Learning Non-linear SVM in Input Space for Image Classification

Gaurav Sharma, Frédéric Jurie, Patrick Pérez

Noname manuscript No.
(will be inserted by the editor)

Learning Non-linear SVM in Input Space for Image Classification

Gaurav Sharma · Frédéric Jurie · Patrick Pérez

Received: date / Accepted: date

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.

G. Sharma
GREYC CNRS UMR 6072
Universite de Caen Basse-Normandie, France
E-mail: grvsharma@gmail.com

F. Jurie
GREYC CNRS UMR 6072
Universite de Caen Basse-Normandie, France
E-mail: frederic.jurie@unicaen.fr

P. Pérez
Technicolor
E-mail: patrick.perez@technicolor.com

1 Introduction

Image classification is one of the central problems of computer vision. Recent works have addressed various image domains, *e.g.* outdoor scenes (Lazebnik et al, 2006; Sharma et al, 2012), indoor scenes (Juneja et al, 2013; Quattoni and Torralba, 2009), object images (Everingham et al, 2007; van Gemert et al, 2008; Harzallah et al, 2009; Sanchez et al, 2013), human attributes (Joo et al, 2013; Sharma et al, 2012, 2013). 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 (2003) and Csurka et al (2004) 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 (2006) 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 (Everingham et al, 2007; Harzallah et al, 2009; Lazebnik et al, 2006). However, exploiting 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 (Boureau et al, 2010, 2011; Gao et al, 2010; Huang et al, 2011; Wang et al, 2010; Yang et al, 2009b) gives better results with inexpensive linear SVM. On the other hand, in parallel, methods were proposed to make nonlinear classifiers efficient (Maji and Berg, 2009; Maji et al, 2008; Perronnin et al, 2010; Vedaldi and Zisserman, 2012). Balancing this trade-off between encoder and classifier complexity has remained an important question for image classification.

More recently, convolutional neural networks (CNN) based features were shown to perform very well for large scale image classification tasks, by Krizhevsky et al (2012). Parameters learnt from the large scale datasets were shown to be transferable to mid scale datasets *e.g.* by Oquab et al (2014). Further, such CNN features were also used as local features, and were encoded and pooled similar to the traditional local features, for image classification by Liu et al (2014) and for texture classification by Cimpoi et al (2014). This kept the encoding and pooling methods relevant, albeit replaced the traditional local features using either the last layer outputs or the convolutional filter banks learnt with the CNN architecture.

In the present paper, we stay in the traditional setup and 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 Sharma and Jurie (2013)), 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 (Everingham et al, 2007; van Gemert et al, 2008; Harzallah et al, 2009).

For the second contribution, we first note that, while the recently proposed complex encodings, *e.g.* Sanchez et al (2013); Wang et al (2010), 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 (Chatfield et al, 2011). 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 (Everingham et al, 2007; van de Sande et al, 2010).

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 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× faster for complete testing and 33× 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 (Sharan et al, 2009), (ii) MIT Indoor Scenes (Quattoni and Torralba, 2009), (iii) Human Attributes (Sharma and Jurie, 2011) and (iv) Pascal VOC 2007 (Everingham et al, 2007).

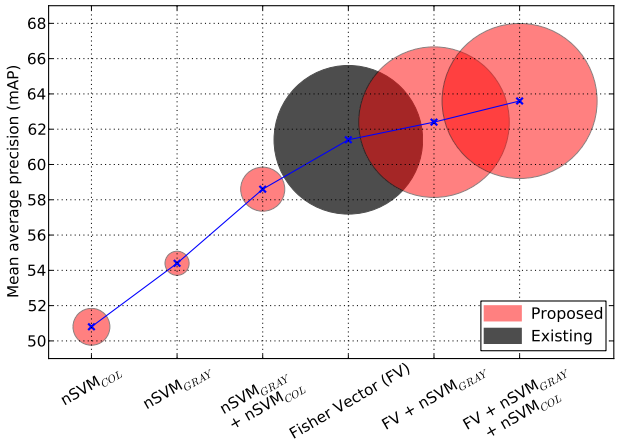

Fig. 1 The classification performances (mAP) on the Pascal VOC 2007 (Everingham et al, 2007) dataset: (i) the proposed nonlinear SVM method (denoted nSVM, in red/light) with complementary features, *i.e.,* gray (subscript GRAY) and color (subscript COL); (ii) Fisher Vector (Sanchez et al, 2013), an existing 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.

1.1 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 fea-

tures such as in BoF (Csurka et al, 2004) and HOG (Dalal and Triggs, 2005) and (ii) the use of margin maximizing classifiers such as SVMs (Scholkopf and Smola, 2001). Systems built on them have led to state-of-the-art performance on image classification (Krapac et al, 2011b; Lazebnik et al, 2006; Sharma et al, 2012) and object detection (Felzenszwalb et al, 2010; Harzallah et al, 2009; Vedaldi et al, 2009).

