A Mixture Density Based Approach to Object Recognition for Image Retrieval

J. Dahmen, M.O. Güld, H. Ney, K. Beulen

Lehrstuhl für Informatik VI
RWTH Aachen - University of Technology
D-52056 Aachen, Germany
{dahmen, beulen, gueld, ney}@informatik.rwth-aachen.de

Abstract

In the last few years, statistical classifiers based on Gaussian mixture densities proved to be very efficient for automatic speech recognition. The aim of this paper is to find out how well such a 'conventional' statistical classifier performs in the field of image object recognition (for future use within a content-based image retrieval system) We present a mixture density based Bayesian classifier and compare the results obtained on some well known image object recognition tasks with other state-of-the-art classifiers. Furthermore, we propose a new, robust variant of a linear discriminant analysis for feature reduction as well as a new classification approach, called the 'virtual-test-sample' method, which significantly improves the recognition performance of the proposed classifier.

1 Introduction

In this paper we compare a mixture density based Bayesian classifier to other state-of-the-art classifiers such as support vector machines, artificial neural nets or decision trees. In our approach, we model the observed data either by Gaussian mixture densities or kernel densities, where parameter estimation is performed using the Expectation-Maximization algorithm (EM-algorithm) in combination with Linde-Buzo-Gray clustering. As we make use of *appearance based pattern recognition*, that is we interpret each pixel of an image as a feature, the resulting feature vectors are high-dimensional. To avoid the difficulties of estimating covariance matrices in a high-dimensional feature space, we use a new variant of a linear discriminant analysis (LDA) to transform the data into an appropriate, low-dimensional feature space. The proposed LDA variant proved to be numerically more robust than a conventional LDA. To evaluate the performance of the proposed classifier, we apply it to the widely used US Postal Service corpus (USPS) for handwritten digits, as well as to the Max Planck Institute's Chair Image Database (CID). The obtained error rates of 3.4% on USPS (performing the virtual-test-sample method proposed in Section 3) and 0.4% on CID prove our classifier to be state-of-the-art (cp. Section 8). Our long-term goal is the use of the proposed algorithms in a statistical image retrieval system for the automatic indexing of images (with an image being composed of multiple objects). A first application of the statistical classifiers described in this paper within a (medical) image retrieval system can be found in [Dahmen et al., 2000].

1.1 Object Recognition for Image Retrieval

At first sight, the relationship between statistical object recognition and image retrieval might not be apparent. Yet, if we interpret an image as being composed of multiple objects, we can use object recognition algorithms for *automatic image indexing* by defining object classes that are suitable for a given domain. If the chosen domain is natural images, suitable classes would be 'grass', 'leaves' or 'sky', in other cases we might chose the classes 'crowd', 'car' or 'face'. Looking at object recognition this way, image retrieval can be understood as a 'coarse-to-fine' object recognition problem. In a certain application, detection of the object 'crowd' might be sufficient (coarse search), in another, we might be interested in pictures showing certain people, i.e. we would look for known faces within the 'crowd'-area (fine search). Thus, image indexing can be achieved via object recognition in many practical applications. Retrieval can then be performed on the extracted image indices, using suitable distance measures. Therefore, we will concentrate on statistical object recognition algorithms in the following.

It should be noted that a mixture density based statistical classifiers tends to do computational complex calculations in the training phase. After training, classification can be done very efficiently. Thus, the proposed statistical classifiers are especially suited for use in practice. Moreover, computational complexity can be further reduced by using discriminative training criteria (such as the maximum-mutual information criterion) instead of the maximum-likelihood standard approach. More information on that topic can be found in [Dahmen et al., 1999].

1.2 Related Work

While appearance based image object recognition is pretty common, the use of statistical classifiers such as the one we propose is not. MOGHADDAM & PENTLAND use Gaussian Mixtures, too, but do not focus on the statistical point of view of object recognition [Moghaddam & Pentland, 1997]. Moreover, they use a principal components analysis (PCA) for feature analysis, just as many other researchers do, among them [Turk & Pentland, 1991] or [Murase & Nayar, 1995]. Yet - from a theoretical point of view - using an LDA seems more suitable, as an LDA - in opposite to the PCA - yields the optimal linear transformation of the original data into a low-dimensional feature space with respect to class separability. One reason is that estimating an LDA transformation matrix can be difficult, especially if only few, high-dimensional training data is available. Therefore, SWETS ET AL. use an LDA after reducing the dimensionality of the feature vectors using a PCA [Swets & Weng, 1998]. The virtual-test-sample method we propose in Section 3 was motivated by KITTLER's research on classifier combination schemes [Kittler, 1998]. It should also be noted that many researchers evaluate their algorithms on problem specific data, making it hard to compare the efficiency of different algorithms. For that reason, we apply our classifier to widely used standard corpora such as USPS, allowing for a meaningful comparison of the results obtained by the different classifiers. In case varying training and/ or testing data is used, a fair comparison is not possible.

