Learning Non-linear SVM in Input Space for Image Classification

Gaurav Sharma, Frédéric Jurie, Patrick Pérez

Learning Non-linear SVM in Input Space for Image Classification

Gaurav Sharma, Frédéric Jurie and Patrick Pérez

Abstract—The kernel trick enables learning of non-linear decision functions without having to explicitly map the original data to a high dimensional space. However, at test time, it requires evaluating the kernel with each one of the support vectors, which is time consuming. We propose a novel approach for learning non-linear support vector machine (SVM) corresponding to commonly used kernels in computer vision, namely (i) Histogram Intersection, (ii) χ^2, (ii) Radial Basis Function (RBF) and (iv) RBF with χ^2 distance, without using the kernel trick. The proposed classifier incorporates non-linearity while maintaining $O(D)$ testing complexity (for D-dimensional space), compared to $O(D \times N_{sv})$ (for N_{sv} number of support vectors) when using the kernel trick. We also promote the idea that such efficient non-linear classifier, combined with simple image encodings, is a promising direction for image classification. We validate the proposed method with experiments on four challenging image classification datasets. It achieves similar performance w.r.t. kernel SVM and recent explicit feature mapping method while being significantly faster and memory efficient. It obtains competitive performance while being an order of magnitude faster than the state-of-the-art Fisher Vector method and, when combined with it, consistently improves performance with a very small additional computation cost.

Index Terms—Support vector machine, margin maximization, kernel methods, image classification.

1 Introduction

Image classification is one of the central problems of computer vision. Recent works have addressed various image domains, *e.g.* outdoor scenes [1], [2], indoor scenes [3], [4], object images [5], [6], [7], [8], human attributes [2], [9], [10]. The standard pipeline for an image classification system is (i) extract local image features, (ii) encode them, (iii) aggregate (or pool) the encodings to make a fixed length image representation and then finally (iv) use a classifier to learn the decision boundaries between the different classes. The seminal works of Sivic and Zisserman [11] and Csurka *et al.* [12] introduced the *bag-of-features* (BoF) representation in the computer vision community. It is based on encoding local features by using vector quantization and aggregating them by simple zeroth order statistics, *i.e.,* histogramming/counting over the quantization bins. Since such a simple pooling leads to loss of spatial information, Lazebnik *et al.* [1] proposed using spatial pyramid (SP) pooling. When used with non-linear classifiers, *e.g.* kernel support vector machines (SVMs), SP lead to state-of-the-art systems [1], [5], [6]. However, using simple statistics required the use of kernel SVMs, which at test time necessitated computation of a large number (order of number of training images) of kernel evaluations and hence were quite expensive. Driven by this limitation, on one hand, researchers started focusing on offloading the complexity from the classifier step to the encoding and aggregation step, demonstrating that using better coding (using higher order statistics) and aggregation [13], [14], [15], [16], [17], [18] gives better results with inexpensive linear SVM. On the other hand, in parallel, methods were proposed to make non-linear classifiers efficient [19], [20], [21], [22]. Balancing this trade-off between encoder and classifier complexity has remained an important question for image classification.

In the present paper, we focus on simple encodings and efficient complex classifiers. As a first contribution, we propose to learn efficient nonlinear SVM directly in the input space (a preliminary version of this part appeared in [23]), with popular and successful kernels used in computer vision, namely (i) Histogram Intersection, (ii) χ^2, (ii) Radial Basis Function (RBF) and (iv) RBF with χ^2 distance. Among these different kernels, we find the RBF-χ^2 to be particularly interesting as it is known to give the state-of-the-art performance on image classification tasks [5], [6], [8].

For the second contribution, we first note that, while the recently proposed complex encodings, *e.g.* [7], [18], lead to state-of-the-art performance, they are relatively slower than the simpler bag-of-features encoding with approximate nearest neighbor based hard quantization of local features [24]. Also, it has been demonstrated empirically that using complementary features, *e.g.* based on gray and color, leads to improvement in performance, albeit with expensive RBF-χ^2 kernel SVMs [5], [25].

Motivated by this observation, we empirically investigate the speed vs. performance trade-off by using the proposed RBF-χ^2 SVM with fast and simple statistics based on complementary features. We show that the efficient nonlinear classifiers, as proposed, when used

- *Gaurav Sharma (http://www.grvsharma.com) and Patrick Pérez (http://www.technicolor.com/en/patrick-perez) are with Technicolor.*
- *Frédéric Jurie (http://jurie.users.greyc.fr) is with the GREYC CNRS UMR 6072, Universite de Caen Basse-Normandie, France.*
- *Part of the work was done when GS was with the Universite de Caen Basse-Normandie.*

with simple statistics of complementary features lead to substantial gain in performance at a very small cost in computational time. This allows us to design systems at different operating points balancing time and performance (Fig. 1). Most interestingly, we find that it is possible to reach 95% of the performance of state-of-the-art methods (which use complex encoding and linear SVM) using complementary features with proposed non-linear RBF-χ^2 SVM, while being $11\times$ faster for complete testing and $33\times$ faster when excluding the common feature extraction part (which could run at relatively negligible cost on dedicated hardware such as GPU). Further, it is possible to improve the state-of-the-art while adding a very small run time cost. We obtain such improvements consistently on four recent publicly available challenging image classification datasets: (i) Flickr Materials [26], (ii) MIT Indoor Scenes [4], (iii) Human Attributes [27] and (iv) Pascal VOC 2007 [5].

1.1 Related Work

The two main ingredients of many successful approaches for visual recognition are (i) the representation of images by distributions (*i.e.,* histograms) of visual features such as in BoF [12] and HOG [28] and (ii) the use of margin maximizing classifiers such as SVMs [29]. Systems built on them have led to state-of-the-art performance on image classification [1], [2], [30] and object detection [6], [31], [32].

The standard formulation for learning classifiers is the SVM primal formulation (Eq. 2, see [29] for more details) which allows the learning of a linear classification boundary in the space of (images represented as) distributions. However, general visual tasks, *e.g.* scene or object based classification of unconstrained images, are very challenging due to the presence of high variability due to viewpoint, lighting, pose, *etc.* and linear decision boundaries are not sufficient. Many competitive methods in image classification [1], [5] and object detection [6], [31], thus, use non linear classifiers. Such non linear classifiers are obtained by using the *kernel trick* with the dual formulation (Eq. 3) of the SVM. The SVM dual formulation only requires the dot products between the vectors and so a nonlinear *kernel* function $k(\mathbf{x}_1, \mathbf{x}_2)$ is used which implicitly defines a (non linear) mapping $\phi : \mathbb{R}^D \rightarrow \mathcal{F}$ of input vectors to a high (potentially infinite) dimensional *feature* space with $k(\mathbf{x}_1, \mathbf{x}_2) = \langle \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) \rangle$. With the kernel trick, a linear decision boundary in the feature space is learned which corresponds to a non linear decision boundary in the input space. Such kernel based SVMs have been shown to improve the performance of linear SVMs in many visual tasks (*e.g.* classification and detection) by a significant margin, *e.g.* [1], [6], [31].

While the dual formulation allows the learning of non linear decision boundaries, the computation of classifier decision for a test vector $\mathbf{x}$,

$$f(\mathbf{x}) \propto \sum_{i=1}^{N_{sv}} c_i k(\mathbf{x}, \mathbf{x}_i) \tag{1}$$

(c_i being the model parameters), depends on kernel computation with *all support vectors* $\{\mathbf{x}_i \in \mathbb{R}^D | i = 1 \ldots N_{sv}\}$. Hence, the test time and space complexities becomes $O(D \times N_{sv})$[1] vs. $O(D)$ for the linear case (where $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$). In practice, N_{sv} is of the order of number of training examples, and this leads to significant cost in terms of time and space. Such high cost makes it impractical for kernel based classifiers to be used for large scale tasks, *e.g.* object detection, in which the classifier has to be applied to more than 100,000 windows per image [6], [31] or large scale image classification [34] with thousands of classes. Similarly, it makes them impractical to use with limited capability mobile devices in consumer applications, *e.g.* smart-phone/tablet applications for object or landmark recognition or for real time object based video editing.

Traditionally, classifiers based on non linear SVMs have led to the best results on image classification problems with the standard BoF [11], [12] image representation (*e.g.* the high ranking entries of the PASCAL VOC 2007 competition [5]). However, such classifiers incur a very high testing cost, as explained above. To address

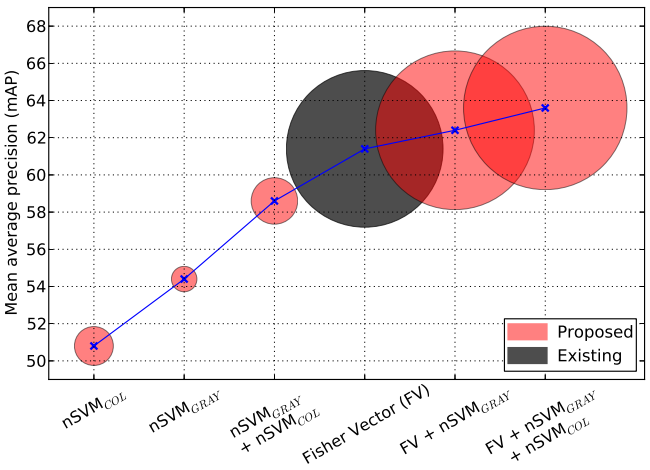

Fig. 1. The classification performances (mAP) on the Pascal VOC 2007 [5] dataset: (i) the proposed non-linear SVM method (denoted nSVM, in red/light) with complementary features, *i.e.,* gray (subscript GRAY) and color (subscript COL); (ii) Fisher Vector [7], an existing state-of-the-art method (denoted FV, in black/dark); (iii) combinations of the two, by late fusion (denoted FV + nSVM, in red/light). The areas of the disks are proportional to the testing times of the respective methods. As argued in text, 'nSVM$_{GRAY}$+nSVM$_{COL}$' and 'FV+nSVM$_{GRAY}$+nSVM$_{COL}$' are especially appealing.