The standard formulation for learning classifiers is the SVM primal formulation (Eq. 2, see Scholkopf and Smola (2001) 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 high variability in viewpoint, lighting, pose, *etc.* and linear decision boundaries are not sufficient. Many competitive methods in image classification (Everingham et al, 2007; Lazebnik et al, 2006) and object detection (Harzallah et al, 2009; Vedaldi et al, 2009), thus, use non linear classifiers. Such non linear classifiers are obtained by using the *kernel trick* with the dual formulation 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.* Harzallah et al (2009); Lazebnik et al (2006); Vedaldi et al (2009).

While the dual formulation allows the learning of non linear decision boundaries, the computation of classifier score 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 become $O(D \times N_{sv})$[1] vs. $O(D)$ for the linear case (where $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$). 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 (Harzallah et al, 2009; Vedaldi et al, 2009) or large scale image classification (Perronnin et al, 2012) 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 (Csurka et al, 2004; Sivic and Zisserman, 2003) image representation (*e.g.* the high ranking entries of the PASCAL VOC 2007 competition (Everingham et al, 2007)). However, such classifiers incur a very high testing cost, as explained above. To address this problem of efficiency, approaches have primarily taken one of the following two directions.

1.2 Efficient classification.

Methods were proposed to reduce, primarily, the test time complexity of kernel SVMs. Bruges (1996) 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.* Downs et al (2001); Lee and Mangasarian (2001); Osuna and Girosi (1998); Scholkopf et al (1997).

Recently, Maji et al (2008) 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.* Bottou and Bousquet (2008); Chang et al (2008); Fan et al (2008); Singer and Srebro (2007); Thorsten (2006), has been quite popular recently. To this end, Williams and Seeger (2000, 2001), Smola and Scholkopf (2000) and Fine and Scheinberg (2001) used Nystrom's approximation. Perronnin et al (2010) applied Nystrom's approximation to each dimension for additive kernels for image classification. In a data independent way, Rahimi and Recht (2007) proposed to use random Fourier features to map the vectors and Raginsky and Lazebnik (2009) proposed to construct binary codes corresponding to shift invariant kernels. Maji and Berg (2009) approximated the feature map corresponding to the HI kernel while in a

[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 (Rubner et al, 2000)), in the present work we consider only those with linear complexities

more general approach, Vedaldi and Zisserman (2012) 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.

1.3 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 (van Gemert et al, 2010; Liu et al, 2011; Perronnin et al, 2006; Winn et al, 2005). Later works proposed even more sophisticated coding and pooling methods, *e.g.* sparse coding (Boureau et al, 2010, 2011; Gao et al, 2010; Huang et al, 2011; Wang et al, 2010; Yang et al, 2009b; Zhang et al, 2013) 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 (Zhou et al, 2010) and Fisher Vectors (Jaakkola and Haussler, 1998; Sanchez et al, 2013), which, used with linear SVMs, are part of the current state-of-the-art methods for image classification (Chatfield et al, 2011).

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 (Bruges, 1996; Kwok and Tsang, 2004; Mika et al, 1998). However, the objective 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 non-linear 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 (Chatfield et al, 2011), the complex encoding method of Fisher Vectors (Sanchez et al, 2013) was shown to be the state-of-the-art image representation, at that time, and has been applied to many classification tasks (Akata et al, 2013; Chatfield et al, 2011; Juneja et al, 2013; Sanchez et al, 2011). In our experiments, we show that the proposed 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).

1.4 Neural Networks.

Our work is also closely related to neural networks (Bishop, 1996), particularly kernel neural networks *e.g.* Bishop (1996); Park and Sandberg (1991); Rauber and Berns (2011); Xu et al (2001). Our method can be seen as a simple kernel neural network with one hidden unit with a parametrized kernel as the activation function. However, we arrive at this architecture from the perspective of non-linear classification using kernel SVMs. We, thus, keep the associated regularization which is otherwise a difficult and actively studied problem in neural networks, using *e.g.* regularization by weight decay, early stopping and training with transformed data (Bishop, 2006).

2 Approach

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

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

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` Fan et al (2008)) 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

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

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:

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

with

$$\phi : \mathbb{R}^D \rightarrow \mathcal{F} \tag{5}$$

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 Scholkopf and Smola (2001) for detailed discussion). Learning a kernel based non-linear SVM with the primal formulation, using similarly the kernel trick, is also possible (Chapelle, 2007) 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:

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

with l being the *hinge loss* function

$$l(y, \delta) = \max(0, 1 - y\delta) \tag{7}$$

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 (Mika et al, 1998; Scholkopf and Smola, 2001). 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.*,

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

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

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

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:

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

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

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

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

and hinge loss computations in the second term involve computing

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

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

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

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,

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

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

	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 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.

2.1.1 Histogram intersection kernel.

The generalized HI kernel is given by

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

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

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

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.,* $x_d \geq 0\ \forall,\ d = 1, \ldots, D$.

