Log-Linear Mixtures for Object Class Recognition

Tobias Weyand[1]
Tobias.Weyand@rwth-aachen.de
Thomas Deselaers[12]
deselaers@vision.ee.ethz.ch
Hermann Ney[1]
ney@informatik.rwth-aachen.de

[1] Computer Science Department
RWTH Aachen University
Aachen, Germany

[2] Computer Vision Laboratory
ETH Zurich
Zurich, Switzerland

Abstract

We present log-linear mixture models as a fully discriminative approach to object category recognition which can, analogously to kernelised models, represent non-linear decision boundaries. We show that this model is the discriminative counterpart to Gaussian mixtures and that either one can be transformed into the respective other. However, the proposed model is easier to extend toward fusing multiple cues and numerically more stable to train and to evaluate. Experiments on the PASCAL VOC 2006 data show that the performance of our model compares favourably well to the state-of-the-art despite the model consisting of an order of magnitude fewer parameters, which suggests excellent generalisation capabilities.

1 Introduction

Recognising object classes is one of the fundamental problems in computer vision and many approaches have been presented in the past. In recent years most approaches in this area (with very few exceptions, *e.g*. Torralba et al. [25]) have focused on local image descriptors, *e.g*. [2, 4, 7, 8, 10, 11, 13, 14, 17, 20, 22, 23, 24].

One very common approach to determine whether an object of a certain category is contained in an image is the bag-of-visual-words approach [3, 4, 24], which consists of three steps: (1) local descriptors are extracted from all images, (2) a vector quantisation technique is applied in order to obtain low-dimensional (often in the order of a few thousand) histograms of local features, (3) a (commonly discriminative) model is used as a classifier to determine whether the object of interest is present in the images.

This approach, although in practise leading to good results, has several problems. One problem is the vector quantisation step, in which most appearance information is deliberately discarded. The simplest approach to avoiding hard quantisation is the use of soft cluster assignments (*e.g*. [7]), but this only superficially avoids this problem. Boiman et al. [2] also observed this problem and instead of vector quantisation use a nearest neighbour classifier similar to Paredes et al. [21], which leads to improved performance. Hegerath et al. [10] use a Gaussian mixture classifier directly in order to avoid hard quantisation. Their model,

BMVC 2009 doi:10.5244/C.23.30

although it is generative in principle, is refined using a discriminative training step in order to improve results. Minka [19] noted that discriminative training of generative models is not a theoretically clean method. Lasserre et al. [13] propose a model for object recognition following Minka's ideas, which allows to smoothly fade between discriminative and generative models. Grabner et al. [8] extend a boosted, discriminative model by modifying the training criterion in order to also incorporate generative aspects. Note, that most bag-of-visual-words models also effectively fuse generative and discriminative models: the quantisation step often applies a generative model (*e.g.* Gaussian mixture densities obtained using k-means) in order to obtain a fixed length feature vector which is then classified using a discriminative model.

Another problem with the bag-of-visual-words approaches is that it is generally difficult to incorporate spatial information. A direct approach is to consider triplets of features [27], but most approaches apply post-processing steps in order to verify spatial consistency [11, 22]. Leibe et al. [14] use a visual codebook and each word votes in a continuous voting space for the centre of the object to be detected, which allows to jointly recognise and segment the object of interest with pixel-level accuracy.

In the approach presented here, we address all these problems jointly: (i) we avoid hard vector quantisation, (ii) the model is fully discriminative, and (iii) different information cues, such as appearance and position information, can easily be fused. Furthermore, we show in experiments, that the results obtained with the proposed method are comparable to the state of the art, although the obtained models are very small in the number of parameters and thus very likely robust w.r.t. overfitting. Additionally, we show that the model is closely related to the Gaussian mixture model of Hegerath et al. [10] (*cp.* Section 2.3), but avoids the inelegant training by using a generalised log-linear model of the class posteriors.

