Eigenboosting: Combining Discriminative and Generative Information *

Helmut Grabner Peter M. Roth Horst Bischof
Graz University of Technology
Institute for Computer Graphics and Vision
{hgrabner, pmroth, bischof}@icg.tugraz.at

Abstract

A major shortcoming of discriminative recognition and detection methods is their noise sensitivity, both during training and recognition. This may lead to very sensitive and brittle recognition systems focusing on irrelevant information. This paper proposes a method that selects generative and discriminative features. In particular, we boost classical Haar-like features and use the same features to approximate a generative model (i.e., eigenimages). A modified error function for boosting ensures that only features are selected that show a good discrimination and reconstruction. This allows a robust feature selection using boosting. Thus, we can handle problems where discriminant classifiers fail while still retaining the discriminative power. Our experiments show that we can significantly improve the recognition performance when learning from noisy data. Moreover, the feature type used allows efficient recognition and reconstruction.

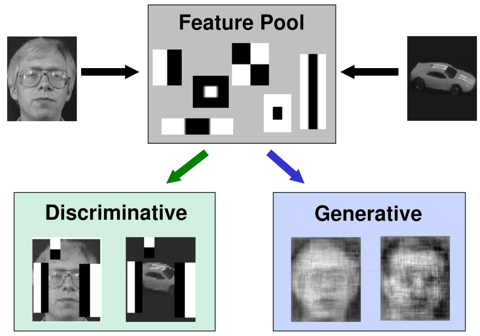

Figure 1. Combining discriminative and generative information by using a shared feature pool. In addition to discriminative classifying the features can be used to reconstruct a previously learned object (face) while the reconstruction for other objects (*e.g.*, cars) fails completely.

1. Introduction

When computing a classifier for object recognition one faces two main philosophies: generative and discriminative models. Formally, the two categories can be described as follows: Given an input x and and a label y then a generative classifier learns a model of the joint probability $p(x, y)$ and classifies using $p(y|x)$ which is obtained by using Bayes' rule. In contrast, a discriminative classifier models the posterior $p(y|x)$ directly from the data or learns a map from input to labels: $y = f(x)$.

Generative models such as principal component analysis (PCA) [11], independent component analysis (ICA) [10], or non-negative matrix factorization (NMF) [13] try to find a suitable representation of the original data (by approximating the original data by keeping as much information as possible). As they provide sufficient reconstructive power these methods are capable of dealing with missing or occluded pixels. Thus, several methods were proposed that allow robust recognition (*e.g.*, [4, 14]) as well as robust learning (*e.g.*, [5, 23]).

In contrast, discriminant classifiers such as linear discriminant analysis (LDA) [3], support vector machines (SVM) [27], or boosting [7] were designed for classification tasks. Given the training data and the corresponding labels the goal is to find optimal decision boundaries. Compared to generative methods this allows to train more specific classifiers having a higher recognition rate. In fact, several studies (*e.g.*, [3, 17]) have shown that discriminative classifiers outperform generative models (if enough training data is available). Thus, many applications use discriminative classifiers instead of generative classifiers. Compared to generative models discriminative models have two main drawbacks: (a) discriminant models are not robust, whether

*This work has been supported by the Austrian Joint Research Project Cognitive Vision under projects S9103-N04 and S9104-N04 and the FFG project AUTOVISTA 813395 under the FIT-IT programme. Part of this work has also been carried out as part of the K-plus Competence center ADVANCED COMPUTER VISION funded under the K-plus program. In addition, this work has been sponsored by the EU FP6-507752 NoE MUSCLE IST project.

in the training nor in the recognition stage and (b) a huge amount of labeled training data is necessary.

To overcome these drawbacks several approaches (*e.g.*, [6, 9, 15, 16, 19, 20]) have been proposed that combine discriminative and generative models to get the best of both worlds: the discriminative power and the robustness. In addition, the paper [12] provides a theoretical discussion on this topic. Most of these approaches are based on two-stages (*e.g.*, [9, 15, 16, 19]). In the first stage a generative model is estimated and in the second stage a discriminant classifier is built from the generative model. Holub *et al*. [9] proposed to use a probabilistic constellation model as generative model and to use SVM on Fisher Scores to estimate the discriminative classifier. Lin *et al*. [16] applied a probabilistic PCA to compute a distribution for the positive samples in the first stage. In the second stage they estimate a new distribution for the negative samples by learning a linear projection. Other authors proposed to use a clustering algorithm in the generative stage and a neural network classifier (*e.g.*, multi-layer perceptron [15], pairwise neural network [19]) in the discriminative stage. In contrast, the problem of robust discriminative classification was addressed by Fidler *et al*. [6]. They proposed a robust LDA approach that constructs a basis that contains all discriminative information but in addition also contains sufficient reconstructive information to enable robust reconstructions. Boosting Haar-like features allows to train efficient classifiers. Thus, Roth *et al*. [20] proposed the conservative learning framework where they additionally estimate a PCA model which serves as supervisor for the boosted discriminative model and thus ensures robustness.