2.1.2 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 $x_d \geq 0$)

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

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

2.1.3 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

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

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** $r \leftarrow r/10$ **end if**
end for

The required subgradients are then given by

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

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

2.1.4 RBF-χ^2 kernel.

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

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

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 Learning using SGD

We learn the non-linear SVM (nSVM) directly, without the kernel trick, by optimizing the primal (14) w.r.t. $(\mathbf{w}, b) \in \mathbb{R}^{D+1}$. We follow Sharma and Jurie (2013) and use stochastic gradient descent (SGD). The required gradients for the regularization and score functions are summarized in Table 1. Following Akata et al (2013), 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. 5). 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.

2.3 Relation with pre-image and reduced set methods

While many of the current methods 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 (Maji et al, 2008; Perronnin et al, 2010; Vedaldi and Zisserman, 2012), 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 (Bruges, 1996; Mika et al, 1998; Scholkopf and Smola, 2001) 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

$$\sum_{i=1}^{N} y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) \approx k(\mathbf{w}, \mathbf{x}),\ \forall\ \mathbf{x} \in \mathbb{R}^D, \tag{28}$$

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 (Chapelle, 2007). Hence we propose instead to integrate approximations motivated by reduced set formulations directly into the original optimization of regularized loss minimization.

Our method is thus closely related to previous approaches *e.g.* Keerthi et al (2006) proposed a iterative greedy forward selection for choosing a small basis vectors (*cf.* support vectors) for reducing the complexity of the SVM. Joachims and Yu (2009) proposed a cutting-plane based algorithm to learn the basis vectors and Cotter et al (2013) proposed to approximate the dense SVM solution with a sparse one, among many others. These approaches either try to approximate the solution obtained by conventional kernel SVM or try to learn from scratch a low complexity sparse classifier. We are similar in spirit to the second type of methods,

as we also propose to learn a reduced complexity kernel SVM. We demonstrate empirically that using just one 'basis vector' we can obtain performances competitive to the state-of-the-art computer vision systems while being much faster, and using a scalable stochastic gradient based learning algorithm.

3 Experimental results

We use the following publicly available datasets to evaluate the various aspects of the proposed method. Fig. 2 gives some examples of the kind of images the test databases contain.

Flickr Materials dataset[2] (Sharan et al, 2009) is a challenging dataset with 10 material categories, *e.g.* glass, leather, fabric. The dataset was created manually by downloading images from Flickr.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] (Quattoni and Torralba, 2009) 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] (Sharma and Jurie, 2011) 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 (Felzenszwalb et al, 2010) 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] (Everingham et al, 2007) 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 is a reference 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 (Csurka et al, 2004), one based on gray SIFT (Lowe, 2004) and other based on opponent SIFT, which has been shown to be a good color descriptor (van de Sande et al, 2010). 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 $1\times1, 2\times2$ and 3×1 partitions. We use `vlfeat` library (Vedaldi and Fulkerson, 2008) 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 Sanchez et al (2013), following Chatfield et al (2011), 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. Such free parameter was also used in a similar margin maximization framework albeit in the context of nonlinear metric learning (Kedem et al, 2012). In the present case, 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 for linear SVM) that *all* of the training examples

[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. 2 Example labeled images from the datasets used for experimental evaluation.

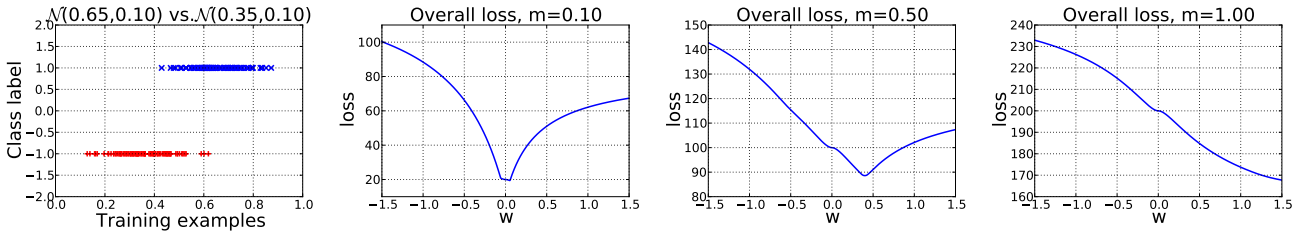

Fig. 3 The loss function as a function of w for different values of m, corresponding to χ^2 kernel for synthetic 1-D examples generated randomly from two normal distributions $\mathcal{N}(0.65, 0.1)$ vs. $\mathcal{N}(0.35, 0.1)$. The leftmost figure plots (x,y) for the randomly sampled points and the next three figures plot the loss function for $m = 0.1, 0.5, 1.0$. See § 3.1 for discussion.

