AUTOMATIC TARGET RECOGNITION IN INFRARED IMAGERY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

TUBA MAKBULE BAYIK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF THESIS
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2004

Approval of the Graduate School of Natural and Applied Sciences

_____

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____

Prof. Dr. Mübeccel Demirekler
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Assoc. Prof. Dr. Aydın Alatan
Supervisor

Examining Committee Members

Prof. Dr. Kemal Leblebicioğlu (Chairman) (METU,EE)_____

Assoc. Prof. Dr. Aydın Alatan (METU,EE)_____

Assoc. Prof. Dr. Gözde Bozdağı Akar (METU,EE)_____

Instructor Çağatay Candan (METU,EE)_____

Assoc. Prof. Dr. Yasemin Yardımcı (METU,IS)_____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Tuba Makbule BAYIK

ABSTRACT

AUTOMATIC TARGET RECOGNITION IN INFRARED IMAGERY

Bayık, Tuba Makbule

M.S., Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Aydın Alatan

September 2004, 66 Pages

The task of automatically recognizing targets in IR imagery has a history of approximately 25 years of research and development. ATR is an application of pattern recognition and scene analysis in the field of defense industry and it is still one of the challenging problems. This thesis may be viewed as an exploratory study of ATR problem with encouraging recognition algorithms implemented in the area. The examined algorithms are among the solutions to the ATR problem, which are reported to have good performance in the literature. Throughout the study, PCA, subspace LDA, ICA, nearest mean classifier, K nearest neighbors classifier, nearest neighbor classifier, LVQ classifier are implemented and their performances are compared in the aspect of recognition rate. According to the simulation results, the system, which uses the ICA as the feature extractor and LVQ as the classifier, has the best performing results. The good performance of this system is due to the higher order statistics of the data and the success of LVQ in modifying the decision boundaries.

Keywords: Automatic Target Recognition (ATR), Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Independent Component Analysis (ICA), Learning Vector Quantization (LVQ)

ÖZ

KIZILÖTESİ GÖRÜNTÜLERDE OTOMATİK HEDEF TANIMA

Bayık, Tuba Makbule

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Doç.Dr. Aydın Alatan

Eylül 2004, 66 Sayfa

Yaklaşık 25 yıllık araştırma ve geliştirme tarihi olan kızılötesi görüntülerde otomatik hedef tanıma konusu, örüntü tanıma ve görüntü çözümleme alanlarının savunma sanayisindeki bir uygulama alanıdır. Otomatik Hedef Tanıma konusu, hala bu alanın ilgi çekici problemlerinden biridir. Bu tez, çarpıcı tanıma algoritmalarının otomatik hedef tanıma problemine uygulamalarını içerir ve konunun keşif çalışması olarak okunabilir. İncelenen algoritmalar, literatürde bu konuda başarılı sonuçlar verdikleri bildirilen çözüm yöntemleridir. Bu çalışmada Ana Bileşen Çözümlemesi, indirgenmiş uzayda Doğrusal Ayırtaç Çözümlemesi, Bağımsız Bileşen Çözümlemesi, en yakın ortalama sınıflayıcısı, K komşu sınıflayıcısı, en yakın komşu sınıflayıcısı ve Öğrenen Vektör Nicemlemesi algoritmaları gerçeklenmiş ve elde edilen sonuçların tanıma oranı açısından gösterdikleri performanslar sunulmuştur. Benzetimlerde özellik bulucu olarak Bağımsız Bileşen Çözümlemesi, sınıflayıcı olarak da Öğrenen Vektör Nicemlemesi metodunun 2 numaralı uzantısını kullanan dizge, en iyi sonuçları vermiştir. Bu

sonuçlar, verilerin yüksek dereceli korelasyonuna ve Öğrenen Vektör Nicemlemesi metodunun karar sınırlarını tanımlamadaki başarısına bağlıdır.

Anahtar Kelimeler: Otomatik Hedef Tanıma, Ana Bileşen Çözümlemesi, Doğrusal Ayırtaç Çözümlemesi, Bağımsız Bileşen Çözümlemesi, Öğrenen Vektör Nicemlemesi

For mommy and her Gandhi who kept us alive

ACKNOWLEDGEMENTS

I am greatly thankful to my supervisor Assoc. Prof. Dr. Aydın Alatan for his remarkable affect on every step of this thesis. Throughout this period, by his appreciated advices, thought systems and cheerful personality, had I the chance of improvement.

I would like to thank Salim Sırtkaya for his cooperation, support and precious friendship right from the beginning of the master's period.

My parents, my aunt PerÇet, various friends - now all over the world, and my dearest 'fearless crab' Pelin, thanks for 'bearing my black face'.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ATR	Automatic Target Recognition
DR	Dimension Reduction
IC	Independent Component
ICA	Independent Component Analysis
IR	Infrared
LDA	Linear Discriminant Analysis
LVQ	Learning Vector Quantization
OLVQ	Optimized Learning Vector Quantization
MFC	Microsoft Foundation Class
NN	Nearest Neighbor
PC	Principal Component
PCA	Principal Component Analysis

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION AND MOTIVATION

Automatic target recognition (ATR) task, also can be referred to as *weapon vision* [8], is one of the challenging problems of the defense industry. The aim of an ATR system is to remove the role of man from the process of target detection and recognition and hence, implementing a real-time and reliable system of high performance.

The automatic target recognition term originated with the Low Altitude Navigation and Targeting Infrared for Night (LANTIRN) program in the early 1980's. Prior to the LANTIRN program, little had been done in the area, which became known as ATR [1].

In the most general sense, an ATR system is composed of a target detector and recognizer. Such an inclusive system of ATR covers the problems of preprocessing, detection, segmentation, classification, tracking and aim-point selection. Figure 1-1 represents the block diagram of such a typical system.

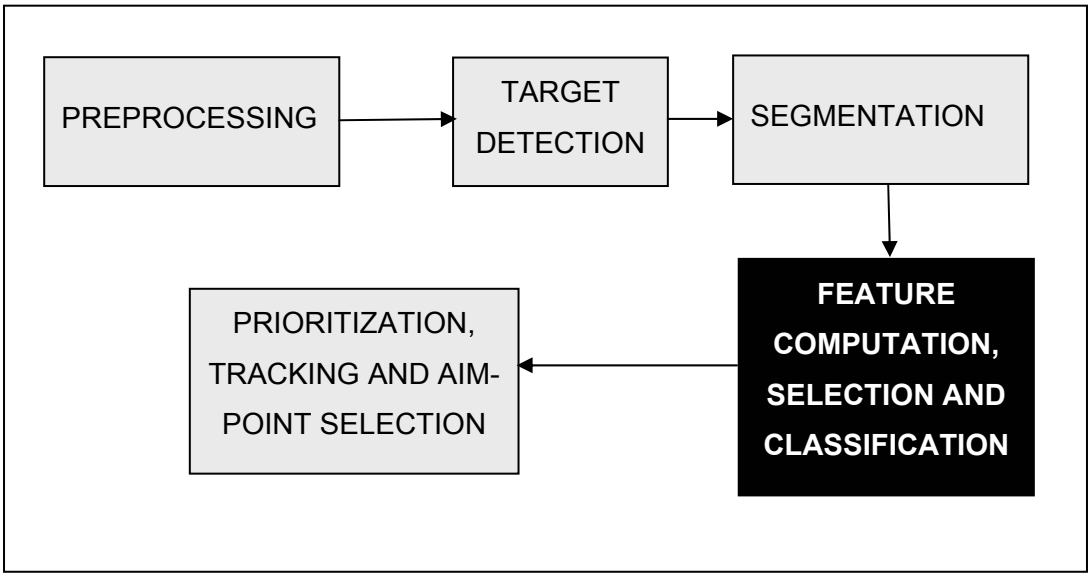

Figure 1-1: Basic block diagram of a typical ATR system. Black box represents the scope of this thesis

Briefly, the preprocessing is the step that improves target contrast and reduces noise and clutter present in the image. Examples of preprocessing functions are noise suppression, focus control, adaptive contrast enhancement, etc [7].

Target detection is the process of localizing those areas in the image where a potential target is likely to be present. Conventionally, the detection techniques are based on the contrast between target and its immediate background. Most of the techniques can be adapted to detect either light or targets. Once a potential target is localized, it is extracted from the background as accurately as possible in the segmentation step.

Feature computation and classification is the process of associating detected targets with target classes. Throughout this thesis, ATR refers to the classification task of infrared images, specifically, to the decision algorithm, that classifies the still images or video sequences into one of the predefined target classes using the information extracted through the examination of the pre-collected data, so-called the training set.

Prioritization is the process of assigning priorities to the targets in the field of view. This information, which is pre-stored, is normally based on the type of the target and the probability of its correct classification. The prioritized target is then tracked. Aim-point selection involves the determination of the critical aim-point of a target. A stored feature vector corresponding to the target class and aspect is used for aim-point designation.

ATR systems, combining different sensor outputs, such as visual images or SAR images, can be designed to increase the overall performance. Parallel algorithms utilizing semantic, contextual and structural information would perform well.

An ATR system may be used as guidance to armed forces operating under inferior weather conditions or at night. Moreover, with ATR, remotely controlled vehicles or cruise missiles are also designed. Remotely piloted vehicles are another application area of the ATR systems. The performance of such systems is dictated by the large volumes of data requiring analysis and by the short timelines required by the target acquisition scenarios. ATR is a high leverage technology, and the challenge is to go beyond human-aided capabilities to automatic, autonomous systems [8].

1.2 CHALLENGES IN ATR IN INFRARED IMAGERY

Infrared imagery contains the thermal radiations emitted by objects. It is an effective method to cluster heat-generating targets. Main advantages of IR sensors are that they are passive and involve one-way propagation. Moreover, they penetrate fog, haze and dust, operate at day and night, and therefore offer good performance even under inappropriate operating conditions.

Figure 1-2 shows a typical IR image of size 240x320 used in this study. At the center of the image, there exists a loaded truck with white (hot) tires. At the background, buildings with hot windows and vegetation are observed.

Figure 1-2: An example of IR image

The biggest handicap of utilizing the IR imagery reported in the literature and encountered during this study is the difficulty to obtain such images, moreover there is no canonical and public data set to evaluate the performance of the implemented algorithms. The implemented ATR systems, developed with a limited data set might result in high false alarm rates in practical applications. The reason for this inconsistency lays under the sensor deficiencies and the infrared signature characteristics of the targets and the background. ATR is particularly difficult if there is structured background and low signal-to-noise ratio. The viewing direction of the target is another issue to be considered.

IR images of targets exhibit seasonal variations and changes according to the time of the day. The image of the same target under identical conditions may even vary according to the recent history of the target (the tires of a truck get brighter after long rides, the heaters inside may be on or off etc.). Another limitation with the IR is the occlusion problem. The vegetation and terrain may obscure the targets in infrared imagery, whereas there are different sensing systems to penetrate, such obstacles such as SAR. Therefore, IR images are relatively high in irrelevant information and variability.

1.3 OUTLINE OF THE THESIS

Chapter 2 represents a brief survey of ATR methods. The algorithms standing out among the recently developed algorithms each with a brief explanation are involved.

Chapter 3 includes the dimension reduction methods, which is critical for the application, since the system is implemented under a limited data set and the target images constitute high dimensional data involving great amount of irrelevant information. Among the dimension reduction methods, recognized methods of PCA, LDA and subspace LDA, as well as the relatively new topic of ICA, are discussed in detail.

Chapter 4 is devoted to the pattern classifiers. Pattern classification basics, traditional methods of NN, k-NN, nearest mean and k-Mean classification are overviewed and details of LVQ method are conferred.

In Chapter 5, the performances of the algorithms, which are discussed in detail, are analyzed graphically and with tables.

Finally, in Chapter 6, the thesis gets to a conclusion and simulation results are compared.

CHAPTER 2

OVERVIEW OF AUTOMATIC TARGET RECOGNITION METHODS

There are plenty of algorithms developed and being developed in the area of ATR. According to [9], these algorithms fall into one of the three general categories of statistical, neural and model based approaches. In the first two categories, the information to be used in the classification task is implicitly extracted, while in the latter one, a model database is constructed either with CAD tools or from the real data, and then comparison is achieved among these templates.

In the following subsections, the prominent algorithms of these categories are summarized.

2.1 STATISTICAL METHODS

Statistical Pattern Recognition is one of the major approaches in pattern recognition discipline. In statistical methods, a target image is represented with a set of characteristic measurements, called *feature vectors* [10]. Features constitute points in a d-dimensional space, and the feature space is formed in a decision theoretic basis with the a priori knowledge of underlying distribution and/or the statistical properties of a set of known samples, namely the training set. The goal is to form the feature space such that the features of the targets belonging to different classes are clustered at different regions of the feature space. Statistical methods include projection based methods, such as Principal Component Analysis, Linear

Discriminant Analysis and Independent Component Analysis, etc, each of which will be explained in this thesis.

In ATR, patterns are images of targets and this constitutes a very high dimensional space of vectors. The aim for using statistical methods in ATR is reducing the dimension and overcoming the problems of the limited data sets with which the feature extractors are trained. Statistical methods are especially satisfactory for patterns with well-behaved distributions.

- *Principal Component Analysis:*

Principal Component Analysis (PCA) method is a conventional method of face recognition and is widely applied in to appearance-based recognition [4]. This method is applicable to the ATR problem provided that a region of interest is detected within the input image [9]. PCA is based on the correlation between image pixels, and its utility to image analysis is in part due to the correlation between nearby pixels in a real-world image. The idea behind PCA is to express the data in a lower dimensional space with the minimum error in the least square sense. This means that the criterion of the feature extraction with PCA is the minimization of the reconstruction error. It should be noted that PCA is an unsupervised method. The details and mathematical reasoning of PCA are further discussed in Chapter 3.

- *Linear Discriminant Analysis:*

Linear Discriminant Analysis (LDA), just like PCA, aims to reduce the dimensionality of the data and searches for appropriate axes to project the data onto [6]. However, in LDA; the criterion of feature extraction is the separability of different classes. LDA is based on the analysis of within and between scatter matrices of the data. It is a supervised method, since the data are analyzed according to the classes, which they belong to. The LDA method is thoroughly discussed in Chapter 3.

2.2 NEURAL NETWORK METHODS

Neural network systems are inspired from the human neural system. ATR needs methods to represent targets and backgrounds that are both sufficiently descriptive, yet robust to signature and environmental variations. Neural networks offer potentially powerful collective-computation techniques for designing special purpose hardware, which can implement fast optimization for a number of computational vision and multi-sensor fusion methods [1]. The existence of powerful learning algorithms is one of the main strengths of the neural network approach. There are a number of neural network inspired techniques, which can be used for the selection or development of maximally discriminating feature sets. The classical multilayer perceptron, convolutional neural networks, modular neural networks are examples of neural techniques in ATR literature. The major drawback of the NN systems is the *overfitting problem* [14]. Overfitting means that the solution fits the training data, but it does not represent the underlying function. In other words, algorithm memorizes the system rather than generalizing it.

