Recognition Using Hybrid Classifiers

Margarita Osadchy, Daniel Keren, and Dolev Raviv

Abstract—A canonical problem in computer vision is category recognition (e.g., find all instances of human faces, cars etc., in an image). Typically, the input for training a binary classifier is a relatively small sample of positive examples, and a huge sample of negative examples, which can be very diverse, consisting of images from a large number of categories. The difficulty of the problem sharply increases with the dimension and size of the negative example set. We propose to alleviate this problem by applying a "hybrid" classifier, which replaces the negative samples by a prior, and then finds a hyperplane which separates the positive samples from this prior. The method is extended to kernel space and to an ensemble-based approach. The resulting binary classifiers achieve an identical or better classification rate than SVM, while requiring far smaller memory and lower computational complexity to train and apply.

Index Terms—Object recognition, object detection, large scale learning

1 Introduction

One of the central problems in computer vision is recognizing objects in realistic scenes. We deal with the classification problem, defined as predicting whether at least one object of a given class is present in an image. The basic recipe for this kind of problem has been 1) constructing a bag of visual words or spatial pyramids [29] of multiple features, 2) vector quantization, 3) training SVM classifiers with histogram intersection [29] or other additive kernels [36], [39], and 4) integrating classifiers using voting or MKL [11]. Recent work focused on devising new and better features and kernels (e.g., [47]), various coding strategies (e.g., [17]), etc. Most of these methods adopt a one-against-rest strategy for training SVM classifiers, in which the positive class is composed of samples from a single class and the negative class comprises samples from all remaining classes. When the number of classes is relatively small, the one-against-rest training scheme was shown to be as good as multi-class classifiers [40]. However, in real problems, the negative class—i.e., the background—is often much richer and includes all object categories except the positive class. When the number of classes is large, the one-against-all scheme faces two major problems: extremely unbalanced training sets, and high computational complexity [39].

Unbalanced sets. It is a common knowledge that when trained on unbalanced sets, the class boundary learned by SVMs can be severely skewed towards the smaller class, and it becomes very sensitive to noise [1]. Several approaches have been proposed to solve this problem (a review of previous work is provided in [1], [26]), including setting different penalties for misclassifying the positive class relative to the negative one, various weighting techniques, undersampling the majority class or oversampling the minority class, adjusting the class boundary based on the spatial distribution of the support vectors, and various combinations of the above. All these methods, however, do not address the complexity issue. Thus using weighted SVM or any other of these methods as a one-against-rest classifier for a large data set is problematic, especially when a kernel classifier is applied, since the number of support vectors linearly increases with the number of training examples [49].

High computational complexity. Kernel SVM was shown to be the most successful among one-against-rest classifiers for object recognition tasks [13], [29], [55]. However, it cannot be used in large-scale problems, because its training is slow and requires large memory. Further, its prediction rule is too expensive when the number of support vectors is large. The running time of kernel SVM is proportional to the number of its support vectors, which tends to linearly increase with the size of the training set [49]. There are several solutions to this problem, such as kernel approximations (e.g., [46], [31], [21]), locally linear SVMs [27], pruning the support vectors (e.g., [7], [9]), etc. Alas, all these methods trade accuracy for efficiency. Using an explicit mapping to the feature space, followed by linear classification [39], [36], [53] showed excellent accuracy, but is applicable only for certain types of kernels.

To summarize, solutions for the case of unbalanced sets exist, however they are computationally inefficient; there are also solutions which efficiently approximate the kernel classifier, but they are not designed for unbalanced sets. Further, adding a new category requires retraining all the one-against-rest classifiers, making the approach even more problematic.

In current visual classification problems, the negative class approaches the complement of the positive class and thus it can be viewed as a general "background class". In this work we propose classifiers that are specifically designed to separate a class from a rich background. By "background" we mean *all images* except the category to be recognized. Learning this background from samples is highly problematic. We suggest *replacing the background samples by a distribution.* The idea is straightforward—instead of

• *The authors are with the Computer Science Department, University of Haifa. E-mail: {rita, dkeren}@cs.haifa.ac.il, dolev.raviv@gmail.com.*

Manuscript received 14 June 2014; revised 3 Aug. 2015; accepted 4 Aug. 2015. Date of publication 6 Aug. 2015; date of current version 11 Mar. 2016. Recommended for acceptance by D. Ramanan.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2465910

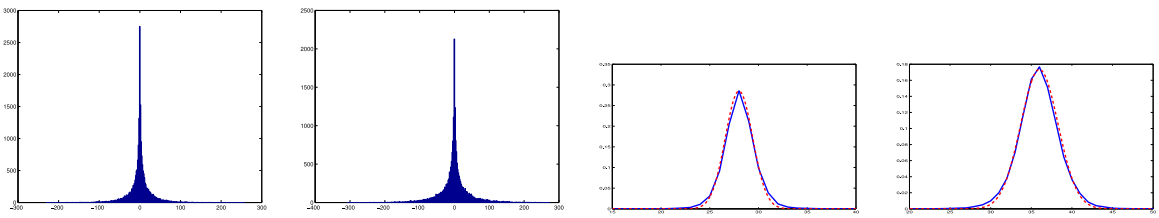


Fig. 1. Examples of 1D random projections of the background class. The two histograms on the left correspond to grey-level with 8×8 filter size (as common in work on natural image statistics). The projections are clearly non-Gaussian. The other two histograms correspond to BoW of SIFT features (the blue solid line). The projections are very close to Gaussians (dashed red line).

minimizing the *number* of background samples in the classifier's acceptance region, we minimize the *probability volume* of the background prior in the acceptance region. This formulation eliminates the problem of unbalanced training sets (since there are no negative samples) and the high complexity due to the large number of "negative" support vectors. From now on we shall refer to this type of classifiers as *hybrid*. The notion of hybrid classifier we use here characterizes the mixed input in the training phase: samples from the positive class versus probability distribution on the background.

The idea of a hybrid classifier was first introduced in [38], but the solution proposed there was restricted to grey-level images and applied a very simple prior, which is not robust to illumination variation and image deformations. Further, only a linear classifier was presented. Here we extend the basic paradigm to realistic scenes and propose the following contributions:

1. Although accurately modeling the background is difficult, we observe that in classification tasks one typically seeks to separate the values of the two categories (or in this case, a single category and the background) *after* they were projected (either linearly, as in linear SVM, or by a more complicated function, e.g., a kernel) into the real line. We show that the projection of a complicated background can be well-approximated by a simple distribution (e.g., Gaussian). Thus, we suggest that as the number of image categories in the background class increases, the method described here will become even more suitable.
2. We apply priors on robust features, such as Bag of Words constructed from densely sampled SIFT features [4], [34]. The prior assumed in [38] was based on the observation that typical images are "smooth", that is, most of their energy is concentrated in the low frequencies. Although BoW features obviously do not posses this property, we show that they can be successfully used with the hybrid classifiers, suggesting that the basic paradigm is very general and can be applied to other features and domains.
3. We develop a kernel version of the hybrid classifier, which is much more efficient than kernel SVM in both training and classification, while it enjoys an even better classification accuracy.
4. We demonstrate that a hybrid linear classifier can be used in an ensemble of linear classifiers, yielding better performance than other linear classifiers, such as linear SVM and LDA.

1.1 Modeling the Background Distribution

Compared to a single object class, the background distribution is so wide that it can be assumed to be approximately equal to the distribution of all natural images, hence we can use this distribution to model the background class (this model will therefore be applicable to *all* single classes one wishes to detect, thus drastically reducing training complexity). Modeling the distribution of natural images is, however, a challenging task. A number of energy-based models have been proposed to learn this distribution from examples (e.g., [41], [48], [54], [56]). These models attempt to find a set of linear filters in order to decompose the image into channels, as well as the corresponding energy functions. Training most of these models is very long, which is not a burden if it is computed once and then used for an application that employs a fixed prior. We are interested in determining a suitable prior on natural images and applying it to classification. In light of this we need to evaluate, during training, the probability of background images to be accepted by the classifier; this probability reflects the percentage of false positives, which the classifier seeks to minimize. Such an evaluation is performed for each choice of parameters for the candidate classifier.

Since the final step in classification consists of thresholding a scalar-valued function, we are interested in modeling the projections or outputs of scalar-valued functions applied on the space of natural images. Modeling projections of natural images has also been studied in low-level vision. The most striking difference between the functions applied to features commonly used in object recognition and the linear filters applied to grey-levels in low-level vision [54] is the form of the distribution they produce. Applying linear filters, such as derivative-like filters, wavelets etc. to natural images, represented by grey-levels, produces outputs whose distribution is highly non-Gaussian—it is peaked at zero and has heavy tails [48] (Fig. 1). We are interested in non-linear functions of grey-levels, such as Bag of Words [4], constructed from SIFT features [34]. Our experiments suggest that projections of these representations are Gaussian-like (Fig. 1). As elaborated in Section 2, this allows to efficiently approximate the distribution of the projections of the background class. Gaussian modeling of the negative set was also used in [15] and showed close to state-of-the-art detection results.

