Fast Image Classification with Reduced Multiclass Support Vector Machines

Marco Melis, Luca Piras, Battista Biggio, Giorgio Giacinto, Giorgio Fumera, and Fabio Roli

Dept. of Electrical and Electronic Engineering, University of Cagliari
Piazza d'Armi, 09123 Cagliari, Italy
{marco.melis,luca.piras,battista.biggio,giacinto,fumera,roli}@diee.unica.it
WWW home page: http://pralab.diee.unica.it

Abstract. Image classification is intrinsically a multiclass, nonlinear classification task. Support Vector Machines (SVMs) have been successfully exploited to tackle this problem, using one-vs-one or one-vs-all learning schemes to enable multiclass classification, and kernels designed for image classification to handle nonlinearities. To classify an image at test time, an SVM requires matching it against a small subset of the training data, namely, its support vectors (SVs). In the multiclass case, though, the union of the sets of SVs of each binary SVM may almost correspond to the full training set, potentially yielding an unacceptable computational complexity at test time. To overcome this limitation, in this work we propose a well-principled reduction method that approximates the discriminant function of a multiclass SVM by jointly optimizing the full set of SVs along with their coefficients. We show that our approach is capable of reducing computational complexity up to two orders of magnitude without significantly affecting recognition accuracy, by creating a super-sparse, budgeted set of virtual vectors.

1 Introduction

In the last decade, Support Vector Machines (SVMs) [23] have gained increasing popularity in the field of image classification, due to their high generalization capability [25, 14, 1]. In addition, the introduction of novel kinds of feature descriptors, like the Scale-Invariant Feature Transform (SIFT) [15] and the Histogram of Oriented Gradients (HoG) [8], extracted following the Bag-of-Words (BoW) paradigm and the spatial pyramid framework [11], has caused a significant increase in the dimensionality of the corresponding feature spaces. This change, along with the ability of SVMs to retain a high generalization capability even in high-dimensional feature spaces, has favored a wide diffusion of SVMs in image classification tasks.

Under this setting, high-dimensional image descriptors in combination with linear classifiers are used. The use of linear classifiers is usually motivated by computational efficiency reasons. This is especially important when dealing with a large number of classes and images, even if it may not attain a very high

classification accuracy [14, 1]. To overcome this drawback, the use of kernel-based approaches has become widely popular. Although being frequently used, this approach has the disadvantage of requiring a large number of computations during testing, as it requires matching each test image against a potentially large number of images in the training set. For instance, to classify a test image, an SVM requires computing the kernel values between the test image and the so-called Support Vectors (SVs), whose number increases linearly with the training set size [20, 6]. Usually, in image classification, researchers aim to optimize the training phase and use parallel computing to manage the complexity at test time while preserving classification accuracy [14].

The use of nonlinear classifiers, besides bringing clear benefits in terms of classification performance, demands for a higher complexity at test time. In fact, if linear classifiers can classify a test image by simply computing a scalar product between its feature vector and the set of learned feature weights [23], the use of kernels requires a much higher number of comparisons, as mentioned above. It is thus clear that enabling the use of kernel-based methods on large image datasets while retaining a reduced computational complexity at test time can be considered a relevant open research issue. In the field of pattern recognition, diverse methods have been proposed to tackle this problem. In particular, several methods have been proposed to reduce the number of SVs in SVMs [21, 19] but, to the best of our knowledge, no one has been ever exploited for image classification purposes.

Although SVMs have been designed for binary classification, in object recognition and image classification tasks they have to deal with several classes. To this end, several multiclass extensions have been considered (see Sect. 2).

In this paper, we propose a novel algorithm that can drastically reduce the number of required matchings without significantly affecting recognition accuracy. To this end, our algorithm creates a small set of virtual support vectors, and jointly optimizes the objective function of all SVMs (one for each class) at once. In particular, our algorithm optimizes a unique, budgeted set of virtual vectors along with an optimal set of coefficients for their combination (see Sect. 3). It is also worth noting that the proposed method may be exploited to speed up other non-parametric approaches besides SVMs, making it suitable for a wider range of pattern recognition tasks.

The reported results show that our approach is capable of reducing computational complexity up to two orders of magnitude, while only worsening the recognition accuracy of about 5% in the worst case (see Sect. 4).