- Convolutional Neural Network:

Convolutional Neural Network (CNN) is reported to be an effective method for character recognition [2]. This neural approach then applied in face recognition and also ATR problems.

CNN is an adaptation of the well-known multilayer back-propagation network which is an accepted technique of recognizing high dimensional data. CNN compensates the deficiency of the back propagation network for the invariance with respect to translations or local distortions. The robustness is achieved via implementing three ideas: local receptive fields, shared weights and spatial sub-sampling.

In Figure 2-1, an illustrative CNN structure is shown. The network consists of an input layer, several hidden layers, and an output layer.

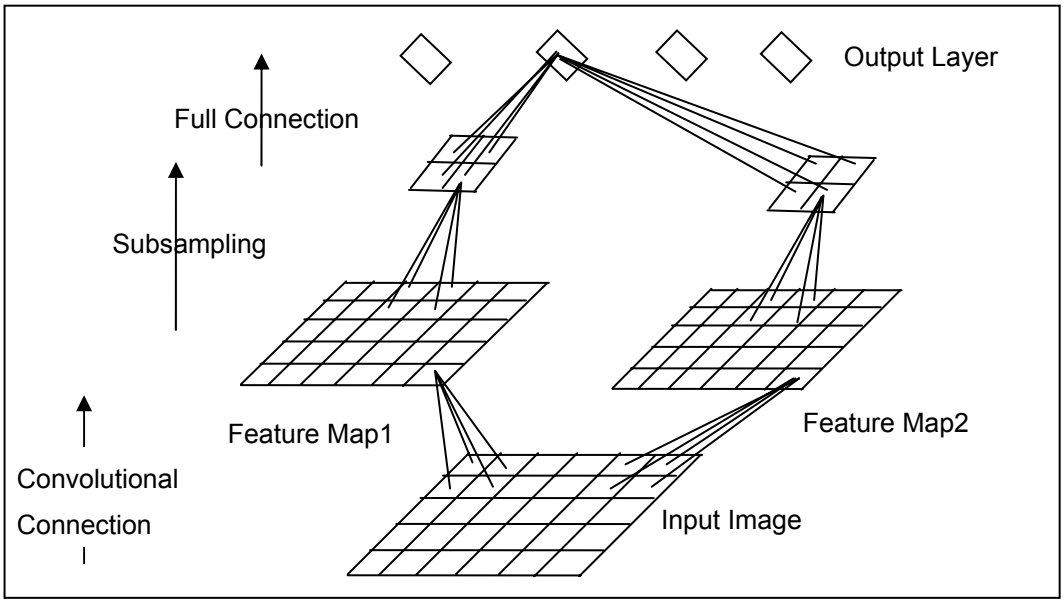

Figure 2-1: CNN configuration

In CNN, the image itself constitutes the input layer. Each node corresponds to a pixel in the input image. The hidden layers receive input from a small neighborhood of the previous plane. This is denoted as *local receptive fields* in the preceding paragraph. Shared weight means that all nodes of a plane use the same weights. In order to extract various features, multiple planes can be used in each plane. Sub-sampling step is held out via another hidden layer where local averaging and sub-sampling operation is carried out. Notice that, sub-sampling reduces the resolution of the feature map, consequently, it reduces the sensitivity of the output to shifts and distortions.

In the application of CNN to ATR [9], the convolutional kernel is designed as a Canny edge detector. In this approach, different convolutional kernel sizes are utilized, specifically kernels of size 3x3, 5x5 and 3x5. The kernels of different sizes are employed to detect features at different scales and orientations.

- Modular Neural Network

Modular Neural Networks (MNN), involves a hierarchical neural network architecture using a mixture of expert modular neural network, with each expert

consisting of a committee of neural networks [15]. In the mixture of expert modular network, each expert is trained for a particular subset of data vectors (target region). A gating network is then trained to select or combine the outputs of expert networks to form the final output. The partitioning of the dataset into several subsets is based on the similarity of target silhouettes. The partitioning is guided by intuition and confirmed by experiment. In the committee of networks, each member network receives distinct inputs, which are features extracted from one local region of the target image.

Apparently, this method is not shift-invariant, a centering algorithm preceding the MNN upgrades the performance. The main advantage of this method is its robustness to occlusion, since the input is a particular region of the target image. It should also be noted that decomposing the image results in simpler networks, which allow better generalization, and saves processing time.

Figure 2-2 shows the MNN-based approach introduced by [16]. The image is partitioned into six disjoint regions. A sub-sampled version of the input image is also added as the input to the network, and it is partitioned. The committee of networks receives features of directional variance in 5x5 image blocks. The classification decisions of the individual committee members are combined using *stacked generalization*. Stack generalization is a scheme for minimizing the generalization error rate [17].

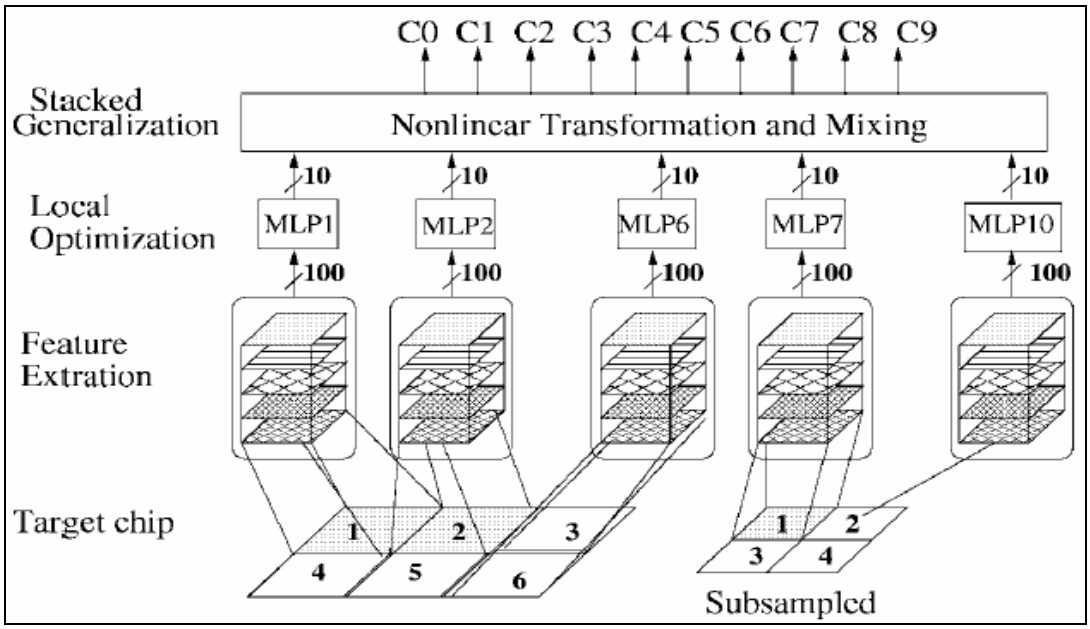

Figure 2-2: The MNN-based scheme offered for ATR

- Learning Vector Quantization :

Learning vector quantization (LVQ), as discussed thoroughly in Chapter 4, is a popular method of globally modifying decision boundaries. LVQ has an iterative learning scheme in which a number of class representatives are searched to define the distribution of the data as accurately as possible especially at the class boundaries. After the learning procedure is completed, the test procedure is a simple nearest neighbor classifier.

LVQ is proposed by [15], [16] to classify targets in IR imagery. In their methods, training images are separated into target-aspect groups. Each target-aspect group contains one target type within a restricted range of viewing angles. The training images are then decomposed into wavelet sub-bands. Each wavelet sub-band of each target-aspect group is clustered using the k-means algorithm [11] in order to create a set of code vectors. The LVQ algorithm is then applied to the code vectors to enhance discriminatory ability.

2.3 MODEL-BASED METHODS

Model based methods are based on physical properties of sensor, scene and the target. In model-based approaches, first, a model database is constructed, and then local features extracted from an image are compared with the prestored target models. This technique reduces dependence on large training sets. However, practical modeling of spectrum of operational reality is far from solved. There are different approaches in model-based methods, such as Hausdorff metric-based matching and geometric hashing.

- Hausdorff Metric Based Matching:

The Hausdorff distance is a measure of dissimilarity of two sets of points in their least similar matches. It is defined to be the maximum of the minimum distances from all members of two point sets.

Formally, if A and B are two finite sets of points, the Hausdorff distance is defined as

$$H(A,B) = \max\{h(A,B), h(B,A)\} \tag{2-1}$$

where h(A,B) is the norm of distances between points of B in the neighborhood of A and the points of A.

The Hausdorff distance does not involve any point matching algorithms. As an ATR algorithm, the test image is assigned to the label of the class whose template is closest in the Hausdorff sense, with the intensity edges of images and templates being used as the point sets.

- Geometric Hashing:

Geometric hashing is another method of representing and matching the data on the basis of affine transformations and hash tables. The main benefit of this scheme is the invariance under affine transformations. It is based on an intensive offline model learning stage, where model information is indexed into a hash table using minimal transformation invariant features. It is reported to be a

successful recognition method of both 2-D and 3-D objects in cluttered scenes from an arbitrary viewpoint [21]. The mathematical approximations, explained in the following paragraph, on which the approach is based, are especially suitable for bodies, which are relatively far from the camera.

Given two different images of the same flat object, we may assume that, there exists a non-singular 2x2 matrix **A** and a 2-D translation vector **b**, such that each point **x** in the first image is translated to the corresponding point **Ax**+**b** in the second image.

Assume that the model objects and scenes are described by sets of interest points, like corners, end points, which are invariant under affine transformation. Now, the recognition problem is a point-set matching task, where one is given a set of model point-sets and an observed point-set. We look for a transformed subset of some model point-set, which matches a subset of the observed point-set.

For the geometric representation of the objects, assume that an arbitrary set of m points belonging to a rigid body are given. The three ordered non-collinear points (e_0, e_1, e_2) define an affine basis under which any feature point P can be represented by a doublet (α, β) satisfying $P = \alpha(e_1 - e_0) + \beta(e_2 - e_0) + e_0$. Hence, a representation, which is invariant under affine transformations, is obtained.[21] Accordingly, the m points are represented by their coordinates in the affine basis triplet. Representing object points by coordinates in all possible affine bases removes the dependency of the algorithm to the basis, therefore, occlusion problem is overcome. For each affine basis, the coordinates (α, β) of all other m-3 model points in the affine coordinate frame defined by the basis triplet, are computed. After each such coordinate are quantized, they are used as an index to a hash table, and the (model, basis) pair is recorded there.

Given the test image, first, features are extracted (assume n features are extracted). Test data is associated with the label of the model that gets the most of the entries to its location in the hash table.

2.4 DISCUSSION

Other than presented here, there are also several variations of such algorithms proposed for the ATR problem. However, no single approach is likely to be the solution to all ATR problems [34], [35], [36], [37], [38]. Nevertheless, by applying the most useful techniques to each part of the problem, the progress is accelerating. The most successful ATR systems will probably blend several algorithmic techniques to achieve satisfactory performance. Due to the nature of IR signature, the performance of the system is heavily dependent on operating conditions.

Under the operating conditions of the ATR systems developed in this study, there are no occluded parts of the targets. The view angle under which data is collected is fixed, but there are some seasonal changes between data. Under such conditions, the canonical pattern recognition approaches are likely to perform well. In any algorithm, due to the high dimensionality of the data, the curse of dimensionality problem is still an important issue. Overfitting and computational problems are to be considered. Some dimension reduction techniques are implemented to overcome this issue. LVQ, as one of the best methods of decision boundary representations, is implemented throughout this study, as well as some conventional classifiers, to evaluate the system experimentally.

CHAPTER 3

AUTOMATIC TARGET RECOGNITION WITH DIMENSION REDUCTION

3.1 MOTIVATIONS IN DIMENSION REDUCTION PROBLEM

Due to advances in data collection capabilities, researchers in such domains of engineering, astronomy, economics, statistics encounter an increasing number of variables associated with each observation. These high dimensional datasets present many mathematical challenges, as well as some opportunities. As an important problem in high dimensional datasets for many cases, not all the measured variables are 'important' for understanding the underlying phenomena of interest [19].

Having large amounts of high dimensional sensory data to process, analyze or store, dimension reduction is needed for

- Visualization
- Data compression for transmission or storage
- Decreasing computation time and memory usage
- Change of representation for statistical pattern recognition and modeling

Dimension reduction is the problem of finding a k-dimensional representation of a d-dimensional random variable, with $k<d$, that captures the content in the original data with respect to some criterion.

There exist different criteria for each dimension reduction problem, such as minimizing the reconstruction error, preserving distances or maximizing likelihood with respect to some model [18].

Within the context of pattern recognition, high dimensionality introduces the well-known limitation, which is denoted by [10] as “the curse of dimensionality”. For linear or quadratic classifiers, the required number of training samples depends linearly or quadratically on the data dimensionality. Furthermore, the training sample set size needs to increase exponentially, in order to effectively estimate the multivariate densities needed to perform nonparametric classification [6].

In the following chapters, the most common linear approaches to avoid the problem of dimensionality, namely Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) and a relatively new approach, Independent Component Analysis (ICA) are studied.

3.2 PRINCIPAL COMPONENT ANALYSIS

In the mean squared error sense, Principal Component Analysis (PCA) is the best linear dimension reduction technique. It is also known as the Singular Value Decomposition or the Karhunen-Loève Transform [19]. Since it is based on the covariance matrix of the variables, it is a second order method, therefore PCA considers the pair-wise relationships between variables of observation set (e.g. pixels in the image database).

PCA looks for orthogonal basis functions for which the components of the signal are uncorrelated. The main aim of the PCA is to reduce the dimensionality by finding orthogonal linear combinations of the original variables with the largest variance. An N-dimensional random variable has N principal components. However, for many datasets, the first several principal components retain most of the variance, so that the rest can be discarded with minimal loss of information.