1.2 Incorporating the Prior in a Classifier

There are different ways of incorporating a prior into a model or classifier. In [32] a Bayesian approach is employed, in which the prior information about object categories is learned from previously observed models or unrelated categories and is incorporated as a probability density function

of the parameters of the generative model. Here, we consider the problem of object recognition against a general background, and the prior we use is on the background class. We don't have any prior on the object class, thus an application of the Bayesian method is not straightforward.

Using a prior on the marginal distribution, or using unlabeled images, is the essence of semi-supervised or transductive learning [2]. In these techniques, a classifier is trained on labeled samples from both classes and the prior is incorporated by the assumption that $p(x)$ influences $p(y|x)$ (where x is the data and y represents the labels). The relation between the two depends on the method. In our method we not only assume an influence of $p(x)$ on $p(y|x)$, but also use the much stronger assumption that the marginal distribution is approximately equal to the background distribution.

The Fisher kernel [18] allows incorporating prior knowledge about the data distribution into a discriminative framework. This is done by comparing the gradient of the log-likelihood of the data item with respect to the model with a given set of parameters. This is very different from our approach, as in our case we model the distribution of the background using a generative model and learn a discriminative model between the positive examples and that distribution.

Using a Gaussian approximation of the negative set for object detection was also suggested in [15], as part of an LDA model, which was trained to separate a single positive example (or a cluster of similar positive examples) from the background. Similarly to our proposed method, they learned a Gaussian model of the background only once, using images of all classes. This model was used to train exemplar LDA, achieving detection results comparable to exemplar SVM [35], but at much lower training complexity. The same model was used to whiten a HOG representation of images, in order to cluster them in more coherent clusters than those obtained by using euclidean distance. Training linear classifiers for clusters rather than for each positive example allowed to reduce detection time.

While there is some similarity between our work and [15] in the Gaussian approximation of the background, the idea of using a simple Gaussian model of the image space as a background approximation in training linear classifiers was introduces earlier, in [38]. Also, the motivation of using a prior instead of examples in our case has a probabilistic meaning: we minimize the overall probability volume of the background prior in the acceptance region. In [15], the positive examples are approximated by their mean, which is inaccurate in the presence of even moderate inner class variation. Our experiments show (see Section 4) that the proposed hybrid linear classifier consistently outperforms the model proposed in [15] when applied to clusters obtained using whitened features. Finally, in this paper, we extend the hybrid formulation to kernel, which performs substantially better than linear classifiers.

2 Hybrid Classifiers

We propose to incorporate the background prior in a hybrid classifier $f(x)$, which is trained to attain positive scores on the samples of the target class and for which $\int_H \Pr(z)dz$ is very small, where $\Pr(z)$ represents the background distribution, and H is the acceptance region of $f(x)$ (i.e., all x for which $f(x) \geq 0$). Thus the standard constraints of excluding background *samples* are replaced by a *single* constraint of excluding a large volume of background *probability*. One can see some resemblance between the hybrid classifier and one class SVMs [42], [51]. These, however, implicitly assume that the background is isotropically distributed, while here we don't make that assumption.

2.1 Linear Classifier

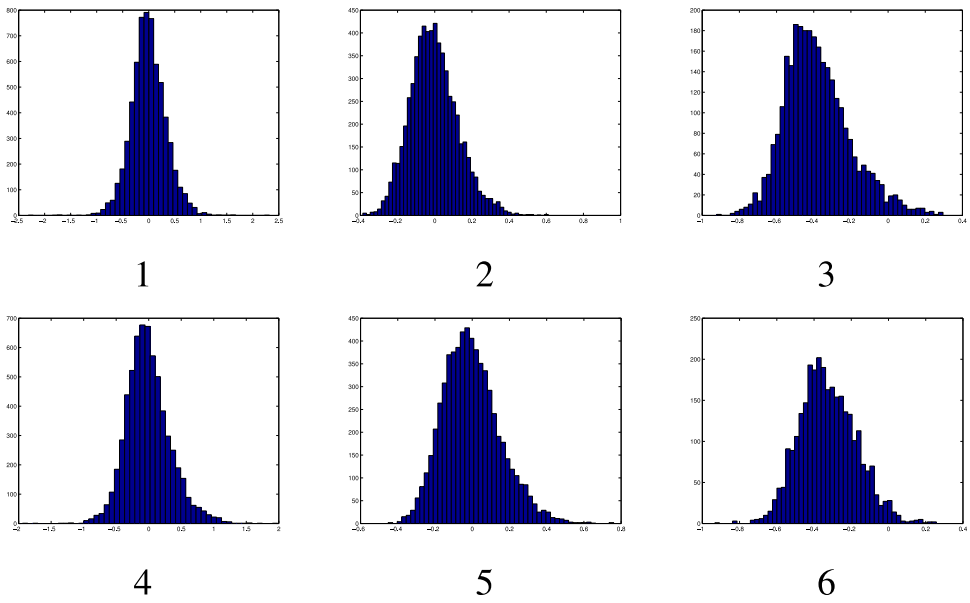
We search for a separating hyperplane $(\mathbf{w}, b)$ which yields a maximum margin between itself and the positive samples, under the constraint that the integral of the probability density of the background (natural images) in its acceptance region, $\mathcal{H} = \{\mathbf{x}|\mathbf{w} \cdot \mathbf{x} \geq b\}$, is small. We bound the probability of natural images to fall in the acceptance region: $\Pr(\mathbf{w} \cdot \mathbf{x} \geq b) < \delta$, where the constant δ is close to zero. Diaconis and Freedman [5] showed that if high-dimensional data consists of independent and identically distributed random vectors, then its projections are almost surely close to Gaussian. We empirically demonstrate that this proposition holds for one-dimensional random projections applied to two diverse sets of images: Caltech-256 [14] and Scenes-15 [29]. We used all 30,607 images of 256 categories from Caltech-256 and 3,000 images of 15 scenes from Scene-15. Images in Caltech-256 are quite diverse and objects appear in various scales and orientations; images in Scenes-15 contain many objects. Thus, these sets can serve as an approximation to the set of natural images. We used the BoW representation provided in [11] [1] for Caltech-256, and three-level pyramids of BoW [29] for Scenes-15. We tested hundreds of random projections for both sets, and all of them are well-approximated by one-dimensional Gaussians (Fig. 1, the two histograms on the right). We show in Appendix A the KL divergence of an average random projection from normal as the function of number of categories in the background class. This experiment shows that the KL divergence drops rapidly and then saturates, reaching very low values for a larger number of categories. The first two images in Table 1 show that the distribution of the projections which correspond to the learned classifiers is quite similar to the distribution of random projections, which supports the Gaussian assumption. Our experiments show that the Gaussian approximation of the projections bounds the background probability in the acceptance region of the learned classifiers in all tests conducted (see Section 4).

In order to obtain a closed-form, general expression for the distribution of the projections, we first estimate the mean and covariance matrix of the high-dimensional distribution, denoted $\bar{\mathbf{x}}$ and $\mathbf{\Sigma}_x$ respectively. Then, the projection defined by taking a scalar projection with a vector $\mathbf{w}$ is a random variable with mean $\mathbf{w}^T\bar{\mathbf{x}}$ and variance $\mathbf{w}^T\mathbf{\Sigma}_x\mathbf{w}$. Following the previous discussion, we approximate this variable by a Gaussian; thus the probability of a background image to be accepted by the classifier is

$$\Pr(\mathbf{w} \cdot \mathbf{x} \geq b) = \frac{1}{2}\left[1 - \mathrm{erf}\left(\frac{1}{\sqrt{2}} \frac{b - \mathbf{w}^T\bar{\mathbf{x}}}{\sqrt{\mathbf{w}^T\mathbf{\Sigma}_x\mathbf{w}}}\right)\right]. \quad (1)$$

1. http://www.vision.ee.ethz.ch/~pgehler/projects/iccv09/index.html

TABLE 1
Examples of 1D Projections of Test Images on Separating Hyperplanes Corresponding to Different Hybrid Classifiers: The First Four Distributions Correspond to Classifiers Trained on Different Categories from Caltech-256, the First Two—Linear Classifier, the Third and Fourth—SPM Kernel Classifiers; the Last Two Correspond to SPM Kernel Trained on Two Different Categories from Scene-15



For the approximation to be valid, the real (empirical) probability of a background image to lie in a certain half-space should be close to the one derived from the prior (or the "theoretical probability"). The empirical probability can be estimated by testing many randomly chosen background images, while the theoretical probability can be computed (as in Eq. (1)). We tested the similarity between the two probabilities on Caltech-256 and Scenes-15; results are presented in Fig. 2. We used disjoint sets to estimate the mean and covariance of $\mathbf{x}$ and the probability that a natural image falls in the "positive" (acceptance) half-space. We randomly chose $\mathbf{w}$, constraining its norm to be 1, and a value for b in the range [-0.5, 0,5]. For each choice of $(\mathbf{w}, b)$ we computed the expression in Eq. (1) and used it as the x-coordinate of a point in the scatter plot in Fig. 2. The y-coordinate represents the empirical probability, and it is computed as the actual percentage of the images that fall in the positive half-space. The scatter plot supports the validity of the proposed approximation. A similar relation has been shown in [38] for the class of natural images represented in the frequency domain. This suggests that such relations hold for different features and data sets.