All methods described above combine discriminative and generative information on image level only. Moreover, most of them are multistage methods where the (final) discriminative classifier is trained on a (pre-processed) generative model. The main contribution of this paper is to propose a method that combines a discriminative model and a generative model on the feature level. In order to do this we need features that are discriminative and provide reconstructive feasibilities at the same time. For example, Haar-like features fulfill both criteria. It is well known [28] that this feature type can be used to train powerful discriminative classifiers. In addition, Tao *et al*. [25] showed that binary bases (in fact, Haar-like features can be considered as binary basis functions) can be used to reconstruct gray-value images. The discriminative and generative power of Haar-like features is shown in Figure 1 by two specific examples. Given a face model learned from Haar-like features a face can be described and reconstructed by the same features which is not the case for a non-face image (*e.g.*, a car).

Having features that cover both, discriminative and generative information, we can combine discriminative and generative information on the feature level. In particular, we apply boosting for feature selection on Haar-like features. Therefore, we define a new error-function that additionally includes the generative information of a feature which allows a robust feature selection using boosting. Thus, we can learn a discriminative classifier even from degraded input images.

The remaining paper is organized as follows: Section 2 reviews some theoretic issues that are relevant later on. Section 3 introduces the new Eigenboosting method and defines the modified discriminative/generative boosting error-function. In Section 4 the proposed method is evaluated on two well known databases. Finally, conclusions are drawn in Section 5.

2. Preliminaries

Before we introduce the Eigenboost method we need to discuss the two main components of the system and to introduce the notation.

2.1. Boosting for Feature Selection

In general, Boosting is a widely used technique in machine learning for improving the accuracy of any given learning algorithm. In fact, boosting converts a weak learning algorithm into a strong one. In the following we focus on the AdaBoost algorithm that was introduced by Freund and Schapire [7].

Given a training set $\mathcal{X} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_L, y_L)\}$, where $\mathbf{x}_i \in \mathbb{R}^P$ is a sample and $y_i \in \{-1, +1\}$ is the corresponding label, and an initial uniformly distributed weight distribution with $p(\mathbf{x}_i) = \frac{1}{L}$. In each boosting iteration n a new weak classifier[1] $h_n^{weak}(\mathbf{x}) : \mathbf{x} \to [-1, 1]$ and the corresponding weight α_n is calculated according to the training error over all samples $\mathcal{X}$ and $p(\mathbf{x})$. In addition, the probability $p(\mathbf{x})$ is updated such that it is increased for samples that were misclassified and decreased for samples that were classified correctly. Thus, the algorithm focuses on the difficult examples.

The process is repeated and at each boosting iteration a new weak classifier is added until a certain stopping condition is met (*e.g.*, a given number of weak classifiers). Finally, a strong classifier $h^{strong}(\mathbf{x})$ is computed as linear combination of a set of N weak classifiers $h_n^{weak}(\mathbf{x})$:

$$h^{strong}(\mathbf{x}) = \mathrm{sign}\left(\sum_{n=1}^{N} \alpha_n h_n^{weak}(\mathbf{x})\right). \tag{1}$$

The weights α_n can be obtained by

[1] A weak classifiers is a classifier that has to perform only slightly better than random guessing (*i.e.*, for a binary decision task, the error rate must be less than 50%). It is obtained by applying a learning algorithm (*e.g.*, by applying statistical learning for a decision stump).

$$\alpha_n = \frac{1}{2}\ln\left(\frac{1+r_n}{1-r_n}\right), \;\; r_n = \sum_{l=1}^{L} p(\mathbf{x}_l) h_n^{weak}(\mathbf{x}_l) y_l \quad (2)$$

which was shown by Schapire and Singer in [22] where they proved strong bounds for the training and generalization error of AdaBoost. Hence, boosting minimizes the error on the training set:

$$\frac{1}{L}\sum_{l=1}^{L}\left[h^{strong}(\mathbf{x}_l) \neq y_l\right] = \frac{1}{L}\sum_{l=1}^{L} h^{strong}(\mathbf{x}_l) y_l. \quad (3)$$

This error is bounded by

$$\frac{1}{L}\sum_{l=1}^{L} exp\left(-\sum_{n=1}^{N}\alpha_n h_n^{weak}(\mathbf{x}_l) y_l\right) = \prod_{n=1}^{N} Z_n, \quad (4)$$

where

$$Z_n = \sum_{l=1}^{L} p(\mathbf{x}_l) exp(-\alpha_n \underbrace{h_n^{weak}(\mathbf{x}_l) y_l}_{e_n}) \quad (5)$$

and $h_n^{weak}(\mathbf{x}_l) \in [-1, +1]$. Minimizing Z_n on each round n, boosting greedily minimizes the training error which finally yields the weights defined in (2). Note, the optimization problem and therefore the weights α_n are related to the error of the weak hypotheses:

$$e_n = h_n^{weak}(\mathbf{x}_l) y_l. \quad (6)$$

Boosting for feature selection was first introduced by Tieu and Viola [26] and has been widely used for different applications (*e.g.*, face detection [28]). The main idea is that each feature corresponds to a single weak classifier and boosting selects an informative subset from these features.

