Image Classification Using Super-Vector Coding of Local Image Descriptors

Xi Zhou[1], Kai Yu[2], Tong Zhang[3], and Thomas S. Huang[1]

[1] Dept. of ECE, University of Illnois at Urbana-Champaign
[2] NEC Laboratories America, Cupertino, CA
[3] Department of Statistics, Rutgers University

Abstract. This paper introduces a new framework for image classification using local visual descriptors. The pipeline first performs a nonlinear feature transformation on descriptors, then aggregates the results together to form image-level representations, and finally applies a classification model. For all the three steps we suggest novel solutions which make our approach appealing in theory, more scalable in computation, and transparent in classification. Our experiments demonstrate that the proposed classification method achieves state-of-the-art accuracy on the well-known PASCAL benchmarks.

1 Introduction

Image classification, including object recognition and scene classification, remains to be a major challenge to the computer vision community. Perhaps one of the most significant developments in the last decade is the application of *local features* to image classification, including the introduction of "bag-of-visual-words" representation that inspires and initiates a lot of research efforts [1].

A large body of work investigates probabilistic generative models, with the objective towards understanding the semantic content of images. Typically those models extend the famous topic models on bag-of-word representation by further considering the spatial information of visual words [2][3].

This paper follows another line of research on building discriminative models for classification. The previous work includes SVMs using pyramid matching kernels [4], biologically-inspired models [5][6], and KNN methods [7][8][9]. Over the past years, the nonlinear SVM method using spatial pyramid matching (SPM) kernels [4][10] seems to be dominant among the top performers in various image classification benchmarks, including Caltech-101 [11], PASCAL [12], and TRECVID. The recent improvements were often achieved by combining different types of local descriptors [10][13][14], without any fundamental change of the underlying classification method. In addition to the demand for more accurate classifiers, one has to develop more practical methods. Nonlinear SVMs scale at least quadratically to the size of training data, which makes it nontrivial to handle large-scale training data. It is thus necessary to design algorithms that are computationally more efficient.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part V, LNCS 6315, pp. 141–154, 2010.

1.1 Overview of Our Approach

Our work represents each image by a set of local descriptors with their spatial coordinates. The descriptor can be SIFT, or any other local features, computed from image patches at locations on a 2D grid. Our image classification method consists of three computational steps:

1. *Descriptor coding*:
   Each descriptor of an image is nonlinearly mapped to form a high-dimensional sparse vector. We propose a novel nonlinear coding method called Super-Vector coding, which is algorithmically a simple extension of Vector Quantization (VQ) coding;

2. *Spatial pooling*:
   For each local region, the codes of all the descriptors in it are aggregated to form a single vector, then vectors of different regions are concatenated to form the image-level feature vector. Our pooling is base on a novel probability kernel incorporating the similarity metric of local descriptors;

3. *Image classification*:
   The image-level feature vector is normalized and fed into a classifier. We choose linear SVMs, which scale linearly to the size of training data.

We note that the *coding-pooling-classification* pipeline is the *de facto* framework for image scene classification. One notable example is the SPM kernel approach [4], which applies average pooling on top of VQ coding, plus a nonlinear SVM classifier using Chi-square or intersection kernels.

In this paper, we propose novel methods for each of the three steps and formalize their underlying mathematical principles. The work stresses the importance of learning good coding of local descriptors in the context of image classification, and makes the first attempt to formally incorporate the metric of local descriptors into distribution kernels. Putting all these together, the overall image classification framework enjoys a linear training complexity, and also a great interpretability that is missing in conventional models (see details in Sec. 2.3). The most importantly, our method demonstrates *state-of-the-art* performances on the challenging PASCAL07 and PASCAL09 image classification benchmarks.

2 The Method

In the following we will describe all the three steps of our image classification pipeline in detail.

2.1 Descriptor Coding

We introduce a novel coding method, which enjoys appealing theoretical properties. Suppose we are interested in learning a smooth nonlinear function $f(x)$

defined on a high dimensional space $\mathbb{R}^d$. The question is, how to derive a good coding scheme (or nonlinear mapping) $\phi(x)$ such that $f(x)$ can be well approximated by a *linear function* on it, namely $w^\top \phi(x)$. Our only assumption here is that $f(x)$ should be sufficiently smooth.

