# Invariant Image Object Recognition using Gaussian Mixture Densities

Von der Fakultät für Mathematik, Informatik
und Naturwissenschaften der Rheinisch-Westfälischen
Technischen Hochschule Aachen
zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom–Informatiker Jörg Dahmen

aus

Jülich

Berichter:
Universitätsprofessor Dr.-Ing. H. Ney
Universitätsprofessor Dr.-Ing. D. Meyer-Ebrecht

Tag der mündlichen Prüfung: 29. Oktober 2001

Diese Dissertation ist auf den Internetseiten der
Hochschulbibliothek online verfügbar.

# Danksagung

An dieser Stelle möchte ich allen Personen meinen Dank aussprechen, die - auf unterschiedlichste Art und Weise - das Zustandekommen dieser Arbeit ermöglicht haben.

Besonderer Dank gilt Herrn Prof. Dr.-Ing. Hermann Ney, der mich zu Beginn meiner Arbeit dazu ermutigte, auf dem Gebiet der statistischen Mustererkennung in Bildern zu promovieren und eine entsprechende Aktivität am Lehrstuhl für Informatik VI zu initieren. Sein stetiges Interesse an meinen Forschungsarbeiten hatte einen erheblichen Einfluß auf diese Arbeit.

Danken möchte ich auch Herrn Prof. Dr-Ing. Dietrich Meyer-Ebrecht vom Lehrstuhl für Messtechnik und Bildverarbeitung der RWTH Aachen, der sich sofort bereiterklärte, die Funktion des Zweitgutachters für diese Arbeit zu übernehmen. Unsere zahlreichen Diskussionen haben mir geholfen, die Brücke zwischen "klassischer" Bildverarbeitung und statistischer Mustererkennung zu schlagen.

Besonders danken möchte ich auch meinem i6-Bildverarbeitungskollegen und ehemaligem Diplomarbeiter Daniel Keysers, für buchstäblich zahllose Diskussionen, Ideen, Implementierungen, eine hervorragende Zusammenarbeit und nicht zuletzt für die gemeinsamen "außeruniversitären" Aktivitäten. Ebenso gilt mein Dank meinen ehemaligen Hiwis und Diplomarbeitern Mark Oliver Güld, Ralf Perrey, Thomas Theiner und Alex Crämer, die mich bei vielen Implementierungen und Experimenten sehr unterstützt haben.

Danken möchte ich auch Achim Sixtus, Michael Motter und Oliver Bender für die unermüdliche Unterstützung in Fragen der Rechnerbetreuung. Weiterhin danke ich allen aktuellen und ehemaligen Kollegen am Lehrstuhl, besonders Klaus Beulen, der mich in der Anfangsphase meiner Arbeit sehr unterstützt und mitgeholfen hat, den Stein dieser Arbeit ins Rollen zu bringen. Danken möchte ich auch Ralf Schlüter für seine Unterstützung bei den Experimenten zum diskriminativen Training und Wolfgang Macherey für viele interessante Diskussionen.

Ganz besonders bedanken möchte ich mich aber bei meiner Familie: Bei meiner Frau Anne und meiner (noch) kleinen Tochter Pia, die mich jederzeit unterstützt haben, mir vor allem aber auch das nötige "Abschalten" von der Arbeit ermöglicht und mich nach den - zum Glück wenigen - schlechten Tagen jederzeit wieder aufgerichtet haben. Bei meinen Eltern Karin und Günter Dahmen, die meine Studien und die sich daraus ergebende Promotion überhaupt erst ermöglicht haben - und weit mehr als nur das. Danke für alles! Danke sagen möchte ich auch meinen Geschwistern Katja und Klaus Dahmen, die mich auf meinem Weg jederzeit unterstützt und ermutigt haben.

## Abstract

In this work, a statistical image object recognition system is presented, which is based on the use of Gaussian mixture densities in the context of the Bayesian decision rule. Optionally, to reduce the number of free model parameters, a linear discriminant analysis is applied. This baseline system is then extended with respect to the incorporation of invariances. To do so, we start by suitably multiplying the available reference images. This idea is then applied to the observations to be classified, too, yielding the novel 'Virtual Test Data' method, which has some desirable advantages over classical classifier combination approaches. Furthermore, global invariances are incorporated by using the so-called tangent distance. In this work, tangent distance is embedded into a statistical framework, which for instance leads to a modified, more reliable estimation of the mixture density parameters. Furthermore, tangent distance is extended to compensate not only for global, but also for local image transformations (*distorted tangent distance*).

A large part of the experiments was performed on the well known US Postal Service standard corpus for handwritten digit recognition. Furthermore, the proposed classifier was successfully applied to the recognition of medical radiographs, red blood cells as well as to the Columbia University Object Image Library (COIL-20) and the Max-Planck Institute's Chair Image Database. The obtained error rate of 2.2% on the US Postal Service corpus is the best error rate published so far on this particular data set.



## Zusammenfassung

In dieser Arbeit wird ein statistisches Objekterkennungssystem für Bilder vorgestellt, welches auf der Verwendung von Gauß'schen Mischverteilungen im Kontext der Bayes'schen Entscheidungsregel beruht. Zur Reduktion der freien Modellparameter wird dabei optional eine lineare Diskriminanzanalyse verwendet. Dieses Basissystem wird dann um die Berücksichtigung von Invarianzen erweitert. Zu diesem Zweck werden zunächst die vorhandenen Trainingsdaten geeignet vervielfacht. Diese Idee wird dann auf zu klassifizerende Testdaten übertragen und liefert die neue 'Virtual Test Data' Methode, die einige Vorzüge gegenüber Methoden der Klassifikatorkombination aufweist. Weiterhin wird eine Berücksichtigung globaler Invarianzen durch die Verwendung der sogenannten Tangentendistanz erreicht. Diese wird in der vorliegenden Arbeit in einen statistischen Rahmen eingebettet, was unter anderem zu einer modifizierten, zuverlässigeren Schätzung der Mischverteilungsparameter führt. Außerdem wird die Tangentendistanz um die Berücksichtigung lokaler Bildtransformationen erweitert (*distorted tangent distance*).

Ein Großteil der Experimente wurde auf dem bekannten US Postal Service Standard-Korpus für die Erkennung handgeschriebener Ziffern durchgeführt. Außerdem wurde der vorgestellte Klassifikator erfolgreich angewandt auf die Klassifikation medizinischer Röntgenbilder, roter Blutzellen sowie auf die Columbia University Object Image Library (COIL-20) und die Chair-Image Database des Max-Planck Instituts. Die auf dem US Postal Service Korpus erzielte Fehlerrate von 2.2% ist dabei die bislang beste publizierte Fehlerrate auf dieser Datensammlung.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

In the last years, the use of a statistical classification approach [Devroye$^+$ 1996, Fukunaga 1990, Duda & Hart 1973, Devijver & Kittler 1982] proved to be very successful in various fields of pattern recognition, among them speech recognition [Ney$^+$ 1998, Sixtus$^+$ 2000] and machine translation [Och & Ney 2000]. Furthermore, it is widely accepted that in speech recognition, the use of Gaussian mixture densities (GMD) – in combination with hidden Markov models – defines the state-of-the-art approach to tackle this particular problem. Motivated by these experiences, the goal of this work is to find out how well a mixture density based classifier performs in the field of image object recognition and how it compares to commonly accepted state-of-the-art approaches such as artificial neural nets (ANN) [Rojas 1993] or support vector machines (SVM) [Vapnik 1995]. Throughout this work, to achieve a meaningful comparison of different classification approaches, the proposed classifier is applied to different well known standard corpora, for which many results of other research groups have been reported (cp. Chapter 2).

Object recognition in images is a very important task in many real-world applications, among them

- the recognition of handwritten characters and digits, which greatly improves the interaction between humans and computers.
- industrial applications, such as robot vision or quality control in industrial manufacturing processes. In this case, possible defects are interpreted as objects which are to be detected.
- medical applications, such as the automated evaluation of medical image data. Typical tasks are for instance counting cells in a medical probe or classifying tumors as malignant or benign.
- image or video indexing. Interpreting an image as to be composed out of multiple objects, an image index can be automatically obtained by detecting and classifying the objects present in a given scene. Given suitable similarity measures, image or video retrieval can then be obtained using object recognition algorithms.

- biometric applications, such as fingerprint or face recognition. These applications are crucial for the successful implementation of state-of-the-art security systems.

Although the above list is far from being complete, it should emphasize the fact that object recognition is an important tool that many practical applications require. As the considerations in the following chapters show, state-of-the-art results can be obtained by simply applying the experiences gained in speech recognition to the problem of object recognition in images (i.e. use Gaussian mixtures and Fisher's linear discriminant analysis in a Bayesian context). Yet – not surprisingly – superior results can be obtained by taking into consideration the special properties of image data. Among these, the incorporation of invariances into the classifier (with respect to certain image transformations) plays a very important role. For instance, if a robot shall be able to seize a certain object, he must first be able to recognize the object regardless of his position (vertical shift, horizontal shift, rotation etc.) or scale. Contrary to this example, where a full invariance is desirable (for instance a full rotation invariance), there are other applications in which one might be interested in partial invariances only, also called *transformation tolerance* sometimes. For instance, a slightly rotated version of the digit '6' is still a '6', yet a full rotation invariance would confuse the classes '6' and '9' in many cases. Thus, optical character recognition is one example for an application where only partial invariance is needed.

Transformations affecting the whole image – as is the case for affine transformations [Lehmann$^+$ 1997, pp. 324 ff.] such as scaling, rotation or shift – are called *global transformations* in the following. In many applications, *local transformations* of a given image play an important role, too. For instance, the position of the scribor[1] in medical radiographs is not normalized. Therefore, two more or less identical images may only differ by the position of the respective scribors, rising the need for local transformation models (also called local perturbation models sometimes). Otherwise, the varying scribor position in the two images might lead to a misclassification. The incorporation of such invariances - global and local - into a statistical, Gaussian mixture density based classifier is one of the key issues of this thesis.

Throughout this work, the main emphasis is put on evaluating the effectiveness of the proposed statistical methods, which is done by applying them to standard image corpora (especially the US Postal Service handwritten digits recognition task) and by comparing the obtained error rates to those reported by other research groups. Nevertheless, the practical applicability of the methods is also shown by applying them to two practical medical problems, namely the classification of radiographs respectively red blood cells.

## 1.1 Pattern Recognition

The problem to be solved by pattern recognition algorithms is the following: Given a signal $s$ belonging to a class $k, k = 1, ..., K$, a decision function is to be constructed

[1] The scribor is a data field containing all necessary patient information, such as patient name, date of birth etc..

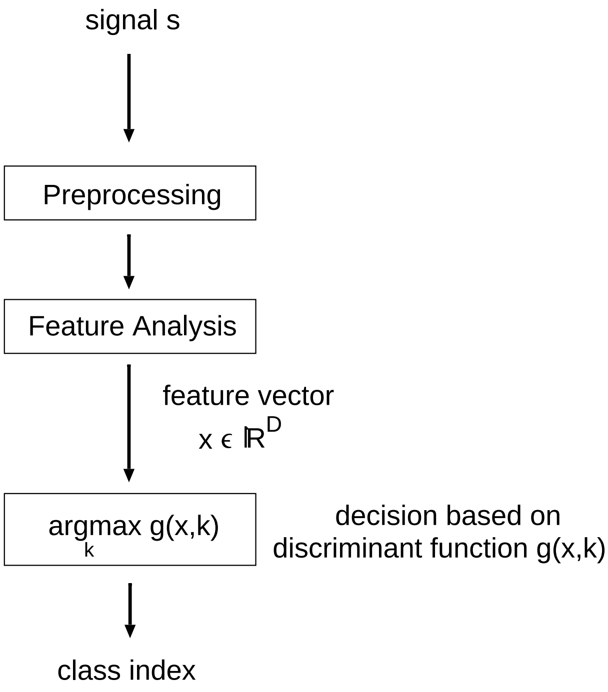


Figure 1.1: Typical structure of a recognition system.

which determines the class the signal belongs to. To do so, usually a number of features is extracted from the original signal, which form the so-called feature vector $x \in \mathbb{R}^D$.

Thus, a decision function

$$\begin{aligned} r : \mathbb{R}^D &\rightarrow \{1, ..., K\} \\ x &\longmapsto r(x), \end{aligned} \tag{1.1}$$

must be determined. In many cases, this is done using a discriminant function $g(x, k)$:

$$r : \; x \longmapsto \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \; \{g(x, k)\} \tag{1.2}$$

where the criterion for the discriminant function usually is

$$\begin{aligned} g(x, k) &\longmapsto 1 \qquad \text{for the “right” class} \\ g(x, k) &\longmapsto 0 \qquad \text{for the “false” class} \end{aligned} \tag{1.3}$$

Figure 1.1 illustrates the basic structure of a classifier. Usually, the feature analysis step is preceded by a preprocessing step. In image object recognition for instance, this could be a grayscale normalization or a segmentation of the objects present in the given image.

### 1.1.1 Bayes Rule

In statistical pattern recognition, the Bayesian decision rule is often used to model $g(x, k)$. In that particular case, the class $k$ is chosen which maximizes the posterior probability $p(k|x)$ given an observation $x$ to be classified:

$$\begin{aligned} r(x) &= \operatorname*{argmax}_k \{p(k|x)\} \\ &= \operatorname*{argmax}_k \left\{ \frac{p(k) \cdot p(x|k)}{\sum_{k'=1}^{K} p(k') \cdot p(x|k')} \right\} \end{aligned} \tag{1.4}$$

As the denominator of Equation (1.4) does not depend on $k$, it can be neglected for classification purposes, arriving at

$$r(x) = \operatorname*{argmax}_k \{p(k) \cdot p(x|k)\}, \tag{1.5}$$

where $p(k)$ is called the *prior probability* of class $k$ and $p(x|k)$ is the *class-conditional probability* for the observation $x$ given class $k$. It can be shown that Bayes rule is optimal with respect to the expected number of errors in case the true distributions $p(k)$ and $p(x|k)$ are known [Duda & Hart 1973, pp. 10-39]. Note that this implies the assumption of a cost function assigning cost one to a misclassification and cost zero to a correct classification. It does not hold for the case of weighted error functions that is some errors might be 'more expensive' than others. For instance, a false-positive cancer detection in a medical application could be a 'cheap' misclassification (as the following examinations will show that the patient does not suffer from cancer), whereas a false negative result should result in high costs (as the patient is regarded to be healthy, delaying the necessary cancer therapy).

Since the true distributions are usually unknown, one has to choose suitable models for $p(k)$ and $p(x|k)$ in order to use Bayes rule in real world applications. The free parameters of these models are then estimated during the training phase. Throughout this work, the training phase is supervised that is one is given training data as a set of labelled pairs $(x_n, k_n)$, $n = 1, \ldots, N$ where $x_n$ is a feature vector belonging to class $k_n$. This training data is then used to estimate the free model parameters. Further information on this statistical approach can be found in Chapter 4.

The performance of a classifier is usually measured by its error rate on a particular data set. Thus, a certain number of observations is classified and the error rate is being defined as the ratio between the number of misclassifications and the total number of trials performed.

### 1.1.2 Maximum Likelihood Parameter Estimation

A widely used method for parameter estimation given a set of training data is *maximum likelihood estimation*. Consider a density function $p(x|k, \lambda_k)$ that depends on a parameter set $\lambda_k$, which in turn depends on the class $k$ to be modeled. For each class, a number of $N_k$ training samples $x_{1k}, ..., x_{nk}, ..., x_{N_k k} \in \mathbb{R}^D$ is given, resulting in the *likelihood function*

$$\lambda_k \longmapsto \prod_{n=1}^{N_k} p(x_{nk}|k, \lambda_k) \tag{1.6}$$

respectively the *log-likelihood function*

$$\lambda_k \longmapsto \sum_{n=1}^{N_k} \log p(x_{nk}|k, \lambda_k) \tag{1.7}$$

Now, the so called *maximum likelihood estimator* $\hat{\lambda}_k$ is defined by

$$\begin{aligned} \hat{\lambda}_k &:= \underset{\lambda_k}{\operatorname{argmax}} \left\{ \prod_{n=1}^{N_k} p(x_{nk}|k, \lambda_k) \right\} \\ &= \underset{\lambda_k}{\operatorname{argmax}} \left\{ \sum_{n=1}^{N_k} \log p(x_{nk}|k, \lambda_k) \right\} \end{aligned} \tag{1.8}$$

i.e. the maximum likelihood estimation of the free model parameters maximizes the (log) likelihood function. Note that in this case, parameter estimation is performed separately for each class $k$. Contrary to this, the term *discriminative training* is used for approaches that take the posterior probability as a criterion for the training phase, for example

$$\lambda \longmapsto \prod_{n=1}^{N} p(k_n|x_n, \lambda) \tag{1.9}$$

respectively the logarithm

$$\lambda \longmapsto \sum_{n=1}^{N} \log p(k_n|x_n, \lambda) \tag{1.10}$$

These methods are called discriminative, because they take into account the relation between the classes and thus aim at optimizing class separability. Note that classifiers such as artificial neural nets or support vector machines (see below) are inherently discriminative. Further information on maximum likelihood respectively discriminative training is given in Chapter 4.

### 1.1.3 Feature Reduction

To reduce the number of free model parameters that have to be estimated, it is sometimes advisable to perform a feature reduction step on the original feature vectors. The general idea is to find some suitable function $\varphi : \mathbb{R}^D \longmapsto \mathbb{R}^d, d << D$, which maps the original feature vectors $x \in \mathbb{R}^D$ into an appropriate d-dimensional subspace (i.e. $\varphi(x) = y$, where $x \in \mathbb{R}^D, y \in \mathbb{R}^d$). In the following, two well known methods to determine the desired mapping function are described.

#### 1.1.3.1 Principal Components Analysis

The principal components analysis (PCA) is a linear transformation that aims at minimizing the expected reconstruction error

$$||x - \hat{x}||^2 \tag{1.11}$$

where $\hat{x} = \varphi^{-1}(y)$. It can be shown that the PCA can be computed as follows:

In a first step, compute the eigenvectors and eigenvalues of the empirical covariance matrix $\Sigma$ of the data, where

$$\Sigma = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)\cdot(x_n - \mu)^T, \tag{1.12}$$

$$\mu = \frac{1}{N}\sum_{n=1}^{N} x_n. \tag{1.13}$$

In a second step, compute the projection of the data into the subspace spanned by the first $d$ *principal components* of $\Sigma$ (i.e. the eigenvectors with the $d$ largest corresponding eigenvalues).

Besides minimizing the expected reconstruction error as given in Equation (1.11), the PCA has the interesting property that it decorrelates the feature space. Thus, the empirical covariance matrix (1.12) is diagonal. Normalizing the length of the eigenvectors by dividing each component by the square root of the corresponding eigenvalue, the covariance matrix becomes the matrix of identity. This transformation is sometimes called a whitening transformation. Further information on the PCA and the whitening transformation can be found in [Fukunaga 1990]. Note that no class information is used when computing the PCA. Thus, although it is often used in pattern recognition task, nothing can be said about the discriminative power of the computed features.

#### 1.1.3.2 Linear Discriminant Analysis

Contrary to the principal components analysis, the linear discriminant analysis (LDA) aims at maximizing the class separability of the transformed data. The LDA can be computed as follows:

In a first step, compute the within-class-scatter matrix $S_w$ and the between-class-scatter matrix $S_b$:

$$S_w = \sum_{k=1}^{K}\sum_{n=1}^{N_k}(x_{nk} - \mu_k)\cdot(x_{nk} - \mu_k)^T \tag{1.14}$$

$$S_b = \sum_{k=1}^{K} N_k\cdot(\mu_k - \mu)\cdot(\mu_k - \mu)^T \tag{1.15}$$

and compute the eigenvectors and eigenvalues of the matrix $S_w^{-1}\cdot S_b$. In a second step, compute the projection of the data into the subspace spanned by the first $d$ principal

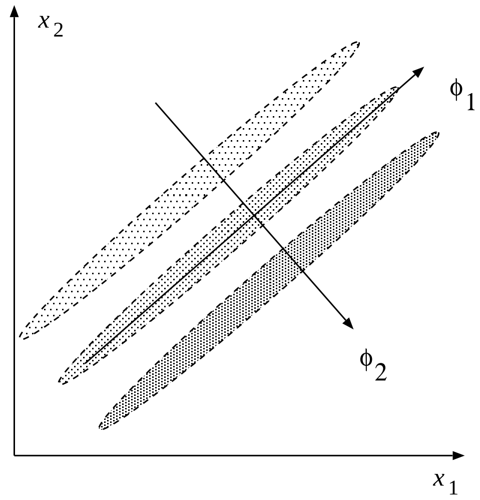


Figure 1.2: The 'Adidas-Problem': Behaviour of LDA vs. PCA

components of $S_w^{-1} \cdot S_b$. To avoid the inversion of $S_w$, the LDA can also be computed by solving a general eigenvalue problem in $S_w$ and $S_b$ [Duda & Hart 1973].

As the overall mean vector $\mu$ is a linear combination of the class-specific mean vectors $\mu_k$, the maximal rank of $S_b$ is $K - 1$. Thus, application of a linear discriminant analysis yields a maximum of $K - 1$ features. To overcome this shortcoming in the presence of only few classes, the creation of *pseudoclasses* is suggested in this work. This is done by performing a cluster analysis on the available data and by interpreting each of the resulting clusters as a pseudoclass. For instance, in the US Postal Service digit recognition experiments, four pseudoclasses are created per class, yielding a reduced feature space of 39 dimensions. An example of the different behaviour of LDA and PCA is given in Figure 1.2. The principal component analysis aims to preserve as much variance as possible in the reduced features. Thus, the data is projected onto direction $\Phi_1$. In opposite to this, the linear discriminant projects the data onto direction $\Phi_2$, resulting in reduced features which do only preserve a small part of the variance contained in the original data, but which allow for a perfect separation of classes.

It should be noted here that in almost any practical case feature reduction means a loss of information, as it can be shown that the information gained from an additional feature is always greater or equal to zero [Fukunaga 1990]. Yet, this loss is usually compensated by a more reliable parameter estimation in the reduced feature space (cp. experimental results in Chapter 9).

## 1.2 Examples for Object Recognition Systems

In this section, well known pattern recognition/ object recognition approaches are briefly described. The performance of these approaches on typical object recognition tasks is considered in Chapter 2.

### 1.2.1 Nearest Neighbour Classification

One of the simplest classifying approaches and at the same time maybe the best known example for a recognition system is a 1-Nearest Neighbour classifier (1-NN). In this case, an observation $x$ is classified as belonging to the class $k$, to which its nearest neighbour from the training data belongs:

$$r(x) = \underset{k}{\operatorname{argmin}} \left\{ \min_{n=1,\ldots,N_k} ||x - x_{nk}|| \right\} \tag{1.16}$$

where $x_{nk}$ is the n-th reference image, $N_k$ is the number of reference images of class $k$ and $||...||$ is an arbitrary distance measure. In many cases, (squared) Euclidean distance is chosen. The approach can be modified by taking into account the $k$-nearest neighbours. Nearest neighbour based classifiers are often blamed for the amount of memory they require and their computational complexity. Therefore, many techniques have been developed to suitably reduce the number of reference vectors required, among them the editing or condensing techniques [Devijver & Kittler 1982]. These techniques try to reduce the available references to those lying near class-borders in feature space. Thus, they are somewhat related to the idea of support vector machines. Another method to reduce the number of references (which is directly related to support vector machines) is the reduced set method [Burges 1996] Yet, on todays state-of-the-art computers these drawbacks are somewhat alleviated and nearest neighbour techniques are applicable in many real-world problems. Throughout this work – because it is very easy to implement – a 1-nearest neighbour based classifier is often used to produce baseline error rates to be compared with more sophisticated approaches: "*Nearest neighbour classifiers are extremely simply and always worth trying as a benchmark with any classification task.*" [Hastie$^+$ 1998].

### 1.2.2 Artificial Neural Nets

Artificial neural nets (ANN) try to mimic the operation of the human brain [Rojas 1993]. An artificial neural net usually consists of multiple layers of connected nodes, called *neurons* (it can be shown that one hidden layer (i.e. a net with input-, output- and one additional layer) is sufficient to model an arbitrary function [Ney 1999]). At each node a weighted sum of all input signals is computed. The output of a node is then computed to be a non-linear function of this weighted sum (usually, a sigmoid function is used). In many cases, given observations $x \in \mathbb{R}^D$ coming from $K$ classes, the input layer of an artificial neural net consists of $D$ neurons and the output layer of $K$ neurons. The output neuron with maximal activation then determines the class to which an observation is classified. Once the topology of the net has been chosen (number of layers, number of nodes, which neurons are connected etc.), the training problem is to chose the required weight coefficients in such a way that the net 'explains' the available training data as well as possible (usually, a mean squared error criterion is used). One of the best known training procedure for artificial neural nets is the *error-backpropagation* method [Rojas 1993, pp. 149]. Interestingly, it can be shown that the expected error rate of an artificial neural net is minimized, if the outputs of the net equal the posterior probabilities $p(k|x)$ [Ney 1995].

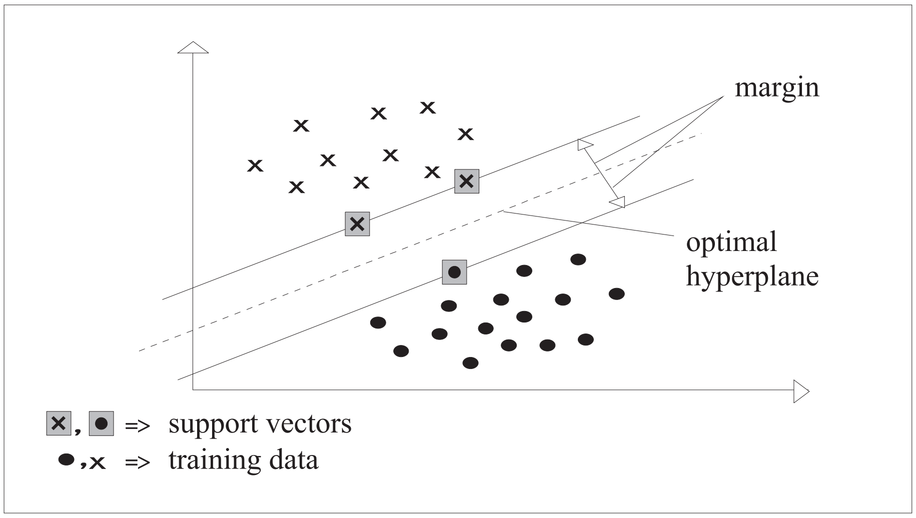


Figure 1.3: 2D example of a SVM: Support vectors and optimal hyperplane.

### 1.2.3 Support Vector Machines

The support vector machine (SVM) approach is originally a linear classifier for two-class problems, which can be applied to $K$-class problems by reformulating these as $K$ two-class problems [Vapnik 1995, Cortes$^+$ 1995, Schölkopf 1997]. Given the training data, a support vector machine computes that particular hyperplane in feature space, which separates both classes in an 'optimal' way. The sought for hyperplane is optimal in the sense that it has maximal distance from both classes (cp. Figure 1.3). This property is desirable, as such a hyperplane is expected to have the best generalization properties (that is, it is likely to also produce good classification results on unseen data). For that reason, the support vector machine is also called *optimal margin classifier*. It can be shown that the computation of this hyperplane can be obtained by solving a quadratic optimization problem, for which efficient algorithms are known [Künzi 1979]. An interesting property of the optimal hyperplane is the fact that it is not affected by all reference vectors, but only by those lying closest to it. These vectors are called *support vectors* (see Figure 1.3).

To overcome the drawback that a support vector machine only realizes a linear classifier, the given data is implicitly projected into a very high-dimensional feature space. Linearly separating the data in this space now yields a highly non-linear classifier in the original feature space. It can be shown that explicitly mapping the data into this high-dimensional space is not necessary in many practical applications. Because a support vector machine is based on the computation of scalar products, it is sufficient to introduce so-called kernel functions. Calculation of the scalar product between two vectors in the high-dimensional space can then be shown to equal the application of certain kernel functions to the result of the scalar product in the original feature space [Vapnik 1995, pp. 133 ff.].

Recently, the idea of the support vector machine has been extended to the probabilistic relevance vector machine [Tipping 2000].

## 1.3 Incorporation of Invariances

The classifying approaches presented above are not invariant to image transformations inherently. Instead, they are only approximately invariant with respect to image variations that are present in the training data. Therefore, this section briefly describes several approaches to gain additional invariance properties, which is especially helpful in the presence of rather small training data sets, which do not contain all possible variations of the given data. It should be noted that this is the case for most practical applications.

### 1.3.1 Normalization

Normalization of the given images affects the preprocessing step of a general classifying system as shown in Figure 1.1. The aim of a normalization process is to construct a canonical representation of a given object, in which the transformations considered are eliminated. For instance, to achieve invariance to additive illumination changes, it is sufficient to normalize all given images to have a mean graylevel of zero. A more complex normalization procedure can be performed to obtain invariance with respect to rotation, scale and translation of images (RST-invariance) [Güld 2000, Wood 1996]:

- Compute the center of gravity and translate the origin to that point (translation-invariance).
- Normalize for the average radius (scale-invariance).
- Rotate such that the direction of the maximum variance coincides with the x-axis (rotation-invariance).

A drawback of such normalization procedures is the fact that they often depend on a segmentation of the objects contained in an image and that they may be very sensitive to noise. Furthermore, moment-based normalization steps (as the computation of the center of gravity in the above procedure) only yield meaningful results if the intra-class variability of the objects regarded is negligible [Süße 1999].

### 1.3.2 Extraction of Invariant Features

Another approach to achieve transformation invariance is to extract invariant features in the feature analysis step. In the literature, one can find many approaches to do so, among them

- the computation of invariant moments. Usually, the moments proposed by Hu [Hu 1962] or Zernike [Perantonis$^+$ 1992] are used.
- translation-invariant features based on the power spectrum of the Fourier transform [Wood 1996] (the power spectrum being defined as the squared magnitude of the Fourier spectrum).
- rotation, translation and scale invariant features based on the Fourier-Mellin transformation [Reddy$^+$ 96].

- features based on (color) histograms [Siggelkow+ 98, Smith & Chang 1996, Schiele & Crowley 1996, Zhang+ 1995].
- translation invariant features based on monomials [Schulz-Mirbach 1992,1995].

A common problem of invariant feature extraction methods is that in many cases, a significant part of the information contained in the original images is lost. For instance, in most Fourier-based methods, the phase information of the Fourier spectrum is discarded. Using the invariant moments as proposed by Hu, all the information contained in the regarded object is reduced to seven moments, which obviously implies a considerable loss of information. Thus, it is not guaranteed that invariant features are discriminative features at the same time. As an example, the mapping of each image to a constant value results in a perfectly invariant, yet at the same time completely useless feature. Therefore, *complete* invariant features are of special interest, as these only eliminate the degrees of freedom of the respective transformations. Examples for such features are translation invariant features based on monomials [Burkhardt+ 92, Schulz-Mirbach 1992].

In the following, the Fourier transform based extraction of RST invariant features is briefly described, as these features are used in the red blood cell experiments conducted throughout this work. The calculations are given for 1D signals, but can analogously be extended to (and also hold for) 2D signals (see for instance [Schalkoff 1989, pp. 90 ff.]).

#### 1.3.2.1 Shift Invariance via Fourier Transform

The continuous 1D-Fourier transform $H(\omega)$ of a signal $h(t)$ is defined as

$$\mathcal{F}\{h(t)\} = H(\omega) := \int_{-\infty}^{\infty} h(t) \cdot e^{-i\omega t} dt \tag{1.17}$$

Thus, for the Fourier transform of a translated function one obtains

$$\begin{aligned}
\mathcal{F}\{h(t-t_0)\} &= \int_{-\infty}^{\infty} h(t-t_0) \cdot e^{-i\omega t} dt \\
&= e^{-i\omega t_0} \int_{-\infty}^{\infty} h(t-t_0) \cdot e^{-i\omega (t-t_0)} dt \\
&\overset{\tau=t-t_0}{=} e^{-i\omega t_0} \int_{-\infty}^{\infty} h(\tau) \cdot e^{-i\omega \tau} d\tau \\
&= e^{-i\omega t_0} \cdot H(\omega)
\end{aligned} \tag{1.18}$$

Obviously,

$$
\begin{aligned}
|e^{-i\omega t_0}| &= \sqrt{(\cos(-\omega t_0))^2 + (\sin(-\omega t_0))^2} \\
&= \sqrt{(\cos(\omega t_0))^2 + (-\sin(\omega t_0))^2} = 1
\end{aligned}
$$

and it follows

$$|\mathcal{F}\{h(t - t_0)\}| = |\mathcal{F}\{h(t)\}| \tag{1.19}$$

Thus, the magnitude of the Fourier spectrum is invariant with respect to translation. Therefore, in many cases the squared magnitude (called the power spectrum) is applied for shift invariant pattern recognition. Other interesting properties of the Fourier transform, which are used in the following to extend the approach presented above to the extraction of RST invariant features, are:

- The Fourier spectrum is rotation variant (that is, a rotation of the image results in a rotation of the spectrum) and
- it is inversely variant with respect to scaling (enlarging the image shrinks the spectrum).