In order to rephrase the PCA method in mathematical terms, consider M observations of N-dimensional random variable $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & . & . & . & x_N \end{bmatrix}^T$ with the

i^{th} observation being denoted as $\mathbf{x}^{(i)}$. $\mathbf{x}^{(i)}$ can be expanded in any set of orthonormal basis vectors $\boldsymbol{\varphi}_j$ as

$$\mathbf{x}^{(i)} = \kappa_1^{(i)}\boldsymbol{\varphi}_1 + \kappa_2^{(i)}\boldsymbol{\varphi}_2 + ... + \kappa_N^{(i)}\boldsymbol{\varphi}_N \tag{3-1}$$

where orthonormality indicates the relation

$$\boldsymbol{\varphi}_i{}^T\boldsymbol{\varphi}_j = \begin{cases} 1, i = j \\ 0, i \neq j \end{cases} \tag{3-2}$$

It is further desired that the coefficients κ_i satisfy the relation

$$\mathrm{E}\{\kappa_i\kappa_j\} = \begin{cases} \varsigma_i^2, i = j \\ 0, i \neq j \end{cases} \tag{3-3}$$

which means that the coefficients are uncorrelated, if the random variable has zero mean (otherwise, they are statistically orthogonal). It can be shown that [3] the eigenvectors of the covariance matrix (correlation matrix for random variables with nonzero means) are the unique solution to $\boldsymbol{\varphi}_i$ to satisfy the desired conditions.

A further property of this decomposition is the equality between the total variation and the sum of the eigenvalues λ_i of the covariance matrix. Truncating the expansion to use some number P(<N) of basis vectors, the average energy in the error process, referred to as *the mean-square error*, equals [19]

$$\sum_{i=P+1}^{N} \lambda_i \tag{3-4}$$

If the eigenvectors, corresponding to the largest P eigenvalues of the covariance matrix are preserved, the optimal representation with P basis vectors, with respect to the mean-square error, is achieved.

Notice the following practical issues: When the density function of the random vector **x** is not known, the expectations and moments can be simply estimated from the data. The mean of **x** can be estimated as

$$\mathbf{\Psi} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}^{(i)} \tag{3-5}$$

and defining the MxN zero mean data matrix to be

$$\mathbf{X} = \left[\mathbf{x}^{(1)} - \mathbf{\Psi} \quad \mathbf{x}^{(2)} - \mathbf{\Psi} \quad \ldots \quad \mathbf{x}^{(M)} - \mathbf{\Psi}\right] \tag{3-6}$$

the covariance matrix can be estimated as

$$\mathbf{C} = \frac{1}{M}\mathbf{X}\mathbf{X}^{(T)} \tag{3-7}$$

In Figure 3-1 and Figure 3-2, some typical IR target image regions are illustrated, as well as their mean image.

Figure 3-1: Examples of IR target images

Figure 3-2: Mean image of target images

In the task of target recognition, target images are random vectors to be analyzed. Note that images, which are 2-dimensional arrays of size N_xxN_y can be represented as vectors of dimension N_x*N_y . Typically, images constitute a high

dimensional space (40*80 in our implementation), and M, the number of observations, is far less than the image space dimensions (250 in our implementation).

Computationally, it is quite costly to determine the eigenvectors of the N_x*N_y by N_x*N_y covariance matrix that is formed for images of typical sizes. Furthermore, only M – 1 eigenvalues will be nonzero, consequently having meaningful corresponding eigenvectors. In their task of face recognition, Turk and Pentland [4] offered an elegant remedy in which the eigenvalue decomposition is carried out on MxM matrices. Their methodology is as follows:

Consider again the zero mean data matrix **X** and the associated covariance matrix **C,** defined in the previous paragraphs. Now, consider the eigenvectors $\mathbf{v}_i$ of $\mathbf{X}^T\mathbf{X}$:

$$\mathbf{X}^T\mathbf{X}\mathbf{v}_i = \mu_i\mathbf{v}_i \tag{3-8}$$

After multiplying the equation with **X**,

$$\mathbf{X}\mathbf{X}^T\mathbf{X}\mathbf{v}_i = \mu_i\mathbf{X}\mathbf{v}_i \tag{3-9}$$

it is straightforward to rewrite the equation in terms of the covariance matrix **C**

$$\mathbf{C}\mathbf{X}\mathbf{v}_i = \mu_i\mathbf{X}\mathbf{v}_i \tag{3-10}$$

and observe that $\mathbf{X}\mathbf{v}_i$ corresponds to φ_i , the eigenvectors of **C**.

Therefore, to extract the eigenvectors of **C** which correspond to the nonzero eigenvalues, extracting the eigenvectors of MxM dimensional matrix $\mathbf{L} = \mathbf{X}^T\mathbf{X}$, and then multiplying the eigenvectors of **L** with **X** avoids a significant amount of computational effort. Depending on the constraints of the problem, the dimension can further be decreased to some number $P \leq M$ by taking into account only the P highest valued eigenvalues.

After the basis vectors are obtained, the variation of any image $\mathbf{I}$ from the mean image $\boldsymbol{\Psi}$ can be projected onto the new coordinates, and a lower dimensional feature vector $\boldsymbol{\kappa}$ can be obtained. Defining a basis matrix $\boldsymbol{\Phi} = \left[\boldsymbol{\varphi}_1 \quad \boldsymbol{\varphi}_2 \quad ... \quad \boldsymbol{\varphi}_P\right]^T$, the feature vector of the image $\mathbf{I}$ can simply be calculated via a matrix multiplication:

$$\boldsymbol{\kappa} = \boldsymbol{\Phi}\left(\mathbf{I} - \boldsymbol{\Psi}\right) \tag{3-11}$$

Figure 3-3 shows the first 15 eigenimages $(\boldsymbol{\Phi}_1 ... \boldsymbol{\Phi}_{15})$ of the data set, whose typical examples are shown in Figure 3-1.

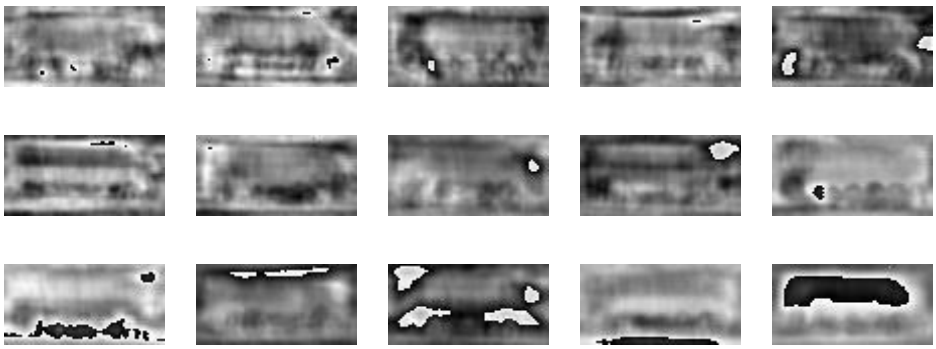

Figure 3-3: First 15 eigen images of the data set

3.3 SUBSPACE LINEAR DISCRIMINANT ANALYSIS

As stated in the previous section, given M observations of an N-dimensional random vector, PCA aims to code the data to retain most of the variance of the data. In PCA, while searching for the optimal representation, all observations are considered to belong to a single distribution and the class information to which the observations belong is not utilized. Unlike PCA, Linear Discriminant Analysis (LDA) searches for the orientations for which the projected data for each class are well separated. Therefore, a measure of discrimination between projected data is optimized. The most fundamental and widely used technique is the Fisher's linear

discriminant [6],[10], in which the objective function is stated in terms of the scatter of the projected points.

For the data to be well separated, the means of the classes should be distant enough with respect to their variations. More clearly, the data of the same classes is desired to be dense while the means of classes are scattered widely. A measure of the scatter within class χ_i can be obtained via the scatter matrix, defined as:

$$\mathbf{S}_i = \sum_{\mathbf{x}\in\chi_i} (\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^T \tag{3-12}$$

where $\mathbf{m}_i$ stands for the sample mean

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x}\in\chi_i} \mathbf{x} \tag{3-13}$$

For a c-class problem, the within-class scatter matrix is defined to keep track of the scatter within all of the classes:

$$\mathbf{S}_W = \sum_{i=1}^{c} \mathbf{S}_i \tag{3-14}$$

Moreover, the scatter between classes is measured as the distinction between the means of the classes and is represented via the between-class scatter matrix:

$$\mathbf{S}_B = \sum_{i=1}^{c} n_i(\mathbf{m}-\mathbf{m}_i)(\mathbf{m}-\mathbf{m}_i)^T \tag{3-15}$$

If the projection is represented in matrix form with matrix W, such that the projected samples **y** are computed as $\mathbf{y}=W^T\mathbf{x}$; the within-class and between-class scatter matrices after the projection are easily shown to be

$$\tilde{\mathbf{S}}_W = W^T \mathbf{S}_W W \text{ and } \tilde{\mathbf{S}}_B = W^T \mathbf{S}_B W \tag{3-16}$$

A simple scalar measure of scatter is the determinant of the scatter matrix, and utilizing of all the above definitions, yield the following criterion function

$$J(W) = \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|} \tag{3-17}$$

where $|\bullet|$ denotes the determinant operation.

It can be shown [10] that the columns of an optimal W are generalized eigenvectors that correspond to the largest eigenvalues in

$$\mathbf{S}_B \mathbf{w}_i = \lambda_i \mathbf{S}_W \mathbf{w}_i \tag{3-18}$$

The subspace LDA method simply makes use of the above discussion of LDA on the data which is projected onto its Principal Components of Section 3.2 beforehand. As stated in [5], the combination of these two methods solves the generalization/overfitting problem based on the training samples to new testing samples.

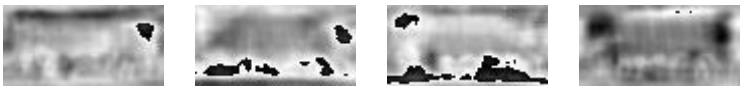

Figure 3-4: Subspace LDA images of the data set

3.4 INDEPENDENT COMPONENT ANALYSIS

Independent Component Analysis (ICA) is a more recently developed method of linearly representing the multivariate data. In ICA, the aim is to minimize the statistical dependence of the components of a representation [28]. The method

is applicable to many different problems, such as blind source separation, blind deconvolution, and feature extraction. Independence is much stronger than the uncorrelatedness sought in the classical method PCA, since uncorrelatedness involves only second order statistics, while independence involves all the higher order statistics. Actually, the main aim of ICA is not necessarily dimension reduction and there are overcomplete versions of ICA where the number of ICs is even larger than the dimension of the data [26].

Similar to the previous formulation, consider a set of observed random vectors **x** of dimension N. Assume there exists an unobservable set of random variables $\mathbf{s} = \begin{bmatrix} s_1 & s_2 & \dots & s_P \end{bmatrix}^T$, which are statistically independent and are mixed using an unknown linear transformation to form observable random variable **x**

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{3-19}$$

Note that, the set of random variables s_i are denoted as statistically independent, if their joint probability density function is the product of each variable's marginal density function, i.e.

$$f(s_1, s_2, ..., s_P) = f(s_1) f(s_2) ... f(s_P) \tag{3-20}$$

ICA aims to estimate the independent random vectors $\mathbf{s}_i$, referred to as *hidden variables* or *sources*, and the *mixing matrix,* **A,** from the observations of **x**. In the estimation procedure the mixing matrix is assumed to be square for simplicity. After estimating **A**, its inverse **W** can be computed and an estimate of the independent components can be obtained

$$\mathbf{y} = \mathbf{W}\mathbf{x} \tag{3-21}$$

It is well known that [30], given two independent random variables $\mathbf{s}_1$ and $\mathbf{s}_2$, their sum $\mathbf{s}_1+\mathbf{s}_2$ is more Gaussian than the participating variables $\mathbf{s}_1$ and $\mathbf{s}_2$. Among all the variables, which are generated by summing these two variables, the sum $1x\mathbf{s}_1+0x\mathbf{s}_2$ is the least Gaussian. Therefore, given a set of observed random variables, which are generated from some independent components, these

independent components can be estimated by maximizing a measure of nongaussianity.

Information theory might be utilized as the main guide for studying independence. Information theory introduces the concept of *entropy*, a quantity of average information that the observer obtains via a random variable, $\mathbf{x}$, [23], [30]:

$$H(\mathbf{x}) = -\int \log(f(\mathbf{x}))f(\mathbf{x})d\mathbf{x} \tag{3-22}$$

Considering the definition of entropy, *mutual information* is defined

$$I(\mathbf{x},\mathbf{y}) = H(\mathbf{x}) + H(\mathbf{y}) - H(\mathbf{x},\mathbf{y}) \tag{3-23}$$

This quantity can be interpreted as a measure of information about **y**, that is obtained by observing the random variable **x**. It should be noted that, I(**x**,**y**)=0, if **x** and **y** are statistically independent.

It is also proven that [29] if g is the cumulative density function of the ICs generating the random variable, maximizing the joint entropy of **Y** = g(**U**) minimizes the mutual information between estimates of the ICs $\mathbf{u}_i$. In other words, the entropies of $\mathbf{y}_i$ tend to be minimized, as the joint entropy of **Y** is maximized. For a given covariance matrix, it is well known that the Gaussian distribution has the maximum entropy. Minimizing mutual information is roughly equivalent to minimizing the entropy and therefore amounts to searching for components that are far from Gaussian.

Another tool to study independence is *kurtosis* [19], which is defined as

$$kurtosis(\mathbf{x}) = \mathrm{E}\{\mathbf{x}^4\} - 3(\mathrm{E}\{\mathbf{x}^2\})^2 \tag{3-24}$$

This concept accounts for the sparseness of a distribution. Sparse data are highly kurtotic. Maximizing kurtosis is equivalent to maximizing the joint entropy.

The basic principles of the information theory stated above, construct the main methodology of ICA. Any approximations or formulations keeping track of mutual information, other than those presented here, would suffice as well.

Practically, there are various algorithms to estimate the ICs. Mainly, these algorithms involve optimization procedures with some constraint function that gives a measure of independence. Generally, the number of ICs is determined by the number of observed vectors, by assuming a square mixing matrix for simplicity. The independent axes found in ICA are not necessarily orthogonal, therefore. They change relative distance between data points, and also alters the angles between data points, which affects similarity measures such as cosines. With the cosine similarity measure, the distance between two vectors $\mathbf{x_1}$ and $\mathbf{x_2}$ is given as

$$d = \frac{\mathbf{x}_1 \cdot \mathbf{x}_2}{\|\mathbf{x}_1\| . \|\mathbf{x}_2\|} \tag{3-25}$$

where the operator ||.|| stands for the Euclidean norm of a vector.

While searching for a smaller number of components than the original dimension, the dimension may be reduced beforehand using another method, such as PCA, or since the ICA is a linear generative model, the original variables may be replaced with a number of their principal components [25],[26]. Such techniques are also present for ordering the ICs according to their response to the constraint function and keep as many extrema as the desired number of dimensions.