Let us consider a general unsupervised learning setting, where a set of bases $C \subset \mathbb{R}^d$, called *codebook* or *dictionary*, is employed to approximate any x, namely,

$$x \approx \sum_{v \in C} \gamma_v(x) v,$$

where $\gamma(x) = [\gamma_v(x)]_{v \in C}$ is the coefficients, and sometimes $\sum_v \gamma_v(x) = 1$. By restricting the cardinality of nonzeros of $\gamma(x)$ to be 1 and $\gamma_v(x) \geq 0$, we obtain the Vector Quantization (VQ) method

$$v_*(x) = \arg\min_{v \in C} \|x - v\|,$$

where $\|\cdot\|$ is the Euclidean norm (2-norm). The VQ method uses the coding $\gamma_v(x) = 1$ if $v = v_*(x)$ and $\gamma_v(x) = 0$ otherwise. We say that $f(x)$ is β Lipschitz derivative smooth if for all $x, x' \in \mathbb{R}^d$:

$$|f(x) - f(x') - \nabla f(x')^\top (x - x')| \leq \frac{\beta}{2} \|x - x'\|^2.$$

It immediately implies the following simple function approximation bound via VQ coding: for all $x \in \mathbb{R}^d$:

$$\left| f(x) - f\big(v_*(x)\big) - \nabla f\big(v_*(x)\big)^\top \big(x - v_*(x)\big) \right| \leq \frac{\beta}{2} \left\| x - v_*(x) \right\|^2. \tag{1}$$

This bounds simply states that one can approximate $f(x)$ by $f\big(v_*(x)\big) + \nabla f\big(v_*(x)\big)^\top \big(x - v_*(x)\big)$, and the approximation error is upper bounded by the quality of VQ. It further suggests that the function approximation can be improved by learning the codebook C to minimize this upper bound. One way is the K-means algorithm

$$C = \arg\min_C \left\{ \sum_x \min_{v \in C} \|x - v\|^2 \right\}.$$

Eq. (1) also suggests that the approximation to $f(x)$ can be expressed as a linear function on a nonlinear coding scheme

$$f(x) \approx g(x) \equiv w^\top \phi(x),$$

where $\phi(x)$ is called the *Super-Vector* (SV) coding of x, defined by

$$\phi(x) = \left[s\gamma_v(x), \gamma_v(x)(x - v)^\top \right]_{v \in C}^\top \tag{2}$$

where s is a nonnegative constant. It is not difficult to see that $w = [\frac{1}{s} f(v), \nabla f(v)]_{v \in C}$, which can be regarded as unknown parameters to be estimated. Because $\gamma_v(x) = 1$ if $v = v_*(x)$, otherwise $\gamma_v(x) = 0$, the obtained $\phi(x)$

a is highly sparse representation, with dimensionality $|C|(d+1)$. For example, if $|C| = 3$ and $\gamma(x) = [0, 1, 0]$, then

$$\phi(x) = \Big[\underbrace{0, \ldots, 0}_{d+1 \text{ dim.}}, \underbrace{s, (x-v)^\top}_{d+1 \text{ dim.}}, \underbrace{0, \ldots, 0}_{d+1 \text{ dim.}}\Big]^\top \tag{3}$$

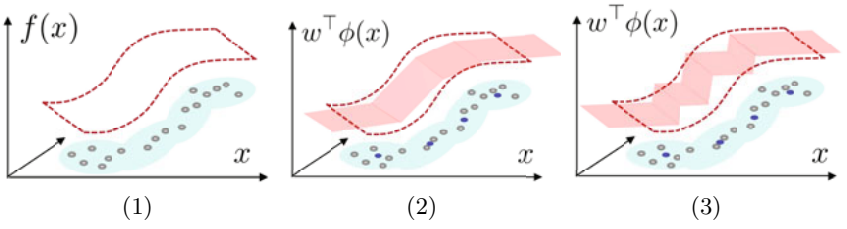

Fig. 1. Function $f(x)$ approximated by $w^\top\phi(x)$

As illustrated in Figure 1, $w^\top\phi(x)$ provides a piece-wise linear function to approximate a nonlinear function $f(x)$, as shown in Figure 1-(2), while with VQ coding $\phi(x) = [\gamma_v(x)]^\top_{v\in C}$, the same formulation $w^\top\phi(x)$ gives a piece-wise constant approximation, as shown in Figure 1-(3). This intuitively suggests that SV coding may achieve a lower function approximation error than VQ coding. We note that the popular bag-of-features image classification method essentially employs VQ to obtain histogram representations. The proposed SV coding is a simple extension of VQ, and may lead to a better approach to image classification.