2 Databases used for Experiments

To enable other research groups to compare their methods with the one proposed in this paper, we will give a short description of the corpora we used.

2.1 The Chair Image Database

The databases we used in our experiments are USPS and the Max Planck Institute's CID. The latter one (`ftp://ftp.kyb.tuebingen.mpg.de/pub/chair_dataset`) consists of computer generated images of office chairs out of 25 classes. There are different training sets available, but we only used the largest one with 400 different 3D-views per class, 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 16×16 pixels sized grayscale image (see Figure 1). 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 1280-dimensional [Blanz et al., 1996].

2.2 The US Postal Service Task

The USPS database (`ftp://ftp.kyb.tuebingen.mpg.de/pub/bs/data/`) is a well known handwritten digit recognition task. It contains 7291 training objects and 2007 test objects.

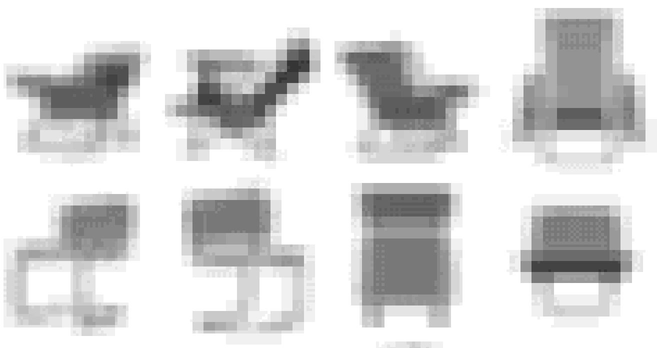

Figure 1: Example images taken from CID

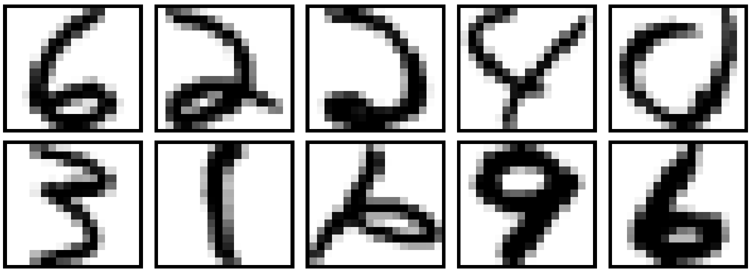

Figure 2: Example images taken from USPS

The digits are isolated and represented by a 16×16 pixels sized grayscale image (see Figure 2). Making use of *appearence based pattern recognition*, we interpret each pixel as a feature in our experiments. The USPS recognition task is known to be hard, with a human error rate of about 2.5% on the testing data [Simard et al., 1993].

A typical drawback of statistical classifiers is their need for a large amount of training data, which is not always available. To overcome this difficulty in our USPS experiments, we create additional *virtual training data* by shifting each image by one pixel into eight directions. Doing so, we on the one hand get a more precise estimation for the model parameters of our classifier and on the other hand we obviously incorporate invariance to slight translations. This procedure leads to 65.619 training samples which are then used to train our system. Note that the translated images are of size 18×18 pixels, as we want to guarantee that no pixel belonging to a digit gets shifted out of the image. Thus, the respective feature vectors are 324-dimensional (before feature reduction). Note that despite creating virtual training data, we still use the original USPS training set. We neither add new training data, nor do we remove training samples. We just use a-priori knowledge to create additional data.

