Diplomarbeit im Fach Informatik
Rheinisch-Westfälische Technische Hochschule Aachen
Lehrstuhl für Informatik 6
Prof. Dr.-Ing. H. Ney

Deformations and Discriminative Models for Image Recognition

vorgelegt von:
Tobias Gass
Matrikelnummer 231497

Gutachter:
Prof. Dr.-Ing. H. Ney
Prof. Ph. D. G. Lakemeyer

Betreuer:
Dipl.-Inform. T. Deselaers

Erklärung

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Alle Textauszüge und Grafiken, die sinngemäss oder wörtlich aus veröffentlichten Schriften entnommen wurden, sind durch Referenzen gekennzeichnet.

Aachen, im Juli 2008

Tobias Gass

Abstract

In this work we present approaches to incorporate domain-knowledge into discriminative classifiers. In particular, we investigate the incorporation of the image distortion model into log-linear models and support vector machines. Discriminative models are a well-known technique in many fields of machine learning and pattern recognition and the image distortion model has recently been shown to be a very effective means of modelling variability in images of handwritten characters, however, so far it was impossible to fuse the advantages of these two different approaches. In order to incorporate the IDM into the log-linear model, we re-investigate the probabilistic formulation of the IDM, and the relationships between Gaussian models and log-linear models, which allows for a direct integration of the IDM into the log-linear framework, where an alignment of an image to a prototype is considered a hidden variable. Alternating optimisation techniques allow for effective training of log-linear models with hidden variables. Furthermore,we have investigated how to combine SVMs with the IDM. The performance of the different models is examined on two standard tasks, the USPS tasks, which, due to its small size, allows for extensive testing and tuning of models and the MNIST database, which is considered a standard benchmark in OCR applications. The results obtained compare favourably well to other, comparable models. In particular, we report the best error rate using a single density model on the USPS task.

Acknowledgments

I would like to express my gratitude to all people who made this work possible.

First, I would like to thank Prof. Dr.-Ing. Hermann Ney for the interesting possibilities at the Chair of Computer Science VI at the RWTH Aachen. I have been a student researcher in the image recognition group since fall 2004 and lots of knowledge and ideas originate from this work.

Furthermore, I would like to thank Thomas Deselaers for his supervision of this work, lots of ideas, helpful suggestions and interesting discussions.

I would like to thank Prof. Ph. D. Gerhard Lakemeyer as well, who kindly accepted to attend this work.

Also, I would like to thank the other members of the image recognition group at the Chair of Computer Science 6 - Tobias Weyand, Pascal Steingrube, Jens Forster, Jan Ziegeldorf, Stephan Jonas and Phillippe Dreuw - for helpful comments, hints, interesting discussions and partly proof reading this manuscript. Additionally, I would like to thank all other student researchers for lots of fun in the hiwi-room.

Of course special thanks go to my parents and grandparents for all the things they made possible and for their continuing support. Finally I would like to thank Charlotte for her interest and support.

Contents

Chapter 1

Introduction

Many applications for image recognition demand for efficient and precise methods for applying class labels to unseen images. For example, modern postal services recognise hand-written addresses with postal codes automatically, which has similar requirements as recognising the content of forms completed by hand writing. Additional applications are face recognition, *e.g.* in automatic surveillance systems or in user authentification processes. Furthermore, the ever-growing amount of medical imagery demands for efficient means of classifying the content in order to support medical diagnostics.

A recognition method is considered precise, if the number of images labelled (=classified) falsely is small compared to the total number of images to be classified. Since a common approach for the classification of an unseen image is to find the most similar images in a database with known class labels and apply the most common class label to the unseen image (known as k nearest neighbour (KNN)), it is straight forward to increase classification performance by finding suitable similarity (or dissimilarity=distance) measures. The most general and thus most common approach is to use the l_2-norm (Euclidean distance) as dissimilarity measure, which already leads to good results in a wide variety of tasks. However, *e.g.* even marginally shifted objects in an image can lead to high Euclidean distances and thus low similarity. In order to increase recognition performance, it is thus profitable to incorporate domain knowledge into the similarity function. For images, it is desirable to obtain similarity measures which are invariant to global or local transformations like rotation or scaling because they usually do not change the class of the depicted content. The most basic approach to incorporate this kind of invariance is to apply the transformations of interest to the training data, effectively enriching the variability of the training data. This has been done by Simard [2003], who trains artificial neural networks (ANNs) using randomly deformed training images. Additionally, Haasdonk et al. [2004] propose to use invariant features obtained by integrating over the transformation space. A more implicit method to incorporate domain knowledge is to model deformations of images directly. For example, Keysers et al. [2000] use tangent vectors at pixel positions in order to obtain a distance measure which is invariant w.r.t. small local transformations. Uchida and

Sakoe [2005] give an overview over several elastic matching techniques for optical character recognition (OCR), which are at least partly computationally very complex. In [Keysers et al., 2007, Keysers, 2006, Gollan, 2003, Keysers et al., 2004b], several different approaches to explicitly modelling deformations with second- to zero-order complexity are presented, compared and evaluated on a variety of tasks like OCR and medical image recognition. It is shown that using a zero-order model with additional features like Sobel gradients and local context leads to extremely good recognition accuracy, while keeping the computational demands low and the model simple.

Most of the presented similarity measures have mainly been used in nearest neighbour (NN)-based approaches, which on the one hand are conceptually very simple and lead to good baseline results, on the other hand become very slow when the amount of training data grows. In many applications it is thus favourable to train a model on the data with known class labels and use this model to classify unseen images.

In the classification literature, two approaches are most commonly distinguished: *generative* models aim at representing the data as good as possible, while *discriminative* models try to find parameters which separate the classes as good as possible. A simple generative model is Gaussian density (GD) classifier, which represents each class by its mean and variance and can easily be extended to mixtures of Gaussian densities. It is a common approach to further tune the parameters of a generative model discriminatively. Hegerath et al. [2006] show how image object recognition can be improved considerably by retraining Gaussian mixture densities of image patches. Common discriminative models are support vector machines (SVMs) [Schölkopf, 1997], which do not model a probability distribution but instead directly model the decision boundaries and log-linear models (LLMs) [Darroch and Ratcliff, 1972], which have many applications in speech recognition [Och and Ney, 2002] or image annotation [Deselaers et al., 2007]. For most classification tasks, discriminative models outperform generative ones due to the fact that the optimisation criterion of the discriminative models is chosen w.r.t. to exact that task.

Few efforts have been made to combine the advantages of modelling domain knowledge by finding suitable representations of image deformations and modelling a statistical distribution or boundary specific to the considered task. In Gollan [2003], the authors present Gaussian mixture densities (GMDs) with different deformation models for OCR. As far as discriminative approaches are concerned, Haasdonk and Keysers [2002] present an SVM with a tangent-distance based kernel and Haasdonk and Bahlmann [2004] give a general approach for using arbitrary distance functions in SVMs. DeCoste and Schölkopf [2002] propose to use smoothing and image shifts in their *virtual* SVM. Simard [2003] uses a convolutional neural network which is trained using training data that was obtained by randomly de-

forming the training images and thus the neural network learns about deformed images in the same way as it learns about the initial training data.

It is the main goal of this work to further investigate the fusion of discriminative models with knowledge of the image domain given by deformation invariance. We focus on using the image distortion model (IDM), which has been shown to lead to very good recognition accuracy despite a comparably low computational complexity. We shortly revisit this model in Chapter 2 and discuss the according model-free and generative approaches. In Chapter 3, we present a novel IDM kernel for SVMs. In Chapter 4, the IDM is reformulated in a probabilistic manner in order to obtain means to re-train the GMDs with IDM discriminatively. In Chapter 5 we present the main result of this work, where the IDM is incorporated into an LLM. The models are evaluated experimentally in Chapter 6. Chapter 7 concludes this work with a summary of the work done and an outlook on possible further research.

Chapter 2

Image Distortion Model

In many fields of image recognition, distance functions like the Euclidean distance are used to compare images given a suitable representation. This metric has the advantage of being conceptually simple, easy to implement and can be calculated efficiently where time complexity grows linearly compared to the size of the image descriptor. However, the Euclidean distance makes little to no use of specific information in images and thus often lacks in performance compared to methods exploiting domain-specific knowledge. Figure 2.1 shows this kind of errors where the Euclidean distance is used to find the most similar image in some database (here: US Postal Service (USPS)) and assign the most common class label of the closest images (called neighbours) to the query image. In each row, the query image is left-most and the database images with the smallest Euclidean distance follow to the right. It can be seen that a small Euclidean distance can lead to misclassification if the variability of the images is high, therefore it is favourable to model the variability implicitly in the distance function. One approach to do that is modelling invariance to some common transformations in the image domain like rotation, scaling and others. This kind of invariance can be modelled by warping reference images in order to best match some query image. An overview of some deformation schemes is given in Keysers et al. [2007], where zero- to second order deformation models are successfully applied to tasks like OCR or medical image recognition. In this context, the order of a deformation model describes the complexity of constraints imposed on the pixel-warping function. It is two-dimensional if the warping of a pixel fully depends on the warping of all other pixels. This problem is NP-complete as shown by Keysers and Unger [2003] and thus only solvable by approximation like simulated annealing [Hromkovic and Oliva, 2002]. Pseudo-two-dimensional models do not take all possible deformations into account and are thus less expensive in terms of time complexity. One possibility is to impose constraints only column-wise, where the resulting hidden Markov model (HMM) is of first order and the resulting matching algorithm is likewise named a first-order deformation model.

Finally, zero order models do not take placement of neighbouring pixels into account when looking for the best match of a pixel. In the IDM used throughout this

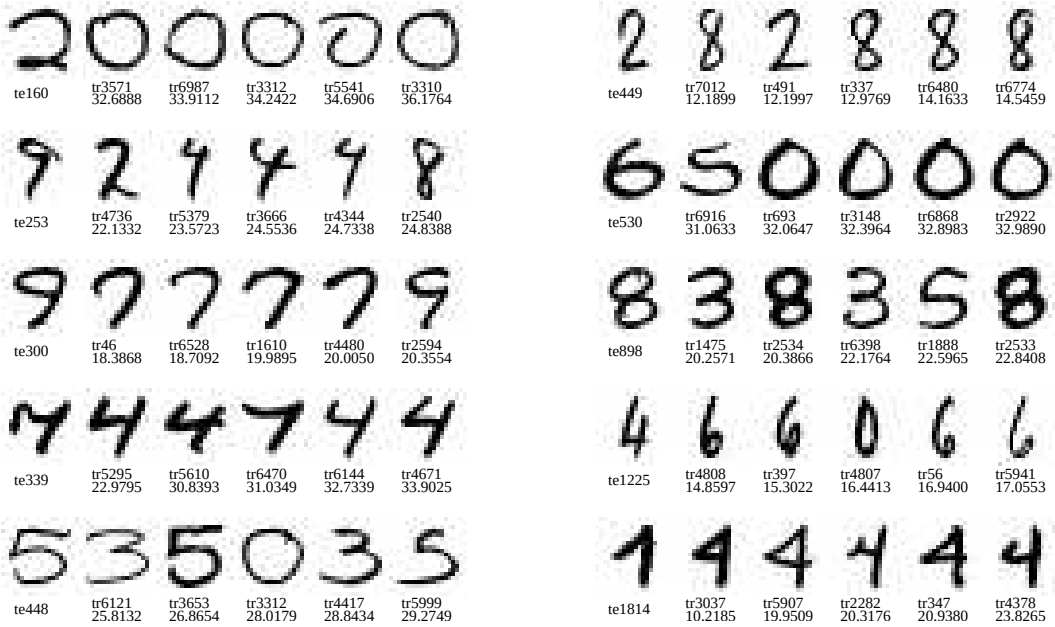

Figure 2.1. Examples of Euclidean distance leading to wrong classification on the USPS task

thesis, the only constraints for a pixel matching is the absolute warping distance. This leads to a computationally efficient model, which is reported to give extremely good results on suitable tasks.

Given two images A and B to be matched, with

$$\begin{aligned} A &= \{a_{ij}\}, \quad 1 \le i \le I, \quad 1 \le j \le J, \quad a_{ij} \in \mathbb{R}^{\mathsf{D}} \\ B &= \{b_{xy}\}, \quad 1 \le x \le X, \quad 1 \le y \le Y, \quad b_{xy} \in \mathbb{R}^{\mathsf{D}} \end{aligned} \tag{2.1}$$

the matching of a pixel coordinate (ij) of image A to a coordinate (x, y) of image B is given by the warping function

$$(x_{11}^{IJ}, y_{11}^{IJ}), \quad (i, j) \to (x, y) = (x_{ij}, y_{ij}). \tag{2.2}$$

The resulting matching scheme is visualised in Figure 2.2 and the according distance function d_{idm} is defined as follows:

$$d_{\mathrm{idm}}(A, B) = \sum_{ij} \min_{\substack{x=(i-W),..,(i+W) \\ y=(j-W),..,(j+W)}} g(A, ij, B, xy), \tag{2.3}$$

where $g(A, ij, B, xy)$ is a local distance function given pixel (ij) of image A and pixel (xy) of image B, W is the maximum warprange, and the best matching of

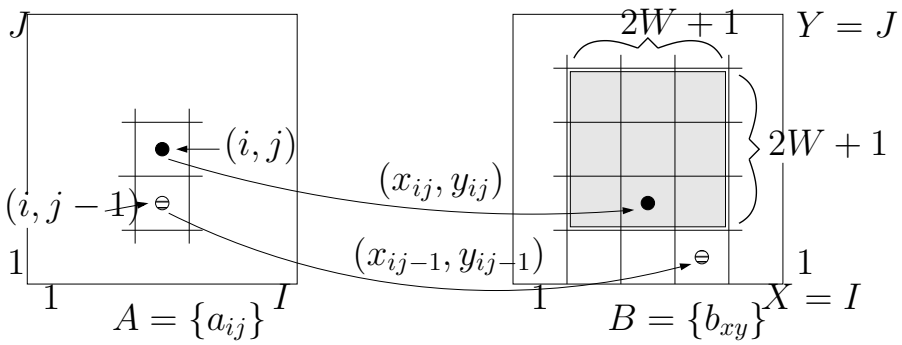

Figure 2.2. Warping scheme of the image distortion model. A pixel (i, j) is matched to pixel $x_{ij}, y_{ij})$ by searching the best matching pixel inside the warping area defined by the maximum warp range W.