#### 1.3.2.2 The Fourier Mellin Transform

As can be seen from the above, shift invariant pattern recognition can be obtained using the invariance property of the Fourier transform. If RST-invariance is desired, this can be achieved with variants of the Fourier transform, for instance the *Mellin transform.* This a Fourier transform evaluated over an exponential scale, which is invariant under the scaling transformation [Reddy$^+$ 96, Perrey 2000]. If aspects of the Fourier and Mellin transform are combined with a transformation to polar coordinates of an image (resulting in a circular Fourier, radial Mellin transform), one can achieve invariance with respect to rotation, scaling and translation simultaneously. The resulting transform is called *Fourier-Mellin transform* and can be calculated in the following way [Reddy$^+$ 96, Wood 1996]:

(1) Calculate the power spectrum of the Fourier transform of the two-dimensional input.
This is invariant under translation.

(2) Convert the power spectrum to polar coordinates.
This converts rotations to translations.

(3) Perform a complex-log mapping.
This converts scalings to translations.

(4) Calculate another two-dimensional Fourier transform power spectrum.
This is rotation-, scale- and translation-invariant.

The resulting features are now RST-invariant, but it should be noted that a lot of information is lost due to usage of only magnitudes in steps (1) and (4). An example of this transform is given in Figure 1.4. It shows a rotation example for the image of a red blood cell.

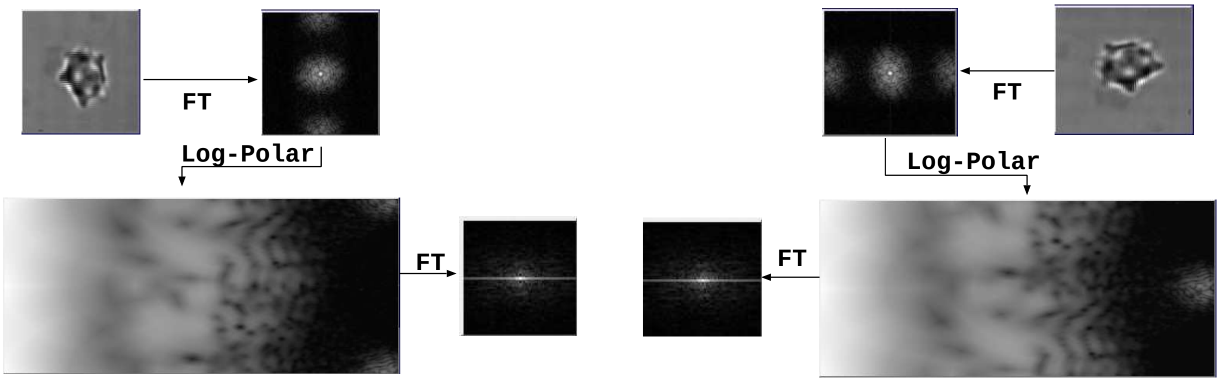


Figure 1.4: RST-invariant feature extraction: A 90° rotation example. Note that the image rotation becomes a vertical shift in the log-polar plane.

### 1.3.3 Invariant Distance Measures

While normalization and the extraction of invariant features aim at the elimination of the considered transformations before the actual classification process takes place, invariance can also be incorporated directly into the classifier. This can be done by using *invariant distance measures*. An invariant distance measure - in the ideal case - would have the property that the distance between two patterns is always equal to the minimum distance between the 'best matching' transformed instances of those patterns. Since the orbits that arise from regarding the set of all possible transformations of a pattern form a *manifold* in pattern space, this ideal invariant distance is called manifold distance. A manifold is a locally Euclidean space together with a differential structure, which has the same local properties as $\mathbb{R}^D$, but may have different global properties. One can also think about a manifold as a generalization of surfaces in $\mathbb{R}^D$ [Keysers 2000a]. The main problem with the notion of a manifold distance is that it is in most cases a very hard problem to determine the minimum distance, because the manifolds are difficult to handle. Furthermore, the required manifolds do not have an analytic representation in many cases [Simard+ 1993].

Since probability density functions are often based on a distance function, one can use invariant distance measures to define transformation invariant probability distributions. For instance, regarding the negative logarithm of a Gaussian distribution

$$\begin{aligned}
\mathcal{N}(x|\mu,\Sigma) &= \frac{1}{\sqrt{\det(2\pi\Sigma)}}\ \exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right] \\
&= \frac{1}{\text{norm}}\ \exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right]
\end{aligned}$$

one obtains

$$-\log\left\{\mathcal{N}(x|\mu,\Sigma)\right\} = \frac{1}{2}\cdot(x-\mu)^T\Sigma^{-1}(x-\mu) + \log\{\text{norm}\}. \tag{1.20}$$

Thus, the negative logarithm of a probability distribution can be interpreted as a distance measure. On the other hand, one can show that starting from a distribution invariant with respect to some transformation, an invariant distance measure can be derived. The two concepts may therefore be regarded as equivalent.

One of the most common distance measures used in pattern recognition is (squared) Euclidean distance. For images $x$ and $\mu$ of size $I \times J$ pixels, the squared Euclidean distance is defined by

$$d(x,\mu) = ||x-\mu||^2 = \sum_{i=1}^{I}\sum_{j=1}^{J} ||x_{ij}-\mu_{ij}||^2. \tag{1.21}$$

Other distance measures are for instance the dot product between two vectors $x, \mu \in \mathbb{R}^D$

$$x^T \cdot \mu = \sum_{d=1}^{D} x_d \cdot \mu_d, \tag{1.22}$$

which is related to the angle $\theta$ between $X$ and $\mu$

$$\theta = \arccos\frac{x^T\cdot\mu}{||x||\ ||\mu||} \Leftrightarrow \cos\theta = \frac{x^T\cdot\mu}{||x||\ ||\mu||}, \tag{1.23}$$

where the cosine of the angle is also called normalized dot product. A connection to the Euclidean distance is given by the relation

$$||x-\mu||^2 = ||x||^2 - 2x^T\cdot\mu + ||\mu||^2 \tag{1.24}$$

which - given that $||x|| = ||\mu|| = 1$ - can be simplified to

$$||x-\mu||^2 = 2(1 - x^T\cdot\mu), \tag{1.25}$$

The above distance measures are not invariant with respect to variations in the images like affine transformations, in fact they are very sensitive to such distortions. In the context of image object recognition SIMARD introduced a new locally invariant distance measure called tangent distance [Simard$^+$ 1993]: "Memory-based classification algorithms such as radial basis functions or K-nearest neighbors typically rely on simple distances (Euclidean, dot product...), which are not particularly meaningful on pattern vectors. More complex, better suited distance measures are often expensive and rather ad-hoc (elastic matching, deformable templates). We propose a new distance measure which (a) can be made locally invariant to any set of transformations of the input and (b) can be computed efficiently."

Examples for other invariant distance measures include:

- Extensions to the Hausdorff distance [Hutten+ 1999] or the local pixel distance [Smith+ 1994].
- Perturbation models, such as the extended Levenshtein distance [Moore 1979] or two-dimensional warping [Uchida+ 1998].

In the experiment conducted during this work, the use of tangent distance proved to be especially effective. It is therefore presented in more detail in Chapter 5. Furthermore, a probabilistic interpretation of tangent distance is given in Chapter 7.

### 1.3.4 Data Multiplication

Finally, a rather simple method to incorporate invariances into a statistical classifier is to multiply the available reference images using transformations that respect class membership. This method is also called 'data augmentation' by some authors. The basic idea is the following: If the classifier should be invariant with respect to image shifts, this can be obtained by simply applying shifts to the reference images and use the in that way augmented training data to train the system. Thus, the final classifier is approximately invariant to image shifts, as these have been 'seen' in the training step. This approach is described in detail in Chapter 6. It is furthermore extended to the test data as well, resulting in the proposed virtual test data method (VTS). Furthermore, we compare the virtual test sample method (which is basically an approach to perform combined classification) to conventional classifier combination schemes [Kittler+ 1998].

Although the creation of virtual data seems to be a rather naive approach, the best recognition results reported so far on the well known MNIST database were obtained by making extensive use of virtual data creation in combination with boosted artificial neural nets [Drucker+ 1993]. In fact, several million reference images were created starting from the available 60,000 original references. More information on this topic is given in Chapter 2. Furthermore, the idea of the invariant support vector machine is basically a support vector machine trained on virtually created training samples [Schölkopf+ 1996]. The only difference is that in this case only previously determined support vectors get multiplied.

## 1.4 Related Work

While appearance based image object recognition is common in the pattern recognition community, the use of invariant statistical classifiers such as the one proposed throughout this work is not. Among the few groups using this approach are Moghaddam & Pentland, who also use Gaussian mixture densities for view-based image recognition.

Yet, invariances are only accounted for by assuming an appropriate training set and by performing a suitable image normalization [Moghaddam & Pentland 1997]. SCHIELE & CROWLEY use local receptive field histograms as features within a Bayesian classifier, but do not use mixture densities to model the required probability densities [Schiele & Crowley 1996]. Instead, first-order statistics are applied (namely the receptive field histograms themselves). HINTON et al. apply tangent distance to define a modified version of a principal components analysis within a linear autoencoder based classifier [Hinton+ 1995]. This approach is similar to computing a maximum approximation within a mixture density based classifier. Furthermore, HASTIE et al. propose the computation of suitable prototype vectors from a given training set with respect to tangent distance. The approach was successfully used to speed up nearest neighbour classification (by using just a few prototype vectors instead of the possibly large, full training set) [Hastie+ 1995]. Not surprisingly, as tangent distance originated from the field of artificial neural nets [Simard+ 1998], many authors such as SCHWENK & MILGRAM use it in this context [Schenk & Milgram 1995]. SIMARD himself also used tangent distance within a 1-nearest neighbour setting, but unfortunately used a modified version of the US Postal Service database in his experiments. Thus, the best error rates obtained on this database were reported by SCHÖLKOPF et al., who applied the support vector machine [Schölkopf 1997], which was brought to the attention of the pattern recognition community by VAPNIK (see for instance [Vapnik 1995]). Incorporating invariances into this approach yields the so called invariant support vector machine, which is basically a support vector machine trained on virtually extended training data [Schölkopf+ 1996]. The idea of creating virtual training data is also applied to a boosted ensemble of artificial neural nets by DRUCKER et al., who reported the best results on the MNIST database [Drucker+ 1993]. Furthermore, LECUN et al. incorporated prior knowledge about the structure of a given recognition problem (in this case US Postal digit recognition) into an artificial neural net, proving that such a net is not necessarily a completely black box approach [Bottou+ 1994]. As for the incorporation of invariances into the recognition process, an interesting review of possible approaches to do so is given in [Wood 1996].

Finally, the virtual test sample method for combined classification (proposed in Section 6 of this work) was motivated by KITTLER's research on classifier combination schemes [Kittler+ 1998]. It should also be noted that some of the aforementioned groups use datasets which are not standardized (i.e. taken from special projects), so that a direct comparison of the obtained recognition results is not possible. This is one reason for the wide variety of databases used throughout this work, which allow for a comparison of the obtained results with most of the groups mentioned above.

# Chapter 2

# Databases and State-of-the-Art

In this chapter, the databases used in the experiments (cp. Chapter 9) are briefly presented, including the state-of-the-art results that have been reported by other international research groups.

## 2.1 The US Postal Service Task

The most important database for the experiments in this work is the well known US Postal Service Database (USPS). It consists of handwritten, isolated and normalized images of handwritten digits which were taken from from US zip codes. The images are sized 16×16 pixels and are quantized to 256 grayscales. The database contains a 7,291 references images and 2,007 test images and can be downloaded via FTP at `ftp://ftp.kyb.tuebingen.mpg.de/pub/bs/data`. The USPS test set is known as a hard recognition task, which can be inferred from the surprisingly high human error rate of 2.5% [Simard$^+$ 1993].

Figure 2.1 shows some example images for each of the ten classes taken from the USPS corpus. Despite of the normalization there is still a large variability in the data, which the classifier needs to take into account. Furthermore one can see segmentation artifacts, as is the case for the image of the digit '8' in the last row.

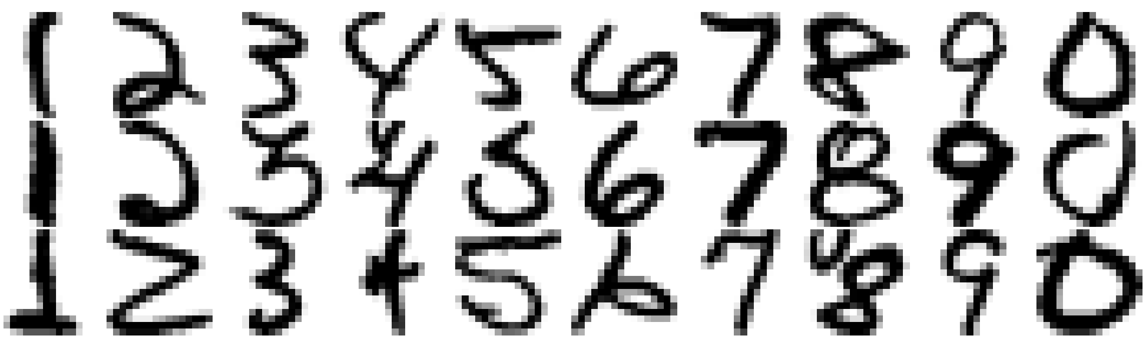

Figure 2.1: Example images taken from the USPS test set.

Table 2.1: Results reported on USPS.

| Author | Method | Error [%] |
|---|---|---|
| Simard$^+$, 1993 | Human Performance | 2.5 |
| Vapnik, 1995 | Decision Tree C4.5 | 16.2 |
| Vapnik, 1995 | Two-Layer Neural Net | 5.9 |
| Tipping, 2000 | Relevance Vector Machine | 5.1 |
| Simard$^+$, 1998 | Five-Layer Neural Net | 4.2 |
| Schölkopf, 1997 | Support Vector Machine | 4.0 |
| Schölkopf$^+$, 1998 | Invariant Support Vector Machine | 3.0 |
| Drucker$^+$, 1993 | Boosted ANN | *2.6 |
| Simard$^+$, 1993 | Tangent Distance | *2.5 |

*: 2,418 machine printed digits were added to the training set

One disadvantage of the corpus is that there exists no development test set, which leads to effects known as 'training on the testing data', yet this drawback holds for each of the research groups performing experiments on USPS. Ideally, a development test set would be used to determine the best parameters for the classifiers and the final results would be obtained from one run on the test set itself. Nevertheless, a comparison of 'best performing' algorithms may lead to valid conclusions. In [Hastie$^+$ 1998] the authors compare the performance of different algorithms on the USPS database and comment the subject with the following: "Although there is an official test set of data to be used to evaluate different methods, it can be overused. For example, a group may attempt tens or hundreds of different configurations, but only report the results of the best. These caveats hold for any technique with tunable parameters, but are especially pertinent for neural networks which have many."

On the other hand a definite advantage of the USPS task is the availability of many recognition results reported by international research groups, allowing for a meaningful (keeping in mind the above consideration) comparison of different classification approaches. Some of the results that have been reported on USPS can be found in Table 2.1. As can be seen, the best results reported so far using the original training and test set is 3.0% and was obtained by Schölkopf using an invariant support vector machine.

Note that despite the creation of virtual data – as it is performed throughout this work – the algorithms presented here are still based on the original USPS datasets, as virtual data creation only enriches the data by using transformations of the available images, i.e. by making use of available a-priori knowledge (applying a suitably small affine transformation does not affect the class membership of a digit). Contrary to [Drucker$^+$ 1993] or [Simard$^+$ 1993], no new images are added to the datasets.

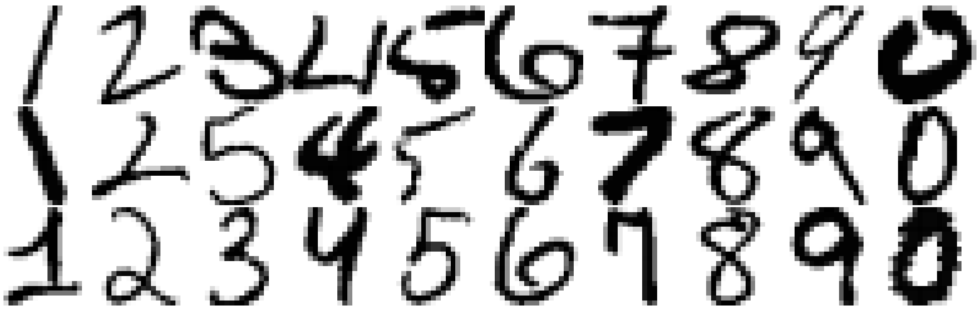

Figure 2.2: Example images taken from the NIST database.

Table 2.2: Results reported on MNIST.

| Author | Method | Error [%] |
|---|---|---|
| Simard$^+$, 1993 | Human Performance | 0.2 |
| Bottou$^+$, 1994 | Linear Classifier | 8.4 |
| LeCun 1990 [Bottou$^+$ 1994] | ANN (LeNet1) | 1.7 |
| Meinicke$^+$, 1993 | Local PCA, GMD | 1.6 |
| Cortes 1995 [Simard$^+$ 1998] | Support Vectors | 1.1 |
| LeCun 1995 [Bottou$^+$ 1994] | ANN (LeNet4) | 1.1 |
| Simard$^+$, 1993 | Tangent Distance | 1.1 |
| LeCun 1995 [Vapnik 1998] | ANN (LeNet5) | 0.9 |
| Schölkopf$^+$, 1998 | Invariant Support Vectors | 0.8 |
| Drucker$^+$, 1993 | Boosted ANN | 0.7 |

## 2.2 The MNIST Database

The modified National Institute of Standards and Technology handwritten digits database (MNIST) is very similar to the USPS database. The main differences are that the images are not normalized and that the corpus is much larger. It contains 60,000 reference images and 10,000 test images. The data is given as 20×20 pixel sized images with with 256 grayscales. The MNIST database is available via the WWW at `http://www.research.att.com/~yann/ocr/mnist/`. Some examples from the NIST corpus are shown in Figure 2.2, which illustrate the effects of normalization if compared to Figure 2.1.

The MNIST task is generally considered to be easier than the USPS task for two reasons. On the one hand, the human error rate of this particular task is reported to be 0.2%, although it has not been determined for the whole test set [Simard$^+$ 1993]. On the other hand, the (almost ten times) larger training set allows machine learning algorithms to generalize better. Concerning the relationship between training set size and classification performance, it is said in [Smith$^+$ 1994] that increasing the training set size by a factor of ten cuts the error rate by half in many cases.

Table 2.3: Results reported on CID.

| Author | Method | Error Rate [%] |
|---|---|---|
| Blanz$^+$, 1996 | Support Vectors | 0.3 |
| Kressel, 1998 | Polynomial Classifier | 0.8 |

The same arguments as for the USPS data concerning the absence of a development test set and the availability of recognition results from other research groups also hold for the MNIST database. Some of the results that have been reported on MNIST can be found in Table 2.2.

## 2.3 The Chair Image Database

The Chair Image Database (CID), which can be downloaded via FTP at `ftp://ftp.kyb.tuebingen.mpg.de/pub/chair_dataset` consists of computer generated images of office chairs out of 25 classes. Example images taken from the CID database can be seen in Figure 2.3.

There are different training sets available, but in the experiments of this work only the largest one with 400 different 3D-views per class was used, summing up to a total of 10,000 training samples. The test set consists of 2,500 images, i.e. a hundred views per class, where each object is represented by a $16 \times 16$ pixels sized grayscale image. Feature vectors for each object are part of the database, each of them consisting of the original grayscale image and four orientation dependent gradient images. Thus, the resulting feature vectors are 1,280-dimensional [Blanz$^+$ 1996]. Some of the results that have been reported on CID can be found in Table 2.3.

Figure 2.3: Example images taken from the CID database.

## 2.4 The IRMA Task

In this section, a brief description of the medical IRMA database is given, as well as a short introduction into the RWTH Aachen - University of Technology IRMA project (Image Retrieval in Medical Applications) [Lehmann$^+$ 2000a].

### 2.4.1 The IRMA Database

The IRMA radiograph database consists of medical radiograph images taken from the RWTH Aachen - University of Technology IRMA project (cp. Section 2.4.2). The images come from daily routine and are secondary digital, that is they have been scanned from conventional film-based radiographs. All images were scanned using 256 grayscales, with the image sizes ranging from about $200 \times 200$ pixels (e.g. a radiograph of a single finger) to about $2,500 \times 2,500$ pixels (e.g. a chest radiograph). The anonymized images reflect the distribution of images in the Department of Diagnostic Radiology and were labelled into six classes by an expert. The corpus consists of 110 abdomen, 706 limbs, 103 breast, 110 skull, 410 chest and 178 spine radiographs, summing up to a total of 1,617 images. Furthermore, a smaller set of 332 images exists which is used for testing purposes. The original images are scaled down to a common height of 32 pixels for classification purposes (keeping the original aspect ratio). It should be noted that this rescaling-step does not produce a significant decrease in recognition rate, but speeds up the recognition system considerably [Dahmen$^+$ 2000a, Theiner 2000, Dahmen$^+$ 2001c].

Figure 2.4 shows prototypical example images for each of the six classes. Because the images were taken from daily clinical routine, the IRMA database is subject to considerable intra-class variance, which is demonstrated in Figure 2.5 for the class ‘chest’. Thus, despite the fact that the IRMA database is only a six-class problem, radiograph classification is a hard problem. Besides the considerable variation in radiograph quality and the aforementioned intra-class variance (caused by different doses of X-rays, varying orientations, images with and without pathologies, changing scribor position[1] etc.), there is a strong visual similarity between many images of the classes abdomen and spine respectively skull and spine, as can be seen in Figure 2.4.

Because there are only 1,617 images available, a leaving-one-out approach was adopted for cross validation, that is the database served as training and development test set, classifying each image while using the remaining 1616 as training set. After parameter adjustment the classifier was evaluated on the set of 332 additional radiographs. Thus, the final result does not suffer from ‘training on the testing data’.

One drawback of the IRMA database is the fact that so far only few comparison results exist, which are shown in Table 2.4. Nevertheless it was chosen in this work, as it shows the wide variety of possible applications of the proposed classifier, which not only produces state-of-the-art results on handwritten digits, but also on this completely different dataset of medical radiographs.

[1] The scribor is a data field containing patient information.

Figure 2.4: Example radiographs taken from the IRMA database. Top-left to bottom-right: abdomen, limbs, breast, skull, chest and spine.

Table 2.4: Results reported on the IRMA database.

| Author | Method | ER [%] |
|---|---|---|
| Theiner 2000 | Cooccurrence Matrices | 29.0 |
| Bredno+ 2000a | Active Contour Model | 51.1 |

To describe the context to which this classification task belongs, a short description of the IRMA system is given below [Lehmann+ 2000a]. The most important point for the motivation of radiograph classification is that in secondary digital image archives, the anatomic class labels are usually not existing. Even in primary digital DICOM archives, the anatomic region information is in many cases incorrect or missing [Kohnen+ 2001].

### 2.4.2 An Overview of the IRMA system

From the medical point of view there exist three major applications for automated content based image retrieval [Lehmann+ 2000a]:

(1) automatic retrieval of relevant images for follow-up studies within a picture archiving system,

(2) searching for representative images of known diseases and

(3) scientific and educational studies on X-ray patterns.

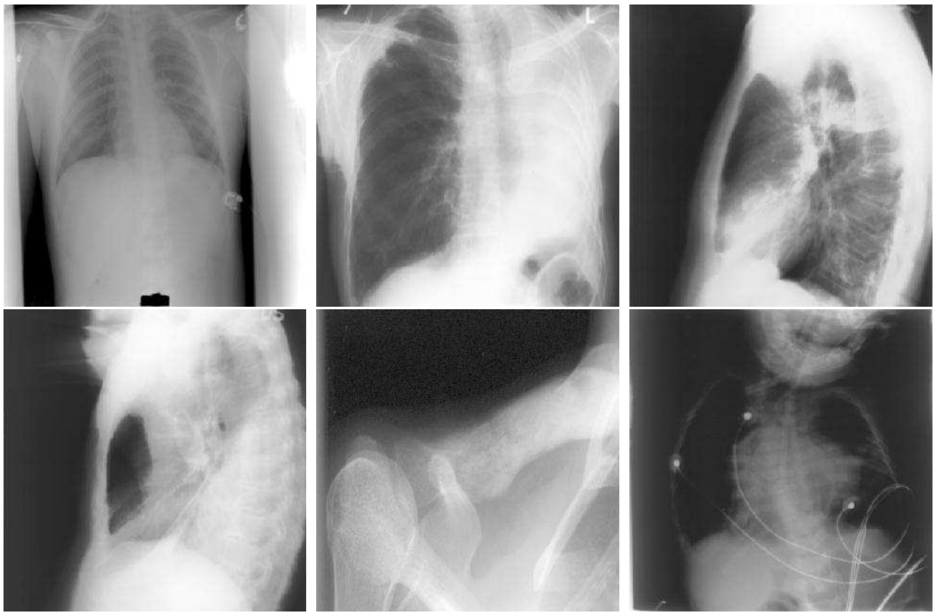

Figure 2.5: Variations within the class ‘chest’.

In contrast to common approaches to image retrieval, the IRMA concept is based on a strict logical and algorithmic separation of the following steps to enable complex image content understanding (cp. Figure 2.6):

- image-categorization (based on global features)
- image-registration (in geometry and contrast)
- feature extraction (based on local features)
- feature selection (category and query dependent)
- indexing (multiscale blob-representation)
- identification (incorporate a-priori knowledge)
- retrieval (on blob-level)

To enable complex queries for medical purpose, the information retrieval system must be familiar with the class of a given image prior to query processing, as this information is of great interest for the following IRMA steps. For example, searching a pulmonal tumor in a skull radiograph is senseless (as - by definition - a pulmonal tumor is always located in the lungs), and ultrasound images need different processing than radiographs (as the characteristics of an ultrasound image greatly differ from those of a radiograph). Thus, if a radiologist is searching the image database for all radiographs showing a pulmonal tumor, the IRMA system only processes radiographs which are classified as ‘chest’ (or have a posterior probability for ‘chest’ that is higher than a user-defined threshold). On all pictures fulfilling this constraints, the (probably computational more expensive) search for tumors is done, for instance by using local textural features as proposed in [Vogelsang$^+$ 1997] or statistical classifiers such as proposed in [Dahmen$^+$ 2000b]. The

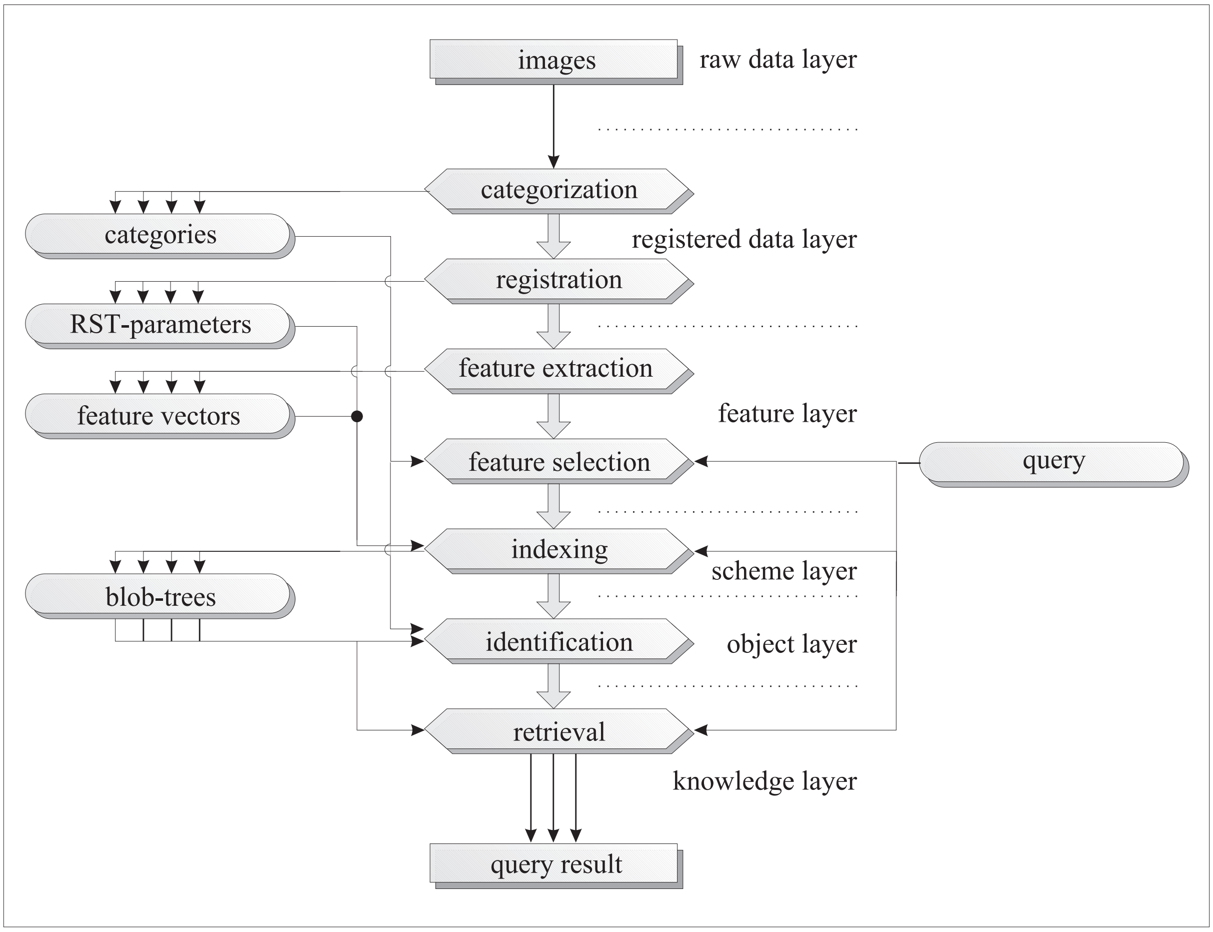


Figure 2.6: The IRMA architecture.

categorization step therefore not only reduces the computational complexity needed to answer an IRMA query, it will also most probably reduce the 'false-alarm'-rate of the system, improving its precision.

In the IRMA system, three major classes are defined: image modality (physical), anatomic region (anatomical) and image orientation (technical). In a first step, six anatomic regions are distinguished: (1) abdomen, (2) limbs, (3) breast, (4) skull, (5) chest and (6) spine. These instances build subclasses resulting in hierarchically structured IRMA-categories. While modern DICOM imaging devices provide information required for image classification (at least theoretically, as this information is often wrong or missing [Kohnen$^+$ 2001]), automatic content based classification is required for fast archiving of images acquired by film-based modalities such as radiographs. Once the class of a given image has been determined using global features, subsequent IRMA processing steps can use this information to extract problem specific features needed to answer complex queries. As classification is not necessarily unique (a chest radiograph might be labelled 'chest' and 'spine' at the same time), this step is called 'categorization' within the IRMA system. Thus, each image can be linked to several categories and the likelihood for each of these is also stored in the IRMA database. Therefore, classifiers used for categorization should be rather sensitive than specific.

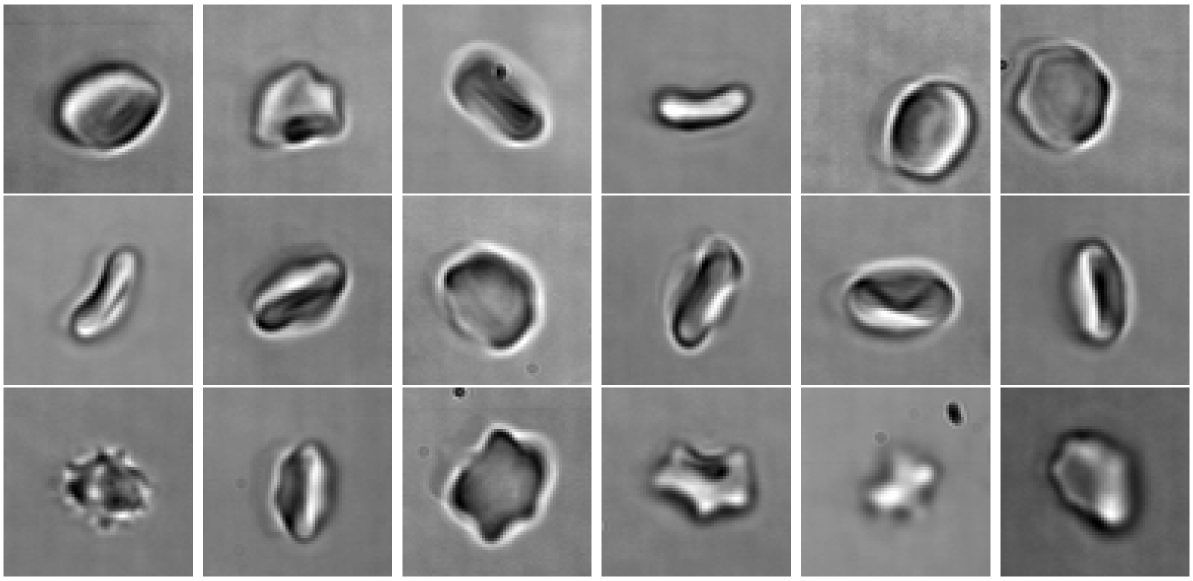

Figure 2.7: RBC example images, top to bottom: stomatocytes, discocytes, echinocytes.