2.2 Spatial Pooling

Pooling. Let each image be represented as a set of descriptor vectors x that follows an image-specific distribution, represented as a probability density function $p(x)$ with respect to an image independent back-ground measure $d\mu(x)$. Let's first ignore the spacial locations of x, and address the spacial pooling later. A kernel-based method for image classification is based on a kernel on the probability distributions over $x \in \Omega$, $K : \mathcal{P} \times \mathcal{P} \mapsto \mathbb{R}$. A well-known example is the Bhattacharyya kernel [15]:

$$K_b(p, q) = \int_\Omega p(x)^{\frac{1}{2}} q(x)^{\frac{1}{2}} d\mu(x).$$

Here $p(x)$ and $q(x)$ represent two images as distributions over local descriptor vectors, and $\mu(x)$ is the image independent background measure. Bhattacharyya kernel is closely associated with Hellinger distance, defined as $D_h(p, q) = 2 - K_b(p, q)$, which can be seen as a principled symmetric approximation of the

Kullback Leibler (KL) divergence [15]. Despite the popular application of both Bhattacharyya kernel and KL divergence, a significant drawback is the ignorance of the underlying similarity metric of x, as illustrated in Figure 2. In order to avoid this problem, one has to work with very smooth distribution families that are inconvenient to work with in practice. In this paper, we propose a novel formulation that explicitly takes the similarity of x into account:

$$\begin{aligned} K_s(p,q) &= \int_\Omega \int_\Omega p(x)^{\frac{1}{2}} q(x')^{\frac{1}{2}} \kappa(x,x') d\mu(x) d\mu(x') \\ &= \int_\Omega \int_\Omega p(x)^{-\frac{1}{2}} q(x')^{-\frac{1}{2}} \kappa(x,x') p(x) q(x') d\mu(x) d\mu(x') \end{aligned}$$

where $\kappa(x,x')$ is a RKHS kernel on Ω that reflects the similarity structure of x. In the extreme case where $\kappa(x,x') = \delta(x-x')$ is the delta-function with respect to $\mu(\cdot)$, then the above kernel reduces to the Bhattacharyya kernel.

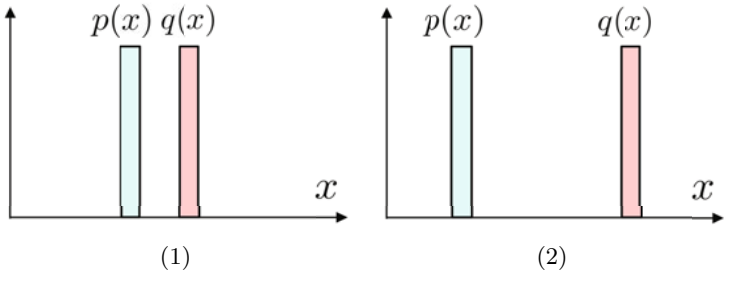

Fig. 2. Illustration of the drawback of Bhattacharyya kernel: in both cases their density kernels $K_b(p,q)$ remain to be the same, equal to 0

In reality we cannot directly observe $p(x)$ from any image, but a set X of local descriptors. Therefore, based on the empirical approximation to $K_s(p,q)$, we define a kernel between sets of vectors:

$$K(X,X') = \frac{1}{NN'} \sum_{x \in X} \sum_{x' \in X'} p(x)^{-\frac{1}{2}} q(x')^{-\frac{1}{2}} \kappa(x,x') \tag{4}$$

where N and N' are the sizes of the descriptor sets from two images.

Let $\kappa(x,x') = \langle \phi(x), \phi(x') \rangle$, where $\phi(x)$ is the SV coding defined in the previous section. It is easy to see that $\kappa(x,x') = 0$ if x and x' fall into different clusters. Then we have

$$K(X,X') = \frac{1}{NN'} \sum_{k=1}^{|C|} \sum_{x \in X_k} \sum_{x' \in X'_k} p(x)^{-\frac{1}{2}} q(x')^{-\frac{1}{2}} \kappa(x,x')$$

where X_k is the subset of X fallen into the k-th cluster. Furthermore, if we assume that $p(x)$ remains constant within each cluster partition, i.e., $p(x)$ gives rise to a histogram $[p_k]_{k=1}^{|C|}$, then

$$K(X, X') = \frac{1}{NN'} \sum_{k=1}^{|C|} \left\langle \frac{1}{\sqrt{p_k}} \sum_{x \in X_k} \phi(x), \frac{1}{\sqrt{q_k}} \sum_{x' \in X'_k} \phi(x') \right\rangle$$

The above kernel can be re-written as an inner product kernel of the form $K(X, X') = \langle \Phi(X), \Phi(X') \rangle$, where

$$\Phi(X) = \frac{1}{N} \sum_{k=1}^{|C|} \frac{1}{\sqrt{p_k}} \sum_{x \in X_k} \phi(x).$$

Therefore functions in the reproducing kernel Hilbert space for this kernel has a linear representation $f(X) = w^\top \Phi(X)$. In other words, we can simply employ $\Phi(X)$ as nonlinear feature vector and then learn a linear classifier using this feature vector. The effect is equivalent to using nonlinear kernel $K(X, X')$ between image pairs X and X'.