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 plotting the loss function, shown in Fig. 3 (for χ^2 kernel). We see that the loss function changes as we vary m, since the learning focuses on 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 we indeed 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. 5 shows typical 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 (Everingham et al, 2007) dataset, for the different kernels. We use `libsvm` library by Chang and Lin (2001) with precomputed kernels for the kernel SVM results. We see that the proposed method achieves slightly lower results as the ker-

	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 (Everingham et al, 2007) 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	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 (Everingham et al, 2007) (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 (Vedaldi and Zisserman, 2012) (with Histogram intersection kernel).

nel 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 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 (2012) which computes a finite dimensional map approximating the kernel. Such mappings thus enable one to compute linear classifiers in the mapped space. It was shown by Vedaldi and Zisserman (2012) that this feature mapping obtains better results that the one by Maji and Berg (2009).

We use `vlfeat` library by Vedaldi and Fulkerson (2008) to compute the FM corresponding to Vedaldi and Zisserman (2012) for the histogram intersection and χ^2 kernels. We use `liblinear` by Fan et al (2008) (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. 4 shows the per class performances of our method vs. kernel SVM (using `libsvm` by Chang and Lin (2001)

Regularization	k_h	k_c	k_{rc}	k_{re}
Yes	54.2	53.9	55.2	40.9
No	53.1	52.5	53.6	38.2

Table 5 Performance (mAP) of the proposed nonlinear SVM, with and without regularization, on the Pascal VOC 2007 (Everingham et al, 2007) dataset with Histogram intersection (k_h), χ^2 (k_c), RBF-χ^2 (k_{rc}) and RBF-Euclidean (k_{re}) kernels.

with precomputed kernels) and FM by Vedaldi and Zisserman (2012), on the Pascal VOC 2007 (Everingham et al, 2007) and Human Attributes (HAT) (Sharma and Jurie, 2011) 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 (2012).

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× bigger for explicit feature map compared to our method and (empirically) our method is about 19× 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 Effect of regularization

As discussed in §1.1, the proposed nonlinear SVM can also be seen as regularized kernel neural network with only one hidden unit with the parametrized kernel function as the activation function. Regularization in neural networks is an important issue, lacking which the performance degrades as overfitting occurs. There have been different ways of regularizing a neural network *e.g.* by limiting the number of hidden units, using weight decay while training and training with transformed examples to achieve invariance (Bishop, 1996). In the proposed method, the regularization is derived from the max-margin principle owing to the SVM perspective. Table 5 shows the result of the proposed nSVM with

