Pattern Recognition 43 (2010) 2476–2484

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Object classification by fusing SVMs and Gaussian mixtures

Thomas Deselaers [a,b,*], Georg Heigold [b], Hermann Ney [b]

[a] *Computer Vision Laboratory, ETH Zurich, Zurich, Switzerland*
[b] *Human Language Technology and Pattern Recognition Group, RWTH Aachen University, Aachen, Germany*

ARTICLE INFO

Article history:
Received 12 October 2009
Received in revised form
9 January 2010
Accepted 1 February 2010

Keywords:
Support vector machine
Gaussian mixtures
Discriminative classifiers
Generative classifiers
Local-feature-based object recognition

ABSTRACT

We present a new technique that employs support vector machines (SVMs) and Gaussian mixture densities (GMDs) to create a generative/discriminative object classification technique using local image features. In the past, several approaches to fuse the advantages of generative and discriminative approaches were presented, often leading to improved robustness and recognition accuracy. Support vector machines are a well known discriminative classification framework but, similar to other discriminative approaches, suffer from a lack of robustness with respect to noise and overfitting. Gaussian mixtures, on the contrary, are a widely used generative technique. We present a method to directly fuse both approaches, effectively allowing to fully exploit the advantages of both. The fusion of SVMs and GMDs is done by representing SVMs in the framework of GMDs without changing the training and without changing the decision boundary. The new classifier is evaluated on the PASCAL VOC 2006 data. Additionally, we perform experiments on the USPS dataset and on four tasks from the UCI machine learning repository to obtain additional insights into the properties of the proposed approach. It is shown that for the relatively rare cases where SVMs have problems, the combined method outperforms both individual ones.

1. Introduction

Two major approaches to the classification of patterns are well known: generative approaches and discriminative approaches. Both have been successfully applied to object classification and both have their own advantages and disadvantages. For object classification in images nearly all approaches nowadays strongly build on the use of local features. Generative approaches such as those presented by Fergus et al. [11], Mikolajczyk et al. [23] try to find an optimal representation of the original data by keeping as much information as possible. Generative methods can be trained from partly or even unlabelled data and normally allow for a reconstruction of the most likely prototype for each modelled class. Generative methods can be built very robustly. Discriminative methods, such as those presented by Bosch et al. [2], Moosmann et al. [25], Shotton et al. [30], Viola and Jones [32], require fully labelled training data, can be applied very quickly and often show better recognition accuracy than their generative counterparts. The biggest problem of many discriminative approaches is that they are prone to overfitting, which requires significant extra effort to be overcome [2,36].

Clearly, both approaches have their advantages and several authors have tried to combine the approaches to benefit from both. One common approach to join the two worlds is a two stage method: using a generative model to create a fixed length representation of the image, which then is classified using a discriminative technique [6,9,14,19]. Cazzanti et al. [3] combine the two approaches the other way round: the extract discriminative features using their similarity discriminant analysis and then apply a generative model for classification.

A direct approach to joining the two principles is proposed by Minka [24] and used in an object recognition framework by Lasserre et al. [17] which allows to seamlessly blend from a fully discriminative model to a fully generative model. Grabner et al. [12] modify a discriminative, boosted model to account for reconstruction in addition to the discriminatory performance and a clear performance boost for noisy data was observed. Lin et al. [20] take the opposite approach and boost Gaussians as weak classifiers. Hegerath et al. [13] present a Gaussian mixture density classifier for patch-based object recognition which, in principle, is a generative model but which is refined by discriminatively changing the cluster-weights. The discriminative refinement of a generative model can in some cases be shown to be identical to directly training a discriminative model [17,24] if done properly. The model presented in [17] which also resembles

[*] Corresponding author at: Computer Vision Laboratory, ETH Zurich, Zurich, Switzerland. Tel.: +41 789 43 6669.
E-mail addresses: deselaers@vision.ee.ethz.ch, thomas@deselaers.de (T. Deselaers).
URL: http://thomas.deselaers.de (T. Deselaers).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.02.002

a mixture, thus is a much cleaner way to achieve a similar goal. Do and Artieres [8] use SVMs to improve the recognition of sequential data. The proposed model increases the discriminative power of generative models by using a support vector machine to improve a mixture of generative models.

Among the discriminative models, support vector machines (SVMs) are very popular in many domains since they have very good performance in many cases and can be applied to many problems in machine learning. Some of the two-staged generative/discriminative approaches mentioned above use SVMs for the second stage [9] and in [33] a kernel for direct application to a local feature-based image representation is presented. SVMs, which do not model a probability distribution, are not open to the ideas presented in [17] and can thus not easily be extended to incorporate generative concepts.

For dimensionality reduction Yang et al. [35] propose a model that allows to combine the advantages of kernel principal component analysis (PCA) and linear discriminant analysis (LDA). The resulting method is a hybrid generative/discriminative dimensionality reduction method.

Despite the fact that it is relatively easy to find a good set of parameters for training an SVM, which makes SVMs one of the most successful and best understood approaches, LeCun et al. [18] observe that in some cases tuning the parameters of an SVM to obtain optimal performance turns an SVM into "*little more than a glorified template matcher*". This is in accordance to the observation addressed here that an SVM (with radial basis function (RBF) kernel, which is probably the most commonly used kernel) in some cases has a large portion of the training data as support vectors (SVs) and thus it degenerates to a *discriminatively weighted* kernel densities classifier. This degeneration can be interpreted as effectively overfitting to the training data.

In this paper, we present an approach that fuses an SVM with a generatively trained GMD classifier and thereby profits from the advantages of both techniques. The idea, how SVMs and GMDs can be fused was published in [7] but not evaluated as thoroughly and not applied to object recognition. A close connection between Gaussian mixtures and SVMs was already discussed by Schölkopf et al. [28], but to the best of our knowledge, the direct fusion of both approaches has not yet been investigated. For the two approaches, to be fused, we first convert the SVM into a GMD with identical decision boundary. This conversion allows to compute posterior probabilities $p(c|x)$ for class c of observation x and class conditional probabilities $p(x|c)$ for the obtained GMD. These probabilities, however, must not be considered to be the true probabilities for the underlying SVM but are just an interim instrument to allow for the combination. To obtain probabilities from an SVM, other methods have been proposed, e.g., by Platt [26], Seeger [29], Sollich [31] where a sigmoid function is fit onto the distance of an observation to the hyperplane to obtain probabilities. SVMs and GMDs could be combined by computing their individual posterior probabilities Kittler [16] and combining these, however, the here proposed method is not a late combination of two different classifiers, but a unified framework, to fuse the two classification methods into one joint classifier.