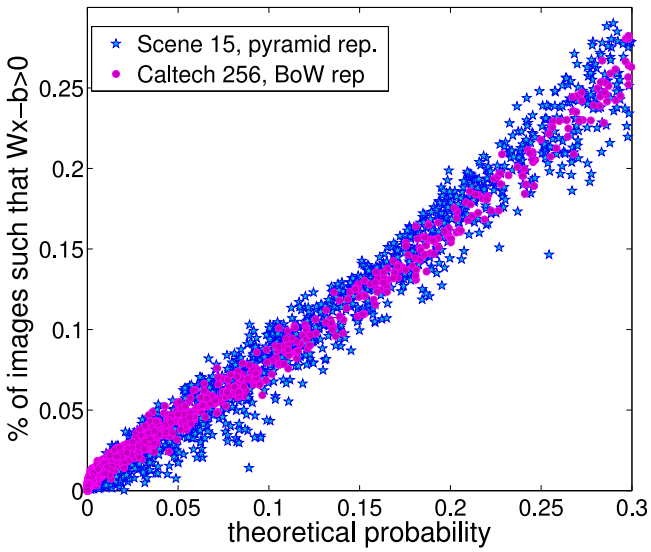


Fig. 2. Relation between the percentage of natural images in the acceptance region (a half-space) and the Gaussian approximation in Eq. (1), tested on the Caltech-256 and Scenes-15 data sets. The plot is zoomed on the [0, 0.3] interval of the probability, which is more relevant to our purpose, as the probability volume of the background in the acceptance region should be small.

Based on the above observations, the constraint on the probability of background misclassification is given by:

$$\Pr(\mathbf{w}\cdot\mathbf{x}\geq b)=\frac{1}{2}\left[1-\mathrm{erf}\left(\frac{1}{\sqrt{2}}\frac{b-\mathbf{w}^T\bar{\mathbf{x}}}{\sqrt{\mathbf{w}^T\Sigma_x\mathbf{w}}}\right)\right]\leq\delta. \quad (2)$$

Since we seek to minimize $\Pr(\mathbf{w}\cdot\mathbf{x}\geq b)$, we assume that $\delta<1/2$, and thus $\gamma\triangleq\sqrt{2}\mathrm{erf}^{-1}(1-2\delta)>0$. By formulating the constraint in Eq. (2) in terms of γ, and rearranging, we obtain a convex constraint:

$$\gamma\sqrt{\mathbf{w}^T\Sigma_x\mathbf{w}}+\mathbf{w}^T\bar{\mathbf{x}}-b\leq 0. \quad (3)$$

A more general argument, which does not require the Gaussian approximation assumption, can be applied to justify the constraint in Eq. (3). To show this we apply a result from [28], which states that for a half space $S=\{\mathbf{w}\cdot\mathbf{y}\geq b\}$, and all distributions y with expectation $\bar{\mathbf{y}}$ and covariance matrix Σ_y:

$$\sup_{\mathbf{y}\sim(\bar{\mathbf{y}},\Sigma_y)}\Pr(\mathbf{w}\cdot\mathbf{y}\geq b)=\frac{1}{1+d^2},d^2=\frac{(b-\mathbf{w}^t\bar{\mathbf{y}})^2}{\mathbf{w}^t\Sigma_y\mathbf{w}}. \quad (4)$$

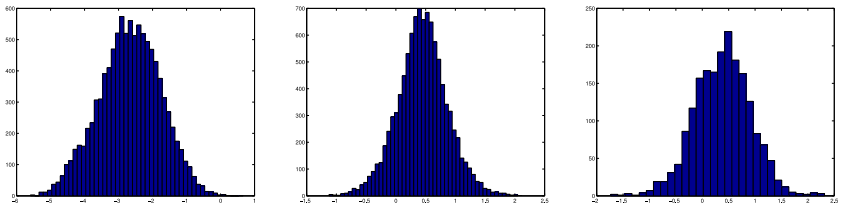


Fig. 3. Histograms of values of the histogram intersection (left), χ^2 (middle), and SPM (left) kernels with randomly selected parameters, applied to many background samples represented by a BoW of SIFT features on Caltech-256.

Now, instead of constraining the probability, we constrain its supremum over all distributions for $\mathbf{x}$ having mean $\bar{\mathbf{x}}$ and covariance Σ_x. Using Eq. (4) we obtain:

$$\sqrt{\frac{1-\delta}{\delta}}\sqrt{\mathbf{w}^T\Sigma_x\mathbf{w}} + \mathbf{w}^T\bar{\mathbf{x}} - b \leq 0 \tag{5}$$

which turns out to be the same as Eq. (3) with $\gamma = \sqrt{\frac{1-\delta}{\delta}}$.

We now define the linear hybrid classifier as the solution to the following optimization problem: given a set $\{\mathbf{x_i}\}_{i=1}^n$ of positive examples, minimize $\|\mathbf{w}\|^2$, subject to $\mathbf{w}\cdot\mathbf{x}_i - b \geq 1$ $(i = 1,\ldots,n)$ and the probability constraint in Eq. (3). This formulation resembles the usual SVM algorithm, but with the many constraints on the negative examples replaced by *one* constraint on the probability. Note that the background slackness is controlled by the parameter δ. Adding slacks to the positive samples could be done similar to SVM:

$$\begin{aligned} \underset{\mathbf{w},b,\xi}{\text{minimize}} \quad & \|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & \mathbf{w}\cdot\mathbf{x}_i - b \geq 1 - \xi_i, \\ & \xi_i \geq 0, \\ & \gamma\sqrt{\mathbf{w}^T\Sigma_x\mathbf{w}} + \mathbf{w}^T\bar{\mathbf{x}} - b \leq 0. \end{aligned} \tag{6}$$

γ is a parameter that controls the probability volume of the negative class in the positive side of the classifier and can be found using cross-validation.

2.2 Kernel Classifier

We use a standard kernel decision function:

$$f(\mathbf{x}) = \mathbf{sign}\left(\sum_{i=1}^{l}\alpha_i K(\mathbf{s}_i,\mathbf{x}) - b\right),$$

where α_i, $\mathbf{s}_i$, and b are the model parameters. The $\mathbf{s}_i$'s are chosen from a set of unlabeled training examples, as described later.

To compute the probability

$$\Pr\left(\sum_{i=1}^{l}\alpha_i K(\mathbf{s}_i,\mathbf{x}) \geq b\right)$$

on the background class, we define a random variable in kernel space, $\mathbf{z} = [z_1,\ldots,z_l]^t$, where $z_i \triangleq K(\mathbf{s}_i,\mathbf{x})$ $(i = 1,\ldots,l)$, ($\mathbf{x}$ is a random variable in input space, representing the background). Then, we write the probability constraint as

$$\Pr\left(\sum_{i=1}^{l}\alpha_i z_i \geq b\right) \leq \delta. \tag{7}$$

This constraint has the same form as in our linear classifier. Similarly, we can apply the Gaussian approximation and obtain the same expression as in Eq. (3), with the only difference that $\mathbf{x}$ is replaced by $\mathbf{z}$, which is obtained by applying a non-linear function $K(\mathbf{s}_i,\mathbf{x})$. Thus the constraint is

$$\gamma\sqrt{\alpha^t\Sigma_z\alpha} + \alpha^t\mu_z - b \leq 0, \tag{8}$$

where μ_z is the mean and Σ_z the covariance matrix of $\mathbf{z}$.

Next, we check the validity of the proposed approximation for several kernels commonly used in object recognition: the histogram intersection, χ^2, and the spatial pyramid match (SPM) [29] kernels. Fig. 3 shows examples of outputs of these kernels. To create random projections in kernel space we randomly chose 1,000 samples as $\mathbf{s}_i, i = 1\ldots1{,}000$, and 1,000 scalars as $\alpha_i, i = 1\ldots1{,}000$, and evaluated, using a diverse collection of images $\mathbf{x}$, the value of $\sum\alpha_i K(\mathbf{s}_i,\mathbf{x})$ (where $K()$ are the above-mentioned kernels). Table 1 (images 3-6) depicts examples of projections on learned classifiers. These distributions do not differ much from the random projections, which supports the Gaussian assumption.

