesis is explained briefly. Process of obtaining marble images, contents of marble image database, marble classes and their typical visual properties are described to give a better understanding of the study.

#### 3.1 Acquisition of Marble Images

Marble images database is collected by Dokuz Eylül University, Torbalı Vocational School and Civil Engineering Department under the scope of TÜBİTAK MAG 104M358 research project. Marble blocks are from the marble mine near Saruhanlı, Manisa, Turkey. Blocks are in the form of 7x7x7 cm cubes as seen in Figure 3.1. Each face of the marble cube blocks is polished to have a better visual perception which is important for the image acquisition (Figure 3.2).



Figure 3.1 A polished marble block as a 7x7x7 cm cube.

Acquisitions of images are realized in a closed container with good illumination provided by fluorescent lamps. A digital camera with high sensitivity charged coupled device sensor is used. Image acquisition system is seen in Figure 3.3.

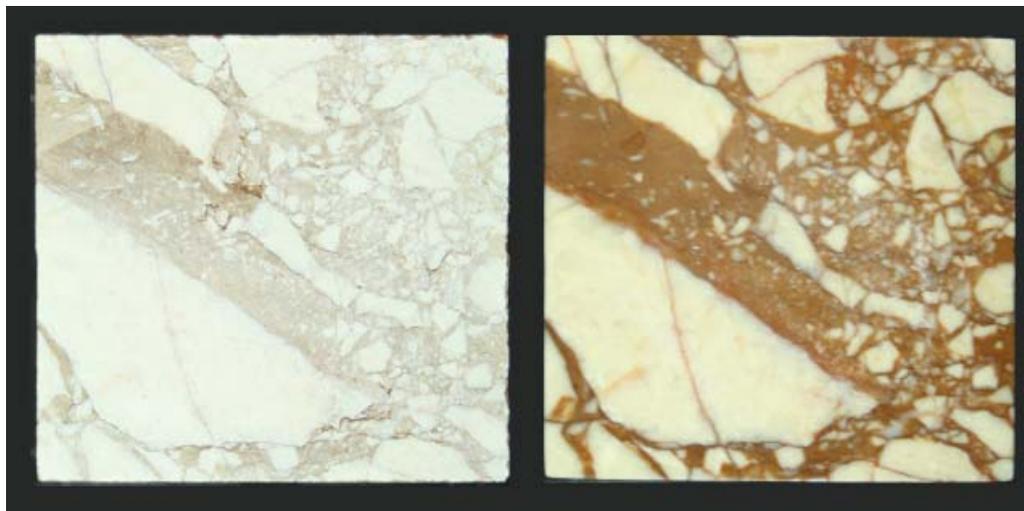


Figure 3.2 Marble block before and after polishing, visual difference is clear.

Raw images are captured at 2304x3456 resolutions with the black background. Since background information is not relevant, background is cropped and images are scaled down to 315x310 resolutions to reduce computational cost during the simulations.



Figure 3.3 Image acquisition system.

### 3.2 Image Database and Quality Classes

After acquisition of images and preprocessing them (disconnection of black background area and downscaling manually), they are classified into four quality groups by Dokuz Eylül University Civil Engineering Department experts. Color scheme, homogeneity and size, orientation, distribution of veins are used as classification criteria by the experts. Four quality groups may be titled as;

- i) Homogenous limestone,
- ii) Limestone with veins,
- iii) Fine grains (limestone) separated by cohesive matrix,
- iv) Homogenous cohesive matrix.

Cohesive matrix stands for collection of veins which are unified and construct a larger area of material. Typical samples from each group are seen in Figure 3.4.

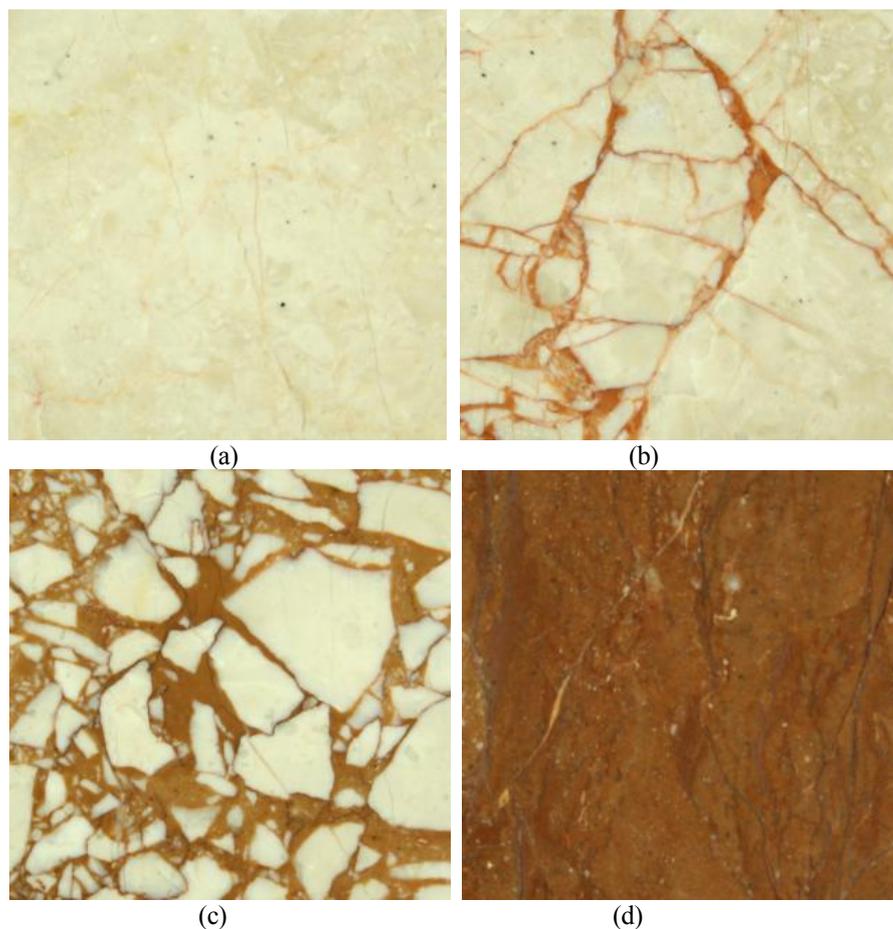


Figure 3.4 Typical sample images from each group; (a) Homogenous limestone, (b) Limestone with veins, (c) Images containing grains separated by cohesive matrix, (d) Homogenous cohesive matrix.

There are 193 pieces of marble cubes available, hence image database consists of 1158 (193x6) images. At the end of classification by experts, distribution of marble samples into quality groups is given in Table 3.1.

Table 3.1 Distribution of marble samples into quality groups.

	Quality Group 1	Quality Group 2	Quality Group 3	Quality Group 4
Number of samples	172	388	411	187
Percentage (%)	14.85	33.51	35.49	16.15

## **CHAPTER FOUR**

### **SIMULATIONS AND RESULTS**

In this chapter, classification schemes are explained, simulations, application details and simulation results are given. Computational processes and simulations are realized using the MATLAB technical computational language and its ready software toolbox components.