Finally, we point out that weighting by histogram p_k is equivalent to treating density $p(x)$ as piece-wise constant around each VQ basis, under a specific choice of background measure $\mu(x)$ that equalizes different partitions. This representation is not sensitive to the choice of background measure $\mu(x)$, which is image independent. In particular, a change of measure $\mu(\cdot)$ (still piece-wise constant in each partition) leads to a rescaling of different components in $\Phi(X)$. This means that the space of linear classifier $f(x) = w^\top \Phi(X)$ remains the same.

Spatial Pyramid Pooling. To incorporate the spatial location information of x, we apply the idea of spatial pyramid matching [4]. Let each image be evenly partitioned into 1×1, 2×2, and 3×1 blocks, respectively in 3 different levels. Based on which block each descriptor comes from, the whole set X of an image is then organized into three levels of subsets: X_{11}^1, X_{11}^2, X_{12}^2, X_{21}^2, X_{22}^2, X_{11}^3, X_{12}^3, and X_{13}^3. Then we apply the pooling operation introduced in the last subsection to each of the subsets. An image's spacial pyramid representation is then obtained by concatenating the results of local pooling

$$\Phi_s(X) = \left[\Phi(X_{11}^1), \Phi(X_{11}^2), \Phi(X_{12}^2), \Phi(X_{21}^2), \Phi(X_{22}^2), \Phi(X_{11}^3), \Phi(X_{12}^3), \Phi(X_{13}^3)\right]$$

2.3 Image Classification

Image classification is done by applying classifiers based on the image representations obtained from the pooling step. Here we consider the task of finding whether a particular category of objects is contained in an image or not, which can be translated into a binary classification problem. We apply a *linear* SVM that employs a hinge loss to learn $g(X) = w^\top \Phi_s(X)$. We note that the function is nonlinear on X since $\Phi_s(X)$ is a nonlinear operator.

Interestingly, the image-level classification function is closely connected to a real-valued function on local descriptors. Without loss of generality, let's assume that only global pooling is used, which means $\Phi_s(X) = \Phi(X)$ in this case.

$$g(X) = w^\top \Phi(X) = \frac{1}{N} \sum_{k=1}^{|C|} \frac{1}{\sqrt{p_k}} \sum_{x \in X_k} w^\top \phi(x) = \frac{1}{N} \sum_{k=1}^{|C|} \frac{1}{\sqrt{p_k}} \sum_{x \in X_k} g(x) \qquad (5)$$

where $g(x) = w^\top \phi(x)$. The above equation provides an interesting insight to the classification process: a patch-level pattern matching is operated everywhere in the image, and the responses are then aggregated together to generate the score indicating how likely a particular category of objects is present. This observation is well-aligned with the biologically-inspired vision models, like Convolution Neural Networks [16] and HMAX model [6], which mostly employ feed-forward pattern matching for object recognition.

This connection stresses the importance of learning a good coding scheme on local descriptors x, because $\phi(x)$ solely defines the function space of $g(x) = w^\top \phi(x)$, which consequently determines if the unknown classification function can be well learned. The connection also implies that supervised training of $\phi(x)$ could potentially lead to further improvements.

Furthermore, the classification model enjoys the advantages of interpretability and computational scalability. Once the model is trained, Eq. (5) suggests that one can compute a response map based on $g(x)$, which visualizes where the classifier focuses on in the image, as shown in our experiments. Since our method naturally requires a linear classifier, it enjoys a training scalability which is linear to the number of training images, while nonlinear kernel-based methods suffer quadratic or higher complexity.

3 Discussion and Further Improvement

Our approach is along the line of recent works on unsupervised feature learning for image classfiication, especially, learning *sparse representations* e.g., [17][5][18] [19] [20]. In theory our work is more related to local coordinate coding (LCC) [19], which points out that in some cases a desired sparsity of $\phi(x)$ should come from a *locality* of the coding scheme. Indeed, the proposed SV coding leads to a highly sparse representation $\phi(x)$, as defined by Eq. (2), which activates those coordinates associated to the neighborhood of x. As the result, $g(x) = w^\top \phi(x)$ gives rise to a local linear function (i.e., piece-wise linear) to approximate the unknown nonlinear function $f(x)$. But, the computation of SV coding is much simpler than sparse coding approaches.