After categorization, the image is registered to a prototype which has been previously defined by an expert or by a statistical data analysis. In the following feature extraction step it is distinguished between so called 'category-free' features (which are suitable for all categories, i.e. a gradient image) and 'category-specific' features, (i.e. segmentation of the ribs in a chest radiograph [Dahmen+ 1997, Vogelsang+ 1998] ). In the feature selection step, appropriate features for a given query are chosen. One possibility to do this is performing a linear discriminant analysis. In the indexing step, a compact representation of the given query image and the features extracted is created. Based on each set of feature images, the query image is segmented into relevant regions. Region representation (at multiple scales) will then be done via blobs. This hierarchical multiscale approach will allow the user to retrieve from entire images as well as from regions of interest. The blob-identification step might be useful for queries concerning details defined within organs or other objects in an image. In the final retrieval step, the query is processed via suitable distance measures defined on the entire image or on blob-level respectively.

## 2.5 The Red Blood Cell Task

In the red blood cells experiments, a database of 5,062 images that were expert labelled as *stomatocyte*, *echinocyte* or *discocyte* was used, where each cell is represented by a 128×128 pixels sized grayscale image (see Figure 2.7). The images were taken in a capillary where the RBC showed their native shapes without applied forces during sedimentation [Schönfeld+ 1989]. With only 5,062 images available - similar to the IRMA data - the dataset was not divided into a single training and test set. Instead, a cross-validation approach is applied in the experiments, that is the data are splitted into ten subsets. Each data set is then used for testing while the remaining nine sets are used for training, with the overall error rate being the mean over all subset error rates. Note that although all images are used as test and training images, the according training and test sets are strictly disjoint in all cases. A drawback of the RBC database is the lack of results obtained by competing classification methods. The only 'comparison result' reported is a human error rate of $\geq 20\%$ [Fischer 1999].

As for the motivation of this RBC task, it should be noted that in standard tests, drugs that induce shape changes to red blood cells are often used to examine whether the cell membrane still acts in a well known way. This is done by comparing induced shape changes with the known behaviour on drugs [Deuticke$^+$ 1990]. This comparison is usually performed by a human expert and therefore time and cost consuming, stressing the need for automatic classification.

## 2.6 The COIL-20 Database

The Columbia University Object Image Library (COIL-20, [Murase & Nayar 1995]) consists of images taken from 20 different 3D-objects, which are viewed from varying points of view (the database is ready for download via WWW at `http://www.cs.columbia.edu/CAVE/research/softlib/coil-20.html`). Each image contains a single object (which is subject to different lighting conditions) and is given in 256 grayscales. There are 1,440 reference images of size $128 \times 128$ pixels available (called *processed* data), as well as 360 test images of size $448 \times 416$ pixels (called *unprocessed* data). Although the test images do belong to only 5 classes, the problem is still treated as a 20-class problem in the experiments. Furthermore, to guarantee that training and test set are sufficiently different, only images with odd rotation angle are used as references and only images with even rotation angle as test scenes. Thus, a number of 720 reference images and 180 test images remains.

Concerning the state of the art it should be noted that only few authors report error rates for the whole data set. In fact, most authors use COIL-20 in a modified version, for instance to investigate on the behaviour of a recognition system in presence of inhomogeneous backgrounds and the like, because it is very easy to modify the COIL-20 data. Therefore, a direct comparison of different COIL-20 results is hard. Nevertheless it was chosen for some experiments on object localization and object recognition. Interesting publications using COIl-20 are for instance:

- In [Murase & Nayar 1995], the authors realize a real-time segmentation based recognition system for the COIL-20 data, reporting an error rate of 0% (using 720 unavailable test scenes which differ from the 360 mentioned above).

- In [Baker$^+$ 1996], the authors present a recognition system optimized for fast recognition of COIL-20 objects. The experiments were conducted using only the 1,440 reference images (processed image set), which were split into two disjoint subsets. For this particular setting, the authors report an error rate of 0%.

- In [Pösl$^+$ 1998], the authors use a small subset of the available COIL-20 images for experiments dealing with inhomogeneous backgrounds and localization of known objects. In the experiments conducted, either the class of the object was known


Figure 2.8: The 20 different objects of the COIL-20 references.

and the task is to detect it in the scene, or the position of the object is known and the according class label is to be determined. Furthermore, as only a subset of the images was used, no error rates for the complete COIL-20 data are given.

Example images taken from COIL-20 are given in Figure 2.8. Because of the fact that there are only 72 reference images available per class, a nearest neighbour based classifier was used in the COIL-20 experiments throughout this work. Furthermore, as all of the reference images of each class have a different orientation, applying a mixture approach appears to be meaningless (as there is only one view per rotation angle available).
Furthermore, illumination invariance proved to be an important point in the COIL-20 experiments. Therefore, the resulting feature vectors $x$ were normalized to have unit length, i.e.

$$\hat{x} = \frac{1}{\|x\|} \cdot x. \tag{2.1}$$

Note that this normalization step, too, justifies the use of a nearest neighbour based classifier on that particular data set. Transforming all vectors to have unit length, the resulting feature vectors lie on the surface of a hypersphere in feature space, contradicting the assumption of a Gaussian distribution.

# Chapter 3

# Scientific Goals

The aim of this work is to build up a framework for efficient image object recognition. In particular, the following goals can be formulated:

I. *Invariant Statistical Classifier:*
The use of Gaussian mixture densities proved to be very efficient in the field of speech recognition and is widely accepted to be the state-of-the-art approach. Thus, in order to investigate how well such a classifier performs in image object recognition, the classifier to be developed is based on Gaussian mixtures, too. Furthermore, taking into consideration the special properties of image data (which differ considerably from the properties of audio signals), the system shall be invariant with respect to transformations such as rotation, scale and translation.

II. *State-of-the-Art Results:*
The main goal of this work is to realize a classifier which obtains state-of-the-art results. That is, recognition accuracy is the most important point, neglecting questions such as the computational complexity of the classifier. For a meaningful evaluation, the methods presented are applied to well known image object recognition tasks, where most of the experiments are performed on the US Postal Service database. On the one hand, this is done because it is generally regarded to be hard task. On the other hand, there exist lots of results reported by international research groups, which allow for a comparison of the different training and classification approaches used (cp. Chapter 2.1).

III. *Segmentation-free Approach:*
Following the experiences gained in speech recognition, the system to be developed shall not rely on object segmentation. This especially holds for the IRMA and the COIL-20 databases. Throughout this work, a holistic appearance based approach is chosen. Thus, in order to present all information contained in an image to the system, each pixel (respectively its grayvalue) of the input images is used as a feature. Optionally, a statistical data analysis can be performed to reduce the feature space, for instance by using a linear discriminant analysis.

IV. *Multiplying the Test Data:*
Investigations are performed concerning how the approach of creating virtual data can be extended to the test data. In this work, a probabilistic justification of the use of virtual testing data is given, yielding the novel virtual test sample method (VTS), which has some desirable advantages over conventional classifier combination schemes, such as discussed in [Kittler$^+$ 1998].

V. *Extension of Tangent Distance:*
Throughout this work, the use of tangent distance within the statistical classifier proved to be very efficient. Yet, tangent distance has the drawback that it only considers global (e.g. affine) image transformations. Thus, investigations are performed on how to incorporate local image transformations (local perturbations) into tangent distance.

VI. *Probabilistic Framework:*
The investigations performed throughout this work are to be embedded within a statistical framework. This especially holds for the novel probabilistic interpretation of tangent distance as presented in Chapter 7 and for the justification of the virtual test sample method.

# Chapter 4

# The Baseline System

In this section the statistical baseline classifier is described, which is based on the use of Gaussian mixture densities in the context of the Bayesian decision rule. In the course of this work, this baseline system will be extended with regard to the incorporation of invariances. Furthermore, a new approach to combined classification will be derived, called the virtual test sample method.

## 4.1 Gaussian Mixtures in Bayesian Context

To classify an observation $x \in \mathbb{R}^d$ the Bayesian decision rule (cp. Chapter 1.1.1) is applied here [Duda & Hart 1973, pp. 10-39]

$$x \longmapsto r(x) = \underset{k}{\operatorname{argmax}} \{p(k) \cdot p(x|k)\} \tag{4.1}$$

where $p(k)$ is the *a priori* (or *prior*) probability of class $k$, $p(x|k)$ is the *class conditional* probability for the observation $x$ given class $k$ and $r(x) \in \{1, ..., K\}$ is the classifier's decision. As neither $p(k)$ nor $p(x|k)$ are known, models have to be chosen for the respective distributions and their parameters have to be estimated by using the training data. In the handwritten digits experiments, the prior probabilities are set to

$$p(k) = \frac{1}{K}, \ k = 1, ..., K \tag{4.2}$$

as it is not obvious why a certain digit should have a higher prior probability than another without any context information. Otherwise, relative frequencies are used, i.e.

$$p(k) = \frac{N_k}{N}, \ k = 1, ..., K \tag{4.3}$$

where $N_k$ is the number of reference images of class $k$ and $N$ is the total number of images available.

The class conditional densities $p(x|k)$ are modeled by using Gaussian mixture densities or kernel densities respectively. A Gaussian mixture is defined as a linear combination of Gaussian component densities $\mathcal{N}(x|\mu_{ki}, \Sigma_{ki})$, leading to the following expression for the class conditional probabilities:

$$\begin{aligned} p(x|k) &= \sum_{i=1}^{I_k} c_{ki} \cdot \mathcal{N}(x|\mu_{ki}, \Sigma_{ki}) && (4.4) \\ \mathcal{N}(x|\mu_{ki}, \Sigma_{ki}) &= \frac{1}{\sqrt{\det(2\pi\Sigma_{ki})}} \exp\left[-\frac{1}{2}(x-\mu_{ki})^T \Sigma_{ki}^{-1}(x-\mu_{ki})\right] && (4.5) \end{aligned}$$

where $I_k$ is the number of component densities used to model class $k$, $c_{ki}$ are weight coefficients (with $c_{ki} > 0$ and $\sum_{i=1}^{I_k} c_{ki} = 1$, which is necessary to ensure that $p(x|k)$ is a probability density function), $\mu_{ki}$ is the mean vector and $\Sigma_{ki}$ is the covariance matrix of component density $i$ of class $k$.

To avoid the problems of estimating a covariance matrix in a high-dimensional feature space, i.e. to keep the number of parameters to be estimated as small as possible, pooled covariance matrices are used in the experiments:

- *class specific variance pooling* :
  In this case, only a single covariance matrix $\Sigma_k$ is estimated for each class $k$:

$$\Sigma_{ki} = \Sigma_k = \sum_{i=1}^{I_k} \frac{N_{ki}}{N_k} \cdot \Sigma_{ki} \tag{4.6}$$

- *global variance pooling* :
  Here, only a single covariance matrix $\Sigma$ is estimated for all densities considered:

$$\Sigma_{ki} = \Sigma = \sum_{k=1}^{K} \frac{N_k}{N} \cdot \Sigma_k \tag{4.7}$$

  where $\Sigma_k$ is the class specific covariance matrix of class $k$ as defined in Equation (4.6).

Furthermore, in most experiments (if nothing else is said), a diagonal covariance matrix is used, i.e. a variance vector (some experiments on USPS using a full covariance matrix are given in Appendix C.1). Note that this does not mean a loss of information, as a mixture density of that special form can still approximate any given density function with arbitrary precision [Wilson 2000]. Maximum-likelihood parameter estimation is now done using the Expectation-Maximization (EM) algorithm [Dempster$^+$ 1977] combined with a Linde-Buzo-Gray based clustering procedure [Linde$^+$ 1980].

## 4.2 Parameter Estimation

This section deals with estimating the mixture density parameters, i.e. $\{c_{ki}, \mu_{ki}, \Sigma_{ki}\}$. To do so, the EM-algorithm, a maximum likelihood parameter estimation approach for data with so-called hidden variables, is used. Here, the unknown membership relation between observations $x_n$ and mixture components $\mathcal{N}(x_n|\mu_{ki}, \Sigma_{ki})$ is the hidden variable.

### 4.2.1 The Expectation-Maximization Algorithm

The Expectation-Maximization algorithm is an iterative approach to estimating the parameters of some unknown probability density function with a hidden variable $i$. Its application to mixture densities is described in [Dempster+ 1977], where the index of some density which an observation belongs to is interpreted as hidden variable. This assignment is expressed as a probability $p(i|x_n, k, \lambda)$, with $k$ being the class index, $i$ the density index and $\lambda = \{c_{ki}, \mu_{ki}, \Sigma_{ki}\}$. Applying the EM-algorithm to the problem at hand, one obtains the following reestimation formulae:

$$p(i|x_n, k, \lambda) = \frac{c_{ki} \cdot \mathcal{N}(x_n|\mu_{ki}, \Sigma_{ki})}{\sum_{i'} c_{ki'} \cdot \mathcal{N}(x_n|\mu_{ki'}, \Sigma_{ki'})} \tag{4.8}$$

$$\gamma_{ki}(n) = \frac{p(i|x_n, k, \lambda)}{\sum_{n'} p(i|x_{n'}, k, \lambda)} \tag{4.9}$$

$$\hat{c}_{ki} = \frac{1}{N_k} \sum_{n=1}^{N_k} p(i|x_n, k, \lambda) \tag{4.10}$$

$$\hat{\mu}_{ki} = \sum_{n=1}^{N_k} \gamma_{ki}(n) \cdot x_n \tag{4.11}$$

$$\hat{\Sigma}_{ki} = \sum_{n=1}^{N_k} \gamma_{ki}(n) \cdot [x_n - \hat{\mu}_{ki}][x_n - \hat{\mu}_{ki}]^T \tag{4.12}$$

The iteration is started by estimating the parameters $c_{ki}$, $\mu_{ki}$ and $\Sigma_{ki}$, yielding the initial $p(i|x_n, k, \lambda)$. Using this $p(i|x_n, k, \lambda)$, the parameters $\lambda$ can be re-estimated by setting $c_{ki} := \hat{c}_{ki}$, $\mu_{ki} := \hat{\mu}_{ki}$ and $\Sigma_{ki} := \hat{\Sigma}_{ki}$, yielding a better estimation for $p(i|x_n, k, \lambda)$. This procedure repeats until the parameters converge or until a certain number of iteration has been performed.

In the experiments, the number of densities to be trained per mixture as well as their initial parameters are defined by repeatedly splitting mixture components, that is a Linde-Buzo-Gray [Linde+ 1980] inspired method is used.

To overcome the problem of choosing the initial parameter values, the iteration is started by estimating a single density for each class $k$, which is easily possible. A mixture density is then created by splitting single densities, i.e. a mixture component $ki$ is splitted by modifying the mean vector $\mu_{ki}$ using a suitable distortion vector $\epsilon$. In the experiments, fast convergence was obtained by choosing $\epsilon$ to be a fraction of the respective variance vector. This method proved to be very efficient for modeling emission probabilities in speech recognition [Ney 1990]. Thus, one obtains two new mean vectors $\mu_{ki}^{+} = \mu_{ki} + \epsilon$ and $\mu_{ki}^{-} = \mu_{ki} - \epsilon$, that is a mixture density with mixture components $\mathcal{N}(x|\mu_{ki}^{+}, \Sigma_{ki})$ and $\mathcal{N}(x|\mu_{ki}^{-}, \Sigma_{ki})$. The mixture density parameters can now be re-estimated using Equations (4.8)-(4.12). This splitting procedure repeats until the desired number of densities is reached. In the experiments, in order to get reliable estimations for the unknown parameters, a density $i$ belonging to class $k$ is only splitted, if

$$\sum_{n=1}^{N_k} p(i|x_n, k, \lambda) \geq N_{Split} \tag{4.13}$$

holds, using $N_{Split} = 4.0$ here.

Note that choosing the number of mixture components is a problem, as the (log-) likelihood of the model keeps improving with the number of densities increasing (thus, there exists no clear maximum), which sometimes leads to overfitting effects, i.e. bad generalization properties. In the experiments conducted, the number of densities per mixture is chosen with respect to the obtained error rate on a development test set (if the latter is advisable). The optimal parameters determined on such a set are then used for the recognition of unseen data. In literature, there are a many approaches to choose the number of densities, among them minimum description length [Rissanen 1978] or Bayesian approaches [Roberts$^{+}$ 1998].

### 4.2.2 Maximum Approximation

One can also define a maximum approximation of the Expectation-Maximization algorithm by defining:

$$p(i|x_n, k, \lambda) = \begin{cases} 1 & : \text{ 'best' single density } i \\ 0 & : \text{ otherwise} \end{cases} \tag{4.14}$$

In this case, each training vector is assigned to only one density, namely the one yielding the best explanation. Because of that, instead of calculating any $p(i|x_n, k, \lambda)$ only the maximum $\mathcal{N}(x_n|\mu_{ki}, \Sigma_{ki})$ remains to be found (thus, the sum in Equation (4.4) is replaced by the maximum operation). This approximation is justified by the exponential decay of a Gaussian probability function, which usually leads to one dominating $p(i|x_n, k, \lambda)$. Therefore, the computational complexity of the training- and recognition-step can be reduced without a significant deterioration of recognition accuracy in many cases.

Nevertheless, the motivation of the maximum approximation is somewhat historical. On todays computers, there is no need to perform such an approximation. Therefore, it is not used in the experiments conducted throughout this work.

### 4.2.3 Discriminative Training

A drawback of the conventional maximum likelihood training of the mixture density parameters is the fact, that each class is handled separately in training. In opposite to this, discriminative training procedures, such as the *maximum mutual information* criterion (MMI) presented here, optimize the *a posteriori* probabilities of the training samples and hence the class separability.

Given labelled training data $(x_n, k_n), n = 1, .., N$, with $x_n$ being a feature vector of class $k_n$ the MMI criterion is defined as

$$F_{MMI}(\lambda) = \sum_{n=1}^{N} \log \frac{p(k_n) \cdot p(x_n|k_n, \lambda)}{\sum_{k=1}^{K} p(k) \cdot p(x_n|k, \lambda)} \qquad (4.15)$$

where the prior probabilities $p(k)$ are assumed to be given. A maximization of the MMI criterion defined above therefore tries to simultaneously maximize the class conditional probabilities of the given training samples and to minimize a weighted sum over the class conditional probabilities of all competing classes. Thus, the MMI criterion optimizes the class separability.

In the following, discriminative reestimation formulae for the mixture density parameters $\lambda$ will be presented, using global variance pooling. Furthermore, a maximum approximation is used, that is sums of probabilities are approximated by the maximum addend. Performing *extended Baum-Welch* parameter optimization on the MMI criterion yields the following reestimation formulae for the means $\mu_{ki}$, global diagonal variances $\sigma^2$ and mixture weights $c_{ki}$ of Gaussian mixture densities (for further details on that topic, the reader is referred to [Schlüter & Macherey 1998]). Note that for ease of representation, the dimension index $d = 1, ..., D$ is skipped in the following formulae.

$$\hat{\mu}_{ki} = \frac{\Gamma_{ki}(x) + Dc_{ki}\mu_{ki}}{\Gamma_{ki}(1) + Dc_{ki}} \qquad (4.16)$$

$$\hat{\sigma}^2 = \frac{\sum_k D(\sigma^2 + \sum_i c_{ki}\mu_{ki}^2)}{KD} - \sum_{ki} \frac{\Gamma_{ki}(1) + Dc_{ki}}{KD} \hat{\mu}_{ki}^2 \qquad (4.17)$$

$$\hat{c}_{ki} = \frac{\Gamma_{ki}(1) + Dc_{ki}}{\Gamma_k(1) + D} \qquad (4.18)$$

with iteration constant $D$. $\Gamma_{ki}(g(x))$ and $\Gamma_k(g(x))$ are discriminative averages of functions $g(x)$ of the training observations, defined by

$$\Gamma_{ki}(g(x)) = \sum_n \delta_{i,i_{k,n}} \left[\delta_{k,k_n} - p(k|x_n,\lambda)\right] g(x_n) \tag{4.19}$$

$$\Gamma_k(g(x)) = \sum_i \Gamma_{ki}(g(x)) \tag{4.20}$$

$\delta_{i,j}$ is the *Kronecker delta*, i.e. given a training observation $x_n$ of class $k_n$, $\delta_{i,i_{k,n}} = 1$ only if $i$ is the 'best-fitting' component density $i_{k,n}$ given class $k$ and $\delta_{k,k_n} = 1$ only if $k = k_n$. For fast but reliable convergence of the MMI criterion, the choice of the iteration constant $D$ is crucial. Although there exists a proof of convergence [Baum+ 1967], the size of the iteration constant guaranteeing convergence yields impractical small step-sizes, i.e. very slow convergence. In practice, fastest convergence is obtained if the iteration constants are chosen such that the denominators in Equations (4.16) - (4.18) and the according variances are kept positive:

$$D = h \cdot \max_{k,i} \left\{ D_{min}, \frac{1}{c_{ki}} \left( \frac{1}{\beta_k} - \Gamma_{ki}(1) \right) \right\} \tag{4.21}$$

$$\begin{aligned} D_{min} = \max_d & \frac{-\Gamma(x^2) + \alpha\Gamma(1) + \sum_{k,i} [2\Gamma_{ki}(x) - \Gamma_{ki}(1)\mu_{ki}]\mu_{ki}}{K(\sigma^2 - \alpha)} \\ & + \frac{\sum_{ki} \beta_k (\Gamma_{ki}(x) - \Gamma_{ki}(1)\mu_{ki})^2}{K(\sigma^2 - \alpha)} \end{aligned} \tag{4.22}$$

Here, $D_{min}$ denotes an estimation for the minimal iteration constant guaranteeing the positivity of the variances and the *iteration factor* $h > 1$ controls the convergence of the iteration process, high values leading to low step sizes. The constants $\beta_k > 0$ are chosen to prevent overflow caused by low-valued denominators. In the experiments, parameter initialization is done using ML training, $\alpha$ denotes the minimum variance allowed and $\beta_k$ is chosen to be

$$\frac{1}{\beta_k} = \max_i(|\Gamma_{ki}(1)|) + 1. \tag{4.23}$$

## 4.3 Kernel Densities

In the case of kernel densities (also called Parzen windows or Parzen densities sometimes) [Devroye+ 1996, pp. 147-153], each training sample $x_n$ of class $k_n$ defines a Gaussian single density $\mathcal{N}(x|x_n, \Sigma_{k_n})$ with an estimated class-specific covariance matrix $\Sigma_{k_n}$. Thus, each training sample itself is interpreted as the center of a Gaussian. Therefore, kernel densities can be interpreted as the extreme case of a Gaussian mixture density, where the class conditional probabilities are modeled via

$$p_{KD}(x|k) = \frac{1}{N_k} \sum_{n=1}^{N_k} \mathcal{N}(x|x_n, \Sigma_{k_n}). \tag{4.24}$$

In the experiments, the procedure described above is used to train a mixture density for each class $k$ with class-specific covariance matrix $\Sigma_k$. Then, for all training samples $x_n$ belonging to class $k$

$$\Sigma_{k_n} = \alpha \cdot \Sigma_k \tag{4.25}$$

is defined with some factor $\alpha > 0$ (of course, a globally pooled covariance matrix can also be used). Note that a maximum-likelihood estimation of $\Sigma_{k_n}$ fails in the case of kernel densities, as the optimum is obtained for the respective variances approaching zero [Ney 1999]. In some experiments, the test error rate is investigated as a function of the $\alpha$. Also, a comparison between kernel densities and a nearest neighbour classifier is performed, as one would expect the kernel density error rate to converge to that of nearest neighbour with $\alpha \to 0$. Furthermore it should be noted that a 1-nearest neighbour classifier can be regarded as a maximum approximation to kernel densities.

## 4.4 Invariance Properties of the Baseline System

The statistical baseline approach presented above is not invariant with respect to image transformations so far. In fact, it assumes that all relevant image transformations are present in the reference images. If that assumption holds, there is no real need to incorporate any further invariances into the classifier, because the transformed images have a significant contribution to the estimation of the mixture density parameters in the training phase. Yet, in many practical applications, the amount of the available training data is strictly limited for a number of reasons.

On the one hand, acquiring reference images might be difficult or expensive. This is for instance the case in medical imaging, where a large number of probably expensive tests would have to be performed to gather a large number of data. Another example are industrial manufacturing processes, where the gathering of large amounts of data might be infeasible, because this might disturb the production process itself (implying high costs). On the other hand, the possible variations the images might be subject to can be that high, that trying to capture all relevant variations of the objects is infeasible. Furthermore, a vast amount of training data has a strong influence on the computational complexity not only of the training, but also of the recognition phase (as a large amount of training data usually leads to an increased number of mixture components, thus increasing the time required to classify an observation). Therefore, an explicit incorporation of invariances into a classification system is desirable. In the next chapters, two methods are described to do so. Chapter 5 deals with invariant distance measures, whereas Chapter 6 will discuss possibilities to enrich the available image data.

# Chapter 5

# Invariant Distance Measures

In the experiments concerning the incorporation of invariances into the statistical classifier, using of invariant distance measures proved to be the best choice [Dahmen+ 2000d, Perrey 2000, Keysers 2000a]. In this chapter, such invariant distance measures are dealt with. To compensate for global image transformations Simard's tangent distance is used, whereas a simple *image distortion model* is proposed to compensate for local image transformations. Both approaches are successfully coupled to form a distance measure called *distorted tangent distance*. This novel distance measure considerably improves the efficiency of the original tangent distance approach, especially on the IRMA dataset (cp. Chapter 9).

## 5.1 Tangent Distance

In 1993, SIMARD et al. proposed an invariant distance measure called *tangent distance*, which proved to be especially effective for optical character recognition. The authors pointed out that reasonably small transformations of certain objects (like characters or, as also investigated in this work, radiographs) do not affect class membership. Simple distance measures like the Euclidean distance or the dot product between two vectors do not account for this, instead they are very sensitive to transformations like scaling, translation, rotation or axis deformations.

When an image $x$ of size $I \times J$ is transformed (e.g. scaled and rotated) with a transformation $t(x, \alpha)$ which depends on $L$ parameters $\alpha \in \mathbb{R}^L$ (e.g. the scaling factor and the rotation angle), the set of all transformed patterns

$$M_x = \{t(x, \alpha) : \alpha \in \mathbb{R}^L\} \subset \mathbb{R}^{I \times J} \tag{5.1}$$

is a manifold of at most $L$ dimensions. The distance between two images can now be defined as the minimum distance between their according manifolds, being truly invariant with respect to the $L$ transformations under consideration.

Unfortunately, computation of this distance is a hard optimization problem and the manifolds needed have no analytic expression in general. Therefore, small transformations of an image $x$ can be approximated by a tangent subspace $\hat{M}_x$ to the manifold $M_x$ at the

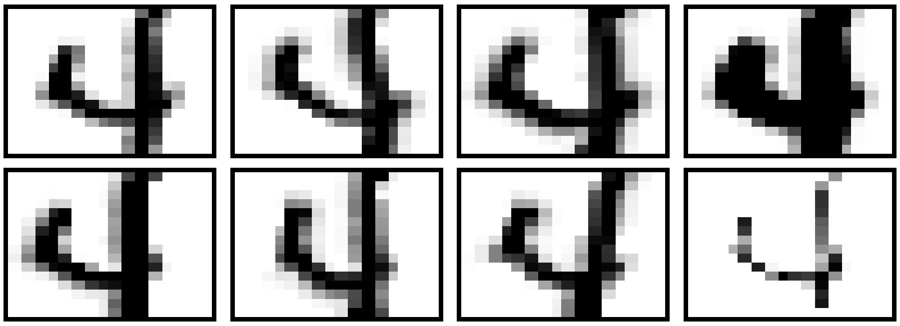

Figure 5.1: Example images generated via tangent approximation, using affine and line thickness transformations. Original image is at top-left.

point $x$. Those transformations (called *tangent approximation* in the following) can be obtained by adding to $x$ a linear combination of the vectors $x_l$ that span the tangent subspace. Thus, the manifold $M_x$ can be first-order approximated by:

$$\hat{M}_x = \{x + \sum_{l=1}^{L} \alpha_l \cdot x_l \ : \alpha \in \mathrm{I\!R}^L\} \subset \mathrm{I\!R}^{I \times J} \tag{5.2}$$

where $\alpha$ again denotes the vector-notation of the $\alpha_l$. Now, the single-sided tangent distance $D_{ST}(x, \mu)$ is defined as

$$D_{ST}(x, \mu) = \min_{\alpha}\{\|x + \sum_{l=1}^{L} \alpha_l \cdot x_l - \mu\|^2\} \tag{5.3}$$

The so-called tangent vectors $x_l$ can be shown to be the derivations of the transformations with respect to the respective transformation parameter and can be easily computed using finite differences between the original image $x$ and a reasonably small transformation of $x$ [Simard$^+$ 1993]. Example images that were computed using (5.2) are shown in Figure 5.1. In this example, handwritten digits were chosen, as they are especially suited to demonstrate the effects of the tangent approximation.

One can also define a double-sided tangent distance

$$D_{DT}(x, \mu) = \min_{\alpha,\beta}\{\|x + \sum_{l=1}^{L} \alpha_l \cdot x_l - \mu - \sum_{l'=1}^{L} \beta_l \cdot \mu_{l'}\|^2\}, \tag{5.4}$$

yet this dramatically increases the computational complexity without yielding a significant improvement in recognition accuracy in most cases [Simard$^+$ 1993]. A schematic visualization of the distances discussed here is shown in Figure 5.2

In the experiments, the tangent vectors for translations (2), rotation, scaling, axis/ diagonal deformations (2) and line thickness were computed as proposed by SIMARD. In the IRMA experiments, the line thickness tangent is not meaningful and therefore replaced by a 'brightness' tangent (for further information, see Chapter 5.1.2).

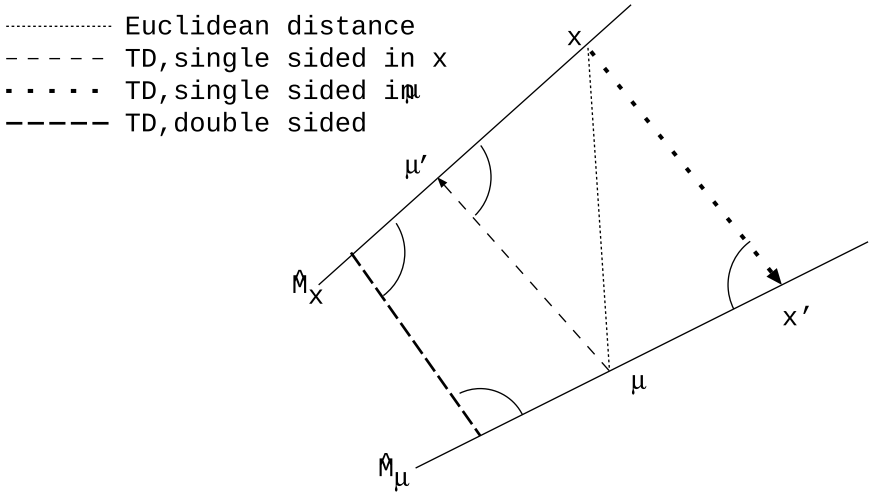


Figure 5.2: Schematic illustration of single-/ double-sided tangent distance.

Conceptionally, the single sided tangent distance is computed as follows:

1) Compute tangent vectors for the observation $x$.

2) Compute an orthonormal basis for the tangent subspace (using a singular value decomposition [Press+ 1992, pp. 59-67]).

3) Compute the projection $\mu'$ of a reference image $\mu$ into the tangent subspace of the observation $x$.

4) Compute $D_T(x, \mu) = D(\mu', \mu)$.

Note that steps 1) and 2) can be computed in advance, if the tangent vectors are applied on the side of the references (this will be done throughout this work, if nothing else is said. Some experiments on that topic are given in Appendix C.2), only steps 3) and 4) have to be performed while classifying. Given that the tangent vectors are orthogonal, this can be done efficiently in a single step by computing

$$D_T(x, \mu) = \|x - \mu\|^2 - \sum_{l=1}^{L} \frac{[(x-\mu)^t \cdot x_l]^2}{\|x_l\|^2} \tag{5.5}$$

### 5.1.1 Computing the Tangent Vectors

In his original work, Simard computed seven tangent vectors: six for affine transformations and a seventh for line thickness. These transformations proved to be especially effective for the recognition of handwritten digits, yet tangent distance can be applied to any transformation with known derivation.