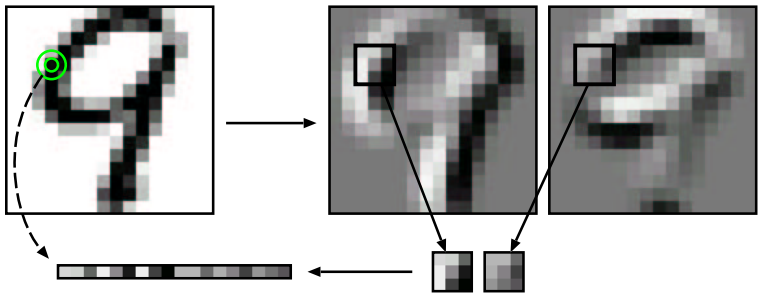

Figure 2.3. Feature expansion with local context and Sobel features. For each pixel horizontal and vertical Sobel gradients are computed. Then, each pixel is represented by the local area of either the grey valued pixel or the corresponding gradient images. In the latter case, the two separate areas are joined to form one local representation of the pixel.

A and B is defined by the local minima of d_{idm}. It has been shown by Keysers et al. [2007] that the best results can be achieved if each pixel is represented by the local context of its horizontal and vertical Sobel gradients as demonstrated in Figure 2.3. Furthermore, a function is added which penalises warping by absolute warping distance to avoid too extreme deformations.

2.1 Model-free Approach

The approach to deformation-invariant modelling as described before was to find a suitable distance function which is invariant to certain kinds of transformations. This distance function has been used in non-parametric models at first, where the most basic one is the NN approach. This model is widely used as baseline system

and often incorporates the Euclidean distance to find the closest observation μ in a labelled training set $\{\mu_1, \cdots, \mu_N\}$ to a query image X. The class label of μ is then assigned to X. It is now straightforward to replace the (Euclidean) distance function used before with the IDM-distance. The NN approach using IDM-distance has been successfully applied to a variety of tasks like OCR [Keysers et al., 2007] and the recognition of medical radiographs [Keysers et al., 2004a] [Deselaers and Ney, 2008], but requires N distance computations for each classification which comes to importance if the number of training images is large and the distance computation is not trivial. Compared to Euclidean distance computation, IDM distance calculation is slower by a factor of approximately $(2w+1)^2$ [Keysers et al., 2007], although it should be noted that no training is necessary.

2.2 Gaussian Models with IDM

Since there are many environments where computationally cheap classification is essential like input character recognition on hand-held devices, it is imperative to find models which fuse the advantages of the IDM and fast classification timings. One approach is to reduce a given training set to a number of prototypes, so in order to classify a query image it is only necessary to match it to all so called reference images. In the Euclidean space, this results in GDs, where each class is represented by the mean μ of all associated data and the respective covariance matrix, which is usually replaced by a pooled, diagonal covariance matrix. This method is well known in classical pattern recognition [Duda et al., 2001], with many applications *e.g.* in speech recognition or image object recognition [Deselaers et al., 2006].

The Bayesian decision rule can be rewritten

$$\begin{aligned} \hat{c} &= \arg\max_{c=1\ldots C} \{p(c|X)\} && (2.4) \\ &= \arg\max_{c=1\ldots C} \{p(c)p(X|c)\}, && (2.5) \end{aligned}$$

and assigns a class $\hat{c}$ to an observation X so that the product of the class prior $p(c)$ and the emission probability $p(X|c)$ is maximised. The emission probability is then modelled by a Gaussian distribution $\mathcal{N}(X|\mu_c, \sigma)$, where parameters μ_c and σ are usually trained with a standard expectation maximization (EM) algorithm which optimises the log-likelihood. If IDM is to be used, the distance function of the emission probability is changed, so

$$p(X|c) = \frac{1}{\sqrt{\det(2\pi\sigma)}} \exp\left\{-\frac{1}{2}\frac{||X-\mu_c||^2}{\sigma^2}\right\} \qquad (2.6)$$

Figure 2.4. Means of training image classes (top) and prototypes trained with IDM (bottom)

changes to:

$$p(X|c) = \frac{1}{\sqrt{\det(2\pi\sigma)}} \exp\left\{-\frac{1}{2}\frac{d_{\text{idm}}(X,\mu_c)}{\sigma^2}\right\}. \tag{2.7}$$

2.2.1 EM-Algorithm

It has been shown by Keysers et al. [2007] that using deformation-invariant distances for classification with GDs leads to decreased performance compared to Euclidean distance if the parameters are not trained according to the new distance measure, too. Since no closed form solution exists to compute the parameters, they are updated iteratively using the EM algorithm [Dempster et al., 1977]. With the IDM, this algorithm works as follows:

```
WHILE(iter<maxIter AND NOT convergence reached):

   E-step: Deform all images in the cluster to the mean image.

   M-step: Re-estimate mean as average of deformed images.
```

This algorithm is repeated until convergence or until a maximum number of iterations has been reached. Figure 2.4 shows that this method results in visually sharper prototypes, which improve recognition performance considerably.

2.2.2 Mixture Densities

It is a common approach [Duda et al., 2001] to extend the GD model to GMDs. This means that instead of using one cluster, each class is represented by a mixture of multiple clusters. This method has been successfully applied to a variety of tasks like image object recognition [Dahmen et al., 2001] and automatic speech recognition (ASR) [Lööf et al., 2006]. To derive a mixture of densities from a single

cluster, the cluster mean is usually disturbed by adding/subtracting fractions of the corresponding variance or mean and then reestimating the cluster memberships of the according training data. This process can be iterated to derive a larger number of densities. In most cases, densities which have no associated data after membership re-estimation are discarded. The emission probability in Equation (2.4) changes as follows:

$$p(X|c) = \sum_{e=1}^{E} p(e|c)p(X|c,e) = \sum_{e=1}^{E} p(e|c)\mathcal{N}(X|\mu_{ce},\sigma) \tag{2.8}$$

Here, μ_{ce} denotes the mean of density e of class c, which determine the according distribution $\mathcal{N}(X|\mu_{ce},\sigma)$ with pooled diagonal covariance. This distribution can cope with the IDM as presented in Section 2.2 and is again trained using a EM algorithm as follows:

```
WHILE(iter<maxIter AND NOT convergenceReached):

   E-step: Re-estimate cluster memberships using the
           emission probability.

   M-step: Compute cluster mean using the EM-algorithm
           for single densities with idm.
```

Eigendeformations

In the literature, densities are usually split by disturbing the mean using a fraction of the corresponding mean or variance [Linde et al., 1980]. However, since IDM is used throughout this work it is possible to further exploit the already computed deformations. So far, it has been proposed by Uchida and Sakoe [2003] to compute so called eigendeformations in order to penalise out-of-class deformations during the classification process. These eigendeformations are computed as follows:

For each image X the horizontal and vertical shift d_{x_i}, d_{y_i} is computed for each pixel $0 \leq i < N$ while comparing the image to a prototype μ_{ce}. This warping is computed by the IDM mechanism introduced before:

$$d_{x_i}, d_{y_i} = \underset{\substack{dx=-W,\ldots,W \\ dy=-W,\ldots,W}}{\arg\min}\ g(X, x_i, y_i, \mu_{ce}, x_i + d_x y_i + d_y) \tag{2.9}$$

Then, all pixel shifts are gathered to form a so-called deformation field

$$F(X, \mu_{ce}) = (d_{x_0}, d_{y_0}, \ldots, d_{x_N}, d_{y_N}) \tag{2.10}$$

Figure 2.5. Disturbance of mean with variance and eigendeformations

Now, principal component analysis (PCA) can be applied on all deformations of training images to their respective cluster centre yielding the most important in-class deformations sorted by the corresponding eigenvalues.

It is now straightforward to apply these deformations to the cluster mean in positive and negative direction either directly, or deforming the mean and then adding/subtracting the deformed mean from the old one in order to achieve suitable cluster splitting. Figure 2.5 visually compares splitting the cluster by variance and by eigendeformations. It can be seen that adding/subtracting variance only leads to a change in contrast, where eigendeformations model in-class variability and thus should lead to a better representation of the training data in the different clusters.

Chapter 3

Support Vector Machines

SVMs are a modern, well understood classifier [Schölkopf and Smola, 2002]. Instead of training a statistical model from which class posteriors can be derived SVMs directly model the decision boundary. The training of the SVMs is known to be convex and thus guaranteed to always find the optimal solution, and it is trained using maximum margin to avoid overfitting. The discriminant function of an SVM distinguishes between the classes -1 and 1 and is commonly defined as follows:

$$r(X) \quad = \quad \operatorname{sgn}\left\{\sum_{v_i \in \mathcal{S}} \alpha_i K(X, v_i) + \beta\right\}. \tag{3.1}$$

Here, $r(X)$ is the class label assigned to observation X, $v_i \in S$ is the set of support vectors (SVs) with weighting coefficients α_i defining the separating hyperplane, K is a kernel function and β is bias term. The resulting quadratic problem is usually solved using Platts SMO algorithm (Platt [1998]).

3.1 Kernel types

SVMs project data into a high-dimensional space where usually a separating hyperplane can be found. Instead of doing that explicitly, SVMs make use of the so called kernel-trick which allows to perform all necessary computations in the original space while implicitly performing the operation in higher dimensional space. Any kernel satisfying Mercer's theorem [Mercer and Forsyth, 1909] like the following commonly used kernels is able to use this trick [Schölkopf, 1997]:

- linear kernel:
$$K_{\text{lin}}(X, V) = X'V \tag{3.2}$$
- polynomial kernel:
$$K_{\text{poly}}(X, V) = (X - V)^d \tag{3.3}$$
- radial basis function (RBF)-kernel:
$$K_{\text{rbf}}(X, V) = \exp(-\gamma||X - V||^2) \tag{3.4}$$

The RBF-kernel is most commonly used and is able to separate most data well, therefore leading to good classification results. The RBF-kernel is based upon the Euclidean distance, which is not invariant to domain specific global transformations. In Chapter 2.1, it has been shown that replacing the Euclidean distance with the IDM distance allows for this kind of invariance. The next section therefore presents the new IDM-kernel for SVMs.

3.1.1 IDM-Kernel

Replacing distances in several kernels has been investigated before [Haasdonk and Bahlmann, 2004, Haasdonk and Keysers, 2002], but so far, the IDM has not been incorporated despite being a simple and good model for dealing with invariance in images. The basic idea is to exchange the Euclidean distance in the RBF kernel with the IDM distance as follows:

$$K_{\text{idm}}(X, V) = \exp\left(-\gamma \cdot d_{\text{idm}}(X, V)\right) \tag{3.5}$$

The corresponding kernel matrix is not symmetric, and thus the SVM is not guaranteed to converge to the global optimum anymore. This problem can be circumvented by replacing the simple IDM distance between X and V by the average of both deformation directions.

$$K_{\text{idm}}(X, V) = \exp\left(-\gamma \frac{d_{\text{idm}}(X, V) + d_{\text{idm}}(V, X)}{2}\right) \tag{3.6}$$

However, the resulting kernel matrix is not necessarily positive definite since a perfect match between non-identical images can be found theoretically, violating the identity constraints and triangle inequality.

A training algorithm for SVM with indefinite kernels has been proposed by Luss and D'Aspremont [2008]. This algorithm is not used for the proposed kernel, since it has been shown by Haasdonk [2005] that SVM can handle indefinite kernel functions.

3.2 Multiclassing

The decision rule presented in Section 3 discriminates between classes -1 and 1, though in real-world tasks it is necessary to discriminate between a larger number of classes. The following section shortly presents two common methods to cope with this so called multi-class problem.

3.2.1 One against the rest

The first and most basic approach to implement multiclassing for SVMs is the one against the rest (OATR) approach [Hsu and Lin, 2002]. One SVM is trained per class to discriminate this class versus all other classes leading to C classifiers with C denoting the number of classes present in training data. The resulting decision can be computed jointly, where the same SV can have a different weight for different classes which can even be negative.

$$r(X) = \arg\max_c \left\{ \sum_{v_i \in \mathcal{S}_c} c\alpha_i K(X, v_i) + \beta_c \right\}. \tag{3.7}$$

3.2.2 One against one

In the one against one (OAO) approach first presented by Knerr et al. [1990], one classifier is trained for each pair of classes. This results in $C(C-1)/2$ classifiers to be trained. To classify a new observation, each of these classifiers is applied and votes for the 'winning' class. Then, the class with the highest number of votes is chosen. Let $r_{(c,c')}(x)$ be the decision rule for the classifier discriminating between classes c and c', then the decision rule can be written as

$$r(X) = \arg\max_c \left\{ \sum_{c \neq c'} \delta(c, r_{(c,c')}(X)) \right\} \tag{3.8}$$

$$= \arg\max_c \left\{ \sum_{c \neq c'} \delta \left(c, \arg\max_{k \in \{-1,1\}} \left\{ \sum_{v_i \in S_{(c,c')}} k\alpha_i K(X, v_i) + \beta_{(c,c')} \right\} \right) \right\}. \tag{3.9}$$

This decision rule has the downside that, in particular for moderate numbers of classes, it is common that two classes have the same numbers of votes and that therefore the decision is not clear.

3.3 Efficient implementation

Despite applying the kernel trick, evaluating the respective kernel might still be computationally expensive. This is especially true for non-trivial distance measures like the tangent distance or the IDM. Since finding optimal parameters like γ is crucial for classification performance, it is necessary to find effective means of computing the kernel functions. This is fairly straightforward for the proposed IDM kernel, where all IDM distances can be precomputed and the remaining evaluation of the quasi-RBF kernel is fairly cheap, resulting in the ability to efficiently estimate parameters for example in grid search using cross validation (CV).

Chapter 4

Discriminative Training of Generative Models

Given generative model such as the GMDs described in Chapter 2.2.2, it is a common approach to further enhance classification performance by re-training the obtained model discriminatively. Generative models are trained according to the maximum likelihood (ML) criterion on the emission probabilities, while the discriminative training optimises the maximum mutual information (MMI) criterion, which is basically a ML on the class posteriors. One big advantage is the usage of out-of-class data in the discriminative training procedure compared to modelling each class separately as in the generative training. The basic principles have been described in Schlüter [2000], and have been deployed successfully in many applications. Dahmen et al. [1999] use discriminatively trained GMDs for OCR, and achieve superior results compared to ML-trained GMDs, especially for low numbers of densities per class.

When training a model generatively, the ML criterion is optimised so the best set of parameters $\hat{\theta}_{ML}$ maximises the product of the emission probabilities over all observation given the respective class and class prior. On the other hand, the MMI criterion optimises parameters with regard to the product of the class posteriors, thus training class separability.

$$\hat{\theta}_{ML} = \arg\max_{\theta} \left\{ \prod_{n} p_\theta(c) p_\theta(x_n|c_n) \right\} \quad \hat{\theta}_{MMI} = \arg\max_{\theta} \left\{ \prod_{n} p_\theta(c_n|x_n) \right\}. \quad (4.1)$$

The previous definition of the MMI-criterion can be rewritten as a function of the set of parameters Θ and transferred to the logarithmic space in order to ease computability, which is possible because the logarithm is monotonous and thus does not change the maximum of the function.

$$F(\Theta) = \sum_{n} \log(p_\Theta(c_n|X_n)) \quad (4.2)$$

In this chapter, a common algorithm for discriminative training is presented. Then, the IDM is reformulated in a probabilistic manner in order to allow its

incorporation in the discriminative training procedure. Finally, update rules for some of the parameters are given and some special topics concerning the parameters are covered.