Our method can be further improved by considering a soft assignment of x to bases C. Recall that the underlying interpretation of $f(x) \approx w^\top \phi(x)$ is the the approximation

$$f(x) \approx f\big(v_*(x)\big) + \nabla f\big(v_*(x)\big)^\top \big(x - v_*(x)\big)$$

which essentially uses the unknown function's Taylor expansion at a nearby location $v_*(x)$ to interpolate $f(x)$. One natural idea to improve this is using several neighbors in C instead of the nearest one. Let's consider a soft K-means that computes $p_k(x)$, the posterior probability of cluster assignment for x. Then the function approximation can be handled as the expectation

$$f(x) \approx \sum_{k=1}^{|C|} p_k(x) \Big[f\big(v_k\big) + \nabla f\big(v_k\big)^\top \big(x - v_k\big) \Big]$$

Then the pooling step becomes a computation of the expectation

$$\Phi(X) = \frac{1}{N} \left[\frac{1}{\sqrt{p_k}} \sum_{x \in X} p_k(x)(x - v_k + s) \right]_{k=1}^{|C|}$$

where $p_k = \frac{1}{N} \sum_{x \in X} p_k(x)$, and s comes from Eq. (2). This approach is different from the image classification using GMM, e.g., [21][22]. Basically, those GMM methods consider the distribution kernel, while ours incorporates nonlinear coding into the distribution kernel. Furthermore, our theory requires the stickiness to VQ – the soft version requires all the components share the same *isotropic* diagonal covariance. That means a much less number of parameters

Table 1. Comparison of different coding methods, on PASCAL VOC 2007 test set

AP (%)	VQ	GMM	SV	SV-soft
aeroplane	39.9	74.4	77.5	78.9
bicycle	44.0	57.9	67.2	68.4
bird	27.7	45.7	47.0	51.9
boat	53.8	68.9	73.9	71.5
bottle	15.8	26.2	27.2	29.8
bus	48.5	63.0	66.9	70.3
car	63.4	77.2	81.4	81.6
cat	38.6	54.6	61.1	60.2
chair	45.8	53.0	53.7	54.5
cow	27.4	42.7	49.3	48.2
dining_table	32.7	46.9	55.1	56.8
dog	36.0	43.1	44.6	44.9
horse	66.7	77.7	77.7	80.8
motorbike	43.6	60.2	66.2	68.8
person	73.1	83.6	84.8	85.9
potted_plant	25.9	28.2	28.5	29.6
sheep	22.8	42.3	46.7	47.7
sofa	41.9	51.2	56.1	57.7
train	60.0	75.6	79.2	81.7
tv/monitor	27.0	44.1	51.1	52.9
average	41.7	55.8	59.8	61.1

to estimate. Our experiment confirms that our approach leads to a significantly higher accuracy.

4 Experiments

We perform image classification experiments on two datasets: PASCAL VOC 2007 and PASCAL VOC 2009. The images in both datasets contain objects from 20 object categories and range between indoor and outdoor scenes, close-ups and landscapes, and strange viewpoints. The datasets are extremely challenging because of significant variations of appearances and poses with frequent occlusions. PASCAL VOC 2007 consists of 9,963 images which are divided into three subsets: training data (2501 images), validation data (2510 images), and test data (4952 images). PASCAL VOC 2009 consists of 14,743 images and correspondingly are divided into three subsets: training data(3473 images), validation data(3581 images), and testing data (7689 images).

All the following experiment results are obtained on the testing datasets, except the comparison experiment for different codebook sizes $|C|$ (Table 4), which is performed on PASCAL VOC 2007 validation set. We use the PASCAL toolkit to evaluate the classification accuracy, measured by average precision based on the precision/recall curve.

Table 2. Comparison of our method with top performers in PASCAL VOC 2007

AP (%)	QMUL	TKK	XRCE	INRIA(flat)	INRIA(GA)	Ours
aeroplane	71.6	71.4	72.3	74.8	77.5	**79.4**
bicycle	55.0	51.7	57.5	62.5	63.6	**72.5**
bird	41.1	48.5	53.2	51.2	**56.1**	55.6
boat	65.5	63.4	68.9	69.4	71.9	**73.8**
bottle	27.2	27.3	28.5	29.2	33.1	**34.0**
bus	51.1	49.9	57.5	60.4	60.6	**72.4**
car	72.2	70.1	75.4	76.3	78.0	**83.4**
cat	55.1	51.2	50.3	57.6	58.8	**63.6**
chair	47.4	51.7	52.2	53.1	53.5	**56.6**
cow	35.9	32.3	39.0	41.1	42.6	**52.8**
dining_table	37.4	46.3	46.8	54.9	54.9	**63.2**
dog	41.5	41.5	45.3	42.8	45.8	**49.5**
horse	71.5	72.6	75.7	76.5	77.5	**80.9**
motorbike	57.9	60.2	58.5	62.3	64.0	**71.9**
person	80.8	82.2	84.0	84.5	**85.9**	85.1
potted_plant	15.6	31.7	32.6	36.3	36.3	**36.4**
sheep	33.3	30.1	39.7	41.3	44.7	**46.5**
sofa	41.9	39.2	50.9	50.1	50.6	**59.8**
train	76.5	71.1	75.1	77.6	79.2	**83.3**
tv/monitor	45.9	41.0	49.5	49.3	53.2	**58.9**
average	51.2	51.7	55.6	57.5	59.4	**64.0**