Various implementations of ICA are reported [22], [24], [25], [26], [29], [31]. The method used in this study is called as FastICA [28] and makes use of *negentropy* as the constraint and Newton Iteration scheme as the optimization method. Negentropy is defined as a slightly modified version of differential entropy:

$$J(y) = H(y_{gauss}) - H(y) \tag{3-26}$$

where y_{gauss} is a Gaussian random variable of the same covariance matrix y. Due to the properties mentioned above, negentropy is always non-negative and equals

zero iff y has Gaussian distribution. Negentropy has the additional interesting property that it is invariant under invertible linear transformations [31], The estimation of negentropy is difficult, therefore in practice, some approximations have to be used. Note that, even in cases where an approximation is not very accurate, it can be used as a measure of non-gaussianity that is consistent in the sense that it is always non-negative, and equal to zero if y has Gaussian distribution.

The following equation can be shown to approximate negentropy [28]

$$J(y) \propto \left[\mathrm{E}\{G(y)\} - \mathrm{E}\{G(\gamma)\}\right]^2 \tag{3-27}$$

where γ is a Gaussian variable of zero mean and unit variance, and G is some nonquadratic function. In particular, choosing G, which does not increase fast, one obtains more robust estimators.

Assume the following relation: $g(\cdot) = \dot{G}(\cdot)$. FastICA uses $g_1(u) = \tanh(a_1 u)$ and $g_2(u) = ue^{-\frac{u^2}{2}}$, as nonlinear functions with $1 \le a \le 2$. This approximation is fast to compute, yet robust. Algorithm for calculating the p^{th} IC is as follows:

Step1. initialize w_p: $w_{old} = w_p$

Step 2. calculate the new vector: $w_{new} = \mathrm{E}\{x g({w_p}^T x)\} - \mathrm{E}\{x g'({w_p}^T x)\} w_p$

Step 3. normalize the calculated vector and update the IC: $w_p = \frac{w_{new}}{\|w_{new}\|}$

Substep1. $w_p = w_p - \sum_{j=1}^{p-1} w_p^T w_j w_j$

Substep 2. $w_p = \frac{w_p}{\|w_p\|}$

Step 4. if not converged, i.e. $\left| w_{old} \bullet w_p \right| \neq 1$, return to a2.

The steps numbered labeled with ‘Substep’ exist to ensure that different ICs are calculated by making use of decorrelation.

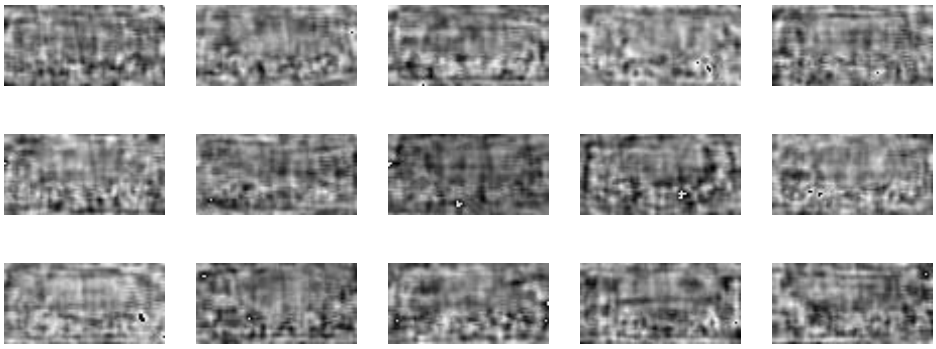

Figure 3-5: 15 IC images of the data set, calculated with the FastICA method

In the following figures, an illustration of dimension reduction techniques are seen. With MATLAB’s statistical toolbox, two sets of Gaussian variables are constructed. The projection axes are shown in the figures.

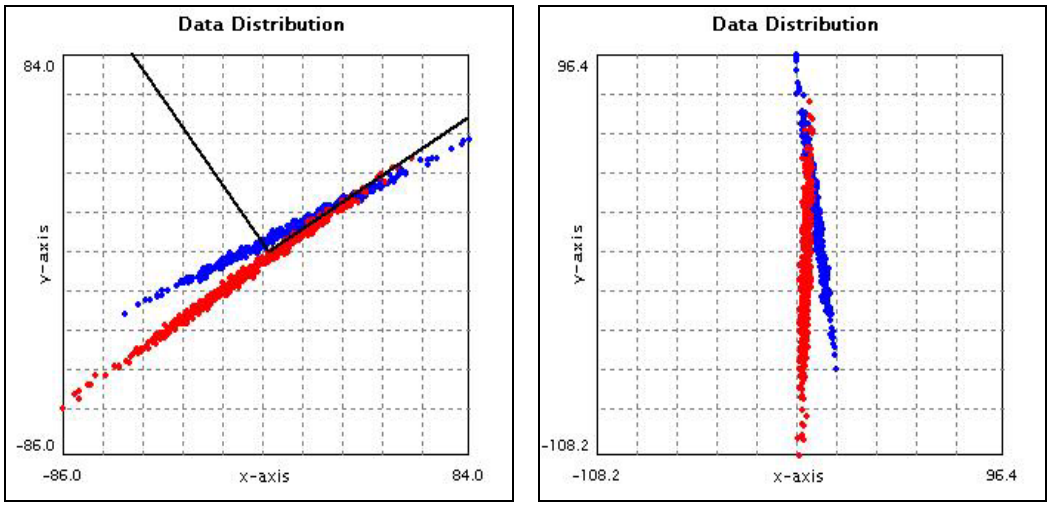

Figure 3-6: PCA results on the first constructed data set

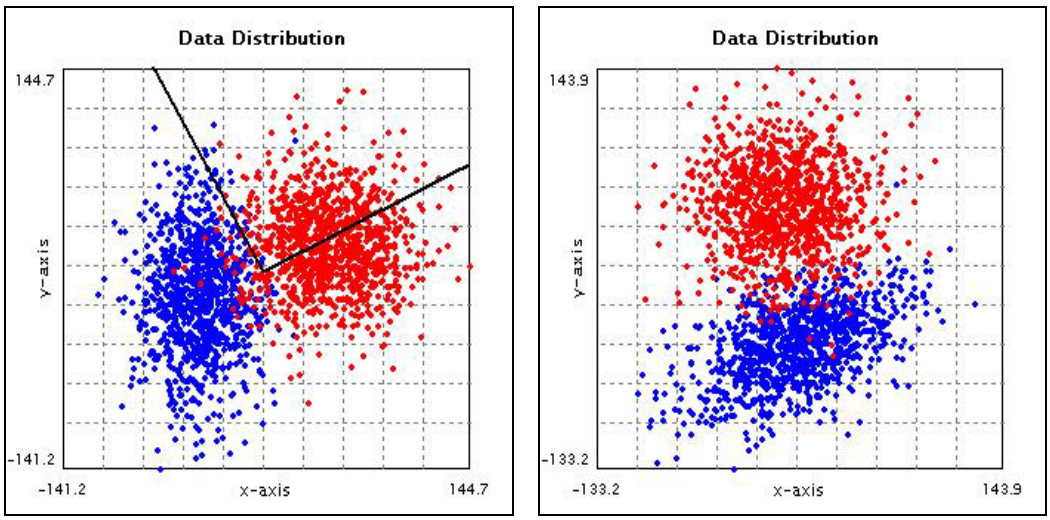

Figure 3-7: PCA results on the second constructed data set

Figure 3-6 and Figure 3-7. shows a two dimensional, two class distributions. Left figures show the class distributions and principal components of the corresponding distributions, while the right figures show the projected data onto the principal axes. As these illustrations indicate, PCA acts as if all data belongs to the same distribution and the PCA axes point to the directions of the extremum

variations of the data. It is observed that PC's are orthogonal, and the separability between classes is not altered after projection.

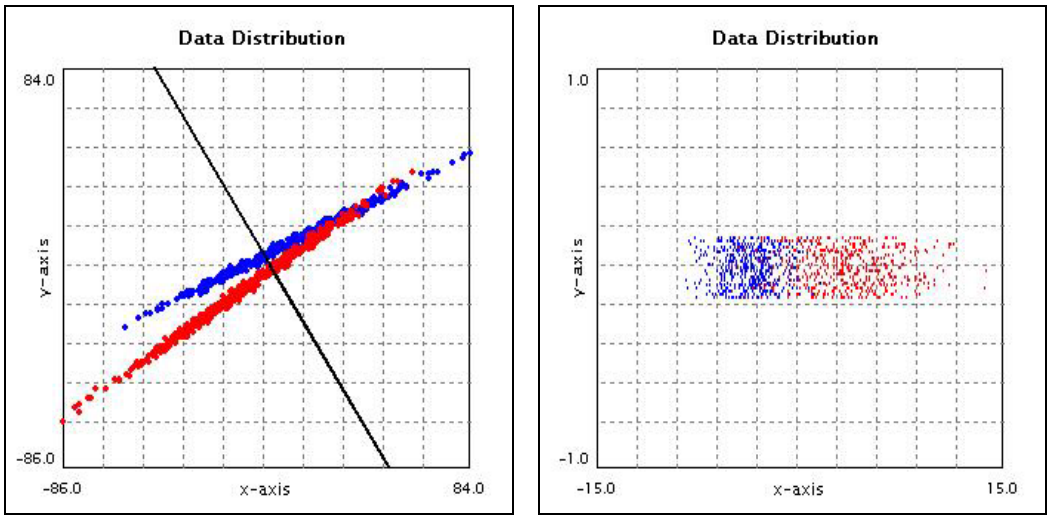

Figure 3-8:LDA results on the first constructed data set

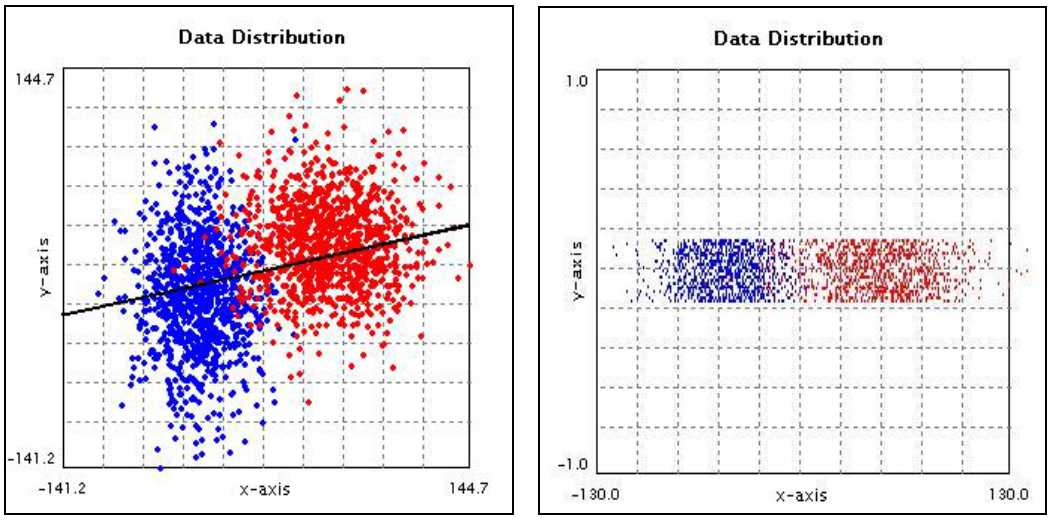

Figure 3-9: LDA results on the second constructed data set

Figure 3-8 and Figure 3-9 are organized like the preceding figures. Left images belong to class distributions and projection axes calculated via LDA. Right

images show the projected data onto the computed axes. As seen from the figures, LDA resulted in a projection axis that maximizes the class separability on single dimension. Dimension is reduced to 1 since we are dealing with a 2 class problem.

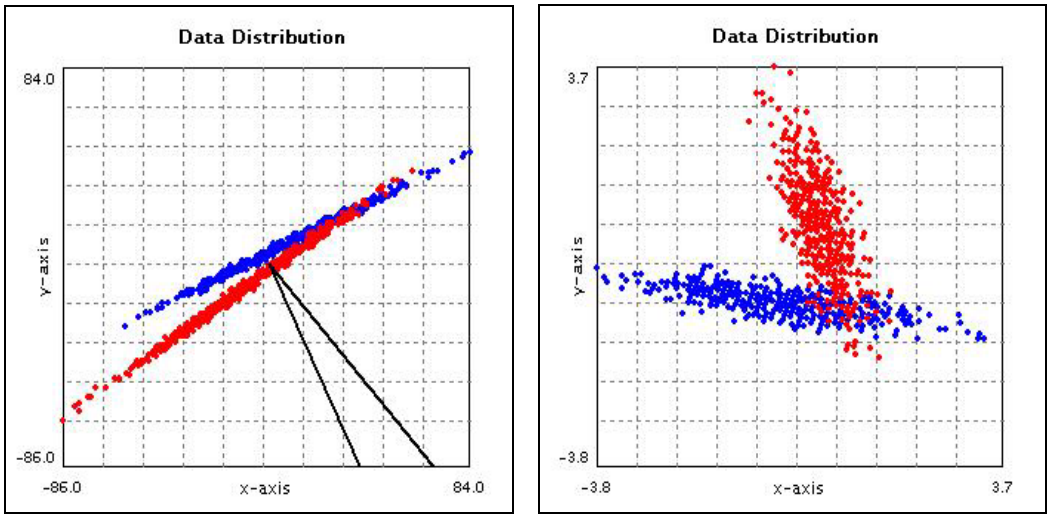

Figure 3-10:ICA results on the first constructed data set

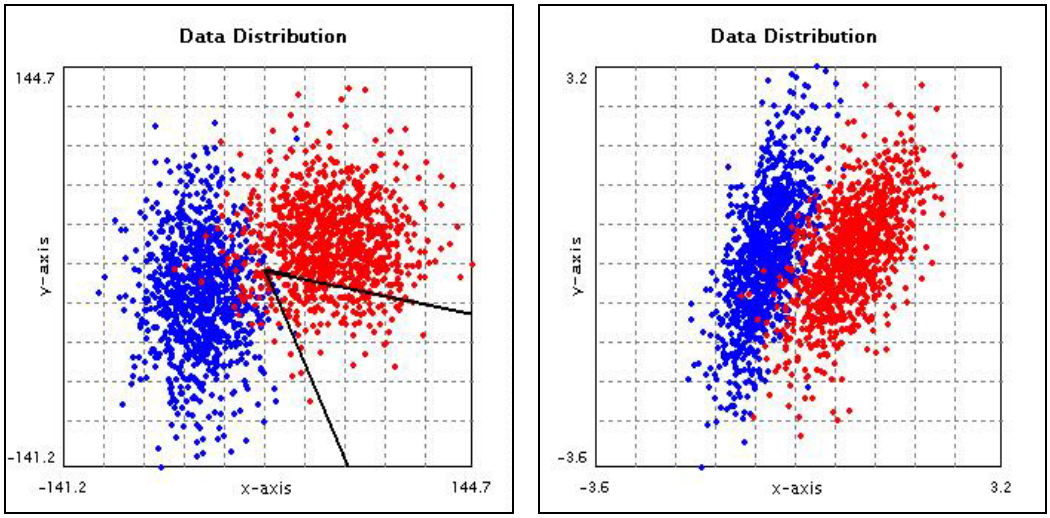

Figure 3-11:ICA results on the second constructed data set

Left images on Figure 3-10 and Figure 3-11 show the IC's of two different distributions whose PC's and LDA axes are indicated in the preceding axes. The computed IC's are clearly non-orthogonal, and the relative distances between points are not preserved.