The result from [28] can be applied to the kernel classifier as well (here we consider the supremum over all distributions for $\mathbf{z}$ having mean μ_z and covariance Σ_z), which leads to the following constraint:

$$\sqrt{\frac{1-\delta}{\delta}}\sqrt{\alpha^t\Sigma_z\alpha} + \alpha^t\mu_z - b \leq 0 \tag{9}$$

which has the same form as Eq. (8).

We now formulate the following convex optimization problem to learn the hybrid kernel classifiers. Given a set $\{\mathbf{x}_j\}_{j=1}^n$ of positive examples:

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & \sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j K(\mathbf{s}_i,\mathbf{s}_j) + C\sum_{k=1}^{n}\xi_k \\ \text{subject to} \quad & \sum_{i=1}^{l}\alpha_i K(\mathbf{s}_i,\mathbf{x}_j) - b \geq 1 - \xi_j \quad \forall j = 1,\ldots,n; \\ & \gamma\sqrt{\alpha^t\Sigma_z\alpha} + \alpha^t\mu_z - b \leq 0. \end{aligned} \tag{10}$$

Here we use a standard kernel regularizer as an objective function (Eq. (10)).

We next address the question of choosing the $\mathbf{s}_i$'s which define the classifier $f(\mathbf{x}) = \sum_i\alpha_i K(\mathbf{s}_i,\mathbf{x})$. For a standard kernel SVM, the s_i's are automatically chosen from the samples; we, however, do not use negative samples, so the $\mathbf{s}_i$ have to be determined in advance. The basic idea is to find a family $\{\mathbf{s}_i\}$ such that the span of $K(\mathbf{s}_i,\mathbf{x})$ approximates all the functions $K(\mathbf{s},\mathbf{x})$, where s ranges over the sample space.

A similar problem has been addressed in [46]: find a subset of indices $I = \{i_1, \ldots, i_m\} \subset [t]$ (where t is the size of the full kernel matrix K) such that $\tilde{K}_i = \sum_{j=1}^{m} K_{i_j} T_{ji}$, where K_i are the columns of the kernel matrix, T is an $m \times t$ matrix containing the expansion coefficients for an approximation of the columns of K. I and T are chosen to minimize the Frobenius norm $\|\tilde{K} - K\|_{Frob}$. A greedy, probabilistic algorithm from [46] chooses I in $O(mt)$ time complexity per index. Here we define K to be the kernel matrix of the unlabeled training samples and $\mathbf{s}_i = \tilde{K}_i$; then we apply the algorithm from [46] for finding $\tilde{K}_i$.

In our formulation, $\mathbf{s}_i$'s represent the background, and are chosen prior to the training of the classifiers. Thus we will refer to this set as the "common" $\mathbf{s}_i$'s. To learn the classifier for a specific class, we add its positive examples to the common $\mathbf{s}_i$'s and run the optimization in Eq. (10). This is much faster than training kernel SVM, as far less parameters need to be optimized over. Our experiments also suggest that the number of common $\mathbf{s}_i$'s required to represent a rich background is small, and does not increases as the number of background categories increases.

2.3 Complexity

Standard SVM training requires $O(N^3)$ time and $O(N^2)$ space complexities, where N is the training set size. Gradient based methods can train linear SVM in $O(N)$ [20]. The SGD solver PEGASOS [43] does not even depend on the size of the training set, and was used to learn linear classifiers in large-scale vision applications [39], [50]. However, linear SVMs have been reported to be inferior to non-linear SVMs on BoW [39]. Directly applying non-linear SVMs is impractical for very large image collections. Kernel approximation methods enjoy a lower computational complexity than kernel SVM [22]. While these methods consider balanced problems, the main computational burden in one-against-all training is due to the negative ("rest") class. The hybrid classifiers proposed here significantly reduce the amount of computations, since they replace the constraints on the negative examples with a single probability constraint and do not use negative examples in the training stage.

The training of hybrid classifiers consists of two steps. The first is performed only once and includes the selection of $\mathbf{s}_i$'s (only for the kernel classifier) and the estimation of the background covariance matrix. The second step is the actual training of the classifier, which is done per object class and thus repeated a number of times equal to the number of classes one wishes to recognize. Hereafter we denote by n the number of positive examples per category, by m the number of common $\mathbf{s}_i$'s, by p the number of unlabeled samples for selecting $\mathbf{s}_i$'s, and by C the number of categories comprising the background class in the one-against-rest training phase. The complexities of the proposed classifiers are studied next.

2.3.1 Linear

a) Estimation of the background covariance matrix. Even though an accurate estimation of the covariance matrix of a high-dimensional random variable requires many samples, here we are only interested in its 1D projections, thus an approximation of the covariance matrix suffices. We observed that the number of background samples required to derive this approximation is relatively small: in the Caltech 256 experiments, increasing the number of samples beyond five per category had a negligible effect on the projection's parameters as well as on the performance. Note that the background covariance matrix has to be estimated *only once*, and then it is applied for training classifiers for *all* classes.

b) Training a classifier per category. Our optimization has only $n + 1$ constraints (n positive examples and one probability constraint), while the number of constraints in one-against-all SVM training is nC. Another important advantage is that we do not need to keep a huge number of negative examples in memory, which allows using off-the-shelf solvers for convex optimization, even for a large scale classification problems. The classification process is the same as for linear SVM.

2.3.2 Kernel

a) Choosing $\mathbf{s}_i$'s. To find the common $\mathbf{s}_i$'s we use the algorithm from [46], which runs in $O(mp)$ per vector, thus the entire process runs in $O(m^2 p)$. The selection is performed only once, and even for a very rich background, the size of the basis m is small (about 200). (Section 4.1.4).

b) Estimation of the background covariance matrix. The size of the covariance matrix is $(m+n)^2$, of which the block of size $m \times m$ is identical for all classes (since the common $\mathbf{s}_i$'s do not depend on the class one wishes to recognize), and the block including the class-specific $\mathbf{s}_i$ of size $n \times (m+n)$, which must be estimated for each class. Typically both n and m are quite small (see Section 4.1.4), thus estimating the covariance matrix is not a burden.

c) Training a category classifier. We optimize over $m + n$ parameters, compared to nC in kernel SVM trained in one-against-rest manner. Similarly to the linear case, our formulation has $n + 1$ constraints. The space complexity is $O(m+n)^2$, compared to $O(C^2 n^2)$ in SVM.

d) Classification using kernel classifier. In kernel SVM, the number of kernel evaluations required to classify an input image is equal to the number of support vectors, which is linear in the size of the training set [49]. The number of kernel evaluations when applying the kernel hybrid classifier is $(m + n)$, which is typically small and independent of the number of categories one wished to recognize (Section 4 provides an empirical study, showing that beyond a small number of categories the number of $\mathbf{s}_i$'s does not increase).

3 Using Hybrid Classifiers in an Ensemble

Several recent papers [15], [24], [33], [35] proposed using an ensemble of linear hyperplanes for object recognition. In ensemble methods, the key property of the individual classifiers from which an ensemble is formed is *diversity* [6]. Diversity is achieved by partitioning the input space either spatially or by using different types of features. A more formal definition of diversity is statistical independence. Independent linear classifiers were first introduced in [24], and it was shown empirically that the number of false positives produced by a cascade of these classifiers decreases

exponentially in the number of classifiers. A similar idea was used in [33] for constructing a minimum correlation ensemble of SVM classifiers, providing improvement over boosting SVM classifiers trained on different subsets of the data.

In [15], [35] diversity was achieved by partitioning the positive examples into small subsets, while keeping the negative class intact: [35] built an ensemble of linear SVMs, each of which is trained with a single positive example; [15] used an ensemble of LDA classifiers, trained for clusters of examples from the same class. Next, we briefly discuss these methods and show the advantages of using the hybrid linear classifier instead of SVM or LDA.

3.1 Minimum Correlation Ensemble

In the minimum correlation ensemble [33], the basic classifiers are computed sequentially, using an SVM-like formulation with an additional term which measures the correlation of the current classifier with the previous ones. Then, these classifiers are applied to a validation set and their outputs are stacked, forming a new feature vector; these vectors are used to train a strong classifier using GentleBoost [10] over decision stumps. Thus in the first stage, the directions of the classifiers are defined by the SVMs satisfying the minimum correlation requirement, and in the second stage, optimal thresholds for each SVM are computed. The minimum correlation requirement in [33] is formulated in terms of training examples:

$$r = \sum_{j\in\{p,n\}} \left(\frac{\langle \mathbf{w}_i^T(X_j - \bar{X}_j), \mathbf{w}_k^T(X_j - \bar{X}_j)\rangle}{\|\mathbf{w}_i^T(X_j - \bar{X}_j)\|\|\mathbf{w}_k^T(X_j - \bar{X}_j)\|} \right)^2, \quad (11)$$

where $\mathbf{w}_i$ are the previously computed classifiers ($i = 1,\ldots,k-1$), $\mathbf{w}_k$ is the current classifier, and X_j are matrices consisting of samples, where p and n refer to positive and negative examples correspondingly.