Table 3. Comparison of our method with top performers in PASCAL VOC 2009

AP (%)	LEOBEN	LIP6	LEAR	FIRSTNIKON	CVC	UVASURREY	OURS
aeroplane	79.5	80.9	79.5	83.3	86.3	84.7	**87.1**
bicycle	52.1	52.3	55.5	59.3	60.7	63.9	**67.4**
bird	57.2	53.8	54.5	62.7	**66.4**	66.1	65.8
boat	59.9	60.8	63.9	65.3	65.3	67.3	**72.3**
bottle	29.3	29.1	**43.7**	30.2	41.0	37.9	40.9
bus	63.5	66.2	70.3	71.6	71.7	74.1	**78.3**
car	55.1	53.4	66.4	58.2	64.7	63.2	**69.7**
cat	53.9	55.9	56.5	62.2	63.9	64.0	**69.7**
chair	51.1	50.7	54.4	54.3	55.5	57.1	**58.5**
cow	31.3	33.8	38.8	40.7	40.1	46.2	**50.1**
dining_table	42.9	43.9	44.1	49.2	51.3	54.7	**55.1**
dog	44.1	44.6	46.2	50.0	45.9	53.5	**56.3**
horse	54.8	59.4	58.5	66.6	65.2	68.1	**71.8**
motorbike	58.4	58	64.2	62.9	68.9	70.6	**70.8**
person	81.1	80.0	82.2	83.3	85.0	**85.2**	84.1
potted_plant	30.0	25.3	39.1	34.2	**40.8**	38.5	31.4
sheep	40.2	41.9	41.3	48.2	49	47.2	**51.5**
sofa	44.2	42.5	39.8	46.1	49.1	49.3	**55.1**
train	74.9	78.4	73.6	83.4	81.8	83.2	**84.7**
tv/monitor	58.2	60.1	66.2	65.5	**68.6**	68.1	65.2
average	53.1	53.6	56.9	58.9	61.1	62.1	**64.3**

In all the experiments, 128-dimensional SIFT vectors are extracted over a grid with spacing of 4 pixels on three patch scales (16x16,25x25 and 31x31). The dimension of descriptors is reduced to 80 by applying principal component analysis (PCA). The codebooks C are trained on one million randomly sampled descriptors. The constant s is chosen from $[0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$ via cross-validation on the training set.

4.1 Comparison of Nonlinear Coding Methods

Our first experiment investigates image classification using various nonlinear coding methods. The goal is to study which coding method performs the best under linear SVM classifiers. These methods are: (1) VQ coding – using Bhattacharyya kernel on spatial pyramid histogram presentations; (2) GMM – the method described in [22]; (3) SV – the super-vector coding proposed by this paper; (4) SV-soft – the soft version of SV coding, where $[p_k(x)]_k$ for each x is truncated to retain the top 20 elements with the rest elements being set zero.

Table 1 shows the experiment results with different coding methods on PASCAL VOC 2007 test dataset. In all the cases $|C| = 512$ bases/components are used for coding. SV and SV-soft both significantly outperform other two competitors. SV-soft is slightly better than SV. In the rest of the experiments we apply SV-soft for classification.

Table 4. The influence of codebook sizes $|C|$, on PASCAL VOC 2007 validation set

AP (%)	$\|C\|$ =256	$\|C\|$ =512	$\|C\|$ =1024	$\|C\|$ =2048
aeroplane	77.7	77.9	77.9	78.7
bicycle	55.6	57.2	58.2	58.7
bird	51.0	53.5	54.4	54.0
boat	66.3	66.9	67.1	68.9
bottle	25.5	29.8	31.5	31.9
bus	56.2	59.7	60.9	60.0
car	78.8	79.6	79.8	80.5
cat	59.5	61.4	62.3	62.4
chair	56.4	56.6	56.8	58.0
cow	40.0	43.6	45.6	44.3
dining_table	52.7	58.8	61.1	60.7
dog	42.3	46.5	48.7	47.1
horse	72.5	72.1	72.2	74.4
motorbike	65.7	68.7	70.1	70.5
person	79.8	81.0	81.6	81.7
potted_plant	23.3	22.9	22.5	23.2
sheep	30.2	33.9	35.5	32.0
sofa	52.2	54.7	55.9	57.3
train	80.2	81.2	81.4	82.5
tv/monitor	55.0	56.4	57.2	57.9
average	56.0	58.1	59.0	59.2