The object recognition approach presented is based on the assumption that objects consist of parts and these parts can be modelled more or less independently which is a common assumption in the object recognition literature [9,11,14]. Here the parts are represented by local features extracted at interest points and classified individually. To classify the image, the respective classification decisions are combined. In the course of the experiments, we observe that SVMs are not suitable for this approach since the problem of classifying the individual local features is too "inseparable" for an SVM to be solved and thus the SVM shows the degeneration as described by LeCun et al. [18]. The combination of the SVM with GMD is shown to be an effective smoothing of the SVM. Experimentally it is shown that this smoothing greatly reduces the negative effects from overfitting.

The remainder of this paper is structured as follows: In Section 2 we describe the local feature extraction for the SVM-based object recognition system presented in Section 3 and the GMD-based object recognition system presented in Section 4. In Section 5 the fusion of SVMs and GMDs into a unified framework is presented. In Section 6 we present and discuss experiments on the PASCAL VOC 2006 task and to get a deeper insight into the proposed methods we also evaluate it on the USPS dataset and on four tasks from the UCI machine learning repository. Finally, the paper is concluded.

2. Feature extraction for object classification

For each image, we extract up to 200 SIFT descriptors [21] at the top-200 Difference-of-Gaussian (DoG) interest points. SIFT descriptors [21] are local image descriptors designed to be invariant with respect to image translation, scaling, and rotation. They are partially invariant to illumination changes and they are robust to local geometric distortion.

We do not evaluate different types of features in this work, but strictly follow the procedure described by Lowe [21] since here we want to focus on the influence of the model rather than the impact of carefully chosen descriptors. SIFT features, however, were shown to perform well on a wide range of different tasks [22] and need only little additional tweaking. In the following, an image X is represented by the set of L local features $x_1^L = \{x_1 \ldots x_L\}$.

3. Object classification using local features with an RBF-kernel SVM

SVMs, being a modern, well understood and widely used classifier, directly predict the label of an observation. An SVM commonly discriminates between two classes: -1 and 1 using the decision rule

$$r : X \rightarrow \{-1,1\}, X \mapsto r(X) = \operatorname{sgn}\left(\sum_{v_i \in \mathcal{S}} \alpha_i K(X, v_i) + \alpha_0\right) \quad (1)$$

$$= \operatorname{sgn}\left(\sum_k \sum_{v_i \in \mathcal{S}_k} \alpha_i K(X, v_i) + \alpha_0\right) \quad (2)$$

to classify the observation X where K is a kernel function, the v_i are the support vectors (SVs) and the a_i are the corresponding weights, α_0 is a bias term.

Here, an image X is represented by a variable number of local SIFT features. To classify an image, we classify each of these local features x individually and determine the class of the whole image by combining the individual classification decisions. To allow for effectively combining, not only the resulting class but also the distance to the decision hyperplane is considered. The distance to the decision hyperplane can be considered to be proportional to a class-conditional emission probability, i.e., we assume that given a class, every local feature vector x which is far away from the hyperplane is likely to be emitted from this class, and conversely, for every vector which is close to the hyperplane, the probability that this vector comes from the class is low. Thus, we can write

$$p(x|k) \propto \sum_{v_i \in \mathcal{S}_k} k \alpha_i K(x, v_i) + \alpha_0 \quad (3)$$

where S_k is the set of SVs for class k, i.e., those SVs with positive α_i for class $k=+1$ and those with negative α_i for class $k=-1$, and the α_i are the corresponding weights, α_0 is the bias term.

Given these probabilities, we apply Bayes' decision rule, assume that the patches of an image are independent, and come to the following decision rule to classify an image X represented by a set of local features $\{x_1 \ldots x_L\}$.

$$X \mapsto r(\{x_1^L\}) = \operatorname*{argmax}_k \{p(k|\{x_1^L\})\} = \operatorname*{argmax}_k \{p(k)p(\{x_1^L\}|k)\}$$
$$= \operatorname*{argmax}_k \left\{ p(k) \prod_{l=1}^{L} p(x_l|k) \right\} \tag{4}$$

Note that the $p(x_l|k)$, which are the values obtained from the right-hand side of Eq. (3), are not properly normalised with respect to x. This, however, does not affect the decision and thus the missing normalisation does not matter here.

4. Object classification using local features with Gaussian mixture densities

Gaussian mixture models are a *generative* model: for each object class a class-dependent mixture $p(x|k)$ is used. To decide which object is depicted in an image, again Bayes' decision rule is used:

$$X \mapsto r(\{x_1^L\}) = \operatorname*{argmax}_k \{p(k|\{x_1^L\})\} = \operatorname*{argmax}_k \{p(k) \cdot p(\{x_1^L\}|k)\}$$
$$= \operatorname*{argmax}_k \left\{ p(k) \cdot \prod_{l=1}^{L} p(x_l|k) \right\} \tag{5}$$

where $\{x_1^L\}$ denotes the set of patches $x_1, \ldots, x_L$ extracted from image X. We model $p(x_l|k)$ as untied Gaussian mixture densities with class-wise pooled diagonal covariances, i.e.,

$$p(x_l|k) = \sum_{i=1}^{I_k} p(i|k) \cdot p(x_l|i,k) = \sum_{i=1}^{I_k} p(i|k) \cdot \mathcal{N}(x_l|\mu_{ki}, \Sigma_k) \tag{6}$$

where class k is represented by I_k clusters, $p(i|k)$ are the cluster weights and $\mathcal{N}(x_l|\mu_{ki}, \Sigma_k)$ is the Gaussian representing the i-th cluster of class k with mean μ_{ki} and covariance Σ_k. Without loss of generalisation, we assume that $\Sigma_k = \sigma\mathbf{1}$, where $\mathbf{1}$ is the identity matrix. This can always be achieved by decorrelating and rescaling the observations.

These mixtures are trained using the EM algorithm to maximise the likelihood $\prod_{n=1}^{N} \prod_{l=1}^{L_n} p(x_l|k)$ [5] by starting with an initial Gaussian over all observations which is iteratively split and reestimated until a certain number of densities is obtained. Densities with too few observations are deleted to ensure stable estimation.