3 The Virtual-Test-Sample Method

Similar to creating virtual training data, we propose the following virtual-test-sample method (VTS). Using our a-priori knowledge about the data again (shifting an image does not change its class), we create virtual test samples by shifting each image into eight directions. We then classify each of these virtual test samples separately and use the sum rule [Kittler, 1998] to come to a final decision for the original test sample. Note that in case of VTS, the motivation for the sum rule differs from that proposed by KITTLER. To justify the sum rule in the case of using multiple classifiers to classify a single test pattern, he assumed that the posterior probabilities computed by the respective classifiers do not differ much from the prior probabilities. In contrast to this, using multiple test patterns and a single classifier, the sum rule simply follows from the fact that the transformations considered are mutually exclusive, if we assume that the respective prior probabilities are equal (e.g. the prior probability for a right shift should be the same as for a left shift, which seems reasonable).

The basic idea behind this method is that we are able to use classifier combination rules and their benefits without having to create multiple classifiers. Instead, we simply create virtual test samples. Thus, classifying a pattern using VTS has the same computational complexity as compared to using any other classifier combination scheme, yet the (computationally expensive) training phase remains unaffected. Despite its simplicity, VTS proved to be very efficient (cp. Section 8).

4 Feature Reduction

To reduce the number of model parameters to be estimated as described in Section 6, transforming the data into some low-dimensional feature space is advisable. To do so, we propose the following LDA variant: In a first step, we use the training data to estimate a whitening transformation matrix W [Fukunaga, 1990, pp. 26-29]. The data is then transformed using W and is afterwards called *white*, that is the class conditional covariance matrix

$$\hat{\Sigma} = \frac{1}{N} \sum_{n=1}^{N} (x_n - \mu_{k_n})(x_n - \mu_{k_n})^t \tag{1}$$

is the matrix of identity. N is the number of training samples, given as labelled pairs (x_n, k_n), where x_n is the observation of training sample n with according class index k_n and μ_{k_n} is the mean vector of class k_n, In a second step, we generate K prototype vectors of the form $\mu_k - \mu$, where K is the number of classes, μ_k is the mean vector of class k and μ is the overall mean vector. We now transform those vectors into an orthonormal basis. To avoid the numerical instabilities of the classical Gram-Schmidt approach (caused by rounding errors), this is done by using a singular value decomposition [Press et al., 1992, pp. 59-67]. The SVD yields a maximum of $(K-1)$ base vectors, as the overall mean vector is a linear combination of the class specific mean vectors. By projecting the original feature vectors into that subspace we obtain the reduced feature vectors, which will be used in the following.

The proposed method proved to be more robust than the conventional computation of the LDA transformation matrix [Duda & Hart, 1973, pp. 114-123], but (in the absence of numerical problems) still gives the same results as compared to $(K-1)$ LDA features (see Section 8). In case fewer than $(K - 1)$ features are wanted, an LDA can be used in a second step, on the previously reduced features. If more features are needed, it is advisable to create so-called ‘pseudoclasses’ by clustering the training data and by interpreting each cluster as a pseudoclass. In the case of the USPS database we obtained our best results by creating four pseudoclasses per class (the resulting feature vectors being 39-dimensional). Note that the pseudoclasses are created by using the algorithms described in Section 6, i.e. by training mixture densities.

5 Gaussian Mixture Densities

To classify an observation $x \in \mathbb{R}^d$ we use the Bayesian decision rule [Duda & Hart, 1973, pp. 10-39]

$$x \longmapsto r(x) = \underset{k}{\operatorname{argmax}} \{p(k)p(x|k)\} \tag{2}$$

where $p(k)$ is the *a priori* probability of class k, $p(x|k)$ is the *class conditional* probability for the observation x given class k and $r(x)$ is the classifier's decision. As neither $p(k)$ nor $p(x|k)$ are known, we have to choose models for the respective distributions and estimate their parameters by using the training data. In our experiments, we set $p(k) = \frac{1}{K}$ for each class k (which holds for the chair image database and is assumed for the USPS corpus, as it is not obvious why a certain digit should have a higher prior probability than another) and model $p(x|k)$ 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:

$$p(x|k) = \sum_{i=1}^{I_k} c_{ki} \cdot \mathcal{N}(x|\mu_{ki}, \Sigma_{ki}) \tag{3}$$

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), μ_{ki} is the mean vector and Σ_{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, we make use of pooled covariance matrices in our experiments:

- *class specific variance pooling* :
  estimate only a single Σ_k for each class k, i.e. $\Sigma_{ki} = \Sigma_k \; \forall \; i = 1, ..., I_k$
- *global variance pooling* :
  estimate only a single Σ, i.e. $\Sigma_{ki} = \Sigma \; \forall \; k = 1, ..., K$ and $\forall \; i = 1, ..., I_k$

Furthermore, we only use a diagonal covariance matrix, i.e. a variance vector. Note that this does not mean a loss of information, as a mixture density of that form can still approximate any density function with arbitrary precision. Maximum-likelihood parameter estimation is now done using the Expectation-Maximization (EM) algorithm [Dempster et al., 1977] combined with a Linde-Buzo-Gray based clustering procedure [Linde et al., 1980].