1. While noting that there are kernels with polynomial complexities, *i.e.,* $O(D^n)$ with $n > 1$ (*e.g.* Earth movers' distance based kernels with worst case exponential complexity [33]), in the present work we consider only those with linear complexities

this problem of efficiency, approaches have primarily taken one of the following two directions.

Efficient classification. Methods were proposed to reduce, primarily, the test time complexity of kernel SVMs. Bruges [35] gave a method to approximate the decision function of kernel SVM using a reduced set of vectors (w.r.t. the set of all support vectors). Many other works were then proposed in a similar spirit, *e.g.* [36], [37], [38], [39].

Recently, Maji *et al.* [19] showed that SVM classifier decision corresponding to the histogram intersection (HI) kernel can be computed in logarithmic (w.r.t. N_{sv}) time and also proposed a constant (w.r.t. N_{sv}) time and space approximation for the same. Mapping features to another, higher dimensional, yet finite space where the inner product of the transformed vectors approximates the kernel and then using recent fast linear SVM classification methods, *e.g.* [40], [41], [42], [43], [44], has been quite popular recently. To this end, Williams and Seeger [45], [46], Smola and Scholkopf [47] and Fine and Scheinberg [48] used Nystrom's approximation. Perronnin *et al.* [21] applied Nystrom's approximation to each dimension for additive kernels for image classification. In a data independent way, Rahimi and Recht [49] proposed to use random Fourier features, Raginsky and Lazebnik [50] proposed to construct binary codes corresponding to shift invariant kernels. Maji and Berg [20] approximated the feature map corresponding to the HI kernel and Vedaldi and Zisserman [22] approximated general additive kernels, *e.g.* HI and χ^2. The advantage of resorting to such explicit mappings is that they allow the use of linear classification methods with the feature mapped vectors, but the drawback is that each data point has to be explicitly mapped, which has a cost.

Stronger Encodings. Simple extensions of BoF, *e.g.* from hard assignment to a single quantization bin to soft assignments of local features to multiple bins, were shown to improve performance [51], [52], [53], [54]. Later works proposed even more sophisticated coding and pooling methods, *e.g.* sparse coding [13], [14], [15], [16], [17], [18], [55] with max pooling reporting very good classification performances with linear classifier. Works were also reported using higher order statistics of features for coding, *e.g.* Super Vectors [56] and Fisher Vectors [7], [57], which, used with linear SVMs, are the current state-of-the-art methods for image classification [24].

In addition to all the methods cited above, our method is also loosely related to the pre-image and reduced set problem in kernel methods [35], [58], [59]. However, the objectives of those works is in sharp contrast with the proposed method. We comment more on this in § 2.3.

In this paper, we take an (as far we know) unexplored route and show that it is possible to learn nonlinear classifiers directly in the input space without using the dual formulation and the kernel trick, achieving similar classification performance at improved speed and memory requirements.

In a recent empirical study [24], the complex encoding method of Fisher Vectors [7] was shown to be the state-of-the-art image representation and has been applied to many classification tasks [3], [24], [60], [61]. In our experiments, we show that our method leads to 95% of the performance of this state-of-art method, while being an order of magnitude faster, and, when combined with it, leads to consistent improvements at a very small additional cost (Fig. 1).

2 Approach

Support vector machine (SVM) primal formulation, *i.e.*,

$$\min_{\mathbf{w}\in\mathbb{R}^D} \frac{\lambda}{2}||\mathbf{w}||_2^2 + \frac{1}{N}\sum_{i=1}^{N}\xi_i \qquad (2)$$
$$\text{sb.t.} \quad y_i\mathbf{w}^\top\mathbf{x}_i \geq 1-\xi_i \text{ and } \xi_i \geq 0, \ \forall\ i=1\dots N,$$

is a standard formulation to learn a linear classifier, where $\mathbf{w}$ is a normal to the linear decision hyperplane and $(\mathbf{x}_i, y_i) \in \mathbb{R}^D \times \{-1,+1\}$, $i = 1\dots N$, are the N training vector and label pairs. The optimization problem is convex and well studied, and many standard libraries (*e.g.* `liblinear` [44]) exist for solving it. However, only a linear decision boundary (*i.e.*, a hyperplane parametrized by $\mathbf{w}$) can be learned with this formulation. General visual tasks, *e.g.* scene or object based classification of unconstrained images, are very challenging due to the high variability caused by changes in viewpoint, lighting, pose, *etc.* and linear decision boundaries are not sufficient. To allow learning more complex nonlinear decision boundaries, the dual formulation of the problem

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^N} \sum_{i=1}^{N}\alpha_i + \left(\frac{1}{2}-\frac{1}{\lambda}\right)\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\ \mathbf{x}_i^\top\mathbf{x}_j \qquad (3)$$
$$\text{sb.t.} \quad 0 \leq \alpha_i \leq \frac{1}{N}, \ \forall\ i=1\dots N,$$

is used. In this formulation, the *kernel trick* can then be mobilized whereby dot products are replaced by a suitable kernel function k such that:

$$k(\mathbf{x}_i,\mathbf{x}_j) = \langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\rangle_{\mathcal{F}} \qquad (4)$$

with

$$\phi : \mathbb{R}^D \to \mathcal{F} \qquad (5)$$

being a *feature map* from the *input* space $\mathbb{R}^D$ into a high (potentially infinite) dimensional Hilbert *feature* space $\mathcal{F}$ where the classes are hoped to be linearly separable. In the kernelized version of dual problem (3), dot products $\mathbf{x}_i^\top\mathbf{x}_j$ are replaced by kernel-based similarities $k(\mathbf{x}_i,\mathbf{x}_j)$ and map ϕ is not explicitly required (see [29] for detailed discussion). Learning a kernel based non-linear SVM with the primal formulation, using similarly the kernel trick, is also possible [62] but less common in practice.

Towards the goal of learning a nonlinear classifier in input space, we start with the SVM problem (unconstrained formulation equivalent to Eq. 2) in feature space, obtained by mapping the input space vectors using the feature map ϕ:

$$\min_{\mathbf{w}_\phi \in \mathcal{F}} \underbrace{\frac{\lambda}{2}||\mathbf{w}_\phi||^2_{\mathcal{F}} + \frac{1}{N}\sum_{i=1}^{N} l(y_i, \langle \mathbf{w}_\phi, \phi(\mathbf{x}_i)\rangle_{\mathcal{F}})}_{L_\phi(\mathbf{w}_\phi)}, \quad (6)$$

with l being the *hinge loss* function

$$l(y, \delta) = \max(0, 1 - y\delta) \quad (7)$$

and where $\mathbf{w}_\phi \in \mathcal{F}$ denotes the (parameters of the) linear decision boundary in the feature space. We note that arbitrary vectors in feature space might not have *pre-images* relative to ϕ in input space [29], [59]. Hence, we denote by $\mathbf{w} \in \mathbb{R}^D$ either the pre-image of $\mathbf{w}_\phi \in \mathcal{F}$ if it exists or the best approximate pre-image otherwise, *i.e.*,

$$\mathbf{w}_\phi \approx \phi(\mathbf{w}). \quad (8)$$

We can now derive a new objective function in input space, $L(\mathbf{w}) = L_\phi(\phi(\mathbf{w}))$, which reads:

$$L(\mathbf{w}) = \frac{\lambda}{2}||\phi(\mathbf{w})||^2_{\mathcal{F}} + \frac{1}{N}\sum_{i=1}^{N} l(y_i, \langle \phi(\mathbf{w}), \phi(\mathbf{x}_i)\rangle_{\mathcal{F}}). \quad (9)$$

This objective is same as the original objective upto the approximation introduced due to the possible non-existence of a pre-image for the solution $\mathbf{w}_\phi$ of original problem. Said differently, by minimizing $L(\mathbf{w})$ in input space, we solve kernel SVM problem (6) under the constraint that the normal to separating plane in feature space is in $\phi(\mathbb{R}^D)$.

Although we are working with non-negative input vectors such as bag-of-features histograms, we expect the vector $\mathbf{w}$ to be negative as well, in general. In that case, we can see the $\mathbf{w}$ vector as a combination of two non-negative vectors with disjoint supports:

$$\mathbf{w} = \mathbf{w}_+ - \mathbf{w}_-, \text{ with } \mathbf{w}_+ \text{ and } \mathbf{w}_- \in \mathbb{R}^D_+, \quad (10)$$

where the $\mathbf{w}_+$ ($\mathbf{w}_-$) capture the discriminative information supporting the positive (negative) class.

Given a kernel k, the regularization term in Eq. 9 becomes

$$\mathcal{R}(\mathbf{w}) := ||\phi(\mathbf{w})||^2_{\mathcal{F}} = \langle \phi(\mathbf{w}), \phi(\mathbf{w})\rangle_{\mathcal{F}} = k(\mathbf{w}, \mathbf{w}) \quad (11)$$

and hinge loss computations in the second term involve computing

$$y_i \langle \phi(\mathbf{w}), \phi(\mathbf{x}_i)\rangle_{\mathcal{F}} = y_i f(\mathbf{x}_i; \mathbf{w}) \quad (12)$$

where $f(\cdot; \mathbf{w}) := k(\mathbf{w}, \cdot)$ acts like a scoring function, parametrized by $\mathbf{w}$. This score function induces a decision boundary in the input space that is non-linear in general (unless k is a monotonic function of dot product in input space).