In this thesis, the article with the title “Automatic System for Quality-Based Classification of Marble Textures” (Alajarin, et. al., 2005) is the main reference which has successful and challenging results in terms of classification performance. Our extended sample database (in terms of number of samples), greater number of classification groups (four as opposed to three) and relatively different visual appearance of the samples give us motivation for applying the similar classification techniques in our applications. Research works performed during this thesis study show that the classification techniques are not as successful as stated in (Alajarin, et. al., 2005) with the database used in our study. Hence research is extended with different neural network architectures and techniques which are also explained in this chapter.

As the starting point, classification scheme in (Alajarin, et. al., 2005) is used which uses SDH method. Different color spaces and neural network architectures are investigated. Then; study is extended with wavelet analysis method which is also a commonly applied feature extraction method used in the literature (Delgado et. al., 2003). Since both methods have reasonable results, combination of both features is also investigated to see the effect on the classification performance. In the end, a cascaded neural network with two stage for classification is investigated with a different approach which is described in the last section.

#### 4.1 Classification Using Sum and Difference Histograms (SDH) Method

The first applied method which is also the main scheme in (Alajarin, et. al., 2005) may be summarized as follows. Color space conversion (if necessary) is performed on colored images. Then scaling of color coordinates is needed so that all components are within the [0-255] interval (color spaces other than RGB and XYZ have at least one component out of the [0-255] range). Sum and Difference Histograms (SDH) are obtained from images on each color layer (since pictures are colored), and then seven statistical features are extracted as explained in Section 2.1.1. Hence each colored marble image is represented by 3 (color layer) x 7 = 21 features. A normalization step is necessary to prevent unintended weighting effects of features. Afterwards, Principal Component Analysis (PCA) is applied to the features to reduce dimensionality. It is known that high dimensionality generally gives worse results on classification problems. In the end, the obtained features are applied to the neural network classifier. Basic processing scheme is seen in Figure 4.1.

Since images are captured as colored, possible effect of the color space is also investigated using four different color spaces. RGB which is the original color space, KL (also known as I1I2I3), YIQ and XYZ color spaces are tested to see the effect on the classification performance. They are linearly converted from RGB space. The conversion matrices are given as;

$$\begin{pmatrix} I1 \\ I2 \\ I3 \end{pmatrix} = \begin{pmatrix} 0.333 & 0.333 & 0.333 \\ 0.500 & 0.000 & -0.500 \\ -0.500 & 1.000 & -0.500 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad \text{Eq. 4.1}$$

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & -0.312 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad \text{Eq. 4.2}$$

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.607 & 0.174 & 0.201 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.116 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad \text{Eq. 4.3}$$

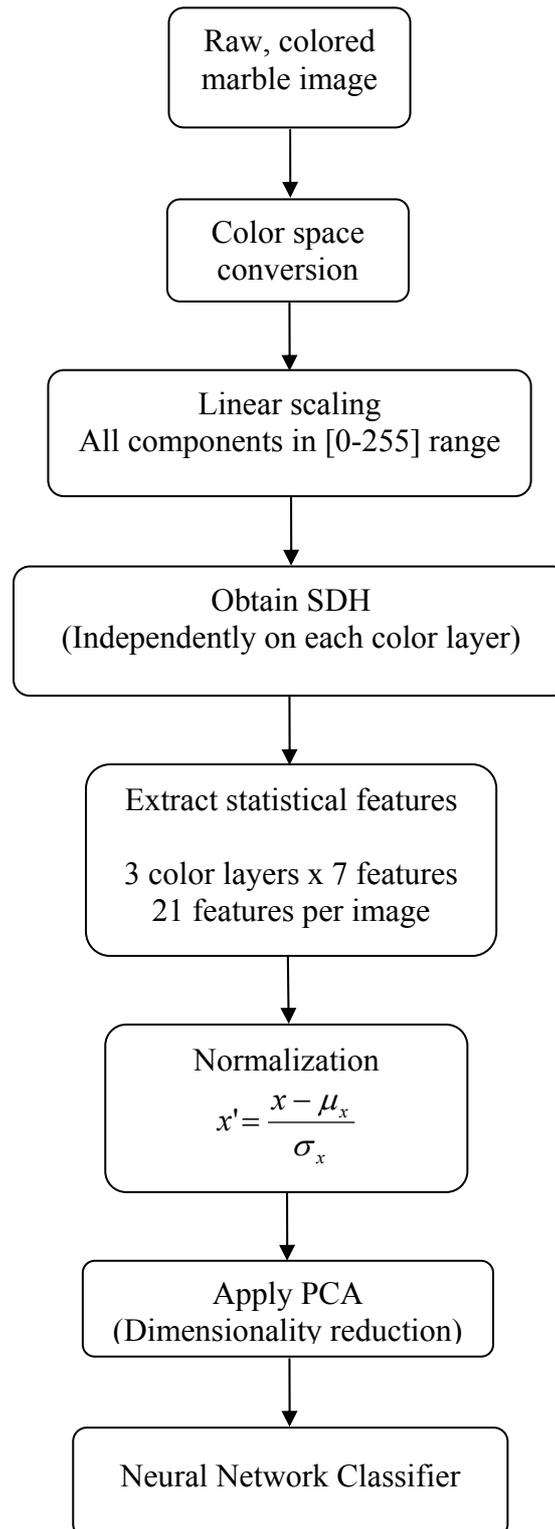


Figure 4.1 Basic processing scheme.

Color components may have negative values after space conversion, which is not suitable for SDH extraction. SDH calculations are defined over positive gray levels. Hence color space conversion is followed by a linear scaling which is necessary to make sure that all color components are within [0-255] interval. Then SDH extraction can be performed on each color component. Since 8 bit images are used in the application  $N_G = 256$  and dimensions of SDH are  $1 \times 511$ . Each image contains 3 color components. As a result  $3 \times 2 \times 511 = 3066$  elements are obtained. That is not suitable to use as a descriptor for an image since its size is too large. Instead, seven statistical descriptors are extracted from SDH which are defined in Section 2.1.1. Hence any marble image may be described by  $3 \times 7 = 21$  descriptors. However, it is considered that dimensionality is still high and a last preprocessing is performed before presenting the features to the neural network.

Normalization of the features is needed to prevent unintentional weighting effect between the features before PCA transformation. It is performed by the simple formulation

$$x'_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j} \quad \text{Eq. 4.4}$$

where  $x'_{i,j}$  is the normalized feature,  $x_{i,j}$  is the unnormalized feature,  $\mu_j$  and  $\sigma_j$ , respectively, are the mean and standard deviation of feature  $j=1, \dots, 21$ , and  $i$  is the image index. Then, Principal Component Analysis (PCA) is applied to normalized feature set. Only the principal components that have a contribution equal or greater than 0.1 % are taken into consideration. Thus, the accumulated variance is above 99.5 %. Different numbers of principal components are selected for different color spaces (see Table 4.1).