2 Log-Linear Mixtures

Log-linear models are well-understood discriminative models for which efficient training methods exist. In log-linear models, the class-posterior $p(c|X)$ for an observation X is directly modelled as

$$p(c|X) \quad = \frac{\exp(g_\theta(X,c))}{\sum_{c'=1}^{C}\exp(g_\theta(X,c'))}, \tag{1}$$

where commonly $g_\theta(X,c) = \alpha_c + \lambda_c^T X$, where $\theta = \{\alpha_c, \lambda_c\}$ are the model parameters to be trained.

One problem with log-linear models is that they can only model linear decision boundaries. To circumvent this problem, the kernel trick can be applied in order to train the model in a higher dimensional feature space [9]. The simplest kernel which can be used is a polynomial kernel of degree 2, which can be implemented directly without applying the kernel trick by changing $g_\theta(X,c)$ to $g_\theta(X,c) = \alpha_c + \lambda_c^T X + X^T \Lambda_c X$ in order to incorporate second order (quadratic) features. The resulting model is commonly called a *log-linear model with second order features*.

Another option to extend this model is to create a log-linear mixture model by incorporating a hidden variable i

$$p(c|X) \quad = \sum_i p(c,i|X) = \frac{\sum_i \exp(g_\theta(X,i,c))}{\sum_{c'=1}^{C}\sum_{i'}\exp(g_\theta(X,i',c'))}, \tag{2}$$

where $g_\theta(X,i,c) = \alpha_{ci} + \lambda_{ci}^T X$. Now, let X be an image, represented by a set of local descriptors $\{x_1^L\} = \{x_1,\ldots,x_L\}$, and assume (analogously to our previous work in [10], cp. also Eq. (12)- (14)) that the individual x_l are independent, then the class posterior becomes

$$p(c|\{x_1^L\}) \quad = \quad \frac{1}{Z(\{x_1^L\})} \prod_{l=1}^{L} \sum_{i=1}^{I} \exp(g_\theta(x_l,i,c)), \tag{3}$$

where $Z(\{x_1^L\}) = \sum_{c'} \prod_l \sum_i \exp(g_\theta(x_l,i,c'))$.

By choosing a sufficiently large number of mixture components I, it is possible to model arbitrarily complex decision boundaries. Additionally, this model can be extended using kernels to add further flexibility. Here, we keep the kernel simple (i.e. linear or polynomial) and investigate the capabilities of the mixture model.

2.1 Training and Maximum Approximation

To train the log-linear mixture model, we compute the derivatives and apply the LBFGS gradient descent method [15]. The training criterion to be maximised is

$$F(\theta) = \sum_n \log p_\theta(c_n|X_n), \tag{4}$$

where X_n is the n-th training image, c_n is the corresponding class, and $p_\theta(c_n,X_n)$ is the model of the class posterior probabilities using the parameters θ. Note, that the training of the proposed model is not a convex optimisation problem.

However, if we apply maximum approximation and keep the assignment i of training observations to the model's components fixed in Eq. (3), we can consider the pairs (c,i) as pseudo classes. Note that maximum approximation is known to be a good approximation for functions from the exponential family and is commonly used in the Viterbi algorithm [6]. Then, the training of the resulting model is identical to training a log-linear model over the pseudo classes. Thus, we obtain a convex optimisation problem. The resulting model is

$$p(c,i|X) \quad = \frac{\exp(g_\theta(X,i,c))}{\sum_{c'}\sum_{i'}\exp(g_\theta(X,i',c'))}. \tag{5}$$

Combining Eq. (3) and Eq. (5) leads to

$$p(c,i_1^L|\{x_1^L\}) \quad = \quad \frac{1}{Z(\{x_1^L\})} \exp(\sum_l g_\theta(x_l,i_l,c)), \tag{6}$$

with $Z(\{x_1^L\})$ unchanged from Eq. (3) and $i_1^L = i_1,\ldots,i_L$ is the assignment of the local descriptors x_l to the model's mixture components.