Adding the minimum correlation term (Eq. (11)) makes the loss function non-convex. To solve this problem, it was suggested in [33] to remove $\|\mathbf{w}_k^T(X_j - \bar{X}_j)\|$ from the expression in Eq. (11), and add the remaining convex part to the SVM objective function with a multiplicative parameter. In order to maintain the desired level of correlation, the multiplicative parameter was increased when the resulting classifier's correlation with the previous ones was higher than a certain threshold (which was set as a parameter). A generalization bound on the individual classifiers was provided, and the resulting classifier was demonstrated to work well on several data sets, although training SVM with the minimal correlation requirement a number of times (when the requirement is not satisfied) renders the training slow.

Eq. (11) can also be written in terms of the covariance matrix of the feature space. Let X represent a matrix of examples sampled from the feature space (shifted to obtain a zero average). The minimum correlation requirement for the classifiers trained on these samples is:

$$r = \frac{(\mathbf{w}_i^T X X^T \mathbf{w}_k)^2}{(\mathbf{w}_i^T X X^T \mathbf{w}_i)(\mathbf{w}_k^T X X^T \mathbf{w}_k)} = \frac{(\mathbf{w}_i^T \Sigma \mathbf{w}_k)^2}{(\mathbf{w}_i^T \Sigma \mathbf{w}_i)(\mathbf{w}_k^T \Sigma \mathbf{w}_k)} < \lambda, \quad (12)$$

where Σ is a covariance matrix approximation of the feature space. Let us denote the convex part of the constraint as

$$\mathbf{v}_j = \frac{\mathbf{w}_i^T \Sigma}{\sqrt{\mathbf{w}_i^T \Sigma \mathbf{w}_i}}.$$

Next, we suggest to replace the linear SVM with a linear hybrid classifier and incorporate the minimum correlation constraint from Eq. (12) in the same way it was derived in [33]. The training starts by constructing a linear hybrid classifier as the first classifier in the ensemble, using the optimization in Eq. (6). The subsequent classifiers are trained by running the following optimization with $\gamma' = 10\gamma$ (which means that we allow a larger probabilistic volume of the background on the positive side of the hyperplane than for the first classifier):

$$\begin{aligned} \underset{\mathbf{w},b,\xi}{\text{minimize}} \quad & \|\mathbf{w}_k\|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{j=1}^{k-1}\eta_j(\mathbf{v}_j\mathbf{w}_k)^2 \\ \text{subject to} \quad & \mathbf{w}_k\cdot\mathbf{x}_i - b \geq 1 - \xi_i, \\ & \xi_i \geq 0, \\ & \gamma'\sqrt{\mathbf{w}_k^T\Sigma_x\mathbf{w}} + \mathbf{w}_k^T\bar{\mathbf{x}} - b \leq 0, \end{aligned} \quad (13)$$

where η_j is a parameter controlling the correlation term. After finding $\mathbf{w}_k$, we check if it satisfies the minimal correlation constraint in Eq. (12). If not, we return to the optimization in Eq. (13) with $\eta_j = 2\eta_j$.

This procedure is essentially the same as in [33], but training the hybrid classifier is much faster than training SVM, and the independence constraint is simpler.

3.2 Ensemble of Exemplar Models

It was shown in [15] that using an LDA classifier instead of linear SVM as an exemplar model [35] significantly reduces training time, since constructing an LDA classifier that separates a single positive example from the background can be solved in simple closed form. In LDA-based classification it is assumed that all classes follow the Gaussian distribution. Using the same covariance matrix for all classes is a common regularization technique, which was adopted in [15]; specifically, they learned the covariance matrix of the feature space once using images from all classes and used the same covariance matrix in training all LDA classifiers.

Let μ and Σ denote the mean and covariance matrix of the feature space. The LDA classifier for a single positive $\mathbf{x}$ has the following form:

$$\mathbf{w} = \Sigma^{-1}(\mathbf{x} - \mu). \quad (14)$$

Next we show that the formulation in Eq. (14) can be obtained using the same probabilistic approach we used to derive the hybrid classifier. A linear classifier that passes through a point $\mathbf{x_p}$ and minimizes the probability of the background in its positive hyper-plane is defined as follows:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}\left[1 - \text{erf}\left(\frac{1}{\sqrt{2}}\frac{b - \mathbf{w}^T\mu}{\sqrt{\mathbf{w}^T\Sigma_x\mathbf{w}}}\right)\right] \\ \text{subject to} \quad & \mathbf{w}^T\mathbf{x_p} - b = 0. \end{aligned}$$

This problem is equivalent to the following maximization problem

$$\begin{aligned} \max_{\mathbf{w}} \quad & \frac{b - \mathbf{w}^T \mu}{\sqrt{\mathbf{w}^T \Sigma \mathbf{w}}} \\ \text{subject to} \quad & \mathbf{w}^T \mathbf{x_p} = b \end{aligned}$$

which can be written as

$$\max_{\mathbf{w}} \frac{\mathbf{w}^T (\mathbf{x_p} - \mu)}{\sqrt{\mathbf{w}^T \Sigma \mathbf{w}}}. \tag{15}$$

It is easy to show that the solution of the problem in Eq. (15) is given by

$$\mathbf{w} \propto \Sigma^{-1} (\mathbf{x_p} - \mu),$$

which is the same as Eq. (14) (since $\mathbf{w}$ defines the direction of the discriminative projection and thus its norm is not important).

To reduce the running time of the exemplar model, it was suggested in [15] to train an exemplar model for a cluster of similar positive examples, instead of a single example. It was also assumed [15] that the positive and negative classes have different means (μ_p—the mean of the cluster and μ—the mean of the feature space), but the same covariance Σ. Under these assumptions, the LDA solution reduces to

$$\mathbf{w} \propto \Sigma^{-1} (\mu_\mathbf{p} - \mu).$$

However, these assumptions are clearly limiting. We suggest to use a hybrid linear classifier as an exemplar model, trained on a cluster of positive examples. We expect it to perform better than LDA, as it is more expressive (the separation boundary depends on *all* positive examples and not only on their mean) and it does not require estimating the covariance matrix of the positive examples. Our experiments show that hybrid classifiers provide better exemplar model than LDA.

4 Experiments

The hybrid classifier is a binary classifier. Its most natural application is object detection, in which the background class is not limited to a predefined number of categories and can be very diverse. Thus the model for a general background, which we handle, is very useful in object detection. Using a binary classifier in one-against-all or other multi-class problems is another potential application of binary classifiers. Unfortunately, it is a heuristic which results in poor performance when the number of classes is large. This was apparent in early experiments on ImageNet. Since SVM did not perform well in one-against-all multi-class setting when the number of classes is large, we do not expect impressive results from the hybrid classifier either. Thus our experiments include only binary tests on multi-class data sets and a full object detection system on PASCAL VOC 2007 [8].

We first show that the linear and kernel hybrid classifiers achieve results comparable with SVM, but at much lower computational complexity when applied to binary imbalanced classification problems. Then we demonstrate that linear hybrid classifiers can successfully be used in ensemble models. Finally, we incorporate a linear hybrid classifier in an object detection system and show that it achieves comparable to R-CNN [12] performance at a fraction of the training cost.

TABLE 2
Average EER

	SVM	weighted SVM	hybrid	LDA
linear	71%	73.9%	73.8%	68.06%
kernel	83.4%	83.6%	84.0%	–

Each number in the table corresponds to the average EER of 256 binary classifiers, produced on a test set constructed from the 256 categories of Caltech 256.

4.1 Single Classifier

Our goal is to recognize a given class against a very rich background. The Caltech 256 [14] data set contains images from 256 diverse classes and thus approximates the set of *all* natural images quite well. The Scene-15 data set [29] contains far fewer classes, but its images are richer than the images containing objects (as in the Caltech dataset), thus it, too, provides an approximation of a rich background. We tested hybrid classifiers on all classes from these data sets, with the following results:

4.1.1 Accuracy

For the Caltech 256 data set, we used the image representation provided in [11] for a codebook with 1,000 words. We compared the performance of linear and kernel hybrid classifiers to linear and kernel SVMs and their weighted versions, trained in one-against-rest manner. We used an SPM kernel [29] in the kernel classifiers. We provide the results of LDA classifier with the constant background covariance matrix for completeness.

We used 30 images per class as a positive sample. In SVM the negative class consequently contained the rest of the classes, resulting in 7,650 samples. For hybrid classifiers we used 1,280 samples from the same domain to estimate the mean and covariance matrix of the background (these images were excluded from the test set).