4.1 Gradient Descent Training

A common approach to train a model given a criterion is to compute the gradient of the criterion with regard to the model and to update the model parameters accordingly. This method is described in Boyd and Vandenberghe [2004] and can be used either directly or in more advanced training algorithms like limited memory BFGS method (LBFGS) [Liu and Nocedal, 1989]. It has the advantage of being conceptually simple, but has the downside that it only finds local minima. This is sufficient given a convex problem, where the only local minimum equals the global minimum.

4.1.1 Derivatives

The computation of the derivatives necessary for the gradient calculation is straightforward and uses standard differentiation techniques and mathematical theorems. The posterior probability is reformulated as a normalised product of class prior and emission probability. Then, the logarithm of the fraction is rewritten as the difference between the logarithm of the numerator and the logarithm of the denominator and the derivatives of the logarithms are computed.

$$\frac{\partial}{\partial\Theta}F(\Theta) = \frac{\partial}{\partial\Theta}\sum_n \log p_\Theta(c_n|X_n) \tag{4.3}$$

$$= \sum_n \frac{\partial}{\partial\Theta}\left(\log\frac{p_\Theta(c_n)p_\Theta(X_n|c_n)}{\sum_{c'}p_\Theta(c')p_\Theta(X_n|c_n)}\right) \tag{4.4}$$

$$= \sum_n \frac{\partial}{\partial\Theta}\left(\log p_\Theta(c_n) + \log p_\Theta(X_n|c_n) - \log\sum_{c'}p_\Theta(c')p_\Theta(X_n|c')\right) \tag{4.5}$$

$$= \sum_n \left(\frac{\partial}{\partial\Theta}\log p_\Theta(c_n) + \frac{\partial}{\partial\Theta}\log p_\Theta(X_n|c_n) - \frac{\sum_{c'}\frac{\partial}{\partial\Theta}p_\Theta(c')p_\Theta(X_n|c')}{\sum_{c'}p_\Theta(c')p_\Theta(X_n|c')}\right) \tag{4.6}$$

The last term then simplifies to

$$\frac{\sum_{c'}\frac{\partial}{\partial\Theta}p_\Theta(c')p_\Theta(X_n|c')}{\sum_{c'}p_\Theta(c')p_\Theta(X_n|c')} = \sum_{c'}p_\Theta(c'|X_n)\frac{\partial}{\partial\Theta}\log p_\Theta(X_n|c_n) \tag{4.7}$$

which makes it possible to rewrite Equation (4.6) as:

$$\frac{\partial}{\partial\Theta}F(\Theta) = \sum_n \left(\frac{\partial}{\partial\Theta} \log p_\Theta(c_n) + \sum_{c'} \left[\delta(c', c_n) - p_\Theta(c|X_n)\right] \frac{\partial}{\partial\Theta} \log p_\Theta(X_n|c_n) \right), \tag{4.8}$$

where $\delta(c', c_n)$ denotes the Kronecker delta. It is now possible to compute derivatives for all parameters in the parameter set Θ given the according derivatives for the emission probability $\log p_\Theta(X_n|c_n)$. Then, parameters can be updated iteratively in the following way:

$$\Theta' = \Theta - \varepsilon \frac{\partial}{\partial\Theta} F(\Theta), \tag{4.9}$$

where ε is a step width which is set so that the criterion improves in each iteration.

4.2 Probabilistic IDM

In order to allow for discriminative training with GMDs including IDM, it is necessary to reformulate the IDM in a probabilistic way, so the emission probabilities can be differentiated. The according probabilistic notation was first described by Keysers [2006], but has never been used.

Here, we formulate the emission probability of an observation X given a prototype μ_c of class c as a zero order HMM, where the states are the pixels (ij) of X and no intra-pixel dependencies are used. Instead, the emission probability of each pixel X_{ij} given the prototype is modelled as sum over the probabilities of all possible matchings. These matchings can be modelled by Gaussian distributions with means μ_{cxy} and pooled diagonal covariance matrix. Additionally, a deformation prior $p_c(xy|ij)$ is used comparable to the deformation penalties described in Gollan [2003]. Since the IDM restricts the warp range, the prior for deformations outside the warp range is zero while it is initialised inversely proportional to a scaled Euclidean distance between pixel positions (ij) and (xy) inside the allowed warping range.

$$p(X|\mu_c) = \prod_{ij} p(X_{ij}|\mu_c) \tag{4.10}$$

$$= \prod_{ij} \sum_{xy} p_c(xy|ij)\; p(X_{ij}|\mu_{xy}) \tag{4.11}$$

$$= \prod_{ij} \sum_{xy} p_c(xy|ij)\; \mathcal{N}(X_{ij}|\mu_{cxy}, \sigma^2) \tag{4.12}$$

$$= \prod_{ij} \frac{1}{\sqrt{2\pi}\sigma} \sum_{xy} \exp\left[\log p_c(xy|ij) - \frac{1}{2\sigma^2}(X_{ij} - \mu_{cxy})^2\right] \tag{4.13}$$

Using maximum approximation in Equation (4.13) this simplifies to the definition of the IDM in generative models as presented in Chapter 2.2. Here, the Euclidean distance of the Gaussian is replaced by the sum over the distance of all pixels to their best matching pixel in the reference image.

$$p(X|\mu_c) \cong \prod_{ij} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[\max_{xy}\left\{\log p_c(xy|ij) - \frac{1}{2\sigma^2}(X_{ij} - \mu_{cxy})^2\right\}\right] \tag{4.14}$$

$$= \frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^{I \cdot J}} \exp\left[-\sum_{ij} \min_{xy}\left\{-\log p_c(xy|ij) + \frac{1}{2\sigma^2}(X_{ij} - \mu_{cxy})^2\right\}\right] \tag{4.15}$$

However, since the computation of gradients is unclear given the minima in the formula, we use the exact version of the emission probability for further calculations, which leads to increased time complexity, but finer parameter estimations since not only the best matching warping has influence on the gradient, but instead all possible matches are used.

4.2.1 Parameter updates

The calculation of the derivatives for most of the parameters is straightforward. In the proposed framework, all parameters of the generative model are trained discriminatively, which results in a diagonal covariance matrix per class and in class-specific deformation priors. These priors have been pooled during the training of the generative model because they could not be trained using the EM algorithm.

As introduced in Chapter 2, the performance of the IDM heavily depends on the features used, where using Sobel gradients and local context are well-known approaches in OCR and medical image annotation [Deselaers and Ney, 2008]. So far, Sobel gradients and local context have been used inside the (pixel-)distance computation $g(A, ij, B, xy)$. Now, given the need to calculate derivatives of the probability distribution, these feature expansions have to be modelled explicitly.

While it is straightforward to incorporate Sobel features by replacing each pixel by its respective horizontal and vertical Sobel gradient, the usage of local context is not that simple. This stems from the fact that the local context areas of neighbouring pixels overlap and thus are interconnected. It is possible to fully take into account this kind of overlap, but preliminary experiments have shown that allowing the local area to fully overlap leads to bad results. This is probably due to good matching pixel warps having influence on all parameters in the area of the pixel warped to, which effectively neglects the IDM design.

Therefore, mild restrictions are applied updating only the parameter warped to (the pixel in the centre of the local context area), while considering all of the surrounding pixels for probability estimation. The derivative for mean μ_c at position xy given full interconnection is as follows

$$\frac{\partial}{\partial \mu_{cxy}} \log p(X, c) = \sum_{ij} \frac{1}{Z_{ij}} \sum_{x',y'} \frac{p(x'y'|ij)}{2\sqrt{\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(X_{ij} - \mu_{cx'y'})^2\right) \cdot \sum_{l_1,l_2} \delta(x' + l_1, x)\delta(y' + l_2, y)\frac{1}{\sigma^2}(X_{i+l_1,j+l_2} - \mu_{cx'+l_1,y'+l_2}), \quad (4.16)$$

and reduces to

$$\frac{\partial}{\partial \mu_{cxy}} \log p(X, c) = \sum_{ij} \frac{1}{Z_{ij}} \sum_{x',y'} \frac{p(x'y'|ij)}{2\sqrt{\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(X_{ij} - \mu_{cx'y'})^2\right) \cdot \left(\frac{1}{\sigma^2}(X_{ij} - \mu_{cx'y'})\right) \quad (4.17)$$

using only the updates of the centre. Here, the sum over l_1, l_2 accounts for the local area used, hence $-\text{lc} \leq l_1, l_2 \leq \text{lc}$ with lc defining the amount of local context used and Z_{ij} is a normalising factor.

Chapter 5

Log-Linear Models with IDM

It is well known that discriminative models often outperform generative models in terms of classification performance. So far, SVMs have been fused with the IDM and generative models have been trained using the IDM with added discriminative training for parameter tuning. This has been done by optimising the parameters of the generative distribution with respect to a discriminative criterion, the MMI criterion, where the class posterior probabilities are modelled as normalised products of prior and emission probability.

Log-linear models (LLMs) have been known since the 1970s [Darroch and Ratcliff, 1972] and model the class posteriors directly:

$$p(c|x) = \frac{\exp(\alpha_c + \lambda_c^T x)}{\sum_{c'=1}^{C} \exp(\alpha_{c'} + \lambda_{c'}^T x)}. \tag{5.1}$$

The parameters α_c, λ_c are optimised according to the discriminative criterion during the training. LLMs have been successfully applied in a variety of tasks, *e.g.* in OCR [Keysers, 2006], medical image retrieval [Deselaers et al., 2007] and image object recognition [Deselaers et al., 2006, 2005]. The training of LLMs is a convex problem and thus it is guaranteed to find the global optimum. Several algorithms like generalised iterative scaling (GIS) [Darroch and Ratcliff, 1972] and LBFGS [Liu and Nocedal, 1989] exist which can find the global optimum for the presented model.

In this chapter, we present two approaches to fuse the IDM with LLMs. First, we describe an initial two-step approach which trains class-dependent diagonal covariance matrices using a LLM and GDs trained with IDM. Then, it will be shown how the probabilistic definition of the IDM can be rewritten in log-linear form and how GDs with IDM can be transformed into LLMs with IDM. Finally, several topics regarding the parameters and different initialisation schemes are discussed, leading to an enclosed model, for which the loss of convexity and the increased training complexity are discussed.

5.1 First Approach

In a more general definition of the LLM presented by Berger et al. [1996], where it is shown that the performance of the LLM depends on a feature function $f_i(X, c)$ suitable for classification. The resulting model then has the following form:

$$p(c|X) = \frac{\exp(\alpha_c + \sum_i \lambda_{ci} f_i(X, c))}{\sum_{c'} \exp(\alpha_{c'} + \sum_i \lambda_{c'i} f_i(X, c'))}. \tag{5.2}$$

Commonly, this function is used to define first- or higher-order features. Here, we propose to use a feature function which incorporates the IDM.

Considering the emission probability of a GD with IDM

$$p(X|c) \propto \exp\left(- \sum_{ij} \frac{\min_{xy} \{g(X, ij, \mu_c, xy)\}}{\sigma^2_{cij}} \right), \tag{5.3}$$

the parameters μ_c and σ_{cij} are obtained according to ML training as described in Chapter 2.2.1. Therefore, a feature function $f_{ij}(X, c)$ can be defined as the distance between pixel (ij) of image X and the best matching pixel of the prototype μ_c.

$$f_{ij}(X, c) = \min_{xy} \{g(X, ij, \mu_c, xy)\} \tag{5.4}$$

In the resulting form of the LLM

$$p(c|X) = \frac{\exp(\alpha_c + \sum_{ij} \lambda_{cij} f_{ij}(X, c))}{\sum_{c'} \exp(\alpha_{c'} + \sum_{ij} \lambda_{c'ij} f_{ij}(X, c'))}, \tag{5.5}$$

it can be seen that the parameters λ_{cij} can be interpreted as proportional to a class-dependent variance

$$\lambda_{cij} = \frac{1}{\sigma^2_{cij}}, \tag{5.6}$$

so the training of model (5.5) can be seen as a discriminative training of the variances of the generative model. One big advantage of this approach is that using arbitrary complex features like Sobel and local context is straight forward by extending the distance function $g(X, ij, \mu_c, xy)$. Most importantly, the GD is easily trained and the training of the LLM in the second step is convex. Furthermore, it is possible to easily extend the model to mixtures by using more than one mean per class, effectively increasing the number of parameters to be optimised.

The aim of this work is to fuse deformations and discriminative models. This means that all parameters of either model should be trained by the combined model,

which is not the case in the first approach presented here. In the next section, we present a novel model which fuses the IDM with LLMs, and is therefore able to train variances, deformation penalties, alignments and means in one single model.

5.2 Log-Linear Model

Since the incorporation of the IDM in Gaussian densities has been explored thoroughly, it is possible to exploit common knowledge about relations between GDs and LLMs in order to obtain a log-linear model with IDM. We shortly revise the relation between the models in absence of a transformation invariant distance measure, and then present the novel model with IDM.