Using an alternating training method, taking turns between updating the model parameters and the mixture component assignments i_1^L, convergence to a local optimum is guaranteed, since the update of the model parameters with fixed mixture component assignment is a convex problem and the update of the mixture component assignment cannot deteriorate the training criterion.

Additionally, the maximum approximation can also be applied in the denominator, which then leads to the following form of the posterior probability

$$p(c|\{x_1^L\}) \quad = \frac{\exp(\sum_l \max_i g_\theta(x_l,i,c))}{\sum_{c'=1}^{C}\exp(\sum_l \max_{i'} g_\theta(x_l,i',c'))}, \tag{7}$$

which can be evaluated efficiently and which we will use throughout the experiments in the following. This form does not have the semi-convexity property described above but in practice can be optimised as efficiently. Note, that during training, the maximising i is fixed during parameter training.

Regularisation. Regularisation is a common technique in many machine learning approaches to reduce overfitting. We use an L_2 regulariser which leads to the following modified training criterion

$$F(\theta,\gamma) = \sum_n \log p(c_n|X_n) - \gamma||\theta||^2, \tag{8}$$

where $||\theta||$ denotes the norm of all model parameters in one vector, and γ is a weighting factor that can be used to control the strength of the regularisation. This is equivalent to imposing an $\mathscr{N}(0,1)$-prior on all model parameters in Bayes' learning.

2.2 Cue Fusion

One advantage of the proposed model over the closely related Gaussian mixtures is that the log-linear approach can very easily be extended to incorporate features of different types. Related approaches (under the name "*maximum entropy approach*") are frequently used for model- and feature combination in natural language processing [1].

Different Local Descriptors. Assuming, we have S different cues with L_s individual local features each, we can easily extend the model to incorporate all the cues, without explicitly having to account for their scaling and location in feature space, resulting in the following form for $p(c|X)$:

$$p(c|X) = \frac{\exp(\sum_s \sum_{l=1}^{L_s} \max_i\{\alpha_{csi} + \lambda_{csi}^\mathsf{T} x_{sl}\})}{\sum_{c'=1}^{C} \exp\left(\sum_s \sum_{l=1}^{L_s} \max_i\{\alpha_{c'si} + \lambda_{c'si}^\mathsf{T} x_{sl}\}\right)}. \tag{9}$$

Note that in this model, the descriptors of the individual cues do not have to be extracted from the same interest points but that they can be entirely unrelated.

Spatial Information. Analogously to the fusion of different appearance cues it is also possible to incorporate other cues such as the spatial layout of the individual features. In order to integrate spatial information suitably it is necessary to formulate features that describe the neighbourhood of the considered point.

In our representation, the neighbourhood of a point is described by the set of descriptors which were extracted at points from its neighbourhood.

That is, a local feature x_l from position $(uv)_l$ is enriched by the set $NN_k(x_l)$ of the descriptors x_γ extracted at the k nearest neighbours $(uv)_\gamma$ of its extraction position.

$$NN_k(x_l) = \left\{ x_\gamma \mid (uv)_\gamma \in \arg_k \min_{\gamma \in \{1,\ldots,L\}} \{||(uv)_l - (uv)_\gamma\}|| \right\}. \tag{10}$$

Given this set of neighbours we extend the discriminant function $g(x,c)$ to incorporate their appearance, which we model using a mixture:

$$g_\theta(x,c) = \sum_{l=1}^{L} \max_i \left\{ \alpha_{ci} + \lambda_{ci}^\mathsf{T} x_l + \sum_{x_\gamma \in NN_k(x_l)} \max_j \left\{ \beta_{cij} + \mu_{cij}^\mathsf{T} x_\gamma \right\} \right\}. \tag{11}$$

The new parameters β_{cij} and μ_{cij} allow us to capture appearance variation among the typical neighbours of a model component. For consistency, we use the maximum approximation analogously to Eq. (7). The size k of the set of neighbours indirectly models the size of the local structure that is considered. Alternatively, one could determine the set of neighbours using a distance threshold resulting in different numbers of neighbours for each x_l.