Considering the group of affine transformations

$$\begin{pmatrix} i' \\ j' \end{pmatrix} = \begin{pmatrix} 1+\alpha_1 & \alpha_2 \\ \alpha_3 & 1+\alpha_4 \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix} + \begin{pmatrix} \alpha_5 \\ \alpha_6 \end{pmatrix} \tag{5.6}$$

the six corresponding tangent vectors can be computed as the derivations $x_1, ..., x_6$ with respect to the according parameters $\alpha_1, ..., \alpha_6$. Contrary to this, the tangent vector for line thickness is of heuristic nature and defined to be the squared image gradient. Following [Vapnik 1998, pp. 506-510] one obtains (here, for ease of notation, $x(i,j)$ denotes the pixel at position $(i,j)$ within an image $x$):

Horizontal translation:
$\alpha_l = 0,\ l = 1,2,3,4,6 \qquad i' = i + \alpha_5 \qquad j' = j$

$$x_1(i,j) = \lim_{\alpha_5 \to 0} \frac{x(i+\alpha_5, j) - x(i,j)}{\alpha_5} = \frac{\partial x(i,j)}{\partial i} \tag{5.7}$$

Vertical translation:
$\alpha_l = 0,\ l = 1,...,5 \qquad i' = i \qquad j' = j + \alpha_6$

$$x_2(i,j) = \lim_{\alpha_6 \to 0} \frac{x(i, j+\alpha_6) - x(i,j)}{\alpha_6} = \frac{\partial x(i,j)}{\partial j} \tag{5.8}$$

Rotation:
$\alpha_l = 0,\ l = 1,4,5,6 \qquad \alpha_2 = -\alpha_3 \qquad i' = i + \alpha_2 j \qquad j' = j - \alpha_2 i$

$$\begin{aligned} x_3(i,j) &= \lim_{\alpha_2 \to 0} \frac{x(i+\alpha_2 j, j-\alpha_2 i) - x(i,j)}{\alpha_2} \qquad (5.9) \\ &= \lim_{\alpha_2 \to 0} \frac{x(i+\alpha_2 j, j-\alpha_2 i) - x(i, j-\alpha_2 i)}{\alpha_2} + \lim_{\alpha_2 \to 0} \frac{x(i, j-\alpha_2 i) - x(i,j)}{\alpha_2} \\ &= j x_1(i,j) - i x_2(i,j) \end{aligned}$$

Scaling:
$\alpha_l = 0,\ l = 2,3,5,6 \qquad \alpha_1 = \alpha_4 \qquad i' = i + \alpha_1 i \qquad j' = j + \alpha_1 j$

$$x_4(i,j) = i x_1(i,j) + j x_2(i,j) \tag{5.10}$$

Axis deformation:
$\alpha_l = 0,\ l = 1,4,5,6 \qquad \alpha_2 = \alpha_3 \qquad i' = i + \alpha_3 j \qquad j' = j + \alpha_3 i$

$$x_5(i,j) = j\, x_1(i,j) + i\, x_2(i,j) \tag{5.11}$$

Diagonal deformation:
$\alpha_l = 0,\ l = 2,3,5,6 \qquad \alpha_1 = -\alpha_4 \qquad i' = i + \alpha_4 i \qquad j' = j - \alpha_4 j$

$$x_6(i,j) = i\, x_1(i,j) - j\, x_2(i,j) \tag{5.12}$$

Additionally, Simard also suggested the use of a seventh tangent vector, which is responsible for the line thickness deformation. Using the squared gradients of the directional shifts, it is intuitively evident that the resulting transformation affects line thickness (cp.

Figure 5.5). A similar effect could also be obtained by simply using absolute values.

Line thickness deformation:

$$x_7(i,j) \quad = \quad (x_1(i,j))^2 + (x_2(i,j))^2 \tag{5.13}$$

Note that the above equations do not exactly describe the respective transformations, in fact they are only local approximations. Furthermore, all transformations considered can be expressed as combinations of the tangent vectors for horizontal and vertical translations. Thus, to compute the required tangent vectors in a practical application, computing these particular tangent vectors is crucial. In the literature, many approaches are known to compute the vertical and horizontal gradients of a discrete image. One of the most prominent among them is the Sobel operator, which also takes into account diagonal gradients. Its four directional variants are shown in Figure 5.3, with the combined Sobel operator being defined as [Lehmann⁺ 1997, p. 213]:

$$S^* = \max\{|S_i|, |S_j|, |S_/|, |S_\backslash|\} \tag{5.14}$$

In the experiments conducted throughout this work, the modified Sobel operator as shown in Figure 5.4 produced slightly better results than the original operator [Keysers 2000a]. The figure depicts the template used for horizontal shifts, with the template used for vertical shifts being a 90° rotated version. Figure 5.5 shows the resulting tangent vectors for three images taken from the US Postal Service database.

### 5.1.2 Illumination Invariance

In the experiments, the tangent vectors for translations (2), rotation, scaling, axis/ diagonal deformations (2) and line thickness were computed as proposed by SIMARD. In the IRMA experiments, the line thickness tangent is not meaningful, so it was replaced by the following 'brightness' model, where an image $x$ is regarded to be subject to multiplicative as well as additive illumination changes, that is

$$t(x(i,j), \gamma_1, \gamma_2) = \gamma_1 \cdot x(i,j) + \gamma_2 \tag{5.15}$$

To compute the according tangent vectors, Equation (5.15) has to be derived with respect to the parameters $\gamma_1$ and $\gamma_2$. Derivation with respect to $\gamma_1$ one obtains the image itself

$$S_i = \frac{1}{4}\boxed{\begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix}} \qquad S_j = \frac{1}{4}\boxed{\begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}}$$

$$S_/ = \frac{1}{4}\boxed{\begin{matrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{matrix}} \qquad S_\backslash = \frac{1}{4}\boxed{\begin{matrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{matrix}}$$

Figure 5.3: The four directional variants of the Sobel operator.

| | | | | |
|---|---|---|---|---|
| | -0.15 | 0 | 0.15 | |
| -0.08 | -0.62 | 0 | 0.62 | 0.08 |
| | -0.15 | 0 | 0.15 | |

Figure 5.4: Template used for horizontal shift tangent calculation.

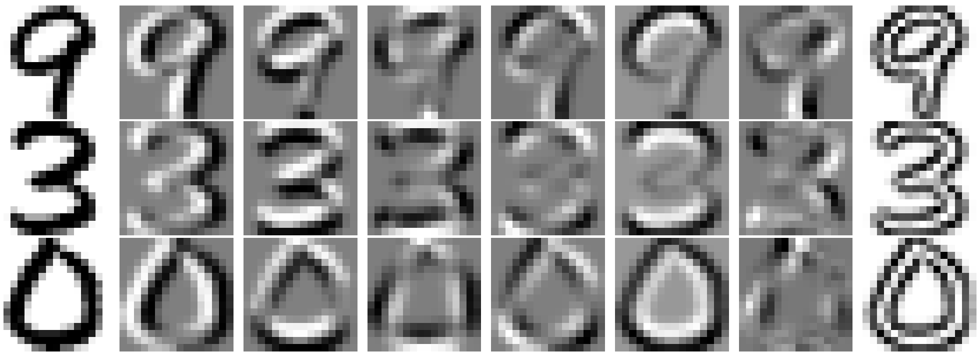

Figure 5.5: Tangent vectors for three USPS images. Left to right: original image, horizontal translation, vertical translation, diagonal deformation, axis deformation, scaling, rotation, line thickness.

as tangent vector, whereas derivation with respect to $\gamma_2$ yields a constant tangent vector.

Two things should be noted on these tangent vectors. On the one hand, the resulting tangent vectors yield an exact description of the respective manifold in this case, because the manifold resulting from applying the transformation given in Equation (5.15) is linear. On the other hand, using this illumination model in double sided tangent distance is senseless, as the null vector is always an element of the respective tangent subspace. Thus, the distance between any two images is zero using the multiplicative brightness model within double sided tangent distance. Because of this, only the additive lightning model (resulting in a tangent vector consisting of constant values) is used in the IRMA experiments.

As discussed above, tangent distance is a very effective means to compensate for small global transformations of an image. In the following, a simple, yet very effective image distortion model for local image variations are presented. The experiments conducted throughout this work show that both approaches work very well, but that the best results are obtained by combining both to *distorted tangent distance*.

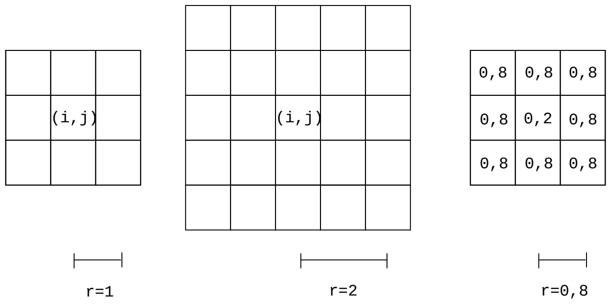


Figure 5.6: Examples for integer and non-integer IDM region sizes.

## 5.2 The Image Distortion Model

The last conceptional step in the computation of the tangent distance still requires the calculation of the (squared) Euclidean distance between an image $\mu$ and its projection $\mu'$ into the tangent subspace of a second image $x$. Although small global transformations have been compensated for by the projection step, this distance is still highly sensitive to *local* transformations of the images, e.g. caused by noise (which is for instance typical for radiographs). Therefore, the following image distortion model (IDM) is proposed:

When calculating the distance between two images $x$ and $\mu$, small local deformations are allowed. That is, the image distortion model does not compute the squared error between a pixel $(i, j)$ in $x$ and its counterpart in $\mu$, but it looks for the 'best-fitting' pixel in $\mu$ within a certain neighbourhood $R_{ij}$ around the corresponding pixel (see Figure 5.7):

$$D_{dist}(x, \mu) = \sum_{i=1}^{I} \sum_{j=1}^{J} \min_{(i',j') \in R_{ij}} \|x_{ij} - \mu_{i'j'}\|^2 \tag{5.16}$$

for images with dimension $I \times J$. Typically $R_{ij}$ is chosen to be square, containing $(2r+1) \times (2r+1)$ image pixels. Thus, choosing $r = 0$ yields Euclidean distance. Note that non-integer region sizes can easily be realized by – for instance – using linear interpolation between pixels. This is visualized in Figure 5.6 [Theiner 2000]. Obviously, this fully unrestricted distortion approach can model wanted as well as unwanted (i.e. meaningless) transformations. Nevertheless, an appropriate choice of $R_{ij}$ leads to a significant improvement, especially in the field of radiograph classification (cp. the experimental results presented in Chapter 9).

### 5.2.1 An extended Distortion Model

Looking at Equation (5.16) it is evident that with increasing neighbourhood $R_{ij}$, the transformations realized by the distortion model violate the assumption that the class-membership of the original input image equals that of the transformed input image. In fact, the distortion distance between almost any two images can be reduced to a value near zero by increasing $R_{ij}$, leading to a significant increase in classification error. To compensate for this, a cost function $C(i, i', j, j')$ is introduced, which models the costs

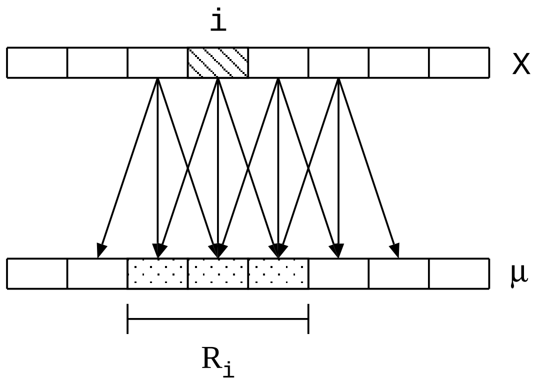


Figure 5.7: One-dimensional example of the distortion model with r=1.

for deforming a source pixel $x_{ij}$ in the input image to a target pixel $\mu_{i'j'}$ in the reference image:

$$D_{dist}(x,\mu) = \sum_{i=1}^{I}\sum_{j=1}^{J} \min_{(i',j')\in R_{ij}} \{\|x_{ij} - \mu_{i'j'}\|^2 + C(i,i',j,j')\} \tag{5.17}$$

There are at least two methods to estimate $C(i,i',j,j')$ for a given problem:

- Choose $C(i,i',j,j')$ empirically, e.g. by using a $\delta$-weighted Euclidean distance between the source pixel $x_{ij}$ and the target pixel $\mu_{i'j'}$. This way, small local transformations are preferred to most probably unwanted long-range pixel transformations.

- Learn $C(i,i',j,j')$ by using training samples and a maximum-likelihood approach. That is, do 'meaningful' transformations in training and choose $C(i,i',j,j')$ using relative frequencies of possible transformations. The more often a certain transformation was performed in training, the lower its cost in the recognition process.

In the experiments conducted throughout this work, the empirical model to choose $C(i,i',j,j')$ is used [Dahmen+ 2000a], arriving at

$$D_{dist}(x,\mu) = \sum_{i=1}^{I}\sum_{j=1}^{J} \min_{(i',j')\in R_{ij}} \left\{\|x_{ij} - \mu_{i'j'}\|^2 + \delta \cdot \left( \|i-i'\|^2 + \|j-j'\|^2 \right) \right\}. \tag{5.18}$$

Figure 5.8 visualizes the effects of using different region sizes without any cost function (i.e. $\delta = 0$), whereas Figure 5.9 visualizes the effect of using different weights for the cost function using constant a neighbourhood $r = 1$. In both cases, an image taken from the US Postal Service database (leftmost) is 'morphed' into another image (rightmost), which either belongs to a competing class (top row) or to the same class (bottom row). As can be seen, choosing suitable values for both parameters, the allowed range of possible transformations can be steered effectively.

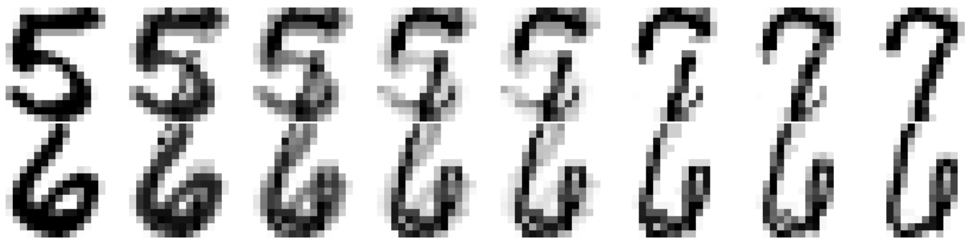

Figure 5.8: Effects of increasing r using $\delta = 0$. Left to right: r= 0.0, 0.2, 0.5, 0.8, 0.9, 1.0, 1.5, 2.0.

Figure 5.9: Effects of increasing $\delta$ using $r = 1.0$. Left to right: $\delta = 0.0,\ 1.0,\ 2.0,\ 3.0,\ 4.0$

### 5.2.2 Distorted Tangent Distance

So far, two different invariant distance measures have been discussed. The first one, tangent distance, aims at the compensation of *global* image transformations such as rotations or shifts, whereas the second one, the image distortion model, compensates for *local* image transformation, such as caused by image noise. In this sense, both approaches are somewhat orthogonal to each other. Thus, combining both into a single distance measure sounds reasonable, as the effects of both approaches are likely to be additive.

The combination of both methods is done as follows:

- Given two images $x$ and $\mu$, compute the tangent vectors for $\mu$ (single-sided approach). Now, compute the optimal tangent approximation $x'$ for the image $x$ given $\mu$ and its tangent vectors (of course one could also use the tangent vectors for $x$ or apply a double sided approach). This optimal approximation can be regarded as a registered version of the image $x$ with respect to $\mu$.

- Now, compute the image distortion model distance between $x'$ and $\mu$.

This distance measure is called *distorted tangent distance* in the following and proved to be especially effective on the IRMA task. This is not surprising, as the image distortion model was developed taking into consideration the special properties of medical radiographs.

### 5.2.3 Thresholding

In the IRMA experiments conducted throughout this work, a simple method called *thresholding* is also applied to obtain local invariances. The idea is to simply restrict the maximum distance between two pixels by introducing a distance threshold $d_{max}$. Now, the computation of – for instance – the squared Euclidean distance between two images is computed as follows:

$$d(x, \mu) = \sum_{i=1}^{I} \sum_{j=1}^{J} \min\{d_{max}, \|x_{ij} - \mu_{ij}\|^2\} \tag{5.19}$$

In the experiments, $S$=3500 is used, with the maximum possible distance between to pixels being 255×255=65,025, as the IRMA images are normalized to grayvalues between 0 and 255.

# Chapter 6

# Virtual Data Creation

In the previous chapter, a number of invariant distance measures have been introduced. Another possibility to incorporate invariances into a classifier is to create virtual data using transformed variants of the available images.

## 6.1 Creating Virtual Training Data

A typical drawback of statistical classifiers is their need for a large amount of training data, which is crucial for reliable parameter estimation but not available in many applications. One possibility to overcome this shortcoming is to create virtual data, which is a common approach in pattern recognition. The basic idea is to choose transformations that respect class membership and apply these to the reference images. The resulting augmented training set is then used to train the free parameters of the classifier. For instance, in the US Postal Service experiments conducted in the course of this work, a shift transformation was applied to multiply the available training data. Choosing $\pm 1$ pixel shifts into the directions of the 8-neighbourhood of a pixel, the training set size is extended from 7,291 to 65,619 images, i.e. by a factor of nine. Using other transformations such as rotations or variations of line thickness did not improve the best results obtained on this particular task. Besides resulting in more reliable parameter estimation, creation of virtual training data also incorporates invariances into the classifier (as transformed version of the reference images have been seen in training). One example for the successful application of virtual training data are the experiments performed by Drucker et al. on the MNIST handwritten digits task. Making use of excessive virtual training data creation (multiplying the available 60,000 images to some million training examples) in combination with a boosted artificial neural net, the authors reported the best known error rate of 0.7% on that particular task in 1993 [Drucker+ 1993].

It should be noted that the creation of virtual data makes sense, even if combined with tangent distance. This is because tangent distance is only approximately invariant, in this case for instance approximately invariant with respect to image shifts. Thus, enriching the training data with shifted copies of the original images yields a better approximation of the real manifolds, as the shifted images lie exactly on it. This matter is illustrated in Figure 6.1.

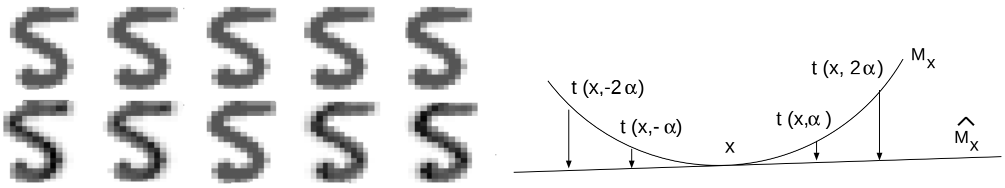


Figure 6.1: Left: Images obtained by shifting a digit and by finding the closest point in the tangent space, original image in the middle. The upper row shows the shifted images with the closest tangent approximation in the lower row. Right: Schematic illustration - the transformation $t$ is a horizontal shift here and $\alpha$ corresponds to the displacement of one pixel.

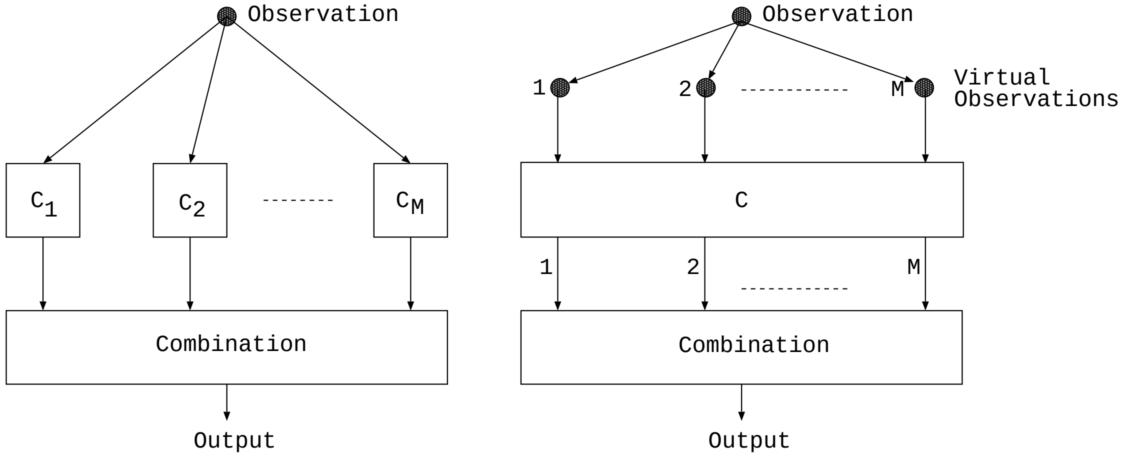


Figure 6.2: Classifier Combination (left) vs. the Virtual Test Sample method (right).

Throughout this work, the idea of creating virtual training data is extended to the creation of virtual test data. This topic is dealt with in the following Section.

## 6.2 Creating Virtual Test Data

The idea of creating virtual test data was inspired by classifier combination schemes. In classifier combination, a number of different classifiers $C_1, C_2, ..., C_M$ is trained instead of just one. An observation is then classified by each of the classifiers separately and – using methods called classifier combination schemes – a final decision for the original observation is derived. Two well known methods for the construction of classifiers to be combined are *bagging* [Breiman 1994] and *boosting* [Freund & Schapire 1996]. Contrary to this, the basic idea of the virtual test sample method is to use multiple observations $x_1, ..., x_m$ (generated from the original observation $x$ to be classified) and a single classifier $C$ instead of using multiple classifiers and a single observation. Both approaches are visualized in Figure 6.2.

Before introducing the virtual test sample method (VTS) in Chapter 6.2.2, a brief introduction is given regarding classifier combination schemes.

### 6.2.1 Classifier Combination Schemes

The general idea of classifier combination is quite simple: Given a particular pattern recognition problem, the goal is usually to implement a system which achieves the best possible recognition results on unseen data. Thus, in may cases, a variety of pattern recognition approaches is evaluated and the one performing best is chosen to solve the task at hand. Unfortunately, in that approach, all other systems that have been developed are useless. In opposite to this, the idea of classifier combination is to use all classifiers $C_m, m = 1, ..., M$ for classification and to come to a final decision for an observation to be classified by combining their outputs in a suitable way.

In the last years, a number of classifier combination schemes have been proposed [Kittler$^+$ 1998]. Note that if such combination rules should be meaningful, the outputs of the classifiers must be normalized. Thus, it is assumed here that each classifier $C_m$ computes posterior probabilities $p_m(k|x)$ for each class $k = 1, ..., K$ given the observation $x \in \mathbb{R}^D$, which are normalized in the sense that $\sum_{k=1}^{K} p(k|x) = 1$ by definition. At this point, it should be noted that - for instance - the outputs of an artificial neural net approximate such posterior probabilities [Ney 1995]. Thus, this normalization comes for free in many applications of neural networks (assuming that a sufficient amount of training data is available).

Probably the easiest way to come to a combined decision is *majority vote*. In this case, each classifier $m$ votes for the class $k$ maximizing $p_m(k|x)$. The observation $x$ is then classified as belonging to the class with most votes. To avoid ties, i.e. the case that two or more classes have the same number of votes, weighted votes can be introduced. In that case, the vote of each classifier could – for instance – be weighted with the according posterior probability (the approach being similar to the sum rule presented below).

Another combination scheme commonly used in literature is the *product rule*. Here the decision function for the combined decision is:

$$r(x) = \underset{k}{\operatorname{argmax}} \left\{ \prod_{m=1}^{M} p_m(k|x) \right\} \tag{6.1}$$

A drawback of the product rule is the fact that it is very sensitive to 'outliers'. If a single classifier produces a posterior probability near zero for a class $k$, the decisions of the other classifiers become more or less irrelevant, as the resulting product will quite probably be close to zero, too.

Therefore, in many practical cases, a combination scheme called *sum rule* is used, which is more robust with respect to the beforementioned outliers. In this case, the decision rule for the combined decision becomes:

$$r(x) = \underset{k}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} p_m(k|x) \right\} \tag{6.2}$$

In [Kittler+ 1998] the authors discuss a number of classifier combination schemes, among them the schemes presented above. Furthermore, an analysis of their theoretical properties is presented, i.e. what probabilistic assumptions have to be made to justify a given classifier combination scheme? Interestingly, the assumption

$$p(k) \sim p(k|x) \tag{6.3}$$

leads to a theoretical justification of the sum rule (which - according to the authors - yielded the best results in their recognition experiments). Note that it is difficult to understand why this approach should work in a practical application, as the important fact that the posterior probabilities depend on $x$ is neglected here. In other words, the sum rule for classifier combination is based on the rather strong assumption that the features extracted from the observations contain no discriminatory information, i.e. they are meaningless. Yet, Kittler also showed in his work that the good performance of the sum rule could possibly be explained by its error tolerance. In fact, he showed that for the sum rule, errors in estimating the real (and therefore usually unknown) posterior probabilities are dampened, while for instance in the case of the product rule, these estimation errors are amplified. For more details on this topic, see [Kittler+ 1998].

In the following, the sum rule is derived in the framework of the virtual test sample method. It is shown that in this case, derivation of the sum rule is straightforward and that no assumption such as given in Equation (6.3) is needed.

### 6.2.2 The Virtual Test Sample Method

The basic idea of the *virtual test sample method* (VTS) is to create a number of virtual test samples $x(\alpha) = t(x, \alpha),\ \alpha \in \mathcal{M}$, with $M = |\mathcal{M}|$, where $t(x, \alpha)$ is a transformation with parameters $\alpha \in \mathbb{R}^L$ respecting class-membership. For instance, in the case of the US Postal Service database, $\pm 1$ pixel shifts were applied, i.e. $M = 9$. As an image cannot be shifted into different directions at the same time, the resulting 'events' $x(\alpha),\ \alpha \in \mathcal{M}$ can be regarded as being mutually exclusive. Thus, a final decision for the original observation can be computed as follows:

$$\begin{aligned}
x \longmapsto r(x) \quad &= \quad \underset{k}{\operatorname{argmax}} \{p(k|x)\} \\
&= \quad \underset{k}{\operatorname{argmax}} \left\{ \sum_{\alpha \in \mathcal{M}} p(k, \alpha|x) \right\} \\
&= \quad \underset{k}{\operatorname{argmax}} \left\{ \sum_{\alpha \in \mathcal{M}} p(\alpha|x) \cdot p(k|x, \alpha) \right\} \\
&\overset{model}{=} \quad \underset{k}{\operatorname{argmax}} \left\{ \sum_{\alpha \in \mathcal{M}} p(\alpha) \cdot p(k|x(\alpha)) \right\}
\end{aligned} \tag{6.4}$$

In the above, the simultaneous occurrence of an observation $x$ and a parameter vector $\alpha \in \mathbb{R}^L$ is modeled by the virtual test sample $x(\alpha)$, i.e. by applying the respective transformation to the observation. Furthermore, we model $p(\alpha|x)$ by $p(\alpha)$ in the experiments. Thus, to come to a final decision for the original observation, we only have to add the posterior probabilities $p(k|x(\alpha))$, weighted with the prior probabilities $p(\alpha)$ of the transformation parameters. In the experiments conducted throughout this work these transformation parameters are assumed to be uniformly distributed. Thus, Equation (6.4) reduces to

$$x \longmapsto r(x) = \underset{k}{\operatorname{argmax}} \left\{ \sum_{\alpha \in \mathcal{M}} p(k|x(\alpha)) \right\} \tag{6.5}$$

Note that the only assumption made here is that the virtual test samples created are mutually exclusive. As such a sample is the result of applying a unique transformation to the given observation, this assumption seems reasonable.

### 6.2.3 Properties of the Virtual Test Sample Method

The proposed virtual test sample methods has the following properties (in comparison to the sum rule as used in classifier combination), which are discussed in the following:

I) *Computational complexity:*
The computational complexity of the recognition step using VTS is the same as compared to classifier combination. That is, the computational complexity generally increases by a factor of $M$ (because in both approaches, $M$ posterior probabilities have to be computed). Yet, the computational complexity of the VTS training phase is significantly lower than that of classifier combination schemes, as only a single classifier has to be trained. This is especially important for statistical classifiers, where the training step is computationally expensive in many cases.

II) *Theoretical basis:*
In contrast to the derivation of the sum rule in the framework of classifier combination, VTS sum rule is straightforward to derive, with the assumption of mutual exclusiveness of the $x(\alpha)$ sounding reasonable.

III) *Increased transformation tolerance/ invariance:*
Obviously, by creating virtual test samples, invariance properties with respect to the transformations used for virtual test data creation are incorporated into the classifier.

IV) *Ease of implementation & effectiveness:*
VTS is very simple to embed into an existing classifier, assuming a suitable normalization of the classifier's output. Furthermore, using VTS significantly reduces the USPS error rates in the experiments conducted throughout this work. For real-time applications – similar to classifier combination – VTS is obviously straightforward to parallelize, as it is inherently parallel.

V) *Applicable together with classifier combination:*
In principle, VTS and classifier combination can be used at the same time. In fact, the best VTS recognition result obtained on the US Postal Service database could be slightly improved by combining the VTS method with classical classifier combination using the sum rule (cp. results as presented in Section 7).

VI) *Incorporation of prior knowledge about transformation probabilities:*
Finally, it is possible to incorporate prior knowledge into VTS classification via an appropriate choice of the probabilities $p(\alpha)$ (model) respectively $p(\alpha|x)$ (exact solution). Although this was not done in the experiments conducted throughout this work, it might be a desirable property in other practical applications. For instance, these probabilities could be learned from the training data.

# Chapter 7

# Probabilistic Framework for Tangent Distance

In this chapter, the concept of tangent distance as proposed in Chapter 5 is embedded into a statistical framework. It is shown that computation of single-sided tangent distance can be interpreted as using a structured covariance matrix, if the tangents are applied on the side of the references [Dahmen+ 2000d, Keysers 2000a, Keysers+ 2000b]. Because of this, other approaches to impose certain structures on the covariance matrix are also dealt with here.

## 7.1 Probabilistic Interpretation of Tangent Distance

For the theoretical considerations presented here, two cases are distinguished, namely variations in the references respectively the observations. Furthermore, it is shown that tangent distance can also be applied in the case that no prior knowledge concerning the variation contained in the images is available. In that case, the tangent vectors are estimated and computed as the principal components of certain covariance matrices (being the result of a maximum likelihood estimation approach of the tangent vectors).

In the following considerations, integration over the unknown parameter $\alpha \in \mathbb{R}^L$ is performed (which represents the transformation parameters) in order to obtain the probability density function $p(x|\mu, \Sigma)$:

$$\begin{aligned}
p(x|\mu,\Sigma) &= \int p(x,\alpha|\mu,\Sigma)\, d\alpha \\
&= \int p(\alpha|\mu,\Sigma)\cdot p(x|\mu,\alpha,\Sigma)\, d\alpha \\
&= \int p(\alpha)\cdot p(x|\mu,\alpha,\Sigma)\, d\alpha
\end{aligned} \tag{7.1}$$

Note that $p(\alpha|\mu,\Sigma) = p(\alpha)$, as $\alpha$ is assumed to be independent of the parameters $\mu$ and $\Sigma$. Furthermore, it is assumed that the distribution of the $\alpha_l$ is a Gaussian with mean 0 and covariance matrix $\gamma^2 I$, i.e.

$$p(\alpha) = \mathcal{N}(\alpha|0, \gamma^2 I). \tag{7.2}$$

Two models for $p(x|\mu,\alpha,\Sigma)$ will be investigated: In the first model, the variations will be applied on the side of the reference parameters, whereas in the second model variations of the observations are dealt with. Note that in the following calculations, we assume that

$$\mu_l^T \Sigma^{-1} \mu_{l'} = \delta_{l,l'}, \tag{7.3}$$

where $\delta_{l,l'}$ is the Kronecker delta, which is equal to one for $l = l'$ and zero otherwise.

### 7.1.1 Variations in the Reference Images

First it is assumed that the references $\mu$ are subject to certain transformations, which are modelled using the (assumedly known) tangent vectors $\mu_l$. The variation of the references is modelled via the tangent approximation presented in Chapter 5.1:

$$\mu(\alpha) = \mu + \sum_{l=1}^{L} \alpha_l \mu_l \tag{7.4}$$

Thus, one obtains the following density function (assuming a normal distribution) for a given parameter vector $\alpha \in \mathbb{R}^L$:

$$\begin{aligned} p(x|\mu,\alpha,\Sigma) &= \mathcal{N}(x|\mu + \sum_{l=1}^{L} \alpha_l \mu_l, \Sigma) && (7.5) \\ &= \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp\left( -\frac{1}{2}(\mu + \sum_{l=1}^{L} \alpha_l \mu_l - x)^T \Sigma^{-1} (\mu + \sum_{l=1}^{L} \alpha_l \mu_l - x) \right) \end{aligned}$$

By inserting this term into Equation (7.1) and integrating over $\alpha$, one arrives at (the proof is given in Appendix B.1 and was motivated by [Ney 2000a]):

$$\begin{aligned} p(x|\mu,\Sigma) &= (1+\gamma^2)^{-\frac{L}{2}} \cdot \det(2\pi\Sigma)^{-\frac{1}{2}} \cdot \\ & \exp\left[ -\frac{1}{2} \left( (\mu - x)^T \Sigma^{-1} (\mu - x) - \sum_{l=1}^{L} \frac{((\mu - x)^T \Sigma^{-1} \mu_l)^2}{(\frac{1}{\gamma^2} + \mu_l^T \Sigma^{-1} \mu_l)} \right) \right] && (7.6) \end{aligned}$$

Interestingly, the exponent in Equation (7.6) yields Mahalanobis distance for $\gamma \to 0$ and tangent distance for $\gamma \to \infty$. In the latter case, it can be interpreted as computing

Mahalanobis distance and subtracting the fractions of the total distance which occur along the direction of the tangent vectors. Although infinite variance of the transformation parameters $\alpha$ is not necessary for the probabilistic interpretation of tangent distance, in the experiments no improvements could be obtained by restricting $\gamma$ to 'small' values. These experiences match with that reported by SIMARD, who observed that the minimizing values for the $\alpha_l$ are usually small [Simard$^+$ 1998].