Distribution of the first two principal components in RGB space is given in Figure 4.2. The other color spaces show similar distributions. It is seen that at the end of the preprocessing, there are meaningful distributions in two dimensional spaces.

Table 4.1 Variance in percentage of principal components selected for each color space.

PC#	RGB	KL	YIQ	XYZ
1	72.5	70.519	70.52	72.602
2	16.78	13.048	12.698	17.401
3	5.6214	6.1097	6.3426	5.4531
4	3.2316	3.9553	4.1184	3.2947
5	0.63811	2.6054	2.4481	0.58438
6	0.57832	1.3642	1.3348	0.28437
7	0.24064	0.80426	0.81536	0.16947
8	0.16452	0.39937	0.49263	0.12763
9	-	0.33362	0.3621	-
10	-	0.26302	0.24554	-
11	-	0.17125	0.20015	-
12	-	0.15234	0.16279	-
13	-	0.12019	0.11785	-
sum (%)	99.75	99.85	99.86	99.92
# of comp. used for min 0.1 % fraction	8	13	13	8

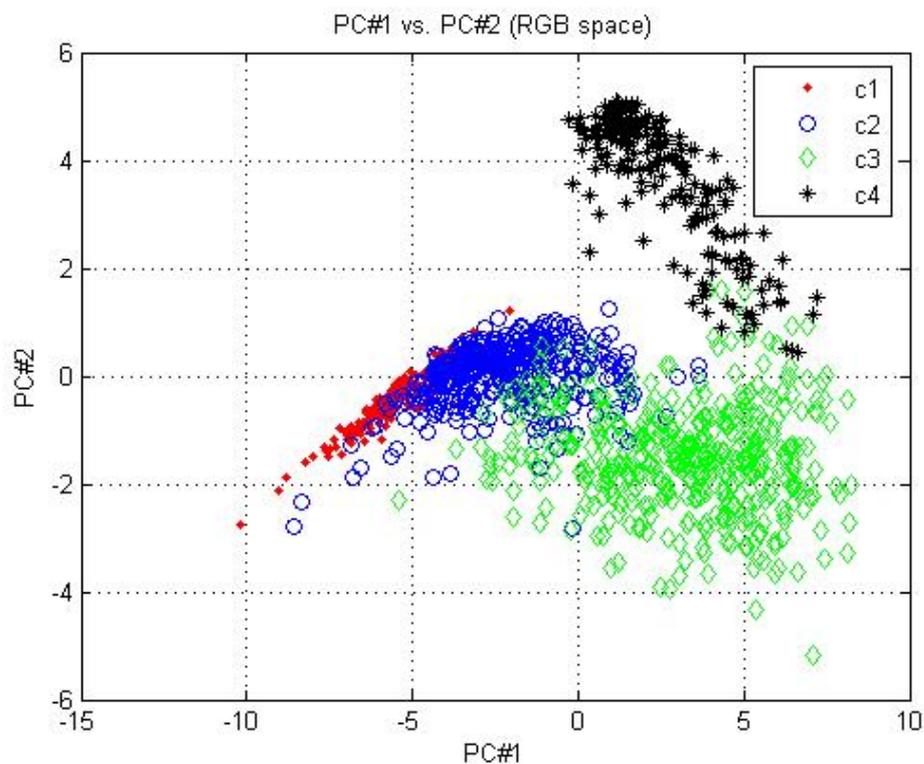


Figure 4.2 1st and 2nd Principal Components in RGB space (c1: group 1, c2: group 2, c3: group 3, c4: group 4).

After all preprocessing, features are presented to neural network for training and testing. Training and testing are performed with hold out method. Sample space is randomly divided into training and test spaces so that percentage of each group in the spaces is roughly close to each other (the same training/test spaces are also preserved for the evaluation of all classifiers). Distribution of training and test samples used in simulations are given in Table 4.2.

Table 4.2 Number of samples in training and test spaces.

	Training space	Test space	Total
Group 1	94	78	172
Group 2	301	87	388
Group 3	250	161	411
Group 4	111	76	187
Total	756	402	1158

The final step of the neural network classifier is realized with different architectures. MLP, PNN and RBF type networks are tested to see the effect of network structure on the classification performance. The following sections give the application details about the classifiers and performance results.

#### ***4.1.1 Multi Layer Perceptron (MLP) Classifier Evaluation***

Three stage MLP architecture is evaluated with one hidden layer and one output layer. There are six hidden neurons with nonlinear sigmoid activation functions and four output neurons with linear activation functions, one for each class. Network is trained with adaptive gradient descent algorithm with the network parameters; 0.0001 for goal error, 0.01 learning coefficient, 15000 training epochs, 1.05 and 0.7 as the ratios for increasing and decreasing learning rate, respectively, 1.04 as the maximum performance increase threshold.

Performance of the network is evaluated over performance metrics such as correct classification rate (CC), false positive rate (FP), false negative rate (FN), sensitivity

(SE), specificity (SP), positive predictive value (PPV) and negative predictive value (NPV). For a brief description on performance metrics, please see Appendix A.

Evaluation results with MLP network for different color spaces are given in Table 4.3. Results are promising, but needs improvement when compared to other works in the literature (Alajarin, et. al., 2005). Manipulations on parameters of the MLP (i.e. number of hidden neurons, number of training epochs) do not affect the network performance drastically.

Table 4.3 Classification performance of MLP network on different color spaces for SDH features.

color space	group	CC	FP	FN	SE	SP	PPV	NPV
RGB	G1	96.5174	2.0619	7.6923	92.3077	97.5309	90.0000	98.1366
	G2	91.7910	5.4201	14.9425	85.0575	93.6508	78.7234	95.7792
	G3	94.5274	1.3158	10.5590	89.4410	97.9253	96.6443	93.2806
	G4	99.2537	0.7519	0.0000	100.0000	99.0798	96.2025	100.0000
KL	G1	95.5224	2.3438	11.5385	88.4615	97.2222	88.4615	97.2222
	G2	92.5373	4.0323	17.2414	82.7586	95.2381	82.7586	95.2381
	G3	94.7761	2.0997	8.0745	91.9255	96.6805	94.8718	94.7154
	G4	98.7562	1.2594	0.0000	100.0000	98.4663	93.8272	100.0000
XYZ	G1	97.2637	1.5345	6.4103	93.5897	98.1481	92.4051	98.4520
	G2	92.2886	4.3127	17.2414	82.7586	94.9206	81.8182	95.2229
	G3	94.2786	2.3747	8.6957	91.3043	96.2656	94.2308	94.3089
	G4	99.2537	0.7519	0.0000	100.0000	99.0798	96.2025	100.0000
YIQ	G1	97.5124	0.5102	10.2564	89.7436	99.3827	97.2222	97.5758
	G2	94.0299	4.4974	8.0460	91.9540	94.6032	82.4742	97.7049
	G3	95.2736	1.3055	8.6957	91.3043	97.9253	96.7105	94.4000
	G4	98.7562	1.2594	0.0000	100.0000	98.4663	93.8272	100.0000

#### 4.1.2 Radial Basis Function Networks (RBFN) Classifier Evaluation

Another network architecture, RBFN, is also evaluated. RBFN uses the same number of neurons as the number of input training patterns. Hence there are 756 hidden units in the hidden layer and 4 units at the output layer. Spread parameter of the network is determined by trial and error method. Performance is evaluated over different spread parameters and the best one is selected as the classifier parameter. Performance results are seen in Table 4.4. The results seem similar to results obtained with MLP.