2.3 The Relationship to Gaussian Mixtures

The presented model has an interesting relationship to Gaussian mixtures. In the following, we show, how Gaussian mixtures can be transformed into the log-linear form of our model, and how one can be interpreted as the respective other [12]:

We start from the posterior of a Gaussian mixture model

$$\begin{aligned}
p(c|\{x_1^L\}) &= \frac{1}{Z(x)} p(c) \prod_{l=1}^{L} \sum_{i=1}^{I} p(i|c) \mathcal{N}(x_l|\mu_{ci}, \Sigma_{ci}) && (12)\\
&= \frac{1}{Z(x)} p(c) \prod_{l=1}^{L} \sum_{i=1}^{I} \frac{p(i|c)}{|2\pi\Sigma_{ci}|^{\frac{1}{2}}} \exp\left(-\tfrac{1}{2}(x_l-\mu_{ci})^T \Sigma_{ci}^{-1}(x_l-\mu_{ci})\right) && (13)\\
&= \frac{1}{Z(x)} \prod_{l=1}^{L} \sum_{i=1}^{I} \exp\left(\tfrac{1}{L}\log(p(c)) + \log(p(i|c)) - \tfrac{1}{2}\log(|2\pi\Sigma_{ci}|)\right. && (14)\\
&\qquad \left. -\tfrac{1}{2}\left(x_l^T \Sigma_{ci}^{-1} x_l - 2x_l^T \Sigma_{ci}^{-1}\mu_{ci} + \mu_{ci}^T \Sigma_{ci}^{-1}\mu_{ci}\right)\right)
\end{aligned}$$

Then, it can be seen that choosing the parameters $\alpha_{ci}, \lambda_{ci}$, and Λ_{ci} as follows leads to the log-linear model in Eq. (3).

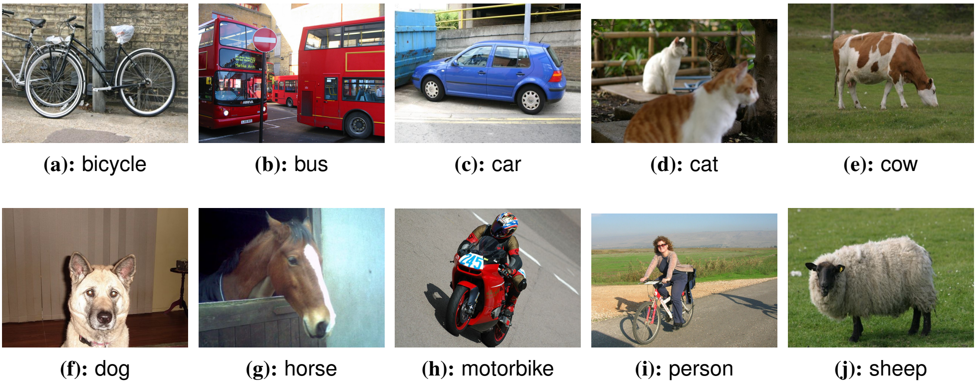
$$\begin{aligned}
\alpha_{ci} &= \frac{1}{L}\log(p(c)) + \log(p(i|c)) - \frac{1}{2}\log(|2\pi\Sigma_{ci}|) - \frac{1}{2}\mu_{ci}^T \Sigma_{ci}^{-1}\mu_{ci} && (15)\\
\lambda_{ci} &= \Sigma_{ci}^{-1}\mu_{ci} && (16)\\
\Lambda_{ci} &= -\frac{1}{2}\Sigma_{ci}^{-1}. && (17)
\end{aligned}$$

By solving the above equations for the parameters μ, Σ, and $p(i|c)$ (and possibly making Σ_{ci} positive definite by adding a positive constant to the diagonal), a Gaussian mixture can be obtained from a log-linear mixture.