CHAPTER 4

CLASSIFIERS

4.1 PATTERN CLASSIFICATION BASICS

The classical model of a pattern recognizer involves a sensor, a feature extractor and a classifier, where classification is defined to be the assignment of the features into one of the specified classes.

The classifiers are mainly designed using statistical, structural or neural network approaches [10]. While the structural methods expose a description of the pattern rather than classifying it, the neural networks are black box approaches supervised according to a reward/punishment scheme. Finally, the statistical methods partition the feature space into class decision regions such that the probability of error or the cost of error is minimized. Unless fuzzy sets are used, statistical methods find out disjoint regions.

In some statistical methods, the underlying probability density of the pattern is assumed to be belonging to some of the classical parametric densities. In such problems known as the parametric approaches, the unknown parameters of the densities are tried to be estimated to find out the optimal classifier. However, the classical densities rarely fit the actual pattern density, and nonparametric approaches are employed in many practical problems [6]. The goal in some nonparametric approaches is to estimate the density functions $p(x|w_j)$ from training patterns, and can be thought of a generalization of the histogram approach. The others aim to a posteriori probabilities $P(w_j|x)$ bypassing the estimation and directly go into the decision functions.

The basic idea behind the probability estimation can be stated in terms of the binomial distribution [6],[10]. Having n independently drawn samples of the probability density function p(x), binomial distribution gives the probability of k of the samples falling into the region R

$$P_k = \binom{n}{k} P^k (1-P)^{n-k} \tag{4-1}$$

with the expected value E{k} = nP. Since the binomial distribution peaks sharply about the mean, k/n is a good estimate for P. This probability is equals

$$P(\mathbf{x} \in R) = \int p(\mathbf{x}')d\mathbf{x}' \tag{4-2}$$

and assuming R is very small and encloses a volume V

$$P(\mathbf{x} \in R) = p(\mathbf{x})V \tag{4-3}$$

These results yield the following estimation of p(**x**)

$$p(\mathbf{x}) \approx \frac{k/n}{V} \tag{4-4}$$

Thus, a reasonable estimate for the a posteriori probability is

$$P_n(w_i \mid x) = \frac{p_n(x, w_i)}{\sum p_n(x, w_j)} = \frac{k_i}{k} \tag{4-5}$$

For the minimum error rate, we select the category most frequently represented within the cell.

4.1.1 NEAREST NEIGHBOR CLASSIFIER

When there exists a set of labeled training samples, assigning a test sample to the class associated with the training sample, which is closest to the test

data, is known as the nearest neighbor classification [10]. The ‘closeness’ of the data are calculated using a proper distance measure, and the most widely used distance measure is the Euclidean distance. This kind of classifier is a special case of the k-nearest neighbor classifier, which is discussed in the next section, with k=1.

4.1.2 K-NEAREST NEIGBOR CLASSIFIER

With the k-nearest neighbor classifier, the k closest sample data to the test data are considered, and the test data is labeled with the most frequently observed class among the k nearest neighbors.

4.1.3 NEAREST MEAN CLASSIFIER

The mean vector of each class is calculated using the known labeled data, and the nearest mean classifier labels the test data in the class whose mean vector happens to be the closest to the test data.

4.1.4 K-MEANS CLASSIFIER

K-means algorithm is an unsupervised method as the training data are unlabeled. ‘K’ stands for the number of classes to partition the feature space into. The k-means algorithm starts with a random initialization of the k mean vectors. The train data are clustered around these mean vectors according to the nearest neighbor rule, and each mean vector is restructured by calculating the mean of its cluster. Restructuring is repeated until the mean vectors are not updated anymore or the maximum number of iterations is reached.

4.2 LEARNING VECTOR QUANTIZATION

Another popular approach in supervised learning is the learning vector quantization (LVQ) where the decision boundaries are represented via a set of code vectors that are globally modified [12]. Each class is assigned to a set of feature vectors called the code vectors and every coordinate in feature space is

assigned to the class of the closest code vector. There are variations of the LVQ algorithm [12], [15], [33], [32] that ensures faster convergence of the decision algorithms or closer partitioning of the feature space to the optimal decision boundaries.

The original LVQ algorithm starts with a set of optimal number of initial code vectors and the code vectors are updated iteratively via a learning rate until the criterion of learning is achieved. Since the decision boundaries are represented by the code vectors, lying close to the class borders, a good approximation of the a posteriori probability is not necessary everywhere [32]. It is more important to place the code vectors such that the nearest neighbor classification minimizes the average expected misclassification probability.

In the following sections, main algorithms of LVQ are explained. Through these sections, x(t) represents a labeled training sample, $m_i(t)$ represents the sequential values of the code vectors. $\alpha(t)$ is the learning rate of the procedure. Let the nearest code vector to the input pattern is $m_c(t)$.

4.2.1 THE LVQ1

The following equations define the basic learning vector quantization algorithm, so-called LVQ1 [12].

$$\mathbf{m}_c(t+1) = \begin{cases} \mathbf{m}_c(t) + \alpha(t)\left[\mathbf{x}(t) - \mathbf{m}_c(t)\right] & \text{if } \mathbf{x} \text{ and } \mathbf{m}_c \text{ belong to the same class} \\ \mathbf{m}_c(t) - \alpha(t)\left[\mathbf{x}(t) - \mathbf{m}_c(t)\right] & \text{if } \mathbf{x} \text{ and } \mathbf{m}_c \text{ belong different classes} \end{cases} \quad (4\text{-}6)$$

and

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) \text{ for i} \neq \text{c} \quad (4\text{-}7)$$

Note that, this approach is supervised, since we know each train data's class label. In LVQ1, after the code vectors are initialized, each train data is observed for a chosen number of iterations. In an iteration, every element of the training set is examined, and for each train data, the code vector, which is the nearest to the train data is found. If the data is correctly classified, the winning

code vector is moved towards the train data by some scaled value of the distance between the data and the winning code vector. This scale is called as the *learning rate*. If the training data is misclassified, the code vector is moved away from the misclassified data by the distance between the winning code vector and the train data, scaled with the learning rate. Note that, the code vectors, other than the winning vector are not updated.

The learning rate is chosen between 0 and 1 and is usually made to decrease monotonically with time.

Figure 4-1 shows how LVQ1 method modifies the decision boundary. There are two classes of Gaussian variables, each class represented with a different color. The black points are code vectors. Each data falling onto the light blue area would be labeled as ‘blue’, while any data that falls into light red region would be labeled as ‘red’. Left image shows the decision areas after random initialization of code vectors, while the right image shows the updated decision regions after 5 iterations. Note that, there are 5 code vectors for each class, and learning rate is chosen to be 0.015. The algorithm converged after 5 steps.

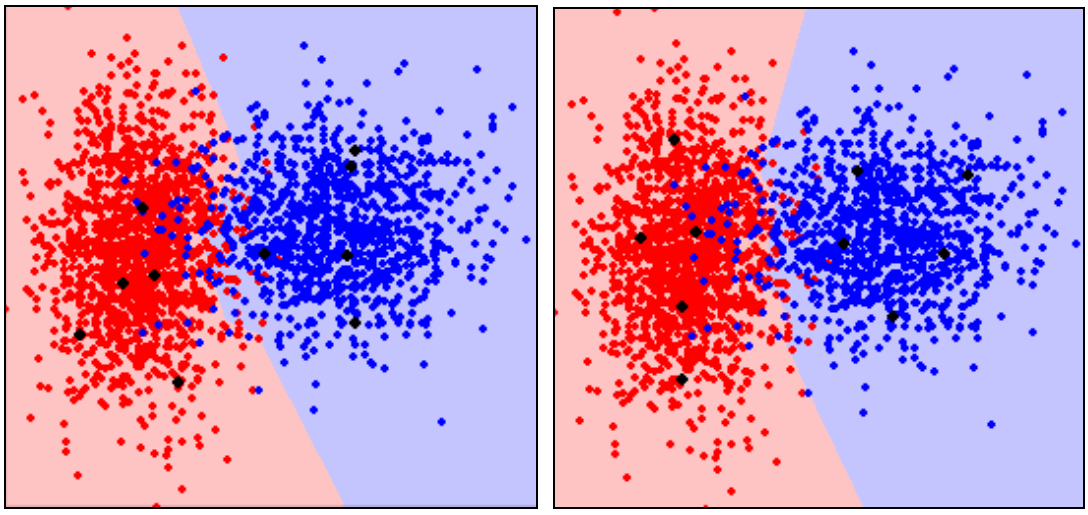

Figure 4-1: Class Boundaries and Code Vectors of a distribution in 2 dimensional feature space at initialization and after convergence

4.2.2 THE OPTIMIZED LEARNING RATE LVQ1 (OLVQ1)

OLVQ1 is a modification of the LVQ1 in the way that an individual learning rate $\alpha_i(t)$ is assigned to each code vector $m_i(t)$ and is updated in each iteration for the fastest convergence of the algorithm [1].

$$\mathbf{m}_c(t+1) = \begin{cases} \mathbf{m}_c(t) + \alpha_c(t)\left[\mathbf{x}(t) - \mathbf{m}_c(t)\right] \text{ if } \mathbf{x} \text{ and } \mathbf{m}_c \text{ belong to the same class} \\ \mathbf{m}_c(t) - \alpha_c(t)\left[\mathbf{x}(t) - \mathbf{m}_c(t)\right] \text{ if } \mathbf{x} \text{ and } \mathbf{m}_c \text{ belong different classes} \end{cases} \quad (4\text{-}8)$$

and

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) \text{ for i} \neq \text{c} \quad (4\text{-}9)$$

The preceding equations can be rephrased in the following form:

$$\mathbf{m}_c(t+1) = \{\left[1 - s(t)\alpha_c(t)\right]\mathbf{m}_c(t) + s(t)\alpha_c(t)\mathbf{x}(t) \quad (4\text{-}10)$$

where s(t) = +1 if the classification is correct, and s(t) = -1 otherwise.

For the statistical accuracy of the learned code vectors, it is obvious that the corrections, which are made at different times, should be of approximately equal magnitude. In other words, the same training set, when ordered differently, should yield the same solution under the same initialization. As the previous equation states, the updated code vector contains the traces of the last input pattern through the last term and the traces of the previous input patterns through the first term. The contribution factor of x(t) in this learning step is $\alpha_c(t)$, whereas this factor is [1-s(t) $\alpha_c(t)$] $\alpha_c(t-1)$ for x(t-1). For x(t) and x(t-1) to contribute equally to the code vector, their contribution factors must be equal. Approximately equal contributions yield the recursive formulation:

$$\alpha_c(t) = \frac{\alpha_c(t-1)}{1 + s(t)\alpha_c(t-1)} \quad (4\text{-}11)$$

However, since the learning rate may rise, its value should be restricted to unity.

4.2.3 THE LVQ2

In LVQ2, two code vectors, $m_i(t)$ and $m_j(t)$, that are nearest neighbors to x(t) are updates instead of one. One of them must be the nearest neighbor in the same class with the test pattern while the other must belong to another class. Moreover, x(t) must fall into a window between m_i and m_j. The window is defined to have the width w

$\min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > s$ where $s = \frac{1-w}{1+w}$ and d_i represents the Euclidean distance between x and m_i.

With the figures below, the effects of LVQ algorithm parameters; namely the learning rate, the size of the training data set, the number of codebook vectors; on the recognition rate, are graphically presented. Recognition rate is the percentage of correct classifications among the test samples.

A set of simulated data is used. The data set consists of 3 two-dimensional classes with Gaussian distribution of different means and variances. Figure 4-2 shows this data set with each class indicated with a different color. The mean and standard deviation pairs $((\mu_1, \mu_2), (\sigma_1, \sigma_2))$ for these classes are ((55,72.2),(19.5,20.8)), ((35,167),(29.5,32)) and ((-14.5,31.5),(24.5,32)). Each class has 500 samples.

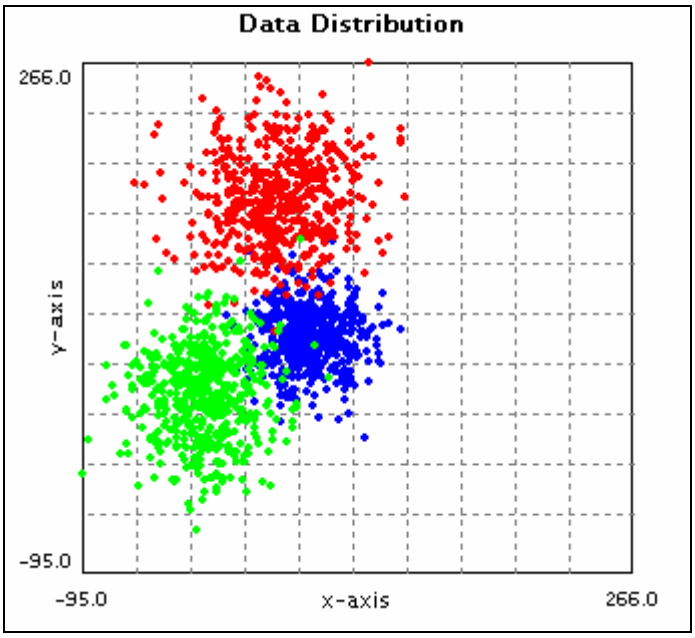

Figure 4-2: Simulated data set, which is used in comparisons

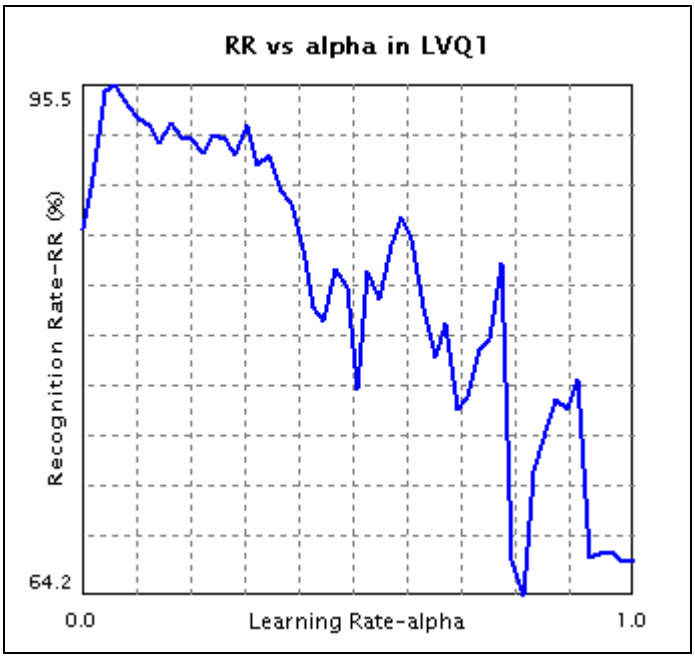

Figure 4-3: Evaluation of the effect of learning rate on performance

The LVQ1 algorithm is run with different learning rates, and as it is observed in Figure 4-3, the simulations show that the recognition rate versus learning rate is a unimodal function with a global optimum. This optimum may depend on the number of iterations, since for larger values of learning rate, oscillations with iterations might be observed. During simulations, 5 code vectors per class are used, and the iteration number is 10.