2 Image Classification with Visual Descriptors

Classifying a scene depicted in an image amounts to labeling it among a set of categories, according to its semantic meaning. In recent years, scene classification has been an active and important research topic, ranging from computer vision to content-based image retrieval, as witnessed by the large number of related approaches proposed in the last decade [13, 11, 26]. Despite this, a number of challenging aspects in scene classification can be still considered open

issues, including inter-class similarity, intra-class variability, and the wide range of illumination and scale changes. Along with the considerable progress made in this field, tougher challenges have been posed by researchers, in terms of more difficult benchmark datasets, *i.e.*, bigger datasets with an increasing number of images: 8-category scenes [16], 13-category scenes [13], 15-category scenes [11], and 397-category scenes (SUN-dataset) [24].

Feature extraction and classification algorithms play an important role in scene classification problems [5, 18]. Regarding feature descriptors, researchers have recently employed histograms of local descriptors instead of global image features. The former are indeed able to better model the content of images in order to fill the semantic gap between low-level features and high-level concepts.

The most famous approach uses the so-called bag-of-features paradigm to model visual scenes in image collections [13]. This approach has been first exploited with SIFT descriptors [15] but it has been quickly used also with other descriptors. One of the main drawbacks of the bag-of-features representation is that it does not account for spatial information. To overcome this limitation, an efficient extension of this approach, called spatial pyramid matching (SPM), has been proposed in [11]. It exploits spatial relationships between neighboring local regions. Compared with methods based on low-level features, both the aforementioned approaches achieve very good results for multiple scene classification, although they suffer a high computational cost and generate very high-dimensional feature spaces.

Due to their good generalization ability also in the presence of high-dimensional feature spaces, SVMs are among the most used classifiers in scene classification tasks [7, 24, 26]. SVMs have been designed for binary classification, but they can be exploited for multiclass classification by decomposing the multiclass problem into several two-class sub-problems, *e.g.*, using the One-vs-One (OVO) and the One-vs-All (OVA) approaches. The first method trains each binary classifier on two out of N classes and builds $N(N-1)/2$ classifiers, subsequently combined through majority voting. Conversely, the second approach constructs a set of N binary classifiers, each aiming to discriminate one given class from the remaining ones. During classification, a sample is assigned to the class exhibiting the highest *support*, *i.e.*, the one corresponding to the classifier that outputs the most confident prediction.

Another important aspect of statistical learning approaches like SVMs is the choice of the kernel, since an inappropriate kernel can lead to poor performance. There are currently no techniques available to know which kernel to use, so it easy to understand why several authors exploit well-known kernels such as the polynomial kernel or the Radial Basis Function (RBF) kernel. In image classification, however, several studies have investigated this issue, reporting that histogram-intersection kernels usually outperform polynomial and RBF kernels.[2, 3, 7, 11].

3 Reducing Multiclass Support Vector Machines

In this section, we extend the SVM reduction method originally proposed in [4] for binary classification problems to the multiclass classification case. Let us

assume we are given a set $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n \in \mathcal{X}^n \times \mathcal{Y}^n$ of n images along with their labels $y \in \mathcal{Y} = \{1, \ldots, c\}$, being c the number of classes.[1] Training a one-vs-all multiclass SVM on $\mathcal{D}$ amounts to learning a binary SVM for each class $k = 1, \ldots, c$, using the samples of class k as positive training samples, and the remaining ones as negative. Its decision function is then given as:

$$y^\star = \arg\max_{k=1,\ldots,c} g_k(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i^k k(\boldsymbol{x}, \boldsymbol{x}_i) + b^k, \tag{1}$$

where $y^\star$ is the predicted class label, $g_k(\boldsymbol{x})$ is the k^{th} SVM's discriminant function, and the set $\{\alpha_i^k\}_{i=1}^n$ are its *signed* dual coefficients (positive if $y_i = k$, and negative otherwise). Although each binary SVM has a sparse solution, *i.e.*, only a subset of the values in $\{\alpha_i^k\}_{i=1}^n$ are not null (corresponding to its *support* vectors), their number grows linearly with the training set size [20, 6]. Furthermore, in the multiclass case, classifying an input image requires matching it against the set of SVs of each binary SVM, which yields a number of matchings (*i.e.*, kernel computations) equal to the size of the *union* of the sets of SVs of each binary SVM. In the sequel, we refer to this number as m, and, as we will see in Sect. 4, m may be very close to the full training set size n.