Given a set of possible features $\mathcal{F} = \{f_1, ..., f_M\}$ and a weak learning algorithm $\mathcal{L}$. In each iteration step n all features f_j are evaluated on all positive and negative training samples and a hypothesis is generated by applying the learning algorithm $\mathcal{L}$. Finally, the best hypothesis is selected which forms the weak classifier h_n^{weak}. Thus, the selection of a feature depends the error e_n that was defined in (6) and (2), respectively.

2.2. Principal Component Analysis

Given a set of input images $\mathbf{X} = [\mathbf{x}_1, \ldots \mathbf{x}_L] \in \mathbb{R}^{P \times L}$, where $\mathbf{x}_i = [x_{1i}, \ldots, x_{Pi}]^T \in \mathbb{R}^P$ is an individual image represented as a vector. Then the PCA projection $\mathbf{U} = [\mathbf{u}_1, \ldots \mathbf{u}_L] \in \mathbb{R}^{P \times L}$ is obtained by solving the eigenproblem for the covariance matrix $\mathbf{C} \in \mathbb{R}^{P \times P}$ of $\mathbf{X}$ or more efficiently by solving Singular Value Decomposition (SVD) of $\mathbf{X}$ assuming that $\mathbf{X}$ is mean normalized.

The columns of $\mathbf{u}_i = [u_{1i}, \ldots, u_{Pi}]^T \in \mathbb{R}^P$ of $\mathbf{U}$, *i.e.*, the eigenvectors, are arranged in decreasing order with respect to the corresponding eigenvalues. Usually, only K, $K < L$, eigenvectors (those with the largest eigenvalues) are needed to represent a given image $\mathbf{x}$ to a sufficient degree of accuracy as a linear combination of eigenvectors $\mathbf{u}_i$:

$$\tilde{\mathbf{x}} = \sum_{i=1}^{K} a_i \mathbf{u}_i \;\; . \quad (7)$$

The coefficients a_i can be calculated by a standard projection

$$a_i = \mathbf{u}_i^T \mathbf{x} = \sum_{j=1}^{P} u_{ji} x_j \;\; , \;\; i = 1 \ldots K, \quad (8)$$

or, as a robust procedure [14], by solving a system of linear equations

$$x_{r_i} = \sum_{j=1}^{K} a_j u_{r_i,j} \;\; , \;\; i = 1 \ldots Q, \quad (9)$$

evaluated at $Q \geq K$ points $\mathbf{r} = (r_1, \ldots, r_Q)$.

3. Eigenboosting

3.1. Image Representation

The goal of this paper is to train a classifier that takes into account discriminative as well as generative information. Therefore, we need a common low-level representation of the data that covers both aspects. In particular, we decided to use Haar-like features for this purpose and defined a shared feature pool $\mathcal{F} = \{f_1, ..., f_M\}$ containing Haar-like features. It is well known that boosting for feature selection on Haar-like features allows to train powerful discriminative classifiers (*e.g.*, [28]). Thus, the features f_j are used to build weak hypothesis $h_j^{weak}(\mathbf{x})$ following the theory of boosting for feature selection as described before.

But these features can also be considered as binary basis functions that describe a visual dictionary. Tao *et al.* [25] showed that binary bases can be used to reconstruct gray-value images which can efficiently be done by using integral images [28]. Examples of such basis functions obtained from Haar-like features are shown in Figure 2.

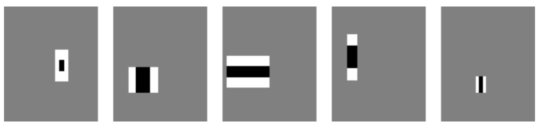

Figure 2. Haar-like binary basis functions: for illustration the original values were normalized; a black pixel represents -1, a white pixel 1, and a gray pixel 0.