Hence, the complete feature space optimization can be written (approximately) in input space as minimizing

$$L(\mathbf{w}) = \frac{\lambda}{2}k(\mathbf{w}, \mathbf{w}) + \frac{1}{N}\sum_{i=1}^{N} \max(0, 1 - y_i k(\mathbf{w}, \mathbf{x}_i)). \quad (13)$$

Minimizing L_ϕ w.r.t. $\mathbf{w}_\phi$ in the feature space is a convex problem (in input space). It is solved using the kernel trick which evades the need of explicitly specifying ϕ. However, at test time, to compute the prediction for a test image, computing kernels with all the support vectors (which are of the order of number of training images) is required.

In this paper, instead of minimizing the convex objective L_ϕ (Eq. 6) in feature space, we propose to directly minimize the nonlinear and non-convex objective L (Eq. 13) in input space.

2.1 Nonlinear SVM with important kernels

We now show how nonlinear SVM learning problem can be formulated and learned directly in the input space for four kernels popular in computer vision namely Histogram Intersection, χ^2, RBF with Euclidean distance and RBF with χ^2 distance function. We start with the optimization (13), written for any general kernel $k(\mathbf{x}, \mathbf{y})$ as,

$$\min_{\mathbf{w}, b} \frac{\lambda}{2}k(\mathbf{w}, \mathbf{w}) + \frac{1}{N}\sum_{i=1}^{N} \max[0, m - y_i(k(\mathbf{w}, \mathbf{x}_i) + b)], \quad (14)$$

where we have (i) replaced the unit margin with a free parameter m (we comment more on this in § 3.1) and (ii) added, in the scoring function, a bias term $b \in \mathbb{R}$ to be learned along with $\mathbf{w}$. The latter is critical for RBF kernels as their range is $\mathbb{R}^+$. With this view we now consider the four different kernels. For each, we give the expression for the kernel which we use with Eq. 14 and derive the analytical expressions for the subgradients that will be used to learn the classifier by means of a stochastic gradient descent algorithm. As we shall see, defining some of these kernels with the traditional distances leads however to some technical problems. We remedy them by introducing shifted versions of the distances.

Histogram intersection kernel. The generalized HI kernel is given by

$$k_h(\mathbf{x}, \mathbf{y}) = \sum_{d=1}^{D} \frac{x_d y_d}{|x_d y_d|} \min(|x_d|, |y_d|), \quad \text{(HI kernel)}$$

where the subscript denotes the coordinates of the vector, *i.e.*,

$$\mathbf{x} = (x_1, \ldots, x_D). \quad (15)$$

	RBF-Euclidean (k_{re})	Histogram intersection (k_h)	χ^2 (k_c)	RBF-χ^2 (k_{rc})
$f(\mathbf{x};\mathbf{w}) = k(\mathbf{w},\mathbf{x})$	$\exp\left(\frac{1}{\gamma}\sum_{d=1}^{D} x_d w_d\right)$	$\sum_{d=1}^{D} \frac{x_d w_d}{\lvert x_d w_d\rvert} \min(\lvert x_d\rvert, \lvert w_d\rvert)$	$\sum_{d=1}^{D} \frac{2x_d w_d}{\lvert x_d\rvert+\lvert w_d\rvert}$	$\exp\left(\frac{1}{\gamma}\sum_{d=1}^{D} \frac{2x_d w_d}{\lvert x_d\rvert+\lvert w_d\rvert}\right)$
$\nabla_{w_d} f(\mathbf{x};\mathbf{w})$	$\frac{x_d}{\gamma} k_{re}(\mathbf{w},\mathbf{x})$	1 if $\lvert w_d\rvert < x_d$, 0 ow	$\frac{2x_d\lvert x_d\rvert}{(\lvert x_d\rvert+\lvert w_d\rvert)^2}$	$\frac{2x_d\lvert x_d\rvert}{\gamma(\lvert x_d\rvert+\lvert w_d\rvert)^2} k_{rc}(\mathbf{w},\mathbf{x})$
$\mathcal{R}(\mathbf{w}) = k(\mathbf{w},\mathbf{w})$	$\exp(\frac{1}{\gamma}\lvert\lvert\mathbf{w}\rvert\rvert_2^2)$	$\lvert\lvert\mathbf{w}\rvert\rvert_1$	$\lvert\lvert\mathbf{w}\rvert\rvert_1$	$\exp(\frac{1}{\gamma}\lvert\lvert\mathbf{w}\rvert\rvert_1)$
$\nabla_{w_d}\mathcal{R}(\mathbf{w})$	$\frac{2w_d}{\gamma}\exp(\frac{1}{\gamma}\lvert\lvert\mathbf{w}\rvert\rvert_2^2)$	$\frac{w_d}{\lvert w_d\rvert}$	$\frac{w_d}{\lvert w_d\rvert}$	$\frac{w_d}{\gamma\lvert w_d\rvert}\exp(\frac{1}{\gamma}\lvert\lvert\mathbf{w}\rvert\rvert_1)$

TABLE 1
The score and regularization functions with their derivatives for four important popular computer vision kernels. Note that RBF-Euclidean kernel k_{re}, χ^2 kernel k_c and RBF-χ^2 kernel k_{rc} are defined in a slightly unusual form, assuming that input vectors $\mathbf{x}$ are ℓ_2 (resp. ℓ_1) normalized for the former (resp. the two others). See text for details.

The subgradients for the regularization and scoring function are given by

$$\nabla_{w_d} k_h(\mathbf{w},\mathbf{w}) = \nabla_{w_d}||\mathbf{w}||_1 = \frac{w_d}{|w_d|}, \tag{16}$$

$$\nabla_{w_d} k_h(\mathbf{w},\mathbf{x}) = \begin{cases} 1 & \text{if } |w_d| < x_d, \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

where we have used the fact that we are working with histograms, *i.e.*, $x_d \geq 0\ \forall,\ d = 1,\ldots,D$.

χ^2 kernel. The traditional χ^2 kernel is based on the χ^2 distance and is given by

$$k_{\bar{c}}(\mathbf{x},\mathbf{y}) = c - \frac{1}{2}\sum_{d=1}^{D}\frac{(x_d - y_d)^2}{|x_d| + |y_d|}, \tag{18}$$

where c is a fixed constant. However, using this form of the kernel leads to no regularization as $k_{\bar{c}}(\mathbf{w},\mathbf{w}) = c$ is independent of $\mathbf{w}$. When the vectors $\mathbf{x},\mathbf{y}$ are ℓ^1 normalized,

$$k_{\bar{c}}(\mathbf{x},\mathbf{y}) = (c-1) + \sum_{d=1}^{D}\frac{2x_d y_d}{|x_d| + |y_d|}, \tag{19}$$

which suggests to define instead a generalized χ^2 kernel as

$$k_c(\mathbf{x},\mathbf{y}) = \sum_{d=1}^{D}\frac{2x_d y_d}{|x_d| + |y_d|}. \qquad (\chi^2 \text{ kernel})$$

With this definition, the required subgradients are given by (with $x_d \geq 0$)

$$\nabla_{w_d} k_c(\mathbf{w},\mathbf{w}) = \frac{w_d}{|w_d|}, \tag{20}$$

$$\nabla_{w_d} k_c(\mathbf{w},\mathbf{x}) = \frac{2x_d|x_d|}{(|x_d| + |w_d|)^2}. \tag{21}$$

RBF-Euclidean kernel. The usual definition of radial basis function (RBF) kernels is given by

$$k(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{1}{\gamma}\mathcal{D}^2(\mathbf{x},\mathbf{y})\right), \tag{22}$$

where $\mathcal{D}(\cdot)$ is the corresponding distance function. Similar to the χ^2 kernel above, if we define the kernel this way the regularizer comes out to be $\mathcal{R}(\mathbf{w}) = k(\mathbf{w},\mathbf{w}) = \exp(0) = 1$, independent of $\mathbf{w}$, *i.e.*, no regularization. For the Euclidean distance we have,

$$\mathcal{D}_E^2(\mathbf{x},\mathbf{y}) = ||\mathbf{x}-\mathbf{y}||^2 = 2 - 2\mathbf{x}^\top\mathbf{y} \tag{23}$$

if the vectors are ℓ^2 normalized. Hence we define the RBF-Euclidean kernels as

$$k_{re}(\mathbf{x},\mathbf{y}) = \exp\left(\frac{1}{\gamma}\mathbf{x}^\top\mathbf{y}\right). \qquad \text{(RBF-Euclidean kernel)}$$

The required subgradients are then given by

$$\nabla_{w_d} k_{re}(\mathbf{w},\mathbf{w}) = \frac{2w_d}{\gamma}\exp\left(\frac{1}{\gamma}||\mathbf{w}||_2^2\right), \tag{24}$$

$$\nabla_{w_d} k_{re}(\mathbf{w},\mathbf{x}) = \frac{x_d}{\gamma}\exp\left(\frac{1}{\gamma}\mathbf{w}^\top\mathbf{x}\right). \tag{25}$$

RBF-χ^2 kernel. Similarly to previous construct, we define the generalized RBF-χ^2 kernel as

$$k_{rc}(\mathbf{x},\mathbf{y}) = \exp\left(\frac{1}{\gamma}\sum_{d=1}^{D}\frac{2x_d y_d}{|x_d| + |y_d|}\right). \qquad \text{(RBF-}\chi^2\text{ kernel)}$$

The required subgradients are then given by

$$\nabla_{w_d} k_{rc}(\mathbf{w},\mathbf{w}) = \frac{w_d}{\gamma|w_d|}\exp\left(\frac{1}{\gamma}||\mathbf{w}||_1\right), \tag{26}$$

$$\nabla_{w_d} k_{rc}(\mathbf{w},\mathbf{x}) = \frac{2x_d|x_d|}{\gamma(|x_d| + |w_d|)^2} k_{rc}(\mathbf{w},\mathbf{x}). \tag{27}$$

2.2 Linear decision function as a convex relaxation for histogram intersection kernel

The optimization problem (14) is a non-convex problem and to gain some insight on the nature of the problem and the feasibility of finding a resonable solution, we present a possible method to solving it.