Our goal is to reduce the number of required matchings m to a much smaller number r, by approximating each SVM's discriminant function $g_k(\boldsymbol{x})$ with a much *sparser* linear combination $h_k(\boldsymbol{x})$, such that all functions $h_k(\boldsymbol{x})$, for $k = 1, \ldots, c$ share the *same* set of SVs $\boldsymbol{z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_r) \in \mathcal{X}^r$, but have a different set of weighting coefficients $\boldsymbol{\beta}_k = (\beta_1^k, \ldots, \beta_r^k) \in \mathbb{R}^r$. In other words, we aim to approximate the decision function given by Eq. (1) as:

$$y^\star = \arg\max_{k=1,\ldots,c} h_k(\boldsymbol{x}) = \sum_{j=1}^{r} \beta_j^k k(\boldsymbol{x}, \boldsymbol{z}_j) + b^k \ . \tag{2}$$

To find the coefficients $\{\boldsymbol{\beta}_k\}_{k=1}^c$ and the shared SVs $\boldsymbol{z}$, we extend our recent work in [4] to the multiclass case. In that work, inspired by the earlier work in [19], we proposed a reduction method based on the idea of minimizing the squared Euclidean distance between the values of g_k and h_k computed on the training points, with respect both to $\boldsymbol{\beta}_k$ and to the choice of the SVs $\boldsymbol{z}$. In practice, we did not require the SVs $\boldsymbol{z}$ to be samples of $\mathcal{D}$, but allow for the creation of *novel*, *virtual* vectors. In the multiclass case, the initial formulation in [4] can be modified by considering k distinct SVMs that share the same SVs $\boldsymbol{z}$, as:

$$\min_{\boldsymbol{\beta}, \boldsymbol{z}} \Omega = \sum_{k=1}^{c} \sum_{i=1}^{n} u_i \left(h_k(\boldsymbol{x}_i) - g_k(\boldsymbol{x}_i)\right)^2 + \lambda \boldsymbol{\beta}_k^\top \boldsymbol{\beta}_k \ , \tag{3}$$

where the scalars $u_1, \ldots, u_n$ can be used to balance the contribution of each point $\boldsymbol{x}_i$ to the empirical loss (*e.g.*, if classes are unbalanced), the regularizer

[1] For simplicity, we assume here that each image can belong only to one class, *i.e.*, we focus on single-label classification. Although our approach can be easily extended to the multi-label classification case, we leave this investigation to future work.

Algorithm 1 Reduced Multiclass SVM (RMSVM), adapted from [4]

Input: the training data $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$; the kernel function $k(\cdot,\cdot)$; the parameters C and λ; the initial vectors $\{\boldsymbol{z}_j^{(0)}\}_{j=1}^r$; the gradient step size η; a small number ϵ.
Output: The coefficients $\boldsymbol{\beta}$ and the SVs $\{\boldsymbol{z}_j\}_{j=1}^r$.

Learn a one-vs-all multiclass SVM on $\mathcal{D}$, with kernel $k(\cdot,\cdot)$ and regularizer C.
Compute $\{\boldsymbol{g}_k\}_{k=1}^c$ by classifying $\mathcal{D}$ with each binary SVM.
Set the iteration count $q \leftarrow 0$.
Compute $\{\boldsymbol{\beta}_k^{(0)}\}_{k=1}^c$ (Eq. 5) using $\boldsymbol{z}_1^{(0)}, \ldots, \boldsymbol{z}_r^{(0)}$.
repeat
  Set $j \leftarrow \mathrm{mod}(q, r) + 1$ to index a support vector.
  Compute $\frac{\partial \Omega}{\partial \boldsymbol{z}_j}$ using Eq. (6).
  Increase the iteration count $q \leftarrow q + 1$
  Set $\boldsymbol{z}_j^{(q)} \leftarrow \boldsymbol{z}_j^{(q-1)} + \eta \frac{\partial \Omega}{\partial \boldsymbol{z}_j^{(q-1)}}$.
  if $\boldsymbol{z}_j^{(q)} \notin \mathcal{X}$, **then** project $\boldsymbol{z}_j^{(q)}$ onto $\mathcal{X}$.
  Set $\boldsymbol{z}_i^{(q)} = \boldsymbol{z}_i^{(q-1)}$, $\forall i \neq j$.
  Compute $\{\boldsymbol{\beta}_k^{(q)}\}_{k=1}^c$ (Eq. 5) using $\boldsymbol{z}_1^{(q)}, \ldots, \boldsymbol{z}_r^{(q)}$.
until $\left| \Omega\left(\boldsymbol{\beta}^{(q)}, \boldsymbol{z}^{(q)}\right) - \Omega\left(\boldsymbol{\beta}^{(q-1)}, \boldsymbol{z}^{(q-1)}\right) \right| < \epsilon$
return: $\boldsymbol{\beta} = \boldsymbol{\beta}^{(q)}$, and $\boldsymbol{z} = \boldsymbol{z}^{(q)}$.