Table 4.4 Classification performance of RBF network on different color spaces for SDH features.

color space	group	CC	FP	FN	SE	SP	PPV	NPV
RGB sp=0.65	G1	94.5274	1.5789	20.5128	79.4872	98.1481	91.1765	95.2096
	G2	90.5473	5.7692	19.5402	80.4598	93.3333	76.9231	94.5338
	G3	89.8010	7.7562	8.0745	91.9255	88.3817	84.0909	94.2478
	G4	96.7662	0.5141	14.4737	85.5263	99.3865	97.0149	96.7164
KL sp=1.4	G1	96.5174	0.7732	14.1026	85.8974	99.0741	95.7143	96.6867
	G2	92.5373	3.7634	18.3908	81.6092	95.5556	83.5294	94.9527
	G3	90.5473	8.5165	4.3478	95.6522	87.1369	83.2432	96.7742
	G4	96.5174	0.0000	18.4211	81.5789	100.0000	100.0000	95.8824
XYZ sp=0.65	G1	95.2736	1.5666	16.6667	83.3333	98.1481	91.5493	96.0725
	G2	91.7910	3.7940	21.8391	78.1609	95.5556	82.9268	94.0625
	G3	91.0448	8.1967	3.7267	96.2733	87.5519	83.7838	97.2350
	G4	97.0149	0.0000	15.7895	84.2105	100.0000	100.0000	96.4497
YIQ sp=1	G1	95.7711	0.7792	17.9487	82.0513	99.0741	95.5224	95.8209
	G2	91.2935	5.4496	17.2414	82.7586	93.6508	78.2609	95.1613
	G3	91.0448	6.8306	6.8323	93.1677	89.6266	85.7143	95.1542
	G4	97.5124	0.2551	11.8421	88.1579	99.6933	98.5294	97.3054

#### 4.1.3 Probabilistic Neural Networks (PNN) Classifier Evaluation

The last classifier network is realized by PNN. Like RBFN, PNN also uses the same number of hidden neurons as the number of input training patterns. Network is built with 756 pattern units and 4 category output neurons. Window width parameter is selected as 0.1 and changed to different values to see its effect on the performance. It is seen that change of window width parameter has no important effect on the performance. Classifier performance metrics are given in Table 4.5 with the window width parameter set to 0.1.

Evaluation results show that color space has no important effect on the performance of the classifier since the performance results are similar for different color spaces. Hence RGB color space may be used in such an application, since it is the natural capturing color space and it reduces the computational cost during simulations. When performances of different classifier networks are compared, it is seen that, although results are similar, MLP has slightly better performance. On the other hand, MLP has the advantages of less storage and fast response while the training phase takes longer time compared to RBFN and PNN type classifiers. In

Table 4.5 Classification performance of PNN on different color spaces for SDH features.

color space	group	CC	FP	FN	SE	SP	PPV	NPV
RGB	G1	95.2736	2.0888	14.1026	85.8974	97.5309	89.3333	96.6361
	G2	90.7960	6.3014	16.0920	83.9080	92.6984	76.0417	95.4248
	G3	94.2786	1.5831	10.5590	89.4410	97.5104	96.0000	93.2540
	G4	98.7562	1.2594	0.0000	100.0000	98.4663	93.8272	100.0000
KL	G1	95.2736	2.3499	12.8205	87.1795	97.2222	88.3117	96.9231
	G2	91.5423	5.7065	14.9425	85.0575	93.3333	77.8947	95.7655
	G3	94.7761	1.5748	9.3168	90.6832	97.5104	96.0526	94.0000
	G4	99.5025	0.5000	0.0000	100.0000	99.3865	97.4359	100.0000
XYZ	G1	94.7761	2.3622	15.3846	84.6154	97.2222	88.0000	96.3303
	G2	89.0547	7.2626	20.6897	79.3103	91.7460	72.6316	94.1368
	G3	93.0348	2.4064	11.8012	88.1988	96.2656	94.0397	92.4303
	G4	98.7562	1.2594	0.0000	100.0000	98.4663	93.8272	100.0000
YIQ	G1	93.0348	3.4759	19.2308	80.7692	95.9877	82.8947	95.3988
	G2	89.8010	6.6482	19.5402	80.4598	92.3810	74.4681	94.4805
	G3	95.0249	1.8325	8.0745	91.9255	97.0954	95.4839	94.7368
	G4	99.2537	0.5013	1.3158	98.6842	99.3865	97.4026	99.6923

spite of having reasonable results, classification performance needs to be improved as compared with (Alajarin, et. al., 2005) in which there are superior results. Hence study is continued with another feature extraction method, wavelet analysis which is also investigated in the literature (Delgado et. al., 2003).

#### 4.2 Classification Using Wavelet Analysis Method

Wavelet analysis method, which is briefly explained in Section 2.2, is used to see its effect on the classification performance. 3 level of discrete wavelet decomposition (DWT) scheme is used in the application and three different features are extracted from each level of wavelet decomposition which are mean, median and variance. Wavelet analysis is performed on the gray level version of the image, hence a  $36 \times 1$  feature vector represents each image. Then PCA is applied to reduce the dimensionality so that 98.6 % of the total data is preserved while the feature vector dimension is reduced to  $22 \times 1$  (components whose contribution equal or greater than 1 % are selected).

MLP type network is used as the classifier with the same parameters used in Section 4.1.1. Evaluation results are given in Table 4.6. As seen in the table, the wavelet based features give similar performance to the SDH based features.

Table 4.6 Classification performance of MLP for wavelet features.

group	CC	FP	FN	SE	SP	PPV	NPV
G1	95.7711	2.8571	7.6923	92.3077	96.6049	86.7470	98.1191
G2	93.0348	3.2086	18.3908	81.6092	96.1905	85.5422	94.9843
G3	94.2786	2.6385	8.0745	91.9255	95.8506	93.6709	94.6721
G4	98.0100	1.2690	3.9474	96.0526	98.4663	93.5897	99.0741

Since the results are not as good as expected another new approach which is explained in the next section, is applied to obtain improved results.