5.2.1 Relationship of GDs and LLMs

Keysers et al. [2002a] have shown that a GD classifier can be transformed into an LLM. The decision rule of a single Gaussian classifier (with pooled covariance matrix, uncorrelated and variance normalised data) is

$$\begin{aligned} r(X) &= \arg\max_c \{p(c|X)\} && (5.7)\\ &= \arg\max_c \left\{ \frac{p(c)p(X|c)}{p(X)} \right\} && (5.8)\\ &= \arg\max_c \left\{ \frac{p(c)}{p(X)} \frac{1}{(2\pi\sigma)^{D/2}} \exp\left(-\frac{1}{2} \frac{||X-\mu_{ci}||}{\sigma^2} \right) \right\} && (5.9) \end{aligned}$$

and can be transformed into a log-linear classifier of the following form

$$r(X) = \arg\max_c \left\{ \frac{\exp(\alpha_c + \lambda_c^T X)}{\sum_{c'} \exp(\alpha_c + \lambda_{c'}^T)} \right\} \tag{5.10}$$

by rewriting

$$\begin{aligned} &\frac{1}{(2\pi\sigma)^{D/2}} \exp\left(-\frac{1}{2} \frac{||X-\mu_{ci}||}{\sigma^2} \right)\\ &= \exp\left(-\frac{D}{2}\log(2\pi\sigma) - \frac{1}{2\sigma}\mu_c^T\mu_c + \frac{1}{2\sigma}2\mu_c^T X - \frac{1}{2\sigma}X^T X \right) && (5.11) \end{aligned}$$

and choosing

$$\begin{aligned} \alpha_c &= -\frac{D}{2}\log(2\pi\sigma) - \frac{1}{2\sigma}\mu_c^T\mu_c && (5.12)\\ \lambda_c &= \frac{1}{2\sigma}2\mu_c && (5.13) \end{aligned}$$

5.2.2 Obtaining an LLM with IDM

We show a similar approach to derive an LLM incorporating IDM from a Gaussian density with IDM. Consider the probabilistic interpretation of the IDM given in Equation 4.13:

$$p(X|c) = \prod_{ij} \sum_{xy} p(xy|ij)\ p(X_{ij}|c). \tag{5.14}$$

Using Bayes' theorem, it is possible to derive the class-posterior

$$p(c|X) = \frac{p(c) \prod_{ij} \sum_{xy} \{p(xy|ij)\ p(X_{ij}|c)\}}{\sum_{c'} p(c') \prod_{ij} \sum_{xy} \{p(xy|ij)\ p(X_{ij}|c)\}}, \tag{5.15}$$

which can be extended to describe the probability of a set of pixel alignments $(xy)_{11}^{IJ}$ as well:

$$p\left(c, (xy)_{11}^{IJ} | X_{11}^{IJ}\right) = \frac{p(c) \prod_{ij} p((xy)_{ij}|(ij), c) \cdot p(X_{ij}|c, (xy)_{ij})}{\sum_{c} \sum_{\left[(xy)_{11}^{IJ}\right]} \prod_{ij} p((xy)_{ij}|(ij), c) \cdot p(X_{ij}|c, (xy)_{ij})} \tag{5.16}$$

Note that it is necessary to normalise the probability with regard to all possible alignments. We now model $p(X_{ij}|c, (xy)_{ij})$ as GDs and obtain

$$p\left(c, (xy)_{11}^{IJ} | X_{11}^{IJ}\right) = \frac{1}{Z(X)} p(c) \prod_{ij} p((xy)_{ij}|(ij), c) \cdot \mathcal{N}(X_{ij}|\mu_{c(xy)_{ij}}, \sigma) \tag{5.17}$$

$$= \frac{1}{Z(X)} p(c) \prod_{ij} p((xy)_{ij}|(ij), c) \cdot \frac{1}{(2\pi\sigma)^{D/2}} \exp\left(-\frac{1}{2} \frac{||X - \mu_{c(xy)_{ij}}||}{\sigma^2}\right) \tag{5.18}$$

where $Z(X)$ is the normalising factor. We can now rewrite Equation (5.18) as

$$p\left(c, (xy)_{11}^{IJ} | X_{11}^{IJ}\right) = \frac{1}{Z(X)} \exp\Bigg(\log\left(p(c)\right) + \sum_{ij} \log\left(p((xy)_{ij}|(ij), c)\right) - \frac{D}{2} \log(2\pi\sigma) - \frac{1}{2\sigma} \mu_{c(ij)_{xy}}^T \mu_{c(ij)_{xy}} + \frac{1}{2\sigma} 2\mu_{c(ij)_{xy}}^T X_{ij} - \frac{1}{2\sigma} X_{ij}^T X_{ij} \Bigg). \tag{5.19}$$

Analogously to Equation (5.12), we can now obtain a log-linear model by choosing

$$\alpha_c = \log\left(p(c)\right) - \frac{D}{2}\log\left(2\pi\sigma\right) \tag{5.20}$$

$$\alpha_{c(xy)_{ij}ij} = \log\left(p((xy)_{ij}|(ij),c)\right) - \frac{1}{2\sigma}\mu^T_{c(xy)_{ij}}\mu_{c(xy)_{ij}} \tag{5.21}$$

$$\lambda_{c(ij)_{xy}} = \frac{1}{2\sigma}2\mu_{c(xy)_{ij}}, \tag{5.22}$$

which results in the following model:

$$p\left(c,(xy)^{IJ}_{11}|X^{IJ}_{11}\right) = \frac{1}{Z(X)}\exp\left(\alpha_c + \sum_{ij}\alpha_{c(xy)_{ij}ij} + \lambda^T_{c(xy)_{ij}}X_{ij}\right) \tag{5.23}$$

With this model, all parameters of the IDM can be trained by the same principles as the original log-linear model. Note that the $\alpha_{c(xy)_{ij}ij}$ partly model the deformation prior and thus depend on the originating pixel position (i,j) and the pixel alignment $(xy)_{ij}$, which leads to $D\cdot(2\cdot W+1)^2$ parameters given dimensionality D and maximum warprange W. Although the convexity of the resulting training problem is not given, simple methods like gradient descent can be applied to find a suitable set of parameters.

An example of trained λs using regular LLMs compared to λs obtained by a LLMs with IDM is given in Figure 5.1. Both models have been initialised with zeros, and it can be seen that the model trained w.r.t. deformations allows for edge detection since the Sobel gradients are visible in the trained parameters.

Parameter Overview

Note that in Equations 5.14-5.23 a multivariate distribution is assumed, although all parameters are scalars if the original image representation is preserved. However, it has already been mentioned in Chapter 2 that it is necessary to use local features in order to achieve good results with the IDM. In order to clarify the dimensionality of the parameters, Table 5.1 gives an overview of the influences of different feature settings on the dimensionality of the parameters. The overall number of parameters depends on the maximum warprange as well. This stems from the fact that the α_{cijxy} model the deformation prior and parts of the distance and thus depend on originating pixel position and warping target as well. Several approaches to tying these parameters are presented in Section 5.2.6.

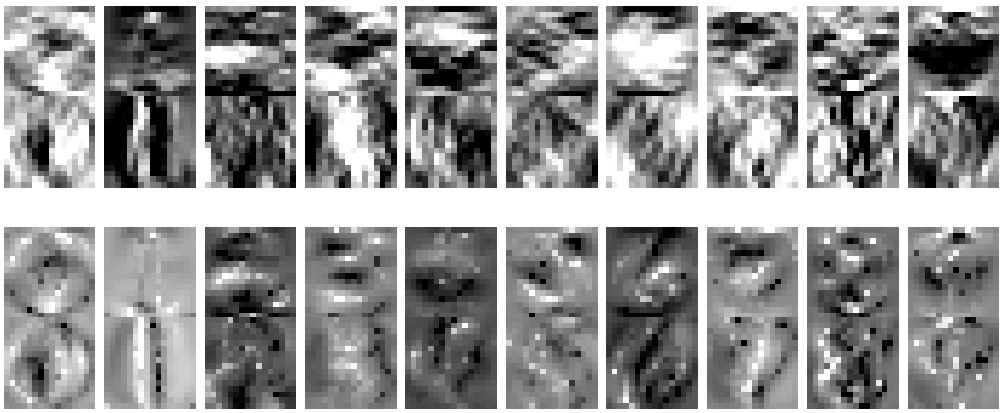

Figure 5.1. Comparison of λs with horizontal and vertical Sobel components. The top row show the parameters trained using a regular LLM and in the bottom row λs trained by a LLM with IDM are depicted. Each column corresponds to the parameters of one class.

Table 5.1. Parameters of the log-linear model. Here, L defines the size of the local area.

	X_{ij}	λ_{cxy}	α_{cijxy}	Λ_{cxy}
No local context (LC)	1	1	1	1
Sobel	2	2	2	2
Gray context ($L \times L$)	L^2	L^2	1	(diag) L^2
Sobel context ($L \times L$)	$2L^2$	$2L^2$	2	(diag) $2L^2$

5.2.3 Maximum Approximation

Revisiting Equation 5.23 we can see that calculating the full denominator as given by:

$$p\left(c, (xy)_{11}^{IJ} | X_{11}^{IJ}\right) = \frac{\exp\left(\alpha_c + \sum_{ij} \alpha_{c(xy)_{ij}ij} + \lambda^T_{c(xy)_{ij}} X_{ij}\right)}{\sum_{c'} \sum_{\left[(xy)_{11}^{IJ}\right]} \exp\left(\alpha_{c'} + \sum_{ij} \alpha_{c'(xy)_{ij}ij} + \lambda_{c'(xy)_{ij}}\right)} \tag{5.24}$$

is not feasible by means of computational complexity since the total amount of different matchings between an image and a prototype is exponential in the number of pixels. If we allow a maximum warprange of 2, we have 25 possible warping positions for each pixel (for simplicity, we consider the images to be padded). This means that for an image with 256 pixels there are 25^{256} different alignments. To circumvent this problem, maximum approximation is applied in the denominator,

which leads to

$$p\left(c,(xy)_{11}^{IJ}|X_{11}^{IJ}\right) \approx \frac{\exp\left(\alpha_c + \sum_{ij}\alpha_{c(xy)_{ij}ij} + \lambda_{c(xy)_{ij}}^T X_{ij}\right)}{\sum_{c'}\max_{\left[(xy)_{11}^{IJ}\right]}\exp\left(\alpha_{c'} + \sum_{ij}\alpha_{c'(xy)_{ij}ij} + \lambda_{c'(xy)_{ij}}^T X_{ij}\right)}. \tag{5.25}$$

Since the exponential function is monotonous and $\alpha_{c'}$ does not depend on the alignment $\left[(xy)_{11}^{IJ}\right]$ we can rewrite Equation (5.25)

$$p\left(c,(xy)_{11}^{IJ}|X_{11}^{IJ}\right) \approx \frac{\exp\left(\alpha_c + \sum_{ij}\alpha_{c(xy)_{ij}ij} + \lambda_{c(xy)_{ij}}^T X_{ij}\right)}{\sum_{c'}\exp\left(\alpha_{c'} + \max_{\left[(xy)_{11}^{IJ}\right]}\sum_{ij}\alpha_{c'(xy)_{ij}ij} + \lambda_{c'(xy)_{ij}}^T X_{ij}\right)} \tag{5.26}$$

$$= \frac{\exp\left(\alpha_c + \sum_{ij}\alpha_{c(xy)_{ij}ij} + \lambda_{c(xy)_{ij}}^T X_{ij}\right)}{\sum_{c'}\exp\left(\alpha_{c'} + \sum_{ij}\max_{(xy)_{ij}}\left[\alpha_{c'(xy)_{ij}ij} + \lambda_{c'(xy)_{ij}}^T X_{ij}\right]\right)}. \tag{5.27}$$

The log-linear model described by this formula can be trained using alternating optimisation, as described in Section 5.2.4. Since the IDM generally bases on the best match given a pixels possible warping targets, we can formulate an approximation of the class posterior probability using maximum approximation in the numerator as well.

$$p(c|X) \approx \frac{\exp\left(\alpha_c + \sum_{ij}\max_{(xy)_{ij}}\left[\alpha_{c(xy)_{ij}ij} + \lambda_{c(xy)_{ij}}^T X_{ij}\right]\right)}{\sum_{c'}\exp\left(\alpha_{c'} + \sum_{ij}\max_{(xy)_{ij}}\left[\alpha_{c'(xy)_{ij}ij} + \lambda_{c'(xy)_{ij}}^T X_{ij}\right]\right)} \tag{5.28}$$

5.2.4 Alternating Optimisation

Equation (5.26) encloses two optimisation problems. The first problem is to estimate the best alignment for each observation and the second problem aims at finding the best possible set of parameters $\hat{\Theta}$ according to the MMI-criterion. Bezdek and Hathaway [2003] have shown that given the convexity of each subproblem, then the resulting alternating approximation problem is convergent. Therefore, we use the following training algorithm in order to find optimal parameters for the proposed log-linear model with IDM.

- Step A: Parameter estimation with given alignment $\left[(xy)_{11}^{IJ}\right]_n$

$$\hat{\Theta} = \arg\max_{\Theta} \left\{ \sum_n \log p_\Theta(c_n, \left[(xy)_{11}^{IJ}\right]_n | X_n) \right\} \tag{5.29}$$

- Step B: Re-align with given parameter set Θ

$$\left[(xy)_{11}^{IJ}\right]_n := \arg\max_{\left[(xy)_{11}^{IJ}\right]'_n} \left\{ p_\Theta(c_n, \left[(xy)_{11}^{IJ}\right]'_n | X_n) \right\} \tag{5.30}$$

The convexity of each subproblem is not investigated theoretically here, but experiments have shown that this algorithm leads to very good and robust results (see Section 6.3.3).

5.2.5 Second order features

It is commonly known that using second order features in log-linear models can increase performance significantly. This approach equals using full covariance matrices in Gaussian densities, although the training of the second order features in the LLMs is usually more robust due to the simplicity of the model and better training algorithms.

We can define two cases for using second order features in LLMs with IDM:

1. Second order features are used equivalently to full covariance matrices in GDs.
2. Covariances are modelled locally for each pixel with according local context.

In the following both approaches are presented in detail.

Full covariance

In order to obtain a LLM with second order features, we consider the emission probability to be modelled by a Gaussian with full covariance matrix:

$$\mathcal{N}(X|\mu_c, \Sigma_c) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp\left(-\frac{1}{2} (X - \mu_c) \Sigma^{-1} (X - \mu_c) \right) \tag{5.31}$$

By rewriting, we obtain

$$\mathcal{N}(X|\mu_c, \Sigma_c) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp\left(-\frac{1}{2} \sum_{ij} \sum_{i'j'} (X_{ij} - \mu_{cij}) \Sigma^{-1}_{iji'j'} \left(X_{i'j'} - \mu_{ci'j'} \right) \right). \tag{5.32}$$

Now, we consider alignments $(xy)_{ij}$ for each pixel (ij), Equation (5.32) changes to

$$\frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp\left(-\frac{1}{2}\sum_{ij}\sum_{i'j'}\left(X_{ij}-\mu_{c(xy)_{ij}}\right)\Sigma^{-1}_{(xy)_{ij}(x'y')_{i'j'}}\left(X_{i'j'}-\mu_{c(x'y')_{i'j'}}\right)\right), \tag{5.33}$$

and can be rewritten in log-linear form as

$$\exp\left(\alpha_c + \sum_{ij} X_{ij}\lambda_{c(xy)_{ij}} + \sum_{ij}\sum_{i'j'} X_{ij}X_{i'j'}\Lambda_{c(xy)_{ij}(x'y')_{i'j'}}\right) \tag{5.34}$$

applying similar assignments as given in Equation (5.20), although omitting parameters $\alpha_{c(xy)_{ij}ij}$ for simplicity. Parameter estimation for the full model can be formulated analogously to the first-order model, but the computation of the alignments is NP-hard since the maximisation of Equation (5.34) with regard to the alignment requires simultaneously maximising all pixel alignments, which effectively leads to a second order deformation model. We have already shown in Section 5.2.3 that the number of possible alignments for an image with 256 pixels and maximum warp range 2 is 25^{256}, which is computationally not feasible.