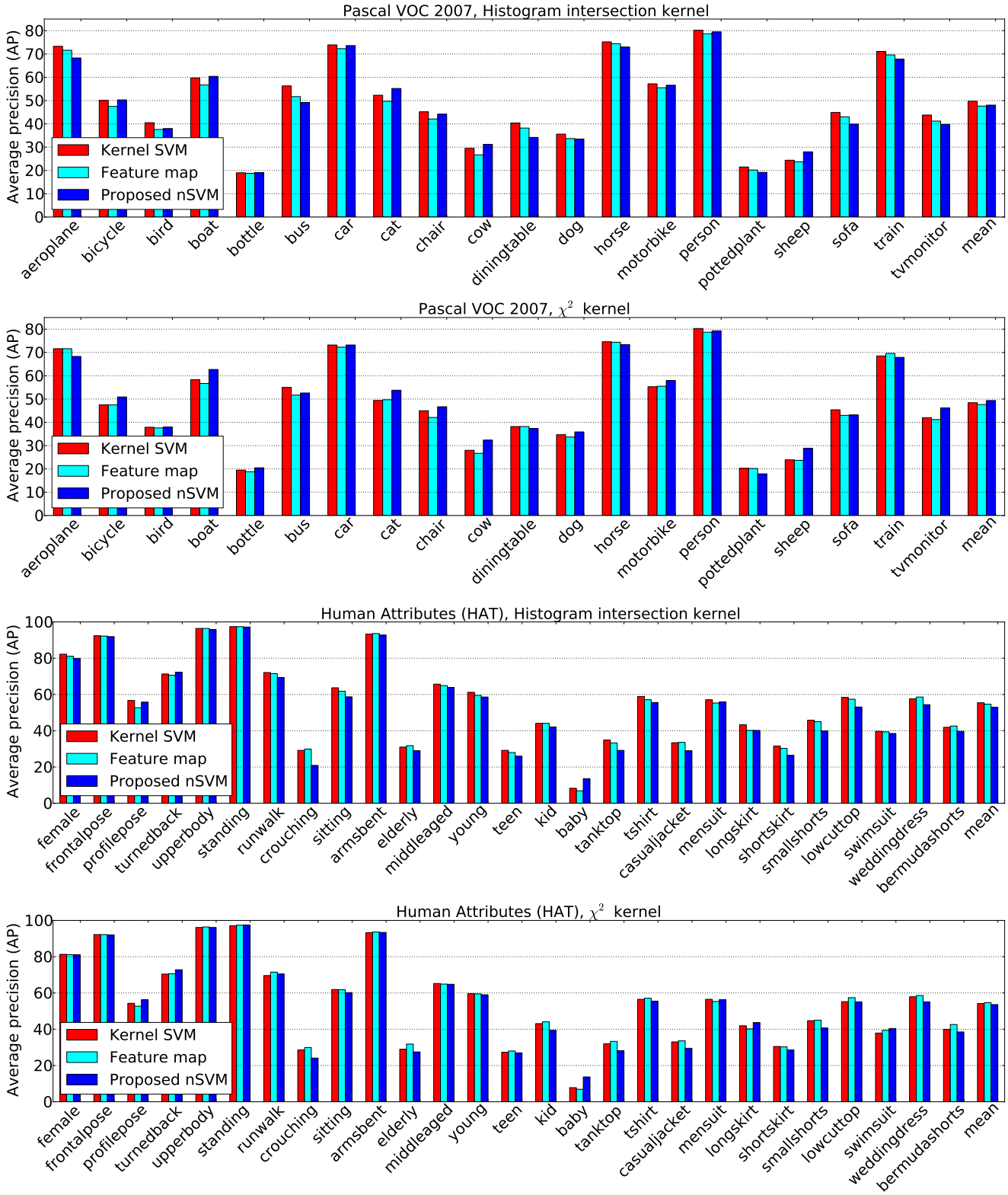

Fig. 4 The average precisions for different classes (and the mean AP) of the Human Attributes (HAT) dataset (Sharma and Jurie, 2011) and Pascal VOC 2007 (Everingham et al, 2007) dataset (image classification task) for (i) Kernel SVM (`libsvm` by Chang and Lin (2001)), (ii) the explicit feature mapping of Vedaldi and Zisserman (2012) (iii) the proposed method (nSVM), for histogram intersection and χ^2 kernels.

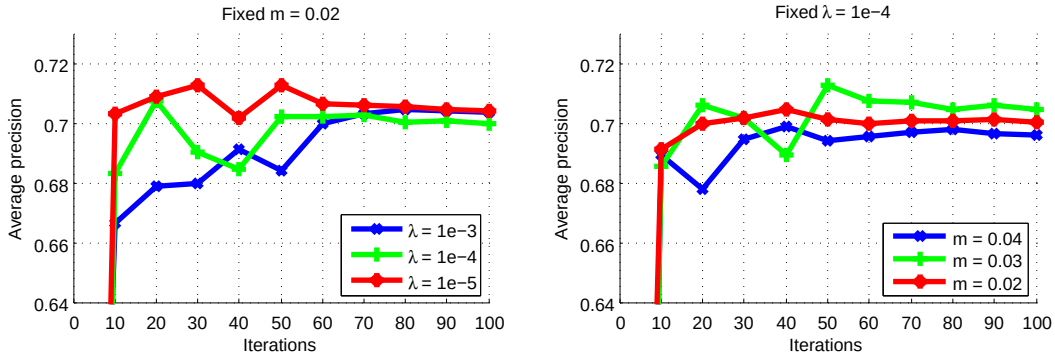

Fig. 5 The average precisions for different values of (left) regularization parameter λ and (right) hinge loss parameter m, for a typical convergence of the proposed method.

	nSVM$_{GRAY}$	nSVM$_{COL}$	nSVM$_{GRAY}$ +nSVM$_{COL}$	Fisher Vector	Fisher Vector +nSVM$_{GRAY}$	Fisher Vector +nSVM$_{GRAY}$ +nSVM$_{COL}$
Flickr Materials	50.2	51.9	57.3	56.1	56.7	59.7
MIT Scenes	53.5	50.5	58.0	62.7	63.3	64.6
Human Attributes	57.2	59.2	61.5	64.2	65.0	65.6
Pascal VOC 2007	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 (Sanchez et al, 2013) 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).

and without regularization (with gray SIFT features). We can observe that the regularization in the proposed method adds performance consistently for the different important kernels in computer vision, highlighting an advantage of the method.

3.4 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 (van de Sande et al, 2010), along with those of Fisher Vectors (Sanchez et al, 2013) and the various combinations of the methods using late fusion on the four datasets: Flickr Materials (Sharan et al, 2009), Human Attributes (Sharma and Jurie, 2011), Indoor Scenes (Quattoni and Torralba, 2009), and Pascal VOC 2007 (Everingham et al, 2007).

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 (Sanchez et al, 2013) while being 34× (75×) faster in testing (encoding) times and at least 10× 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 Fisher Vectors while being 11× (33×) faster in testing (encoding) times and at least 5× 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 Sanchez et al (2013), 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.