The resulting model uses a quadratic decision function $g_\theta(\cdot)$, because we started from a Gaussian model with full, class-specific covariance matrix. If the Gaussian model uses class and cluster pooled covariances, the second order terms cancel out, and the resulting model is *first order* log-linear.

3 Experimental Evaluation

In this section, we evaluate our model and examine the influence of various parameters on its performance.


(a): bicycle **(b):** bus **(c):** car **(d):** cat **(e):** cow

(f): dog **(g):** horse **(h):** motorbike **(i):** person **(j):** sheep

Figure 1. PASCAL 2006 Example Images. On example image for each of the 10 classes considered in the PASCAL 2006 evaluation.

3.1 Dataset

We use the PASCAL Visual Object Classes Challenge (VOC) 2006 dataset [5][1] for evaluation. The set consists of ten classes comprising man-made objects, animals and people. For each class, the task is to detect whether an object of the class is present in an image. For each task, about 2600 training images and about 2700 testing images are given. The performance on this dataset is evaluated in AUC on the ROC curve. An example image from each of the ten classes considered in the PASCAL VOC 2006 is given in Figure 1.

3.2 Experimental Setup

We employ a wavelet interest point detector [16] and extract appearance-based features of size 19x19 pixels at the 200 most salient interest points. The patches are reduced to a dimensionality of 40 using PCA transformation. For all our experiments, the regularisation weight was tuned on the training data. We found that, in particular for the setups with linear $g_\theta(\cdot)$, the regularisation weights are very small. For the experiments in the following we initialise the log-linear mixture from a mixture of Gaussians in order to have a good starting point for the training procedure.

In the following we present the experimental evaluation of the proposed model. First, we evaluate the performance of our model depending on the number of mixture components used and the impact of first and second order features. We also evaluate the impact of spatial and additional appearance cues. Finally, we compare the obtained results to state of the art methods and to the closely related Gaussian mixture densities.

Number of mixture components. First we evaluate the basic parameters of the proposed model and compare it to the standard log-linear models from which we started. In Figure 2 (a), the obtained AUCs for different numbers of mixture components are shown. The leftmost point in the plot corresponds to standard log-linear models. It can be seen that a very small number of mixture components leads to suboptimal results. This is probably due to insufficient flexibility in the model. By adding more mixture components, flexibility is added to the model and results are improved until they level out at about 32 mixture components per class. Thus, in the following, we perform all experiments with this setting.

[1] http://www.pascal-network.org/challenges/VOC/voc2006

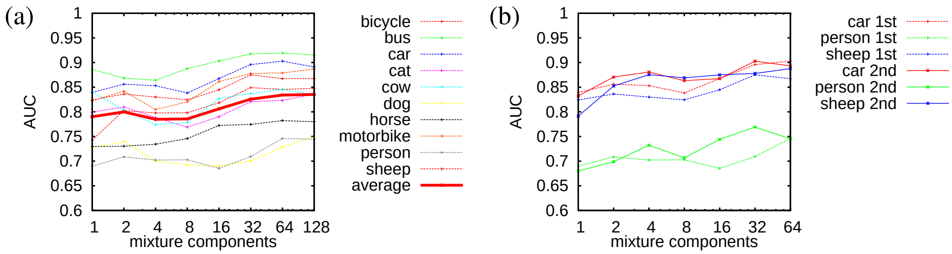


Figure 2. **(a) Number of mixture components.** The plot shows obtained results as a function of the number of mixture components for the individual classes (in dotted lines) and for the average over all classes (think solid line). **(b) Second order models.** A comparison of models using first and second order features respectively for three classes from the PASCAL dataset

For better insight into the obtained results, in the following, we give the average over all 10 classes (AVG_{10}) and the results for the three classes "car" (as a representative for a relatively easy, rigid, man-made class), "person" (as a representative for a strongly articulated class), and "sheep" (as a representative for a natural, but still rather rigid class).