5. Fusing support vector machines and Gaussian mixtures

As described above, SVMs are a discriminative classifier and GMDs are a generative classifier. In the following, we first describe how SVMs with RBF kernel can be represented in the form of GMDs without changing the decision boundary and then describe how two GMDs can be fused to profit from their individual advantages.

Fig. 1 shows an overview of recognition phase for the different models (SVM (Section 3), GMD (Section 4), fused SVM and GMD (Section 5)). The applied system is the same in all three cases, only the emission probabilities $p(x_l|k)$ depend on the underlying model.

5.1. Approximating SVMs using GMDs

Since SVMs are designed to discriminate only two classes, here we consider two cases: first we describe the transformation for the two-class case and then we extend this transformation to the multi-class case.

Two-class case: It is well known that SVMs as well as GMDs can in principle model arbitrary decision boundaries and thus can theoretically represent the respective other without any loss of accuracy or generalisation ability. This theoretical feature, however, does not pose an advantage as the most difficult thing for any classifier normally is to find the model parameters, and thus it is not clear how to benefit from the theoretical equivalence here.

For the case of SVMs with an exponential RBF kernel, a close similarity between SVMs and GMDs can be observed. Starting from the general form of the decision function, we show that GMDs and SVMs are in fact equivalent and, even more, that either one can be represented as the respective other without changing the decision boundary.

Consider the decision rule of a standard SVM in Eq. (1). This equation can be rewritten as

$$r(X) = \arg\max_{k \in \{-1,1\}} \left\{ \sum_{v_i \in \mathcal{S}_k} k\alpha_i K(X, v_i) + \alpha_0 \right\} \tag{7}$$

$$= \arg\max_{k \in \{-1,1\}} \left\{ \sum_{v_i \in \mathcal{S}_k} k\alpha_i \exp(-\gamma \|X - v_i\|^2) + \alpha_0 \right\} \tag{8}$$

where $\mathcal{S}_k$ is the set of SVs v_i from class k, α_i is the corresponding weight, γ is a kernel parameter, and α_0 is the learned bias.

The decision rule of a GMM in Eq. (5) can also be rewritten as

$$r(X) = \operatorname*{argmax}_k \left\{ \sum_i p(k)p(i|k) \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left(-\frac{1}{2} \frac{\|x - \mu_{ki}\|^2}{\sigma^2} \right) \right\}. \tag{9}$$

Now it can be seen that Eqs. (8) and (9) are identical except for the α_0 if the means μ_{ki} and the SVs v_i correspond. In fact, a GMD can be transformed into an SVM (and vice versa) by setting

$$k\alpha_i = p(k)p(i|k) \frac{1}{(2\pi\sigma^2)^{D/2}} \tag{10}$$

$$\gamma = \frac{1}{2\sigma^2} \tag{11}$$

$$\mu_{ki} = v_i \quad \text{for all } v_i \in \mathcal{S}_k \tag{12}$$

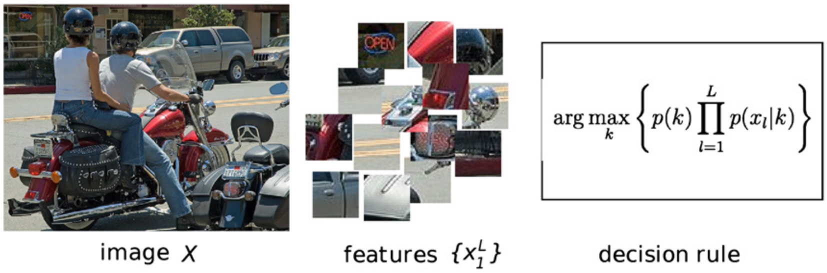

Fig. 1. Scheme of the object classification system. Left: input image, center: extracted local features, right: decision rule.

α_0 can be sufficiently well approximated by an additional density with arbitrary mean and very high variance and a cluster weight proportional to α_0. D is the dimensionality of the observations. The fact that α_i can be negative, which is not allowed for the probabilities in the GMDs, can easily be worked around by adding an SV to the other class with weight $-\alpha_i$, which does not affect the decision boundary and can smoothly be transformed to a density in the GMD model.

Thus GMDs and SVMs can represent the same decision boundaries for the two class case and either representation can be obtained from the other as described above keeping the decision boundaries constant. Thus, the main difference between a GMD and an SVM with RBF kernel is the training method and the optimisation criterion.

Multi-class case: The earliest used implementation for SVM multi-class classification is probably the "*one-against-the-rest*" (also known as "*one-against-all*") method, which has been used to extend other binary classifiers to multi-class problems before [15]. Therefore, not a single classifier is trained to discriminate between all classes at once but a classifier is trained for each class to discriminate it from all other classes and the decision is drawn according to the scores from these individual decisions.

The decision rule in this case is

$$r(x) = \operatorname*{argmax}_{k} \left\{ \sum_{v_i \in \mathcal{S}_k} k\alpha_i K(x, v_i) + \alpha_k \right\} \tag{13}$$

where the parameters for each class k are optimised in individual training procedures considering the two-class problem where all competing classes are considered to be from class -1 and class k is considered to be class 1.

Here, the relationship to the GMD classifier is similar to the two-class case, if this SVM is converted into a GMD classifier, each SV becomes a mixture mean, we assume a pooled, diagonal covariance matrix with identical entries for each dimension inversely proportional to γ and the cluster weights are given through the weights α_i of the SVs:

$$p(k)p(i|k)\frac{1}{(2\pi\sigma^2)^{D/2}} = \alpha_i \tag{14}$$

$$\mu_{ki} = v_i \in \mathcal{S}_k \tag{15}$$

$$\frac{1}{2\sigma^2} = \gamma \tag{16}$$

Again, it is necessary to address the class-wise constant bias terms α_k which can be substituted by very diffuse Gaussians (one per class) with an arbitrary mean and a weight proportional to α_k. Negative weights α_i are compensated by adding respective densities to all other classes.

Note that the same transformation can be applied if the SVM is trained to jointly discriminate all classes as described by Weston and Watkins [34] because the same decision rule is applied there and only the training is done differently.