4.2 Comparison with State-of-the-Art Results

In this section we compare the performance of our method with reported state-of-the-art results on the PASCAL VOC 2007 and 2009 benchmarks. In both cases, we train the classifier on the training set plus the validation set, and evaluate on the test set, with $|C|$ fixed as 2048. Table 2 compares the experiment results by our approach with the top performances in PASCAL VOC 2007 dataset,[1] while Table 3 compares our results with the top results in PASCAL VOC 2009 dataset.[2] In both cases, our method significantly outperforms the competing methods on most of the object categories. We note that most of those compared methods extend the SPM nonlinear SVM classifier by combing multiple visual descriptors/kernels, while our method utilizes only SIFT features on gray images. This difference highlights the significant success of the proposed approach. Note that in Table 3 we do not compare with the winner team NEC-UIUC's result, because as far as we know, they combined an object detection model, i.e. using the information of the provided bounding boxes, to achieve a higher accuracy.

[1] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/workshop/everingham_cls.pdf

[2] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2009/workshop/everingham_cls.pdf

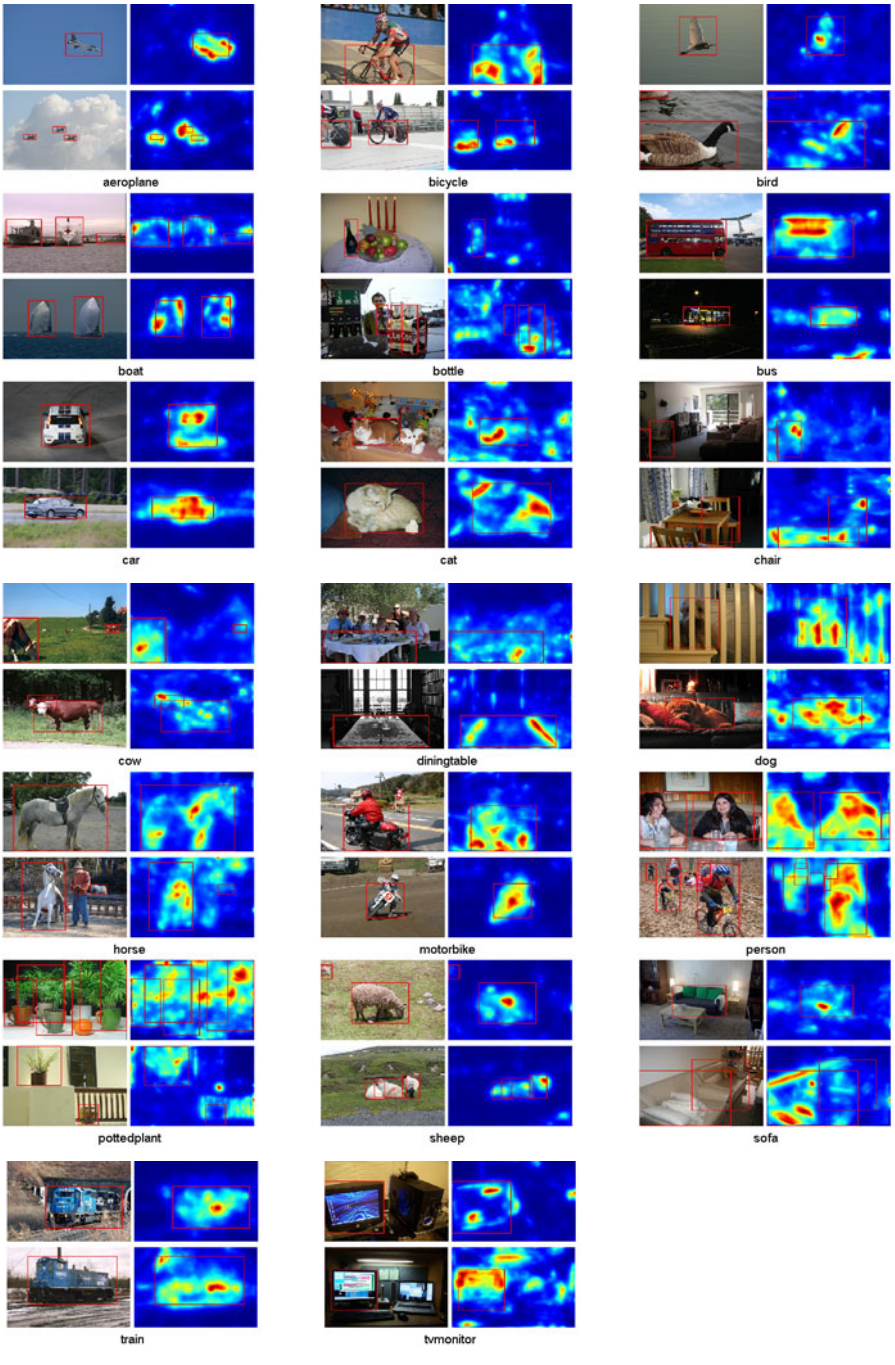

Fig. 3. Visualization of the learned patch-level function $g(x)$ on image examples from PASCAL-09. The relationship between $g(x)$ and the image classification function $g(X)$ is shown in Eq. 5. The figures show that $g(x)$ has a good potential for object detection.