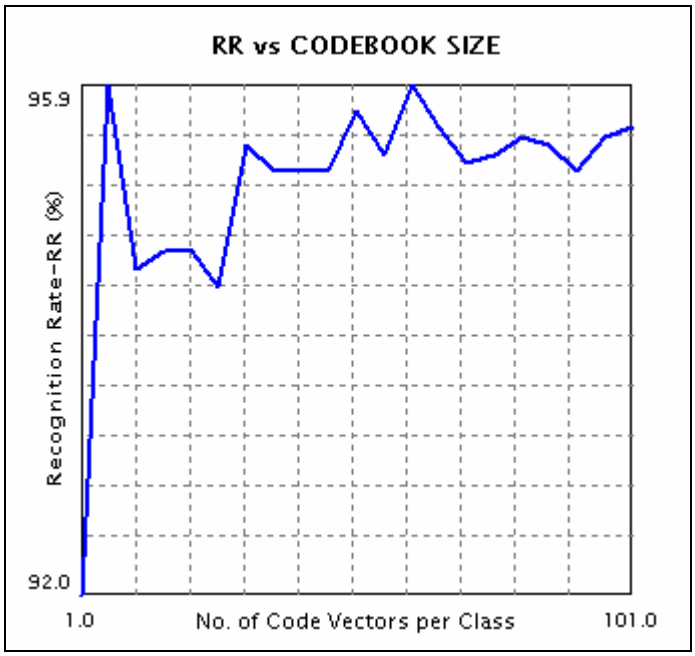

Figure 4-4: Effect of the size of the Code Book on performance

As seen in Figure 4-4, increasing the number of code vectors does not always increase the recognition rate. After 5 code vectors per class, the recognition rate reside within the range of 95.3%-95.9%. While choosing the number of code vectors, codebook size should be chosen in accordance with the number of observations. A large number of code vectors may result in poor reasoning, if there are a small number of observations per vector. During this simulation, learning rate is chosen to be 0.04, and the number of iterations is 10.

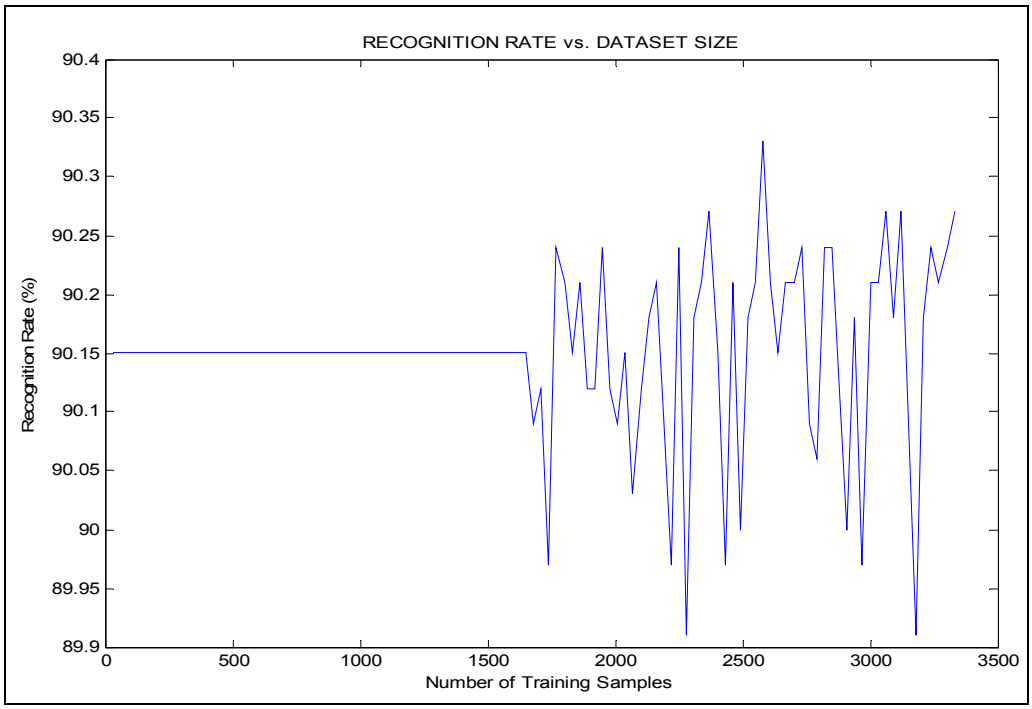

Figure 4-5: Effect of the size of training data set on performance

As explained with the previous simulation, data set size and the number of code vectors need to be considered together. With well-behaved distributions, with an appropriate number of code vectors, performance is not altered considerably with increasing observations. In Figure 4-5, the effect of the training set size on recognition rate is presented. The learning rate is chosen to be 0.2 in this simulation.

CHAPTER 5

COMPARATIVE ANALYSIS VIA SIMULATIONS

In this chapter, the experimental results of previously explained algorithms are presented. The algorithms are implemented in C++, compiled using MS Visual C++ 6.0 IDE. The application is realized using MFC 6.0 Library for user interfaces and DirectX 8.0, GDIPlus Library, and CDirectShowWrapper and CGdiplusWrapper classes of TUBITAK-BILTEN for video and image capturing and viewing.

The algorithm can be examined in two phases. The first phase is that of detection, and the second is the recognition phase. The detection phase input is infrared movies in MPEG1 format and recognition is carried out on captures of the detected images of targets.

The data are collected by a stationary camera, observing vehicles from a fixed distance and fixed viewing angle. There two sets of movies, which are recorded at different times of the year, June 2003 and February 2004. The database is divided into two disjoint sets for test and training data, according to seasonal changes. There are 5 classes in the resulting database: Trucks, buses, vans, minibuses, tankers. Table 5-1 summarizes the classes and class populations of the collected data.

Table 5-1: Statistical properties of the test and the training sets

	Trucks	Buses	Vans	Minibuses	Tankers	Total
Training Set	33	26	22	44	13	**138**
Test Set	7	4	39	14	1	**65**

5.1 PREPROCESSING AND TARGET DETECTION

In detecting the targets in infrared imagery, user intervention is required for background processing. User selects a region of interest with a stationary background scene.

After the user inputs a stationary background region, tracking loop begins. This loop lasts until the user exits or stops the application. The first step of this loop is to directly compare the corresponding pixels of the stationary region of interest and the frame to be processed, i.e. subtraction of two consecutive regions in two frames.

The second step is binarization, i.e. thresholding of each pixel of the difference image.

$$D_{jk} = \begin{cases} 1 \quad \text{if} \quad |\mathrm{F(x,y,j)} - \mathrm{F(x,y,k)}| > \mathrm{T} \\ 0 \quad \text{,otherwise} \end{cases} \tag{5-1}$$

The results of thresholding are quite noisy, especially because the background is highly cluttered in the data.

In order to remove the regions that are unlikely to be object parts, morphological operators are used. Dilation and erosion are used one after another to remove the noise and recover back all the object parts. As the morphological operator, a 3x3 structure in Figure 5-1, which has a checkerboard pattern, is utilized.

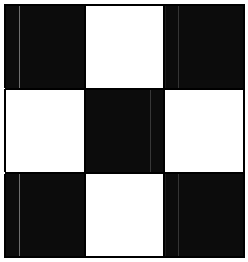

Figure 5-1: The pattern used during morphological operations

Size filter is also used after morphological operators to decrease the false alarm rates. The frames that result in a foreground area that is smaller than 100 pixels are eliminated. A bounding rectangle for the detected target is calculated by scanning the binary image to find the maximum and minimum object pixels in both directions.

In Figure 5-2, samples of typical background and target images, which are used during this study, are presented. Figure 5-3 shows the difference image between the figures presented in Figure 5-2, and the result of thresholding step on the difference image. Finally, Figure 5-4 shows the morphological operation result.

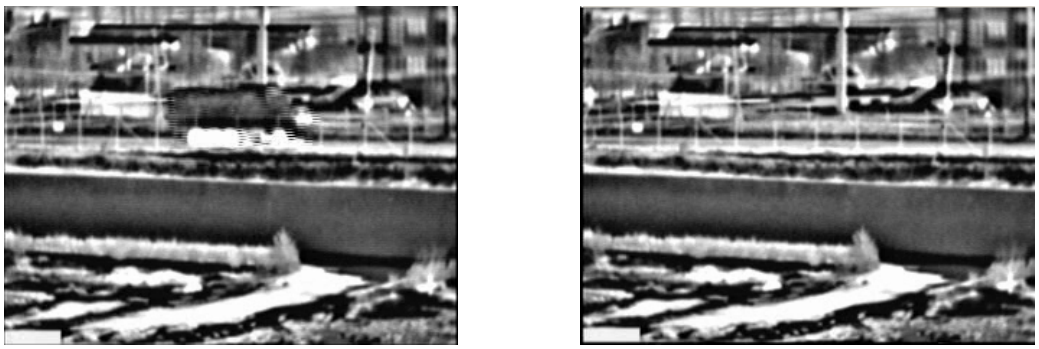

Figure 5-2: Samples of background and target scenes

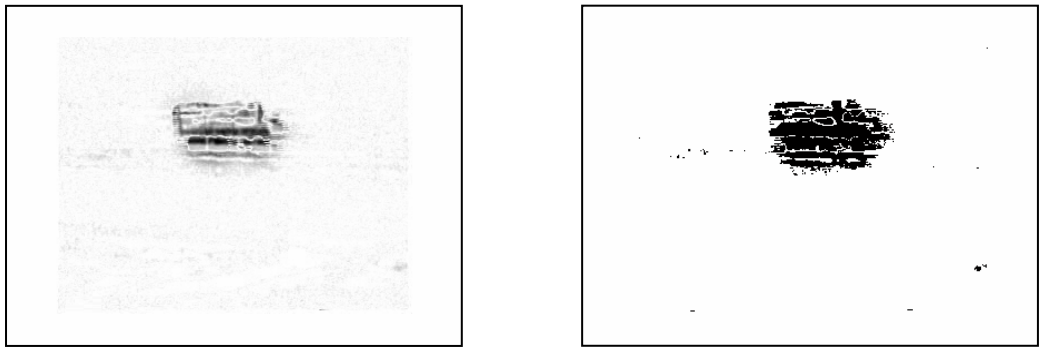

Figure 5-3: Samples of difference image and its thresholding result

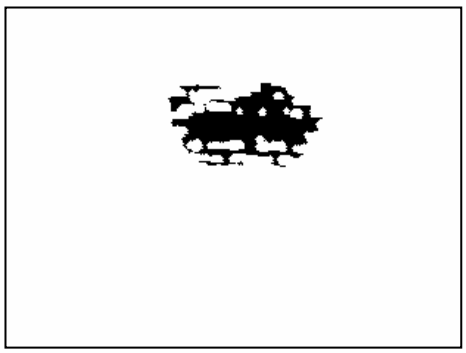

Figure 5-4: Sample of morphological operation

The target image, whose bounding rectangle is calculated, is extracted from the overall image, and normalized to the size of 40x80 pixels by bicubic interpolation. Then the recognition algorithms are employed. The comparative results of the recognition phase are presented in the next section.

5.2 SIMULATION RESULTS

There are two kinds of illustrative results, which are presented. The first presentation is via confusion matrix, and the second type is the graphical plots. In the confusion matrix, the first column represent the actual classes of targets, and the first row represent the labels assigned to the data. Diagonal entities are the

correct classifications, while the non-diagonal entities are the misclassifications. In the simulations, Euclidean distance is utilized unless otherwise stated.

5.2.1 PCA vs subspace LDA

In Table 5-2 and Table 5-4, an expected improvement in subspace LDA is observed. In Table 5-3, results, when dimensions are reduced further with PCA, are presented. The recognition rate decreases as it is expected, but is still satisfactory when the dramatic decrease in the dimension is considered. LDA's discriminatory approach results in higher recognition rate.

Table 5-2: Results of PCA algorithm

DR Technique			PCA, trained with 138 images		
Dimension Reduced To			100		
Classifier			Nearest Neighbor		
Classifier Training			138 representatives		
	TRUCK	**BUS**	**VAN**	**MINIBUS**	**TANKER**
TRUCK	5	1	1	0	0
BUS	0	2	2	0	0
VAN	5	6	23	5	0
MINIBUS	0	0	6	8	0
TANKER	0	0	0	1	0
Recognition Rate : 58.46%					

Table 5-3: Results of PCA algorithm with fewer dimensions

DR Technique			PCA, trained with 138 images		
Dimension Reduced To			20		
Classifier			Nearest Neighbor		
Classifier Training			138 representatives		
	TRUCK	**BUS**	**VAN**	**MINIBUS**	**TANKER**
TRUCK	4	1	2	0	0
BUS	0	4	0	0	0
VAN	6	8	22	3	0
MINIBUS	0	1	9	4	0
TANKER	0	0	0	1	0
Recognition Rate : 52.308%					

Table 5-4: Results of subspace LDA algorithm

DR Technique			Subspace LDA, trained with 138 images		
Dimension Reduced To			20		
Classifier			Nearest Neighbor		
Classifier Training			138 representatives		
	TRUCK	**BUS**	**VAN**	**MINIBUS**	**TANKER**
TRUCK	5	1	1	0	0
BUS	0	3	1	0	0
VAN	6	7	24	2	0
MINIBUS	0	0	6	8	0
TANKER	0	0	0	1	0
Recognition Rate : 61.5%					

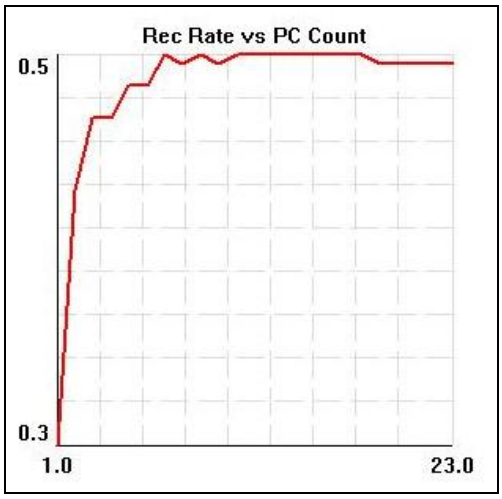

Figure 5-5: Recognition Rate vs the number of Principal Axes used. PCA is trained with 25 images

As Figure 5-5 illustrates, increasing the number of principal components, we obtain a stationary recognition rate. This is because, the principal components corresponding to small valued eigenvalues does not carry much energy, therefore does not carry much information relevant to the classification.