3.5 Comparison with the state-of-the-art

Tables 6, 7, 8 and 9 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 (Everingham et al, 2007) (Tables 4 and 6), 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 (Yang et al, 2009a). We achieve similar performance as Harzallah et al (2009), who used object detection, in addition, to improve the classification score, leading to a high performing but very slow method. Recently, Sanchez et al (2013) 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× slower than us. Also, note that the performance of FV on color features (52.6 mAP), as reported in Sanchez et al (2013), 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 (Quattoni and Torralba, 2009) (Tables 4 and 9), nSVM with gray SIFT BoF performs (53.5 mAP) similar to more complex locality constrained linear coding (LLC) with max pooling (Wang et al, 2010) (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 (Juneja et al, 2013) (63.2 mAP).

On the Human Attributes dataset (Sharma and Jurie, 2011) (Tables 4 and 7), nSVM with gray SIFT BoF performs competitively at 57.2 mAP w.r.t. current methods learning adaptive spatial partitions (Sharma and Jurie, 2011) (53.8 mAP), and learning class-wise or global part dictionaries, *i.e.,* Sharma et al (2013) (58.7 mAP) and Joo et al (2013) (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 (Sharan et al, 2009) (Tables 4 and 8), 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 Liu et al (2012)), descriptors (54.0 mAcc Hu et al (2011)) and classification methods (55.8 Timofte and Van Gool (2012) and 44.6 Liu et al (2010) 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

Method	mAP	Remarks
Challenge winners	59.4	Several features with learnt weights
van Gemert et al (2008)	60.5	Several color (and gray) features
Yang et al (2009a)	62.2	Mult kernel learning
Chatfield et al (2011)	61.7	Fisher Vectors (FV)
Sanchez et al (2013)	63.9	FV gray and color
Harzallah et al (2009)	63.5	Object detection
Present	63.6	

Table 6 Comparison with existing methods on the Pascal VOC 2007 dataset (Everingham et al, 2007).

Method	mAP	Remarks
Sharma and Jurie (2011)	53.8	Learnt spatial partitions
Sharma et al (2013)	58.7	Many parts learnt for each class
Joo et al (2013)	59.3	Many parts shared between classes
Present	65.6	

Table 7 Comparison with existing methods on the dataset of Human Attributes (HAT) (Sharma and Jurie, 2011).

Method	mAcc	Remarks
Liu et al (2010)	44.6	Bayesian learning with several features
Timofte and Van Gool (2012)	55.8	Collaborative representation
Liu et al (2012)	48.2	Sorted random projection features
Hu et al (2011)	54.0	Kernel descriptors
Present	57.6	Corresponding to 59.7 mAP

Table 8 Comparison with existing methods (mean class accuracy) on the Flickr Materials dataset (Sharan et al, 2009).

Method	mAP	Remarks
Wang et al (2010)	53.0	Locality constrained linear coding (LLC)
Juneja et al (2013)	43.5	Bag of parts (BoP)
Juneja et al (2013)	63.2	Fisher Vectors + BoP
Sanchez et al (2013)	61.1	Fisher Vectors, as reported in Juneja et al (2013)
Present	64.6	

Table 9 Comparison with existing methods on the MIT Indoor Scenes dataset (Quattoni and Torralba, 2009).

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× faster and requires 7× 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× (75×) faster in testing (encoding) times and more than 10× more memory efficient. Further, adding color based BoF with nSVM leads to up to 96% performance of FV while being 11× (33×) faster in testing (encoding) times and more than 5× 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 Krapac et al (2011a); Lazebnik and Raginsky (2009); Moosmann et al (2008), should increase the performance of our nSVM based systems further.

References

Akata Z, Perronnin F, Harchaoui Z, Schmid C (2013) Good practice in large-scale learning for image classification. PAMI 4, 7

Bishop CM (1996) Neural Networks for Pattern Recognition. Oxford University Press 4, 10

Bishop CM (2006) Pattern recognition and machine learning. Springer 4

Bottou L, Bousquet O (2008) The tradeoffs of large scale learning. In: NIPS 3

Boureau YL, Bach F, LeCun Y, Ponce J (2010) Learning mid-level features for recognition. In: CVPR 2, 4

Boureau YL, Le Roux N, Bach F, Ponce J, LeCun Y (2011) Ask the locals: multi-way local pooling for image recognition. In: ICCV 2, 4

Bruges CJC (1996) Simplified suport vector decision rules. In: ICML 3, 4, 7

Chang CC, Lin CJ (2001) LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm 9, 10, 11

Chang KW, Hsieh CJ, Lin CJ (2008) Coordinate descent method for large-scale l2-loss linear support vector machines. Journal of Machine Learning Research 9:1369–1398 3

Chapelle O (2007) Training a support vector machine in the primal. Neural Computation 19(5):1155–1178 5, 7

Chatfield K, Lempitsky V, Vedaldi A, Zisserman A (2011) The devil is in the details: an evaluation of recent feature encoding methods. In: BMVC 2, 4, 8, 14