An extension of this approach called binary PCA (B-PCA) was proposed by Tang and Tao [24]. The main idea is first to approximate eigenimages from binary basis functions and then to use these eigenimages to reconstruct the original image. This theory also holds for basis functions defined by Haar-like features which allows us to approximate an eigenimage image **u** by

$$\tilde{\mathbf{u}} = \sum_{i=1}^{M} b_i f_i, \tag{10}$$

where f_i is a single feature from the feature pool $\mathcal{F}$. The coefficient b_i can be estimated by

$$b_i = \mathcal{F}^{\dagger} \mathbf{u}_i, \tag{11}$$

where $\mathcal{F}^{\dagger}$ denotes the pseudoinverse of $\mathcal{F}$. In fact, the obtained bases are not orthogonal but for practical purposes the accuracy is sufficient to reconstruct images by PCA projections; we can approximate the eigenbasis to a desired accuracy (increasing the number of features will increase the accuracy of the reconstructions) from the over-complete visual dictionary.

Having this eigenbasis we can reconstruct a given input image **x** by substituting (10) into (7) which yields

$$\tilde{\mathbf{x}} = \sum_{k=1}^{K} a_k \sum_{i=1}^{M} b_{k,i} f_i. \tag{12}$$

The coefficients a_k can be estimated by the standard projection (8) or robustly by (9). Hence, we can easily introduce robustness into our approach which, in fact, is not possible for a standard discriminative approach.

3.2. Discriminative/Generative Model

Given a training set $\mathcal{X}$ of total L samples with L_{pos} positive samples and L_{neg} negative samples and the previously defined feature pool $\mathcal{F}$. To finally train a discriminant classifier we combine a discriminative model and a generative model using a boosting framework. The proposed discriminative/generative learning framework is depicted in Figure 3.

Discriminative classifiers h_j^d are trained using features from $\mathcal{F}$ from positive and negative training samples. In parallel and independently using PCA an eigenbasis is estimated (from the positive samples only). The obtained eigenvectors are approximated by (10) using features from the shared feature pool $\mathcal{F}$. Finally, a discriminative classifier is trained by boosting for feature selection.

To combine the discriminative and generative information we define a new error-function for boosting for feature selection. Therefore, for each feature f_j we first have to estimate the discriminative error e_j^d and the generative error e_j^g.

Adapting the error function for boosting was previously addressed by other authors, *e.g.*, by Hertz *et al.* [8] to learn kernel functions (*KernelBoost* and *DistBoost*) or by Avidan [2] to include spatial information.

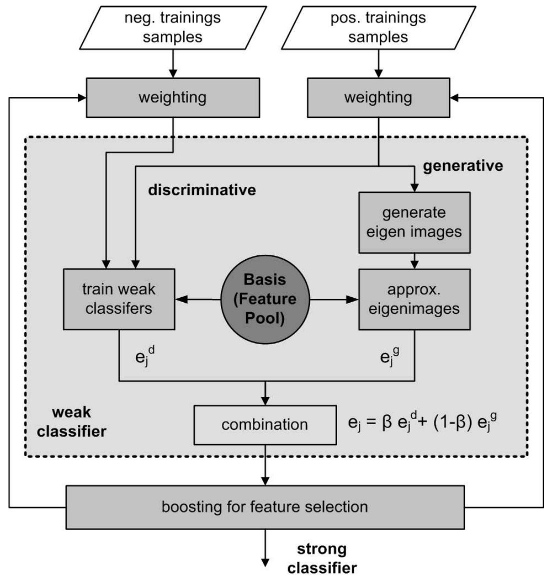

Figure 3. Eigenboosting framework for robust feature selection.

Discriminative Error

The discriminative classifier $h_j^d(\mathbf{x})$ is defined by

$$h_j^d(\mathbf{x}) = p_j \cdot \mathrm{sign}(f_j(\mathbf{x}) - \theta_j), \tag{13}$$

where the threshold θ_j and the parity p_j are defined by

$$\begin{aligned} \theta_j &= |\mu^+ + \mu^-|/2, \\ p_j &= \mathrm{sign}(\mu^+ - \mu^-). \end{aligned} \tag{14}$$

The mean values μ^+ and μ^- are estimated by computing the response for each feature f_j for all images $\mathbf{x}_l$. Based on the decision of the weak hypothesis $h_j^d(\mathbf{x}_l)$ the discriminative error e_j^d for the feature f_j on all training examples can directly be estimated by

$$e_j^d = \frac{1}{L} \sum_{l=1}^{L} h_j^d(\mathbf{x}_l) y_l \tag{15}$$

which is related to the error derived in (3).