6 Parameter Estimation

In this Section we will deal with estimating the mixture density parameters, i.e. c_{ki}, μ_{ki} and Σ_{ki}. To do so, we use the EM-algorithm, a maximum likelihood parameter estimation approach for data with so-called hidden variables. In our case, the unknown membership relation between observations x_n and mixture components $\mathcal{N}(x_n|\mu_{ki}, \Sigma_{ki})$ is the hidden variable. We will present estimation formulae for this problem, as well as a simplified variant of the EM-algorithm, based on a maximum approximation.

6.1 The EM-Algorithm

The EM-algorithm is an iterative approach to estimating the parameters of some unknown probability function with hidden variable i. Its application to mixture densities is described in [Dempster et al., 1977], where the index of some density which an observation belongs to is interpreted as hidden variable. This assignment is expressed as a probability $p(i|x_n, k, \lambda_{ki})$, with k being the class index, i the density index and $\lambda_{ki} = (c_{ki}, \mu_{ki}, \Sigma_{ki})$. Using the EM-algorithm on this problem yields the following estimation formulae:

$$p(i|x_n, k, \lambda_{ki}) = \frac{c_{ki} \cdot \mathcal{N}(x_n|\mu_{ki}, \Sigma_{ki})}{\sum_{i'} c_{ki'} \cdot \mathcal{N}(x_n|\mu_{ki'}, \Sigma_{ki'})} \tag{4}$$

$$\gamma_{ki}(n) = \frac{p(i|x_n, k, \lambda_{ki})}{\sum_{n'} p(i|x_{n'}, k, \lambda_{ki})} \tag{5}$$

$$\bar{c}_{ki} = \frac{1}{N_k} \sum_{n=1}^{N_k} p(i|x_n, k, \lambda_{ki}) \tag{6}$$

$$\bar{\mu}_{ki} = \sum_{n=1}^{N_k} \gamma_{ki}(n) \cdot x_n \tag{7}$$

$$\bar{\Sigma}_{ki} = \sum_{n=1}^{N_k} \gamma_{ki}(n) \cdot [x_n - \bar{\mu}_{ki}][x_n - \bar{\mu}_{ki}]^T \tag{8}$$

with N_k being the number of training samples of class k. We start the iteration by estimating the parameters c_{ki}, μ_{ki} and Σ_{ki}, yielding our initial $p(i|x_n, k, \lambda_{ki})$. Using this $p(i|x_n, k, \lambda_{ki})$, the parameters λ_{ki} can be re-estimated by setting $c_{ki} := \bar{c}_{ki}$, $\mu_{ki} := \bar{\mu}_{ki}$ and $\Sigma_{ki} := \bar{\Sigma}_{ki}$, yielding a better estimation for $p(i|x_n, k, \lambda_{ki})$. This procedure repeats until the parameters converge or until a certain number of iteration has been performed.

In our 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 we are using a Linde-Buzo-Gray [Linde et al., 1980] inspired method. To overcome the problem of choosing the initial values for c_{ki}, μ_{ki} and Σ_{ki}, we start 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 μ_{ki} using a suitable distortion vector ϵ (in our experiments, fast convergence was obtained by choosing ϵ to be a fraction of the respective variance vector). This method proved to be very efficient for modelling emission probabilities in speech recognition [Ney, 1990]. We get 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 (4)-(8). This splitting procedure repeats until the desired number of densities is reached. In our experiments, in order to get reliable estimations for the unknown parameters, a density i belonging to class k is only splitted, if

$$\sum_{n=1}^{N_k} p(i|x_n, k, \lambda_{ki}) \geq N_{Split} \tag{9}$$

holds. In our experiments, we choose N_{Split} to be 4.0.