5.2. *Fusing SVMs and GMDs*

Given two GMDs

$$\mathcal{G}_1 = ((\mu_{11} \ldots \mu_{1I}), (\sigma_{11}^2 \ldots \sigma_{1I}^2), (p_1(1) \ldots p_1(I))) \tag{17}$$

$$\mathcal{G}_2 = ((\mu_{21} \ldots \mu_{2J}), (\sigma_{21}^2 \ldots \sigma_{2J}^2), (p_2(1) \ldots p_2(J))) \tag{18}$$

one trained using the EM algorithm for GMDs and the other obtained by transforming an SVM, it is possible to fuse both GMDs into a single GMD and arbitrarily mix between the two. The new, joint GMD $\mathcal{G}'$ is obtained as

$$\begin{aligned}\mathcal{G}' = &((\mu_{11} \ldots \mu_{1I}, \mu_{21} \ldots \mu_{2J})(\sigma_{11}^2 \ldots \sigma_{1I}^2, \sigma_{21}^2 \ldots \sigma_{2J}^2)\\ &(wp_1(1) \ldots wp_1(I), (1-w)p_2(1) \ldots (1-w)p_2(J)))\end{aligned} \tag{19}$$

where w is a weighting factor allowing to smoothly blend between $\mathcal{G}_1$ (for w=1) and $\mathcal{G}_2$ (for w=0).

Since the cluster weights of $\mathcal{G}_1$ and $\mathcal{G}_2$ are normalised, for $0 \le w \le 1$ the cluster weights of the resulting GMD $\mathcal{G}'$ are also normalised.

The resulting decision boundary, now is chosen according to a combination of the optimisation criteria of the SVM, which optimises classification performance, and the GMD, which optimises data representation. Thus, the resulting decision boundary is not-optimal with respect to either of these criteria, but according to some compromise of these.

In Fig. 2 an example GMD (1 density per class) and three differently parametrised SVMs are visualised for two-dimensional data. It can be seen that the SVMs have, depending on the scale of the kernel γ, many SVs, which is an indicator for possible overfitting. As will be experimentally observed later, overfitting of SVMs to the training data is a problem in cases where the data are very difficult to separate, which commonly goes along with a very high number of SVs. For GMDs, the number of parameters estimated can easily be fixed by the user (i.e. fix number of densities), thus by forcing the number of parameters to be small, overfitting can easily be avoided.

Note that it is typical that a GMD has far fewer densities than an SVM has SVs since in a GMD each density represents a set of observations whereas an SV in an SVM is one training sample.

In Fig. 3, the GMD from Fig. 2(a) is fused with the three different SVMs from Fig. 2(b)–(d) with different weights w_{svm}. The smoothing of the probability distribution and thereby of the decision boundary can clearly be observed. The effect is best observed in the top row of Fig. 3, which shows a combination of the SVM with $\gamma = 0.01$ (cp. Fig. 2(b)) with the GMD (Fig. 2(b)). The before extremely bumpy decision boundary of the SVM is strongly smoothed and only when the SVM gets relatively high weight a slight tendency to overfitting can be observed. Similarly, the decision boundaries for the combinations with the other two SVMs are smoothed when combined with the GMD.

6. Experiments

In the following we present experimental results for two different tasks. First we show the experimental results for the PASCAL 2006 data for the two individual object recognition methods and for their combination and then we present experiments on the well-known USPS database to further analyse the results and show how smoothing an SVM with a GMD can help to rescue clearly overfit classification methods from failing on test data. For both datasets the fused classifier outperforms its individual components. Example images for the PASCAL are given in Fig. 4.

6.1. *PASCAL VOC 2006*

In 2006, the PASCAL network of excellence organised a second visual object classes challenge (VOC) to allow for quantitative comparison of different approaches to object recognition, detection, and identification. The 2006 tasks comprise 10 classes and a total of 5304 images were made available [10]. We apply the above-described methods for the classification task. The data were split into training, development, and testing data. In the

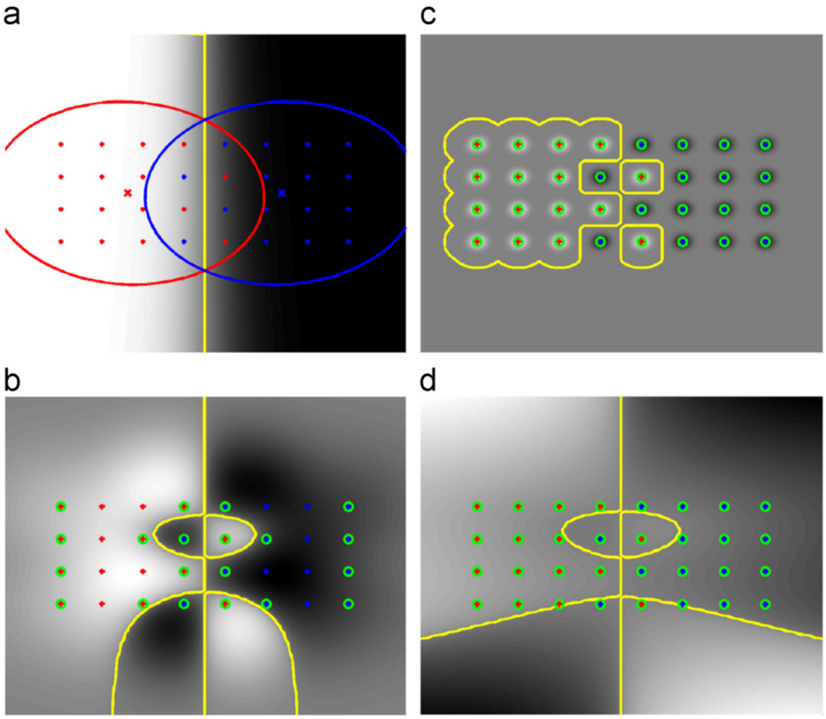

Fig. 2. (a) A single density Gaussian classifier, the variance is given by the ellipse and the mean is denoted by a small star (b)–(d) support vector machines with (b) $\gamma = 100$, (c) $\gamma = 10$, (d) $\gamma = 2$. White areas denote high probabilities for the red class and dark areas denote high probabilities for the blue class, the decision boundary is yellow and SVs are denoted by green circles. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