Generative error

To estimate the generative error (from the positive samples only) we consider the error that would be obtained without the feature f_j. Let

$$\tilde{\mathbf{x}}_{\bar{j}} = \sum_{k=1}^{K} a_k \sum_{m \neq j} b_{k,m} f_m \tag{16}$$

be the reconstruction of the original image **x** using $\{\mathcal{F} \backslash \{f_j\}\}$ and $\tilde{\mathbf{x}}$ be the reconstruction obtained by (12) (*i.e.*, by using the full basis $\mathcal{F}$). As $\tilde{\mathbf{x}}$ is the optimal reconstruction that can be obtained from the basis $\mathcal{F}$ (using a pre-specified number of eigenimages K) we can consider $||\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_{\overline{j}}||$ as an error measure for a single feature f_j. From (12) and (16) we get

$$\begin{aligned} \left|\left|\sum_{k=1}^{K} a_k \sum_{m=1}^{M} b_{k,m} f_m - \sum_{k=1}^{K} a_k \sum_{m \neq j} b_{k,m} f_m\right|\right| = \\ = \left|\left|\sum_{k=1}^{K} a_k b_{k,j} f_j\right|\right| = \left|\sum_{k=1}^{K} a_k b_{k,j}\right| ||f_j|| \, . \end{aligned} \tag{17}$$

Thus, the training error for a feature f_j is related to

$$\left|\sum_{k=1}^{K} a_k b_{k,j}\right| . \tag{18}$$

Hence, we can estimate the training error for a feature f_j over all samples by

$$\epsilon_{\overline{j}} = \frac{1}{L_{pos}} \sum_{l=1}^{L_{pos}} \left|\sum_{k=1}^{K} a_{l,k} b_{k,j}\right| . \tag{19}$$

Finally, the generative error $\epsilon_{\overline{j}}$ is mapped into the interval $[0, 1]$. In particular, the normalized generative error e_j^g can be obtained by

$$e_j^g = g(\epsilon_{\overline{j}}) = \begin{cases} 1 & \epsilon_{\overline{j}} > \theta \\ 0 & \text{otherwise} \end{cases} , \tag{20}$$

where θ is estimated from the expected reconstruction error over all training samples.

Modified boosting error-function

The overall error e_j is estimated as a weighted sum over the discriminative error e_j^d and the generative error e_j^g:

$$e_j = \beta e_j^d + (1 - \beta) e_j^g, \tag{21}$$

where $\beta \in [0, 1]$. The main idea now is to use boosting not on the standard error-function (6) but on this new combined error (21) that incorporates both, generative and the discriminative information. Since we finally train a discriminative classifier we can define the error function for a sample$(\mathbf{x}, y)$ by $e = h(\mathbf{x})y$. Thus, for the feature f_j we get the weak classifier

$$h_j^{weak}(\mathbf{x}) = \beta h_j^d(\mathbf{x}) + (1 - \beta) e_j^g y. \tag{22}$$

When substituting the error defined in (21) (or the definition of the weak classifier (22)) into (5) we finally get that the combined approach minimizes

$$\begin{aligned} &\sum_{\mathbf{x} \in \mathcal{X}} p(\mathbf{x}) \exp\left(-\alpha(\beta h_j^d(\mathbf{x}) + (1-\beta) e_j^g)\right) = \\ &= \sum_{\mathbf{x} \in \mathcal{X}} \left(p(\mathbf{x}) \exp\left(-\alpha(\beta h_j^d(\mathbf{x}))\right) \cdot \exp\left(-\alpha(1-\beta) e_j^g\right)\right) = \\ &= \underbrace{\exp\left(-\alpha(1-\beta) e_j^g\right)}_{\text{generative prior}} \cdot \underbrace{\sum_{\mathbf{x} \in \mathcal{X}} p(\mathbf{x}) \exp\left(-\alpha \beta h_j^d(\mathbf{x})\right)}_{\text{discriminative information}} . \end{aligned} \tag{23}$$

We can interpret this error function as follows: a generative prior (calculated with respect to the weighted positive samples) influences the discriminative error in a multiplicative way, where the parameter β controls the influence. For $\beta = 1$ the generative prior vanishes and we obtain the original boosting error function; for $\beta = 0$ only the generative prior is considered but no discriminative information is included. In fact, by using the prior we introduce robustness to discriminant learning and enable *robust feature selection* using boosting.

Please note, PCA is required only during training. Once training is finished the boosted classifier can be used (*e.g.*, for object detection [28]) on its own and is therefore as efficient as any other boosting approach.

4. Experiments

For our experiments we mainly used the ATT database (former ORL database) [21] and the UIUC Image Database for Car Detection [1]. The databases were split into a training and a test set. In particular, we used 60% of the images for training and 40% for testing. The negative samples were generated randomly from images that do not contain the objects of interest. In the training we build a discriminative classifier containing 20 features; for the generative model five eigenimages were used.

The main contribution of this paper is to introduce robustness to boosting based learning. Thus, we would like to emphasize that the goal of the experiments is to show the benefits of the presented approach (*i.e.*, robust discriminative learning and reconstructing from discriminative features).

4.1. Reconstructive power

First, we show the reconstructive power and the robustness of the Eigenboost method. Thus, we trained a face model from the ATT database by boosting and by the new Eigenboost method and evaluated the obtained models on test images (Figure 4(a)). The first two images are faces from the ATT database where we added an occlusion noise; the others are non-face images from the COIL data base [18].