A common approach to deal with non-convex optimization is to resort to approximation and solve a convex

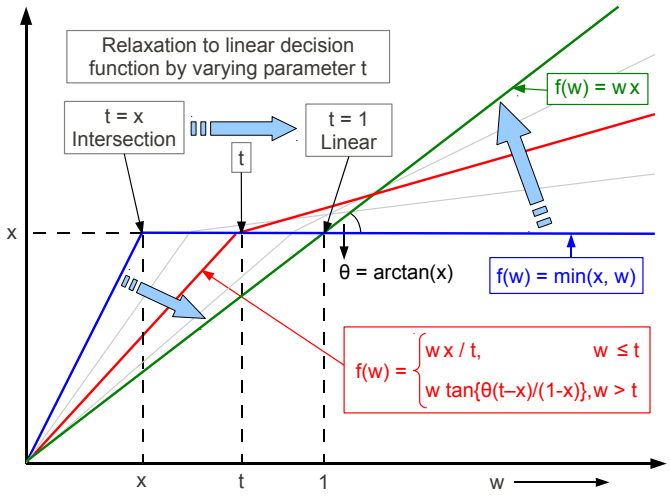

Fig. 2. The linear decision function (green) can be seen as a convex relaxation of the histogram intersection like decision function (blue). If we introduce a parameter $t \in [x, 1]$, we can construct a series of relaxations of the intersection like decision function converging to the convex linear decision function as t varies from x to 1, shown in red in here. Only the first quadrant is shown, the graphs in the third quadrant are obtained by mirroring about the two axes i.e. all functions are odd functions with $f(-x) = -f(x)$.

relaxation of the objective instead. Fig. 2 shows how a series of relaxations of the generalized histogram intersection kernel based decision function can be constructed, with final convergence to linear, and hence, convex decision function (convex optimization problem). An algorithm could, thus, be designed to solve the nonlinear optimization by successive smoothing. The algorithm would proceed by first solving a highly smoothed convex problem (corresponding to linear decision function) and then iteratively solving less smoothed versions, of the objective, initialized with the solution of previous ones.

The intersection decision function (for single dimensional feature $x \in \mathbb{R}$) could be relaxed with the following function, parametrized by t,

$$f_t(w, x) = \begin{cases} \frac{wx}{t} & \text{if} \quad w \leq t, \\ w \tan\left\{\tan^{-1}(x).\frac{t-x}{1-x}\right\} & \text{otherwise}, \end{cases} \tag{28}$$

for $x \in \mathbb{R}_+$ (the function is odd, *i.e.*, $f(w) = -f(-w),\ \forall w \in \mathbb{R}_-$). The parameter $t \in [x, 1]$ ($x \leq 1$ as it is a component of ℓ_1 normalized BoF histogram) controls the amount of smoothness on the objective function. Fig. 2 illustrates the smoothed versions of the function for one dimensional w. We have no smoothing when $t = x$, while $t = 1$ leads to heavily smoothed linear decision function. When the decision function is linear, *i.e.*, $t = 1$, the optimization problem becomes convex.

When using this function for the sum of hinge losses for all d dimensional examples (*i.e.*, many different $\mathbf{x} \in \mathbb{R}^D$) t is replaced with $\max(t, x_d)$ for each dimension of each example. The discussion above, pertaining to the convex relaxation of the objective, changes accordingly.

Similar relaxations could also be constructed for the other kernels and used to solve the optimization with successive relaxations initialized with the solutions to the previous ones.

However, we show below that it is not necessary to use this relaxation based framework as more practical solutions are possible (see section 2.4). The rationale for the above discussion is to give an insight into why gradient descent can work despite the problem is not convex.

2.3 Relation with pre-image and reduced set kernel methods

While many of the current method take the 'forward' path of approximately mapping the features into higher dimensional spaces where the dot products approximate the kernel evaluated in the input space [19], [21], [22], we propose a 'backward' path of mapping the non-linear classification boundary back in to the input space. Our method is thus reminiscent of the reduced set and pre-image problems in kernel methods [29], [35], [59] where the set of support vectors for SVM (or the input vectors for kernel PCA) is reduced to a set with significantly smaller number of vectors such that the relevant calculations are well approximated. Hence, an important motivation for the proposed method can be given as follows. The present scenario could be formulated alternatively as an extreme reduced set problem, where the kernel SVM was first solved obtaining the support vectors and then this set of support vectors were reduced to a single vector (similar to the proposed) $\mathbf{w}$, *i.e.*, optimize for $\mathbf{w}$ such that

$$\sum_{i=1}^{N} y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) \approx k(\mathbf{w}, \mathbf{x}),\ \forall\ \mathbf{x} \in \mathbb{R}^D, \tag{29}$$

where the α_i's are the optimal dual variables, which are strictly positive only for the N_s support vectors among training vectors. However, this would involve solving two optimization problems, one for obtaining the support vectors (and optimal dual variables) and then second for solving the reduced (singleton) set problem. Also, since we are eventually interested in the primal objective (and the kernel SVM is usually solved in the dual) it is known that there is no guarantee that an approximate dual solution will give a good approximate primal solution and, hence, solving the primal directly is beneficial [62]. Hence we propose instead to integrate approximations motivated by reduced set formulations directly into the original optimization of regularized loss minimization.

2.4 Learning using SGD

We learn the non-linear SVM (nSVM) directly, without the kernel trick, by optimizing the primal (14) w.r.t.

Algorithm 1 SGD based learning of nSVM

Input: $(\mathbf{x}_i, y_i)_{i=1:N}$, λ, m and $k \in \{k_h, k_c, k_{re}, k_{rc}\}$
Initialize: $\mathbf{w}$, b and r
for iter $= 1, \ldots, 100$ **do**
  $\sigma \leftarrow$ random_shuffle($[\![1, N]\!]$)
  for $j = 1, \ldots, N$ **do**
    $i = \sigma(j)$
    if $y_i(k(\mathbf{w}, \mathbf{x}_i) + b) < m$ **then**
      $w_d \leftarrow w_d + r y_i \nabla_{w_d} k(\mathbf{w}, \mathbf{x}_i), \ \forall\ d$
      $b \leftarrow b + r y_i$
    end if
    $w_d \leftarrow \frac{w_d}{|w_d|} \ \max[0, |w_d| - r\lambda \nabla_{w_d} k(\mathbf{w}, \mathbf{w})], \ \forall\ d$
  end for
  if iter $= 50$ **do** $\quad r \leftarrow r/10 \quad$ **end if**
end for

$(\mathbf{w}, b) \in \mathbb{R}^{D+1}$. We follow [23] and use stochastic gradient descent (SGD). The required gradients for the regularization and score functions are summarized in Table 1. Following previous works [60], we use a small but constant learning rate r, which we reduce by a factor of 10 in the mid iteration as it leads to a smoother convergence due to annealing (Fig. 6). In the present paper, since we work with bag-of-features histograms, the training examples are non-negative ($x_d \geq 0, \ \forall\ d$), while $\mathbf{w}$ in general is not. When making the update, we do not allow zero-crossing since the regularization term is in general not differentiable at zero. The full learning algorithm used to train models in the present paper is given in Algorithm 1.

3 Experimental Results

We use the following publicly available datasets to evaluate the various aspects of the proposed method. Fig. 3 gives some examples of the kind of images the test databases contains.

Flickr Materials dataset[2] [26] is a challenging dataset with 10 material categories, *e.g.* glass, leather, fabric. The dataset was created manually by downloading images from Flick.com while ensuring large variations *e.g.* in illumination, color, composition and texture, making the dataset very challenging. The evaluation is done with 50 images per class for training and 50 for testing.

MIT indoor scenes dataset[3] [4] contains 67 indoor scene categories, *e.g.* inside airport, inside church, kitchen. There are a total of 15620 images with each class containing at least 100 images. The evaluation is done using 80 training images and 20 test images per class.

Human Attributes (HAT) dataset[4] [27] contains 9344 images of humans with 27 different attributes, *e.g.* small kid, running, wearing jeans, crouching. The dataset was constructed by automatically downloading images based on manually specified human centered queries and then running a state-of-the-art human detector [32] and having the false positives manually pruned. The evaluation is done using the provided split of 7000 training and validation images and 2344 test images.

Pascal VOC 2007 dataset[5] [5] is composed of images containing 20 different categories of objects, *e.g.* horse, airplane, bottle, cow. The images were downloaded from the internet and annotated for the objects. It has been a standard benchmark dataset for image classification, segmentation and object detection. It has 5011 images for training and validation and 4952 images for testing. We report results on the image classification task.

Performance measure. For each dataset, we train a one vs. all binary classifier for each class and report the performance as the average precisions (AP) for each class and the mean average precision (mAP) over all the classes.