Using the relation

$$x^T(A^{-1}+bb^T)x = x^TA^{-1}x + x^Tbb^Tx = x^TA^{-1}x + (b^Tx)^2 = x^TA^{-1}x + (x^Tb)^2 \tag{7.7}$$

Equation (7.6) can be reformulated as

$$\begin{aligned} p(x|\mu,\Sigma) &= (1+\gamma^2)^{-\frac{L}{2}} \cdot \det(2\pi\Sigma)^{-\frac{1}{2}} \cdot \\ & \exp\left[-\frac{1}{2}\Big((\mu-x)^T \underbrace{(\Sigma^{-1} - \sum_{l=1}^{L} \frac{(\mu_l^T\Sigma^{-1})^T(\mu_l^T\Sigma^{-1})}{\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l})}_{(*)}(\mu-x)\Big)\right] \end{aligned} \tag{7.8}$$

where the inverse of the above matrix $(*)$ can be interpreted as a specially structured covariance matrix (with increased variances along the directions of the tangent vectors). In Appendix B.2 it is shown that the resulting covariance matrix is $(\Sigma + \gamma^2 \cdot \sum_{l=1}^{L} \mu_l\mu_l^T)$. Furthermore, it can be shown that

$$\begin{aligned} \det(\Sigma + \gamma^2 \cdot \sum_{l=1}^{L} \mu_l\mu_l^T) &= \det(\Sigma) \cdot \prod_{l=1}^{L} \left(1 + \gamma^2\mu_l^T\Sigma^{-1}\mu_l\right) \\ &= \det(\Sigma) \cdot (1+\gamma^2)^L \end{aligned} \tag{7.9}$$

(the proof is given in Appendix B.3). Thus, computing single sided tangent distance - and applying the tangents on the side of the references - can be interpreted as using a specially structured covariance matrix in a Gaussian distribution:

$$p(x|\mu,\Sigma) = \mathcal{N}(x|\mu,\Sigma') \quad \text{with} \quad \Sigma' = \Sigma + \gamma^2 \sum_{l=1}^{L} \mu_l\mu_l^T \tag{7.10}$$

The case $\gamma \to \infty$ can be interpreted as the case of a degenerated Gaussian distribution, with infinite variance along the direction of the tangent vectors. Alternatively, it can be regarded as a Gaussian distribution in the reduced subspace, which arises from eliminating the tangent vector directions in the original space. Thus, with $x \in \mathbb{R}^D$ and $\alpha \in \mathbb{R}^L$, the resulting subspace is of dimension $D - L$. An advantage of modelling the Gaussian distribution in this reduced subspace is that in this space the model is normalized. Some interesting remarks on this 'normalization problem' can be found in [Hinton$^+$ 1997, Meinicke$^+$ 1999, Moghaddam$^+$ 1996]. For the limiting case $\Sigma = I$, a similar result was derived in [Hastie$^+$ 1998].

The generalisation to $K$-class problems can be done similarly and one obtains

$$p(x|k) = \mathcal{N}(x|\mu_k, {\Sigma'}_k), \qquad \text{where} \qquad {\Sigma'}_k = \text{diag}(\Sigma) + \gamma^2 \sum_{l=1}^{L} \mu_{kl}\ \mu_{kl}^T \tag{7.11}$$

was used in the experiments and

$$\forall k: \quad \mu_{kl}^T \Sigma^{-1} \mu_{kl'} = \delta_{l,l'}, \tag{7.12}$$

### 7.1.2 Variations in the Observations

In the above considerations it was assumed that the references computed in the training phase are subject to certain variations. In the following, variations of the observations are dealt with. Again, these variations are modelled using tangent approximations:

$$x(\alpha) = x + \sum_{l=1}^{L} \alpha_l x_l \tag{7.13}$$

Assuming that the tangent vectors $x_l$ are linear in $x$ (which holds e.g. for affine transformations, but not for the line thickness tangent vector), these tangent approximations can also be given as

$$x(\alpha) = [I + \sum_{l=1}^{L} \alpha_l T_l]\ x, \tag{7.14}$$

where $T_l \in \mathbb{R}^{D\times D}$ is the 'derivation' matrix of the $l$-th transformation considered. Further defining

$$T_\alpha := \sum_{l=1}^{L} \alpha_l T_l \ \text{ and } \ M_\alpha := I + T_\alpha, \tag{7.15}$$

the corresponding density function can be modeled as follows (note that the resulting "distribution" is not normalized):

$$\begin{aligned} p(x|\mu, \alpha, \Sigma) &= \text{“}\mathcal{N}(x(\alpha)|\mu, \Sigma)\text{”} \\ &= \text{“}\mathcal{N}(M_\alpha x|\mu, \Sigma)\text{”} \end{aligned}$$

Normalization can be obtained by applying the transformation on the side of the reference parameters, which can be done like follows:

$$\begin{aligned} p(x|\mu, \alpha, \Sigma) &= \mathcal{N}(x|M_\alpha^{-1}\ \mu, M_\alpha^{-1}\ \Sigma\ {M_\alpha^{-1}}^T) \\ &= \mathcal{N}(x|\mu', \Sigma') \\ &= \frac{1}{\sqrt{\det(2\pi\Sigma')}} \exp\left(-\frac{1}{2}(x-\mu')^T {\Sigma'}^{-1}(x-\mu')\right) \end{aligned} \tag{7.16}$$

Now, since the covariance matrix $\Sigma' = M_\alpha^{-1}\,\Sigma\,M_\alpha^{-1^T}$ depends on $\alpha$, the solution of the integral resulting from Equation (7.1) is far more difficult and so far unknown (cp. Appendix B.4). Yet, one can still compute the mean vector $\mu_T$ and the covariance matrix $\Sigma_T$ of the resulting distribution using the moment method. The Gaussian distribution $\mathcal{N}(x|\mu_T, \Sigma_T)$ can then be considered an approximation to the exact (in general non-Gaussian) distribution $p(x|\mu, \Sigma)$.

The basic idea of the moment method is to replace the expected value

$$E\{f(x)\} = \int f(x) \cdot p(x)\ dx \tag{7.17}$$

of a function $f(x)$ by the empirical average over the given training samples ("sampling")

$$\hat{E}\{f(x)\} = \frac{1}{N}\sum_{n=1}^{N} f(x_n). \tag{7.18}$$

For the case considered here, one obtains

$$\begin{aligned} E\{f(x)\} &= \int f(x) \cdot p(x)\ dx && (7.19)\\ &= \int\int p(x,\alpha) \cdot f(x,\alpha)\ d\alpha\ dx \\ &= \int p(x) \int p(\alpha|x) \cdot f(x,\alpha)\ d\alpha\ dx \\ &\overset{model}{=} \int p(x) \underbrace{\int p(\alpha) \cdot f(x,\alpha)\ d\alpha}_{=:F(x)}\ dx && (7.20) \end{aligned}$$

Now, applying the moment method, one arrives at

$$\hat{E}\{f(x)\} = \frac{1}{N}\sum_{n=1}^{N} F(x_n) \tag{7.21}$$

In the following considerations, the tangent approximations of the references are given by

$$x_n(\alpha) = x_n + \sum_{l=1}^{L} \alpha_l x_{nl}, \tag{7.22}$$

where $x_{nl}$ is the $l$-th tangent vector of the reference $x_n$. Now, the estimation of the first moment (mean vector) and the second centralized moment (covariance matrix) can be obtained by regarding $f(x,\alpha) = x(\alpha)$ respectively $f(x,\alpha) = (x(\alpha) - \mu)(x(\alpha) - \mu)^T$. Furthermore, it is assumed that the distribution of the transformation parameters is a Gaussian with zero mean and covariance $\Sigma_\alpha$ (although the assumption $E\{\alpha\} = 0$ is sufficient for the following calculations), i.e.

$$p(\alpha) = \mathcal{N}(\alpha|0, \Sigma_\alpha) \tag{7.23}$$

Thus, the estimator for the mean vector $\mu_T$ becomes

$$\begin{aligned}
\mu_T &= \int p(x) \int x(\alpha)\; p(\alpha)\; d\alpha\; dx \\
&= \int p(x) \int p(\alpha) \left( x + \sum_{l=1}^{L} \alpha_l x_l \right) \; d\alpha\; dx \\
&= \int p(x) \left( x \int p(\alpha)\; d\alpha + \int p(\alpha) \sum_{l=1}^{L} \alpha_l x_l \; d\alpha \right) \; dx & (7.24)\\
&= \int p(x) x\; dx \\
&= \mu & (7.25)
\end{aligned}$$

The second term in the sum of Equation (7.24) is the expected value of a linear function of the transformation parameters $\alpha \in \mathbb{R}^L$. As the expected value of the $\alpha$ is assumed to be zero (cp. Equation (7.23)), the term vanishes and one obtains $\mu_T = \mu$. That is, the estimation of the mean vectors does not change in the presence of tangent approximations. The following calculations show that – using similar considerations as above – this is not true for the covariance matrix:

$$\begin{aligned}
\Sigma_T &= \int\; p(x) \int (x(\alpha) - \mu)(x(\alpha) - \mu)^T p(\alpha)\; d\alpha\; dx \\
&= \int\; p(x) \int p(\alpha)(x + \sum_{l=1}^{L} \alpha_l x_l - \mu)(x + \sum_{l=1}^{L} \alpha_l x_l - \mu)^T \; d\alpha\; dx \\
&= \int\; p(x) \int p(\alpha) \left[ (x-\mu)(x-\mu)^T + (x-\mu)(\sum_{l=1}^{L} \alpha_l x_l)^T \right. \\
&\quad \left. + (\sum_{l=1}^{L} \alpha_l x_l)(x-\mu)^T + (\sum_{l=1}^{L} \alpha_l x_l)(\sum_{l=1}^{L} \alpha_l x_l)^T \right] \; d\alpha\; dx & (7.26)\\
&= \int\; p(x) \left[ (x-\mu)(x-\mu)^T + \int p(\alpha) \left( (\sum_{l=1}^{L} \alpha_l x_l)(\sum_{l=1}^{L} \alpha_l x_l)^T \right) \; d\alpha \right] \; dx \\
&= \int\; p(x) \left[ (x-\mu)(x-\mu)^T + \sum_{l=1}^{L} x_l \sum_{l'=1}^{L} (\Sigma_\alpha)_{(l,l')} {x_{l'}}^T \right] \; dx \\
&= \Sigma + \int\; p(x) \left[ \sum_{l=1}^{L} x_l \sum_{l'=1}^{L} (\Sigma_\alpha)_{(l,l')} {x_{l'}}^T \right] \; dx
\end{aligned}$$

Note that the second and third term in the sum of Equation (7.26) vanish, as they are the expected value of a linear function of the transformation parameters $\alpha$. Applying the sampling approach of the moment method and using $\Sigma_\alpha = \gamma^2 I$ as in the previous considerations (following from $p(\alpha) = \mathcal{N}(\alpha|0, \gamma^2 I)$), one finally arrives at

$$\Sigma_T = \Sigma + \frac{1}{N} \cdot \gamma^2 \sum_{n=1}^{N} \sum_{l=1}^{L} x_{nl} x_{nl}^T. \tag{7.27}$$

Similar results have been published in [Schölkopf+ 1998] in the context of support vector machines and in [Hastie+ 1998]. Especially the latter work is similar to the approach presented here, but the authors do not embed their results in a statistical framework like the one presented in this work. Furthermore, they report no improvement in classification error rate, whereas in the experiments conducted throughout this work, considerable improvements are obtained (cp. Chapter 9). Obtaining better results than without using the *tangent covariance matrix* (as given in Equation (7.27)) seems reasonable, as the resulting parameters should generalize better, especially if only a small training data set is available.

### 7.1.3 Estimating Tangent Vectors

Finally, the estimation of tangent vectors shall be discussed, which is necessary when no prior information concerning the variations of the data is given. This is for instance the case if features are used that have no interpretation as an image, i.e. where – as an example – affine transformations have no meaningful correspondence. Examples include features resulting from a linear discriminant analysis or features based on moments (cp. Chapter 1.3.2).

In this case, assuming that the number $L$ of tangent vectors sought for is known, a maximum likelihood approach can be applied to estimate suitable vectors $\mu_l,\ l = 1, ..., L$. Given a reference $\mu$ and the covariance matrix $\Sigma$, maximizing the likelihood

$$\max_{\{\mu_l\}} \prod_n \mathcal{N}(x_n|\mu, \Sigma') \quad \text{with} \quad \Sigma' = \Sigma + \gamma^2 \sum_{l=1}^{L} \mu_l \mu_l^T \tag{7.28}$$

is equivalent to minimizing the doubled negative log-likelihood (again, constant terms have been dropped here):

$$\sum_{n=1}^{N} d(x_n, \mu) = \sum_{n=1}^{N} \left[ (\mu - x_n)^T \Sigma^{-1} (\mu - x_n) - \sum_{l=1}^{L} \frac{((\mu - x_n)^T \Sigma^{-1} \mu_l)^2}{\frac{1}{\gamma^2} + \mu_l^T \Sigma^{-1} \mu_l} \right] \tag{7.29}$$

This in turn is equivalent to the maximization (with respect to the $\mu_l$) of

$$\begin{aligned} \sum_{n=1}^{N} \sum_{l} \frac{((\mu - x_n)^T \Sigma^{-1} \mu_l)^2}{\frac{1}{\gamma^2} + \mu_l^T \Sigma^{-1} \mu_l} &= \sum_{l} \sum_{n=1}^{N} \frac{\mu_l^T \Sigma^{-1} (\mu - x_n)(\mu - x_n)^T \Sigma^{-1} \mu_l}{\frac{1}{\gamma^2} + \mu_l^T \Sigma^{-1} \mu_l} \\ &= \sum_{l} \frac{\mu_l^T \Sigma^{-1} S \Sigma^{-1} \mu_l}{\frac{1}{\gamma^2} + \mu_l^T \Sigma^{-1} \mu_l} \end{aligned} \tag{7.30}$$

with $S = \sum_{n=1}^{N}(\mu - x_n)(\mu - x_n)^T$ being the scatter matrix of the data. This is maximized when the vectors $(\Sigma^{-\frac{1}{2}})^T\mu_l$ correspond to the $L$ eigenvectors with the largest eigenvalues of the matrix $(\Sigma^{-\frac{1}{2}})^T S \Sigma^{-\frac{1}{2}}$, its principal components.[1] For a proof one only needs to consider the constraint that the vectors $(\Sigma^{-\frac{1}{2}})^T\mu_l$ are orthonormalized and the problem is similar to finding the principal components for a given covariance matrix, leading to an eigenvalue equation (see e.g. [Fukunaga 1990, pp. 431-435]).

For example, assuming $\Sigma = \sigma^2 I$ (as is the case in a minimum distance setting with Euclidean distance) this implies using the directions of largest variance of the data. In a more general case one might consider using the global covariance matrix for $\Sigma$ and the class specific covariance matrix for $S$. This is equivalent to performing a global whitening transformation for a transformation of parameter space and then employing the $L$ eigenvectors with the largest eigenvalues of the class specific empirical covariance matrix as tangent vectors.

Note that in this maximum likelihood setting no meaningful solution is obtained for the case $\Sigma = S$. In this case the expression to be maximized reduces to

$$\max_{\{\mu_l\}} \sum_l \frac{\mu_l^T \Sigma^{-1} \mu_l}{\frac{1}{\gamma^2} + \mu_l^T \Sigma^{-1} \mu_l} \tag{7.31}$$

This term is a constant for $\gamma \to \infty$ and therefore not helpful for finding the 'best' $\mu_l$. For other values of $\gamma$ a further transformation of the expression to

$$\max_{\{\mu_l\}} \sum_{l=1}^{L} \left[ 1 - \frac{1}{\gamma^2} \cdot \frac{1}{1 + \gamma^2 \mu_l^T \Sigma^{-1} \mu_l} \right] \tag{7.32}$$

shows that the only information obtained is that the term $\mu_l^T \Sigma^{-1} \mu_l$ should be maximized. This implies that the length of the vectors $\mu_l$ should grow infinitely, without giving any information on the directions. If unit length is assumed for the tangent vectors, one obtains the directions of smallest variances (as the product contains $\Sigma^{-1}$). This may not be very helpful for practical classification applications, but it makes sense as a result of a maximum likelihood approach, because it minimizes the reconstruction error, retaining most of the information contained in the data.

The usage of the (local) principal components as directions of increased variance has been mentioned in the context of (local) subspace classifiers before, but it is not derived from domain knowledge. Interestingly, this approach can be derived from the probabilistic considerations presented here. For example in [Meinicke$^+$ 1999], the largest principal components are preserved (although not increased, as in the considerations presented here by setting $\gamma \to \infty$), while other directions are assumed to be directions resulting from noise. Yet, no theoretical justification for that approach is given. In [Hinton$^+$ 1995] a mixture of (local) linear models is regarded, where the directions are estimated like in

[1] Here $\Sigma^{-\frac{1}{2}}$ is defined as the matrix for which $\Sigma^{-\frac{1}{2}}\Sigma(\Sigma^{-\frac{1}{2}})^T = I$ holds, which exists, if $\Sigma$ is a non singular covariance matrix. This is also the transformation matrix of the whitening transformation (see [Fukunaga 1990, pp. 28ff]).

principal components analysis, yet again no justification for this approach is given.

Note that the classical use of principal components analysis (i.e. for feature reduction) is usually an approach contrary to the one proposed here. In the feature reduction approach, only distances along the principal components are accounted for, whereas in the approach presented here, distances arising from the principal components are neglected. In this context, a strict distinction between class specific principal components (this work) and global principal components (feature reduction) has to be made.

## 7.2 Structured Covariance Matrices

Interestingly, the use of single sided tangent distance can be interpreted as using a structured covariance matrix within a Gaussian distribution, if the tangent vectors are incorporated on the side of the references (cp. Equation (7.10)). In this framework, the empirical covariance of the data is estimated and modified in such a way, that the variances along the directions of tangent vectors approach infinity. Thus, distances that arise from these directions are neglected in classification. The resulting Gaussian is therefore degenerated, yet if one considers the space dual the subspace spanned by the tangent vectors (i.e. a $D-L$ dimensional space), this degeneration can be circumvented. In the following, another approach to structuring covariance matrices is presented, which is based on pixel neighbourhoods.

The general motivation for the use of structured covariance matrices is to reduce the number of free model parameters that have to be estimated. In principal the following approaches might be chosen:

1) Use a full covariance matrix. This is only advisable if a sufficient amount of training data is available.
2) Use a diagonal covariance matrix. This is the standard approach applied in this work.
3) Use anything in between 1) and 2), for instance band-structured covariance matrices. This approach is presented in the following.

Thus, choosing the structure of the covariance matrices for a given problem is a trade-off between model complexity and reliability of parameter estimation. In many real-world applications, the use of diagonal covariance matrices yields state-of-the art results [Dahmen+ 1998-2001].

Using full covariance matrices for object recognition implies that any two pixels within an image may be correlated. On the other hand, using diagonal covariance matrices, it is assumed that there is no correlation between different pixels at all. Both such approaches are somewhat extreme: the first suffers from a large amount of parameters, whereas the latter may be an unrealistic model in some applications. As a compromise, one could use a full covariance matrix with the restriction that the grayvalue of a given pixel only depends on the grayvalues of its neighbours. Thus, the number

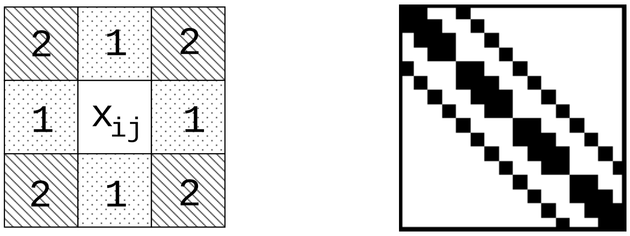


Figure 7.1: Neigbourhoods $N_1$ (1), $N_2$ $(1, 2)$ used (left). Resulting band structure of the inverse covariance matrix $\Sigma^{-1}$ for $N_1$ and $4 \times 4$ pixels sized images (right). Black pixels represent non-zero entries in $\Sigma^{-1}$.

of non-zero entries in the respective inverse covariance matrix can be significantly reduced.

Regarding the neighbourhoods $N_1$ and $N_2$ as shown in Figure 7.1 and assuming that the grayvalue of a pixel $x_{ij}$ only depends on its neighbouring pixels, the respective inverse covariance matrix $\Sigma^{-1}$ has a band structure (this can be shown using Markov random field theory [Li 1995]), with the number of bands increasing as the regarded neighbourhood grows (four bands for $N_1$, eight for $N_2$). Thus, any entry of $\Sigma^{-1}$ that does not lie on the diagonal or the bands is zero. Note that some entries on the first band are zero, too (cp. Figure 7.1). This is due to the fact that wrap-around is not considered, e.g. a pixel at the left border of an image is not a neighbour of the corresponding pixel at the right border.

Considering this, a maximum likelihood estimation of $\Sigma$, i.e.

$$\max_{\{\Sigma\}} \prod_{k=1}^{K} \prod_{n=1}^{N_k} \mathcal{N}(x_{nk}|\mu_k, \Sigma) \tag{7.33}$$

given the training observations $x_{nk}, n = 1, ..., N_k$ of a class $k$) yields the interesting result that only estimations for those entries in $\Sigma$ that lie on the diagonal or the bands can be given. Thus, one knows each entry in $\Sigma$ that is not known in $\Sigma^{-1}$ (where knowledge about the occurences of zeros is available) and vice versa. Hence, an estimation for $\Sigma^{-1}$ (under the constraint that only neighbouring pixels depend on each other) can be found by solving the bilinear equation system

$$\Sigma \cdot \Sigma^{-1} = I \tag{7.34}$$

where $I$ is the matrix of identity. With $\Sigma, \Sigma^{-1} \in \mathbb{R}^{D \times D}$, this yields $D^2$ equations with $D^2$ unknowns. In the experiments conducted throughout this work, the solution of this equation system is obtained by applying the Gauss-Seidel algorithm [Press$^+$ 1992, pp. 864-869].

# Chapter 8

# Towards Complex Object Detection

In the last chapters, a statistical classifier has been presented for single object recognition, where the baseline mixture density based classifier has been extended by using invariant distance measures and by creating virtual data. In this chapter, some possibilities to extend the presented approach to more complex object recognition task are presented [Dahmen+ 2000b, Güld 2000]. These tasks are:

- *Detection of a single object in an image*:
  In the considerations presented so far, it was assumed that the given image contains a single object, with the position of the object varying only slightly with respect to size and position. Contrary to this, object detection is now dealt with, i.e. the realisation of the object is unknown. Here, the task of the system is to determine the position (the scale, ...) and the class index of the object present in the scene.

- *Multi-Object Recognition*:
  In this case, the classifier is presented a scene containing an unknown number of objects. The task of the system is then to determine the number of present objects as well as their position (scale, ...) and the respective class labels. Note that the presence of multiple objects in a scene can be regarded as a problem with inhomogeneous background, because given one particular object, the other objects are a special kind of background for this.

## 8.1 Spotting Single Objects in a Scene

The basic idea for the detection of an object in a complex scene is to apply a *sliding window approach*. That is, a window of fixed size (e.g. $16 \times 16$ pixels) is moved over the given image, where at each image position the image part contained in the sliding window is interpreted as a subimage. Now, using the algorithms presented in the previous chapters, the system checks whether a known object is present in the current subimage. Note that in this case the system must be able to perform a *reject*, because most of the subimages extracted from a single object scene do not contain a known object. Making use of this sliding-window approach, shift invariance is incorporated into the system. Scale invariance is obtained by applying a multiscale approach. Thus, a given image $I$ is processed in multiple scales $I_1, ..., I_S$. Note that other invariances can be incorporated

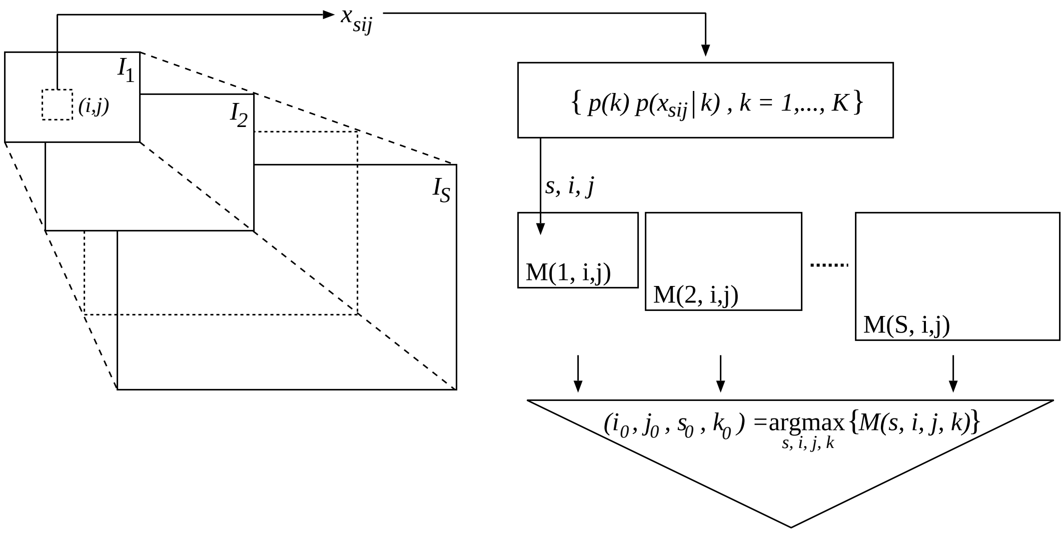


Figure 8.1: Visualization of the Object Detection approach.

by either explicitly using the respective transformations or by using tangent distance (if only small variations are present in the data). Note that – for instance – the combination of a multi-scale or sliding window approach and tangent distance makes sense, as tangent distance is only *approximatively* invariant with respect to *small* image transformations.

In the following, the feature vector being extracted from the subimage $R(s,i,j)$ at position $(i,j)$ of the original image at scale $s$ is denoted $x_{sij}$. Applying the sliding-window approach results in a so called *saliency map* for each scale level. At each position $M(s,i,j)$ in this saliency maps, the posterior probabilities $p(k|x_{sij})$ for the classes $k = 1, ..., K$ at image position $(i,j)$ and scale level $s$ are stored. From this information, the classifier can now determine the position, scale and class of the detected object. For single object detection, this is done by computing

$$r(I) = \operatorname*{argmax}_{k} \left\{ \max_{s,i,j} \; \{p(k|x_{sij})\} \right\} \tag{8.1}$$

A visualization of the approach proposed here is given in Figure 8.1.

Note that in the experiments conducted, the size of the sliding window was chosen to match the size of the available reference images. As a result, the training phase remains the same as in single object recognition (cp. Chapter 4).

### 8.1.1 Confidence in Local Decisions

Equation (8.1) does not take into account the confidence into a local decision and is now modified as follows. Besides computing the posterior probability for a class $k$ given the current local feature vector, the probability that the current feature vector shows a valid object is modelled, too. Thus, Equation (8.1) becomes

$$\begin{aligned}
r(I) &= \operatorname*{argmax}_k \left\{ \max_{s,i,j} \left\{ p(k, x_{sij} \text{ is valid object} | x_{sij}) \right\} \right\} \\
&= \operatorname*{argmax}_k \left\{ \max_{s,i,j} \left\{ p(k | x_{sij} \text{ is valid object}, x_{sij}) \cdot p(x_{sij} \text{ is valid object} | x_{sij}) \right\} \right\} \\
&= \operatorname*{argmax}_k \left\{ \max_{s,i,j} \left\{ p(k | x_{sij}) \cdot p(x_{sij} \text{ is valid object} | x_{sij}) \right\} \right\} \qquad (8.2)
\end{aligned}$$

In Equation (8.2) the assumption is made that the events "$x_{sij}$ is valid object" and "$x_{sij}$ is object of class $k$" are independent, which sounds reasonable. The probability $p(x_{sij}$ is valid object$|x_{sij})$ is now modelled via a Gaussian distribution with mean zero, as this is the minimum distance that can occur in classification:

$$p(x_{sij} \text{ is valid object} | x_{sij}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[ -\frac{1}{2} \frac{(d_{\text{norm}}(x_{sij}) - \mu)^2}{\sigma^2} \right] \qquad (8.3)$$

$$d_{\text{norm}}(x_{sij}) = \frac{\min_{n,k} \left\{ d(x_{sij}, x_{nk}) \right\}}{\max_{s',i',j'} \left\{ \min_{n,k} \left\{ d(x_{s'i'j'}, x_{nk}) \right\} \right\}} \qquad (8.4)$$

The variance is set to $\sigma^2 = \frac{1}{2\pi}$ in the experiments. This is done, because in this case Equation (8.3) equals one if $d_{\text{norm}}(x_{sij}) = 0$ (as $\mu = 0$). Furthermore, $x_{nk}$ is the $n$-th reference image of class $k$. Note that occuring distances $d(x_{sij}, x_{nk})$ between the current feature vector $x_{sij}$ and the references $x_{nk}$ are transformed to values in [0,1] by computing $d_{\text{norm}}$. Figure 8.2 depicts the confidence of a decision with respect to $d_{\text{norm}}$. Introducing a threshold $t$, the subimage under consideration can be rejected, which means that it is likely that there is no known object at position $(i, j)$ and scale level $s$ in the given image. Thus, the decision function finally becomes:

$$r(I) = \begin{cases} r(I) \text{ as given in Equation (8.2);} & \text{if } p(x_{sij} \text{ is valid object} | x_{sij}) > t \\ \text{reject} & \text{else} \end{cases} \qquad (8.5)$$

In the experiments, the threshold $t = 0.3$ is used.

### 8.1.2 Introducing a Handicap Distance

A common problem inherent in the sliding window approach is that in many cases only a small part of the object present in the scene is explained by the respective references, leading to a misclassification of the object or to a wrong number of spotted objects (as a single object is regarded as a group of objects, which all explain only a small part of

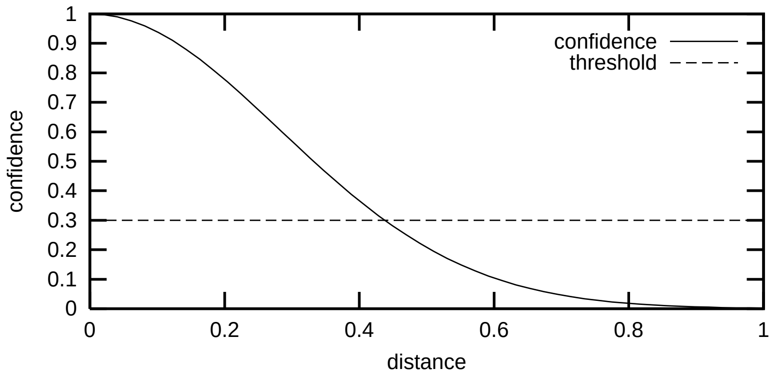


Figure 8.2: Confidence of a local decision with respect to the normalized distance $d_{\text{norm}}$.

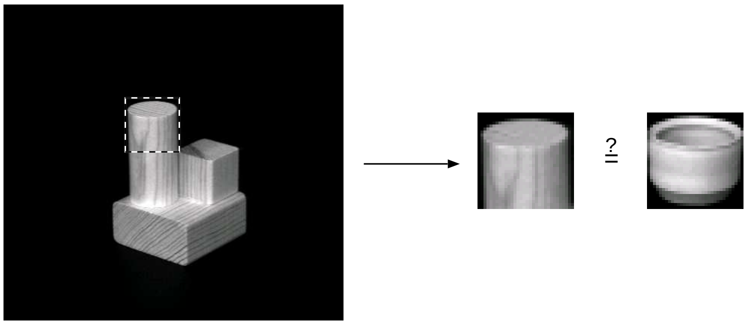


Figure 8.3: Only a small part of the original object is explained, possibly resulting in a misclassification (COIL-20 data).

the scene). An example for this problem is shown in Figure 8.3, where objects from the COIL-20 database are used. This problem is also present in the US Postal Service data, as Figure 8.4 shows. In this case, even small errors in estimating the correct object localisation may have a significant impact on the subsequent object classification.