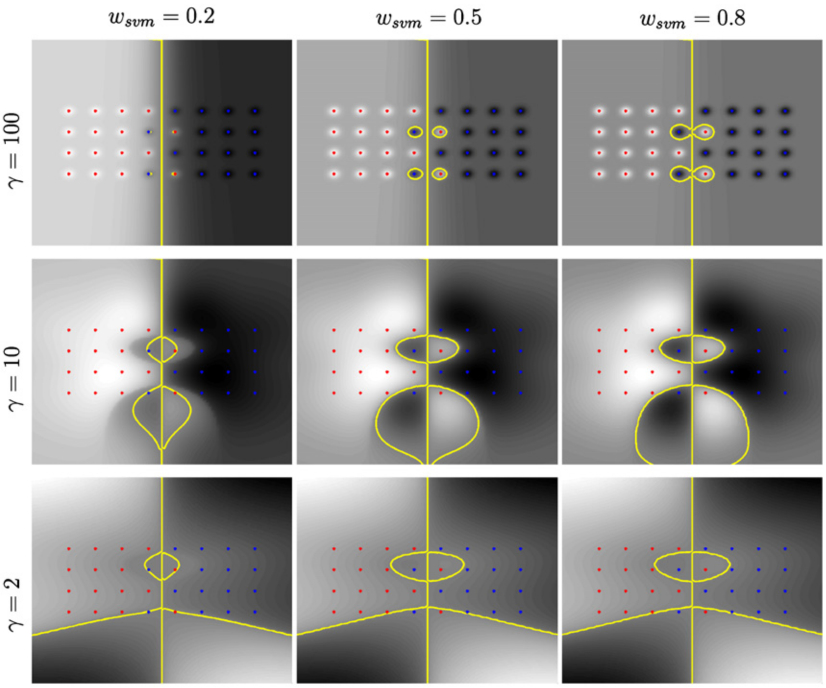

Fig. 3. Fusing the Gaussian classifier from Fig. 2(a) with the SVMs from Fig. 2(b)–(d) using different weights. The decision boundary is plotted as a yellow line. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

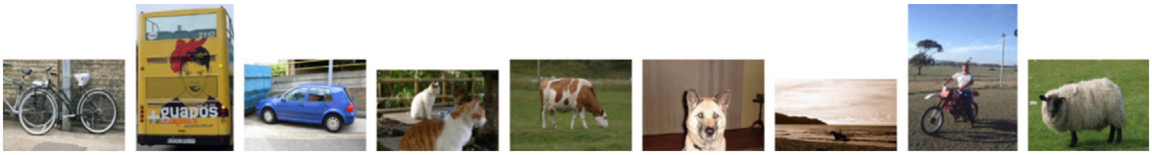

Fig. 4. Example images of the PASCAL VOC 2006 database.

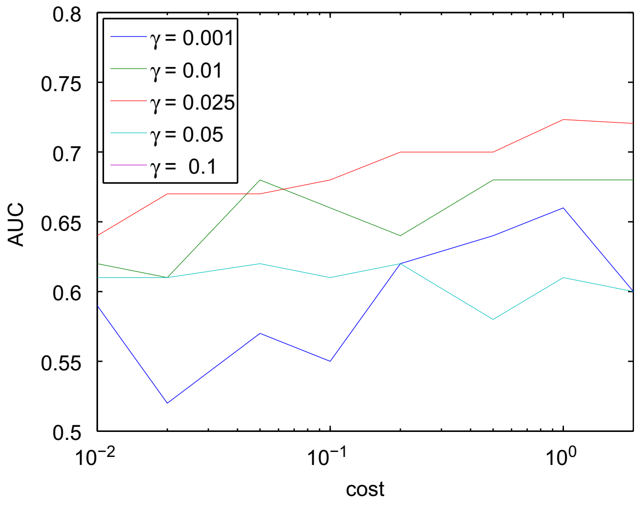

Fig. 5. The results from optimising the SVM parameters for the PASCAL 2006 task on the development data. The individual lines denote different values for the SVM-scale parameter γ.

following, we mainly present experiments on the training vs. development task.

In PASCAL VOC 2006 [10], area under the curve (AUC) is the standard evaluation measure. It measures the area under the receiver operating characteristic (ROC) curve. A ROC curve is a plot of the sensitivity *vs* (1 specificity) for a binary classifier as the decision threshold is varied. This allows for comparing classifiers independently of class distribution and misclassification costs.

SVMs: The object classification method presented in Section 3 is evaluated on the PASCAL VOC 2006 data. To find suitable settings for the SVM, we performed a grid search for the cost parameters C and the scale parameter γ of an SVM with RBF kernel in the PASCAL development data ($C \in \{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0\}$, $\gamma \in \{0.001, 0.01, 0.025, 0.05, 0.1, \}$, i.e., we performed $7 \cdot 5 = 35$ experiments). The average AUC over the classes of the grid search are shown in Fig. 5. Interestingly, we were unable to find parameters for this approach that are able to compete with the results for the (different) approaches applied in the PASCAL VOC 2006 [10]. It can be seen that $\gamma = 0.025$ performs best on the average and that a high C, i.e., large costs for misclassifications of the training data lead to the best results. The best results, when training on the training data and testing on the development data, are given in Table 1. A deeper analysis of the trained models revealed that the models performed poorly for the test data, but have quite good results on the training data and that they consist of a very large number of SVs. The best models trained were those with rather small numbers of SVs, i.e., models with "only" 20–50% of all training vectors as SVs which still is a large amount. If γ and C are chosen such that even fewer SVs are selected (e.g. smaller C), the performance is not better than the results presented here.

GMDs: The object classification method presented in Section 4 is also evaluated for the PASCAL data. Results for 8 and 10 splits (i.e. maximally 256 and 1024 densities per class, respectively) are presented in Table 1. Interestingly, and contrary to our initial expectations, this fully generative method clearly outperforms the discriminative SVM method on the development *and* on the training data. The performance of the 10 split model is on the development data only slightly better than the 8 split model but due to the higher number of parameters shows a stronger overfitting to the training data. Models with even more densities have also been evaluated and found not to perform better.

Table 1
Results on the PASCAL VOC 06 (development data) task using SVMs (top) and GMDs (bottom).