Cimpoi M, Maji S, Vedaldi A (2014) Deep convolutional filter banks for texture recognition and segmentation. arXiv:14116836 2

Cotter A, Shalev-Shwartz S, Srebro N (2013) Learning optimally sparse support vector machines. In: ICML 7

Csurka G, Dance CR, Fan L, Willamowski J, Bray C (2004) Visual categorization with bags of keypoints. In: Intl. Workshop on Stat. Learning in Comp. Vision 1, 3, 8

Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: CVPR 3

Downs T, Gates KE, Masters A (2001) Exact simplification of support vector solutions. In: JMLR 3

Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A (2007) The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html 1, 2, 3, 8, 9, 10, 11, 12, 13, 14

Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research 9:1871–1874 3, 4, 10

Felzenszwalb P, Girshick R, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part based models. PAMI 32(9):1627–1645 3, 8

Fine S, Scheinberg K (2001) Efficient SVM training using low-rank kernel representations. In: JMLR 3

Gao S, Tsang IW, Chia LT, Zhao P (2010) Local features are not lonely–laplacian sparse coding for image classification. In: CVPR 2, 4

van Gemert J, Geusebroek JM, Veenman C, Smeulders A (2008) Kernel codebooks for scene categorization. In: ECCV 1, 2, 14

van Gemert JC, Veenman CJ, Smeulders AW, Geusebroek JM (2010) Visual word ambiguity. PAMI 32(7):1271–1283 4

Harzallah H, Jurie F, Schmid C (2009) Combining efficient object localization and image classification. In: ICCV 1, 2, 3, 13, 14

Hu D, Bo L, Ren X (2011) Toward robust material recognition for everyday objects. In: BMVC 13, 14

Huang Y, Huang K, Yu Y, Tan T (2011) Salient coding for image classification. In: CVPR 2, 4

Jaakkola T, Haussler D (1998) Exploiting generative models in discriminative classifiers. In: NIPS 4

Joachims T, Yu CNJ (2009) Sparse kernel svms via cutting-plane training. Machine Learning 76(2-3):179–193 7

Joo J, Wang S, Zhu SC (2013) Human attribute recognition by rich appearance dictionary. In: ICCV 1, 13, 14

Juneja M, Vedaldi A, Jawahar CV, Zisserman A (2013) Blocks that shout: Distinctive parts for scene classification. In: CVPR 1, 4, 13, 14

Kedem D, Tyree S, Sha F, Lanckriet GR, Weinberger KQ (2012) Non-linear metric learning. In: NIPS 8

Keerthi SS, Chapelle O, DeCoste D (2006) Building support vector machines with reduced classifier complexity. Journal of Machine Learning Research 7:1493–1515 7

Krapac J, Verbeek J, Jurie F (2011a) Learning tree-structured descriptor quantizers for image categorization. In: BMVC 14

Krapac J, Verbeek J, Jurie F (2011b) Modeling spatial layout with Fisher vectors for image categorization. In: ICCV 3

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems 2

Kwok JY, Tsang IW (2004) The pre-image problem in kernel methods. IEEE Transactions on Neural Networks 15(6):1517–1525 4

Lazebnik S, Raginsky M (2009) Supervised learning of quantizer codebooks by information loss minimization. PAMI 31(7):1294–1309 14

Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR 1, 3

Lee YJ, Mangasarian OL (2001) RSVM: Reduced support vector machines. In: SIAM Conf. on Data Mining 3

Liu C, Sharan L, Adelson EH, Rosenholtz R (2010) Exploring features in a bayesian framework for material recognition. In: CVPR 13, 14

Liu L, Wang L, Liu X (2011) In defense of soft-assignment coding. In: ICCV 4

Liu L, Fieguth P, Clausi D, Kuang G (2012) Sorted random projections for robust rotation-invariant texture classification. Pattern Recognition 45(6):2405–2418 13, 14

Liu L, Shen C, Wang L, van den Hengel A, Wang C (2014) Encoding high dimensional local features by sparse coding based fisher vectors. In: Advances in Neural Information Processing Systems 2

Lowe D (2004) Distinctive image features form scale-invariant keypoints. Intl Journal of Computer Vision 60(2):91–110 8

Maji S, Berg AC (2009) Max-margin additive classifiers for detection. In: ICCV 2, 3, 10

Maji S, Berg AC, Malik J (2008) Classification using intersection kernel support vector machines is efficient. In: CVPR 2, 3, 7

Mika S, Scholkopf B, Smola AJ, Muller KR, Scholz M, Ratsch G (1998) Kernel PCA and de-noising in feature spaces. In: NIPS 4, 5, 7

Moosmann F, Nowak E, Jurie F (2008) Randomized clustering forests for image classification. PAMI 30(9):1632–1646 14

Oquab M, Bottou L, Laptev I, Sivic J (2014) Learning and transferring mid-level image representations using convolutional neural networks. In: CVPR 2