To overcome this problem, the so-called *handicap distance* $h_{sij}$ is introduced. To compute this distance, the reference is assumed to be of infinite extent, where pixels that are not part of the real references are regarded to be background. On the COIL-20 and the US Postal Service databases, the background is assumed to have a grayvalue of zero:

$$h_{sij} = \sum_{i',j':(i',j')\notin R(s,i,j)} (I_s(i',j'))^2 \tag{8.6}$$

where $R(s,i,j)$ is the region in the image $I_s$ which is covered by the sliding window. Thus, by adding $h_{sij}$ to the distances between the current feature vector and the references (which can be Mahalanobis, Euclidean or tangent distance), the problems shown in Figures 8.3 and 8.4 can be overcome in many cases. Again, the relationship between distance measures and probability density functions should be pointed out.

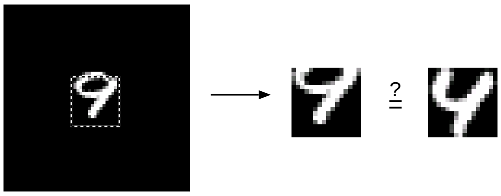

Figure 8.4: Effect of small localisation errors on the classification result on USPS.

## 8.2 Speeding up the Recognition Process

In the experiments conducted throughout this work, the following methods are applied to speed up the recognition process. Basically it is checked – using efficiently computable heuristics – whether the current feature vector possibly contains a known object. If this is not the case, it is rejected without computing Equation (8.5) (a method realizing this functionality is called *rejector* in [Baker+ 1996]).

- Assuming that the background color is zero, subimages with a small squared gray-value sum $m(x_{sij}) = ||x_{sij}||^2$ can be rejected.
- Reject subimages, for which the squared grayvalue sum within the subimage is considerably smaller than the squared grayvalue sum out of this region (handicap). In this case it is quite likely that only a small part of an object is explained (cp. Figure 8.3). This handicap is computed locally for multi-object recognition.
- Compute thresholds $t_{min}, t_{max}$ for certain criteria on the reference images. In the experiments, the minimum and maximum entropy [Lehmann+ 1997] of the reference images was computed. Subimages with an entropy varying considerably from these extreme values are rejected. Here, a variation of 10% was allowed. The entropy is chosen, because it allows the detection of variations in object size, as large scale factors usually lead to homogeneous areas in the image, resulting in a low entropy.

## 8.3 Multi-Object Recognition

The above considerations apply for the case of single object recognition so far. In the following, two methods are presented to extend the proposed system to the detection of the multiple objects. The first approach is based on repeatedly applying a slightly modified system for single object detection, whereas the second one is similar to the one applied in speech recognition.

### 8.3.1 Repeated Detection of Single Objects

One possibility to detect muliple objects in a scene is the repeated application of Equation (8.5), where the areas containing objects are removed from the saliency maps (i.e. these areas are marked as "done" and are ignored in the following detection step). Thus, the multi-object detection system can be described in pseudocode as

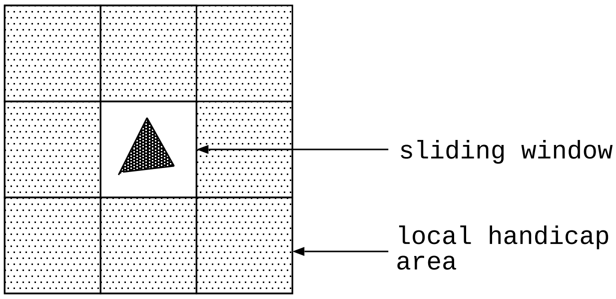


Figure 8.5: Local handicap area as used in the experiments.

(1) `WHILE true DO`
(2) use Equation (8.5)
(3) `IF` ( no more hits )
(4) `THEN exit`
(5) `ELSE` update saliency map, i.e. mark region as "done"
(6) `DONE`

The program terminates, when in step (3) a "reject" decision is met. Removing hits from the map without recomputing the saliency maps under consideration of detected objects is done in order to speed up the system. Alternatively, the saliency maps can be recomputed after each hit.

Obviously, the handicap distance introduced in Equation (8.6) cannot be used for multi-object recognition. Instead, it is modified to a local handicap. In the experiments conducted, the local handicap was computed in a $3 \times 3$ region around the current position of the sliding window (cp. Figure 8.5).

### 8.3.2 A Real Multi-Object Recognition Approach

One drawback of the multi-object classifier as presented above is the fact, that it cannot deal with the case that multiple objects are very close to each other (due to the local handicap). Furthermore, it relies on a rather heuristic reject model. Another approach, which does not suffer from these drawbacks and which is inspired from the approach taken in speech recognition is based on the following classification model:

The scene to be classified contains an unknown number $m = 0, ..., M$ of objects belonging to the classes $k_1, ..., k_M$, which is abbreviated as $k_1^M$ in the following ($M = 0$ representing the special case of an empty scene). Furthermore, reference models $p(x|\mu_k)$ exist for each of the known objects. These references are subject to certain transformations (such as the position of the object in the image, its scale etc.). That is, given transformation parameters $\vartheta_1^M$, the $m$-th reference is mapped to

$$\mu_{k_m} \rightarrow \mu_{k_m}(\vartheta_m). \quad (8.7)$$

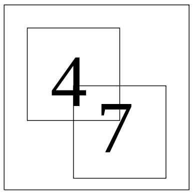


Figure 8.6: The idea of the multi object recognition approach for USPS.

Furthermore, the original scene is partitioned into $M+1$ regions $S_0^M$, where the region $S_m$ is assumed to contain the $m$-th object and region $S_0$ represents the background. In this context, $X_{S_m}$ denotes the feature vector extracted from region $S_m$:

$$X_{S_m} := \{x_{ij} : (i,j) \in S_m\} \tag{8.8}$$

The idea is now to hypothesize all unknown parameters and to look for the hypothesis which best explains the given scene to be classified. Note that this means that any pixel in the scene has to be assigned either to an object or to the class background. Thus, there is no need for a handicap model, as it has been introduced for the sliding window approach above. Furthermore, there is no need for an explicit reject option, as the case that no object is contained in the scene is explicitly a part of the model (namely $M=0$). Formally, the approach can be written as

$$r(\{x_{ij}\}) = \underset{M,k_1^M}{\operatorname{argmax}} \left\{ \max_{\vartheta_1^M} \left\{ p(k_1^M) \cdot \prod_{m=0}^{M} p(X_{S_m} | \mu_{k_m}(\vartheta_m)) \right\} \right\} \tag{8.9}$$

where $\{x_{ij}\}$ denotes the scene to get classified [Ney 2000b].
Furthermore, $p(k_1^M)$ plays essentially the same role as the language model in speech recognition [Martin$^+$ 1998, Martin$^+$ 1999]. In the experiments conducted throughout this work, a uniform distribution was assumed, i.e.

$$p(k_1^M) = \text{const}, \tag{8.10}$$

as the multiple object recognition experiments were done on USPS, using artificially created multi-object scenes. Yet, in a real-world application, prior knowledge can be modelled using a non-uniform distribution for $p(k_1^M)$. Furthermore, due to the nature of the data the approach is applied to (US Postal Service, COIL-20, IRMA), the background model is assumed to be a Gaussian distribution with $\mu_0 = 0$. In future work, the background model should be explicitly learned, as is the case in speech recognition (where the 'silence' model is learned from the training data). Furthermore, for the transformations $\mu_{k_m}(\vartheta_m)$ of the references, only horizontal and vertical translations were considered in the experiments, resulting in rectangular partitions $S_1^M$. Finally, in the experiments conducted on USPS, an overlap of up to 1/3 of the images was allowed. Here, in overlapping image regions, the maximum grayvalue of overlapping pixels was used. The idea of the multi object recognition approach is depicted in Figure 8.6.

Obviously, this real multi-object recognition approach is computationally rather expensive, as a large number of hypotheses have to be dealt with. On the other hand - going by the experiences gained in speech recognition - it can be expected to produce very good recognition results. In the digit experiments conducted, in order to reduce the computational complexity, a single density model was applied (which lead to promising recognition results in the experiments conducted, cp. Chapter 9). In order to be able to apply the approach to large, real-world images, pruning techniques have to be developed (analogue to speech recognition, where such techniques are crucial for real-time speech recognizers [Ortmanns & Ney 2000]).

# Chapter 9

# Experimental Results

In this chapter, experimental results that were obtained using the algorithms proposed in the previous chapters are presented. Basically, this chapter is subdivided into two parts. The first one deals with the single object recognition problem, using the Chair Image, the US Postal Service, the Red Blood Cell and the IRMA data. In the second part, experiments conducted on the COIL-20 database and artificial modifications of the US Postal data are described. These experiments also deal with multi-object recognition and scale invariance as discussed in Chapter 8.

## 9.1 Single Object Recognition

In the following, single object recognition results that were obtained on various datasets are presented.

### 9.1.1 Experiments on the Chair Image Data

The experiments were started on the Chair Image Database (CID) using Gaussian single densities. Without performing any feature reduction, i.e. performing appearance based pattern recognition, the best test error rate obtained was 15.1%, using class specific variance pooling. This high error rate is not surprising, as the CID feature vectors are 1,280-dimensional. Thus, a vast amount of parameters has to be estimated, given only 400 training images per class.

Using a linear discriminant analysis to reduce the feature space to $K-1=24$ dimensions (as $K$=25 for CID) improved the single density error rate to 3.3%, proving the effectiveness of the LDA-based feature reduction approach. In a second step, the reduced feature vectors were used to realize a mixture density based classifier. Figure 9.1 shows the achieved results with respect to different types of variance pooling and the total number of densities used to model the training data. The best result of 0.4% was obtained using global variance pooling and is very well comparable to the results that were reported by other groups (as shown in Table 9.1).

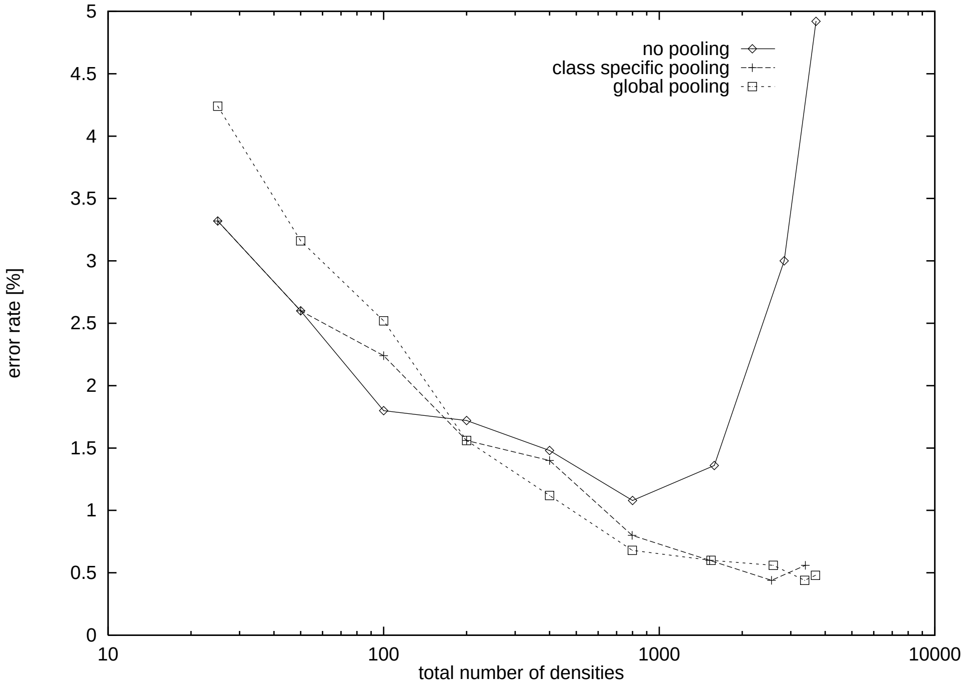


Figure 9.1: CID error rates as a function of the number of densities for three types of variance pooling.

In the case of this database the kernel density approach yields an error rate of 0.4%, too, that is the error rate could not be improved.

Because of the CID classification error rate being close to zero (0.4% error means a total of only 10 misclassifications out of the 2,500 test samples), virtual data creation was not applied to this database. Instead, to achieve a meaningful interpretation of the effects of virtual data creation (and especially the virtual test sample method), the experiments were continued on the US Postal Service task. In the following experiments – if nothing else is explicitly stated – globally pooled, diagonal covariance matrices are used.

### 9.1.2 Experiments on the US Postal Service Data

The first part of the US Postal Service (USPS) experiments were conducted to prove the effeciency of the LDA based feature reduction and the creation of virtual data. After this,

Table 9.1: Results reported on CID.

| Author | Method | Error Rate [%] |
|---|---|---|
| Blanz et al., 1996 | Support Vectors | 0.3 |
| Kressel, 1998 | Polynomial Classifier | 0.8 |
| This work | Gaussian Mixtures | 0.4 |
| | Kernel Densities | 0.4 |

Table 9.2: Results obtained on USPS without feature reduction, using various classifiers.

| Method: | Error Rate [%] | | | |
|---|---|---|---|---|
| | 1-1 | 1-9 | 9-1 | 9-9 |
| Single Densities | 19.5 | 18.3 | 22.3 | 21.5 |
| Mixture Densities | 8.0 | 6.6 | 6.4 | 6.0 |
| Kernel Densities | 6.5 | 5.5 | 5.9 | 5.1 |
| Nearest Neighbour | 6.8 | 5.9 | 6.2 | 5.3 |

Table 9.3: Results obtained on USPS with 39 LDA features, using various classifiers.

| Method: | Error Rate [%] | | | |
|---|---|---|---|---|
| | 1-1 | 1-9 | 9-1 | 9-9 |
| Single Densities | 12.8 | 12.4 | 13.1 | 11.7 |
| Mixture Densities | 6.7 | 5.9 | 4.5 | 3.4 |
| Kernel Densities | 6.3 | 5.3 | 4.2 | 3.4 |
| Nearest Neighbour | 7.0 | 5.9 | 4.9 | 3.6 |

tangent distance is incorporated into the statistical classifier, as discussed in Chapter 7.

#### 9.1.2.1 Feature Reduction & Virtual Data Creation

USPS experiments were started by applying the proposed statistical classifiers to the original USPS data, without creating virtual data and without performing feature reduction. Thus, a single density error rate of 19.5% (i.e. $I_k = 1$ for each class $k$) and a mixture density error rate of 8.0% was obtained. Creating virtual training and testing data, this error rate could be further reduced to 6.0%. Note that - not surprisingly - the single density error rate slightly increases when using virtual training data (as a single prototype has to represent a larger amount of data in this case), yet with the number of model parameters increasing, creating virtual data proves to be superior. With 5.1% error rate, the kernel density based classifier yielded the best error rate obtained without any feature reduction. An overview of the obtained results without feature reduction is shown in Table 9.2 (all results were obtained using globally pooled variances). The notation '*a*-*b*' indicates, that the number of training samples was increased by a factor of $a$ and that of the test samples by a factor of $b$. Thus, $b$=9 implies the application of the virtual test sample method as proposed in Chapter 6.2.

For further experiments on USPS, the dimensionality of the feature space was reduced by applying a linear discriminant analysis. Creating 40 pseudoclasses out of the original ten USPS classes, 39 LDA features were extracted (cp. Chapter 1.1.3.2). Without creating virtual data, the best mixture density error rate obtained was 6.7%, which could be reduced to 6.3% by using kernel densities. Creating virtual training data significantly

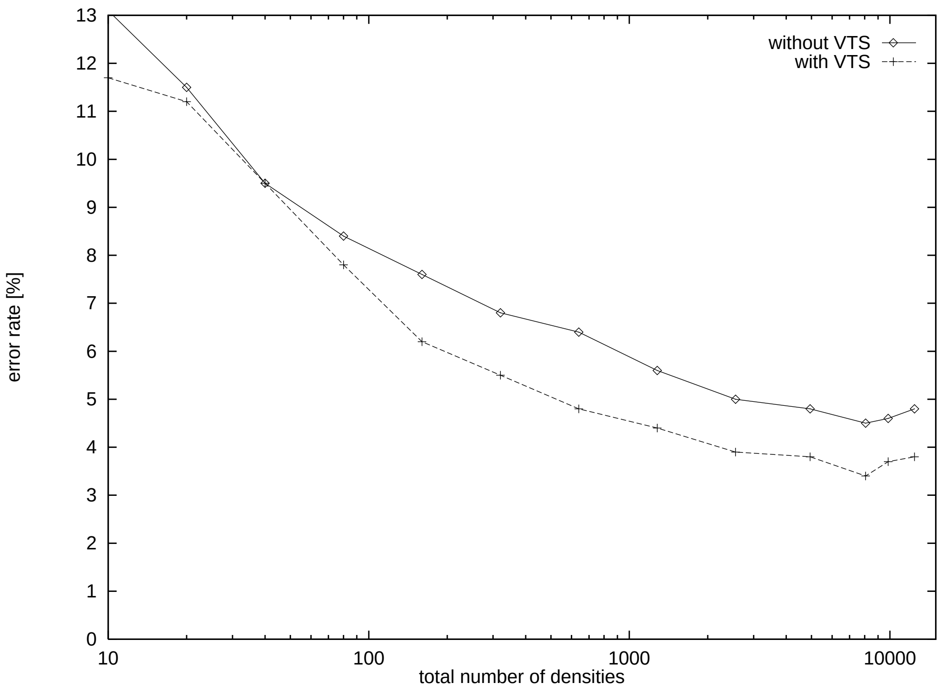


Figure 9.2: LDA Error rates obtained on USPS using globally pooled variances, with and without VTS.

reduced the error rate to 4.5% and 4.2% respectively. This improvement is mainly due to the fact, that variance estimation can be done more reliably in this low-dimensional feature space, especially if there is virtual training data available. Performing the virtual test sample method further reduced the error rate to 3.4%, which is very well comparable to results reported by other groups (cp. Table 9.7). An overview of the obtained results with LDA-based feature reduction is shown in Table 9.3.

Figure 9.2 shows the error rate of a mixture density based classifier for global variance pooling with respect to the total number of densities used to model the observed training data, with and without using the virtual test sample method. Obviously, the error rate drops significantly with the number of parameters increasing (from 13.1% to 4.5% without VTS and from 11.7% to 3.4% with VTS), yet if the the number of densities gets too high, the test error rate slightly increases. This is due to the fact that the probabilistic model is overfitted on the training data, leading to decreasing generalization properties. Strictly speaking, optimizing the density number with respect to the test error rate obtained could be considered as 'training on the testing data', but unfortunately there is no development test set available for the US Postal Service corpus.

Figure 9.4 shows the results obtained by a kernel density based classifier with respect to the variance multiplier $\alpha$ (cp. Chapter 4.3) in comparison to a nearest neighbour classifier (with error rate 4.9%, being independent of the choice of $\alpha$). Examples for nearest neighbour recognition of USPS digits are shown in Figure 9.3. Using kernel densities without VTS, the best mixture density error rate could be improved from 4.5%

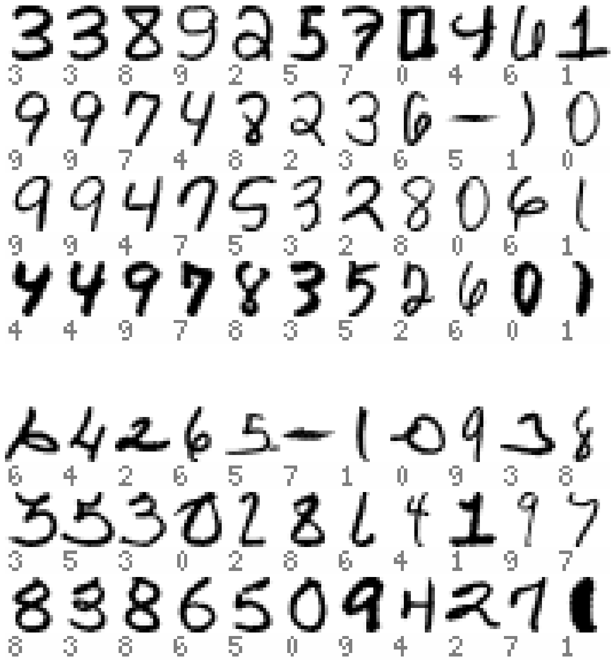


Figure 9.3: Examples of nearest neighbor recognition on USPS (with according class labels) first column: observation, next columns: best references of each class in order of increasing distance to the test pattern. Top four rows: correct classification. Bottom three rows: incorrect classification.

to 4.2%. Not surprisingly, this result is achieved using $\alpha$ =5.1. This is due to the fact that parameter estimation based on a rather small training set tends to underestimate variances. With $\alpha$ >1, this effects is compensated, yet with $\alpha$ getting too large the variances get 'flattened' too much and the error rate increases. Again, using VTS reduced the error rate significantly from 4.2% to 3.4%.

In another experiment, the question why virtual training data reduces the test error rate was investigated. There are two possible reasons for this improvement:

- *Better models for $p(x|k)$ in training*:
  One possible reason for the observed improvements is the fact, that the class conditional probabilities $p(x|k)$ can be estimated more reliably in the training phase, because there is more data to learn from.

- *Improved Feature Reduction*:
  Another possible reason is the improved estimation of the LDA transformation matrix. To construct this matrix, a general eigenvalue problem in the high-dimensional matrices $S_w$ and $S_b$ has to be solved (cp. Chapter 1.1.3.2. Thus, computing the LDA is also subject to estimation problems, as the quality of the transformation depends on the quality of the estimations of $S_w$ and $S_b$.

Table 9.4 shows that in fact both arguments hold. To prove this thesis, the LDA was computed with and without using virtual training data. Using the respective

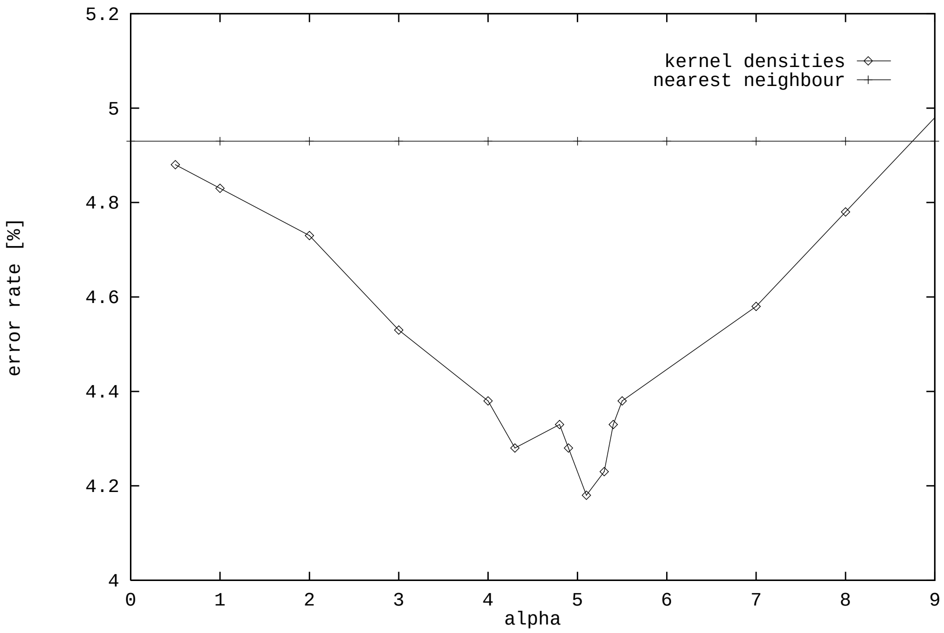


Figure 9.4: Kernel Density error rates on USPS with respect to chosen $\alpha$, compared to a NN-Classifier (using LDA features; NN-error rate is 4.9%).

transformation for feature reduction, the mixture parameters were trained with and without using virtual training data. Remember that using virtual training data (by applying $\pm 1$ pixel shifts to the USPS data), the image size is increased from $16 \times 16$ to $18 \times 18$ pixels.

Finally, some experiments using the discriminative MMI training criterion were conducted (cp. Chapter 4.2.3). The 9-1 mixture density results of this experiment are shown in Table 9.5. As can be seen, discriminative training does not yield better results as compared to maximum likelihood training. Yet, it yields considerably better results for small numbers of densities. Thus, in applications where high computational complexity is a problem – for example in industrial applications – application of discriminative training procedures

Table 9.4: Influence of virtual training data (VTD) with respect to parameter estimation and the estimation of the linear discriminant analysis

| | | Parameter Estimation | |
|---|---|---|---|
| | | no VTD (16x16) | VTD (16x16) |
| LDA Computation, 16x16 | no VTD | 6.7 | 6.3 |
| | VTD | 5.4 | 5.0 |
| | | no VTD (18x18) | VTD (18x18) |
| LDA Computation, 18x18 | no VTD | 7.4 | 5.5 |
| | VTD | 5.9 | 4.5 |

Table 9.5: Comparison of ML/ MMI (h=5, 50 iterations) results for global variance pooling with respect to total number of component densities used

| #component densities | ML Error Rate [%] | | MMI Error Rate [%] | |
|---|---|---|---|---|
| | Train | Test | Train | Test |
| 10 | 17.0 | 13.1 | 11.4 | 10.2 |
| 20 | 13.1 | 12.0 | 6.4 | 8.1 |
| 40 | 10.3 | 9.9 | 3.9 | 6.8 |
| 80 | 8.2 | 9.2 | 2.2 | 5.8 |
| 160 | 6.4 | 8.5 | 1.2 | 6.3 |
| 320 | 4.6 | 6.8 | 0.34 | 5.9 |
| 640 | 3.3 | 6.2 | 0.02 | 5.7 |
| 1280 | 2.2 | 5.6 | 0.02 | 5.4 |
| 4965 | 0.66 | 5.2 | 0.01 | 4.7 |
| 8266 | 0.38 | 4.5 | 0.01 | 4.5 |
| 10360 | 0.38 | 4.6 | 0.01 | 4.6 |

might be interesting. Note that the result obtained in image object recognition support the experiences gained in speech recognition [Dahmen$^+$ 1999, Schlüter & Macherey 1998].

#### 9.1.2.2 Incorporating Tangent Distance

In this chapter, experiments incorporating tangent distance in the proposed classifiers are dealt with. In the following – if nothing else is stated – no feature reduction is applied, because tangent distance is defined on images. In a first experiment, the effect of estimating the proposed tangent covariance matrix (cp. Chapter 7.1.2) is investigated. Interestingly, by simply computing the covariance matrix with respect to all possible tangent approximations of the training data, the error rate can be significantly reduced from 6.0% to 4.3%. A comparison of both variance models can be found in Figure 9.5. Apparently, computing tangent variances in combination with explicitly creating virtual training data is a good means to overcome the difficulties in estimating a covariance matrix in a high dimensional feature space.

In another experiment, the Mahalanobis distance used in the Gaussian component densities was replaced by the single sided tangent distance in the recognition step (that is, the Mahalanobis distance between $x$ and $\mu$ was replaced by the Mahalanobis distance between the respective tangent approximations), whereas the training step was still performed using Mahalanobis distance. This further reduced the error rate from 4.3% to 2.9%. The results of these experiments are shown in Table 9.6.

The best result of 2.9% could be further reduced to 2.7% by calculating the double sided tangent distance in recognition (using a total of about 10.000 mixture components, i.e. on average about 1000 per class). Note that no result better than 3.0% error could be obtained without using tangent variances. On the other hand, using a kernel density based

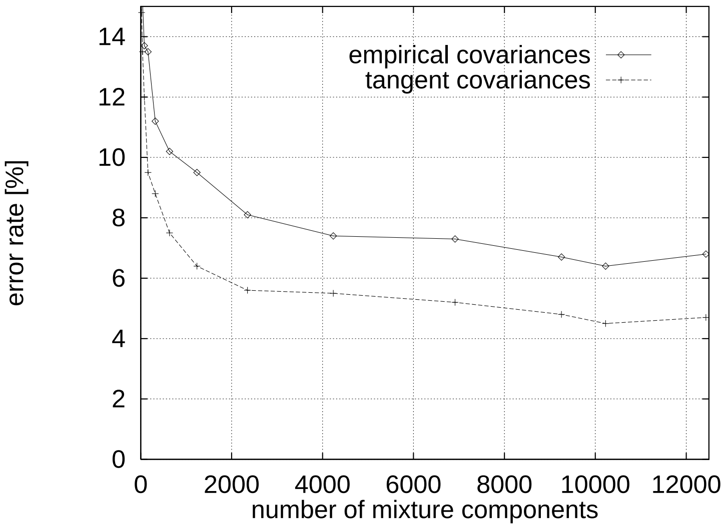


Figure 9.5: Empirical variance vs. tangent variance: error rates with respect to the total number of mixture components used (9-1, no linear discriminant analysis).

Table 9.6: Gaussian mixture densities results on USPS with varying variance estimation and distance measures.

| Method: | Error rate [%] | | | |
|---|---|---|---|---|
| | 1-1 | 1-9 | 9-1 | 9-9 |
| baseline | 8.0 | 6.6 | 6.4 | 6.0 |
| tangent variance, Mahalanobis distance | 6.4 | 4.8 | 4.5 | 4.3 |
| tangent variance, tangent distance | 3.9 | 3.6 | 3.4 | 2.9 |

classifier reduced the error rate to 2.4%. The best result of 2.2% error rate was obtained by combining multiple kernel density based classifiers [Dahmen+ 2000d, Keysers 2000c].

To prove the generalization properties of the presented approach, the best non-bagged kernel density based system was applied to the MNIST task, yielding a state-of-the-art result of 1.0% error rate. Results reported by other groups on that task can be found in Table 2.2.

A comparison of the US Postal Service results obtained throughout this work with that reported by other international research groups can be found in Table 9.7. Note that the result of 2.2% error rate is the best one ever published this particular database, using the original training and testing data (see Table 9.7). The next-best results of 2.5% respectively 2.6% error rate were achieved by by adding about 2,500 machine printed digits to the training set [Drucker+ 1993, Simard+ 1993], therefore they are marked with an asterisk (*). Remember that applying tangent distance at the side of the references can be interpreted as imposing a certain structure on the covariance matrix. Indeed, this 'tangent structure' outperformed the pixel neighbourhood based structures (cp. Chapter 7) in the experiments conducted throughout this work [Dahmen+ 2000e].

In a last experiment, tangent distance was applied to features resulting from a linear discriminant analysis by using the principal components of the class-specific covariance matrices as tangent vectors. This reduced the 1-1 single density error rate from 12.8% to 8.9%, using $L = 7$. Thus, the application of the proposed methods is not restricted to images, there is also hope that they are also suitable in other pattern recognition applications, such as speech recognition (where the tangent vectors have to be learned from the data, as there is no prior knowledge about invariances, as is the case in image processing). In fact, first results that were obtained in automatic speech recognition are promising [Macherey$^+$ 2001].

Experiments on the US Postal data were also performed with Fourier transform based invariants, invariant moments and other features presented in Chapter 1.3.2, yet none of these approaches could improve the best USPS result [Dahmen$^+$ 2000c, Perrey 2000]. Furthermore, using tangent distance also in the training phase yielded no improvement. In comparison to the proposed virtual test sample method, AdaBoost [Freund & Schapire 1996] was used to boost the proposed mixture density based classifier, using features resulting from a linear discriminant analysis. Indeed, AdaBoost reduced the 9-1 error rate from 4.5% to 4.2%, yet the virtual test sample method (reducing the error rate from 4.5% to 3.4%) significantly outperformed this particular boosting method on this particular task.

As for the computational complexity, the standard Gaussian mixture density approach is pretty cheap, requiring less than 0.1 CPU seconds to classify a single pattern (39 LDA features) on a Digital Alpha 500 MHz CPU. Using single sided tangent distance (without feature reduction) takes about 1 CPU second and the computationally expensive double sided tangent distance requires about 50 CPU seconds. Thus, considering error rate versus computational complexity, single sided tangent distance might be considered the best choice for practical applications.

### 9.1.3 Experiments on the IRMA Data

As there are only 1,617 radiographs available in the IRMA database, a *leaving-one-out approach* was adopted here. That is, each image was classified separately, using the remaining 1616 images as references. As already mentioned in Chapter 2.4, the radiographs were scaled down to a standard height of 32 pixels. (Note that this can be done without a significant change in classification error rate, but leads to a considerable system speedup. Performing a 1-nearest neighbour classifier on the radiographs with a squared size of $320 \times 320$ pixels gives a classification error of 18.0%, requiring about 30 CPU seconds on a 500MHz Digital ALPHA CPU to classify a single image. Downscaling the images to a size of $32 \times 32$ pixels, an error rate of 18.1% was obtained, requiring about 0.4 CPU seconds per image [Dahmen$^+$ 2000a].)