$\boldsymbol{\beta}_k^\top \boldsymbol{\beta}_k$ controls overfitting, and λ is a regularization parameter.[2] By denoting with $\boldsymbol{g}_k, \boldsymbol{h}_k \in \mathbb{R}^n$ the values of g_k and h_k for the training points, and with $\mathbf{U} \in \mathbb{R}^{n \times n}$ the diagonal matrix $\mathrm{diag}(\mathbf{U}) = (u_1, \ldots, u_n)$, we can rewrite Eq. (3) in matrix form as:

$$\Omega(\boldsymbol{\beta}, \boldsymbol{z}) = \sum_{k=1}^c \left(\boldsymbol{h}_k^\top \mathbf{U} \boldsymbol{h}_k - 2\boldsymbol{h}_k^\top \mathbf{U} \boldsymbol{g}_k + \boldsymbol{g}_k^\top \mathbf{U} \boldsymbol{g}_k \right) + \lambda \boldsymbol{\beta}_k^\top \boldsymbol{\beta}_k \ . \tag{4}$$

Problem (4) can be solved by iteratively modifying $\boldsymbol{\beta}$ and $\boldsymbol{z}$, as detailed below. The full procedure is given as Algorithm 1. We also report a two-dimensional example in Fig. 1, in which our algorithm reduces the number of SVs of approximately 24 times, from $m = 73$ to $r = 3$.

$\boldsymbol{\beta}$-step. The coefficients $\boldsymbol{\beta}_k$ for each reduced SVM are computed assuming that the SVs $\boldsymbol{z}$ are fixed. This yields a standard ridge regression, which can be analytically solved by deriving Eq. (4) with respect to $\boldsymbol{\beta}_k$, assuming $\boldsymbol{z}$ constant, and then setting the gradient to zero:

$$\boldsymbol{\beta}_k = \underbrace{\left(\mathbf{K}_{\boldsymbol{xz}}^\top \mathbf{U} \mathbf{K}_{\boldsymbol{xz}} + \lambda \mathbb{I} \right)^{-1}}_{\mathbf{M}^{-1}} \underbrace{\left(\mathbf{K}_{\boldsymbol{xz}}^\top \mathbf{U} \right)}_{\mathbf{N}} \boldsymbol{g}_k \ , \tag{5}$$

where $\mathbb{I} \in \mathbb{R}^{r \times r}$ is the identity matrix, and $\mathbf{K}_{\boldsymbol{xz}} \in \mathbb{R}^{n \times r}$ denotes the kernel matrix computed between $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ and the set of SVs $\boldsymbol{z}$.

[2] Here, for convenience, the bias values b^k are set equal to those of the initial SVMs g_k. In general, they can be jointly optimized with the coefficients $\boldsymbol{\beta}_k$, with minor variations to our subsequent derivations.

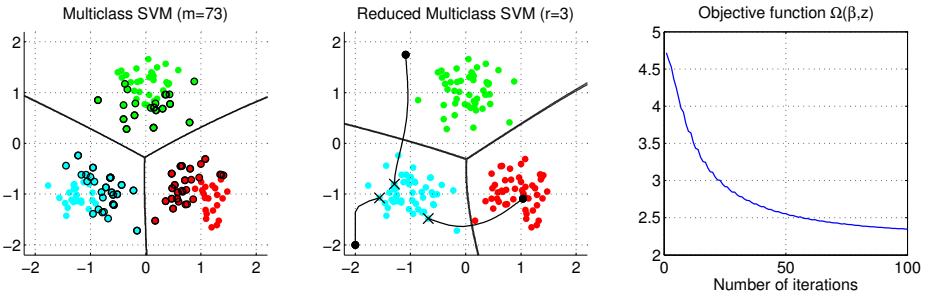

Fig. 1: A two-dimensional classification example with three classes (cyan, green, and red points). *Left:* Decision boundaries (black lines) for the one-vs-all multiclass SVM, that requires $m = 73$ SVs (circled in black). *Middle:* Decision boundaries for our reduced multiclass SVM, using only $r = 3$ SVs (black points). The path followed by each SV during the optimization is also reported (in black), starting from points denoted with '$\times$'. *Right:* Objective function values (Eq. 3) during the minimization process.