6.2 Maximum Approximation

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

$$p(i|x_n, k, \lambda_{ki}) = \begin{cases} 1 & : \text{'best' single density } i \\ 0 & : \text{otherwise} \end{cases} \tag{10}$$

In this case, each training vector is assigned to only one density, namely the best one. Because of that, instead of calculating any $p(i|x_n, k, \lambda_{ki})$ we only need to find the maximum $\mathcal{N}(x_n|\mu_{ki}, \Sigma_{ki})$ (thus, the sum in Eq. 3 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 $p(i|x_n, k, \lambda_{ki})$. Therefore, in many practical cases, the computational complexity of the training- and recognition-step can be reduced without a significant deterioration of recognition accuracy.

7 Kernel Densities

In the case of kernel densities (also called parzen windows or parzen densities sometimes) [Devroye et al., 1996, pp. 147-153], each training sample x_n defines a Gaussian single density $\mathcal{N}(x|x_n, \Sigma_{x_n})$ with an estimated covariance matrix Σ_{x_n}, that is the sample itself is interpreted as mean vector. To classify an observation x, we use the decision function

$$x \longmapsto r(x) = \underset{k}{\text{argmax}} \{p_{KD}(x|k)\} \tag{11}$$

where

$$p_{KD}(x|k) = \frac{1}{N_k} \sum_{n=1}^{N_k} \mathcal{N}(x|x_n, \Sigma_{x_n}). \tag{12}$$

In the experiments, we used the procedure described above to train a mixture density for each class k and for all training samples x_n belonging to density i of class k we define

$$\Sigma_{x_n} = \alpha \cdot \Sigma_{ki} \tag{13}$$

with $\alpha > 0$. We then examine the test error rate as a function of α and we also compare kernel densities with a nearest neighbour classifier (14), as one would expect the kernel density error rate to converge to that of nearest neighbour with $\alpha \to 0$.

$$x \longmapsto r(x) = k\left(\underset{n}{\text{argmax}}\ \mathcal{N}(x|x_n, \Sigma_{x_n})\right) \tag{14}$$

Where $k(n)$ is a function that - given an observation x_n - returns the according class index k_n. Thus, the observation x is assigned to the class that the 'best-fitting' training sample belongs to. Note that this classifier is independent of the choice of α.

8 Results

In this Section we present results for the proposed classifier on USPS and CID. We also compare our results with those reported by other groups, proving them to be state-of-the-art.

8.1 Experiments on CID

We started our experiments on CID using Gaussian single densities. Without feature reduction, the best test error rate we obtained was 15.1%, using class specific variance pooling. Using the proposed subspace method to reduce the feature space to 24 dimensions (as K=25 for CID) improved the error rate to 3.3%, proving the effectiveness of our feature reduction approach. We then used the reduced features to realize a mixture density based classifier. Figure 3 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% is very well comparable to the results that were reported by other groups (as shown in Table 1). In the case of this database the kernel density ap-

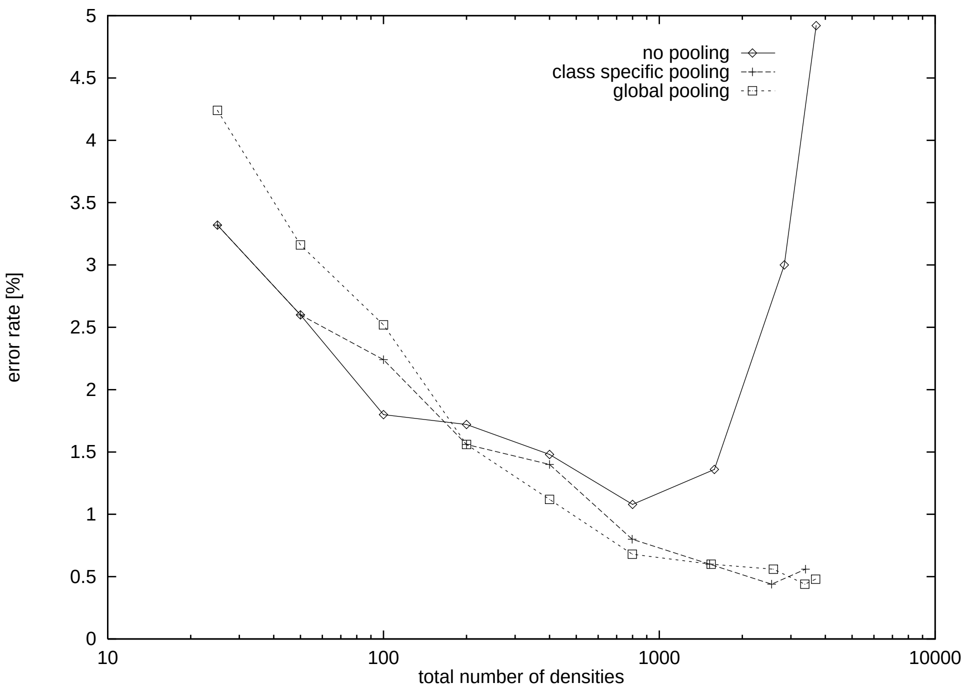

Figure 3: CID error rates as a function of the number of densities for three types of variance pooling (maximum approximation)