To classify the images, the multi-object recognition approach as presented in Chapter 8 was applied, using $M = 1$. Here, the only degree of freedom is a horizontal image shift, as all images have the same height. For the background model, a grayvalue of zero was

Table 9.7: Experimental results reported on the US Postal Service database.

| Author | Method | Error Rate [%] |
|---|---|---|
| Simard+ 1993 | Human Performance | 2.5 |
| Vapnik 1995 | Decision Tree C4.5 | 16.2 |
| Vapnik 1995 | Two-Layer Neural Net | 5.9 |
| Simard+ 1998 | 5-Layer Neural Net | 4.2 |
| Schölkopf 1997 | Support Vectors | 4.0 |
| Schölkopf+ 1998 | Invariant Support Vectors | 3.0 |
| Drucker+ 1993 | Boosted Neural Net | *2.6 |
| Simard+ 1993 | Tangent Distance, 1-Nearest Neighbour | *2.5 |
| This work: | Gaussian Mixtures, VTD, LDA | 4.5 |
| | Gaussian Mixtures, VTD, LDA, VTS | 3.4 |
| | Gaussian Mixtures, VTD, VTS, TD | 2.7 |
| | Kernel Densities, VTD, VTS, TD | 2.4 |
| | Kernel Densities, VTD, VTS, TD, CC | 2.2 |

assumed. Furthermore, a cost term was added depending on the varying image sizes between the observation and the current reference. This was done - for instance - to avoid the matching of a single digit to the spine in a chest radiograph.

The experiments were started by using Mahalanobis distance within a kernel density based classifier (here - due to the high dimensionality of the IRMA data - class specific standard deviations were used instead of diagonal covariance matrices), resulting in an error rate of 14.0%. Using single-sided tangent distance for recognition (on the side of the references), this error rate could be reduced to 13.3%. Interestingly, using the image distortion model with a region size $r = 1$ significantly outperformed tangent distance on this particular dataset, yielding an error rate of 12.1%. In another experiment, it was investigated on the question whether the improvements of tangent distance and the image distortion model are additive. This assumption sounds reasonable, as tangent distance compensates for globale image transformations, whereas the image distortion model deals with local image pertubations. Indeed, using the distorted tangent distance proposed in Chapter 5, the error rate could be further reduced to 10.4%.

In another experiment, the thresholding approach presented in Chapter 5 was applied using $d_{max} = 5000$, in combination with the distance measures discussed above. By doing so, the best error rate could be significantly reduced from 10.4% to 8.2%. Astonishingly, the result of tangent distance in that case is only slightly better than that of Mahalanobis distance (11.1% vs. 11.2%). One thing to be learned from this is that quite probably, using the thresholding approach mimics the behaviour of tangent distance. It should also be noted that in previous experiments all IRMA images were scaled down to a common size of 32×32 pixels prior to classification (more information on that approach is given in [Dahmen+ 2000a]. In these experiments, tangent distance significantly outperformed Mahalanobis distance (with and without the thresholding approach). Thus, it can be

| Distance Measure | Thresholding | |
|---|---|---|
| | no | yes |
| Mahalanobis Distance | 14.0 | 11.2 |
| Tangent Distance | 13.3 | 11.1 |
| Image Distortion Model | 12.1 | 9.0 |
| Distorted Tangent Distance | 10.4 | 8.2 |

Table 9.8: Leaving-one-out IRMA error rates [%] for kernel densities with respect to varying distance measures (with and without thresholding for $d_{max} = 5000$).

assumed that the main effect of tangent distance is the compensation of image shifts (which is now inherent to the classification approach by optimizing over all possible image positions). An overview about the results obtained on the radiograph database is given in Table 9.8.

To make sure that no overfitting occurred in the experiments (due to parameter optimization on the test data), 332 previously unseen radiographs were used as test images and the 1,617 images of the IRMA database as references, using the optimal parameter set determined on the IRMA images. The obtained error rate of 9.0% shows, that the classifier proposed here generalizes very well. Some results reported by other groups on the IRMA data can be found in Table 2.4, proving the obtained result of 8.2% to be excellent.

In a final experiment, the different distance measures discussed above were analysed with respect to their invariance properties, given a transformation $t$. In the experiments, $t$ was chosen to be a translation and the distance between a shifted version of a radiograph and the original image as well as the distances to radiographs from competing classes were computed (in this case, all image were scaled to a common size of 32×32 pixels and new pixels "shifting into" an image were set to graylevel zero). As can be seen in Figure 9.6, the Euclidean distance is highly sensitive to image translations. On the other hand, the tangent distance can nearly compensate one pixel shifts and yields small distances up to 2-3 pixels shifts (cp. Figure 9.7). As expected, the distortion model with $r = 1$ (as shown in Figure 9.8) can fully compensate one pixel shifts, yet with $r$ increasing, the distances to competing classes get smaller rapidly (see Figures 9.8 and 9.9). Thus, large neighbourhoods may lead to decreasing recognition accuracy.

### 9.1.4 Experiments on the Red Blood Cell Data

For the red blood cell experiments, 288 rotation, scale and translation invariant features were extracted using the Fourier Mellin transform as described in Chapter 1.3.2 [Dahmen$^+$ 2000c]. Using these features, a mixture density error rate of 18.8% was obtained (being the average error rate of the ten subsets regarded). This error rate could be further reduced to 15.3% by reducing this 288 dimensional feature space to 47 dimensions by applying a linear discriminant analysis (to previously trained red blood cell pseudo classes). Finally, by using a simple reject rule (the likelihood of the 'best' class must be at least 20% better than that of the second best), the error rate could be reduced to 13.6% at 2.4% reject, with the subset error rates ranging from 10.7% to 16.1%.

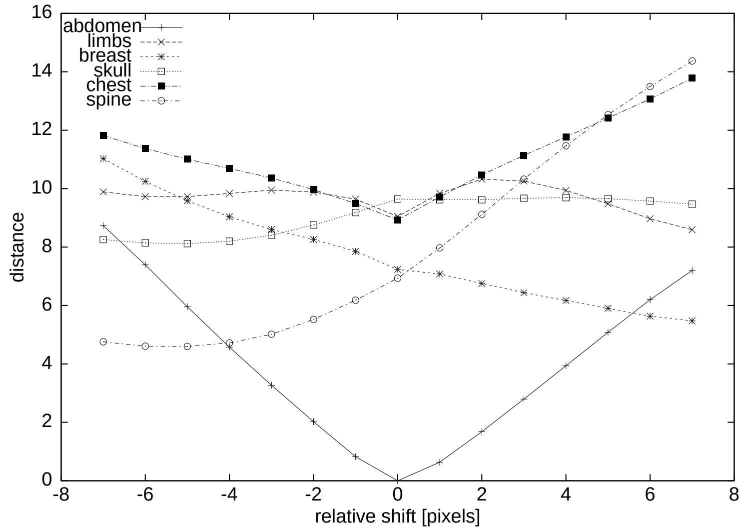


Figure 9.6: Behaviour of Euclidean distance with respect to image shifts.

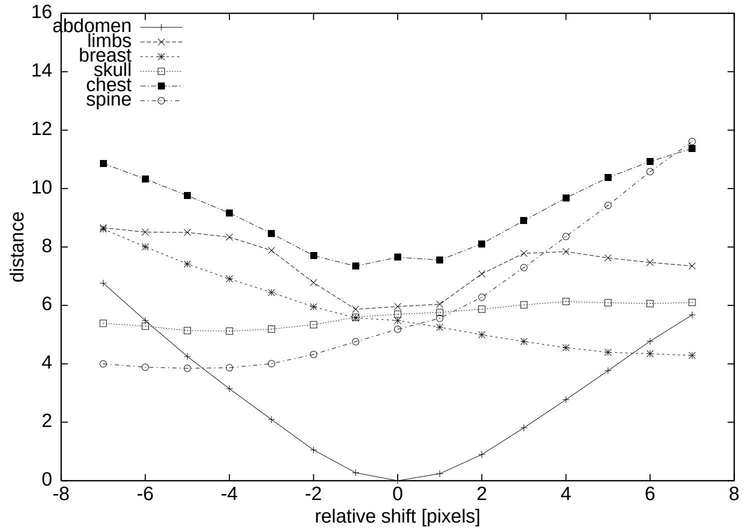


Figure 9.7: Behaviour of tangent distance with respect to image shifts.

Note that a single view of a red blood cell often provides only poor information for classification (e.g. in many cases, stomatocytes and discocytes are hard to distinguish when viewed from above). Therefore, it seems necessary to classify image sequences rather than single images to significantly reduce this error rate. On the other hand, the obtained error rate is significantly lower than the reported human error rate of 20.0% [Fischer 1999]. Thus, the RBC task is an example for a practical application of the methods presented in this work.

## 9.2 Towards Complex Object Recognition

In the following, some experiments towards complex object recognition are described, among them the localisation of an object in a scene (with unknown position and/ or other transformation parameters such as scale) and the detection of multiple objects.

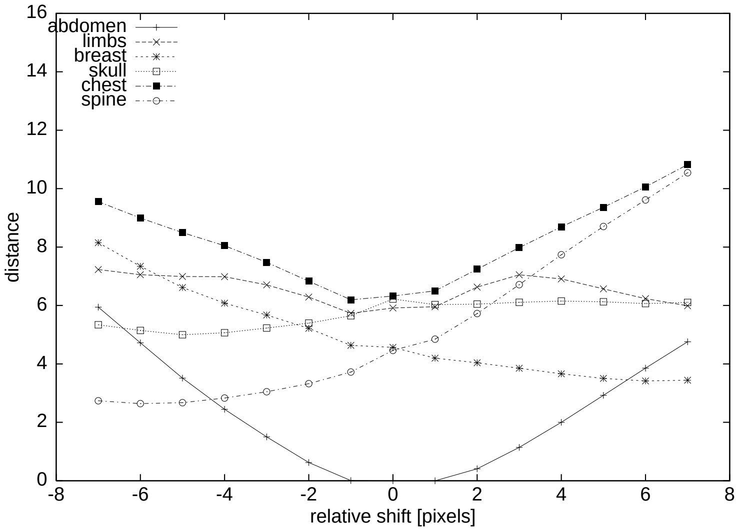


Figure 9.8: Behaviour of the image distortion model with respect to image shifts, using a neighbourhood with $r = 1$.

The experiments were conducted on the COIL-20 database and on modified versions of the US Postal Service database.

### 9.2.1 Experiments on COIL-20

For the experiments on the COIL-20 dataset the sliding window approach was used. Thus, the choice of the following parameters is crucial:

- The size of the sliding window. In the experiments, this parameter was chosen to match the size of the references. The size of the objects to be found is assumed to vary between the size of the reference images and the size of the given scene.

- The number of scale levels used in the multiscale approach. In the experiments, different numbers of scale levels were used. Basically – as one would expect – increasing the number of scale levels improves recognition accuracy at the cost of increased computational complexity [Güld 2000].

Furthermore, thresholds for the rejectors have to be defined, where rejectors based on the entropy and the grayvalue sum of the reference images were used. To speed up the recognition process, the reference images were scaled down from 128×128 pixels to 24×24 pixels. As already mentioned in Chapter 2, only images with odd rotation angle were used as references and only images with even rotation angle as test scenes. It should be noted that by doing so, an observation to be classified always differs from the optimal reference by five degrees. This is contrary to the experiments conducted by Murase & Nayar, where the test scenes (which are unavailable) differed by 2.5 degrees in the worst case [Murase & Nayar 1995]. Thus, the experiments conducted throughout this work can be regarded to be a harder classification task.

Using the COIL-20 data as described above, an error rate of 0% was obtained using a sliding window size of 24×24 pixels, 40 scale levels for the multiscale approach and allowing

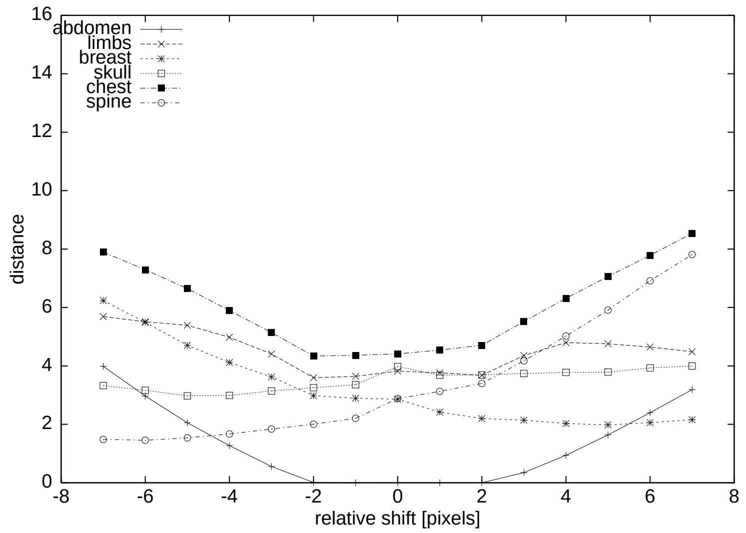


Figure 9.9: Behaviour of the image distortion model with respect to image shifts, using a neighbourhood with $r = 2$.

10% tolerance for the rejectors. It should be noted that other research groups split the COIL-20 "processed" data into a training and a test set (i.e. [Baker$^+$ 1996, Schiele 1997]). Using this splitting, the problem can be treated as an object recognition problem as the US Postal Service database (as training and test images are of the same size) and even a simple 1-nearest neighbour classifier yields an error rate of 0%, using for instance image sizes of 16×16, 24×24 or 32×32 pixels.

### 9.2.2 Experiments on USPS

For the further experiments, modifications of the US Postal Service database were used. In the starting experiments, the first 100 test images were randomly positioned within a 96×96 pixels sized scene (using black background). Furthermore, the images were randomly scaled to a size between 16×16 and 32×32 pixels. For recognition, the sliding window approach with 10 scale levels was applied, with the baseline error rate for the first 100 images being 6.0%. Despite the additional problems (unknown position and unknown scale of the digits), the same error rate was obtained, using the handicap and the rejectors as proposed in Chapter 8. Note that by simply relying on local decisions, the error rate significantly deteriorates to 73% (due to problems as for instance depicted in Figure 8.4).

For the further experiments, multiple US Postal Service digits were randomly positioned within a 96×96 pixels sized scene, where the scale of the digits was again varied between 16×16 and 32×32 pixels. In this case, because of the presence of multiple objects, a local handicap was applied in the sliding window approach. Because of that, for a reliable digit detection, different objects must not be close to each other. Multi-object detection was now done by repeatly applying the single object recognition (again using 10 scale levels), as described in Chapter 8. A resulting example detection is shown Figure 9.10. Apart from the drawback that digits must not be close to each other, this rather simple

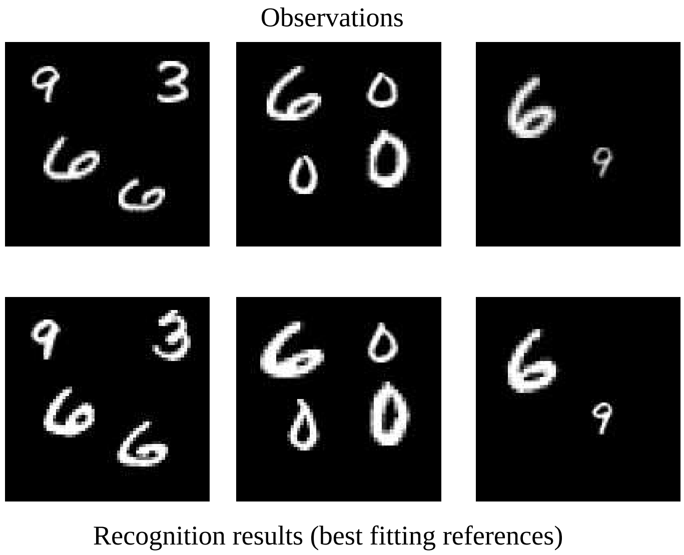


Figure 9.10: Examples for multi-object recognition using the sliding window approach.

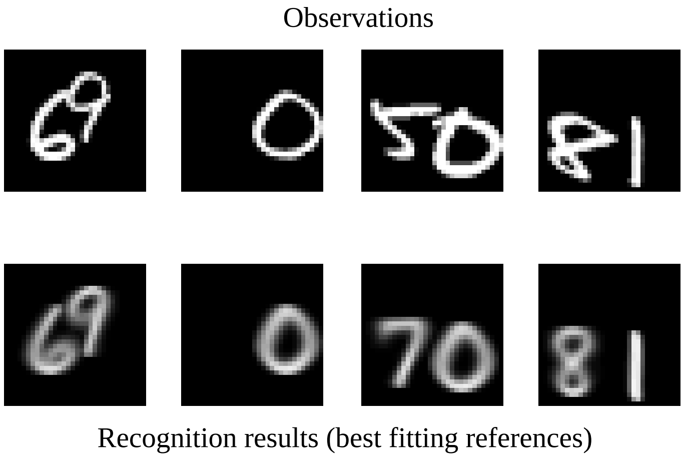


Figure 9.11: Examples for the real multi-object recognition approach.

approach yields surprisingly good results.

To overcome the drawback that the objects must not be close to each other, a final key experiment was conducted using the real multi-object recognition approach as described in Chapter 8. Here, the original US Postal Service digits were randomly placed in a $32\times32$ pixels sized scene (with no scale variations), allowing an overlap of up to 1/3 of the image (in overlapping regions, the maximum grayvalue was used). In the experiments, the background model was assumed to be a Gaussian distribution with zero mean and the prior probabilities $p(k_1^M)$ were assumed to be uniformly distributed. Furthermore, to speed up recognition, a single density model was used. Examples for the resulting recognitions are shown in Figure 9.11. Note that now touching digits can be handled, too (at the cost of increased computational complexity).

# Chapter 10

# Main Contributions

The aim of this work was to implement a statistical classifier based on the use of Gaussian mixture densities, which obtains state-of-the-art error rates on well known standard corpora such as the US Postal Service handwritten digits corpus. In particular, the following achievements were obtained:

I. *Invariant Statistical Classifier & State-of-the-Art Results:*
The experiments conducted throughoutthis work state the thesis, that using a statistical approach for image object recognition is feasible. State-of-the-art results were obtained by using Gaussian mixture densities in the context of the Bayesian decision rule, using Fisher's linear discriminant analysis for feature reduction. As could be expected, these results could be further improved by taking into consideration the special properties of image data, especially the incorporation of invariances. Note that the obtained error rate of 2.2% on the US Postal Service database is the best result ever published on this partucular dataset. An overview of the error rates obtained on the different databases (in comparison to the best result reported by other groups) is given in Table 10.1. More details on that topic can be found in Chapter 9.

| Database | Error Rate [%] | Best reported[%] |
|---|---:|---:|
| US Postal Service | 2.2 | 3.0 |
| MNIST | 1.0 | 0.7 |
| IRMA | 8.2 | 29.0 |
| COIL-20 | 0.0 | 0.0 |
| Red Blood Cells | 13.6 | 20.0 |

Table 10.1: Best error rates obtained throughout this work on various databases in comparison to the best error rates reported by other groups (cp. Chapter 9).

II. *Data Multiplication:*
Throughout this work, the common approach of multiplying the training data has been extended to the test data, resulting in the *virtual test sample method* as presented in Chapter 6. This combined classification approach has some desirable advantages over classical classifier combination schemes. In particular, its theoretical

justification is straightforward (contrary to classifier combination). Furthermore, the virtual test sample method outperformed AdaBoost [Freund & Schapire 1996] (being a classifier combination approach) on the US Postal Service database.

III. *Probabilistic Interpretation and Extension of Tangent Distance:*
Besides extending the invariant distance measure called tangent distance by incorporating a local image distortion model (which proved to be especially effective on the medical IRMA data), a probabilistic interpretation of tangent distance has been given. In particular, it could be shown that computing single sided tangent distance on the side of the references can be interpreted as using a modified covariance matrix within a Gaussian distribution. Furthermore, tangent distance was used to improve parameter estimation, which significantly improved the recognition results obtained on the US Postal Sevice data.

IV. *General Applicability of the Approach & Generalization:*
Finally, it could be shown that the approach is suited for a large number of applications. It not only produced very good results on handwritten digits, but also on the completely different IRMA radiograph and red blood cell databases. Furthermore, first experiments proved, that the approach is also suited for the recognition of more complex scenes, i.e. multi object recognition or the recognition of objects in the presence of inhomogenous background (assuming an appropriate background model). Another important point is the fact that the statistical approach presented here could be shown to generalize well. For instance, using parameters determined on the US Postal database, a surprisingly good MINST error rate of 1.0% was obtained. More information on that topic is given in Chapter 9.

# Chapter 11

# Outlook

For future works, two possible main tasks can be distinguished:

a) Further improvements of the single object recognition approach (being the main topic of this thesis) or

b) concentrating on more complex object recognition tasks such a multi-object recognition (cp. Chapter 8).

Concerning single object recognition (i.e. solving tasks like US Postal Service), the incorporation of invariances into the statistical clasifier could be further improved. For instance, it should be investigated on whether the linear approximation of the manifolds arising in the context of tangent distance can be avoided. At least in the case of image shifts, an exact representation of the manifold seems possible and it should be examined whether the additional computational complexity pays off in terms of recognition accuracy.

Further possible improvements include the virtual test sample method. As was shown in Chapter 6, it is possible to incorporate prior knowledge about the transformation parameters into the recognition process. Throughout this work, the prior probabilities of these parameters were assumed to be uniformly distributed. In future works, the respective probabilities should be learned from a development test set. A similar argument holds for the proposed image distortion model: Here, the function which assigns a cost to each local transformation considered was chosen to be a weighted Euclidean distance between the source and the target pixel. Future experiments should investigate on the question, whether learning this cost function from the training data yields even better results. For instance, meaningful transformations could be performed on the training data, where a transformation that occurs often in the training phase has low costs in recognition.

Finally, if fast recognition algorithms are needed, further investigations should be performed concerning the use of discriminative training criteria, which considerably improved recognition results for a small number of model parameters throughout this work. Among the numerous approaches, discriminative splitting of mixture componenents should be examined [Schlüter 2000].

As for multi-object recognition, the development of suitable background models can be regarded as one of the key issues in order to successfully apply the proposed algorithms to real-world data. Again, experiences gained from speech recognition can be used to learn such a model, being aware of the fact that in this case a 2-dimensional problem is to be dealt with. First approaches towards a statistical background model are given in [Pösl$^+$ 1998].

Furthermore, the multi-object recognition experiments conducted throughout this work showed that the statistical approach yields very promising results. Yet, its rather high computational complexity rises the need for efficient pruning techniques (similar to those being developed in speech recognition) to speed up the recognition process.

Finally, it should be noted that it is hard to compare different multi-object recognition systems in terms of error rate. Apart from the fact that the definition of an error in such an application is highly problem-specific (and subjective), there - to the knowledge of the author - exist no standard databases for this particular problem and thus no results from competing groups which would allow for a meaningful comparison of different approaches. Thus, the creation of such a standardized database seems necessary.

# Appendix A

# List of Abbreviations

| | |
|---|---|
| ANN | Artificial Neural Net |
| CC | Classifier Combination |
| CID | Chair Image Database |
| COIL-20 | Columbia University Object Image library |
| GMD | Gaussian Mixture Density |
| IDM | Image Distortion Model |
| KD | Kernel Density |
| L-1-O | Leaving-One-Out |
| IRMA | Image Retrieval in Medical Applications |
| LDA | Linear Discriminant Analysis |
| ML | Maximum Likelihood |
| MMI | Maximum Mutual Information |
| MNIST | Modified National Institute of Standards and Technology (handwritten digits) database |
| NN | Nearest Neighbour |
| PCA | Principal Components Analysis |
| RBC | Red Blood Cell |
| SVM | Support Vector Machine |
| TD | Tangent Distance |
| USPS | US Postal Service (handwritten digits) database |
| VTD | Virtual Training Data |
| VTS | Virtual Test Sample method |

# Appendix B

# Calculations

Here, calculations omitted in Chapter 7 are given, assuming $\mu_l^T \Sigma^{-1} \mu_{l'} = \delta_{l,l'}$,where $\delta_{l,l'}$ is the Kronecker delta, which is equal to one for $l = l'$ and zero otherwise.

## B.1 Detailed Calculations I

Here, the integration of $p(x, \alpha|\mu)$ is performed (without using maximum approximation)

$$\begin{aligned}
p(x|\mu) &= \int p(x, \alpha|\mu)\ d\alpha \\
&= \int p(\alpha|\mu)\ p(x|\alpha, \mu)\ d\alpha \\
&= \int p(\alpha)\ p(x|\mu(\alpha))\ d\alpha \\
&= \int \frac{1}{\sqrt{2\pi\gamma^2}^L} \exp\left(-\frac{1}{2\gamma^2}\sum\nolimits_l \alpha_l^2\right) \cdot \\
&\qquad \frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \exp\left(-\frac{1}{2}(\mu + \sum\nolimits_l \alpha_l\mu_l - x)^T\Sigma^{-1}(\mu + \sum\nolimits_l \alpha_l\mu_l - x)\right)\ d\alpha \\
&= \frac{1}{\sqrt{2\pi\gamma^2}^L}\frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \cdot \\
&\quad \int \exp\left(-\frac{1}{2}\left(\frac{1}{\gamma^2}\sum\nolimits_l \alpha_l^2 + (\mu + \sum\nolimits_l \alpha_l\mu_l - x)^T\Sigma^{-1}(\mu + \sum\nolimits_l \alpha_l\mu_l - x)\right)\right)\ d\alpha \\
&= \frac{1}{\sqrt{2\pi\gamma^2}^L}\frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \cdot \\
&\quad \int \exp\left(-\frac{1}{2}\left(\frac{1}{\gamma^2}\sum\nolimits_l \alpha_l^2 + (\mu - x)^T\Sigma^{-1}(\mu - x) + (\mu - x)^T\Sigma^{-1}(\sum\nolimits_l \alpha_l\mu_l)\right.\right. \\
&\qquad \left.\left. +(\sum\nolimits_l \alpha_l\mu_l)^T\Sigma^{-1}(\mu - x) + (\sum\nolimits_l \alpha_l\mu_l)^T\Sigma^{-1}(\sum\nolimits_l \alpha_l\mu_l)\right)\right)\ d\alpha \\
&= \frac{1}{\sqrt{2\pi\gamma^2}^L}\frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \exp\left(-\frac{1}{2}\left((\mu - x)^T\Sigma^{-1}(\mu - x)\right)\right) \cdot \\
&\quad \int \exp\left(-\frac{1}{2}\left(\sum\nolimits_l \alpha_l^2(\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l) + 2(\mu - x)^T\Sigma^{-1}(\sum\nolimits_l \alpha_l\mu_l)\right)\right)\ d\alpha
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{\sqrt{2\pi\gamma^2}^L} \frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \exp\left(-\frac{1}{2}\left((\mu-x)^T\Sigma^{-1}(\mu-x)\right)\right) \cdot \\
&\quad \int \exp\left(-\frac{1}{2}\left(\sum_l (\frac{1}{\gamma^2}+\mu_l^T\Sigma^{-1}\mu_l)\left(\alpha_l + \frac{(\mu-x)^T\Sigma^{-1}\mu_l}{(\frac{1}{\gamma^2}+\mu_l^T\Sigma^{-1}\mu_l)}\right)^2\right.\right. \\
&\qquad \left.\left. - \sum_l \frac{((\mu-x)^T\Sigma^{-1}\mu_l)^2}{(\frac{1}{\gamma^2}+\mu_l^T\Sigma^{-1}\mu_l)}\right)\right) d\alpha \\
&= \frac{1}{\sqrt{2\pi\gamma^2}^L} \frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \exp\left(-\frac{1}{2}\left((\mu-x)^T\Sigma^{-1}(\mu-x) - \sum_l \frac{((\mu-x)^T\Sigma^{-1}\mu_l)^2}{(\frac{1}{\gamma^2}+\mu_l^T\Sigma^{-1}\mu_l)}\right)\right) \cdot \\
&\quad \int \exp\left(-\frac{1}{2}\left(\sum_l (\frac{1}{\gamma^2}+\mu_l^T\Sigma^{-1}\mu_l)\left(\alpha_l + \frac{(\mu-x)^T\Sigma^{-1}\mu_l}{(\frac{1}{\gamma^2}+\mu_l^T\Sigma^{-1}\mu_l)}\right)^2\right)\right) d\alpha \\
&= \frac{1}{\sqrt{2\pi\gamma^2}^L} \frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \exp\left(-\frac{1}{2}\left((\mu-x)^T\Sigma^{-1}(\mu-x) - \sum_l \frac{((\mu-x)^T\Sigma^{-1}\mu_l)^2}{(\frac{1}{\gamma^2}+\mu_l^T\Sigma^{-1}\mu_l)}\right)\right) \cdot \\
&\quad \left(\prod_l \sqrt{2\pi}(\frac{1}{\gamma^2}+\mu_l^T\Sigma^{-1}\mu_l)^{-\frac{1}{2}}\right) \int \prod_l \mathcal{N}(\alpha_l| - \frac{(\mu-x)^T\Sigma^{-1}\mu_l}{(\frac{1}{\gamma^2}+\mu_l^T\Sigma^{-1}\mu_l)}, (\frac{1}{\gamma^2}+\mu_l^T\Sigma^{-1}\mu_l)^{-1})\, d\alpha \\
&= \frac{1}{\sqrt{2\pi\gamma^2}^L} \frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \exp\left(-\frac{1}{2}\left((\mu-x)^T\Sigma^{-1}(\mu-x) - \sum_l \frac{((\mu-x)^T\Sigma^{-1}\mu_l)^2}{(\frac{1}{\gamma^2}+\mu_l^T\Sigma^{-1}\mu_l)}\right)\right) \cdot \\
&\quad \left(\prod_l \sqrt{2\pi}(\frac{1}{\gamma^2}+\mu_l^T\Sigma^{-1}\mu_l)^{-\frac{1}{2}}\right) \\
p(x|\mu) &= \left(\prod_l \left(1+\gamma^2\mu_l^T\Sigma^{-1}\mu_l\right)^{-\frac{1}{2}}\right) \cdot \left((2\pi)^D \cdot |\Sigma|\right)^{-\frac{1}{2}} \cdot \\
&\quad \exp\left[-\frac{1}{2}\left((\mu-x)^T\Sigma^{-1}(\mu-x) - \sum_l \frac{((\mu-x)^T\Sigma^{-1}\mu_l)^2}{(\frac{1}{\gamma^2}+\mu_l^T\Sigma^{-1}\mu_l)}\right)\right]
\end{aligned}
$$

## B.2 Detailed Calculations II

Here, it will be shown that the inverse of

$$\left(\Sigma^{-1} - \sum_{l=1}^{L} \frac{(\mu_l^T\Sigma^{-1})^T(\mu_l^T\Sigma^{-1})}{\frac{1}{\gamma^2}+\mu_l^T\Sigma^{-1}\mu_l}\right)$$

is

$$\left(\Sigma + \sum_{l=1}^{L} \gamma^2\mu_l\mu_l^T\right)$$

(by showing that the product equals the matrix of identity $I$).

$$
\begin{aligned}
&\left(\Sigma^{-1} - \sum_{l=1}^{L} \frac{(\mu_l^T\Sigma^{-1})^T(\mu_l^T\Sigma^{-1})}{\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l}\right)\left(\Sigma + \sum_{l=1}^{L}\gamma^2\mu_l\mu_l^T\right) \\
= \;& I - \sum_{l=1}^{L} \frac{(\mu_l^T\Sigma^{-1})^T(\mu_l^T\Sigma^{-1})}{\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l}\Sigma + \sum_{l=1}^{L}\Sigma^{-1}\gamma^2\mu_l\mu_l^T \\
& -\sum_{l=1}^{L}\sum_{l'=1}^{L} \frac{(\mu_l^T\Sigma^{-1})^T\;\overbrace{(\mu_l^T\Sigma^{-1})\;\cdot\;\mu_{l'}}^{=0 \text{ for } l\neq l'}\;\mu_{l'}^T\cdot\gamma^2}{\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l} \\
= \;& I - \sum_{l=1}^{L} \frac{\Sigma^{-1}\mu_l\mu_l^T}{\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l} + \sum_{l=1}^{L}\gamma^2\Sigma^{-1}\mu_l\mu_l^T - \sum_{l=1}^{L}\gamma^2\,\frac{\Sigma^{-1}\mu_l\mu_l^T(\mu_l^T\Sigma^{-1}\mu_l)}{\frac{1}{\gamma^2} + (\mu_l^T\Sigma^{-1}\mu_l)} \\
= \;& I + \sum_{l=1}^{L} \frac{\overbrace{\left[(-1) + 1 + \gamma^2\;\mu_l^T\Sigma^{-1}\mu_l - \gamma^2\;\mu_l^T\Sigma^{-1}\mu_l\right]}^{=0}\cdot\Sigma^{-1}\mu_l\mu_l^T}{\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l} \\
= \;& I
\end{aligned}
\tag{B.1}
$$

## B.3 Detailed Calculations III

Here it will be shown, that

$$
|\Sigma + \sum\nolimits_l \gamma^2\mu_l\mu_l^T| = |\Sigma|\cdot\prod\nolimits_l\left(1+\gamma^2\mu_l^T\Sigma^{-1}\mu_l\right). \tag{B.2}
$$

The proof is given for $L = 1$ (the case $L > 1$ immediately follows by induction). Furthermore, for ease of notation, $b := \gamma\cdot\mu$.

$$
\begin{aligned}
|\Sigma + bb^T| &= |\Sigma^{\frac{1}{2}}|\cdot|I + \Sigma^{-\frac{1}{2}}bb^T\Sigma^{-\frac{1}{2}}|\cdot|\Sigma^{\frac{1}{2}}| \\
&= |\Sigma|\cdot|I + (\Sigma^{-\frac{1}{2}}b)(\Sigma^{-\frac{1}{2}}b)^T| \\
&= |\Sigma|\cdot|I + aa^T|, \text{ where } a = \Sigma^{-\frac{1}{2}}b
\end{aligned}
$$

Now, an orthonormal matrix $U = (\frac{1}{|a|}\cdot a, *, *, ..., *)$ is chosen (with $a, *, ..., * \in \mathbb{R}^D$), which is always possible. Note that for orthonormal matrices $U^T = U^{-1}$ holds.
Thus:

$$
U^Ta = \begin{pmatrix} |a| \\ 0 \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix} = |a|\cdot\begin{pmatrix} 1 \\ 0 \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix}
$$

and therefore

$$(U^Ta)(U^Ta)^T = |a|^2 \cdot \begin{pmatrix} 1 & 0 & . & . & . & 0 \\ 0 & 0 & . & . & . & 0 \\ . & . & . & . & . & . \\ . & . & . & 0 & . & . \\ . & . & . & . & . & . \\ 0 & . & . & . & . & . \end{pmatrix} =: M$$

Now:

$$\begin{aligned} |(I + aa^T)| &= \underbrace{|U^T|}_{=1} \cdot |(I + aa^T)| \cdot \underbrace{|U|}_{=1} \\ &= |(\underbrace{U^TU}_{=I} + U^Taa^TU)| \\ &= |I + U^Ta(U^Ta)^T| \\ &= |I + M| \\ &= 1 + |a|^2 \\ &= 1 + a^Ta \\ &= 1 + (\Sigma^{-\frac{1}{2}}b)^T(\Sigma^{-\frac{1}{2}}b) \\ &= 1 + b^t\Sigma^{-1}b \\ &= 1 + \gamma^2\mu^T\Sigma^{-1}\mu \end{aligned}$$

Finally:

$$|\Sigma + \gamma^2\mu\mu^T| = |\Sigma + bb^T| = |\Sigma| \cdot |I + aa^T| = |\Sigma| \cdot \left(1 + \gamma^2\mu^T\Sigma^{-1}\mu\right)$$

## B.4 Detailed Calculations IV

$$\begin{aligned} p(x|\alpha, \mu, \Sigma) &= \mathcal{N}(x|M_\alpha^{-1}\ \mu, M_\alpha^{-1}\ \Sigma\ M_\alpha^{-1^T}) \\ &= \mathcal{N}(x|\mu', \Sigma') \\ &= \frac{1}{\sqrt{\det(2\pi\Sigma')}} \exp\left(-\frac{1}{2}(x-\mu')^T\Sigma'^{-1}(x-\mu')\right) \end{aligned} \tag{B.3}$$

Now, since the covariance matrix $\Sigma' = M_\alpha^{-1}\ \Sigma\ M_\alpha^{-1^T}$ depends on $\alpha$, the solution of the integral resulting from Equation (7.1) is far more difficult and so far unknown.
Yet, by assuming that $\Sigma' = \Sigma$ for the moment, similar considerations as in the case of variations of the references hold and one obtains (in the necessary calculations - given in detail for the references in Appendix B.1 - the term "$+\sum_{l=1}^{L} \alpha_l\ \mu_l$" is replaced by "$-\sum_{l=1}^{L} \alpha_l\ x_l$" and the negation cancels out in all occurences):

$$
\begin{aligned}
p(x|\mu,\Sigma) \quad = \quad & \left(\prod_{l=1}^{L}\left(1+\gamma^2 x_l^T\Sigma^{-1}x_l\right)^{-\frac{1}{2}}\right)\cdot\det(2\pi\Sigma)^{-\frac{1}{2}}\cdot \\
& \exp\left[-\frac{1}{2}\left((\mu-x)^T\underbrace{(\Sigma^{-1}-\sum_{l=1}^{L}\frac{(x_l^T\Sigma^{-1})^T(x_l^T\Sigma^{-1})}{\frac{1}{\gamma^2}+x_l^T\Sigma^{-1}x_l})}_{(**)}(\mu-x)\right)\right] \qquad \text{(B.4)}
\end{aligned}
$$

Note that even in this case, the resulting distribution cannot be regarded as a degenerated Gaussian, as the matrix $(**)$ depends on $x$ (contrary to the matrix $(*)$ in Equation (7.8)).