For each classifier we computed the EER of the binary classification in which the positive class contained 25 test samples of the corresponding category and the negative class comprised 25 test images per category for all the other categories (in total 6,350 negative examples). We performed training and testing 10 times with random splits into training and test sets and averaged the results. To train the hybrid classifiers we used the CVX optimization package;[2] SVM was trained using the C-SVC option in LIBSVM.[3] Slack, kernel, and probability parameters have been chosen using cross validation. The results are shown in Table 2. The hybrid classifiers outperformed SVM in both the linear and kernel cases, and their accuracy is similar to that of weighted SVM, but the classification and training of the hybrid classifiers enjoys much lower time and space complexities.

The Scenes data set contains only 15 categories. We followed the same test protocol as in the Caltech

2. http://cvxr.com/cvx/download/
3. http://www.csie.ntu.edu.tw/ cjlin/libsvm/

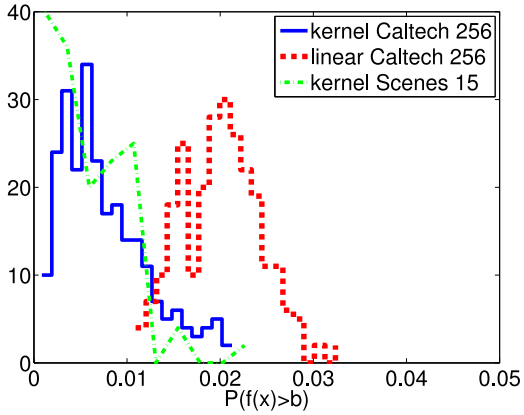


Fig. 4. Histograms of the background empirical probability values in the acceptance region of the hybrid classifiers.

256 experiment, with 30 training and 30 test samples per category, and used the SPM kernel. The average EER rate of the kernel hybrid classifier was 89.36 percent and for kernel SVM it was 89.16 percent.

4.1.2 Probabilistic Model Validation

We tested the validity of the probability constraint for the linear (Eq. (3)) and kernel classifiers (Eq. (8)) by projecting the test set on the separating boundaries corresponding to learned classifiers (we used random splits of data to create different classifiers), and measuring the empirical probability of the background class in their acceptance regions. The histograms of the resulting probabilities are shown in Fig. 4. The value of the probability bound in the training was 0.006 for linear classifiers on Caltech-256, 0.004 for kernel classifiers on Caltech-256 and 0.0032 for kernel classifiers on Scenes-15. As these plots show, the values of the sought probability bound are indeed obtained.

4.1.3 Complexity

In Table 3 the computational and memory requirements of hybrid and SVM kernel classifiers are compared on 256 classes, showing a clear advantage of the proposed method. In our experiments, training kernel SVM using LIBSVM took about four times longer than training the kernel hybrid classifier using CVX. The improvement factor of hybrid versus SVM can be significantly increased by replacing CVX with an optimized package for constrained optimization.

TABLE 3
Comparison of the Computational and Memory Resources for Kernel SVM versus Kernel Hybrid Classifiers for Caltech 256

	SVM (weighted)	hybrid
number of kernel evaluations	600-1,000	230
number of parameters in optimization	7,680	230
number of constraints in optimization	7,680	31
memory usage	450 M	4.5 M

The number of support vectors and of common $\mathbf{s}_i$'s on the much smaller Scenes set was very similar, about 200. The training time of the hybrid classifiers was still faster.

4.1.4 Scalability of the Kernel Hybrid Classifier

To check the scalability of the classifier versus the diversity of the background class, we investigate how the number of $\mathbf{s}_i$'s increases as a function of the number of categories from which the background class is composed. To this end, we used background classes with increasing numbers of categories from the Caltech 256 data set. For each size of the background class (the x-axis) we found the number of vectors required to reach a fixed reconstruction error (the y-axis); here the reconstruction error was set to 0.005 (i.e., on the average the error in approximating a vector was 0.005 of its norm); other error thresholds yielded similar behavior. The plot in Fig. 5 depicts the resulting dependency. The number of vectors is large for a small number of categories and then decreases, and remains nearly constant as the number of categories increases. This behavior—which suggests that the complexity of training and classification does not increase beyond a certain number of categories—can be explained by the fact that we restrict the basis to be a subset of the vectors which need to be approximated. When the background set contains a small number of categories, its diversity is restricted, thus we have to use many vectors to well-approximate the sample set. When the number of categories is large, we can choose fewer—but much better—vectors to approximate the set. At some point the sample is rich enough to allow finding vectors that approximate the entire background class, hence adding more categories does not necessitate increasing the basis. A somewhat similar behavior can be observed when looking at the effective dimension of PCA (defined as a number of eigenvalues that contain the 99 percent of the

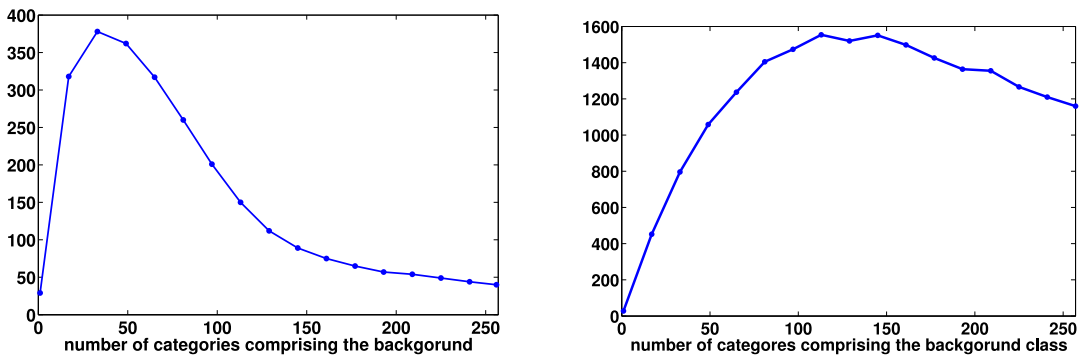


Fig. 5. Left: the relation between the number of vectors required for approximation (with a constant reconstruction error) of unlabeled samples (y-axis) versus the number of categories these samples were taken from (x-axis). Right: the relation between the effective PCA dimension of a set of unlabeled samples (y-axis) versus the number of categories these samples were taken from (x-axis).

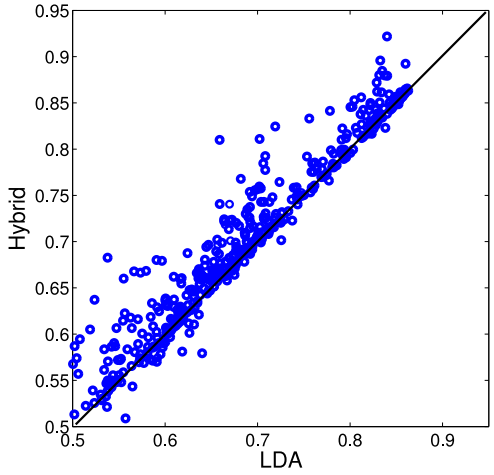


Fig. 6. Comparison of the classifiers which were trained on clusters. Each point represents one cluster, the *x*-coordinate correspond to the AUC of LDA classifier, and the *y*-coordinate to the AUC of a hybrid linear classifier, trained on the same cluster and tested on the all windows from the test set (all object of the category, to which the cluster belongs to, are labeled as positive). The clusters are obtained by partitioning each of 20 categories from PASCAL VOC 2007 into varying number of clusters.

energy of all eigenvectors) as a function of the number of categories (Fig. 5).

To summarize, our experiments show that hybrid classifiers are comparable to SVM when the background set comprises a small number of categories, but as the number of categories increases, the kernel hybrid classifiers becomes much more efficient than kernel SVM. The training of hybrid classifiers is significantly more efficient than training an SVM for any problem size. This is achieved while maintaining classification accuracy better than that of standard SVM, and comparable to that of weighted SVM.

4.2 Ensemble Classifier

We show empirically that linear hybrid classifiers can be incorporated in an ensemble,and that it performs better than an ensemble of other linear classifiers.

4.2.1 Hybrid versus LDA as an Exemplar Model

The ensemble model, proposed in [15], combines LDA classifiers for object detection. Each LDA classifier was trained to separate a subset of images from one category against a general background (estimated using images from all categories). An object category was partitioned into clusters by running normalized cuts [45] on WHO features (whitened HOG) using the cosine of the angle between the feature vectors as an affinity measure. We followed the experimental protocol from [15], which compared different classifiers by training each type of classifier for each cluster and testing them on the test set of PASCAL VOC 2007 [8]. The ground truth for each cluster included all objects of that category.

For each category (20 in total) we used images from the training set and cropped windows corresponding to the object bounding boxes. We represented the cropped image using the Dalal-Triggs variant of HOG features [3] with a fixed number of cells, resulting in fixed length descriptors. The background distribution parameters were learned using windows from all categories. For each category, we performed clustering of feature vectors representing the positive windows with the number of clusters varying from one to ten, removing the clusters with less than three samples. We constructed LDA [16] and Hybrid classifiers (See Section 3.2) for each cluster in the category, and used windows from the validation set for tuning the parameters. We tested the linear classifiers on the test set, in which all windows of that category were labeled as positive and windows from other categories as negatives. Fig. 6 compares the performance of the classifiers trained for each cluster. Each point in Fig. 6 represents a cluster; its x-coordinate corresponds to the AUC of the LDA classifier and its y-coordinate to the AUC of the hybrid classifier. The distribution of the points is above the diagonal line, which shows a clear advantage of the hybrid classifier.