From Figure 4(b) it can be seen that the Haar-like basis function obtained by standard boosting can not be used to

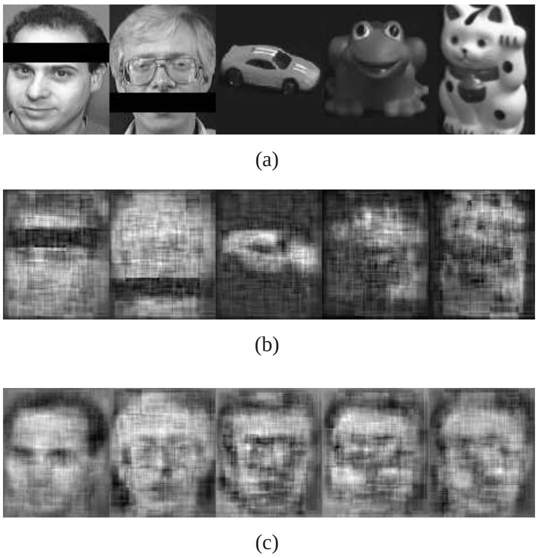

Figure 4. Reconstructive power of Eigenboosting: (a) original test images; (b) reconstructions using learned Haar-like basis functions; (c) reconstructions obtained by Eigenboosting method.

reconstruct the original image. The faces are reconstructed well, but due to the high dimensionality of the (learned) feature space "nearly everything" can be reconstructed. In fact, we can reconstruct the COIL objects even though we have learned faces before. Moreover, the reconstruction is not robust. In contrast, when using the ideas of B-PCA (approximate the eigenimages by binary basis functions) we obtain a more suitable representation. As can be seen from Figure 4(c) the faces are reconstructed correctly (even the occluded pixels) while the reconstruction for the non-faces fails completely.

4.2. ATT Face Database

Next, we trained different classifiers by our proposed method from clean data and evaluated these classifiers an clean data. From the receiver operator characteristic (ROC) curves in Figure 5 it can be seen that including more and more discriminative information (*i.e.*, increasing the parameter β) increases the recognition rate.

This effect can also be seen from Figure 7 where we evaluated the the influence of the parameter β by analyzing the equal error rate point.

When training on noisy images (we included a small constant occlusion in every positive sample) the discriminative method focuses on this local noise. This can be seen in Figure 8(a) where we show the features that were selected by using discriminant information only. Most of these features represent the occlusion that is present in all training images. Moreover, the weights for these features are very high. In contrast, Figure 8(c) shows the features that were selected by using only generative information. It can be

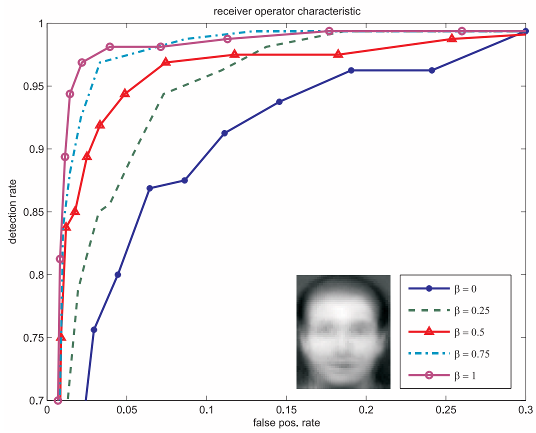

Figure 5. ATT database: ROC curves for discriminative classifiers that were trained on clean data.

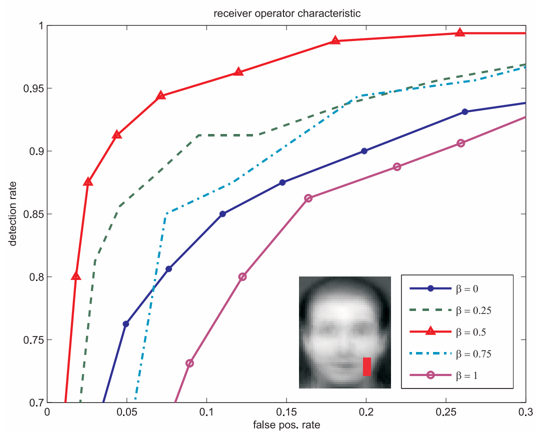

Figure 6. ATT database: ROC curves for discriminative classifiers that were trained on noisy data (small local occlusion noise was added).