Second Order Features. Another option to add flexibility to the model is to incorporate non-linear features into the discriminant function g_θ. In Figure 2 (b), we compare the performance of models with linear and quadratic features for three of the ten classes. It can be seen that the models with second order features slightly outperform those without for small numbers of mixture components. With 32 or more mixture components the models perform approximately equally well, which shows that the added flexibility of the second order model is not advantageous. In the following experiments we will use linear models, since they are faster, more reliably trainable, and, given a sufficient number of mixture components, perform equally well.

Additional Appearance Features. As described in Section 2.2, it is possible to fuse multiple cues in the proposed model. We evaluate this by using combinations of different extraction positions. In addition to the wavelet-based interest point detector from the baseline setup, we use a difference of Gaussians detector and random extraction positions [20] in order to also capture homogeneous image regions.

The results from these experiments are shown in Table 3.2. It can be seen that the additional extraction points clearly improve categorisation. On average, the improvement is 0.03 points AUC, and the improvement is consistent throughout all tasks. For comparison we also performed an experiment where all cues are treated jointly, which is possible in this setup since we extract the same type of descriptor. In further experiments we extracted more wavelet-based interest points per image, but none of these experiments led to any improvement over the baseline results. Thus, we conclude that the separate modelling of these cues allows for a more suitable representation of the data and thus to better recognition accuracy.

Note that it is also possible to incorporate different descriptors such as colour descriptors [26], SIFT features [17], or MSERs [18], but we omit this here for the sake of brevity.

Spatial Features. Results from different setups incorporating the spatial structure of the neighbourhood are given in Table 3.2. To allow for separating the effect from the different appearance cues, we run this experiment on the baseline setup. In these experiments we

Table 1. The AUC results obtained using combinations of different extraction points on three classes of the PASCAL 2006 task and for the average over all 10 classes.

Wavelet	DoG	Random	car	person	sheep	AVG$_{10}$
X			0.88	0.72	0.87	0.82
X	X		0.90	0.75	0.89	0.85
X	X	X	0.91	0.76	0.91	0.86

Table 2. Results using the spatial layout information in our model on the three PASCAL tasks car, person and sheep, and the average over all 10 tasks. For these experiments we kept the number of neighbours at 5 and varied the number J of spatial mixture components in Eq. (11).

J	car	person	sheep	AVG$_{10}$
1	0.88	**0.73**	0.86	0.81
2	**0.90**	0.72	0.84	**0.82**
3	0.87	0.73	0.85	0.80
4	0.88	0.71	0.86	0.81

used $k = 5$ neighbour-points for each descriptor x_l, leading to an average distance of 25 pixels between a descriptor and its neighbours which is slightly below the size of the object structures in this dataset. Like in the baseline setup, we chose a fixed number of 32 mixture components per class. Each of these components contains J *spatial* mixture components which model the appearance of the neighbourhood. It can be seen that setting $J = 2$ seems to be the best choice. This is also consistent with the observation that in general a relatively small number of mixture components is sufficient to obtain good performance. The spatial extension has shown to help with the classification of articulated classes like cat, dog and person as well as classes that show strong variations in appearance, like bus or car. For more uniformly shaped classes like cow and sheep, the extension does not lead to any improvement but might even deteriorate the performance due to overfitting.

In informal experiments we also experimented with different values for k and found the results to be relatively robust between 3 and 10 neighbours (corresponding to neighbourhoods of 10-50 pixels). Using more or less neighbours, hardly any improvement is possible. If too few neighbours are considered, the neighbourhood is not represented well enough, and if too many neighbours are considered the model is weakened in particular for the articulated classes because the considered neighbourhood is too large and thus global spatial arrangements are considered which might not be suitable for these classes.