Local Covariance

If local context is used, the pixels of each image are represented as vectors of the pixel values of their respective local context as described in Section 5.2.2. Then, the emission probability can be modelled as Gaussians on the pixel level with full covariance matrix:

$$p(X_{ij}|c,(xy)_{ij}) = \mathcal{N}(X_{ij}|\mu_{c(xy)_{ij}},\Sigma_{c(xy)_{ij}ij}) \tag{5.35}$$

$$= \frac{\exp\left(-\frac{1}{2}(X_{ij}-\mu_{c(xy)_{ij}})^T\Sigma^{-1}_{c(xy)_{ij}ij}(X_{ij}-\mu_{c(xy)_{ij}})\right)}{\sqrt{2\pi\det(\Sigma_{c(xy)_{ij}ij})}} \tag{5.36}$$

Therefore, one covariance matrix of dimension $(2\cdot l+1)^2$ considering a local area of l pixels around each pixel is trained for each possible pixel-to-pixel alignment, since the overall probability distribution given by Equation (5.28) is maximised over all possible pixel alignments. For 256-dimensional images and maximum warp range set to 2 which leads to $(2*2+1)^2$ possible matchings per pixel, $25\cdot 256 = 6400$ local covariance matrices of dimension $(2\cdot 1+1)^2 = 9$ for $l=1$ are trained, which increases the number of parameters significantly,

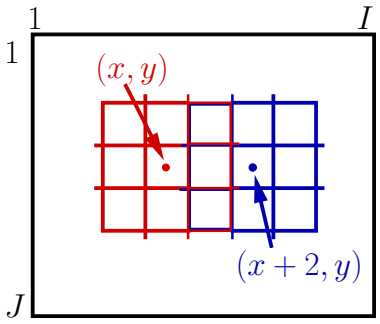

Figure 5.2. Interconnection of λs in Log-Linear Models with IDM. It can be seen that given λs at position (x, y) and $(x + 2, y)$ with a 3x3 local area, the middle row overlaps.

5.2.6 Tying of Parameters

When local context is used, the number of parameters of the log-linear model with IDM can be very high. Furthermore, the total amount of deformation priors is high as well, which leads to several ideas of tying the parameters in order to avoid too fast overfitting. These approaches are presented in the following sections.

Lambdas

So far, feature vectors for observations and parameters alike has been simply extended in the presence of local context. Considering a 3x3 area, which is a well known standard for the USPS task, the number of parameters multiplies by 9. However, given the knowledge about the underlying image domain, these parameters overlap significantly which is shown in Figure 5.2, which is similar to the parameter overlap discussed in Section 4.2.1, where the means of the Gaussian densities overlap the same way as the $\lambda_{c(xy)_{ij}}$ in the log-linear model with IDM.

In order to reduce the number of parameters on the one hand and to exploit the domain knowledge on the other, we propose three methods for dealing with the parameter overlap:

1. Untied $\lambda_{c(xy)_{ij}}$: In this case, each pixel is modelled by its local area separately, which leads to the described enlargement of parameter space.

2. Fully tied $\lambda_{c(xy)_{ij}}$: Here, each pixel is modelled only once, which leads to a local context area to be defined as collection of the other present parameters. This approach is depicted in Figure 5.2.

3. Tied $\lambda_{c(xy)_{ij}}$, centre updates: In the last case, full overlap for probability estimation is assumed, but equivalently to the approach presented in Chapter

4.2.1 only the centre pixel of each local area is updated by its local gradient.

Alphas

The parameters $\alpha_{cij(xy)_{ij}}$ partly model the deformation priors. So far, these priors have been modelled as a table with one entry for each possible pixel-to-pixel mapping. Given a maximum warp range of 2, 25 possible matchings per pixel exist (again, we consider the images to be padded), which leads to $256 \cdot 25 = 6400$ deformation priors per class given 256-dimensional images (*e.g.* USPS). This number is large compared to the number of λs, so we propose four different approaches to tie the $\alpha_{cij(xy)_{ij}}$ and thus effectively reduce the amount of parameters:

1. Deformation prior for each pixel separately: $\alpha_{c(xy)_{ij}ij}$ as table

2. Deformation prior per deformation and direction:

$$\alpha_{c(xy)_{ij}ij} = f_c(x - i, y - j)$$

3. Deformation prior per deformation:

$$\alpha_{c(xy)_{ij}ij} = f_c(|x - i|, |y - j|)$$

4. Deformation prior per deformation distance:

$$\alpha_{c(xy)_{ij}ij} = f_c(\sqrt{(x - i)^2 + (y - j)^2})$$

5.2.7 Initialising the Log-Linear Model

We propose three different approaches to initialising the LLM, which are presented in the following:

Initialisation using GMDs with IDM

Since the log-linear model with IDM has been derived from a Gaussian density with IDM, we initialise the LLMs with the parameters obtained from fully trained GMDs as described by the rules of Equation (5.20). This leads to good alignments in the initialisation, which did not change significantly during the training process. However, having to train a generative model in order to initialise the LLM is not satisfactory, so different approaches have been investigated.

Using Log-Linear Models without IDM

One possibility to initialise a LLM with IDM is to train a LLM without IDM, and allow for deformations once the model is fully trained. Then, the model can be re-trained using the IDM. This proved feasible, although it is difficult to find good alignments without deformation priors. We propose to initialise the deformation priors proportional to the Euclidean warping distance. This approach leads to very good results, although it is necessary to re-align the pixels several times.

Zero-initialisation

Another approach is to initialise all parameters with zeros, as it is commonly done for regular LLMs. The difference with using the IDM is, that alignments have to be computed without an interpretable model. Therefore, we initialise the alignments with the original pixel position, and realign the data according to the alternating optimisation process described in Section 5.2.4.

5.3 Log-Linear Mixtures

Log-linear models can be extended to mixtures nearly exactly the same way single Gaussian classifiers can be extended to Gaussian mixtures. The basics of this extension have been presented by Duda et al. [2001], and a more recent view on the topic is given by Heigold et al. [2008]. The class posteriors of the log-linear mixture models (LLMMs) are modelled as follows:

$$p(c|x) \quad = \quad \frac{\sum_{e=1}^{E} \exp\left(\alpha_{ce} + \lambda_{ce}^T x\right)}{\sum_{c'=1}^{C} \sum_{e'=1}^{E} \exp\left(\alpha_{c'e'} + \lambda_{c'e'}^T x\right)}, \tag{5.37}$$

where E is the number of 'clusters' for each mixture. This model can be trained by the same methods applied to regular LLMs, although it should be noted that the problem is not convex anymore and thus it is not guaranteed that the global optimum can be found. The incorporation of the IDM in this model is simple since it follows exactly the same rules as the model presented in Section 5.2.2:

$$p(c|X) \approx \frac{\sum_{e=1}^{E} \exp\left(\alpha_{ce} + \sum_{ij} \max_{(xy)_{ij}} \left[\alpha_{ce(xy)_{ij}ij} + \lambda_{ce(xy)_{ij}}^T X_{ij}\right]\right)}{\sum_{c'}^{C} \sum_{e'=1}^{E} \exp\left(\alpha_{c'e'} + \sum_{ij} \max_{(xy)_{ij}} \left[\alpha_{c'e'(xy)_{ij}ij} + \lambda_{c'e'(xy)_{ij}}^T X_{ij}\right]\right)} \tag{5.38}$$

Chapter 6

Experimental Results

In this chapter, the models investigated in this thesis are evaluated experimentally. First, an overview of the databases used is given, followed by results of Gaussian models for comparison. Finally, the third section covers discriminative models.

6.1 Databases

The performance of all models has been evaluated on the well-known USPS database of handwritten digits. This database consists of 7921 training and 2007 test images, which allows for fast evaluation of investigated models and the number of published results allows for good comparability. It is generally considered a hard task due to the large variability of images in the test set. The images consist of 16x16 pixels with 1000 distinct grey values, which are originally scaled from $-1, \cdots, 1$. We normalise the images between 0 and 1 for our experiments. In Figure 6.1 some example images are shown.

Compared to realistic tasks, the USPS database is small and it is not certain whether methods performing well on small datasets scale to large datasets. To investigate this effect, additional experiments with the best models on the USPS set have been performed on the modified NIST (MNIST) database, which is also widely used. The MNIST database consists of 60000 training images and 10000 test images with a resolution of 28x28 and normalised grey values. Some example images are given in Figure 6.2.

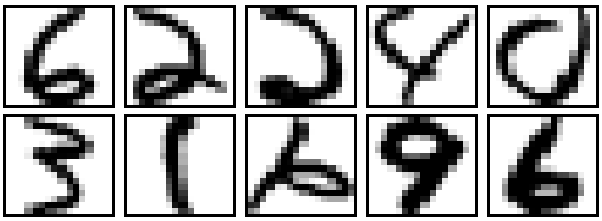

Figure 6.1. Example images of the USPS database.

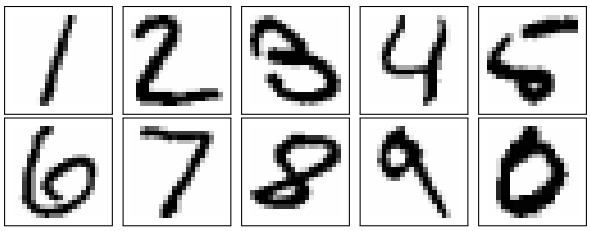

Figure 6.2. Example images of the MNIST database.

Table 6.1. Overview of relevant results on the USPS database.

Method	ER	TD	IDM
kNN	5.6%	3.0%	**2.4**%
OAO-SVM(rbf)	4.6%	3.4%	-
OATR-SVM(rbf)	4.4%	-	-
Gaussian Single Densities	16.3%	9.9%	7.7%
Gaussian Mixture Densities	7.6%	6.5%	3.5%
Log-Linear Model	8.2%	-	-
Log-Linear Mixture Model	3.7%	-	-

In Table 6.1, an overview of already published results on the USPS database with relevance to this thesis is given. The model-free approaches using IDM published by Keysers et al. [2007] perform good as far as recognition accuracy is concerned but are computationally very expensive because of the number of deformations to be computed. This is especially true for the evaluation, but no training is required. The inverse holds for most of the other models. SVMs require a complex training, but reduce the number of necessary comparisons in testing by finding suitable instances of the training data as support vectors. Results published by Haasdonk and Keysers [2002] show good performance with a tangent-distance [Keysers et al., 2002b] based kernel, which provides invariance to some transformations like scaling and rotation. GMDs and LLMMs further reduce model complexity, though depending on the number of densities used. Good results have been achieved by Gollan [2003] by combining GMDs with IDM. The performance of LLMs and LLMMs combined with IDM is investigated in this thesis and presented in the following sections.

6.2 Generative Models

In this section, GMDs are evaluated in combination with IDM. Similar results have been published by Gollan [2003], but in order to fully understand and finally extend

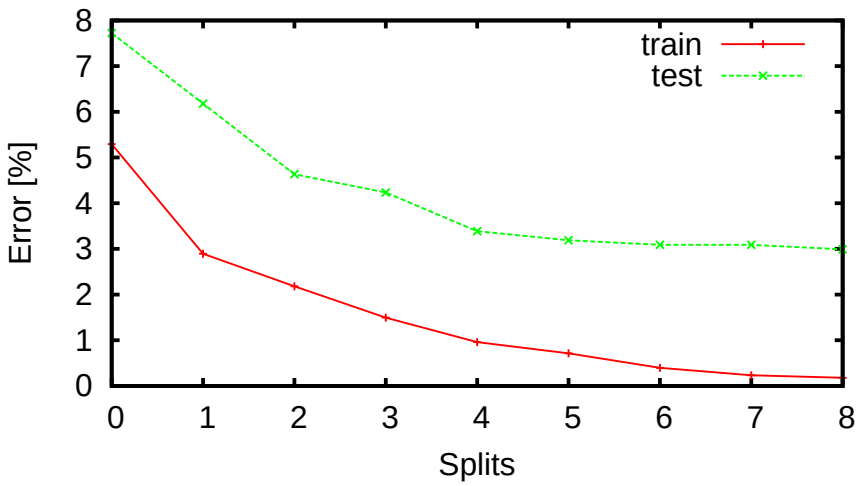

Figure 6.3. Progression of error rate with increasing number of splits on USPS using GMDs with IDM.

the model the results have been reproduced. Furthermore, Gollan [2003] reports error rates for GMDs with up to four splits which corresponds to 16 densities per class. In this thesis experiments with up to 8 splits have been performed resulting in roughly $2^8 = 256$ densities per class. As presented in Chapter 2.2.2, the number of densities does not always exactly match $2^{\#\text{splits}}$ because densities without associated observations are deleted. Figure 6.2 shows the progression of the error rate on the USPS database with increasing number of splits. It can be seen that with more than six splits the error rate on test data does not significantly decrease anymore and train error approaches 0%. The error rate for the six-split model is not much worse than the model-free KNN approach, but is more than ten times faster in testing due to a reduced number of prototypes. In the KNN approach, the similarity of a test image to all training images has to be computed, opposed to only $C \cdot 2^{\#\text{splits}}$ similarity computations given a GMD with C classes. Using more than 6 splits, additional prototypes cannot be estimated robustly because the lack of training data associated with each density and thus the reduced number of deformed images make it difficult to estimate means which represent the data. This leads to artifacts in the prototypes and the error rate does not decrease further.

6.2.1 Eigendeformation splitting

Since the estimation of means for the GMD using IDM involves the computation of the deformations of each image to the cluster means, these deformations can be exploited further by using eigendeformations as proposed by Uchida and Sakoe

Table 6.2. Results using Gaussian Mixture Densities with different splitting approaches.

	Splitting	
Splits	variance	eigendeformations
0	7.7%	-
1	6.1%	6.2%
2	4.6%	4.6%
4	3.4%	3.6%
6	2.9%	3.2%

Figure 6.4. Deforming mean of class 9 with values of k ranging from -3 to 3.

[2003]. This approach is extended as described in Section 2.2.2 and results in a new method to splitting densities w.r.t. the most common deformations in the cluster. It can be seen from Table 6.2 that using eigendeformation splitting approach leads to only slightly worse results than the initial approach where cluster means are disturbed by variance, although not much work has been invested in tuning the parameters since this idea is clearly out of focus of this work. It is likely that it is possible to obtain better results. Possible approaches could include the use of several eigendeformations instead of just the one with the largest eigenvalue or the tuning of parameter k which controls the direction and force of the deformations as demonstrated in Figure 6.4. Here, the PCA component with the largest eigenvalue has been used for the deformation with weight k ranging from -3 to 3. It can be observed that the used deformation results in translation of the upper part of the prototype to the left and right.