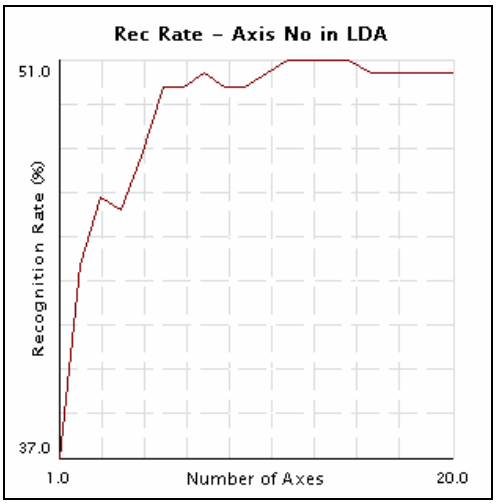

Figure 5-6: Recognition Rate vs Number of Axes used in LDA

As expected, LDA is a better feature extractor than PCA, since it takes the separability issues into account. With subspace LDA, as illustrated in

Figure 5-6 the dimension can be reduced more without downgrading the performance.

5.2.2 PCA vs ICA

Table 5-5 and Table 5-6 compare the results of classification via Euclidean and cosine similarity measures. It is observed that if cosine similarity measure is used, the recognition rate increases as proposed in [25]. Among the dimension reduction methods, ICA and LDA are observed to be more successful in extracting the information that is relevant to the classification.

Table 5-5: Results of ICA algorithm

DR Technique			ICA, trained with 138 images		
Dimension Reduced To			20		
Classifier			Nearest Neighbor		
Classifier Training			138 representatives		
	TRUCK	**BUS**	**VAN**	**MINIBUS**	**TANKER**
TRUCK	1	1	2	3	0
BUS	0	3	1	0	0
VAN	7	0	32	0	0
MINIBUS	0	0	10	4	0
TANKER	1	0	0	0	0
Recognition Rate : 61.53%					

Table 5-6: Results of ICA algorithm with 'cosine-matching'

DR Technique			ICA, trained with 138 images		
Dimension Reduced To			20		
Classifier			Nearest Neighbor with cosine similarity measure		
Classifier Training			138 representatives		
	TRUCK	**BUS**	**VAN**	**MINIBUS**	**TANKER**
TRUCK	2	0	3	2	0
BUS	0	3	1	0	0
VAN	5	2	32	0	0
MINIBUS	0	0	10	4	0
TANKER	1	0	0	0	0
Recognition Rate : 63.0%					

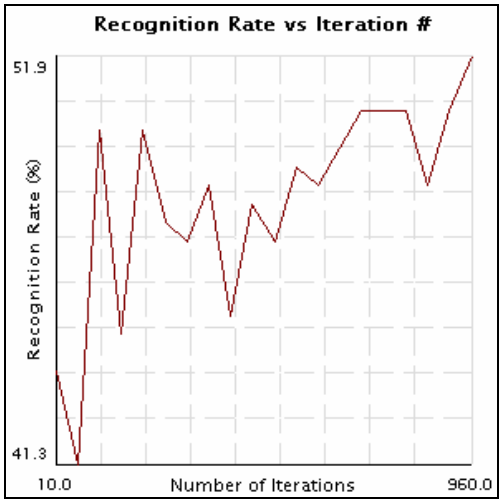

Figure 5-7: ICA-Recognition Rate vs Iteration Count

With the iterative procedure of finding independent components, iteration count improves the performance.

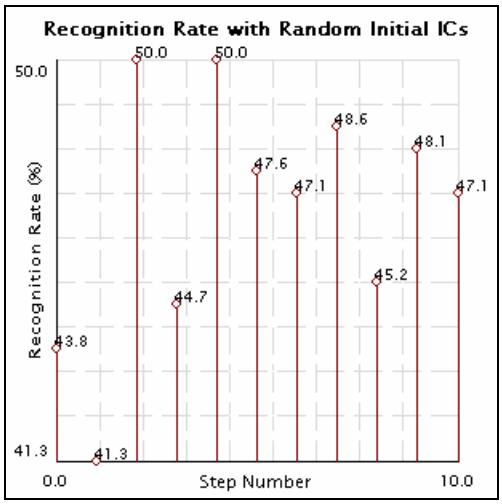

Figure 5-8: ICA with different initial ICs

The initialization of IC's is a critical issue with a low number of iterations.

5.2.3 Simulations by LVQ1 vs Nearest Neighbor

Comparing the results, which are presented in Table 5-2, Table 5-7 and Table 5-5, Table 5-8, the superiority of LVQ1 over nearest neighbor method is apparent. With fewer vectors (a total of 25 vectors are used in LVQ1, while 138 vectors are utilized in NN), higher correct classification rate, as well as better classification time, is achieved. The projections, both onto PC's and IC's are well classified with LVQ1.

Table 5-7: Classification with PCA+LVQ1

DR Technique			PCA, trained with 138 images		
Dimension Reduced To			100		
Classifier			LVQ1		
Classifier Training			Training set population: 138 Codebook : 5 vectors/class, α: 0.2		
	TRUCK	**BUS**	**VAN**	**MINIBUS**	**TANKER**
TRUCK	5	0	0	2	0
BUS	0	1	3	0	0
VAN	2	2	33	2	0
MINIBUS	0	1	8	5	0
TANKER	1	0	0	0	0
Recognition Rate : 67.7%					

Table 5-8: Classification with ICA+LVQ1

DR Technique			ICA, trained with 138 images		
Dimension Reduced To			15		
Classifier			LVQ1		
Classifier Training			Training set population: 138 Codebook : 5 vectors/class, α: 0.2		
	TRUCK	**BUS**	**VAN**	**MINIBUS**	**TANKER**
TRUCK	1	0	3	0	0
BUS	0	3	1	0	0
VAN	0	1	38	0	0
MINIBUS	0	1	9	3	1
TANKER	0	0	0	1	0
Recognition Rate : 69.3%					

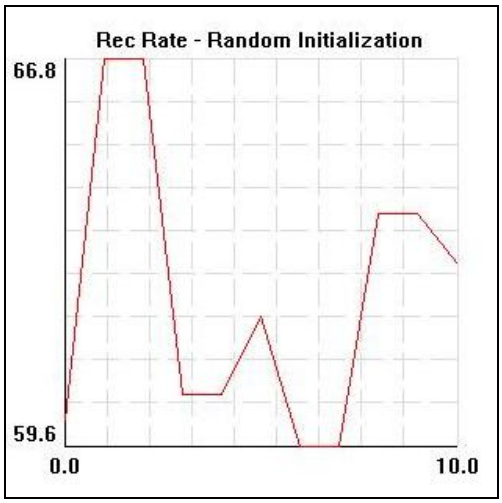

Figure 5-9: LVQ1 is run with different initial code vectors

In Figure 5-9, the x-axis shows 10 different cases, in which various random initial vectors are used. The target images have high order statistical properties and are sensitive to initial vectors. This is due to the local settlement of the code vectors [32].

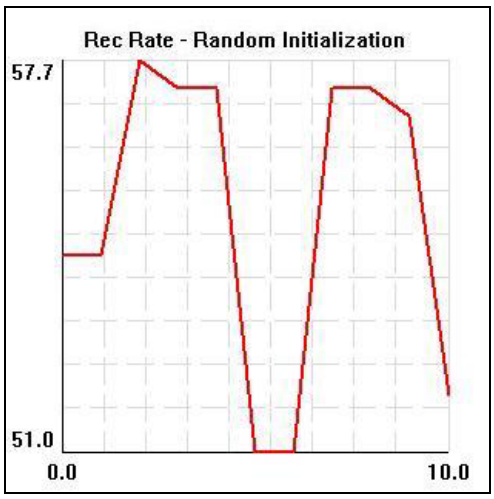

Figure 5-10: LVQ1 is run with different initial code vectors, and each image's histogram is stretched

In Figure 5-10, the effect of a preprocessing step, histogram stretching, on LVQ1 is presented. The same reasoning with Figure 5-9 applies in this illustration. The difference is due to the histogram stretching, which worsens the performance. Histogram stretching, as experimentally observed, is not an appropriate preprocessing technique in ATR with IR, since the IR signatures of each class of vehicles are usually corrupted via such a method.

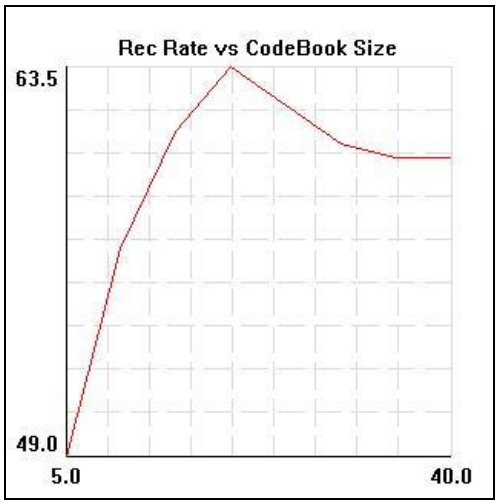

Figure 5-11: LVQ1 Recognition rate vs Codebook size

As observed in Figure 5-11 and stated in Chapter 4, increasing codebook size has an effect of poor reasoning, therefore the optimal number of codebook size should be decided according to the number of observations. Increasing the size of the codebook so that there exist one training data per vector is an example of such a situation. In such a case, performance, the expected classification performance is no more than that of the nearest neighbor classification.

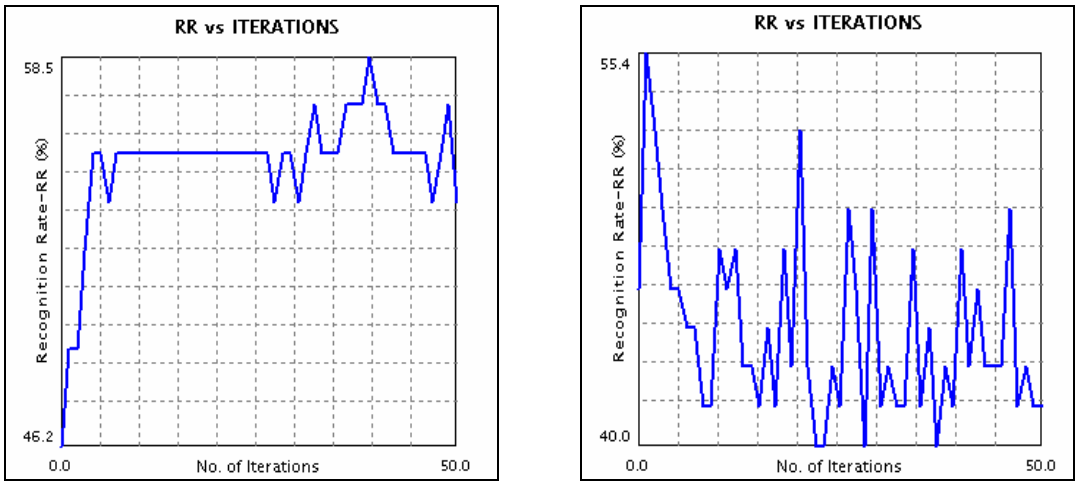

Figure 5-12: LVQ1 trained in different numbers of iterations

As in Figure 5-12, the number of iterations in LVQ1 training is another important parameter to examine by simulations. Left plot presents the results obtained by LVQ1, when dimension is reduced to 10, 3 code vectors per class are computed, alpha is chosen to be 0.01 In the right plot, all parameters except for alpha are the same. Alpha is chosen to be 0.15. The oscillations around 47% are the effect of the high learning rate. Left plot shows that the convergence is reached, and it is reached with a considerably low number of iterations.

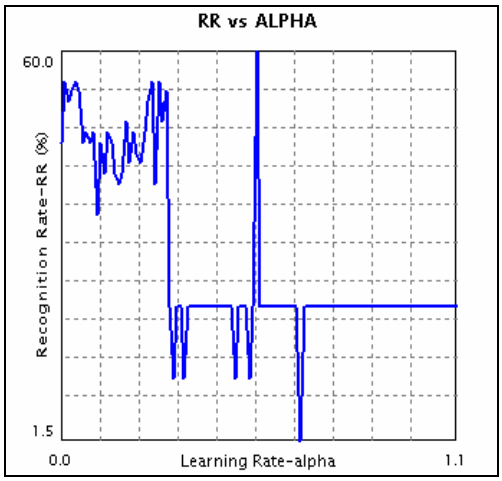

Figure 5-13: LVQ1 Recognition rate vs Learning rate

In LVQ1, learning rate parameter, alpha, may be selected experimentally. As expected, increasing the learning rate dramatically increases the recognition rate. In LVQ, the aim is to represent the decision boundaries as precisely as possible. Therefore, when the code vectors are moved with large steps, the code vectors move far from the decision boundaries, leading the smaller recognition rate as presented in Figure 5-13.

5.2.4 OLVQ1 vs LVQ1

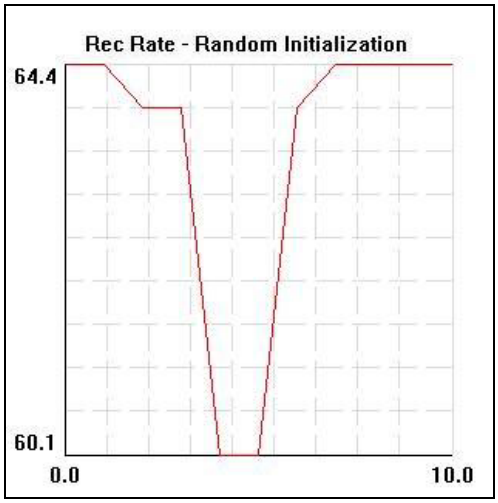

Figure 5-14: OLVQ1 run with different initial code vectors

In terms of recognition performance, OLVQ1 behaves in a similar manner to that of LVQ1, as shown in Figure 5-14. In fact, the three options of LVQ yield almost similar accuracies in most pattern recognition problems, although a different philosophy underlies each [12].