proach yields an error rate of 0.4%, too, that is the error rate could not be further improved. Note that using a conventional linear discriminant analysis instead of the proposed variant we were not able to get a result better than 0.7%.

Because of the CID classification error rate being close to zero (0.4% error means a total of only 10 misclassifications out of the 2500 test samples), we neither created virtual training samples on the CID database, nor did we apply the proposed virtual test sample method to it. Instead, to achieve a meaningful interpretation of the effects of VTS, we continued our experiments on the USPS corpus.

Table 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 2: Results obtained on USPS without feature reduction

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 3: Results obtained on USPS with feature reduction (subspace method)

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

8.2 Experiments on USPS

USPS experiments were started by applying the proposed statistical classifiers to the original USPS data, without creating virtual data and without feature reduction. We obtained a single density error rate of 19.5% (i.e. $I_k = 1$ for each class k) and a mixture density error of 8.0%. 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, the kernel density based classifier yielded the best error rate we obtained without doing any feature reduction. An overview of the obtained results without feature reduction is shown in Table 2, where the mixture density results were obtained using the EM-algorithm and globally pooled variances. The notation 'a-b' indicates, that we increased the number of training samples by a factor of a and that of the test samples by a factor of b. Thus, b=9 indicates that we performed VTS as proposed in Section 3.

For our further experiments on USPS, the dimensionality of the feature space was again reduced as described in Section 4. 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

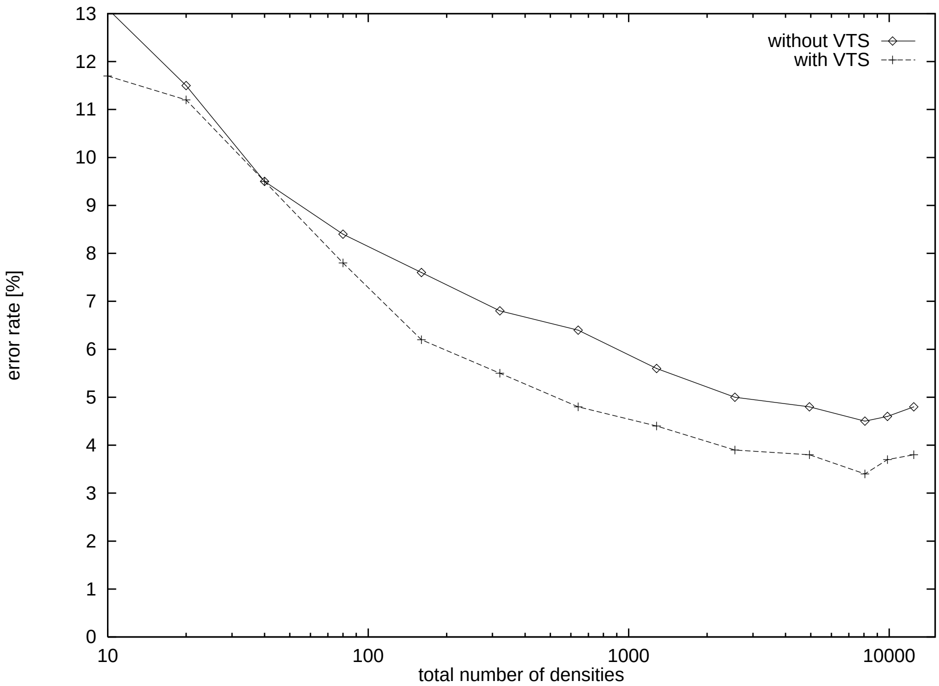

Figure 4: EM error rates on USPS using globally pooled variances, with and without VTS

virtual training data significantly 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 VTS further reduced the error rate to 3.4%, which is very well comparable to results reported by other groups (cp. Table 4).