### 4.3 Classification Using Combination of SDH and Wavelet Analysis Methods

Since two different methods, SDH and wavelet analysis, give reasonable results individually, then the idea of combining both methods of feature extraction seems possible. Combination of different kinds of features might help to classify different groups of sample images. Both features are combined together and then input to an MLP type classifier after application of PCA. MLP network has the same parameters used in section 4.1.1 while the feature vector is reduced from  $57 \times 1$  to  $10 \times 1$  by PCA, so that components whose contribution equal or greater than 2 % are taken into consideration, and hence 80.9 % of the data is preserved. Results are given in Table 4.7. When the results in Table 4.7 are compared with the previous results, it is seen that combining wavelet and SDH features together gives slightly better performance compared to only wavelet features (see Table 4.6). On the other hand, it is not a significant improvement when compared to classification performance using SDH features (Table 4.3).

Table 4.7 Classification performance of MLP for combination of SDH and wavelet features.

group	CC	FP	FN	SE	SP	PPV	NPV
G1	97.2637	1.5345	6.4103	93.5897	98.1481	92.4051	98.4520
G2	94.0299	3.1746	13.7931	86.2069	96.1905	86.2069	96.1905
G3	95.5224	1.8229	6.8323	93.1677	97.0954	95.5414	95.5102
G4	98.7562	1.0076	1.3158	98.6842	98.7730	94.9367	99.6904

#### 4.4 Classification Using Two Stage (Cascaded) Network

In previous two sections, two different feature extraction methods; SDH method and wavelet analysis method, are investigated and combination of those features is also tested. Classification network is selected as nonlinear neural networks. Although results are promising, they are not successful enough contrary to expectations. Results give further motivation to extend the study; hence in this section another kind of classifier is investigated.

When previous simulation results are investigated, it is observed that classification of samples from Groups 2 and 3 are relatively difficult. Analyses on misclassified samples showed that pattern distributions of some of the samples from Groups 2 and 3 are quite similar, hence increasing the separability between Groups 2 and 3 arises as a necessity. This requires another type of classifier where correctly classified samples are taken out of the dataset and a different (probably more complex) feature set is used for the rest of the samples at the next step. Therefore new classification scheme is built up by two stages in a cascaded manner (Acir et. al., 2005), (Selver et. al. 2008). First stage includes pre-classifiers which are realized by perceptrons. The second stage, post-classifier is realized by a nonlinear MLP network. Results show that Group 1 and Group 4 are classified at better rates compared to Group 2 and Group 3 because of visual similarity between samples from Groups 2 and 3. Hence first pre-classifier is aimed to classify samples from Group 1 and non-group 1, while the second one is aimed to classify samples from Group 4 and non-group 4. So that post-classifier is mainly responsible to classify samples from Group 2 and Group 3 which are classified as non-group 1 and non-group 4 by the pre-classifiers. Hence the nonlinear post-classifier has emphasis on Group 2 and Group 3 which are difficult to classify. On the other hand, post-classifier has four outputs so that it still has the capability of classifying samples of Group 1 and Group 4 which may have been misclassified by pre-classifiers. Basic scheme of the two stage cascaded classifier network is seen in Figure 4.3.

One of the discrete perceptrons is trained so that its output should be 1 for definite samples from Group 1 and 0 otherwise. Similarly, other perceptron is trained so that its output is 1 for definite samples from Group 4 and 0 otherwise. A sample which produces 0 at the output of both perceptrons is classified as non-group 1 and non-group 4. Those kinds of samples are passed to the post-classifier for the correct classification. Two stages are briefly explained in the following sections.

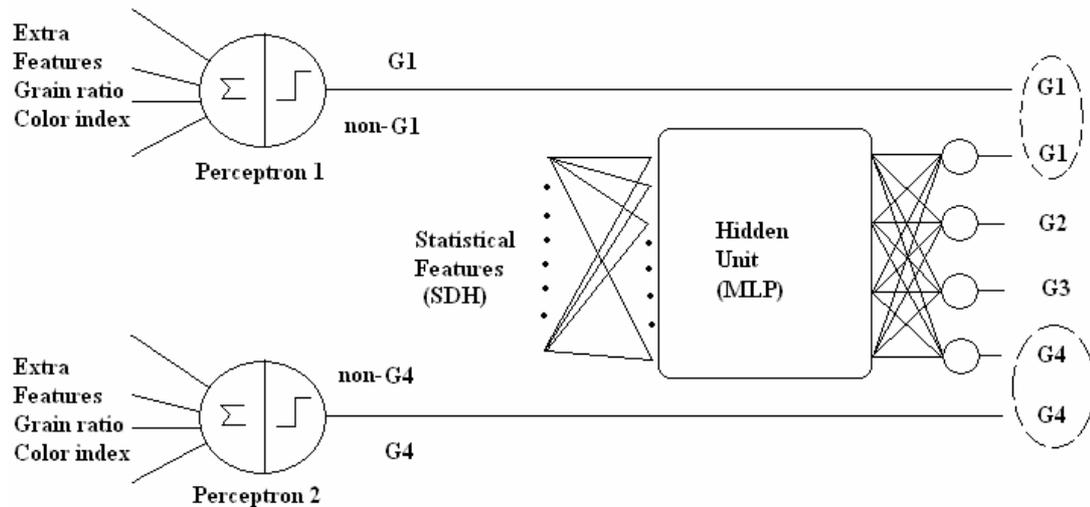


Figure 4.3 Two stage classifier network.

#### 4.4.1 First Stage, Pre-classifier

Discrete perceptrons are used as the pre-classifier which tries to determine a linear separation layer between the classification patterns. Discrete perceptrons use different sets of features for pre-classification. Those are grain area ratio on marble surface which should be high for Group 1 and low for Group 4, and mean values of each color component which give a color index. Color should be another discriminating feature for samples from Group 1 and Group 4 (Group 1 which is limestone, is nearly white while Group 4 which is cohesive matrix intensively, is nearly in brown color) because of materials they contain. Hence there are four different features used for pre-classification.

One of the problems of using perceptrons in this application is that usual perceptron learning algorithm is not proper. It is known that using the four defined

features, classification groups are not linearly separable. In such kind of problems usual perceptron learning algorithm may not reach the optimum solution. Since there is no exact linear separation solution, another modified learning algorithm is needed. Pocket algorithm helps to overcome this bottleneck (Gallant, 1990). In Figure 4.4, an example of linearly non separable sample space is seen. Since there is no linear separation solution, usual perceptron algorithm stops when the specified iteration number is reached. In the end, the obtained solution is random and might be any one of L1, L2 or L3 in Figure 4.4 where L1 is obviously not effective. On the other hand, pocket algorithm has the positive feedback to reach the optimum solution. It keeps the possible solutions (“puts in pocket”) and selects the possible best (optimum) one by means of positive correct classification ratio (L2 in the example).