TABLE 4
Classification Rate Measured as EER, Averaged over 26 Letters

Method	Classification
Min. Corr. Ensemble of linear Hybrid Cl.	91.29%
Min. Corr. Ensemble of linear SVM Cl.	87.21%
Hybrid linear	89.32%
Linear SVM	84.87%
Hybrid RBF kernel	96.02%
RBF kernel SVM	96.47%

4.2.2 Hybrid versus SVM in a Minimum Correlation Ensemble

In the following tests we compared the minimum correlation ensemble of hybrid classifiers, introduced in Section 3.1, with the minimum correlation ensemble proposed in [33]. We also tested single linear and kernel hybrid classifiers and single linear and kernel SVM for a baseline.

The tests were performed on a data set of letters from the UCI Machine Learning Repository [37], which included 16-dimensional feature vectors (statistical moments and edge counts which are scaled to fit into a range of integer values from 0 to 15) for the 26 letters in the English alphabet. The letter images are based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce 20,000 samples. For each letter, we used 100 samples for training, 250 for validation, and the rest for testing (about 400 samples per letter). Since the test set includes 25 times more negatives than positives, which leads to about 96 percent classification rate by just classifying all inputs as negative, we used EER as a more faithful measure of performance (thus our results are not directly comparable to those reported in [33]). Table 4 shows the average classification performance of the tested binary classifiers and Fig. 7 compares Minimum Correlation Ensemble of the linear hybrid classifiers with that of linear SVM classifier and with a single hybrid classifier.

4.3 Detection System Using CNN Features

The most natural application of hybrid classifier is object detection, in which the background class is not limited to a predefined number of categories and can be very diverse. Thus the negative class can be modeled once using many windows from general background (either by using a large number of images from an arbitrary source or by using sub-

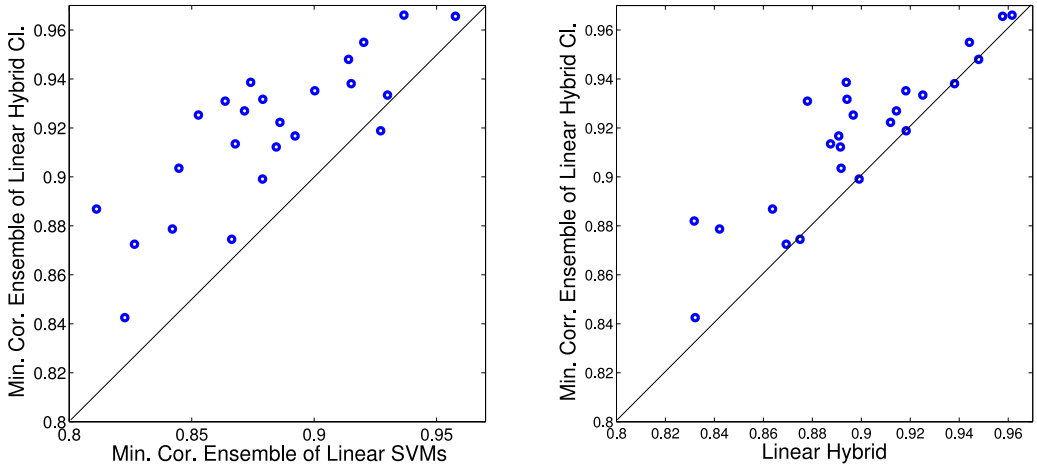


Fig. 7. Comparison of the classification performance on 26 letters, obtained by a minimum correlation ensemble of linear hybrid classifiers with that of minimum correlation ensemble of linear SVM [33] on the left, and of linear hybrid on the right. Each corresponds to 1-EER of single letter.

widows of the real background from the training images, as was done in most detection systems) and used for any target class. Recent work, R-CNN [12] showed excellent detection results on PASCAL-VOC. The main boost in performance was achieved by using features obtained by deep CNN [25], [30], for example, R-CNN which used the final layer of CNN—fc7 [25] as features achieved a mean average precision (mAP) of 54.2 percent on PASCAL VOC 2007, while the previous best result on this set was 34.3 percent [44].

The CNN was first trained on ILSVRC2012 classification data set using image-level annotations only and then fine-tuned (FT) using domain specific data, by replacing the 1,000 outputs to the number of outputs in the domain. The system consists of three steps: first, it extracts around 2,000 bottom-up region proposals using Selective Search [52]; then it computes features for each proposal using a large convolutional neural network (CNN) (using the Caffe [19] implementation of the CNN described in [25]), and finally it classifies each region using class-specific linear SVMs.

We took the R-CNN system with the fc7 features and no bounding-box regression as the test bed for our experiments and replaced the class-specific binary linear SVMs with the hybrid linear classifiers. The rest of the system remained exactly the same. Since the negative class in object detection is very rich and requires many images for training, regular SVM does not perform well due to unbalanced sets. Thus R-CNN ran an iterative process of model refinement, in which training is done several times with the same positive examples and different selection of negative examples. The first model is trained with all negative sub-windows from the first training image, then tested on the second image, and the false positives are included in the negative set. This continues, in the same fashion, for all images in the training set. Obviously, this process requires many rounds of SVM retraining with increasing number of negative examples, a long and tedious process. Instead, we estimated the mean and covariance matrix of the background model using all images and then used them to train hybrid linear classifier for all classes. Table 5 compares the results. We achieve comparable to R-CNN performance at the fraction of the training cost, by removing the need for many rounds of retraining, and also enjoying the faster training of the hybrid classifier. We also tested a hypothesis that the gap in the detection rate between the SVM and hybrid in R-CNN is

TABLE 5
Detection Average Precision (%) on VOC 2007 Test

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN FT - fc7 SVM [12]	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
DPM HSC [44]	32.2	58.3	11.5	16.3	30.6	49.9	54.8	23.5	21.5	27.7	34.0	13.7	58.1	51.6	39.9	12.4	23.5	34.4	47.4	45.2	34.3
R-CNN FT - fc7 SVM random negative sample	57.8	68.4	47.9	35.4	30.7	59.4	68.5	53.5	29.8	55.3	49.9	48.8	54.4	63.7	49.9	28.9	50.9	47.8	51.1	62.9	50.8
R-CNN FT fc7 - HYBRID(1)	57.8	68.1	47.6	36.1	30.8	62.5	68.7	54.2	33.4	55.7	50.4	50.1	58.6	63.3	49.6	28.5	53.9	44.8	51.6	63.0	51.4
R-CNN FT fc7 - HYBRID(2)	58.2	67.5	47.6	36.7	31.2	63.2	68.6	56.2	32.4	55.7	50.2	51.8	57.6	63.4	50.4	28.7	54.0	43.3	53.0	63.6	51.7

The first row shows the results of R-CNN with SVM using hard negative mining on features obtained by CNN pre-trained on ILSVRC 2012 and then fine-tuned on VOC 2007 trainval. The second row presents the best DPM method as a strong baseline. The third row shows the same system as the first row but with the random sample of negatives (background) instead of hard negatives. The last two rows show the results of R-CNN using a hybrid classifier instead of SVM. Row 4 shows the results with constant parameters ($C = 0.001$, $\gamma = 9$) trained using positive samples from trainval set. Row 5 shows the results of R-CNN with hybrid classifier, trained on training set and validated on validation set using different parameters C={0.01, 0.001} and γ between 8 and 14. The model which scored highest was chosen for the test set.

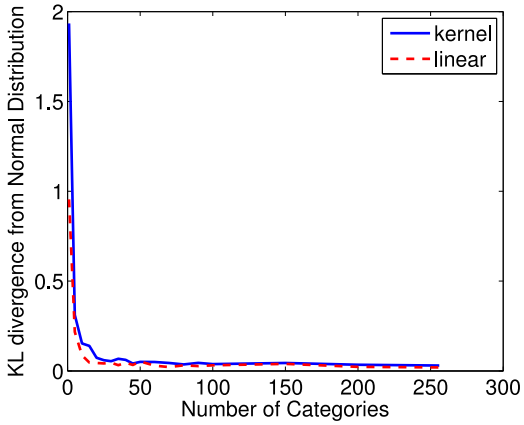


Fig. 8. KL divergence of the projected distribution from normal as a function of a number of categories in the background.

due to hard negative mining, and the training cost could be improved significantly by training SVM with random sampling of negative class for each target object. To this end we sampled 25,000 negative sub-windows from the negative examples (about five sub-windows per image) and trained a single model per object. The mAP of this classifier (row 3 in Table 5) dropped below hybrid classifier, while the cost of training remained higher than hybrid, due to 25,000 negative constraints and larger memory requirements in SVM training.