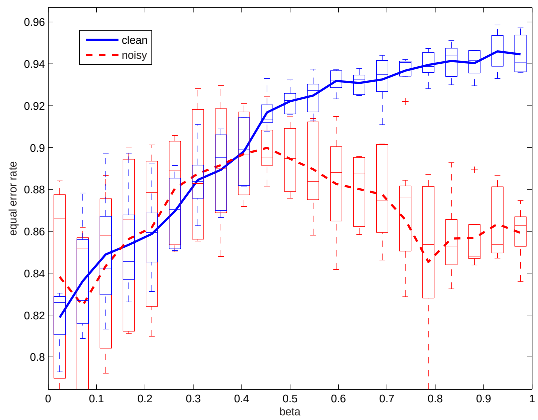

Figure 7. Varying parameter β: (generative information) $0 \leq \beta \leq 1$ (discriminative information). Increasing β increases the performance for clean data; for noisy data the best results are obtained for $\beta \approx 0.5$

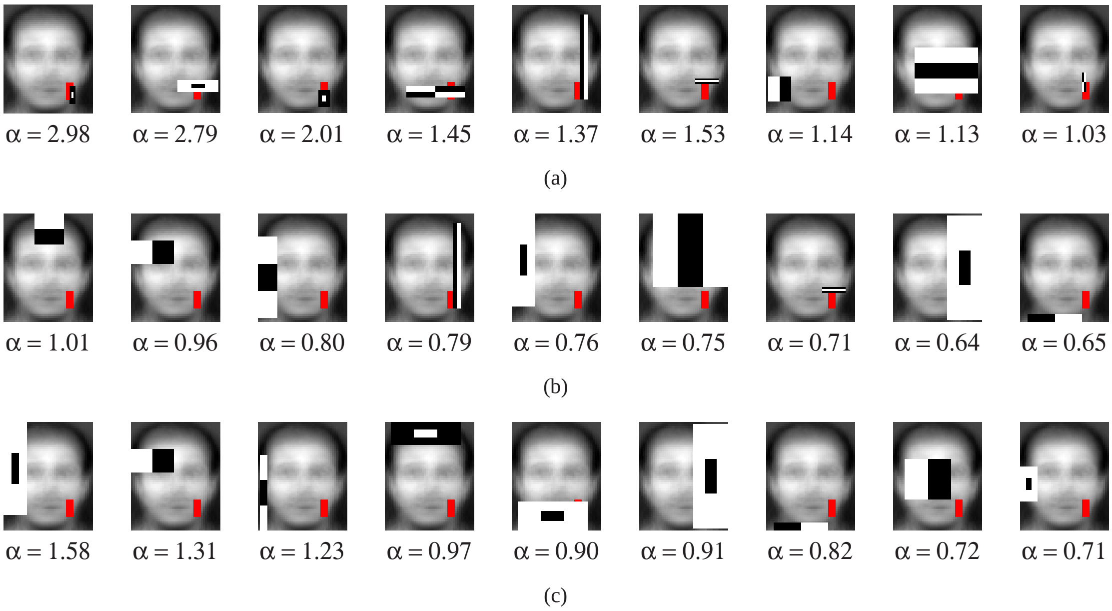

Figure 8. Chosen features with corresponding weights α: (a) classifier trained on noisy data with $\beta = 1$: only discriminative information is used (“standard boosting”); (b) classifier trained on noisy data with $\beta = 0.5$: combining generative and discriminative information; (c) classifier trained on noisy data with $\beta = 0$: only generative information is used. The artificial occlusion noise is marked as a red rectangle.

seen that these features mostly represent the global structure of the original image. In fact, the added occlusion is not modeled. When discriminative and generative information are combined we get the features shown in Figure 8(b). There are still some features left that represent the occlusion noise but compared to the pure discriminant case the weights are much smaller. Thus, the importance of these features was reduced by adding the generative information. This can be seen in Figure 6 and Figure 7. From the ROC curves in Figure 6 it is clear that the performance of the system is improved when using the combined error which captures both, the generative and discriminative information. Figure 7 shows that the performance is increased up to a certain amount of discriminative information ($\beta \approx 0.5$). But by adding further (irrelevant) discriminative information the performance is decreased.

4.3. UIUC Car Database

For the UIUC Car Database we have performed the same experiments as previously described for the ATT face database. As we are mostly interested in robust discriminant learning in Figure 9 we finally show the ROC curves obtained when adding occlusion noise to the training samples. It can be seen that the combined classifiers ($0.25 \leq \beta \leq 0.75$) outperform the extremal cases ($\beta = 0$ and $\beta = 1$).

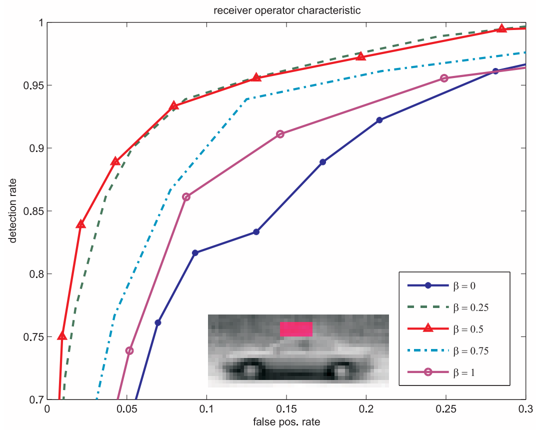

Figure 9. UIUC Car database: ROC curves for discriminative classifiers that were trained on noisy data (small local occlusion noise was added).