Osuna E, Girosi F (1998) Reducing the run-time complexity of support vector machines. In: Proceedings of the International Conference on Pattern Recognition 3

Park J, Sandberg IW (1991) Universal approximation using radial-basis-function networks. Neural computation 3(2):246–257 4

Perronnin F, Dance CR, Csurka G, Bressan M (2006) Adapted vocabularies for generic visual categorization. In: ECCV 4

Perronnin F, Sanchez J, Liu Y (2010) Large-scale image categorization with explicit data embedding. In: CVPR 2, 3, 7

Perronnin F, Akata Z, Harchaoui Z, Schmid C (2012) Towards good practice in large-scale learning for image classification. In: CVPR 3

Quattoni A, Torralba A (2009) Recognizing indoor scenes. In: CVPR 1, 2, 8, 12, 13, 14

Raginsky M, Lazebnik S (2009) Locality-sensitive binary codes from shift-invariant kernels. In: NIPS 3

Rahimi A, Recht B (2007) Random features for large-scale kernel machines. In: NIPS 3

Rauber TW, Berns K (2011) Kernel multilayer perceptron. In: SIBGRAPI conf. on Graphics, Patterns and Images 4

Rubner Y, Tomasi C, Guibas LJ (2000) The earth mover's distance as a metric for image retrieval. Intl Journal of Computer Vision 40(2):99–121 3

Sanchez J, Perronnin F, Akata Z (2011) Fisher vectors for fine-grained visual categorization. In: CVPR Workshops 4

Sanchez J, Perronnin F, Mensink T, Verbeek J (2013) Image classification with the fisher vector: Theory and practice. IJCV 1, 2, 4, 8, 12, 13, 14

van de Sande KEA, Gevers T, Snoek CGM (2010) Evaluating color descriptors for object and scene recognition. PAMI 32(9):1582–1596 2, 8, 12

Scholkopf B, Smola AJ (2001) Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA, USA 3, 5, 7

Scholkopf B, Simard P, Vapnik V, Smola A (1997) Improving the accuracy and speed of support vector machines. In: NIPS 3

Sharan L, Rosenholtz R, Adelson E (2009) Material perception: What can you see in a brief glance? Journal of Vision 9(8):784–784 2, 8, 12, 13, 14

Sharma G, Jurie F (2011) Learning discriminative representation image classification. In: BMVC 2, 8, 10, 11, 12, 13, 14

Sharma G, Jurie F (2013) A novel approach for efficient SVM classification with histogram intersection kernel. In: BMVC 2, 7

Sharma G, Jurie F, Schmid C (2012) Discriminative spatial saliency for image classification. In: CVPR 1, 3

Sharma G, Jurie F, Schmid C (2013) Expanded parts model for human attribute and action recognition in still images. In: CVPR 1, 13, 14

Singer Y, Srebro N (2007) Pegasos: Primal estimated sub-gradient solver for SVM. In: ICML 3

Sivic J, Zisserman A (2003) Video Google: A text retrieval approach to object matching in videos. In: ICCV 1, 3

Smola A, Scholkopf B (2000) Sparse greedy matrix approximation for machine learning. In: ICML 3

Thorsten J (2006) Training linear SVMs in linear time. In: KDD 3

Timofte R, Van Gool LJ (2012) A training-free classification framework for textures, writers, and materials. In: BMVC 13, 14

Vedaldi A, Fulkerson B (2008) VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/ 8, 10

Vedaldi A, Zisserman A (2012) Efficient additive kernels via explicit feature maps. PAMI 34(3):480–492 2, 4, 7, 10, 11

Vedaldi A, Gulshan V, Varma M, Zisserman A (2009) Multiple kernels for object detection. In: ICCV 3

Wang J, Yang J, Yu K, Lv F, Huang T, Gong Y (2010) Locality-constrained linear coding for image classifi-

cation. In: CVPR, vol 0 2, 4, 13, 14
Williams CK, Seeger M (2000) The effect of the input density distribution on kernel-based classifiers. In: ICML 3
Williams CK, Seeger M (2001) Using the nystrom method to speed up kernel machines. In: NIPS 3
Winn J, Criminisi A, Minka T (2005) Object categorization by learned universal visual dictionary. In: ICCV 4
Xu J, Zhang X, Li Y (2001) Kernel neuron and its training algorithm. In: ICONIP 4
Yang J, Li Y, Tian Y, Duan L, Gao W (2009a) Group sensitive multiple kernel learning for object categorization. In: ICCV 13, 14
Yang J, Yu K, Gong Y, Huang T (2009b) Linear spatial pyramid matching using sparse coding for image classification. In: CVPR 2, 4
Zhang T, Ghanem B, Liu S, Xu C, Ahuja N (2013) Low-rank sparse coding for image classification. In: ICCV 4
Zhou X, Yu K, Zhang T, Huang TS (2010) Image classification using super-vector coding of local image descriptors. In: ECCV 4