Class	SVM					
	C=0.2			C=1.0		
		AUC				
	# SV	Train	Test	# SV	Train	Test
Bicycle	56 788	0.87	0.77	62 451	0.98	0.75
Bus	42 586	0.85	0.63	48 210	0.99	0.72
Car	112 848	0.90	0.87	111 580	0.89	0.87
Cat	81 643	0.67	0.59	88 084	0.96	0.73
Cow	45 270	0.70	0.65	51 418	0.90	0.72
Dog	81 150	0.67	0.63	154 309	0.85	0.66
Horse	57 294	0.69	0.62	63 596	0.92	0.63
Motorbike	53 417	0.84	0.69	59 142	0.95	0.69
Person	135 007	0.71	0.65	139 335	0.93	0.70
Sheep	51 790	0.81	0.68	57 466	0.98	0.77
Class	GMD					
	8 splits			10 splits		
		AUC			AUC	
	# dens.	Train	Test	# dens.	Train	Test
Bicycle	512	0.97	0.85	2043	1.00	0.85
Bus	512	0.98	0.86	2024	1.00	0.85
Car	512	0.96	0.88	2046	1.00	0.90
Cat	511	0.93	0.78	2024	0.99	0.80
Cow	512	0.97	0.88	2027	1.00	0.88
Dog	512	0.89	0.73	2026	0.99	0.74
Horse	512	0.99	0.72	2043	1.00	0.73
Motorbike	512	0.99	0.81	2037	1.00	0.81
Person	512	0.90	0.69	2047	1.00	0.70
Sheep	510	0.97	0.86	2021	1.00	0.86

For the SVM experiments, the cost parameter C and the parameter γ were carefully chosen in a grid-search experiment of 35 experiments per class (cp. Fig. 5), but none of these experiments showed better results on the development data. For the GMD experiments, we used 8 and 10 splits. All results are given for the training data and for the development data. Additionally we give the number of support vectors and the number of densities.

Table 2
Results on the PASCAL data for combining the SVM with C=1.0 and $\gamma = 0.025$ with the GMD with 8 splits with equal weighting for both models.

Class	Train	Test	max{SVM, GMD}
Bicycle	1.00	0.82	**0.85**
Bus	1.00	0.82	**0.86**
Car	1.00	**0.89**	0.88
Cat	1.00	**0.80**	0.78
Cow	1.00	**0.88**	**0.88**
Dog	1.00	**0.75**	0.73
Horse	1.00	0.71	**0.72**
Motorbike	1.00	0.77	**0.81**
Person	1.00	**0.70**	**0.70**
Sheep	1.00	**0.87**	0.86

Fused model: Interestingly, although experiments were performed carefully, all data were scaled to be in a reasonable domain and C and γ parameters were carefully chosen, it seems impossible to find a *really* good set of parameters for the SVMs, and in all cases, the number of SVs chosen is very high, which is an indicator for overfitting problems.

Table 2 gives results for combining the SVM model with the GMD model (we choose the SVMs with C=1.0 and the GMDs with 8 splits). We have evaluated different weightings, but a weight of 0.5 performed best. It can be seen that the performance on the training data is improved even further, but the overfitting has no

Fig. 6. Example images from the USPS dataset.

negative effect here, since the results on the test data are improved over the SVM in all cases and over the GMD in most cases (printed in bold face). From this we conclude that the fusion of SVM and GMD effectively combined the advantages from both.

6.2. USPS

The well-known USPS Handwritten Digit Database consists of isolated and normalised images of handwritten digits taken from US mail envelopes scaled to 16 × 16 pixels. The database contains a separate training and test set, with 7291 and 2007 images, respectively.[1] The US Postal Service task is still one of the most widely used reference datasets for handwritten character recognition and allows fast experiments due to its small size. The test set contains a large amount of image variability and is considered to be a "hard" recognition task. Example images from the USPS database are shown in Fig. 6. For the USPS database, several good results using SVMs were published [27]. Here, our objective is not to outperform these results. Instead, we use this task for demonstrating the power of smoothing an overfit SVM using a GMD. For the USPS data, since the images are only 16 × 16 pixels, we do not use the local-feature based approach presented in Sections 3 and 4 but directly use the complete image as feature vector.

SVMs: Table 3 shows results for different parameters C and γ for the training and the test data of the USPS database along with the number of SVs in the trained model. The chosen multi-class voting scheme is one-against-the-rest. In accordance to the experiments described above, the best result on the test data is obtained in the models with the lowest numbers of SVs (bold faced). It is interesting to observe how important carefully choosing the cost parameter C and the scale parameter γ are in creating an SVM and how easily a badly overfitting SVM is created, if parameters are chosen inappropriately. In many cases, such as here, it is rather easy to find a good set of parameters, but in other cases, such as the one described above, it might be very difficult or even impossible. The results in Table 3 are a subset of the results obtained in cross-validation experiments to tune the C and γ parameters. For our analysis, the more interesting cases are those where the SVM overfits. Therefore, we use an SVM which overfits moderately (bold, red) in the following combination experiments.

GMDs: Table 3 gives results for 0–12 splits of GMD on the USPS data, it can be observed that the number of densities does not grow if more than 8 splits are used because due to the sparseness of the data, some densities do not have enough observations to be reestimated robustly and are therefore deleted. Here, the GMDs do not outperform the best SVM but still have competitive results.

Table 3
(a) Results using different scale parameters γ and cost parameters C in the SVM training on the USPS database.

(a) γ	C=0.5			C=1.0		
		ER (%)			ER (%)	
	# SV	Train	Test	# SV	Train	Test
0.001	4344	5.6	9.5	3670	4.5	8.7
0.01	3170	0.9	5.4	2947	0.2	**5.0**
0.02	4118	0.2	5.5	4053	0.0	**5.0**
0.05	5918	0.0	12.2	5824	0.0	11.8
0.08	6411	0.0	38.2	**6359**	**0.0**	**36.0**
0.1	6494	0.0	48.1	6454	0.0	47.6
0.2	6698	0.0	65.6	6656	0.0	65.6
0.5	7057	0.0	71.0	7012	0.0	71.1

(b) # splits	# densities	ER (%)	
		Training	Test
0	10	14.9	18.6
1	18	9.4	13.9
2	36	6.8	9.5
3	72	5.0	8.9
4	144	3.5	7.9
5	287	1.9	6.8
6	550	0.9	6.1
7	860	0.6	6.0
8	935	0.6	5.9
9	956	0.6	6.1
10	945	0.6	5.7
11	991	0.5	5.4
12	958	0.6	5.9

In addition to the classification error rate (ER [%]), we give the total number of support vectors in the model. (b) Results on the USPS database using Gaussian mixture densities with different numbers of densities. The number of densities does not increase anymore after 8 splits due to lack of training data, densities cannot be reestimated reliably and are thus deleted. Further splits and reestimations nonetheless can change the results.