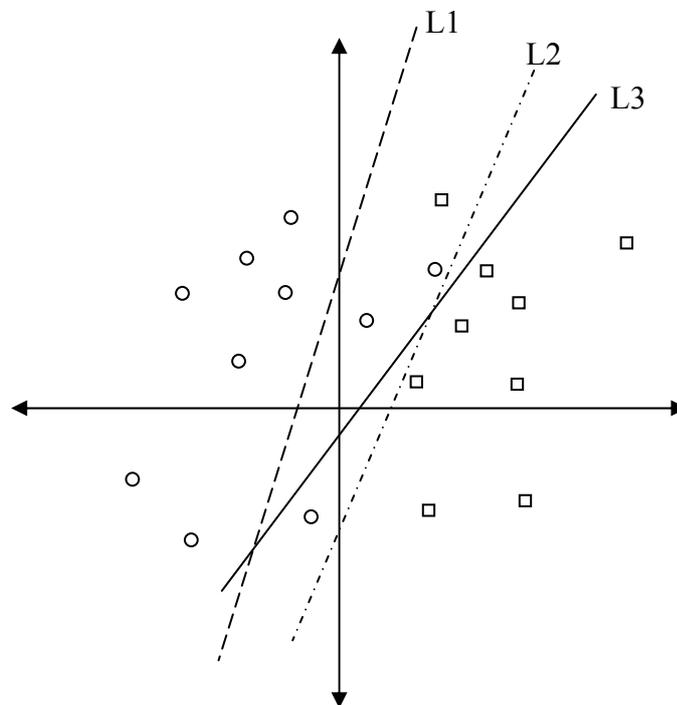


Figure 4.4 Pocket algorithm reaches optimum solution in linearly non separable sample spaces. L2 would be the solution of the pocket algorithm in the example space.

Pocket algorithm is given in Figure 4.5 (Gallant, 1990).

Inputs  $\{x(p) : p = 1 \dots PT\}$ ,  
 Desired outputs  $\{d(p) : p = 1 \dots PT\}$

- 1  $w \leftarrow$  small random values
- 2  $iterations \leftarrow 0$
- 3  $pocketedWeights \leftarrow (0, 0, \dots, 0)$
- 4  $run \leftarrow 0$
- 5  $run_w \leftarrow 0$
- 6 **repeat**
- 7    $modifications \leftarrow 0$
- 8    $iterations \leftarrow iterations + 1$
- 9   **for**  $i \leftarrow 1$  to  $PT$
- 10   **do**
- 11      $p \leftarrow random(1, \dots, PT)$
- 12      $y(p) \leftarrow sign(net(p)) = sign\left(\sum_{k=0}^n w_k x_k(p)\right)$
- 13     **if**  $y(p) \neq d(p)$
- 14       **then**  $modifications \leftarrow modifications + 1$
- 15         $run \leftarrow 0$
- 16        **if**  $(y(p) = 1)$  and  $(d(p) = -1)$
- 17          **then**  $w \leftarrow w - x(p)$
- 18
- 19        **else**  $w \leftarrow w + x(p)$
- 20        **else**  $run \leftarrow run + 1$
- 22     **if**  $run > run_w$
- 23        **then**  $run_w \leftarrow run$
- 24         $pocketedWeights \leftarrow w$
- 25   **until**  $(modifications = 0)$  or  $(iterations = max)$
- 26 **return**  $pocketedWeights$

Figure 4.5 Pocket algorithm.

Pre-classifiers are evaluated using the same training and test sample set given in Table 4.2. Since it is shown that color space has no important effect on the performance only RGB color space is used while evaluating the two stage classifier. Performances of the pre-classifier perceptrons on RGB color space is given in Table 4.8.

Table 4.8 Performance of pre-classifiers.

Pre-classifier	group	CC	FP	FN	SE	SP	PPV	NPV
Percep. 1	G1	95.2736	1.5133	16.9231	83.0769	98.2099	91.8432	96.0265
Percep. 2	G4	99.2040	0.6022	1.0526	98.9474	99.2638	96.9194	99.7537

#### 4.4.2 Second Stage, Post-Classifier

First stage pre-classifiers classify samples from Groups 1 and 4 and thus reduce the number of samples passed onto the post-classifier. Samples which produce 0 outputs at both pre-classifiers are passed onto the second stage. Hence training and test samples of post-classifier are selected as the intersection of the outputs of the pre-classifiers (intersection of non-group 1 and non-group 4 samples which are classified by pre-classifiers). Post-classifier is an MLP type nonlinear classifier and uses the SDH statistical features after dimensionality reduction using PCA. Post-classifier has the same parameters as the networks investigated in Section 4.1.1. Since there is the possibility of misclassification of samples by the pre-classifiers, post-classifier is designed with four outputs. Thus, although post-classifier has main emphasis on Groups 2 and 3, it is also able to classify samples from Groups 1 and 4 which may be misclassified at the first stage. Overall performance of the two stage classifier is given in Table 4.9.

Table 4.9 Performance of two stages (cascaded) network.

group	CC	FP	FN	SE	SP	PPV	NPV
G1	95.7711	3.1169	6.4103	93.5897	96.2963	85.8824	98.4227
G2	92.7861	4.2895	14.9425	85.0575	94.9206	82.2222	95.8333
G3	95.7711	0.7792	8.6957	91.3043	98.7552	98.0000	94.4444
G4	99.7512	0.2494	0.0000	100.0000	99.6933	98.7013	100.0000

#### 4.4.3 A Modified Pre-Classifier for Two Stage Network

It is considered that the first stage in the two stage network should have very low (ideally zero) FP (false positive) rate which means that any of the samples from non-group 1 should not be classified as Group 1 (similar case for Group 4, second perceptron). On the other hand, samples from Group 1 may be classified as non-group 1 (or similarly non-group 4 for the other pre-classifier). So that any misclassified sample from Group 1 or Group 4 is passed onto the nonlinear post-classifier which uses more complicated features and is more accurate compared to pre-classifiers. Hence any misclassified sample from Group 1 or Group 4 has the possibility of being correctly classified by the post-classifier. To achieve zero FP rate a trivial modification is performed on pocket algorithm with the cost of decrease in some performance metrics at the pre-classifiers (such as CC rate). Thus, giving a parallel shift to the separation plane (or line) solution of the pocket algorithm as a result of the trivial modification, no sample from non-group 1 is included in the Group 1 side of the separation plane.