Implementation details. We use dense SIFT features extracted at 8 scales separated by a factor of 1.2, with step size 3 pixels. We use two types of bag-of-features [12], one based on gray SIFT [63] and other based on opponent SIFT, which has been shown to be a good color descriptor [25]. We use k-means to learn a codebook of size 4096 (unless otherwise specified) and do approximate nearest neighbor based hard quantization. We do a three level SPM with $1 \times 1, 2 \times 2$ and 3×1 partitions. We use `vlfeat` library [64] for SIFT, k-means and ANN. We fixed the parameters to $\lambda = 10^{-4}, m = 0.05$, and initialize $\mathbf{w} = 0$, $b = -1$ for the RBF kernels and $b = 0$ for others, for all the experiments. The Fisher Vector is our implementation of [7] (following [24]) in C++ and is called via the mex interface of MATLAB. All times reported are for the computations only, *i.e.*, with all required data completely in memory, and are on a single core/thread of a workstation with an Intel Xeon X5650 2.67 GHz processor running GNU-linux.

In the following, we denote the proposed nonlinear SVM as nSVM and use subscripts 'h', 'c', 'rc', 're' (with k) for Histogram intersection, χ^2, RBF-χ^2 and RBF-Euclidean kernels respectively.

3.1 Free parameter m and sensitivity to m and λ

In the proposed optimization Eq. 14 we introduced a free parameter m instead of the usual unit margin. The motivation for doing so was as follows. Consider Histogram Intersection kernel for instance. Since the scoring function(s) is defined as $k_h(\mathbf{w}, \mathbf{x})$, the maximum (minimum) score achievable is 1 (-1) in the case when $\mathbf{w} = \mathbf{x}$ ($\mathbf{w} = -\mathbf{x}$) (as the vectors $\mathbf{x}$ are ℓ_1 normalized). Hence almost all the vectors will have absolute scores less than 1, *i.e.*, all of them will be inside margin for the usual case of $m = 1$. It is highly unlikely (even

2. http://people.csail.mit.edu/celiu/CVPR2010/FMD/
3. http://web.mit.edu/torralba/www/indoor.html
4. https://jurie.users.greyc.fr/datasets/hat.html
5. http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/

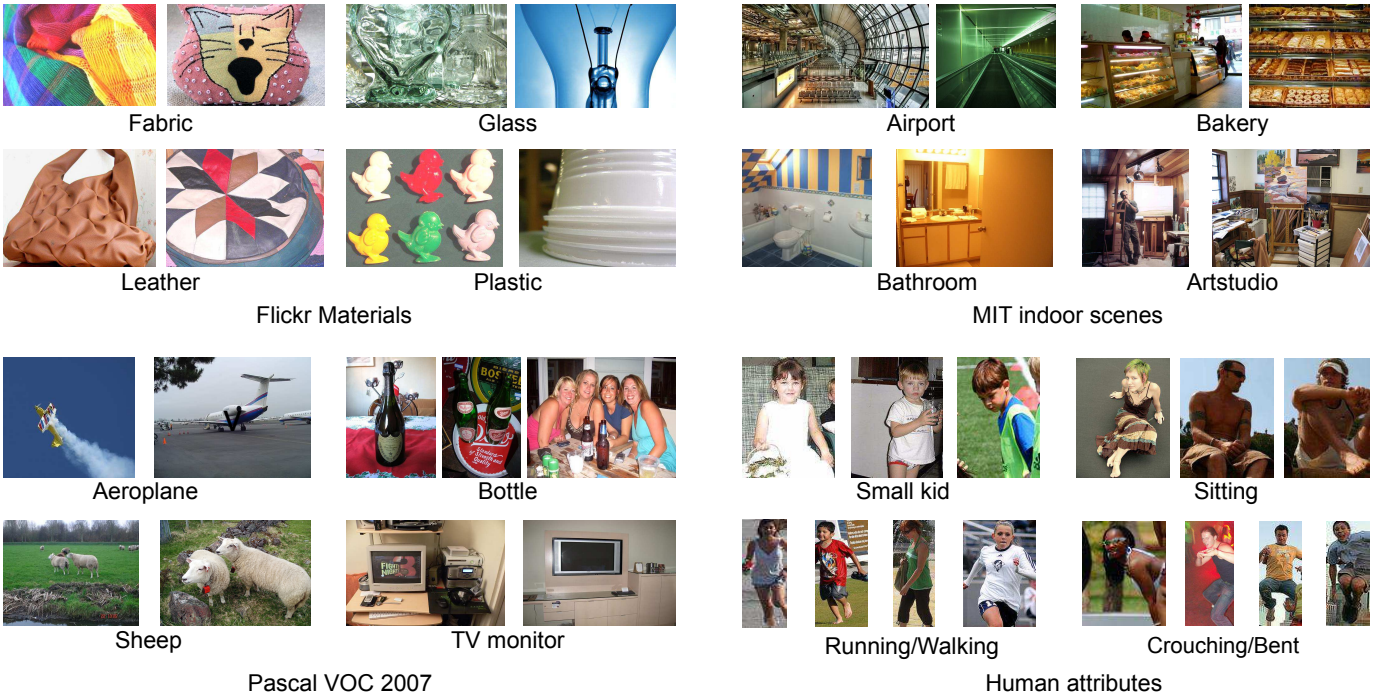

Fig. 3. Example labeled images from the datasets used for experimental evaluation.

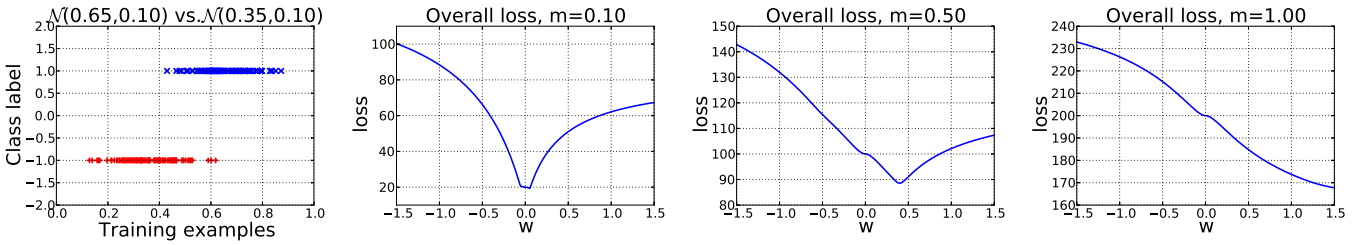

Fig. 4. The loss function as a function of w for different values of m, corresponding to χ^2 kernel for synthetic 1-D examples generated randomly from two normal distributions $\mathcal{N}(0.65, 0.1)$ vs. $\mathcal{N}(0.35, 0.1)$. The leftmost figure plots (x,y) for the randomly sampled points and the next three figures plot the loss function for $m = 0.1, 0.5, 1.0$. See § 3.1 for discussion.

for linear SVM) that *all* of the training examples are support vectors (being inside the margin). Empirically we found in preliminary experiments that with $m = 1$ the method doesn't work. Changing m changes the optimization function and we visualized this in 1-D by generating 100 points randomly from two normal distributions $\mathcal{N}(0.65, 0.1)$ vs. $\mathcal{N}(0.35, 0.1)$ and visualizing the loss function, shown in Fig. 4 (for χ^2 kernel). We see that the loss function changes as we vary m, since the learning focuses in different sets of 'hard examples' which are more likely to become support vectors. On real image data, we expected the learning to focus on harder examples with $m \ll 1$ (as in higher dimensional space, high overlap between $\mathbf{w}$ and all $\mathbf{x}$ is unlikely) and found empirically that smaller values of m work better. The method is not very sensitive to m, once we were in a good range (by preliminary experiment on smaller subset of training set) we could fix m for all experiments.

Fig. 6 shows t ypical test average precision (AP) vs. iterations curves for the proposed learning algorithm with different settings of the two free parameters m and λ. The method converges for a range of λ and m parameters. We found that having a higher rate r initially and then annealing by decreasing the learning rate midway was helpful for convergence, notice the convergence before and after iteration 50.

3.2 Comparison with kernel SVM and explicit feature maps

Tab. 2 gives the performance of the proposed nSVM along with that of the traditional kernel SVM classifier on the Pascal VOC 2007 [5] dataset, for the different kernels. We use `libsvm` library [65] with precomputed kernels for the kernel SVM results. We see that the proposed method achieves slightly lower results as the kernel SVM. For reference, linear SVM obtains 40.1 mAP here. However, the test times for the proposed method is orders of magnitude faster. Instead of comparing test

	k_h	k_c	k_{rc}	k_{re}
Kernel SVM (libsvm)	54.9	55.0	55.5	42.7
nSVM (present)	54.2	53.9	55.2	40.9

TABLE 2
Performance (mAP) of the proposed nonlinear SVM and of kernel SVM on the Pascal VOC 2007 [5] dataset with Histogram intersection (k_h), χ^2 (k_c), RBF-χ^2 (k_{rc}) and RBF-Euclidean (k_{re}) kernels. The performance of linear SVM in same settings is 40.1 mAP.

	Time		Memory	
	Secs	Speedup	Kb	Reduction
Feature maps [22]	3.8	1 (ref)	448	1 (ref)
nSVM (present)	0.2	19×	64	7×

TABLE 3
Testing time, for the test set (with about 5000 images) of Pascal VOC 2007 dataset [5] (with a typical class model, averaged over 10 runs) and memory usage (for keeping model in memory) for the proposed method and the explicit feature maps method of Vedaldi and Zisserman [22] (with Histogram intersection kernel).

times with kernel SVM we compare them with competing methods in the following section.

We compare with a closely related, recently proposed method of explicit feature mapping (FM) by Vedaldi and Zisserman [22] which computes a finite dimensional map approximating the kernel. Such mappings thus enable us to compute linear classifiers in the mapped space. It was shown by Vedaldi and Zisserman [22] that this feature mapping obtains better results that the one by Maji and Berg [20].