Table 4
Combining the SVM with C=1.0 and $\gamma = 0.08$ with different GMDs from Table 3 using different weights w_{gmd} on the USPS dataset.

Split	w_{gmd}=0.2		w_{gmd}=0.5		w_{gmd}=0.8	
	Train	Test	Train	Test	Train	Test
0	0.0	11.7	0.0	10.7	0.2	14.8
1	0.0	9.5	0.0	8.1	0.1	10.3
2	0.0	6.3	0.0	6.1	0.0	8.3
5	0.0	5.7	0.0	5.7	0.0	5.8
10	0.0	5.2	0.0	5.3	0.0	5.3
11	0.0	5.1	0.0	5.2	0.0	5.3
12	0.0	5.3	0.0	5.8	0.0	5.6

Note that a column for w_{gmd}=0.0 is the rightmost column in Table 3(a) and a column for w_{gmd}=1.0 is the given in Table 3(b).

Fused model: To investigate the smoothing of the SVM using a GMD, we chose the SVM trained with $\gamma = 0.08$ and C=1.0 which clearly overfits but does not fail completely (bold in Table 3). This SVM is combined with several of the GMDs trained from the previous section using different weights. The results from these experiments are given in Table 4. It can be observed that none of resulting models performs as badly on the test data as the original SVM and that thus effectively the overfitting problem of the SVM is smoothed away by mixing with the GMD model. Combining a better SVM with a GMD does not lead to improved results over either of the models. In additional experiments, we fused the

[1] Data available from ftp://ftp.kyb.tuebingen.mpg.de/pub/bs

Table 5
Overview of the UCI datasets used, C number of classes; N total number of vectors; D dimensionality of the vectors.

Dataset	C	N	D
Diabetes	2	768	8
German	2	1000	24
Heart[a]	2	270	25
Vehicle	4	846	18

[a] Categorical features were expanded (original dim. 13).

Table 6
Results using SVMs and GMDs on the UCI datasets.

Dataset	SVM		GMD ER (%)		
	ER (%)	SVs (%)	1 dens.	2 dens.	32 dens.
Diabetes	29.9	50.0	28.6	30.5	24.7
German	24.5	54.4	24.0	26.5	30.0
Heart	25.9	56.0	22.2	22.2	27.8
Vehicle	60.2	50.7	53.8	49.1	35.1

We give the result for the SVM using the parameters determined on the data in fivefold cross-validation. For the SVM we also give the number of SVs in percentage. For GMD classifiers, we give three results for each database, using 1, 2, and 32 densities per class, respectively.

better SVM models with different GMD models, but could not outperform the best SVM result (4.6% ER) on these data.

6.3. *UCI datasets*

Additionally, we evaluate the proposed method on four datasets from the UCI machine learning repository[2] [1]. An overview over the datasets used is given in Table 5. These datasets were selected from the UCI repository by selecting those where classification is difficult, i.e., those where reported error rates are rather high. For all experiments we normalised the mean and the variance of all features to 0 and 1, respectively, as recommended for the use with SVMs.

First, we present the experimental results using only SVMs and using only GMDs. We used the default grid search of libsvm [4] in fivefold cross validation (11 values for C, 10 values for γ) to determine the parameters γ and C for the SVM. The results for the SVMs and the GMDs (with 1, 2, and 32 densities/class) are reported in Table 6. GMDs with 1 and 2 densities have only very few parameters and are thus extremely unlikely to overfit. Thirty-two densities were chosen to be a relatively large number of densities that can be reliably estimated on all of these datasets (in the heart-dataset on the average only 4.2 observations are in each density). For the other datasets we also evaluated models with more densities but results were not improved anywhere.

It can be observed that the error rates are in general quite high which shows that the selected tasks can be considered difficult. As expected, the SVMs decided to choose a significant part of the training data as SVs and thus the SVM is on the best way to overfitting. The GMDs mostly have better results (on the test data) than the SVMs, although the SVMs have far better error rates on the training data (not reported here), which is an indicator for overfitting effects.

The results of fusing the classifiers using the SVM and GMDs with 1 and 32 densities are given in Table 7. For these experiments, we do not tune the weight w and set w=0.5 such that GMDs and SVMs have equal influence. For the German-task and the heart task, the fused classifiers outperform their individual components, for the diabetes-task and for the vehicle-task, only the SVM is outperformed and the performance is similar to the GMD alone. Not surprisingly, for the vehicle- and diabetes-tasks the combination has better results if more densities are used, because here the GMDs were better with more densities. We assume that thus effectively the overfitting of the SVM is smoothed away by mixing with the GMD model. Informal experiments showed that for each of these tasks, improvements are possible by using different numbers of densities in the GMD and by using different weights w in the fusion, these results are omitted due to brevity constraints.

Table 7
Results of fusing SVMs and GMDs with w=0.5.

Dataset	ER (%)	
	1 dens.	32 dens.
Diabetes	30.5	27.3
German	22.5	33.0
Heart	22.2	18.5
Vehicle	55.0	35.7

[2] http://archive.ics.uci.edu/ml/index.html

7. Conclusion

We presented a novel generative/discriminative classification method consisting of fusing a generative Gaussian mixture density with a support vector machine with radial basis kernel. We have shown on the PASCAL 2006 task that the combined method is able to overcome overfitting problems of the support vector machine. Further analysis of the observed effects is performed on the USPS database and on four datasets of the UCI machine learning task.

As a conclusion, the proposed technique can be applied in those cases where SVMs suffer from major overfitting problems. However, this is not the case in many situations. SVMs are known to be a well-understood and easily usable classification technique. The PASCAL setup, as presented here, however, is different from most tasks in that respect, as the classification of individual patches is a very hard problem, where a classification boundary is hardly learnable, and thus the SVM tends to overfit, i.e., chooses a huge amount of training samples as SVs. For the USPS dataset, the combination did not lead to improvements, but we could show with a setup where the SVM was chosen to overfit deliberately that robustness is improved. For the UCI tasks the combined models outperform the SVMs and the GMDs in all cases.

References

[1] A. Asuncion, D.J. Newman, UCI machine learning repository, 2007.

[2] A. Bosch, A. Zisserman, X. Muñoz, Image classification using random forests and ferns, in: International Conference on Computer Vision, Rio de Janeiro, Brazil, October 2007.

[3] L. Cazzanti, M.R. Gupta, A.J. Koppal, Generative models for similarity-based classification, Pattern Recognition 41 (7) (2008) 2289–2297.