6.2.2 Sobel Features

As discussed by Keysers et al. [2007], gradients of images are used to further enhance recognition performance with IDM. So far, the single use of these features was to enhance pixel-distance computation and thus lead to a better mapping of an test or training image to a reference image. Since the pixel distances are needed to compute derivatives and thus to train the model discriminatively, using Sobel features inside the distance computation only and using grey-valued prototypes it would make it necessary to re-convert the Sobel derivatives to grey-valued ones. To avoid this

Table 6.3. Results on the USPS database comparing the standard distance-based approach using Sobel, and training separate means for each Sobel layer.

#dens./class	Sobel distance	Sobel features
1	7.5%	6.5%
2	5.1%	6.0%
4	4.6%	4.9%

step, it is possible to transform all data into the Sobel space beforehand and thus model prototypes for each gradient direction separately as presented in Section 2.3. For single densities, this approach results in a better error rate of 6.5% compared to 7.5% when Sobel features are just used inside the IDM distance computation. An overview of error rates using Sobel features and different numbers of splits is given in Table 6.3. The result using 2 densities per class and Sobel features seems to be an outlier and using even more densities marginalises the differences between the models.

6.3 Discriminative Models

In the following, the most important results of this work are reported and discussed. The main goal was to fuse discriminative models with the IDM, hence different existing models have been extended. First, results with SVMs are presented, where the most common kernel function, the RBF kernel is replaced by a kernel capable of dealing with IDM transformations. Since SVMs tend to be complex w.r.t. the number of parameters, LLMs are investigated in addition and finally extended to LLMMs.

6.3.1 Support Vector Machines

As discussed in Chapter 3, SVMs are robust models which are well known for their performance and interpretability. The RBF kernel is preferred for most applications because of its ability to model class boundaries flexibly. Haasdonk and Keysers [2002] report an error rate of 3.0 % using an SVM with a tangent distance based kernel, which is invariant to global deformations like rotation and scaling. In Table 6.4, SVMs with the IDM kernel are compared to SVMs with the regular RBF kernel. It can be seen that the performance is improved considerably, and for the OAO approach the complexity of the model represented by the number of SVs is reduced considerably by 25%. The OATR approach performs equally well, but the different optimisation results in a higher amount of SVs. The performance is hardly different than the model-free KNN approach, where the number of parameters and

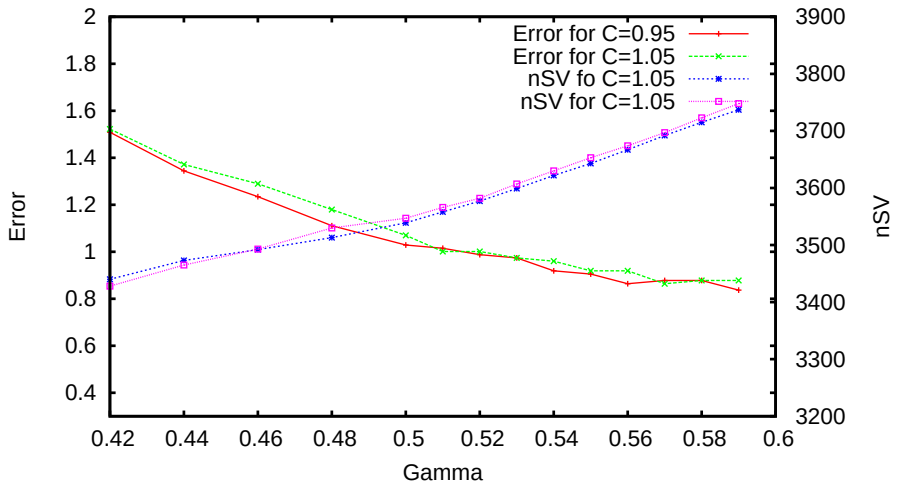

Figure 6.5. Error rate and number of support vectors using five-fold cross validation on USPS-train depending on different gamma and cost parameters.

Table 6.4. Results using support vector machines with RBF- and IDM-Kernel.

	Kernel=RBF		Kernel=IDM	
SVM	ER	# SV	ER	# SV
OAO	4.6%	2561	2.8%	2065
OATR	4.4%	2713	2.8%	3536

thus the evaluation time needed is more than twice as large. When a symmetric distance like the one required for the SVMs kernel is used in the model-free KNN-approach, the error rate is considerably worse (3.4%). Figure 6.5 shows progression of SVM error rate using different values for the parameters γ and C of the SVM. The parameters have been tuned using five-fold CV on the USPS training data. It is clearly observable that the number of SV correlates with the error rate.

6.3.2 Discriminative Training of Generative Models

Since the Gaussian mixture model (GMM) described in Chapter 2.2 already gives good results, it is straightforward to refine its model parameters discriminatively. Table 6.5 shows the clear improvement of the model with and without IDM. The model using both discriminative training and IDM gives the best error rate on

Table 6.5. Results on the USPS database using discriminatively trained Gaussian mixture densities with IDM.

IDM used	GD	disc. train
no	18.5%	8.5%
yes	6.5%	**5.3%**

#dens./class	GMD	disc. train
1	6.5%	5.3%
2	6.0%	4.9%
4	4.9%	-

USPS known so far using single densities without second order features. Keysers et al. [2007] present an error rate of 4.9% using single densities, but use the two-dimensional warping algorithm which includes constraints imposed by neighbouring pixels thus being far more complex than the model used here. Nonetheless, using more densities per class is a major slowdown factor due to the fact that derivatives are computed for all possible alignments of a given training observation to each cluster mean. This leads to extremely slow model training, effectively preventing training of mixture densities, which is shown in Table 6.5. The models using more than one density per class are not fully trained, since one training iteration takes more than one day to compute on a regular 3-GHz machine.

6.3.3 Log-Linear Models

The main goal of this thesis is the joint modelling of discriminative models with IDM. In the following, results are presented using the first generative/discriminative approach, followed by the results achieved using the full LLM with IDM. While the first already gives extremely good results without using mixtures, the full model gives the best known error rate of 4.1% on the USPS database using a single density approach. However, the two-step approach makes it easier to extend the model to mixture densities and is capable of making use of local context, while having the downside that a generative initialisation is necessary to achieve the given results.

Two-step generative/discriminative Approach

The approach presented in Section 5.1 has the advantage that pixel based features like the presented Sobel- and local context features can easily be integrated. The downside is that in the log-linear interpretation only the variances are trained discriminatively and the alignment is fixed. Nonetheless, training is fast due to the small number of parameters updated per class and it is straightforward to use mixture densities by simply computing pixel distances to all cluster means given a GMD model trained with IDM. Figure 6.6 depicts the moderate overfitting curve using single densities only, while Table 6.6 presents results using more than one den-

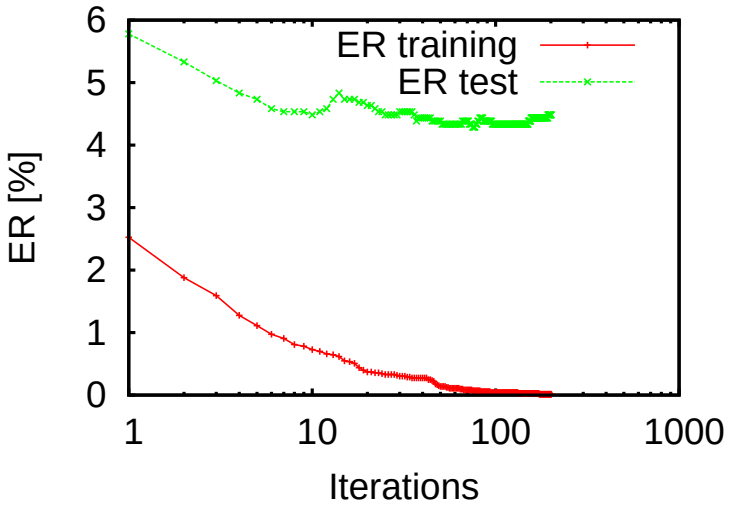

Figure 6.6. Train and test error rates of variance training using one prototype per class.

Table 6.6. Results on the USPS database using the heuristic log-linear approach.

# densities	two-step model	# parameters
10	4.4%	1·(256·10+10)
40	3.5%	4·(256·10+10)
157	**3.0%**	16·(256·10+10)
297	3.5%	32·(256·10+10)

sity per class. However, using more than 16 densities per class leads to decreasing results due to overfitting.

Full Model

It is widely known that LLMs on their own are good classifiers. Without using second order features, an error rate of 8.5% on the USPS task is achieved. Using squared Sobel features this already good error rate can be reduced further to 7.0%. For regular LLMs, the squaring of the Sobel features is necessary, because the gradients are linear combinations of the original features and thus have no effect on the performance of the classifier. However, if deformations are used it is no longer necessary to compute the squares of the gradient since the Sobel features are combined non-linearly. Given the discussed equivalence between LLMs and GMDs even with IDM, it is straightforward to initialise the LLMs with the trained

Table 6.7. Results using GD-initialised LLMs with IDM using different features.

	GD-init		LLM-trained	
Features	Train	Test	Train	Test
Gray values	30.4%	34.4%	6.1%	10.6%
Sobel	3.7%	7.1%	0.7%	**5.4%**
Gray context	6.3%	10.2%	0.6%	6.1%
Sobel context	3.7%	6.5%	2.5%	5.9%

generative models. Table 6.7 gives an overview of error rates using different features in the generative as well as the log-linear model. It can be observed that Sobel features have the strongest impact on error rate both for the generative and the discriminative model since they effectively allow for edge detection. Using grey values alone, the alignment is rather difficult which does not allow for an effective training of the log-linear model. Starting from the bad initialisation of 34.4% error on USPS, the model achieves a fairly good result of 10.6%, which is still worse than a regular LLM.

Incorporating local information like Sobel features or the local context, the generative model improves considerably. The initial alignment in the LLM is better and parameters can thus be updated more reliably. The best results could be achieved using only the Sobel gradients, so feature expansion is fairly moderate using only a two-element representation of each pixel. Using local context in the log-linear framework is not straightforward and is discussed in more detail in Section 6.3.3.

Regularisation

Regularisation is a common method to reduce the effect of over-training [Chen and Rosenfeld, 2000]. Figure 6.7 shows error rate and criterion for both train and test data given different regularisation parameters. The best regularisation parameter was chosen by increasing the parameter slowly and picking the value where the error rate on the training data has its last minimum. With this approach, an error rate of 4.6% on USPS using a log-linear model initialised with a GD has been achieved.

Initialisation

So far, the LLMs have been initialised using generatively trained Gaussian density classifiers with IDM. This has the advantage that the initial alignment is very good

Table 6.8. Error rates on USPS using different regularisation weights in GD-initialised LLM with IDM.

Reg.	Train ER	Test ER
0.05	0.7%	5.3%
0.1	0.4%	4.7%
0.15	0.6%	**4.6%**
0.2	1.0%	4.7%
0.3	1.3%	4.8%

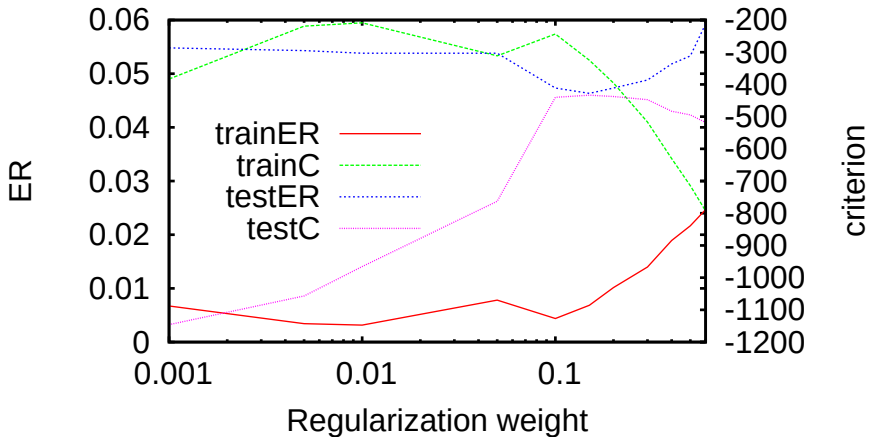

Figure 6.7. Train/test error rate and according criterion of LLM with IDM, initialised with GD and using different regularisation weights.

and has a clear interpretation, but makes a two-step training necessary. To avoid that, several methods for initialisation have been evaluated:

1. using GDs trained with IDM
2. using LLMs trained without IDM as initialisation
3. using zeros as initialisation values

Method 1 is the approach used in the experiments presented before, while method 2 is a first idea to give a suitable, interpretable initialisation while still staying inside the log-linear framework. The latter has the big advantage that all parameters are initialised discriminatively, which makes re-alignment feasible as discussed in the next section. Table 6.9 shows results using the different initialisation schemes. In all models, Sobel has been used disregarding the local context and regularisation

Table 6.9. Error rates on USPS using LLMs with IDM and different initialisations.

Initialisation	Train ER	Test ER
IDM-GD	0.6%	4.6%
zeroinit	5.1%	9.8%
LLM-init	0.2%	5.9%

has been tuned in the same manner as described in Section 6.3.3. The error rates given in Table 6.9 show that using discriminatively trained parameters is already a feasible approach for initialising the LLM, as it improves the result of the regular log-linear model by 2.5% absolute. However, it should be noted that only one initial alignment was computed, which is probably not the optimal alignment given further trained parameters. The effect of this re-alignment is discussed in the next section.

Alternating optimisation

As mentioned before, models initialised with zeros or other models not incorporating deformations tend to lack good initial deformation alignments. As discussed in Chapter 5.2.3, it is possible to alternate between updating the parameters and reestimating the alignments. This is known to converge if both subproblems are convex.

Comparing Table 6.9 and Table 6.10, it can be seen that recomputing the alignments for the GD-initialised model does not change the error rate, which is due to the fact that alignments hardly change if the initial model is trained generatively and already incorporates IDM. This can be observed in Figure 6.8, where the criterion is visually equal in each re-alignment step. In this figure, 20 parameter updates have been conducted in between recomputing the alignments for visualisation purpose. The training of LLMs incorporating IDM with discriminative- or zero-initialisation is depicted in Figures 6.9 and 6.10. Here, alignments have been reestimated every 10 iterations, which results in a much smoother curve. For comparison, we show the visualisation of the training of an LLM initialised with zeros without recomputing the alignments. The curve is similar, but the error rate is bigger. However, significant improvements are achieved w.r.t. the other models, LLM-initialised LLMs with IDM and achieve the best-known error rate of 4.1% on the USPS task using single density models.