We use `vlfeat` library [64] to compute the FM corresponding to Vedaldi and Zisserman's method [22] for the histogram intersection and χ^2 kernels. We use `liblinear` [44] (with ℓ_2 regularized ℓ_1 loss option) to learn SVM with the feature mapped vectors. FM was shown to be more than three orders faster than the kernel SVM (there are $O(10^3)$ support vectors and the scoring a new image/vector requires computing the kernel with each of them). We use the parameter values which gave the best performance for FM *i.e.*, map the original d dimensional BoF vectors to $7d$ dimensional feature space and do classification there. Since we did the experiments using all the features in memory, limited by the RAM of the system, we used a codebook size of 1024 (instead of 4096 as everywhere else) for all three methods.

Fig. 5 shows the per class performances of our method vs. kernel SVM (using `libsvm` [65] with precomputed kernels) and FM [22], on the Pascal VOC 2007 [5] and Human Attributes (HAT) [27] datasets. We get similar performance, on average, compared to FM for both the datasets and both histogram intersection and χ^2 kernels. We conclude that our method for learning a classifier directly in original space achieves essentially similar performance as the explicit feature maps method of Vedaldi and Zisserman [22].

Tab. 3 summarizes the test time and memory usage comparisons. While our method performs a linear scan on the d-dimensional features to calculate the test score (by computing the kernel between $\mathbf{w}$ and the test vector), for explicit feature maps we have to, first, compute the mapping to $7d$ space and then compute a dot product in that space. Hence the model is $7\times$ bigger for explicit feature map compared to our method and (empirically) our method is about $19\times$ faster than explicit feature maps with linear SVM (the time is only due to classifier score computations and excludes the bag-of-features construction time for both methods). The training is also fast, *e.g.* it takes about 45 secs to train a model for one class of Pascal VOC 2007 dataset. We resorted to a conservative training strategy with multiple passes over the data and our training time can be arguably improved quite a bit.

3.3 Performance vs. complexity trade-off

As the RBF-χ^2 kernel usually performs the best among all kernels (see for instance Tab. 2), we report results based on RBF-χ^2 kernel in the following. We denote 'nSVM' the corresponding non-linear SVM learnt with the proposed method, with suffix 'GRAY' or 'COL' based on the type of SIFT, gray or color, that is used. When we combine two methods, denoted by '+', we do so by a simple late fusion (averaging) of the confidence scores of the individual classifiers.

Table 4 shows the performances of nSVMs with RBF-χ^2 kernel trained on bag-of-features based on gray SIFT and opponent SIFT features [25], along with those of Fisher Vectors [7] and the various combinations of the methods using late fusion on the four datasets: Flickr Materials [26], Human Attributes [27], Indoor Scenes [4], and Pascal VOC 2007 [5].

The fastest method, nSVM with gray SIFT features, achieves competitive performance on the four datasets, *i.e.*, 89%, 85%, 89% and 90% respectively of the performance of state-of-the-art FV method [7] while being $34\times$ ($75\times$) faster in testing (encoding) times and at least $10\times$ more space efficient (this is more in practice as, while FVs were negligibly sparse, BoFs with spatial pyramids had up to 40% zeros). We emphasize that efficiency comparisons based only on encoding are equally, if not more, relevant as the local feature extraction is likely to be implemented on fast or even dedicated hardware in many consumer devices, and hence is likely to become relatively negligible for all the methods.

Although nSVM with color features performs relatively poorly alone, when combined with nSVM with gray features it achieves significantly more than either at 102%, 93%, 96% and 95% of the performance of

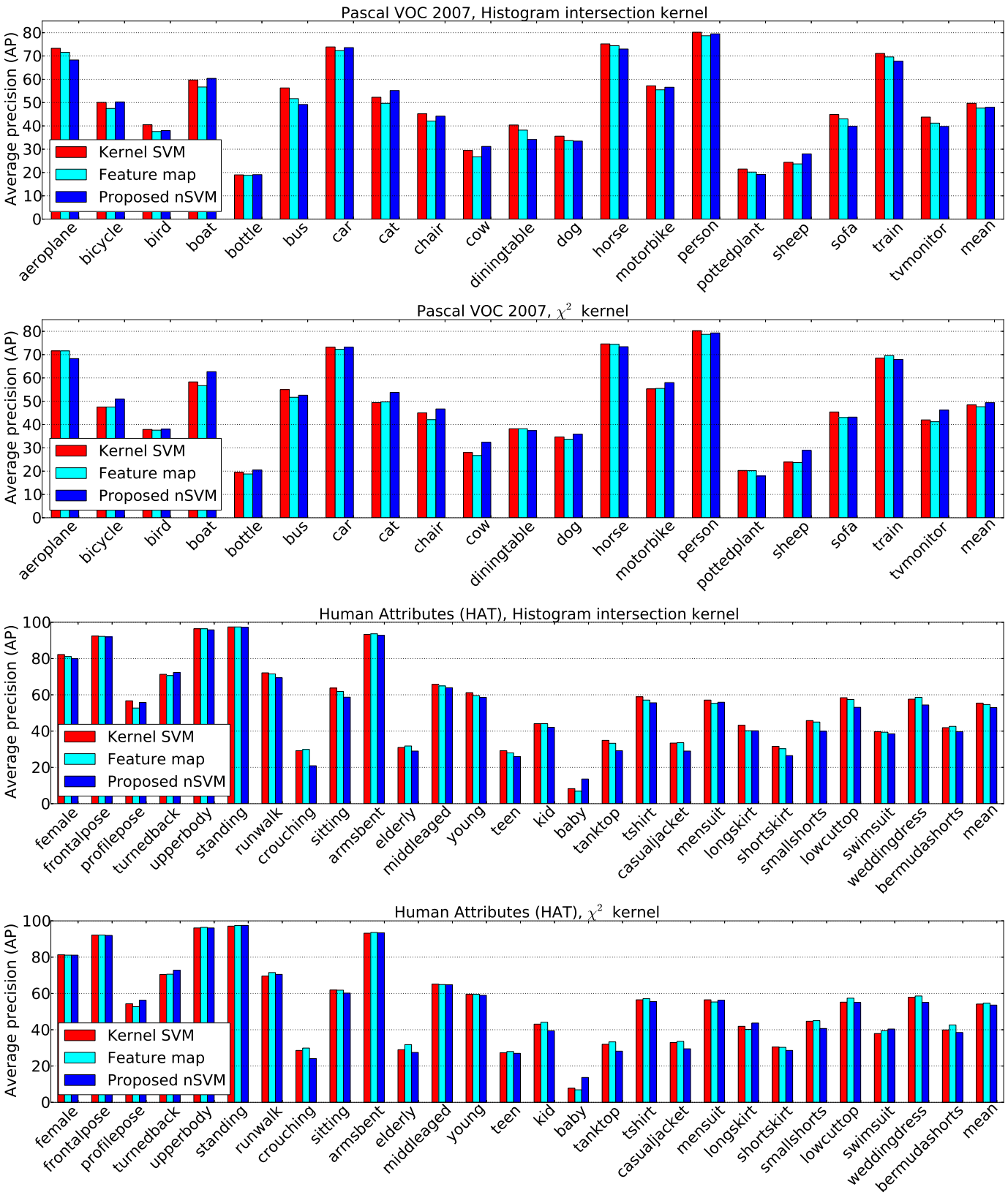

Fig. 5. The average precisions for different classes (and the mean AP) of the Human Attributes (HAT) dataset [27] and Pascal VOC 2007 [5] dataset (image classification task) for (i) Kernel SVM (`libsvm` [65]), (ii) the explicit feature mapping of Vedaldi and Zisserman [22] (iii) the proposed method (nSVM), for histogram intersection and χ^2 kernels.

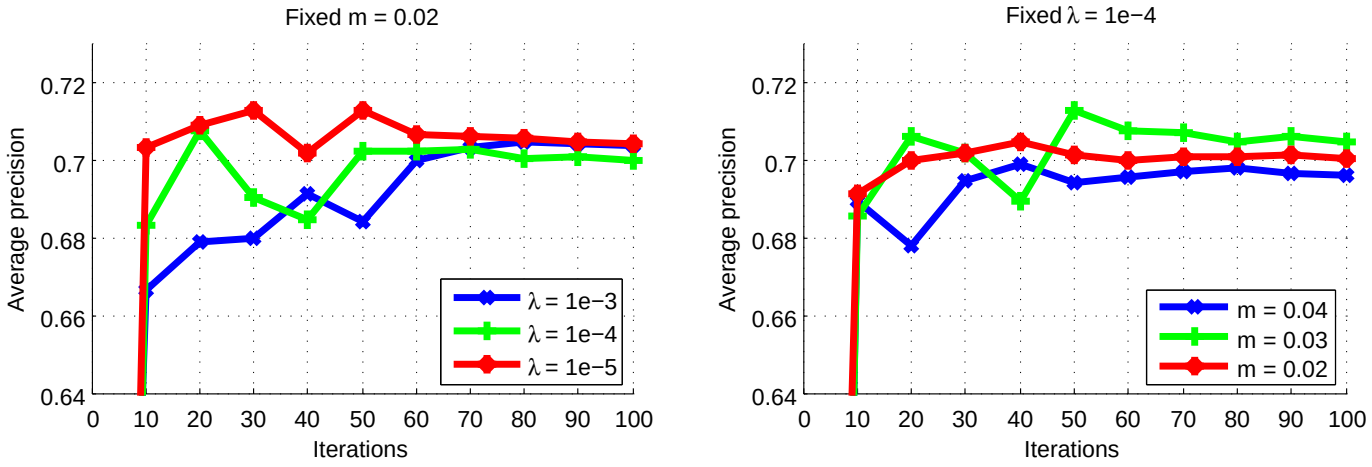

Fig. 6. The average precisions for different values of (left) regularization parameter λ and (right) hinge loss parameter m, for a typical convergence of the proposed method.