[4] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, 2001. Software available at ⟨http://www.csie.ntu.edu.tw/ ~ cjlin/libsvm⟩.

[5] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society Series B 39 (1) (1977) 1–38.

[6] T. Deselaers, D. Keysers, H. Ney, Discriminative training for object recognition using image patches, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, San Diego, CA, June 2005, pp. 157–162.

[7] T. Deselaers, G. Heigold, H. Ney, SVMs, Gaussian mixtures, and their generative/discriminative fusion, in: International Conference on Pattern Recognition, Tampa, Florida, USA, December 2008.

[8] T.M.T. Do, T. Artieres, Learning mixture models with support vector machines for sequence classification and segmentation, Pattern Recognition 42 (12) (2009) 3224–3230.

[9] G. Dorkó, C. Schmid, Object class recognition using discriminative local features. Rapport de recherche RR-5497, INRIA—Rhone-Alpes, February 2005.

[10] M. Everingham, A. Zisserman, C.K.I. Williams, L. Van Gool, The PASCAL Visual Object Classes Challenge 2006 (VOC2006) results, Technical Report, PASCAL Network of Excellence, 2006.

[11] R. Fergus, P. Perona, A. Zissermann, Object class recognition by unsupervised scale-invariant learning, in: IEEE Conference on Computer Vision and Pattern Recognition, Blacksburg, VG, June 2003, pp. 264–271.

[12] H. Grabner, P.M. Roth, H. Bischof, Eigenboosting: combining discriminative and generative information, in: IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, IEEE, New York, June 2007.

[13] A. Hegerath, T. Deselaers, H. Ney, Patch-based object recognition using discriminatively trained Gaussian mixtures, in: British Machine Vision Conference, vol. 2, Edinburgh, UK, September 2006, pp. 519–528.

[14] A. Holub, P. Perona, A discriminative framework for modelling object classes, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, IEEE, San Diego, CA, USA, June 2005, pp. 663–670.

[15] C.-W. Hsu, C.-J. Lin, A comparison of methods for multi-class support vector machines, IEEE Transactions on Neural Networks 13 (2) (2002) 415–425.

[16] J. Kittler, On combining classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (3) (1998) 226–239.

[17] J.A. Lasserre, C.M. Bishop, T.P. Minka, Principled hybrids of generative and discriminative models, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, New York City, NY, USA, June 2006, pp. 87–94.

[18] Y. LeCun, S. Chopra, M.A. Ranzato, F.-J. Huang, Energy-based models in document recognition and computer vision, in: ICDAR, Curitiba, Brazil, October 2007.

[19] Y. Li, L.G. Shapiro, J.A. Bilmes, A generative/discriminative learning algorithm for image classification, in: International Conference on Computer Vision, Beijing, China, October 2005, pp. 1605–1612.

[20] B. Lin, X. Wang, R. Zhong, Z. Zhuang, Continuous optimization based-on boosting Gaussian mixture model, in: International Conference on Pattern Recognition, vol. 1, Hong Kong, August 2006, pp. 1192–1195.

[21] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[22] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (10) (2005) 1615–1630.

[23] K. Mikolajczyk, B. Leibe, B. Schiele, Multiple object class detection with a generative model, in: IEEE Conference on Computer Vision and Pattern Recognition, New York City, NY, USA, June 2006.

[24] T. Minka, Discriminative models, not discriminative training, Technical Report TR-2005-144, Microsoft Research Cambridge, Cambridge, UK, October 2005.

[25] F. Moosmann, B. Triggs, F. Jurie, Fast discriminative visual codebooks using randomized clustering forests, in: Neural Information Processing Systems Conference, Vancouver, BC, Canada, December 2006.

[26] J.C. Platt, Probabilities for support vector machines, in: A. Smola, P. Bartlett, B. Schlkopf, D. Schuurmans (Eds.), Advances in Large Margin Classifiers, Cambridge, MA, USA, 1999, pp. 61–74.

[27] B. Schölkopf, Support Vector Learning, Oldenbourg, Munich, Germany, 1997.

[28] B. Schölkopf, K.-K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, V. Vapnik, Comparing support vector machines with Gaussian kernels to radial basis function classifiers, IEEE Transactions on Signal Processing 45 (11) (1997) 2758–2765.

[29] M. Seeger, Probabilistic interpretations of support vector machines and other spline smoothing models, in: EuroCOLT, Nordkirchen, Germany, March 1999.

[30] J. Shotton, J. Winn, C. Rother, A. Criminisi, TextonBoost: joint appearance, shape and context modeling for multi-class object recognition and segmentation, in: European Conference on Computer Vision, Lecture Notes in Computer Science, vol. 3951, Graz, Austria, Springer, Berlin, May 2006, pp. 1–15.

[31] P. Sollich, Probabilistic interpretation and Bayesian methods for support vector machines, in: ICANN, London, UK, 1999, pp. 91–96.

[32] P. Viola, M.J. Jones, Robust real-time face detection, International Journal of Computer Vision 57 (2) (2004) 137–154.

[33] C. Wallraven, B. Caputo, A. Graf, Recognition with local features: the kernel recipe, in: International Conference on Computer Vision, 2003, pp. 257–264.

[34] J. Weston, C. Watkins, Support vector machines for multiclass pattern recognition, in: Seventh European Symposium on Artificial Neural Networks, April 1999.

[35] J. Yang, Z. Jin, J. yu Yang, D. Zhang, A.F. Frangi, Essence of kernel Fisher discriminant: Kpca plus lda, Pattern Recognition 37 (10) (2004) 2097–2100.

[36] P. Yin, A. Criminisi, J. Winn, I. Essa, Tree-based classifiers for bilayer video segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, June 2007.

About the Author—THOMAS DESELAERS is a researcher at the Computer Vision Laboratory of the ETH Zürich. He received his diploma and his PhD degree from RWTH Aachen University in Aachen, Germany, in 2004 and 2008, respectively.

About the Author—GEORG HEIGOLD is a researcher and a PhD student at the Human Language Technology and Pattern Recognition group of RWTH Aachen University. He holds a diploma in applied physics from ETH Zurich.

About the Author—HERMANN NEY is a full professor and head of the Human Language Technology and Pattern Recognition group at RWTH Aachen University. He received his diploma in physics from University of Goettingen, Germany, and his doctoral degree in electrical engineering from TU Braunschweig, Germany, in 1982.