Comparison to GMDs As described above, the proposed model is closely related to the Gaussian mixtures approach, and thus it opens the possibility to directly compare the discriminative and the generative model. In Figure 3.2, we show the obtained AUC results for three of the ten PASCAL classes for both GMDs and log-linear mixtures with different numbers of mixture components.

It can be seen that the discriminative log-linear mixtures outperform the generative Gaussian models for any number of mixture components. In particular, even log-linear mixtures with few mixture components outperform GMDs with more mixture components. For example on the person task, a GMD with 128 components scores 0.69 AUC whereas a log-linear

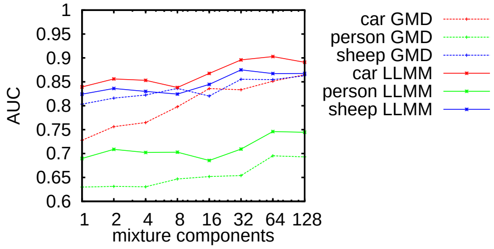


Figure 3. GMDs vs. Log-linear Mixtures. The AUC obtained using different numbers of mixture components in both, GMDs and log-linear mixtures.

Table 3. Comparison of the results of log-linear mixtures on the PASCAL VOC 2006 task with other approaches.

approach	car	person	sheep	AVG_{10}	params.	time
(single) log-linear model	0.86	0.67	0.82	0.79	82	1
log-linear mixture model (baseline)	0.88	0.72	0.87	0.82	2624	35
+ second order features	0.90	**0.77**	0.88	0.85	55104	130
+ additional appearance features	**0.91**	0.76	**0.91**	**0.86**	7744	40
+ spatial layout information	0.90	0.72	0.84	0.82	7872	40
GMDs (tuned + spatial + disc) [10]	0.94	0.72	0.89	0.86	248832	
BoF/Spatial Pyramid Everingham et al. [5]	**0.98**	**0.86**	**0.96**	**0.94**		

mixture with 2 components already achieves a score of 0.71 AUC. This shows that for the task at hand the discriminative model is better suited and is also consistent with the observation of [10] that the discriminative refinement of their model is crucial to obtain good results.

3.3 Comparison to the State of the Art

In Table 3, we compare the obtained results to the state of the art. The top block of the table shows the performance of the different variants of our proposed method and the bottom block shows some of the best results from the literature. For a better overview, we again show only three classes and the average score for all ten PASCAL 2006 classes. The number of model parameters and the time for training the model (relative to the simple log-linear model) are given as well.

It can be seen that our baseline model, consisting of only $2 \cdot 32 \cdot (1+40) = 2{,}624$ parameters per sub-task (we consider binary classifiers, 32 model components, 40 dimensional data) already achieves competitive results and clearly outperforms the simple log-linear model without mixtures, which only has $2 \cdot (1+40) = 82$ parameters. However by adding second order features, the number of parameters grows quadratic in the dimension of the feature space $(2 \cdot 32 \cdot (1+40+\frac{40 \cdot 41}{2}) = 55104$, but the improvement in AUC is relatively low considering the increase of parameters by more than an order of magnitude. On the other hand, by adding additional appearance descriptors, we have $2 \cdot 32 \cdot (1+3 \cdot 40) = 7744$ parameters and obtain better results with less parameters which suggests better generalisation capabilities.

By adding spatial layout information, we have $(2 \cdot 32 \cdot (1+40+(2 \cdot (1+40))) = 7872$ parameters and the results are improved for strongly varying classes.