# Appendix C

# Additional Results

## C.1 Diagonal vs. Full Covariance Matrix

Figure C.1 shows results for a diagonal covariance in comparison to a full covariance matrix with respect to the total number of densities used. Because there is no significant difference in recognition results and full covariance matrices are compuationally much more expensive, diagonal covariance matrices are used throughout this work.

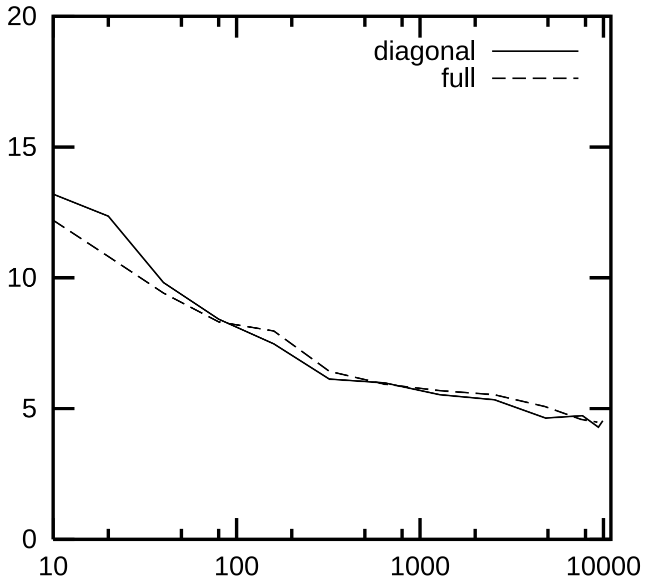


Figure C.1: 9-1 USPS error rates as a function of the number of densities for a globally pooled diagonal/ full covariance matrix.

## C.2 Additional Results using Tangent Distance

In Table C.1, some results on tangent distance are given for 1-1 USPS data. As can be seen, there is no significant difference between applying the tangents on the side of the refercences respectively the observations. Thus, because applying tangent distance on the

Table C.1: 1-1 results on USPS for different tangent distance settings, using kernel densities.

| Method | Tangents on the Side of | Error Rate [%] |
|---|---|---|
| derivative | reference | 3.7 |
| | observation | 3.3 |
| | both (double-sided TD) | 3.0 |
| estimate | reference | 3.8 |

side of the references is computationally cheaper in real applications, this method is used throughout this work. Here, “derivative” means that the tangents were computed using image gradients, whereas “estimate” refers to tangents learned from the data itself (cp. Chapter 7.1.3).

# Bibliography

[Baker$^+$ 1996] S. Baker, S. Nayar, "Pattern Rejection", *International Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, pp. 544-549, June 1996.

[Baum$^+$ 1967] L. Baum, J. Eagon, "An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model for Ecology," *Bulletin of the American Mathematical Society*, Vol. 73, pp. 360-363, 1967.

[Blanz$^+$ 1996] V. Blanz, B. Schölkopf, H. Bülthoff, C. Burges, V. Vapnik, T. Vetter, "Comparison of View–Based Object Recognition Algorithms using Realistic 3D Models", C. von der Malsburg, W. von Seelen, J. Vorbrüggen, B. Sendhoff (eds.): *International Conference on Neural Networks - ICANN'96*, Lecture Notes in Computer Science, Vol. 1112, Springer, Berlin, pp. 251–256, 1996.

[Bottou$^+$ 1994] L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. Jackel, Y. Le Cun, U. Muller, E. Sackinger, P. Simard, V. Vapnik, "Comparison of Classifier Methods: A Case Study in Handwritten Digit Recognition", *12th International Conference on Pattern Recognition*, Jerusalem, Israel, Vol. 2, pp. 77-82, October 1994.

[Bredno$^+$ 1999] J. Bredno, F. Vogelsang, J. Dahmen, T. Lehmann, M. Kilbinger, B. Wein, R. Günther, H. Ney, K. Spitzer, "Eine Entwicklungsumgebung für die interdisziplinäre Zusammenarbeit im Image-Retrieval-Projekt IRMA", *Bildverarbeitung für die Medizin 1999*, Heidelberg, pp. 362-366, March 1999 (in German).

[Bredno$^+$ 2000a] J. Bredno, S. Brandt, J. Dahmen, B. Wein, T. Lehmann, "Kategorisierung von Röntgenbildern mit aktiven Konturmodellen", *Bildverarbeitung für die Medizin 2000*, Munich, pp. 356–360, March 2000 (in German).

[Bredno$^+$ 2000b] J. Bredno, M. Kohnen, J. Dahmen, F. Vogelsang, B. Wein, T. Lehmann, "Synergetic Impact obtained by a Distributed Developing Platform for Image Retrieval in Medical Applications (IRMA)", *Proceedings of the SPIE*, Vol. 3972(33), pp. 321-331, February 2000.

[Breiman 1994] L. Breiman, "Bagging Predictors", Technical Report 421, Department of Statistics, University of California at Berkeley, September 1994.

[Burges 1996] C. Burges, "Simplified Support Vector Decision Rules", *13th International Conference on Machine Learning*, Bari, Italy, pp. 71-78, July 1996.

[Burkhardt+ 92] H. Burkhardt, A. Fenske, H. Schulz-Mirbach, “Invariants for the Recognition of Planar Contour and Gray-Scale Images”, Technical Report TR-402-92-003, Technische Informatik I, TU Hamburg, 1992.

[Cortes+ 1995] C. Cortes, V. Vapnik, “Support Vector Networks”, *Machine Learning*, Vol. 20, No. 3, pp. 273–297, September 1995.

[Dahmen+ 1997] J. Dahmen, F. Weiler, F. Vogelsang, B. Wein, M. Kilbinger, R. Günther, “Rippenkantendetektion in Thoraxröntgenbildern zur Schattenkompensation und Bildananlyse”, *Bildverarbeitung für die Medizin 1997*, Freiburg, pp. 111-116, March 1997 (in German).

[Dahmen+ 1998a] J. Dahmen, T. Lehmann, K. Spitzer, H. Ney, “Image Retrieval für klinische Bilddatenbanken”, *Bildverarbeitung für die Medizin 1998*, Aachen, pp. 442-446, March 1998 (in German).

[Dahmen+ 1998b] J. Dahmen, K. Beulen, H. Ney, “Objektklassifikation mit Mischverteilungen”, *20th Symposium German Association for Pattern Recognition (DAGM)*, Stuttgart, pp. 167-174, September 1998 (in German).

[Dahmen+ 1999] J. Dahmen, R. Schlüter, H. Ney, ”Discriminative Training of Gaussian Mixtures for Image Object Recognition”, *21st Symposium German Association for Pattern Recognition (DAGM)*, Bonn, pp. 205-212, September 1999.

[Dahmen+ 2000a] J. Dahmen, T. Theiner, D. Keysers, H. Ney, T. Lehmann, B. Wein, ”Classification of Radiographs in the ’Image Retrieval in Medical Applications’ System (IRMA)”, *6th International RIAO Conference on Content-Based Multimedia Information Access*, Paris, France, pp. 551-566, April 2000.

[Dahmen+ 2000b] J. Dahmen, K. Beulen, M. Güld, H. Ney, ”A Mixture Density Based Approach to Object Recognition for Image Retrieval”, *6th International RIAO Conference on Content-Based Multimedia Information Access*, Paris, France, pp. 1632–1647, April 2000.

[Dahmen+ 2000c] J. Dahmen, J. Hektor, R. Perrey, H. Ney, “Automatic Classification of Red Blood Cells using Gaussian Mixture Densities”, *Bildverarbeitung für die Medizin 2000*, Munich, pp. 331-335, March 2000.

[Dahmen+ 2000d] J. Dahmen, D. Keysers, M. O. Güld, H. Ney, “Invariant Image Object Recognition using Mixture Densities”, *15th International Conference on Pattern Recognition*, Barcelona, Spain, Vol. 2, pp. 614–617, September 2000.

[Dahmen+ 2000e] J. Dahmen, D. Keysers, M. Pitz, H. Ney, “Structured Covariance Matrices for Statistical Image Object Recognition”, *22nd Symposium German Association for Pattern Recognition (DAGM)*, Kiel, pp. 99–106, September 2000.

[Dahmen+ 2001a] J. Dahmen, D. Keysers, H. Ney, “Combined Classification of Handwritten Digits using the ‘Virtual Test Sample Method”’, to appear in *2nd International Workshop on Multiple Classifier Systems - MCS 2001*, Cambridge, UK, July 2001.

[Dahmen+ 2001b] J. Dahmen, D. Keysers, H. Ney, M. Güld, "Statistical Image Object Recognition using Mixture Densities", *Journal of Mathematical Imaging and Vision*, Kluwer Academic Publishers, Vol. 14, No. 3, pp. 285-296, May 2001.

[Dahmen+ 2001c] J. Dahmen, D. Keysers, M. Motter, H. Ney, T. Lehmann, B. Wein, "An Automatic Approach to Invariant Radiograph Classification", *Bildverarbeitung für die Medizin 2001*, Lübeck, pp. 337-341, March 2001.

[Dempster+ 1977] A. Dempster, N. Laird, D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society*, 39(B), pp. 1-38, 1977.

[Deuticke+ 1990] B. Deuticke, R. Grebe, C. Haest, "Action of Drugs on the Erythrocyte Membrane", *Blood Cell Biochemistry, Vol. 1: Erythroid Cells*, Plenum Press, New York, NY, pp. 475-529, 1990.

[Devijver & Kittler 1982] P. Devijver, J. Kittler, *Pattern Recognition - A Statistical Approach*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[Devroye+ 1996] L. Devroye, L. Györfi, G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, Springer, New York, NY, 1996.

[Drucker+ 1993] H. Drucker, R. Schapire, P. Simard, "Boosting Performance in Neural Networks", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 7, No. 4, pp. 705–719, 1993.

[Duda & Hart 1973] R. Duda, P. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, NY, 1973.

[Fischer 1999] T. Fischer, Department of Physiology, RWTH Aachen University of Technology, personal communication, 1999.

[Freund & Schapire 1996] Y. Freund, R. Schapire, "Experiments with a New Boosting Algorithm", *13th International Conference on Machine Learning*, Bari, Italy, pp. 148-156, July 1996.

[Fukunaga 1990] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, San Diego, CA, 1990.

[Güld 2000] M. Güld, "Inhaltsbasierter Bildzugriff mittels statistischer Objekterkennung", diploma thesis at the Chair of Computer Science VI, RWTH Aachen University of Technology, July 2000 (in German).

[Hastie+ 1995] T. Hastie, P. Simard, E. Säckinger, "Learning Prototype Models for Tangent Distance", *Advances in Neural Information Processing Systems 7*, MIT Press, Cambridge, MA, pp. 999-1006, 1995.

[Hastie+ 1998] T. Hastie, P. Simard, "Metrics and Models for Handwritten Character Recognition", *Statistical Science*, Vol. 13, No. 1, pp. 54–65, January 1998.

[Hinton+ 1995] G.. Hinton, M. Revow, P. Dayan, "Recognizing Handwritten Digits Using Mixtures of Linear Models", *Advances in Neural Information Processing Systems*, Vol. 7, MIT Press, Cambridge, MA, pp. 1015–1022, 1995.

[Hinton+ 1997] G. Hinton, P. Dayan, M. Revow, "Modeling the Manifolds of Images of Handwritten Digits", *IEEE Transactions on Neural Networks*, Vol. 8, No. 1, pp. 65–74, January 1997.

[Hu 1962] M. Hu, "Visual Pattern Recognition by Moment Invariants", *IEEE Transactions on Information Theory* Vol. 8, pp. 179-187, February 1962.

[Hutten+ 1999] D. Huttenlocher, R. Lilien, C. Olson, "View-Based Recognition Using an Eigenspace Approximation to the Hausdorff Measure", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 9, pp. 951–955, September 1999.

[Keysers 2000a] D. Keysers, "Approaches to Invariant Image Object Recognition", diploma thesis at the Chair of Computer Science VI, RWTH Aachen University of Technology, June 2000.

[Keysers+ 2000b] D. Keysers, J. Dahmen, H. Ney, "A Probabilistic View on Tangent Distance", *22nd Symposium German Association for Pattern Recognition (DAGM)*, Kiel, pp. 107–114, September 2000.

[Keysers 2000c] D. Keysers, J. Dahmen, T. Theiner, H. Ney, "Experiments with an Extended Tangent Distance", *15th International Conference on Pattern Recognition*, Barcelona, Spain, Vol. 2, pp. 38-42, September 2000.

[Kittler+ 1998] J. Kittler, M. Hatef, R. Duin, J. Matas, "On Combining Classifiers", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 3, pp. 226-239, March 1998.

[Kohnen+ 2000a] M. Kohnen, F. Vogelsang, B. Wein, M. Kilbinger, R. Günther, F. Weiler, J. Bredno, J. Dahmen, "Kategorisierung von digitalen Röntgenbildern mit parametrisierbaren Formmodellen", *Bildverarbeitung für die Medizin*, Munich, pp. 366-370, March 2000 (in German).

[Kohnen+ 2000b] M. Kohnen, F. Vogelsang, B. Wein, M. Kilbinger, R. Günther, F. Weiler, J. Bredno, J. Dahmen, "Knowledge Based Automated Feature Extraction to Categorize Secondary Digitized Pictures", *Proceedings of the SPIE*, Vol. 3979(70), pp. 709-717, February 2000.

[Kohnen+ 2001] M. Kohnen, H. Schubert, B. Wein, J. Bredno, T. Lehmann, J. Dahmen, "Qualität von DICOM-Informationen in Bilddaten aus der klinischen Routine", *Bildverarbeitung für die Medizin 2001*, Lübeck, pp. 419-423, March 2001 (in German).

[Künzi 1979] H. Künzi, *Nichtlineare Programmierung*, Springer, Berlin, 1979 (in German).

[Laaksonen 1997] J. Laaksonen, "Local Subspace Classifiers", *International Conference on Neural Networks - ICANN'97*, Lecture Notes in Computer Science, Vol. 1327, Springer, Berlin, pp. 637–642, 1997.

[Lehmann+ 1997] T. Lehmann, W. Oberschelp, E. Pelikan, R. Repges, *Bildverarbeitung für die Medizin: Grundlagen, Modelle, Methoden, Anwendungen*, Springer, Berlin, 1997 (in German).

[Lehmann+ 2000a] T. Lehmann, B. Wein, J. Dahmen, J. Bredno, F. Vogelsang, M. Kohnen, "Content-based Image Retrieval in Medical Applications: A Novel Multi-step Approach", *Proceedings of the SPIE*, Vol. 3972(32), pp. 312–331, February 2000.

[Lehmann+ 2000b] T. Lehmann, B. Wein, J. Dahmen, J. Bredno, F. Vogelsang, M. Kohnen, "Ein strukturiertes Konzept zum inhaltsbasierten Zugriff auf medizinische Bildarchive", *Bildverarbeitung für die Medizin*, München, pp. 218-222, März 2000 (in German).

[Li 1995] S. Li, *Markow Random Field Modelling in Computer Vision*, Springer, Tokyo, 1995.

[Linde+ 1980] Y. Linde, A. Buzo, R. Gray, "An Algorithm for Vector Quantizer Design", *IEEE Transactions on Communications*, Vol. 28, No. 1, pp. 84-95, January 1980.

[Macherey+ 2001] W. Macherey, D. Keysers, J. Dahmen, H. Ney, "Improving Automatic Speech Recognition using Tangent Distance", submitted to *7th European Conference on Speech Communication and Technology*, Aalborg, Denmark, September 2001.

[Martin+ 1998] S. Martin, J. Liermann, H. Ney, "Automatic Bigram and Trigram Clustering for Word Classes", *Speech Communication*, Vol. 24, No. 1, pp. 19-37, April 1998.

[Martin+ 1999] S. Martin, C. Hamacher, J. Liermann, F. Wessel, H. Ney, "Assessment of Smoothing Methods and Complex Stochastic Language Modeling", *6th European Conference on Speech Communication and Technology*, Budapest, Hungary, pp. 1939-1942, September 1999.

[Meinicke+ 1999] P. Meinicke, H. Ritter, "Local PCA Learning with Resolution-Dependent Mixtures of Gaussians", *9th International Conference on Artificial Neural Networks*, Edinburgh, UK, pp. 497–502, September 1999.

[Moghaddam+ 1996] B. Moghaddam, C. Nastar, A. Pentland, "A Bayesian Similarity Measure for Direct Image Matching", *13th International Conference on Pattern Recognition*, Vienna, Austria, Vol. 2, pp. 350–358, August 1996.

[Moghaddam & Pentland 1997] B. Moghaddam, A. Pentland, "Probabilistic Visual Learning for Object Representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 696-710, July 1997.

[Moore 1979] R. Moore, "A Dynamic Programming Algorithm for the Distance Between Two Finite Areas", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1, No. 1, pp. 86–88, January 1979.

[Murase & Nayar 1995] H. Murase, S. Nayar, "Visual Learning and recognition of 3-D Objects from Appearance", *International Journal of Computer Vision*, Vol. 14, pp. 5-24, January 1995.

[Ney 1990] H. Ney, "Acoustic Modelling of Phoneme Units for Continuous Speech Recognition", L. Torres, E. Masgrau, M. Lagunas (eds.): *Signal Processing V: Theories and Applications*, Elsevier Science Publishers B.V., 1990.

[Ney 1995] H. Ney, "On the Probabilistic Interpretation of Neural Network Classifiers and Discriminative Training Criteria", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 2, pp. 107–119, February 1995.

[Ney+ 1998] H. Ney, L. Welling, S. Ortmanns, K. Beulen, F. Wessel, "The RWTH Large Vocabulary Continuous Speech Recognition System", *IEEE International Conference on Acoustics, Speech and Signal Processing*, Seattle, WA, pp. 853-856, May 1998.

[Ney 1999] H. Ney, "Mustererkennung und Neuronale Netze", Script to the lecture on Pattern Recognition and Neural Networks held at RWTH Aachen University of Technology, 1999.

[Ney 2000a] H. Ney, "Mustererkennung und Neuronale Netze", lecture on Pattern Recognition and Neural Networks held at RWTH Aachen University of Technology, 2000/2001.

[Ney 2000b] H. Ney, personal communication, April 2000.

[Och & Ney 2000] F. Och, H. Ney, "Improved Statistical Alignment Models", *38th Annual Meeting of the Association for Computational Linguistics (ACL 2000)*, Hongkong, China, pp. 440-447, October 2000.

[Ortmanns & Ney 2000] S. Ortmanns, H. Ney, "Look-Ahead Techniques for Fast Beam Search", *Computer, Speech and Language*, Vol. 14, No. 1, pp. 15-32, January 2000.

[Perantonis+ 1992] S. Perantonis, P. Lisboa, "Translation, Rotation and Scale Invariant Pattern Recognition by High-Order Neural Networks and Moment Classifiers", *IEEE Transactions on Neural Networks*, Vol. 3, No. 2, pp. 241-251, March 1992.

[Perrey 2000] R. Perrey, "Affin-invariante Merkmale für die 2D-Bilderkennung", diploma thesis at the Chair of Computer Science VI, RWTH Aachen University of Technology, February 2000 (in German).

[Pösl+ 1998] J. Pösl, H. Niemann, "Object Localization with Mixture Densities of Wavelet Features", *International Wavelets Conference*, Tanger, Marokko, April 1998.

[Press+ 1992] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, *Numerical Recipes in C*, University Press, New York, NY, 1992.

[Reddy+ 96] B. Reddy, B. Chatterji, "An FFT-based Technique for Translation, Rotation and Scale invariant Image Registration", *IEEE Transactions on Image Processing*, Vol. 5, No. 8, August 1996.

[Rissanen 1978] J. Rissanen, “Modelling by Shortest Data Description”, *Automatica*, Vol. 14, pp. 465-471, 1978.

[Roberts+ 1998] S. Roberts, D. Husmeier, I. Rezek, W. Penny, “Bayesian Approaches to Gaussian Mixture Modelling”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 11, pp. 1133-1141, September 1998.

[Rojas 1993] R. Rojas, *Theorie der neuronalen Netze - Eine systematische Einführung*, Springer, Berlin, 1993 (in German).

[Schalkoff 1989] R. Schalkoff, *Digital Image Processing and Computer Vision*, John Wiley & Sons, New York, NY, 1989.

[Schenk & Milgram 1995] H. Schwenk, M. Milgram, “Transformation Invariant Autoassociation with Application to Handwritten Character Recognition”, *Advances in Neural Information Processing Systems 7*, MIT Press, Cambridge, MA, pp. 991-998, 1995.

[Schiele & Crowley 1996] B. Schiele, J. Crowley, “Probabilistic Object Recognition using Multidimensional Receptive Field Histograms”, *13th International Conference on Pattern Recognition*, Vienna, Austria, Vol. 2, pp. 50-54, 1996.

[Schiele 1997] B. Schiele, “Object Classification based on Visual Classes”, *19th Symposium German Association for Pattern Recognition (DAGM)*, Braunschweig, pp. 403-410, September 1997.

[Schlüter & Macherey 1998] R. Schlüter, W. Macherey, “Comparison of Discriminative Training Criteria,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Seattle, WA, pp. 493-496, May 1998.

[Schlüter 2000] R. Schlüter, “Investigations on Discriminative Training Criteria”, PhD thesis at the Chair of Computer Science VI, RWTH Aachen University of Technology, 2000.

[Schölkopf+ 1996] B. Schölkopf, C. Burges, V. Vapnik, “Incorporating Invariances in Support Vector Learning Machines”, *International Conference on Neural Networks - ICANN'96*, Lecture Notes in Computer Science, Vol. 1112, Springer, Berlin, pp. 47-52, 1996.

[Schölkopf 1997] B. Schölkopf, *Support Vector Learning*, Oldenbourg Verlag, Munich, 1997.

[Schölkopf+ 1998] B. Schölkopf, P. Simard, A. Smola, V. Vapnik, “Prior Knowledge in Support Vector Kernels,” *Advances in Neural Information Processing Systems 10*, MIT Press, Cambridge, MA, pp. 640-646, 1998.

[Schönfeld+ 1989] M. Schönfeld, R. Grebe, “Automatic Shape Quantification of Freely Suspended Red Blood Cells by Isodensity Contour Tracing and Tangent Counting”, *Computer Methods and Programs in Biomedicine*, Vol. 28, pp. 217-224, 1989.

[Schreckenberg+ 2000] M. Schreckenberg, J. Dahmen, M. Güld, G. Schummers, D. Meyer-Ebrecht, H. Ney, "Automatische Endokarderkennung in 3D mit approximierenden Thin-Plate-Spline Modellen unter Einsatz von Gauss'schen Mischverteilungs-Modellen für die lokale Klassifikation", *Bildverarbeitung für die Medizin 2000*, Munich, pp. 351-355, März 2000 (in German).

[Schulz-Mirbach 1992] H. Schulz-Mirbach, "On the Existence of Complete Invariant Feature Spaces in Pattern Recognition", *11th International Conference on Pattern Recognition*, Den Haag, The Netherlands, Vol. 2, pp. 178–182, August/ September 1992.

[Schulz-Mirbach 1995] H. Schulz-Mirbach, "Invariant Features for Gray-Scale Images", *17th Symposium German Association for Pattern Recognition (DAGM)*, Bielefeld, pp. 1-14, September 1995.

[Siggelkow+ 98] S. Siggelkow, H. Burkhardt, "Image Retrieval Based on Local Invariant Features", *International Conference on Signal and Image Processing*, Las Vegas, NV, Oktober 1998.

[Simard+ 1993] P. Simard, Y. Le Cun, J. Denker, "Efficient Pattern Recognition Using a New Transformation Distance", *Advances in Neural Information Processing Systems 5*, Morgan Kaufmann, San Mateo, CA, pp. 50-58, 1993.

[Simard+ 1998] P. Simard, Y. Le Cun, J. Denker, B. Victorri, "Transformation Invariance in Pattern Recognition — Tangent Distance and Tangent Propagation", *Neural Networks: Tricks of the Trade*, Lecture Notes in Computer Science, Vol. 1524, Springer, Berlin, pp. 239–274, 1998.

[Sixtus+ 2000] A. Sixtus, S. Molau, S. Kanthak, R. Schlüter, Hermann Ney, "Recent Improvements of the RWTH Large Vocabulary Speech Recognition System on Spontaneous Speech" *IEEE International Conference on Acoustics, Speech and Signal Processing*, Istanbul, Turkey, pp. 1671-1674, June 2000.

[Smith+ 1994] S. J. Smith, M. O. Bourgoin, K. Sims, H. L. Voorhees, "Handwritten Character Classification Using Nearest Neighbor in Large Databases", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 9, pp. 915–919, September 1994.

[Smith & Chang 1996] J.R. Smith, S.F. Chang, "Tools and Techniques for Color Image Retrieval", *Proceedings of the SPIE*, Vol. 2670, pp. 426-437, 1996.

[Süße 1999] H. Süße, personal communication at the *21st Symposium German Association for Pattern Recognition (DAGM)*, Bonn, September 1999.

[Theiner 2000] T. Theiner, "Inhaltsbasierter Zugriff auf große Bilddatenbanken", diploma thesis at the Chair of Computer Science VI, RWTH Aachen University of Technology, February 2000 (in German).

[Tipping 2000] M. Tipping, "The Relevance Vector Machine", *Advances in Neural Information Processing Systems 12*, MIT Press, Cambridge, MA, pp. 652-658, 2000.

[Uchida+ 1998] S. Uchida, H. Sakoe, "A Monotonic and Continuous Two-Dimensional Warping Based on Dynamic Programming", *14th International Conference on Pattern Recognition*, Brisbane, Australien, Vol. 2, pp. 521–524, August 1998.

[Vapnik 1995] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, 1995.

[Vapnik 1998] V. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, 1998.

[Vogelsang+ 1997] F. Vogelsang, F. Weiler, B. Wein, M. Kilbinger, R. Günther, "Image Content Analysis using Textural Information and Synergetic Classifiers", *European Congress of Radiology ECR 97*, Vienna, Austria, pp. 291, March 1997.

[Vogelsang+ 1998] F. Vogelsang, F. Weiler, J. Dahmen, M. Kilbinger, B. Wein, R. Günther, "Detection and Compensation of Rib Structures in Chest Radiographs for Diagnose Assistance", *Proceedings of the SPIE*, Vol. 3338(1), San Diego, CA, pp. 774–785, 1998.

[Wilson 2000] R. Wilson, "MGMM: Multiresolution Gaussian Mixture Models for Computer Vision", *15th International Conference on Pattern Recognition*, Barcelona, Spain, Vol. 1, pp. 212–215, September 2000.

[Wood 1996] J. Wood, "Invariant Pattern Recognition: A Review", *Pattern Recognition*, Vol. 29, No. 1, pp. 1–17, January 1996.

[Zhang+ 1995] H. Zhang, Y. Gong, C. Low, S. Smoliar, "Image Retrieval based on Color Features: An Evaluation Study", *Proceedings of the SPIE*, Vol. 2606, pp. 212-220, 1995.

## Lebenslauf - Curriculum Vitae

| | |
|---|---|
| Name: | Jörg Dahmen |
| Adresse: | Schloß-Schönau-Straße 48<br>52072 Aachen |
| Geburtstag: | 29. September 1971 |
| Geburtsort: | 52428 Jülich |
| Nationalität: | deutsch |
| Konfession: | römisch-katholisch |
| Eltern: | Günter Dahmen, *06.02.1939<br>Karin Dahmen, geb. Sprenkelmann, *18.07.1939 |
| Geschwister: | Klaus Dahmen, *22.10.1965<br>Katja Dahmen, *26.11.1967 |
| 29. Mai 1999: | Heirat mit Anne Elisabeth Schneider-Dahmen, geb. Schneider |
| 21. Juli 2000: | Geburt der Tochter Pia Katharina Dahmen |

**Schulbildung:**

| | |
|---|---|
| August 1978 – Juli 1982: | Gemeinschaftsgrundschule Süd, Jülich |
| August 1982 – Juni 1991: | Gymnasium Zitadelle, Jülich |

**Wehrdienst:**

| | |
|---|---|
| Juli 1991 – Juni 1992: | Fernmeldebatallion 930, Gerolstein |

**Studium:**

| | |
|---|---|
| Oktober 1992 – Juni 1997: | Informatikstudium an der RWTH Aachen |
| September 1994: | Vordiplom |
| Juni 1997: | Diplom |

**Arbeitstätigkeiten:**

| | |
|---|---|
| Oktober 1994 - Februar 1995: | Studentische Hilfskraft am Lehrstuhl für Informatik V,<br>RWTH Aachen |
| Juni 1995 - April 1996: | Studentische Hilfskraft am Institut für Kunststoffverarbeitung<br>RWTH Aachen |
| Januar 1996 - April 1997: | Tätigkeit bei der Ikarion Software GmbH, Aachen |
| seit August 1997: | Promotionsstudent am Lehrstuhl für Informatik 6,<br>RWTH Aachen |

Unterschrift