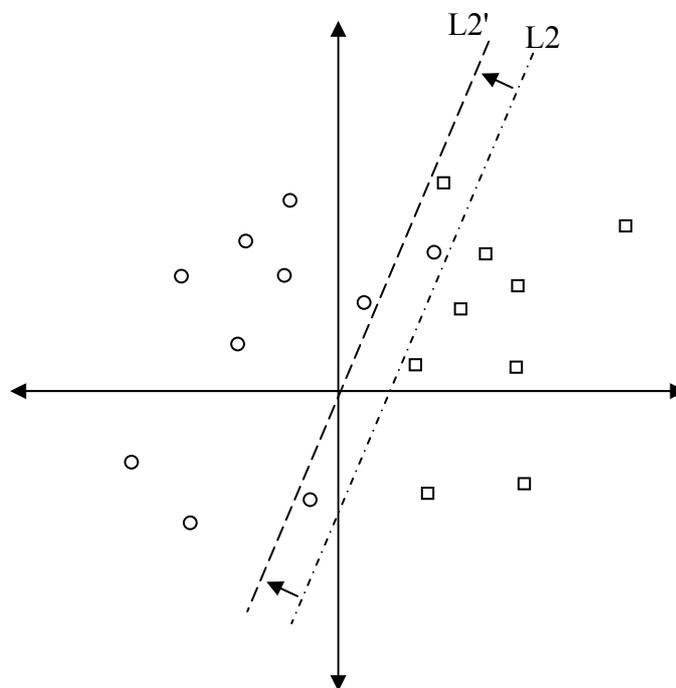


Figure 4.6 Modification on the pocket algorithm solution, giving a parallel shift to the separation plane.

Consider the example in Figure 4.6. Pocket algorithm solution to the given linearly nonseparable space is L2. With the modification on the algorithm, optimum solution line is shifted in a parallel manner so that non of the “□” samples is classified as “○” which is the L2' separation line in Figure 4.6. Resulting cost of the modification is decrease on other performance metrics. Some of the “○” samples are classified as “□” in the example. Those kinds of samples are passed onto the post-classifier for further investigation on the proposed two stage architecture.

Table 4.10 Performance of the pre-classifiers with the modified algorithm.

Pre-classifier	group	CC	FP	FN	SE	SP	PPV	NPV
Percep. 1	G1	93.5821	0.2659	31.7949	68.2051	99.6914	98.1546	92.8699
Percep. 2	G4	99.5025	0.2500	1.3158	98.6842	99.6933	98.6842	99.6933

Performance evaluation results of the two stage network with the modified pocket algorithm are given in Tables 4.10 and 4.11. Table 4.10 shows the effect of modification on the pre-classifier performances. As expected, while FP rate decreases, (especially for the first pre-classifier (perceptron 1)) other performance metrics become worse. On the other hand, as seen in Table 4.11, overall performance of the classifier is improved.

Table 4.11 Performance of two stage (overall) network with the modified algorithm.

group	CC	FP	FN	SE	SP	PPV	NPV
G1	97.0149	0.7692	11.5385	88.4615	99.0741	95.8333	97.2727
G2	93.7811	5.0398	6.8966	93.1034	93.9683	81.0000	98.0132
G3	96.2687	0.7752	7.4534	92.5466	98.7552	98.0263	95.2000
G4	99.5025	0.5000	0.0000	100.0000	99.3865	97.4359	100.0000

## CHAPTER FIVE

### CONCLUSION

In this thesis, image processing and neural networks methods are investigated and applied for marble texture classification. Two basic textural information extraction methods (SDH and wavelet analysis) are investigated to obtain textural descriptors. Feedforward supervised neural networks MLP, RBF and PNN are employed as the classifier network. In the end, a new approach is proposed by designing a cascaded network.

Prior works in the literature show superior results for the SDH method and MLP type classifier (Alajarin, et. al., 2005). Hence the same method was applied to our own database and somewhat worse performance results were obtained. Extended number of samples (1158 samples), number of quality groups (4 quality groups), difference at visual appearance and similar samples belonging to different groups in the database (some samples are critical and difficult to classify, see Figure 5.1 as an example) are the main reasons for the obtained performance results.

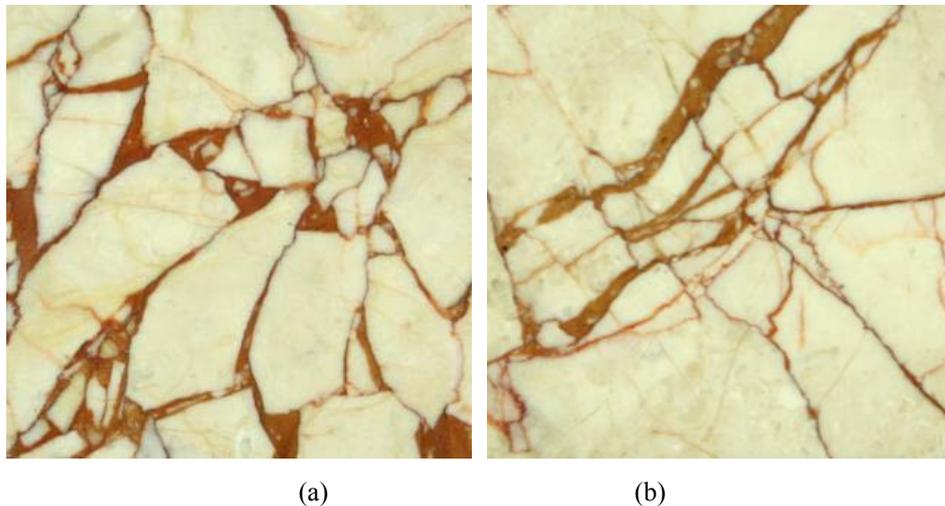


Figure 5.1 Two critical samples from the database; (a) Group 3, and (b) Group 2.

In Section 4.1, SDH method is used as the feature extraction method for different color spaces (RGB, KL, YIQ and XYZ) and different types of neural networks. As seen from the simulation results, color space has no important effect on the

classification performance. Hence RGB color space is used for a fast implementation and to reduce computational complexity (since it is natural capturing color space). On the other hand, MLP type classifier network has slightly better performance in general when compared to RBF and PNN type networks. Also MLP type network has the fast response and less storage advantages while RBF and PNN type networks need more memory (one hidden unit per training sample). Although training MLP network takes more time, since it is only a onetime procedure (performed at the beginning and once) MLP network with the RGB color space descriptors might be more suitable for online systems (i.e. real time classification on an industrial production line) which is not the main research area of this thesis (offline study is performed).

In Section 4.2, another feature extraction method (wavelet analysis) is used to obtain textural descriptors. Classification is performed on only MLP type network, since it is observed that MLP has better performance compared to RBFN and PNN and it also has the advantage of storage. Simulation results show that wavelet features produce no significant difference on classification. On the other hand, SDH and wavelet features are used together in Section 4.3 and similarly no significant difference is observed on the results.