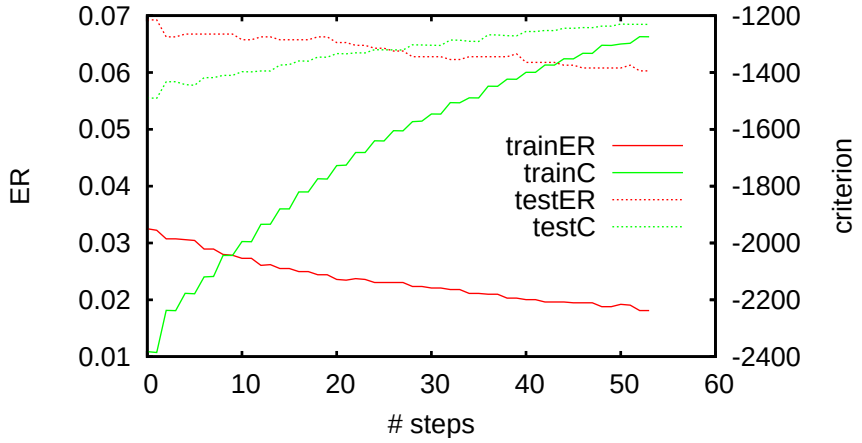

Figure 6.8. Alternating re-Alignment and parameter updates using a generatively initialised LLM with IDM.

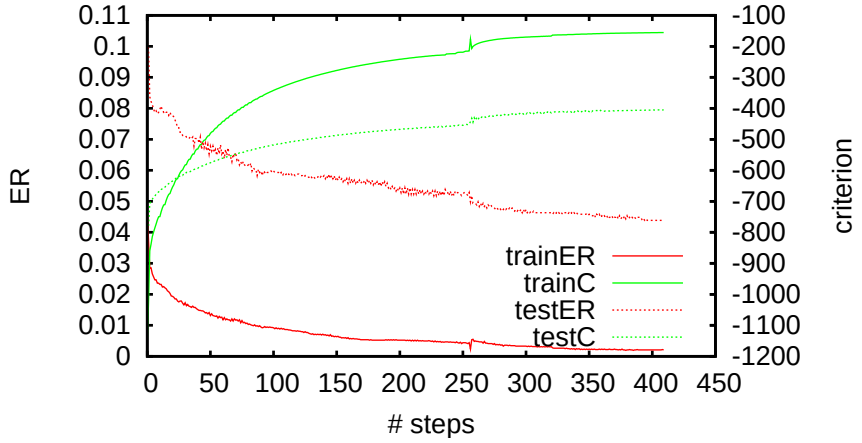

Figure 6.9. Alternating re-alignment and parameter updates using a discriminatively initialised LLM with IDM.

Parameter tying

As discussed in Chapter 5.2.6, the parameters of the log-linear model are connected if IDM is used. So far the λs have been treated as if they were totally independent. Three main approaches to tying the λs have been investigated:

- λs are considered to be completely independent
- λs are fully connected

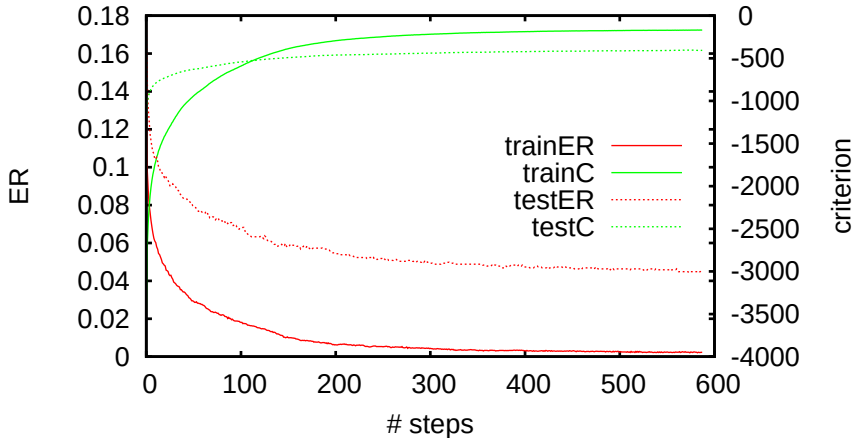

Figure 6.10. Alternating re-alignment and parameter updates using a LLM with IDM, initialised with zeros.

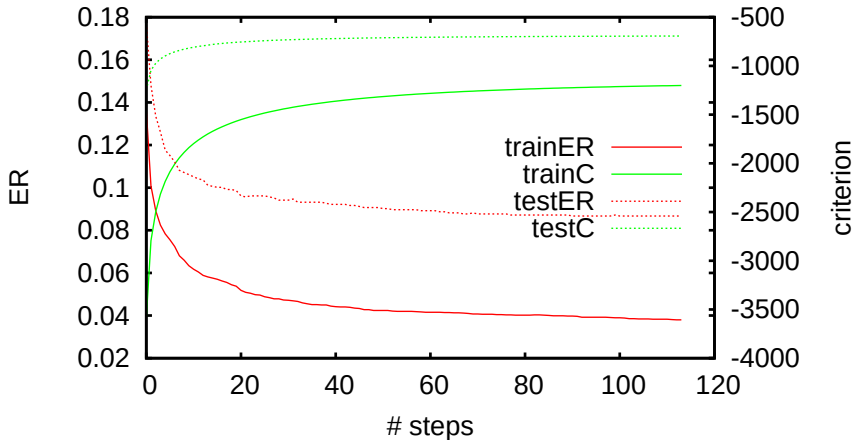

Figure 6.11. In this figure, a LLM with IDM is initialised with zeros, and trained without recomputing the alignments.

- λs are considered to be connected, but just the centre of each parameter is updated

Table 6.11 shows results on USPS using the described approaches and it can be seen that fully tying the λs decreases the results considerably. This is probably due to the fact that misplaced pixels gain influence because not only the positions they are mapped to are affected, but neighbouring pixels as well. If the mentioned constraints concerning centred updates are imposed, the error rate is equal to the

Table 6.10. Alternating re-alignment and parameter updates with different initialisation schemes.

Initialisation	Train ER	Test ER
IDM-GD	0.6%	4.6%
zeroinit	0.2%	4.4%
LLM-init	0.2%	**4.1%**

Table 6.11. Tying of λs in LLM on USPS using regularisation.

	Full tying	Centre update	No tying
Error on test	6.2	4.9	4.9

ad-hoc model, although the number of parameters of the model is lowered considerably. Still, using local context features did not result in superior error rates compared to the model not using local context at all. This is probably originates from the fact that Sobel features already model the local context implicitly, and thus given the increased number of parameters with no additional information the model is prone to overfitting.

Tying the α_{cijxy} as presented in Chapter 5.2.6 results in a clear decrease in performance, since they do not only model the deformation penalty, but some part of the original distance to the pixel as well. This leads to an extremely bad initial alignment, and even realigning during the training does not help because of the limited information given the reduced α_{cijxy}. The best error rate achieved using this schemes is 6.5% on the USPS database.

Full Covariance

Chapter 5.2.5 already mentioned the complexity of second-order features being exponential if IDM is used due to the necessary knowledge of all alignments if a new alignment has to be computed. However, we have tried to reduce this problem by precomputing the alignments without using second-order features and then training the full covariance matrix. The problem with the resulting model is the very high number of parameters which cannot be estimated reliably. Thus, the error rate on training data quickly lowered to zero % while error on test remained high. Nonetheless, using this described approach and diagonal covariance, the same results as with the best method presented so far could be achieved.

Using local covariance matrices the error rate on test becomes considerably worse since the number of parameters is very large and thus the model overfits to the

Table 6.12. Results on the USPS database using log-linear models with IDM and different parameters and initialisations.

Method	ER
GD + IDM + Sobel + lc	6.5%
+LLM	5.7%
+ reg	4.9%
+fully tied	6.2%
+tied centre	4.9%
GD + IDM + Sobel	7.1%
+ LLM	5.5%
+ reg	4.6%
LLM	8.5%
+ Sobel^2	7.0%
+ Sobel + IDM + reg	5.9%
+ reAlign	**4.1%**
LLM + IDM + Sobel + reg + reAlign	**4.4%**

training data.

6.3.4 Summary

Table 6.12 summarises results of LLMs with IDM on the USPS database. It can be seen that omitting local context features leads to superior results compared to using them in the log-linear context. Furthermore, the enclosed LLMs surpasses the two-step approach for which it is necessary to train a GD with IDM beforehand. The best results could be achieved by training a LLM without IDM and then further training it using deformations. However, initialising the LLM incorporating IDM with zeros did not lead to much worse results.

6.4 Results on the MNIST database

In order to verify the validity of the obtained results so far, we have evaluated the most significant models on the MNISTs database as well. Table 6.13 shows the error rates of GMDs with IDM, which are further trained discriminatively. It can be seen that single density estimation already leads to better results compared to performance on the USPS task due to the larger amount of training data. Discrim-

Table 6.13. Results on the MNIST database using discriminatively trained Gaussian (mixture) densities with IDM.

IDM used	GD	disc. train
no	18.0%	8.2%
yes	5.8%	**4.7%**

#densities/class	GMD	disc. train
1	5.8%	4.7%
2	4.7%	4.1%
4	2.9%	-

Table 6.14. Results on the MNIST database using log-linear models with IDM and different parameters and initialisations.

Method	ER
GD + IDM + Sobel	5.8%
+ LLM	4.1%
+ reg	**2.8%**
LLM	9.4%
+IDM + Sobel + reg + reAlign	3.3%
LLM + IDM + Sobel + reg + reAlign	4.2%

inative retraining with IDM improves the single-density error rate from 5.8% to 4.7%. Comparable to the USPS task, discriminative training of Gaussian mixtures with IDM showed not to be feasible due to the computational complexity of the derivative calculation.

Table 6.14 shows results obtained on the MNIST database using LLMs with IDM. Here, it can be seen that GD-initialised LLMs perform slightly better than the enclosed LLMs with IDM. However, this mainly stems from the fact that the recalculation of the alignments is computationally more complex than the calculation of the derivatives and thus it is possible to conduct more training iterations on the GD initialised model, since here, re-alignment is not necessary during the training process. Experiments using LLMMs with IDM have not been conducted due to time constraints.

Chapter 7

Conclusions

In this work, we have investigated the IDM, GDs and LLMs and how to combine them. Starting from the idea that LLMs and Gaussian models are closely related and from deformation-invariance we have developed a deformation invariant LLM. Furthermore, we have investigated how the IDM can be incorporated into SVMs. The main contributions of this work are as follows:

- A probabilistic formulation of the IDM was used to train a deformation-aware Gaussian classifier for OCR (Section 4.2).
- We propose a novel approach to split generatively trained clusters based on eigendeformations by exploiting knowledge of class specific deformations (Section 2.2.2).
- An IDM-kernel for SVMs has been proposed and an error rate of 2.8% on the USPS task has been achieved using this model (Chapter 3).
- We have shown how to use Sobel features directly instead of using them in the distance calculation only. This reduces the error rate from 7.5% to 6.5% using GDs trained with IDM (Section 4).
- Using the probabilistic formulation of the IDM, we re-train the GDs with IDM discriminatively, which improves recognition error rate on the USPS task from 6.5% to 5.3% (Chapter 4).
- The structure of the parameter space using local context in GDs and LLMs has been investigated thoroughly and efficient means have been found to exploit overlaps which lead to increase of model accuracy (Section 5.2.6).
- A first two-step approach to log-linear modelling with IDM, where a generative model using IDM is trained first and then a LLM is trained on pixel distances to the cluster means using class-pooled features. This approach leads to an error rate of 4.4% using single densities and can easily be extended to mixtures, which leads to an error rate of 3.0% using roughly 16 densities per class (Section 5.1).

- We integrate the IDM into an LLM in order to jointly model deformations and a discriminative distribution (Chapter 5).
- We investigate several ways to train and initialise the LLM with IDM:
  - We investigate a full LLM with IDM which is derived from the shared properties of Gaussian and log-linear models (Section 5.2).
  - Three possible initialisation schemes for the LLM with IDM have been investigated (Section 5.2.7):
    - We have initialised the LLM with GDs trained with IDM and further trained the LLM. This approach leads to an error rate of 4.6% on the USPS task.
    - Furthermore, we have initialised the LLM with parameters obtained from a discriminatively trained model without deformations. The training of this model w.r.t. deformations leads to an error rate of 5.9%.
    - In order to obtain a closed-form LLM which incorporates IDM, we have investigated the initialisation of the models with zeros and thus without training another model beforehand, which leads to an error rate of 9.8% on the USPS task.
  - We have investigated the training of all parameters of the LLM using IDM by alternating the optimisation of the model parameters and recomputing the pixel alignments. This approach decreased the error rates of the zero- and discriminative initialisation approaches from 9.8% to 4.4% (zero) and from 5.9% to 4.1% (discriminative), the latter being the best known result on the USPS database using a single density model (Section 5.2.4).
  - Furthermore, second order features for LLM with IDM have been investigated, but no further improvements could be achieved since the model is either too hard, or too strong in the sense that the number of parameters is very high which leads to over-training on the data (Section 5.2.5).

Summarising, it can be said that the fusion of discriminative models with deformations has been shown to be a valuable model for image recognition. Very good performance on the USPS task has been achieved using a computationally cheap model.

In order to obtain state-of-the-art performance, the model complexity is too low which encourages further research on the topic of mixture densities or second order features. Additionally, we feel that using local context for LLMs with IDM can

be investigated more deeply. Furthermore, the computation of the deformations for generative models has led to the idea to split densities w.r.t. to class specific deformations. This approach is promising and might lead to more efficiently trained mixture densities in the future.

List of Figures

List of Tables

Glossary

ANN	artificial neural network.
ASR	automatic speech recognition.
BFGS	Broyden-Fletcher-Goldfarb-Shanno.
CV	cross validation.
EM	expectation maximization.
ER	error rate.
GD	Gaussian density.
GIS	generalised iterative scaling.
GMD	Gaussian mixture density.
GMM	Gaussian mixture model.
HMM	hidden Markov model.
IDM	image distortion model.
IRMA	Image Retrieval in Medical Applications.
KNN	k nearest neighbour.
LBFGS	limited memory BFGS method.
LLM	log-linear model.
LLMM	log-linear mixture model.
ML	maximum likelihood.
MMI	maximum mutual information.
MNIST	modified NIST.