***z*-step.** To update $\boldsymbol{z}$, the objective can be minimized through gradient descent (no analytical solution is available). Its gradient with respect to a given $\boldsymbol{z}_j$ is:

$$\frac{\partial \Omega}{\partial \boldsymbol{z}_j} = 2 \sum_{k=1}^{c} (\boldsymbol{h}_k - \boldsymbol{g}_k)^\top \mathbf{U} \left(\beta_j^k \frac{\partial \mathbf{K}_{\boldsymbol{x}\boldsymbol{z}_j}}{\partial \boldsymbol{z}_j} + \mathbf{K}_{\boldsymbol{x}\boldsymbol{z}} \frac{\partial \boldsymbol{\beta}_k}{\partial \boldsymbol{z}_j} \right) + 2\lambda \boldsymbol{\beta}_k^\top \frac{\partial \boldsymbol{\beta}_k}{\partial \boldsymbol{z}_j} \quad , \tag{6}$$

where $\mathbf{K}_{\boldsymbol{x}\boldsymbol{z}_j}$ is the j^{th} column of $\mathbf{K}_{\boldsymbol{x}\boldsymbol{z}}$, and we use the numerator-layout convention for matrix derivatives, *i.e.*, all the derivatives with respect to $\boldsymbol{z}_j$ are vectors or matrices with the same number of columns as the dimensionality of $\boldsymbol{z}_j$. The term $\frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{z}_j}$ can be obtained by deriving Eq. (5) (before inverting $\mathbf{M}$), which yields:

$$\frac{\partial \boldsymbol{\beta}_k}{\partial \boldsymbol{z}_j} = -\mathbf{M}^{-1} \left(\beta_j^k \mathbf{K}_{\boldsymbol{x}\boldsymbol{z}} + \mathbf{S} \right)^\top \mathbf{U} \frac{\partial \mathbf{K}_{\boldsymbol{x}\boldsymbol{z}_j}}{\partial \boldsymbol{z}_j}, \tag{7}$$

where $\mathbf{S}$ is an $n \times r$ matrix of zeros, with the j^{th} column equal to $(\boldsymbol{h}_k - \boldsymbol{g}_k)$.

Gradient of $k(\boldsymbol{x}_i, \boldsymbol{z}_j)$. Our approach can be readily applied to many numeric kernels, as most of them are differentiable. In our experiments, we will use the exponential χ^2 (exp-χ^2) kernel, given as $k(\boldsymbol{x}_i, \boldsymbol{z}_j) = \exp\left(-\gamma \sum_{l=1}^{d} \frac{(x_{il} - z_{jl})^2}{x_{il} + z_{jl}}\right)$, where x_{il} and z_{jl} are the l^{th} feature of $\boldsymbol{x}_i$ and $\boldsymbol{z}_j$, and d is the dimensionality of the input space. It is easy to see that the l^{th} element of the gradient $\frac{\partial k(\boldsymbol{x}_i, \boldsymbol{z}_j)}{\partial \boldsymbol{z}_j}$ is given as $\gamma(x_{il} - z_{jl}) \frac{3x_{il} + z_{jl}}{(x_{il} + z_{jl})^2} k(\boldsymbol{x}_i, \boldsymbol{z}_j)$.

4 Experiments

In this section, we report a set of experiments to show how significantly our RMSVM algorithm can reduce computations required by a kernel-based approach in an image classification scenario. For a fair comparison with current state-of-the-art approaches, we reproduce the image classification setup originally adopted by Xiao *et al.* [24]. The data, the extracted feature values for each

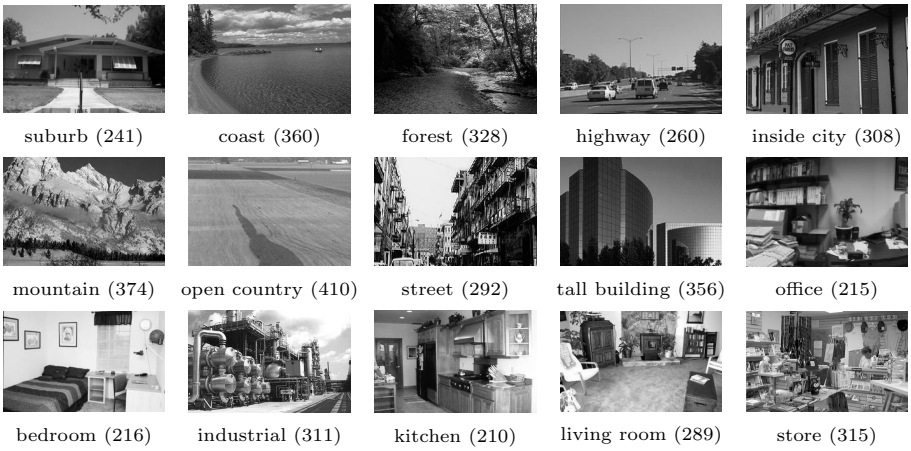

Fig. 2: Example images from the 15-category scenes dataset. We also report the number of available acquisitions for each category.

image, and the training-testing splits are publicly available [9, 24] To implement the multiclass SVM classifier and our regression-based algorithm, we exploit the open-source machine-learning library `scikit-learn` [17]. We test our method by selecting a different number of virtual SVs, fixed in advance (*i.e.*, budgeted).