Applied methods are promising and motivate further research. Hence, considering the necessity of increasing the separation between Groups 2 and 3, another classifier with a new approach is designed. A two stage classification network is built with linear perceptrons at the first stage and a nonlinear MLP type network at the second stage. Main aim was to classify samples from Group 1 and 4 with simple perceptrons before the nonlinear classifier. Nonlinear classifier has more emphasis on samples from Groups 2 and 3 which contain the difficult challenging samples (see Figure 5.1). A post-classifier is designed so that it still has the ability of classifying samples from Group 1 and 4. Since the pre-classifier feature space is linearly nonseparable, pocket algorithm, which finds a linear solution trying to improve the correct classification rate, is used. The second step is to modify the pocket algorithm in a trivial manner. The idea is to keep the false positive (FP) rate (ideally) zero at the

pre-classifier stage (which is simple and may classify probably wrong hence we need minimum number of falsely classified samples), since post-classifier is able to also classify Groups 1 and 4. The post-classifier uses more complex features, and thus critical samples from Groups 1 and 4 are classified correctly in addition to samples from Groups 2 and 3. Final simulation results show that modifying the pocket algorithm improves the results as expected but superior improvements cannot be obtained when compared to only MLP type classifier which is examined in Section 4.1.1. When Table 4.3 (RGB color space) and Table 4.11 are considered, correct classification (CC) rate is slightly improved with the two stage (modified pocket algorithm) network. On the other hand, specificity (SP) and false positive (FP) measures are also improved. Especially for Group 1, bad SP and FP results may cause more dramatic drawbacks (i.e. considering use for decorative purposes, any mistaken sample from Groups 2, 3 or 4 wrongly decided as from Group 1 is undesired).

In this thesis, automatic classification of marble slabs is studied using image processing and neural network methods. Different kinds of classifiers and features are investigated to obtain high performance results. Simulations showed that as the typical appearance of the marble slabs change and the number of quality classes increase, classification performance cannot be increased by using (only) textural descriptors. Some morphological features seem to be needed, such as size and shapes of grains, vein orientation and shapes. Those are not covered under the scope of this thesis study and left as future study possibilities.

**REFERENCES**

- Acharya T., & Ray A. K. (2005). *Image Processing Principles and Applications*. NJ: John Wiley & Sons.
- Acir N., Oztura I., Kuntalp M., Baklan B., & Guzelis C. (January 2005). Automatic detection of epileptiform events in EEG by a three-stage procedure based on artificial neural networks. *IEEE Trans. on Biomed. Eng.*, vol. 52, no. 1, pp. 30-40.
- Alajarin J. M., Delgado J. D. L., & Balibrea L. M. T. (November 2005). Automatic system for quality-based classification of marble textures. *IEEE Trans. on Syst., Man, Cyber. - Part C: Appl. and Reviews*, vol. 35, no. 4, pp. 488-497.
- Delgado J. D. L., Alajarin J. M., & Balibrea L. M. T. (May 2003). Classification of marble surfaces using wavelets. *IEEE Electron. Lett.*, vol. 39, no. 9, pp. 714-715.
- Duda R. O., Hart P. E., & Stork D. G. (2001). *Pattern Classification (2<sup>nd</sup> ed.)*. NY: John Wiley & Sons.
- Gallant S. I. (June 1990). Perceptron-based learning algorithms. *IEEE Trans. on Neural Networks*, vol. 1, no. 2, pp. 179-191.
- Gonzales R. C., & Woods R. E. (2002). *Digital Image Processing (2<sup>nd</sup> ed.)*. NJ: Prentice Hall.
- Graps, A. (June, 1995). An introduction to wavelets. *IEEE Computational Sciences and Engineering*, vol. 2, no. 2, pp. 50-61.
- Ham F. M., & Kostanic I. (2001). *Principles of Neurocomputing for Science & Engineering*. NY: McGraw Hill.

- Haralick R. M., Shanmugam K., & Dinstein I. (November 1973). Textural features for image classification. *IEEE Trans. on Syst., Man, Cyber.*, vol. 3, no. 6, pp. 610-621.
- Haykin S. (1999). *Neural Networks: A Comprehensive Foundation (2<sup>nd</sup> ed.)*. NJ: Prentice Hall.
- Hernandez V. G., Perez P. C., Perez L. G. G., Balibrea L. M. T., Puyosa P. H. (October 1995). Traditional and neural networks algorithms: Applications to the inspection of marble slab. *IEEE Int. Conf. on Syst., Man and Cyber.*, 22-25 October 1995, vol. 5, Page(s): 3960-3965.
- Jolliffe I. T. (2002). *Principal Component Analysis (2<sup>nd</sup> ed.)*. NY: Springer.
- MATLAB Neural Networks Toolbox User Guide, version 5 (September 2006), MA: Mathworks.
- Selver A., Ardalı E., & Akay O. (June 2007). Feature extraction for quantitative classification of marbles. *IEEE 15th Signal Processing and Communications Applicaitons, SIU 2007, 11-13 June, Page(s): 1-4*.
- Selver A., Akay O., Ardalı E., Yavuz B., Önal O., Özden G. (June 2008). Cascaded and hierarchical neural networks for classifying surface images of marble slabs. *IEEE Trans. on Syst., Man, Cyber. - Part C: Appl. and Reviews (in revision)*.
- Sousa J. M. C., & Pinto J. C. (October 2004). Comparison of intelligent classification techniques applied to marble classification. *1st International Conference on Image Analysis and Recognition, ICIAR 2004, Page(s): 802-809*.
- Unser M. (Janurary 1986). Sum and difference histograms for texture classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 1, pp. 118–125.

## APPENDIX A

### A.1 Performance Metrics

To measure the performance of the classifier some statistical metrics are used which are given in Eq. A.1 - A.8.

$$\%CC = 100x(TP + TN) / N \quad \text{Correct classification, Accuracy} \quad \text{Eq. A.1}$$

$$\%FP = 100xFP / (TN + FP) \quad \text{False Positive rate} \quad \text{Eq. A.2}$$

$$\%FN = 100xFN / (TP + FN) \quad \text{False Negative rate} \quad \text{Eq. A.3}$$

$$\%SE = 100xTP / (TP + FN) \quad \text{Sensitivity} \quad \text{Eq. A.4}$$

$$\%SP = 100xTN / (TN + FP) \quad \text{Specificity} \quad \text{Eq. A.5}$$

$$\%PPV = 100xTP / (TP + FP) \quad \text{Positive Predictive Value, Selectivity, Precision}$$

Eq. A.6

$$\%NPV = 100xTN / (TN + FN) \quad \text{Negative Predictive Value} \quad \text{Eq. A.7}$$

$$N = TP + TN + FP + FN \quad \text{Eq. A.8}$$

TP is the number of true positives, TPs are the C<sub>x</sub> patterns classified as C<sub>x</sub> patterns (C<sub>x</sub> represents the class number x, x=1, 2, 3, 4). TN is the number of true negatives, TNs are the non-C<sub>x</sub> patterns classified as non-C<sub>x</sub>. FP is number of false positives, FPs are the non-C<sub>x</sub> patterns classified as C<sub>x</sub>. FN is the number of false negatives, FNs are the C<sub>x</sub> patterns classified as non-C<sub>x</sub>.

Importance of the performance metrics may change depending on the classification application. While CC rate is usually important in a simple classification problem (i.e. discrimination of two fish type for canning), FN rate may be another important metric on some applications (i.e. cancer diagnosis).