NIST	National Institute for Standards and Technology.
NN	nearest neighbour.
OAO	one against one.
OATR	one against the rest.
OCR	optical character recognition.
PCA	principal component analysis.
RBF	radial basis function.
SV	support vector.
SVM	support vector machine.
USPS	US Postal Service.

Appendix A

Software

For the evaluation of all presented models except the SVMs, a software framework developed at the Human Language and Pattern Recognition Group at the Chair of Computer Science 6 has been used and extended. This framework is under active development and implemented training algorithms for GMDs and LLMMs and basic support for the IDM before this work started. In this chapter, a short overview of the functionality added through the course of this work is given by command line examples and according parameter explanations. Not explicitly mentioned here are a variety of helper tools for converting data, which have been implemented using Python[1], BASH[2] scripts, and awk[3].

A.1 Generative Modelling with IDM

In order to train and evaluate a Gaussian density with IDM as described in Section 2.2, which leads to an error rate of 6.5% on the USPS set (Table 6.3), the following command line can be used:

```
gmd -t <trainfile> -T <testfile> --emMode=idm --logPxMode=idm
    --localcontext=1 --warprange=2 --UpdatesBetReest=4
    --sepSob oldtrain evaltrain evaltest save=<model>
```

A full list of all possible parameter settings can be obtained using `gmd --help`. Here, regular parameters are given by a "-" or "–" at the beginning of the parameter name, and additionally a command syntax allowing for simple changes in the program functionality is given. In the above example we specify the following parameters:

- `--emMode=idm` : Here, the mean re-estimation is set to use IDM.
- `--logPxMode=idm` : This parameter sets the emission probability to incorporate IDM.

[1] http://www.python.org
[2] http://www.gnu.org/software/bash
[3] http://www.gnu.org/software/gawk

- `--localcontext=1 --warprange=2` sets maximum warprange and local context to the respective values.
- `--UpdatesBetReest=4` : This parameter sets the maximum number of mean re-estimations during the EM-algorithm.
- `--sepSob` : With this parameter set, horizontal and vertical Sobel gradients are used instead of the original grey values.
- `oldtrain evaltrain evaltest` gives a series of actions which are conducted in that order. Here, `oldtrain` simply trains the GD and the commands `evaltrain evaltest` evaluate the model on train and test data, and finally `save=<model>` saves the model to disk.

The above model can be extended to GMDs by adding the `--MaxSplits=<S>` parameter, which splits the clusters S times, resulting in 2^S densities per class.

A.2 Discriminative Training of Gaussian Densities

In order to further train the model discriminatively as described in Chapter 4, it is possible to simply append further commands to the list of actions as follows:

```
gmd -t <trainfile> -T <testfile> --emMode=idm --logPxMode=idm
    --localcontext=1 --warprange=2 --UpdatesBetReest=4
    --sepSob oldtrain evaltrain evaltest save=<model>
    repeat:next=3:times=50 10trainiter evaltrain evaltest
```

Here, the simple command `repeat:next=3:times=50` is used to repeat the next three commands 50 times. The command `10trainiter` applies 10 iterations of the gradient descent algorithm, which in this example uses the IDM as well (Results given in Table 6.5).

A.3 Log-Linear Models

The same framework also implements training methods for LLM and has been extended to incorporate the IDM during the course of this work. The following command line loads a GD trained with IDM, converts the model to a LLM as described in Section 5.2.2, and trains it using LBFGS:

```
loglinmix -t <trainfile> -T <testfile> --sepSob
          --loadgmd=<model> lbfgs evaltrain evaltest
          -r -R 0.1
```

Given a suitable GD-model trained with IDM, this model achieves an error rate of 4.6% on the USPS task (Table 6.8). The parameters of the IDM are inherited from the GD-model.

A.3.1 Maximum Approximation

Given a LLM trained without IDM, we can load this model and further train it with deformations. To achieve good results, re-computing the pixel alignments is necessary (Section 5.2.3).

```
loglinmix -t <trainfile> -T <testfile> --sepSob --useIDM
          load=<llm> evaltrain evaltest repeat:next=3:times=1000
          lbfgs:MAXITER=10 evaltest reAlignIdm -r -R 0.01
```

In this example, we restrict the LBFGS algorithm to a maximum of 10 iterations to avoid overfitting, use `--reAlignIdm` to compute the pixel alignments and repeat the process 1000 times. With this command line and a suitable LLM, an error rate of 4.1% on the USPS task can be achieved (Table 6.10).

An alternative commandline initialises the model with zeros and applies the same training principles as the model trained above:

```
loglinmix -t <trainfile> -T <testfile> --sepSob --useIDM
          zeroinit=1 evaltrain evaltest repeat:next=3:times=1000
          lbfgs:MAXITER=10 evaltest reAlignIdm -r -R 0.01
```

Here, `zeroinit=<n>` initialises n densities per class with zeros.

A.3.2 Two-step approach

The heuristic approach presented in Section 5.1 has been implemented using a simple tool to compute IDM pixel distances to a set of trained prototypes (Results given in Table 6.6).

```
savePixelDistances -t <prototypes> -T <train/testfile> -i
                   --symm -S=<output>
```

Here, `-i` denotes that IDM is to be used for pixel distance calculation, and with `--symm` set, distances are computed symmetrically. With the resulting output files, a LLM can be trained using the MaxEnt toolkit [Keysers et al., 2002a] , which is also developed at the Chair of Computer Science VI, and which has been extended to allow for class-wise features during the course of this work:

```
ME-cw-1st.i686 -tr <trainfile> -te <testfile>
```

A.4 SVMs with IDM-kernel

We use the widely used LIBSVM [Chang and Lin, 2001] as basis for our implementation. The IDMs kernel has then been implemented, along with efficient means to load pre-computed kernel matrices. Two simple tools have been implemented in order to pre-compute this kernel matrices:

- `saveDistances.cc` computes (symmetric) IDM-distances between sets of images.
- `distanceJFprecomputedKernel.py` computes the suitable kernel representation in LIBSVM format given a distance file.

To allow for efficient parameter tuning, a gridsearch algorithm has been implemented for LIBSVM using n-fold cross validation.

Given a pre-computed IDM-kernel (Section 3.1.1), a OAO-SVM can be trained using the following command line:

```
svm-train -t 5 -g <gamma> -c <cost> -u <kernel> <model file>
```

An n-fold cross validation with grid search for suitable parameters γ and C can be started using the following command line:

```
svm-train -t 5 -z <n> <gstart> <gend> <gsteps>
          <cstart> <cend> <csteps> -u <kernel>
```

Here, `gstart, gend` denote the starting/ending value for γ and `qsteps` is the step size, with similar parameters for tuning the C parameter of the SVM. Finally, a model can be evaluated on testing data by using the following command line:

```
svm-predict -u <testkernel> <model file> <output file>
```

For the classification process, we also provide a 'kernel' instead of the original data to be classified. The model file is the one trained by the SVM before, and class labels for the test data are saved in the output file. Results obtained using this approach are given in Table 6.4.

Bibliography

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22 (1):39–72, March 1996.

James C. Bezdek and Richard J. Hathaway. Convergence of alternating optimization. *Neural, Parallel Sci. Comput.*, 11(4):351–368, 2003. ISSN 1061-5369.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

S. F. Chen and R. Rosenfeld. A survey of smoothing techniques for me models. *IEEE Transactions on Speech and Audio Proccessing*, 8(1):37–50, January 2000.

Jörg Dahmen, Ralf Schlüter, and Hermann Ney. Discriminative training of gaussian mixture densities for image object recognition. In W. Förstner, J.M. Buhmann, A. Faber, and P. Faber, editors, *Deutsche Arbeitsgemeinschaft für Mustererkennung Symposium*, Informatik aktuell, pages 205–212, Bonn, Germany, September 1999. Springer.

Jörg Dahmen, Daniel Keysers, Hermann Ney, and Mark Oliver Güld. Statistical image object recognition using mixture densities. *Journal of Mathematical Imaging and Vision*, 14(3):285–296, May 2001.

J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, 43(5):1470–1480, 1972.

Dennis DeCoste and Bernhard Schölkopf. Training invariant support vector machines. *Machine Learning*, 46(1-3):161–190, 2002.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algoritm. *Journal of the Royal Statistical Society Series B*, 39(1):1–38, January 1977.

Thomas Deselaers and Hermann Ney. Deformations, patches, and discriminative models for automatic annotation of medical radiographs. *Pattern Recognition Letters*, 2008. accepted for publication.

Thomas Deselaers, Daniel Keysers, and Hermann Ney. Discriminative training for object recognition using image patches. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 157–162, San Diego, CA, June 2005.

Thomas Deselaers, Andre Hegerath, Daniel Keysers, and Hermann Ney. Sparse patch-histograms for object classification in cluttered images. In *Deutsche Arbeitsgemeinschaft für Mustererkennung Symposium*, volume 4174 of *Lecture Notes in Computer Science*, pages 202–211, Berlin, Germany, September 2006.

Thomas Deselaers, Tobias Weyand, and Hermann Ney. Image retrieval and annotation using maximum entropy. In C. Peters, P. Clough, F. Gey, J. Karlgren, B. Magnini, D.W. Oard, M. de Rijke, and M. Stempfhuber, editors, *Evaluation of Multilingual and Multi-modal Information Retrieval – Seventh Workshop of the Cross-Language Evaluation Forum, CLEF 2006*, volume 4730 of *LNCS*, pages 725–734, Alicante, Spain, 2007.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. John Wiley & Sons, New York, NY, 2nd edition, 2001.

Christian Gollan. Nichtlineare Verformungsmodelle für die Bilderkennung. Master's thesis, Human Language Technology and Pattern Recognition Group, RWTH Aachen University, Aachen, Germany, September 2003.

Bernard Haasdonk. Feature space interpretation of svms with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):482–492, 2005.

Bernard Haasdonk and Claus Bahlmann. Learning with distance substitution kernels. In *DAGM-Symposium*, Tübingen, Germany, August 2004.

Bernard Haasdonk and Daniel Keysers. Tangent distance kernels for support vector machines. In *International Conference on Pattern Recognition*, pages 864–868, Quebec City, Canada, September 2002.

Bernard Haasdonk, Alaa Halawani, and Hans Burkhardt. Adjustable invariant features by partial haar integration. In *International Conference on Pattern Recognition*, volume 17, Washington, DC, USA, August 2004.

Andre Hegerath, Thomas Deselaers, and Hermann Ney. Patch-based object recognition using discriminatively trained gaussian mixtures. In *British Maschine Vision Conference*, volume 2, pages 519–528, Edinburgh, UK, September 2006.

Georg Heigold, Thomas Deselaers, Ralf Schlüter, and Hermann Ney. Gis-like estimation of log-linear models with hidden variables. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 4045–4048, Las Vegas, NV, USA, April 2008.

Juraj Hromkovic and Waldyr M. Oliva. *Algorithmics for Hard Problems*. Springer-Verlag, New York, Inc., Secaucus, NJ, USA, 2002.

Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, March 2002.

D. Keysers, F.-J. Och, and H. Ney. Maximum entropy and Gaussian models for image object recognition. In *Deutsche Arbeitsgemeinschaft für Mustererkennung Symposium*, pages 498–506, Zürich, Switzerland, September 2002a.

Daniel Keysers. *Modeling of Image Variability for Recognition*. PhD thesis, Human Language Technology and Pattern Recognition Group, RWTH Aachen University, Aachen, Germany, March 2006.

Daniel Keysers and Walter Unger. Elastic image matching is NP-complete. *Pattern Recognition Letters*, 24:445–453, January 2003.

Daniel Keysers, J"org Dahmen, Thomas Theiner, and Hermann Ney. Experiments with an extended tangent distance. In *International Conference on Pattern Recognition*, pages 38–42, Barcelona, Spain, December 2000.

Daniel Keysers, Roberto Paredes, Hermann Ney, and Enrique Vidal. Combination of tangent vectors and local representations for handwritten digit recognition. In *Int. Workshop on Statistical Pattern Recognition*, Lecture Notes in Computer Science, pages 538–547, Windsor, Ontario, Canada, August 2002b. Springer Verlag.

Daniel Keysers, Christian Gollan, and Hermann Ney. Classification of medical images using non-linear distortion models. In *Bildverarbeitung für die Medizin*, pages 366–370, Berlin, Germany, March 2004a.

Daniel Keysers, Christian Gollan, and Hermann Ney. Local context in non-linear deformation models for handwritten character recognition. In *International Conference on Pattern Recognition*, volume 2, pages 511–514, Cambridge, UK, August 2004b.

Daniel Keysers, Thomas Deselaers, Christian Gollan, and Hermann Ney. Deformation models for image recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1422–1435, August 2007.

S. Knerr, L. Personnaz, and G. Dreyfuss. Single-layer learning revisited: a stepwise procedure for building and training a neural network. In J Fogelmann, editor, *Neurocomputing: Algorithm, Architectures and Applications*. Springer, 1990.

Y. Linde, A. Buzo, and R.M. Gray. An algorithm for vector quantization design. In *IEEE Transactions on Communications*, volume 28, pages 84–95, January 1980.

D.C. Liu and J. Nocedal. On the limited memory method for large scale optimization. *Mathmatical Programming B*, 45(3):503–528, 1989.

J. Lööf, M. Bisani, Ch. Gollan, G. Heigold, Björn Hoffmeister, Ch. Plahl, R. Schlüter, and H. Ney. The 2006 RWTH parliamentary speeches transcription system. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 105 – 108, Pittsburgh, PA, September 2006.

Ronny Luss and Alexandre D'Aspremont. Support vector machine classification with indefinite kernels. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 953–960. MIT Press, Cambridge, MA, 2008.

James Mercer and Andrew Russel Forsyth. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 209:415–446, 1909.

F.J. Och and H. Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Annual Meeting of the Assoc. for Computational Linguistics*, pages 295–302, Philadelphia, PA, USA, July 2002. Best Paper Award.

John Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. Burges, and A. Smola, editors, *Support Vector Learning*. MIT Press, 1998.

Ralf Schlüter. *Investigations on Discriminative Training Criteria*. PhD thesis, RWTH Aachen University, Aachen, Germany, September 2000.

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

Bernhard Schölkopf and Alexander J Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, USA, 2002.

Patrice Simard. Best practices for convolutional neural networks applied to visual document analysis. In *7th Int. Conf. Document Analysis and Recognition*, pages 958–962, Edinburgh, Scotland, August 2003.

S. Uchida and H. Sakoe. Eigen-deformations for elastic matching based handwritten character recognition. *Pattern Recognition*, 36(9):2031–2040, September 2003.

Seiichi Uchida and Hiroaki Sakoe. A survey of elastic matching techniques for handwritten character recognition. *IEICE Transactions on Information and Systems*, E88-D(8):1781–1790, 2005.