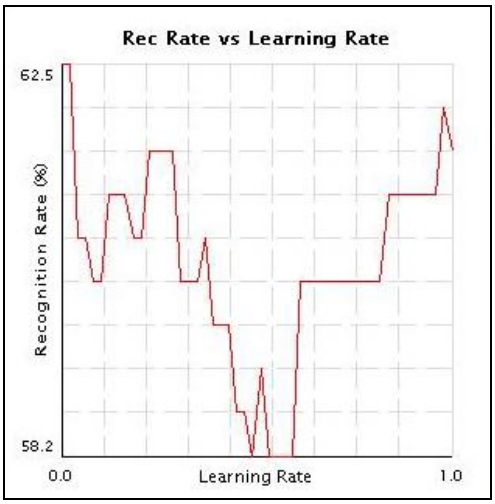

Figure 5-15: OLVQ1 Recognition Rate vs Learning Rate

The effect of initial learning rate in OLVQ1 is not like that of LVQ1, since the learning rate is continuously updated in this modification. The algorithm itself avoids the learning rate to go beyond 1.

5.2.5 Simulations by LVQ2

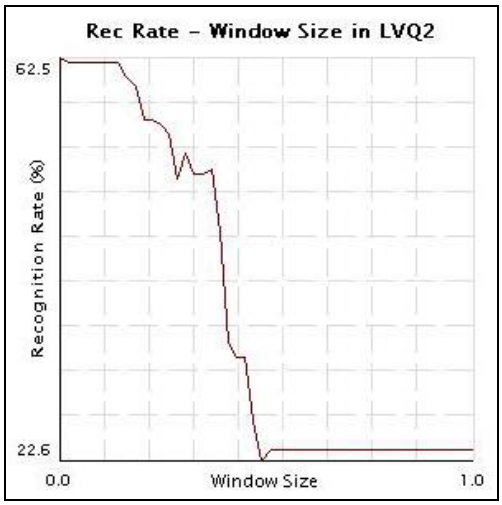

Figure 5-16: LVQ2-Recognition Rate vs Window Size

The window is a region where code vectors are updated. With the window size, the decision critic area is defined. As it is observed in Figure 5-16, window size is a quite critical parameter. Smaller values should be preferred.

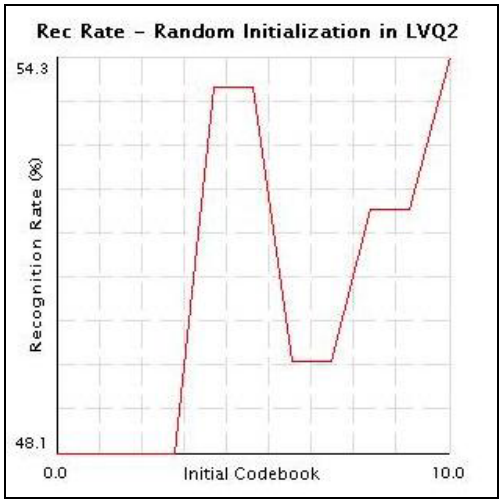

Figure 5-17: LVQ2 is run with different initial code vectors

LVQ2 turns out to be the most sensitive method to the initialization according to the simulations.

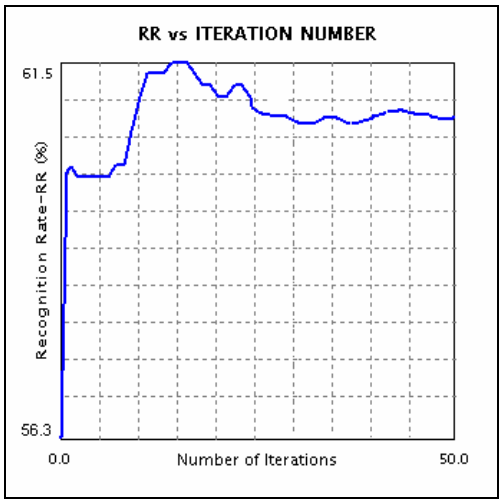

Figure 5-18: LVQ2 Recognition Rate vs Number of Iterations

The LVQ2 is a successful method of defining decision regions.

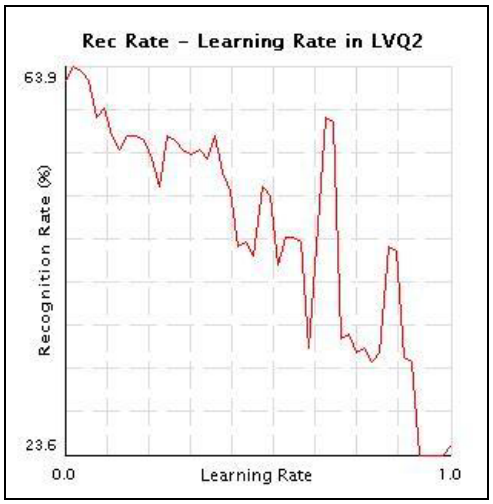

Figure 5-19: LVQ2-Recognition Rate vs Learning Rate

The small learning rate performs better with LVQ2 and a sufficient number of iterations.

5.2.6 Computational Efficiency

Since the ATR is a time-critical task, the computation times of the algorithms are presented in Table 5-9 for comparison. The results are presented in milliseconds and are obtained on a Pentium 4, 2.0 GHz machine.

Table 5-9: Computation times of implemented algorithms

	Specifications	**Computation Time (ms)**
PCA Training	Training set: 138 images of size 40x80	20462
Subspace LDA Training	Training set: 5 classes, 138 images of size 40x80. Dimension reduced to: 20.	21116
ICA Training	Training set: 138 images of size 40x80. Dimension reduced to: 20 , No. of iter.: 1	82395
Projection Algorithm	An image of size 40x80 image onto 20-D space	12
LVQ1 Training	Training set: 20-D, 5 class, 138 vectors Code Vectors/class: 5 , No. of iter: 1	79
OLVQ1 Training	Training set: 20-D, 5 class, 138 vectors Code Vectors per class: 5 , No. of iter.: 1	77
LVQ2 Training	Training set: 20-D, 5 class, 138 vectors Code Vectors per class: 5 , No. of iter.: 1	77
NN Classification	Data Dimension: 20 Number of Representatives: 25 Number of Test Vectors: 138	75

CHAPTER 6

CONCLUSIONS

Throughout this thesis, applications of some pattern recognition techniques on the task of ATR are studied. The investigated algorithms are grouped into two major parts as statistical dimension reduction methods where PCA, subspace LDA and ICA are studied, and pattern classification methods, namely nearest neighbor, k means, k nearest neighbors and LVQ are studied.

One of the major drawbacks of this study is the limited data set with which the algorithms are trained and tested. Though the test and data sets are limited, they are disjoint and are gathered under two different weather conditions, which the infrared sensors are quite sensitive about.

The performance is evaluated based on the recognition rate within this study. Other than recognition rate, computation time is an important issue in ATR tasks, with much of the interest on the test epoch. The decision algorithms are quite satisfactory.

According to the simulation results obtained, of the dimension reduction methods, which can be thought of as feature extractors of the intensity data, the best performing algorithm is ICA, while subspace LDA resulted in the second best performance and the PCA turned out to be the least promising method. LDA, taking the class information of the training set into account, performs better than PCA, as expected. The superiority of the ICA algorithm may be interpreted as a symptom of the non-gaussianity of the image data and that the intensity values are correlated in high orders, the assumption of second order correlatedness is not satisfactory. It also should be noted that, when the tests are carried out in joint sets of test and training samples, PCA and ICA perform closer to each other. However,

with disjoint sets, as illustrated in Chapter 5, ICA is superior to PCA. Better performance of ICA in more challenging test sets may be expected.

Among the classifiers, LVQ has the superior results. This result is not surprising, either. Since LVQ operates on decision boundary regions and iteratively updates the code vectors for better classification, it turns out to be a better classifier than nearest neighbor.

While evaluating the results, it should be kept in mind that all these results are obtained with sets of still IR imagery, and no use of temporal context is made. A remedy for more satisfactory results is sensor fusion algorithms. For instance, SAR outputs, whose employment in the ATR algorithms are reported to be more satisfactory, may be fused with IR images and increase the overall performance of the system. In addition, the viewing angle in collected data are identical, therefore, one needs not consider such a variation. For data sets with different views of targets, aspect windows may offer a good and easy solution. Certainly, to develop solutions for such data sets, there should be sufficient number of samples for every class' each aspect.

The implemented system is not tested with the images where the target parts are occluded. All the implemented algorithms assume that the target images are centered and background is extracted as much as possible. Almost certainly, the algorithms would produce higher false alarm rates under such test sets. This problem may be defeated by training the algorithms with sub-regioning the train images first. Intuitively, optimal sub-regioning is up to experimental results and operating conditions.

REFERENCES

[1] Michael W. Roth, Survey of Neural Network Technology fot Automatic Target Recognition, IEEE Transactions on Neural Networks, Vol. 1, No. 1, March 1990

[2] Yann LeCun, Yoshua Bengio, Convolutional Networks for Images, Speech and Time Series, The Handbook of Brain Theory and Neural Networks, MIT Press, 1995

[3] Charles W. Therrien, Disrete Random Signals and Statistical Signal Processing, Prentice Hall, 1992

[4] M. Turk, A. Pentland, Eigenfaces for Recognition, Journal of Cognitive Neuroscience, Vol.3 No.1, 1991

[5] W. Zhao, R. Chellappa, R.J. Phillips, Subspace Linear Discriminant Analysis for Face Recognition, IEEE Trans. On Image Processing, 1999

[6] R. O. Duda, P. E. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, 1973

[7] Bir Bhanu, Automatic Target Recognition: State of the Art Survey, IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-22, No. 4, July 1986

[8] W. M. Brown, C. W. Swonger, A Prospectus for Automatic Target Recognition, IEEE Transactions on Aerospace and Electronic Systems, Vol. 25, No. 3, May 1989

[9] B. Li, R. Chellappa, Q. Zheng, Experimental Evaluation of FLIR ATR Approaches – A Comparative Study, Computer Vision and Image Understanding 84, 5-24, 2001

[10] Robert J. Schalkoff, Pattern Recognition: Statistical, Structural and Neural Approaches, John Wiley & Sons, 1992

[11] Allen Gersho, Robert M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, 1992

[12] T. Kohonen, Self Organizing Maps, Springer Series in Information Sciences, Vol.30, 1995

[13] Lipchen A. Chan, Nasser M. Nasrabadi, Vincent Mirelli, Multi-Stage Target Recognition Using Modular Vector Quantizers and Multilayer Perceptrons, Proc. Computer Vision Pattern Recognition, pp. 114-119, 1996

[14] K. Hornik, M. Stinchcombe, Multi-Layer Feedforward Networks are Universal Approximators, Neural Networks 2, 1999

[15] Lin-Cheng Wang, Nasser M. Nasrabadi, Composite Classifiers for Automatic Target Recognition, Optical Engineering, Vol.37 No.3, March 1998

[16] L. C. Wang, S. Z. Der, N. M. Nasrabadi, A Committee of Networks Classifier with Multi-Resolution Feature Extraction for Automatic Target Recognition, Proc. IEEE Int. Conf. Neural Networks, Vol.3, 1997

[17] D. H. Wolpert, Stacked Generalization, Neural Networks 5, 1992

[18] Ben Krose, Introduction to Dimension Reduction and Feature Extraction, ASCI Advanced Issues in Neurocomputing, May 2003

[19] Imola K. Fodor, A Survey of Dimension Reduction Techniques, LLNL technical report, June 2002

[20] W. Zhao, R. Chellappa, R.J. Phillips, Subspace Linear Discriminant Analysis for Face Recognition, Proc. 4th Conference on Automatic Face and Gesture Recognition, 2000

[21] Haim J. Wolfson, Model Based Object Recognition by Geometric Hashing, Proc. European Conf. Computer Vision, 1990

[22] Aapo Hyvarinen, Beyond Independent Components, Artificial Neural Networks, Conference Publication No. 470, 7-10 September 1999

[23] Tom Carter, An Introduction to Information Theory and Entropy, Lecture Notes, http://cogs.csustan.edu/~tom/SFI-CCSS, June 2002

[24] Annie X. Guan, Harold H. Szu, A Local Face Statistics Recognition Methodology Beyond ICA and/or PCA, International Joint Conference on Neural Network, pp.1016 –1021, 1999

[25] M. S. Bartlett, J. R. Movellan, T. J. Sejnowski, Face Recognition by Independent Component Analysis, IEEE Transactions on Neural Networks, Vol.13, No.6, November 2002

[26] Imola Fodor, C. Kamath, Using Independent Component Analysis to Separate Signals in the Climate Data, Proceedings of the SPIE, Volume 5102, pp. 25-36 2003

[27] James Stone, Independent Component Analysis: an Introduction, Trends in Cognitive Sciences Vol.6 No.2, 2002

[28] Aapo Hyvarinen, Erkki Oja, Independent Component Analysis: A Tutorial, www.cis.hutfi/projects/ica, April 1999

[29] A. J. Bell, T. J. Sejnowski, Independent Components of Natural Scenes are Edge Filters, Vision Res. Vol 37 No23, 1997

[30] A. Papoulis, Probability, Random Variables and Stochastic Processes, McGraw-Hill, 1991

[31] Aapo Hyvarinen, Survey on Independent Component Analysis, Neural Computing Surveys 2, 1999

[32] A. LaVigna, Nonparametric Classification Using Learning Vector Quantization, PhD. Dissertation, University of Maryland, 1989

[33] T. Bojer, T. Hammer, Relevance Determination in Learning Vector Quantization, Proc. of European Symposium on Artificial Neural Networks, 2001

[34] D. Nair, J. K. Aggarwal, Bayesian Recognition of Targets by Parts in Second Generation Forward Looking Infrared Images, Image and Vision Computing 18, 2000

[35] Clark F. Olson, Daniel P. Huttenlocher, Automatic Target Recognition by Matching Oriented Edge Pixels, IEEE Transactions on Image Processing 6, January 1997

[36] X. Wu, Bir Bhanu, Gabor Wavelet Representation for 3-D Object Recognition, , IEEE Transactions on Image Processing 6, January 1997

[37] S. Z. Der, R. Chellappa, Probe-Based Automatic Target Recognition in Infrared Imagery, IEEE Transactions on Image Processing 6, January 1997

[38] L. I. Perlovsky, W. H. Schoendorf, Model-Based Neural Network for Target Detection in SAR Images, IEEE Transactions on Image Processing 6, January 1997