	nSVM$_{GRAY}$	nSVM$_{COL}$	nSVM$_{GRAY}$ +nSVM$_{COL}$	Fisher Vector	Fisher Vector +nSVM$_{GRAY}$	Fisher Vector +nSVM$_{GRAY}$ +nSVM$_{COL}$
Flickr Materials [26]	50.2	51.9	57.3	56.1	56.7	59.7
MIT Scenes [4]	53.5	50.5	58.0	62.7	63.3	64.6
Human Attributes [27]	57.2	59.2	61.5	64.2	65.0	65.6
Pascal VOC 2007 [5]	55.2	50.8	58.6	61.4	62.4	63.6
Feature dimension	32,768	32,768	65,536	327,680	360,448	393,216
Memory reduction*	10×	10×	5×	1 (ref)	0.91×	0.83×
Encoding time	0.27s	0.34s	0.61s	20.16s	20.43s	21.04s
Speed-up	75×	59×	33×	1 (ref)	0.99×	0.96×
Full test time	0.60s	1.31s	1.92s	20.49s	20.76s	22.68s
Speed-up	34×	16×	11×	1 (ref)	0.99×	0.90×

TABLE 4

The performances and the test time complexities of the proposed method with RBF-χ^2 kernel (denoted as nSVM) with BoF based on gray (subscript GRAY) or color (subscript COL) SIFT features and the Fisher Vector [7] method. The '+' signifies combination of the classifiers by late fusion of their individual scores. The times are for a typical image with about 44k SIFT features on a single core/thread of an Intel Xeon X5650, 2.67Ghz processor. 'Full test time' amounts to the complete chain, SIFT extraction, encoding/pooling and final classification, while 'Encoding time' refers to encoding step only, after SIFT features have been extracted (*The space complexity comparison does not take into account the sparsities of the representations, usually BoF are much sparser than Fisher Vectors).

Fisher Vectors while being $11\times$ ($33\times$) faster in testing (encoding) times and at least $5\times$ more space efficient. The full test time, as reported in Tab. 4, is the end-to-end time *i.e.,* from raw image as input to its test score. It includes computation of all types of features (for the respective methods), their encoding/pooling and finally the test score computation. Hence while the proposed method pays additional cost of computing multiple features, due to inexpensive simple encoding combined with proposed efficient nonlinear classification, it is much faster overall. This shows that simple statistics of complementary features when used with nSVM lead to highly competitive performance on a budget.

On higher time complexities, adding the gray and color features based nSVM to the Fisher Vector consistently leads to improvements of up to 3.6 absolute mAP points. Combining nSVM with just gray features with the Fisher Vectors ('Fisher Vec. + nSVM$_{GRAY}$' in Table 4) only brings a modest improvement (0.6 to 1.0 absolute mAP points); this is in contrast with results in [7], where zeroth order statistics (equivalent to BoFs) did not add anything to the higher order statistics (Fisher Vectors) with linear classifiers. Although this small improvement is not very attractive in practice, recall that adding both gray and color features based nSVMs to FV does lead to consistently important improvements w.r.t. FV alone, over all four datasets, at a small cost in time and space complexities.

3.4 Comparison with the state-of-the-art

Tables 5, 6, 7 and 8 compare the best performing version of our method with existing methods on the four datasets (see Table 4 for the other versions of our approach).

On the Pascal VOC 2007 dataset [5] (Tables 4 and 5), in the original image classification challenge the winning entry used many different features with RBF-χ^2 kernels achieving 59.4 mAP. The kernels for the different features were combined with learnt weights for each class. Similarly, many other works have combined many features using, *e.g.* multiple kernel learning [66]. We achieve similar performance as Harzallah *et al.* [6], who used object detection, in addition, to improve the classification score, leading to a high performing but very slow method. Recently, Sanchez *et al.* [7] reported an mAP of 63.9 which is slightly better than our best result of 63.6. They use Fisher Vectors on both gray and color features and thus would be expected to be about $2\times$ slower than us. Also, note that the performance of FV on color features (52.6 mAP), as reported in [7], is comparable to the performance of proposed RBF-χ^2 nSVM with color SIFT features (50.8 mAP) and significantly lower than nSVM with both gray and color SIFT features (58.6 mAP), while being about an order of magnitude slower (Table 4).

On the Indoor Scenes dataset [4] (Tables 4 and 8), nSVM with gray SIFT BoF performs (53.5 mAP) similar to more complex locality constrained linear coding (LLC) with max pooling [18] (53.0 mAP). Our best result (64.6 mAP) is better than that of a recent method which learns discriminative parts and combines them with Fisher Vectors [3] (63.2 mAP).

On the Human Attributes dataset [27] (Tables 4 and 6), nSVM with gray SIFT BoF performs competitively at 57.2 mAP w.r.t. current methods learning adaptive spatial partitions [27] (53.8 mAP), and learning class-wise or global part dictionaries, *i.e.,* [10] (58.7 mAP) and [9] (59.3 mAP). Color SIFT BoF works slightly better with nSVM for this dataset (59.2) while the best result obtained clearly outperforms all of the existing methods with 65.6 mAP.

On the Flickr Materials dataset [26] (Tables 4 and 7), previous works report mean class accuracy (mAcc), and we do a simple winner-takes-all voting, on confidence scores for the binary classifiers for each class, to calculate mAcc for the proposed method. At 57.6 mAcc (corresponding to our best 59.7 mAP in Table 4) we outperform methods based on different features (48.2 mAcc [67]), descriptors (54.0 mAcc [68]) and classification methods (55.8 [69] and 44.6 [70] mAcc).

4 Conclusion

Making non-linear classification efficient is advantageous for many applications specially with large number of images and categories, *e.g.* large scale classification, and with limited computing resources, *e.g.* in consumer devices like cameras or smart phones.

In the present paper we proposed a method for learning non-linear SVM, corresponding to the four kernels popular in computer vision, directly in the original space, *i.e.,* without using the kernel trick or mapping the features explicitly to high dimensional space corresponding to the kernel. We formulated the non-linear optimization in the original space which corresponds to the linear optimization problem in the high dimensional feature space. We showed experimentally that a stochastic algorithm with subgradients works well in practice. Compared to a recent method for making non linear classification efficient, the proposed method is $19\times$ faster and requires $7\times$ less memory.

We analysed empirically the trade-off between encoder and classifier complexity and strength. While, on one hand we have simple counting/histogram statistics of local features, namely bag-of-features (BoF), with non-linear SVM (nSVM), on the other, we have complex state-of-the-art Fisher vector (FV) encoding with linear SVM. We showed that BoF based on gray SIFT features with the proposed nSVM leads to very fast classifier which can achieve up to 90% of the performance of state-of-the-art FV encoding method while being $34\times$ ($75\times$) faster in testing (encoding) times and more than $10\times$ more memory efficient. Further, adding color based BoF with nSVM leads to up to 96% performance of FV while being $11\times$ ($33\times$) faster in testing (encoding) times and more than $5\times$ more memory efficient. At last, combining the nSVM based system with FV leads to significant improvements (up to 3.6 absolute mAP points) at small space and computation costs.

Finally, we would like to point out that the BoF we used here is the simplest. The large body of existing works that improve BoF, *e.g.* by learning better quantizers [71], [72], [73], should increase the performance of our nSVM based systems further. We hope to explore this in the future.

Acknowledgments

References

[1] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006. 1, 2

[2] G. Sharma, F. Jurie, and C. Schmid, "Discriminative spatial saliency for image classification," in *CVPR*, 2012. 1, 2

[3] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *CVPR*, 2013. 1, 3, 12, 13

[4] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *CVPR*, 2009. 1, 2, 7, 9, 11, 12, 13

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2007. 1, 2, 7, 8, 9, 10, 11, 12, 13

[6] H. Harzallah, F. Jurie, and C. Schmid, "Combining efficient object localization and image classification," in *ICCV*, 2009. 1, 2, 12, 13

[7] J. Sanchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *IJCV*, 2013. 1, 2, 3, 7, 9, 11, 12, 13

Method	mAP	Remarks
Challenge winners	59.4	Several features with learnt weights
van Gemert *et al.* [8]	60.5	Several color (and gray) features
Yang *et al.* [66]	62.2	Mult kernel learning
Chatfield *et al.* [24]	61.7	Fisher Vectors (FV)
Sanchez *et al.* [7]	63.9	FV gray and color
Harzallah *et al.* [6]	63.5	Object detection
Present	63.6	

TABLE 5
Comparison with existing methods on the Pascal VOC 2007 dataset [5].

Method	mAcc	Remarks
Liu *et al.* [70]	44.6	Bayesian learning with several features
Timofte and Van Gool [69]	55.8	Collaborative representation
Liu *et al.* [67]	48.2	Sorted random projection features
Hu *et al.* [68]	54.0	Kernel descriptors
Present	57.6	Corresponding to 59.7 mAP

TABLE 7
Comparison with existing methods (mean class accuracy) on the Flickr Materials dataset [26].

Method	mAP	Remarks
Sharma and Jurie [27]	53.8	Learnt spatial partitions
Sharma *et al.* [10]	58.7	Many parts learnt for each class
Joo *et al.* [9]	59.3	Many parts shared between classes
Present	65.6	

TABLE 6
Comparison with existing methods on the dataset of Human Attributes (HAT) [27].