Dataset. According to [24], we use a widely-used benchmark dataset for image classification [16, 13, 11, 24, 26], *i.e.*, the *15-category scenes dataset*.[3] It consists of fifteen scene categories. Each class has different number of grayscale images, from 200 to 400 acquisitions, with an average size of 300×250 pixels. In Fig. 2 a selection of images of different classes is shown for reference purpose.

Experimental setup. We consider a classification problem where each one-vs-all classifier is trained using a subset of randomly-selected images from each available class, while the remaining ones are used to build the test set. In the training set, the number of samples per category is the same for each class. In the test set, the number of samples per class is different, as it depends on the number of images belonging to each class. Results are averaged over 10 repetitions, considering different training-test pairs.

To compare our results to those obtained in [11, 24], we exploit HoG descriptors as in [8, 24]. Each descriptor consists of 124 feature values, obtained by stacking 2×2 neighboring HoG descriptors each consisting of 31 dimensions. The descriptors extracted from the training images are clustered using the k-means algorithm to identify 300 representative centroids (one per cluster). A histogram of 300 bins is computed from each image. Each bin represents the number of image's descriptors assigned to the corresponding centroid. A number of additional histograms are computed using the same procedure, respectively splitting the image into 2×2 and 4×4 blocks, eventually yielding a total of 21 histograms per image (*i.e.*, $21 \times 300 = 6,300$ features). The exp-χ^2 kernel is

[3] `http://www-cvr.ai.uiuc.edu/ponce_grp/data/`

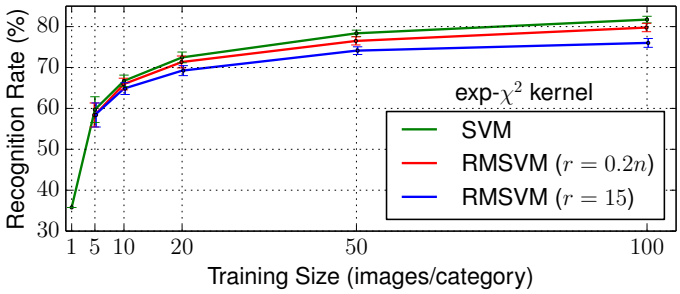

Fig. 3: Recognition rate of the unpruned SVM, and of RMSVM (with different number r of SVs) using the exp-χ^2 kernel on the *15-category scenes dataset*. Results are averaged over 10 repetitions, and reported against an increasing number of samples per class.

used for both SVMs and regressors. According to [24], for each SVM classifier, we set the regularization parameter $C = 1$ and the exp-χ^2 kernel parameter $\gamma = (\frac{1}{n^2}\sum_{i,j}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_j))^{-1}$, yielding $\gamma \approx 0.2$ in each run. The gradient step η and the parameter λ of our RMSVM (see Algorithm 1) are set as $\eta = 0.5$ and $\lambda = 0.1$ by maximizing classification accuracy through a 3-fold cross validation.

Results. Results for the unpruned SVM and RMSVM on the *15-category scenes dataset* are reported in Fig. 3 in terms of recognition rate (*i.e.*, fraction of correctly-classified test images) against an increasing number of training samples per class. For RMSVM, we consider a different number of SVs. In particular, we consider one SV per class (yielding a total of $r = 15$ SVs), and a number of SVs corresponding to the 20% of the training set size ($r = 0.2n$).

It is easy to see that the proposed method for multiclass SVM reduction performs significantly well even when using a very small set of virtual SVs; *e.g.*, in the case of $n = 1500$ training samples (100 images per class), the RMSVM trained with $r = 15$ virtual SVs worsens the recognition rate of less than 5%. This result is exceptionally good considering the extreme reduction rate; in fact, the number of matchings needed is lowered by 100 times. All other reported cases have a proportional behavior, as the number of SVs (*i.e.*, required matchings for classification) found by the standard SVM classifier grows linearly with the training set size [20, 6]. It is worth noting also that the RMSVM with $r = 0.2n$ SVs only worsens the recognition rate of about 1%, while reducing the required number of matchings of 5 times.