In a last experiment, we compared the efficiency of the mixture density based classifier with that of kernel densities. Figure 4 shows the EM error rate for global variance pooling with respect to the total number of densities used to model the observed training data, with and without using VTS. 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 increases again. This is due to the fact that the probabilistic model is overfitted on the training data, leading to decreasing generalization for unseen observations. 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 USPS corpus.

We then used the estimated global variance vector (remember that we only estimate the diagonal elements of Σ_{ki}) to realize a kernel density based classifier. Figure 5 shows the achieved results with respect to α in comparison to a nearest neighbour classifier (with error rate 4.9%, being independent of the choice of α). Using kernel densities without VTS, we could improve our best mixture density error rate from 4.5% to 4.2%. Not surprisingly, this result is achieved using

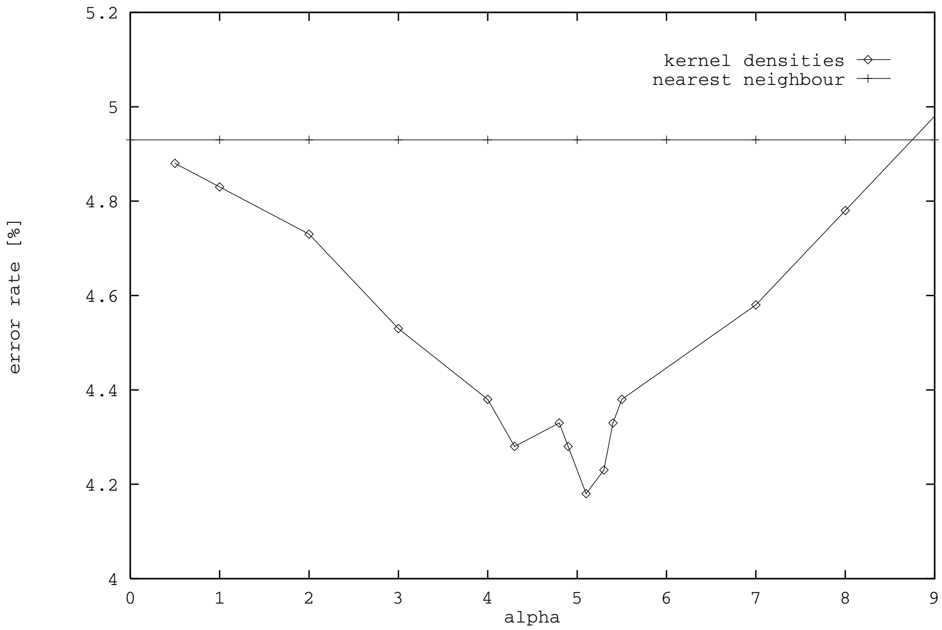

Figure 5: Kernel Density error rates on USPS with respect to alpha, compared to a NN-Classifier (NN-error rate is 4.9%)

α =5.1. This is due to the fact that parameter estimation based on a rather small training set tends to underestimate variances. With $\alpha > 1$ we therefore compensate for this effect, yet with α getting too large the variances get 'flattened' too much and the error rate increases. Again, using VTS reduced the error rate from 4.2% to 3.4%.

An overview of the obtained results with feature reduction (i.e. performing the subspace method) is shown in Table 3. The mixture density results were obtained using EM-algorithm and globally pooled variances. A comparison of our results with that achieved by other state-of-the-art methods can be found in Table 4. Note that we only considered research groups that used the original USPS training and test sets. Otherwise, a comparison of the training and classification methods used is not possible. Other groups for instance improved the recognition performance of their systems to about 2.6% error by adding 2.500 machine printed digits to the training set [Simard et al., 1993, Drucker at al., 1993]. Note that creating virtual data does not violate our assumption that all groups considered should use the same data, as we simply use a-priori knowledge to create virtual training- and/ or test samples. Our algorithms are still based on the original USPS datasets.