4.3 Impact of Codebook Size

In this section we report further experimental results on PASCAL VOC 2007 validation set, to show the impact of codebook size $|C|$ on classification performance. As shown in Table 4, as we increase $|C|$ from 256, to 512, 1024, and 2048, the classification accuracy keeps being improved. But the improvement gets small after $|C|$ goes over 1024.

4.4 Visualization of the Learned Patch-Level Function

As suggested by Eq. 5, a very unique perspective of our method is the "transparency" of the classification model. Once the image classifier is trained, a real-valued function $g(x)$ is automatically obtained on the local descriptor level. Therefore a response map of $g(x)$ can be visualized on test images. In Figure 3, we show the response map (with kernel smoothing) on a set of random images from the PASCAL VOC 2009 test set. In most of the cases, the results are quite meaningful – the target objects are mostly covered by high-valued responses of $g(x)$. This observation suggests a potential to extend the current framework toward joint classification and detection.

5 Conclusion

This paper introduces a new method for image classification. The method follows the usual pipeline but introduces significantly novel methods for each of the steps. We formalizes the underlying mathematic principles for our methods and stresses the importance of learning a good coding of local descriptors in image classification. Compared to popular state-of-the-art methods, our approach is appealing in theory, more scalable in computation, transparent in classification, and produces state-of-the-art accuracy on the well-known PASCAL benchmark.

Acknowledgments. The main part of this work was done when the first author was a summer intern at NEC Laboratories America, Cupertino, CA.

References

1. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV, vol. 1, p. 22 (2004) (Citeseer)
2. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories (2005) (Citeseer)
3. Sivic, J., Russell, B., Efros, A., Zisserman, A., Freeman, W.: Discovering object categories in image collections. In: Proc. ICCV, vol. 2 (2005)
4. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories (2006) (Citeseer)
5. MarcAurelio Ranzato, F., Boureau, Y., LeCun, Y.: Unsupervised learning of invariant feature hierarchies with applications to object recognition. In: Proc. Computer Vision and Pattern Recognition Conference (CVPR 2007) (2007) (Citeseer)

6. Serre, T., Wolf, L., Poggio, T.: Object recognition with features inspired by visual cortex. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, p. 994 (2005) (Citeseer)
7. Zhang, H., Berg, A., Maire, M., Malik, J.: SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In: Proc. CVPR, vol. 2, pp. 2126–2136 (2006) (Citeseer)
8. Makadia, A., Pavlovic, V., Kumar, S.: A new baseline for image annotation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 316–329. Springer, Heidelberg (2008)
9. Torralba, A., Fergus, R., Weiss, Y.: Small codes and large image databases for recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8 (2008)
10. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: Proceedings of the 6th ACM international conference on Image and video retrieval, p. 408. ACM, New York (2007)
11. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer Vision and Image Understanding 106, 59–70 (2007)
12. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes (VOC) Challenge. International Journal of Computer Vision (2009)
13. Varma, M., Ray, D.: Learning the discriminative power-invariance trade-off. In: Proc. ICCV, vol. 2007 (2007) (Citeseer)
14. Marszalek, M., Schmid, C., Harzallah, H., Weijer, J.V.D.: Learning object representations for visual object class recognition. In: Visual Recognition Challange workshop, in conjunction with ICCV (2007)
15. Jebara, T., Kondor, R.: Bhattacharyya and expected likelihood kernels. In: Proceedings of Learning theory and Kernel machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, p. 57. Springer, Heidelberg (2003)
16. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324 (1998)
17. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.: Self-taught learning: Transfer learning from unlabeled data. In: Proceedings of the 24th international conference on Machine learning, p. 766. ACM, New York (2007)
18. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
19. Yu, K., Zhang, T., Gong, Y.: Nonlinear Learning using Local Coordinate Coding. In: NIPS (2009)
20. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Supervised dictionary learning. Adv. NIPS 21 (2009)
21. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: Proc. CVPR (2006) (Citeseer)
22. Zhou, X., Cui, N., Li, Z., Liang, F., Huang, T.: Hierarchical Gaussianization for Image Classification. In: ICCV (2009)