In Fig. 3, we also report the performance of the multiclass SVM trained with one image per class (requiring $n = 15$ matchings at test time). Our results show that an equally-sized set of well-principled optimized SVs can significantly outperform a standard SVM; in particular, the RMSVM using only 15 virtual SVs ($r = 15$, synthesized from a larger training set) achieves recognition rates ranging from 60% to 76%, while that of the unpruned SVM is only 36%. While the training complexity of our approach is increased, computational complexity at test time remains unaffected.

	Number of Matchings					
SVM	$m =$	75	150	300	746.7 ± 1.3	1472.4 ± 4.2
RMSVM (20%)	$r =$	15	30	60	150	300

Table 1: Number of matchings required by the unpruned SVM and RMSVM with $r = 0.2n$ SVs, corresponding to the results reported in Fig. 3.

We finally report an analysis of how well the considered algorithms perform on each scene category in the dataset, by reporting the performance of each of the one-vs-all (binary) base classifiers. In particular, in Table 2 we report the Area Under the ROC Curve (AUC) for each category, and for both the SVM and our RMSVMs using a training set of $n = 1500$ samples, averaged over 10 repetitions. Although our method is able to reliably categorize most of the dataset scenes, some categories, like *store* and *industrial* exhibit higher differences in terms of AUC values with respect to the unpruned SVM. This is mainly due to a very high intra-class variability that may not be thoroughly captured by a significantly-reduced set of SVs.

	suburb	coast	open country	forest	highway	inside city	mountain	street	tall building	office	bedroom	industrial	kitchen	living room	store
SVM	100	98.8	99.6	99.0	97.3	99.1	97.1	99.4	98.8	99.8	96.4	94.5	97.7	97.6	97.0
RMSVM_{15}	99.9	98.3	99.5	98.1	96.6	98.7	95.7	98.9	97.9	99.6	94.1	91.5	96.9	96.8	95.4
RMSVM_{300}	100	98.8	99.7	98.7	96.3	98.9	96.3	99.1	98.3	99.7	96.0	90.2	95.3	94.8	95.1

Table 2: Area Under the ROC Curve (AUC %) for each category, using a training set of 100 samples per class. The performance of the unpruned SVM (requiring $m = 1472.4 \pm 4.2$ matchings per classification) is compared to the RMSVMs, respectively budgeted to $r = 15$ and $r = 300$ matchings.

5 Related Work on SVM Reduction

We have proposed a novel reduction method for *multiclass* SVMs by extending a previously-proposed method for reducing the set of SVs in binary SVMs [4]. The latter method turned out to outperform existing reduction methods [19], as it is not greedy: as ours, it iteratively modifies each SV during the optimization process, while the methods in [19] optimize one SV at a time, without modifying it when the remaining SVs change. Moreover, the former approach can also be used when the kernel function $k(\cdot, \cdot)$ does not satisfy the Mercer condition, *i.e.*, it is not a proper (positive semi-definite) kernel, but a generic similarity function, whereas the approaches in [19] are only suitable for definite kernels. There are other versions of reduced SVMs [10, 12, 6], which are however all devoted to the standard binary formulation of this classifier. To our knowledge, the problem of multiclass SVM reduction has only been more systematically investigated in [22]. Despite comparing this approach with ours remains an interesting future development of this work, it is worth remarking that it considers an independent

reduction problem for each binary SVM in the one-vs-all scheme, then it concatenates the resulting sets of SVs, and retrains each binary SVM. Our method, conversely, jointly learns a common set of SVs for *all* the binary SVMs involved.

6 Conclusions and Future Work

The proposed image classification approach allows us to overcome the limitation of high computational complexity at test time, common to multiclass, nonlinear classification tasks that exploit kernel-based or similar methods, by jointly optimizing a unique, small set of virtual SVs along with an optimal set of coefficients for their combination. We have shown that we can *dramatically* speed up the test phase without significantly affecting the recognition rate given by the use of nonlinear (though differentiable) kernel functions, and required by large multicategory datasets. As future developments of this work, we plan to investigate the use of our multiclass reduction algorithm with *non-differentiable* and *indefinite* kernel functions, as already preliminary considered in [4]. This opens interesting research directions, considering that well-known non-differentiable kernels, like the histogram intersection kernel [3], have demonstrated high recognition rates in various image classification tasks. Another potential future investigation regards the application of our method to speed up other non-parametric approaches besides SVMs; in fact, the function $g(\boldsymbol{x})$ in Eq. 3 (and subsequent derivations) is *not* required to be an SVM's discriminant function, but can be *any* discriminant function (or target variable).