The bottom block of Table 3 shows results from the evaluation [5]. The GMD result from [10] used a combination of 3 individual models, each with 1024 mixture components and diagonal but unpooled covariances leading to $3 \cdot 1024 \cdot (1+40+40) = 248832$ parameters and obtains similar performance. The method “Bag of Features/Spatial Pyramid” is the best result from the evaluation and is joint work by Queen Mary University, London and INRIA Rhone-Alpes. They use a strongly tuned bag-of-visual words approach with a χ^2-pyramid matching kernel. The total number of parameters is unknown to us, but will likely be rather large because the SVM will most probably consist of a huge amount of support vectors. At the moment, our method cannot beat this method. However, by adding more advanced features such as SIFT features and possibly some more advanced spatial models, there is still room for improvement.

4 Discussion and Conclusion

We proposed log-linear mixture models as an extension in order to allow for non-linear decision boundaries in log-linear models and showed that the proposed model is the discriminative counterpart of Gaussian mixture models.

The proposed model allows to create efficient and small (in the number of parameters) models, which still outperform Gaussian approaches and compare favourably well to the state of the art. One key advantage of the proposed model is that it is easy to fuse arbitrary cues such as spatial information or different descriptors in a theoretically sound way without making the model computationally expensive and without imposing any badly funded assumptions.

In the future, we intend to evaluate different descriptors and to improve the modelling of spatial relationships of the individual extracted features.

References

[1] 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.

[2] Oren Boiman, Eli Shechtman, and Michal Irani. In defense of nearest-neighbor based image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, June 2008.

[3] 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.

[4] Gyuri Dorkó and Cordelia Schmid. Object class recognition using discriminative local features. Rapport de recherche RR-5497, INRIA - Rhone-Alpes, February 2005.

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

[6] G. D. Fornay. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[7] J.C. Gemert, J.M. van Geusebroek, C.J. Veenman, and A.W.M. Smeulders. Kernel codebooks for scene categorization. In *European Conference on Computer Vision*, Marseille, France, 2008.

[8] Helmut Grabner, Peter M. Roth, and Horst Bischof. Eigenboosting: Combining discriminative and generative information. In *IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, USA, jun 2007. IEEE.

[9] Trevor Hastie and Robert J. Tibshirani. *Generalized Additive Models*. Chapman & Hall, CRC, 1990.

[10] 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.

[11] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision*, 2008.

[12] 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 2002.

[13] Julia A Lasserre, Christopher M Bishop, and Thomas P Minka. Principled hybrids of generative and discriminative models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 87–94, New York City, NY, USA, June 2006. IEEE.

[14] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3):259–289, 2008.

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

[16] Etienne Loupias, Nicu Sebe, S. Bres, and J.M. Jolion. Wavelet-based salient points for image retrieval. In *International Conference on Image Processing*, volume 2, pages 518–521, Vancouver, Canada, September 2000.

[17] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, February 2004.

[18] J Matas, O Chum, Martin U, and Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, volume 1, pages 384–393, 2002.

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

[20] Eric Noiwak, Frédéric Jurie, and Bill Triggs. Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision*, Lecture Notes in Computer Science, Graz, Austria, May 2006.

[21] R. Paredes, J. Perez-Cortes, A. Juan, and E. Vidal. Local representations and a direct voting scheme for face recognition. In *Workshop on Pattern Recognition in Information Systems*, pages 71–79, Setúbal, Portugal, July 2001.

[22] J. Philbin, O. Chum, M. Isard, J. Sivic, and A Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[23] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost for image understanding: Multi-class ob ject recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23, January 2009.

[24] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, volume 2, pages 1470–1477, October 2003.

[25] Antonio Torralba, Rob Fergus, and Yair Weiss. Small cdes and large databases for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, June 2008. IEEE.

[26] Joost van de Weijer, Theo Gevers, and Arnold W.M. Smeulders. Robust photometric invariant features from the color tensor. *Trans. Image Proc.*, 15(1):118–127, 2006.

[27] C. Lawrence Zitnick, Jie Sun, Richard Szeliski, and Simon Winder. Object instance recognition using triplets of features symbols. Technical report, Microsoft Research, Redmond, WA, USA, 2007.