9 Conclusions and Outlook

In this paper, we presented a mixture density based approach to object recognition that yields state-of-the-art results on well known image recognition tasks (0.4% error rate on the Max Planck

Table 4: Results reported on USPS

Author	Method	Error [%]
Simard et al., 1993	Human Performance	2.5
Vapnik, 1995	Decision Tree C4.5	16.2
Vapnik, 1995	Two-Layer Neural Net	5.9
Simard et al., 1998	Five-Layer Neural Net	4.2
Schoelkopf, 1997	Support Vectors	4.0
Schoelkopf et al., 1998	Invariant Support Vectors	3.0
This work:	Gaussian Mixtures	4.5
	Kernel Densities	4.2
	Mixtures, VTS	3.4
	Kernel Densities, VTS	3.4

Institute's Chair Image Database, 3.4% on the US Postal Service handwritten digits corpus). We are currently improving our feature analysis, that is we are going to extract features that are invariant with respect to linear as well as non-linear image transformations. We will also investigate on the efficiency of invariant distance measures (such as SIMARD's tangent distance [Simard et al., 1993]) within the proposed classifiers (with the tangent distance replacing the Euclidean distance used in our experiments). Future work will also include boosting our mixture density based classifier [Freund & Schapire, 1996, Kittler, 1998], as well as improving the proposed virtual-test-sample method. It will be interesting to see whether one method proves to be superior or whether it makes sense to combine both approaches.

One of our long-term goals is the use of the proposed algorithms as part of a statistical image retrieval system. In some special cases an interpretation of the image retrieval problem as a classification task seems feasible. If we concentrate on certain domains such as traffic images, automatic indexing of images can be achieved via classification, with an image being composed of multiple objects (out of classes such as 'car', 'people', 'face' or 'tree'). Using the generated image indices (e.g. *'a car at position (x,y), with people standing to the right of it'*) similarity between different images can be quantified using a suitable distance measure. First experiments using the algorithms described in this paper within a (medical) image retrieval system can be found in [Dahmen et al., 2000].

References

[Blanz et al., 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.): *Artificial Neural Networks - ICANN’96* , Springer Lecture Notes in Computer Science, Vol. 1112, pp. 251-256, Berlin, 1996.

[Dahmen et al., 1999] J. Dahmen, R. Schlüter, H. Ney, ”Discriminative Training of Gaussian Mixtures for Image Object Recognition”, in W. Förstner, J. Buhmann, A. Faber, P. Faber (eds.): Proceedings of the 21. Symposium of the German Association for Pattern Recognition, pp. 205-212, Bonn, Germany, September 1999.

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

[Dempster et al., 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.

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

[Drucker at al., 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, 1973.

[Freund & Schapire, 1996] Y. Freund, R. Schapire, “Experiments With a New Boosting Algorithm”, Proceedings of the 13th International Conference on Machine Learning, Bari, Italy, July 1996.

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

[Kittler, 1998] J. Kittler, “On Combining Classifiers”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 3, pp. 226-239, March 1998.

[Kressel, 1998] U. Kressel, personal communication, Daimler-Benz-AG, Research and Technology, Department FT3/AB, Ulm, March 1998.

[Linde et al., 1980] Y. Linde, A. Buzo und R. Gray, “An algorithm for Vector Quantizer Design”, *IEEE Transactions on Communications*, Vol. 28, No. 1, pp. 84-95, 1980.

[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.

[Murase & Nayar, 1995] H. Murase, S. Nayar, “Visual Learning and Recognition of 3D Objects from Appearance”, International Journal on Computer Vision, Vol 14, No.1, 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.

[Press et al., 1992] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, *Numerical Recipes in C*, University Press, Cambridge, 1992.

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

[Schoelkopf et al., 1998] B. Schölkopf, P. Simard, A. Smola, V. Vapnik, “Prior Knowledge in Support Vector Kernels”, M. Jordan, M. Kearns, S. Solla (eds.): *Advances in Neural Information Processing Systems 10*, MIT Press, pp. 640-646, 1998.

[Simard et al., 1993] P. Simard, Y. Le Cun, J. Denker, “Efficient Pattern Recognition Using a New Transformation Distance”, S.J. Hanson, J.D. Cowan, C.L. Giles (eds.): *Advances in Neural Information Processing Systems 5*, Morgan Kaufmann, San Mateo CA, pp. 50-58, 1993.

[Simard et al., 1998] P. Simard, Y. Le Cun, J. Denker and B. Victorri, “Transformation Invariance in Pattern Recognition — Tangent Distance and Tangent Propagation”, *Lecture Notes in Computer Science*, Vol. 1524, Springer, pp. 239-274, 1998.

[Swets & Weng, 1998] D. Swets, J. Weng, “Using Discriminant Eigenfeatures for Image Retrieval”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 8, pp. 833-836, March 1998.

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

[Vapnik, 1995] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, pp. 142-143, 1995.