Method	mAP	Remarks
Wang *et al.* [18]	53.0	Locality constrained linear coding (LLC)
Juneja *et al.* [3]	43.5	Bag of parts (BoP)
Juneja *et al.* [3]	63.2	Fisher Vectors + BoP
Sanchez *et al.* [7]	61.1	Fisher Vectors (as reported in [3])
Present	64.6	

TABLE 8
Comparison with existing methods on the MIT Indoor Scenes dataset [4].

[8] J. van Gemert, J.-M. Geusebroek, C. Veenman, and A. Smeulders, "Kernel codebooks for scene categorization," in *ECCV*, 2008. 1, 13

[9] J. Joo, S. Wang, and S.-C. Zhu, "Human attribute recognition by rich appearance dictionary," in *ICCV*, 2013. 1, 12, 13

[10] G. Sharma, F. Jurie, and C. Schmid, "Expanded parts model for human attribute and action recognition in still images," in *CVPR*, 2013. 1, 12, 13

[11] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *ICCV*, 2003. 1, 2

[12] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Intl. Workshop on Stat. Learning in Comp. Vision*, 2004. 1, 2, 7

[13] Y.-L. Boureau, N. Le Roux, F. Bach, J. Ponce, and Y. LeCun, "Ask the locals: multi-way local pooling for image recognition," in *ICCV*, 2011. 1, 3

[14] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *CVPR*, 2010. 1, 3

[15] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR*, 2009. 1, 3

[16] S. Gao, I. W. Tsang, L.-T. Chia, and P. Zhao, "Local features are not lonely–laplacian sparse coding for image classification," in *CVPR*, 2010. 1, 3

[17] Y. Huang, K. Huang, Y. Yu, and T. Tan, "Salient coding for image classification," in *CVPR*, 2011. 1, 3

[18] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*, vol. 0, 2010. 1, 3, 12, 13

[19] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *CVPR*, 2008. 1, 3, 6

[20] S. Maji and A. C. Berg, "Max-margin additive classifiers for detection," in *ICCV*, 2009. 1, 3, 9

[21] F. Perronnin, J. Sanchez, and Y. Liu, "Large-scale image categorization with explicit data embedding," in *CVPR*, 2010. 1, 3, 6

[22] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *PAMI*, vol. 34, no. 3, pp. 480–492, 2012. 1, 3, 6, 9, 10

[23] G. Sharma and F. Jurie, "A novel approach for efficient SVM classification with histogram intersection kernel," in *BMVC*, 2013. 1, 7

[24] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *BMVC*, 2011. 1, 3, 7, 13

[25] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *PAMI*, vol. 32, no. 9, pp. 1582–1596, 2010. 1, 7, 9

[26] L. Sharan, R. Rosenholtz, and E. Adelson, "Material perception: What can you see in a brief glance?" *Journal of Vision*, vol. 9, no. 8, pp. 784–784, 2009. 2, 7, 9, 11, 12, 13

[27] G. Sharma and F. Jurie, "Learning discriminative representation image classification," in *BMVC*, 2011. 2, 7, 9, 10, 11, 12, 13

[28] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005. 2

[29] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001. 2, 3, 4, 6

[30] J. Krapac, J. Verbeek, and F. Jurie, "Modeling spatial layout with Fisher vectors for image categorization," in *ICCV*, 2011. 2

[31] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *ICCV*, 2009. 2

[32] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *PAMI*, vol. 32, no. 9, pp. 1627–1645, 2010. 2, 7

[33] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Intl. Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000. 2

[34] F. Perronnin, Z. Akata, Z. Harchaoui, and C. Schmid, "Towards good practice in large-scale learning for image classification," in *CVPR*, 2012. 2

[35] C. J. C. Bruges, "Simplified suport vector decision rules," in *ICML*, 1996. 3, 6
[36] T. Downs, K. E. Gates, and A. Masters, "Exact simplification of support vector solutions," in *JMLR*, 2001. 3
[37] Y.-J. Lee and O. L. Mangasarian, "RSVM: Reduced support vector machines," in *SIAM Conf. on Data Mining*, 2001. 3
[38] E. Osuna and F. Girosi, "Reducing the run-time complexity of support vector machines," in *Proceedings of the International Conference on Pattern Recognition*, 1998. 3
[39] B. Scholkopf, P. Simard, V. Vapnik, and A. Smola, "Improving the accuracy and speed of support vector machines," in *NIPS*, 1997. 3
[40] J. Thorsten, "Training linear SVMs in linear time," in *KDD*, 2006. 3
[41] Y. Singer and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for SVM," in *ICML*, 2007. 3
[42] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning." in *NIPS*, 2008. 3
[43] K.-W. Chang, C.-J. Hsieh, and C.-J. Lin, "Coordinate descent method for large-scale l2-loss linear support vector machines," *Journal of Machine Learning Research*, vol. 9, pp. 1369–1398, 2008. 3
[44] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008. 3, 9
[45] C. K. Williams and M. Seeger, "The effect of the input density distribution on kernel-based classifiers," in *ICML*, 2000. 3
[46] ——, "Using the nystrom method to speed up kernel machines," in *NIPS*, 2001. 3
[47] A. Smola and B. Scholkopf, "Sparse greedy matrix approximation for machine learning," in *ICML*, 2000. 3
[48] S. Fine and K. Scheinberg, "Efficient SVM training using low-rank kernel representations," in *JMLR*, 2001. 3
[49] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *NIPS*, 2007. 3
[50] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *NIPS*, 2009. 3
[51] L. Liu, L. Wang, and X. Liu, "In defense of soft-assignment coding," in *ICCV*, 2011. 3
[52] F. Perronnin, C. R. Dance, G. Csurka, and M. Bressan, "Adapted vocabularies for generic visual categorization," in *ECCV*, 2006. 3
[53] J. C. van Gemert, C. J. Veenman, A. W. Smeulders, and J.-M. Geusebroek, "Visual word ambiguity," *PAMI*, vol. 32, no. 7, pp. 1271–1283, 2010. 3
[54] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *ICCV*, 2005. 3
[55] T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja, "Low-rank sparse coding for image classification," in *ICCV*, 2013. 3
[56] X. Zhou, K. Yu, T. Zhang, and T. S. Huang, "Image classification using super-vector coding of local image descriptors," in *ECCV*, 2010. 3
[57] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *NIPS*, 1998. 3
[58] J.-Y. Kwok and I. W. Tsang, "The pre-image problem in kernel methods," *IEEE Transactions on Neural Networks*, vol. 15, no. 6, pp. 1517–1525, 2004. 3
[59] S. Mika, B. Scholkopf, A. J. Smola, K.-R. Muller, M. Scholz, and G. Ratsch, "Kernel PCA and de-noising in feature spaces." in *NIPS*, 1998. 3, 4, 6
[60] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Good practice in large-scale learning for image classification," *PAMI*, 2013. 3, 7
[61] J. Sanchez, F. Perronnin, and Z. Akata, "Fisher vectors for fine-grained visual categorization," in *CVPR Workshops*, 2011. 3
[62] O. Chapelle, "Training a support vector machine in the primal," *Neural Computation*, vol. 19, no. 5, pp. 1155–1178, 2007. 3, 6
[63] D. Lowe, "Distinctive image features form scale-invariant keypoints," *Intl. Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. 7
[64] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008. 7, 9
[65] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. 8, 9, 10
[66] J. Yang, Y. Li, Y. Tian, L. Duan, and W. Gao, "Group sensitive multiple kernel learning for object categorization," in *ICCV*, 2009. 12, 13
[67] L. Liu, P. Fieguth, D. Clausi, and G. Kuang, "Sorted random projections for robust rotation-invariant texture classification," *Pattern Recognition*, vol. 45, no. 6, pp. 2405–2418, 2012. 12, 13
[68] D. Hu, L. Bo, and X. Ren, "Toward robust material recognition for everyday objects." in *BMVC*, 2011. 12, 13
[69] R. Timofte and L. J. Van Gool, "A training-free classification framework for textures, writers, and materials." in *BMVC*, 2012. 12, 13
[70] C. Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz, "Exploring features in a bayesian framework for material recognition," in *CVPR*, 2010. 12, 13
[71] J. Krapac, J. Verbeek, and F. Jurie, "Learning tree-structured descriptor quantizers for image categorization," in *BMVC*, 2011. 12
[72] S. Lazebnik and M. Raginsky, "Supervised learning of quantizer codebooks by information loss minimization," *PAMI*, vol. 31, no. 7, pp. 1294–1309, 2009. 12
[73] F. Moosmann, E. Nowak, and F. Jurie, "Randomized clustering forests for image classification," *PAMI*, vol. 30, no. 9, pp. 1632–1646, 2008. 12

Gaurav Sharma is currently a post-doctoral researcher at Technicolor, France. He holds an Integrated M.Tech. (5 years programme) in Mathematics and Computing from the Indian Institute of Technology (IIT) Delhi and a PhD in Applied Computer Science from INRIA (LEAR team) and the University of Caen, France. His primary research interest lies in Machine Learning applied to Computer Vision tasks such as image classification, object recognition and facial analysis.

Frederic Jurie is a professor at the French University of Caen (GREYC - CNRS UMR6072) and associate member of the INRIA-LEAR team. His research interests lie predominately in the area of Computer Vision, particularly with respect to object recognition, image classification and object detection.

Patrick Perez is a Distinguished Scientist and a fellow at Technicolor. He received his engineering degree from Ecole Centrale Paris and PhD degree from University of Rennes. His prior employers include Microsoft Research Cambridge and INRIA. He holds or has held editorial positions for IJCV, TIP and PAMI and has served as area chair for ECCV and ICCV. He is interested in a wide range of theoretical and applied aspects of Machine Learning and Computer Vision.