Acknowledgments. This work has been partly supported by the project "Advanced and secure sharing of multimedia data over social networks in the future Internet" (CUP F71J11000690002) funded by Regione Autonoma della Sardegna, and by the project "Computational quantum structures at the service of pattern recognition: modeling uncertainty" (CRP-59872) funded by Regione Autonoma della Sardegna, L.R. 7/2007, Bando 2012.

References

1. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Good practice in large-scale learning for image classification. IEEE Trans. Pattern Anal. Mach. Intell. 36(3), 507–520 (2014)
2. Barla, A., Franceschi, E., Odone, F., Verri, A.: Image kernels. In: Lee, S.W., Verri, A. (eds.) Pattern Recognition with Support Vector Machines. LNCS, vol. 2388, pp. 83–96. Springer (2002)
3. Barla, A., Odone, F., Verri, A.: Histogram intersection kernel for image classification. In: Int'l Conf. Image Processing (ICIP). pp. 513–516 (2003)
4. Biggio, B., Melis, M., Fumera, G., Roli, F.: Sparse support faces. In: Int'l Conf. Biometrics (ICB). pp. 1–6 (2015)
5. Bosch, A., Muñoz, X., Marti, R.: Which is the best way to organize/classify images by content? Image Vision Comput. 25(6), 778–791 (2007)
6. Chapelle, O.: Training a support vector machine in the primal. Neural Comput. 19(5), 1155–1178 (2007)

7. Chapelle, O., Haffner, P., Vapnik, V.: Support vector machines for histogram-based image classification. IEEE Trans. on Neural Networks 10(5), 1055–1064 (1999)
8. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conf. Computer Vision and Pattern Recognition (CVPR). pp. 886–893 (2005)
9. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Patt. Anal. Mach. Intell. 32(9), 1627–1645 (2010)
10. Keerthi, S.S., Chapelle, O., DeCoste, D.: Building support vector machines with reduced classifier complexity. J. Mach. Learn. Res. 7, 1493–1515 (2006)
11. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 2169–2178 (2006)
12. Lee, Y.J., Mangasarian, O.L.: RSVM: Reduced support vector machines. In: SDM. vol. 1, pp. 325–361 (2001)
13. Li, F.F., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: IEEE Conf. Computer Vision and Pattern Recognition (CVPR). pp. 524–531 (2005)
14. Lin, Y., Lv, F., Zhu, S., Yang, M., Cour, T., Yu, K., Cao, L., Huang, T.S.: Large-scale image classification: Fast feature extraction and SVM training. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 1689–1696 (2011)
15. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int'l Journal of Computer Vision 60(2), 91–110 (2004)
16. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. Int'l Journal of Computer Vision 42(3), 145–175 (2001)
17. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830 (2011)
18. Rifkin, R.M., Klautau, A.: In defense of one-vs-all classification. J. Mach. Learn. Res. 5, 101–141 (2004)
19. Schölkopf, B., Mika, S., Burges, C.J.C., Knirsch, P., Muller, K.R., Rätsch, G., Smola, A.J.: Input space versus feature space in kernel-based methods. IEEE Trans. Neural Networks 10(5), 1000–1017 (1999)
20. Steinwart, I.: Sparseness of support vector machines. J. Mach. Learn. Res. 4, 1071–1105 (2003)
21. Suhr, J.K., Jung, H.G.: Sensor fusion-based vacant parking slot detection and tracking. IEEE Trans. on Intelligent Transportation Systems 15(1), 21–36 (2014)
22. Tang, B., Mazzoni, D.: Multiclass reduced-set support vector machines. In: Proc. Int'l Conf. Machine Learning. pp. 921–928. ICML '06, ACM, New York, NY, USA (2006)
23. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
24. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: SUN database: Large-scale scene recognition from abbey to zoo. In: IEEE Conf. Computer Vision and Pattern Recognition (CVPR). pp. 3485–3492 (2010)
25. Zhang, L., Lin, F., Zhang, B.: Support vector machine learning for image retrieval. In: Int'l Conf. Image Processing (ICIP). pp. 721–724 (2001)
26. Zhou, L., Zhou, Z., Hu, D.: Scene classification using multi-resolution low-level feature combination. Neurocomputing 122, 284–297 (2013)