5 Conclusions

We propose to address the problems arising when training SVM classifiers in a one-against-rest manner, by replacing the negative samples with a distribution representing them. In real visual classification problems, the negative class becomes so rich that it can be viewed as a "background" class and it approaches the distribution of all images. We introduced "hybrid" classifiers, which determine a separating hyperplane between positive samples and this probability distribution, and showed that modeling this distribution is simple, as we are only interested in its projections. Further, we estimate the distribution of the background only once, and then use the same model in training the classifiers for all visual classes. This significantly reduced training complexity, compared to SVM.

We tested the proposed approach in binary classification problems in which the negative class comprises many categories and is much larger than the positive class. In addition to performing well, hybrid classifiers proved to be faster to train and apply than SVM.

Future work will concentrate on alternative models for the background, generalizing the proposed formulation to the multi-class problem, and application to other domains, such as text and video classification.

Appendix A

We tested the validity of the assumption that the distribution of 1D projections of a background comprising many different classes can be approximated by a Gaussian. Fig. 8 shows the KL divergence of an average random projection of the background class from normal distribution as a function of a number of categories in the background. In both linear and kernel projections the KL divergence drops rapidly and then saturates, reaching very low values for a large number of categories.

Acknowledgments

This work has been supported by Israel Science Foundation 839/12.

References

[1] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Proc. 15th Eur. Conf. Mach. Learn.*, 2004, pp. 39–50.
[2] O. Chapelle, A. Zien, and B. Scholkopf, *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.
[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2005, pp. 886–893.
[4] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka, "Visual categorization with bags of keypoints," in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 1–12.
[5] P. Diaconis and D. Freedman, "Asymptotics of graphical projection pursuit," *Ann. Statist.*, vol. 12, pp. 793–815, 1984.
[6] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. Multiple Classifier Syst.*, 2000, pp. 1–15.
[7] T. Downs, K. E. Gates, A. Masters, N. Cristianini, J. Shaw-taylor, and R. C. Williamson, "Exact simplification of support vector solutions," *J. Mach. Learn. Res.*, vol. 2, pp. 293–297, 2001.
[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.* vol. 88, no. 2, pp. 303–338, 2010.
[9] V. Franc and V. Hlavác, "Greedy algorithm for a training set reduction in the kernel methods," in *Proc. 10th Int. Conf. Comput. Anal. Images Patterns*, 2003, pp. 426–433.
[10] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Ann. Stat.*, vol. 28, pp. 337–407, 2000.
[11] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 221–228.
[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 580–587.
[13] K. Grauman and T. Darrell, "The pyramid match kernel. Efficient learning with sets of features" *J. Mach. Learn. Res.*, vol. 8, pp. 725–760, 2007.
[14] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2007-001, 2007.
[15] B. Hariharan, J. Malik, and D. Ramanan, "Discriminative decorrelation for clustering and classification," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 459–472.
[16] T. Hastie, R. Tibshirani, and J. Friedman, The *Elements of Statistical Learning*. New York, NY, USA: Springer, 2001.
[17] Y. Huang, K. Huang, Y. Yu, and T. Tan, "Salient coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1753–1760.
[18] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 487–493.
[19] Y. Jia. (2013). Caffe: An open source convolutional architecture for fast feature embedding [Online]. Available: http://caffe.berkeleyvision.org/
[20] T. Joachims, "Training linear SVMs in linear time," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 217–226.
[21] T. Joachims and C.-N. J. Yu, "Sparse kernel SVMs via cutting-plane training," *Mach. Learn.*, vol. 76, nos. 2/3, pp. 179–193, 2009.
[22] T. Joachims, T. Finley, and C.-N. J. Yu, "Cutting-plane training of structural SVMs," *Mach. Learn.*, vol. 77, pp. 27–59, 2009.

[23] S. S. Keerthi, O. Chapelle, and D. DeCoste, "Building support vector machines with reduced classifier complexity," *J. Mach. Learn. Res.*, vol. 7, pp. 1493–1515, 2006.
[24] D. Keren, M. Osadchy, and C. Gotsman, "Antifaces: A novel, fast method for image detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 7, pp. 747–761, Jul. 2001.
[25] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[26] S. Kotsiantis and D. Kanellopoulos, "Handling imbalanced datasets: A review," *GESTS Int. Trans. Comput. Sci. Eng.*, vol. 30, no. 1, pp. 25–36, 2006.
[27] L. Ladicky and P. H. S. Torr, "Locally linear support vector machines," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 985–992.
[28] G. Lanckriet, L. Ghaoui, C. Bhattacharyya, and M. Jordan, "A robust minimax approach to classification," *J. Mach. Learn. Res.*, vol. 3, pp. 555–582, 2002.
[29] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features. Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2006, pp. 2169–2178.
[30] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[31] Y. J. Lee and O. L. Mangasarian, "RSVM: Reduced support vector machines," in *Proc. SIAM Int. Conf. Data Mining*, 2001.
[32] F.-F. Li, R. Fergus, and P. Perona, "A Bayesian approach to unsupervised one-shot learning of object categories," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 1134–1141.
[33] N. Levy and L. Wolf, "Minimal correlation classification," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 29–42.
[34] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.* vol. 60, no. 2, pp. 91–110, 2004.
[35] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of Exemplar-SVMs for object detection and beyond," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 89–96
[36] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.
[37] A. Asuncion and D. J. Newman, "UCI Machine Learning Repository," Dept. of Inf. Comput. Sci., Univ. California, Irvine, CA, 2007.
[38] M. Osadchy and D. Keren, "Incorporating the Boltzmann prior in object detection using SVM," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2006, pp. 2095–2101.
[39] F. Perronnin, J. Sánchez, and Y. Liu, "Large-scale image categorization with explicit data embedding. in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 2297–2304.
[40] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *J. Mach. Learn. Res.*, vol. 5, pp. 101–141, 2004.
[41] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2005, pp. 860–867.
[42] B. Scholkopf, J. Platt, A. Smola, and R. Williamson, "Estimating the support of a high dimensional distribution," *Neural Comput.*, vol. 13, pp. 1443–1471, 2001.
[43] S. Shalev-Shwartz, Y. Singer, and N. Srebro,"Pegasos: Primal estimated sub-gradient solver for SVM," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 807–814.
[44] X. Ren and D. Ramanan, "Histograms of sparse codes for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 3246–3253.
[45] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[46] A. J. Smola and B. Schölkopf, "Sparse greedy matrix approximation for machine learning," in *Proc. Int. Conf. Mach. Learn.*, 2000, pp. 911–918.
[47] Z. Song, Q. Chen, Z. Huang, Y. Hua, and S. Yan, "Contextualizing object detection and classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1585–1592.
[48] A. Srivastava, A. B. Lee, E. P. Simoncelli, and S.-C. Zhu, "On advances in statistical modeling of natural images," *J. Math. Imaging Vis.*, vol. 18, pp. 17–33, 2003.
[49] I. Steinwart, "Sparseness of support vector machines," *J. Mach. Learn. Res.*, vol. 4 pp. 1071–1105, 2003.
[50] D. Tsai, Y. Jing, Y. Liu, H. A. Rowley, S. Ioffe, and M. J. Rehg, "Large-scale image annotation using visual synset," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 611–618.
[51] D. Tax, "One-class classification," Ph.D. dissertation, Delft Univ. of Technol., Delft, Netherlands, Jun. 2001.
[52] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.* vol. 104, no. 2, pp. 154–171, 2013.
[53] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 480–492, Mar. 2012.
[54] Y. Weiss and W. T. Freeman, "What makes a good model of natural images," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.
[55] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *Int. J. Comput. Vis.*, vol. 73, pp. 213–238, 2007.
[56] S. C. Zhu and D. Mumford, "Prior learning and Gibbs reaction-diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.* vol. 19, no. 11, pp. 1236–1250, Nov. 1997.

Margarita Osadchy received the PhD degree with honors in computer science from the University of Haifa, Israel, in 2002. From 2001 to 2004, she was a visiting research scientist at the NEC Research Institute. During 2004-2005, she was a postdoctoral fellow in the Department of Computer Science at the Technion-Israel Institute of Technology. Since 2005, she has been with the Computer Science Department, University of Haifa. Her main research interests are machine learning, computer vision, and cyber security.

Daniel Keren received the PhD degree from the Hebrew University in 1991, and then spent three years as a postdoctoral researcher at Brown University. Since then, he has been with the Computer Science Department, University of Haifa. His main research interests include computer vision, regularization, and monitoring of large-scale distributed systems.

Dolev Raviv received the BSc degree from the Technion-Israel Institute of Technology majoring in computer engineering in 2012. Currently, he is a graduate student in the Computer Science Department, University of Haifa. His researching is in the field of machine learning and computer vision.