5. Conclusion

In this paper we presented a new visual learning algorithm that combines a discriminative and a generative model. To get a common representation for both models we expressed the learning process at the feature level. Thus, we need features that are discriminative and have reconstructive abilities at the same time. In particular, we use Haar-like features but any other feature type that fulfills both criteria (*e.g.*, Gabor-features) may be applied. For the final discriminative classifier trained by boosting we deduced a modified error-function that minimizes (23) where the generative information is included as a multiplicative prior. As we use robust PCA (the eigenimages are reconstructed from Haar-like binary basis-functions) as generative representation we can introduce robustness to boosting based learning. In particular, the method can be interpreted as mechanism for *robust feature selection* using boosting. In the experiments we demonstrated that when learning form noisy data (*i.e.*, occlusion noise) our Eigenboosting method outperforms a pure discriminative classifier. One of our next steps will be to include different feature types into our system. Moreover, as there exist on-line variants of both, boosting and PCA, an incremental extension of the method is straight forward.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, 2004.

[2] S. Avidan. Spatialboost: Adding spatial reasoning to adaboost. In *Proc. European Conf. on Computer Vision*, pages 386–396, 2006.

[3] P. N. Belhumeur, J. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[4] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *Proc. European Conf. on Computer Vision*, pages 329–342, 1996.

[5] F. de la Torre and M. J. Black. Robust principal component analysis for computer vision. In *Proc. IEEE Intern. Conf. on Computer Vision*, volume I, pages 362–369, 2001.

[6] S. Fidler, D. Skočaj, and A. Leonardis. Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(3):337–350, 2006.

[7] Y. Freund and R. E. Shapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.

[8] T. Hertz, A. B. Hillel, and D. Weinshall. Learning a kernel function for classification with small training samples. In *Proc. Intern. Conf. on Machine Learning*, pages 401–408, 2006.

[9] A. D. Holub, M. Welling, and P. Perona. Combining generative models and fisher kernels for object recognition. In *Proc. IEEE Intern. Conf. on Computer Vision*, volume I, pages 136–143, 2005.

[10] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

[11] I. T. Jolliffe. *Principal Component Analysis*. Springer, 2002.

[12] J. A. Lasserre, C. M. Bishop, and T. P. Minka. Principled hybrids of generative and discriminative models. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 87–94, 2006.

[13] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[14] A. Leonardis and H. Bischof. Robust recognition using eigenimages. *Computer Vision and Image Understanding*, 78(1):99–118, 2000.

[15] Y. Li, L. G. Shapiro, and J. A. Bilmes. A generative/discriminative learning algorithm for image classifications. In *Proc. IEEE Intern. Conf. on Computer Vision*, volume II, pages 1605–1612, 2005.

[16] R.-S. Lin, D. Ross, J. Lim, and M.-H. Yang. Adaptive discriminative generative model and its applications. In *Proc. Conf. on Neural Information Processing Systems*, pages 801–808, 2005.

[17] A. M. Martinez and A. C. Kak. LDA versus PCA. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[18] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-100). Technical Report CUCS-006-96, Department of Computer Science, Columbia University, 1996.

[19] L. Prevost, L. Oudot, A. Moises, C. Michel-Sendis, and M. Milgram. Hybrid generative/discriminative classifier for unconstrained character recognition. *Pattern Recognition*, 26(12):1840–1848, 2005.

[20] P. M. Roth, H. Grabner, D. Skočaj, H. Bischof, and A. Leonardis. On-line conservative learning for person detection. In *Proc. IEEE Workshop on VS-PETS*, pages 223–230, 2005.

[21] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *Proc. Workshop on Applications of Computer Vision*, pages 138–142, 1994.

[22] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[23] D. Skočaj, H. Bischof, and A. Leonardis. A robust PCA algorithm for building representations from panoramic images. In *Proc. European Conf. on Computer Vision*, volume IV, pages 761–775, 2002.

[24] F. Tang and H. Tao. Binary principal component analysis. In *Proc. British Machine Vision Conf.*, volume I, pages 377–386, 2006.

[25] H. Tao, R. Crabb, and F. Tang. Non-orthogonal binary subspace and its applications in computer vision. In *Proc. IEEE Intern. Conf. on Computer Vision*, volume I, pages 864–870, 2005.

[26] K. Tieu and P. Viola. Boosting image retrieval. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 228–235, 2000.

[27] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[28] P. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume II, pages 511–518, 2001.