# Efficient Highly Over-Complete Sparse Coding using a Mixture Model

Jianchao Yang[1], Kai Yu[2], and Thomas Huang[1]

[1] Beckman Institute, University of Illinois at Urbana Champaign, IL
[2] NEC Laboratories America, Cupertino, CA
{jyang29, huang}@ifp.illinois.edu, kyu@sv.nec-labs.com

**Abstract.** Sparse coding of sensory data has recently attracted notable attention in research of learning useful features from the unlabeled data. Empirical studies show that mapping the data into a significantly higher-dimensional space with sparse coding can lead to superior classification performance. However, computationally it is challenging to learn a set of highly over-complete dictionary bases and to encode the test data with the learned bases. In this paper, we describe a mixture sparse coding model that can produce high-dimensional sparse representations very efficiently. Besides the computational advantage, the model effectively encourages data that are similar to each other to enjoy similar sparse representations. What's more, the proposed model can be regarded as an approximation to the recently proposed local coordinate coding (LCC), which states that sparse coding can approximately learn the nonlinear manifold of the sensory data in a locally linear manner. Therefore, the feature learned by the mixture sparse coding model works pretty well with linear classifiers. We apply the proposed model to PASCAL VOC 2007 and 2009 datasets for the classification task, both achieving *state-of-the-art* performances.

**Key words:** Sparse coding, highly over-complete dictionary training, mixture model, mixture sparse coding, image classification, PASCAL VOC challenge

## 1 Introduction

Sparse coding has recently attracted much attention in research of exploring the sparsity property in natural signals for various tasks. Originally applied to modeling the human vision cortex [1] [2], sparse coding approximates the input signal, $\mathbf{x} \in R^d$, in terms of a sparse linear combination of an over-complete bases or dictionary $\mathbf{B} \in R^{d \times D}$, where $d < D$. Among different ways of sparse coding, the one derived by $\ell_1$ norm minimization attracts most popularity, due to its coding efficiency with linear programming, and also its relationship to the NP-hard $\ell_0$ norm in compressive sensing [3]. The applications of sparse coding range from image restorations [4] [5], machine learning [6] [7] [8], to various computer vision tasks [9] [10] [11] [12]. Many efficient algorithms aiming to find such a sparse representation have been proposed in the past several years [13]. Several empirical algorithms are also proposed to seek dictionaries which allow sparse representations of the signals [4] [13] [14].

Many recent works have been devoted to learning discriminative features via sparse coding. Wright *et al.* [10] cast the recognition problem as one of finding a sparse representation of the test image in terms of the training set as a whole, up to some sparse error due to occlusion. The algorithm utilizes the training set as the dictionary for sparse coding, limiting its scalability in handling large training sets. Learning a compact dictionary for sparse coding is thus of much interest [6] [15], and the sparse representations of the signals are used as the features trained later with generic classifiers, e.g., SVM. These sparse coding algorithms work directly on the objects, and are thus constrained to modeling only simple signals, e.g., aligned faces and digits. For general image classification, such as object recognition and scene categorization, the above sparse coding scheme will fail, i.e., it is computationally prohibitive and conceptually unsatisfactory to represent generic images with various spatial contents as sparse representations in the above way.

For generic image understanding, hierarchical models based on sparse coding applied to local parts or descriptors of the image are explored. Ranzato *et al.* [16] proposed a neural network for learning sparse representations for local patches. Raina *et al.* [17] described an approach using sparse coding applying to image patches for constructing image features. Both showed that sparse coding can capture higher-level features compared to the raw patches. Kavukcuoglu *et al.* [18] presented an architecture and a sparse coding algorithm that can efficiently learn locally-invariant feature descriptors. The descriptors learned by this sparse coding algorithm performs on a par with the carefully engineered SIFT descriptors as shown in their experiments. Inspired by the *Bag-of-Features* model and the *spatial pyramid matching* kernel [19] in image categorization, Yang *et al.* [11] proposed the ScSPM method where sparse coding is applied to local SIFT descriptors densely extracted from the image, and a spatial pyramid max pooling over the sparse codes is used to obtain the final image representation. As shown by Yu *et al.* [7], sparse coding is approximately a locally linear model, and thus the ScSPM method can achieve promising performance on various classification tasks with linear SVM. This architecture is further extended in [12], where the dictionary for sparse coding is trained with back-propagation to minimize the classification error.

The hierarchical model based on sparse coding in [11] [12] achieves very promising results on several benchmarks. Empirical studies show that using larger dictionary for sparse coding to map the data into higher dimensional space will generate superior classification performance. However, the computation of both training and testing for sparse coding can be prohibitively heavy if the dictionary is highly over-complete. Although nonlinear regressor can be applied for fast inference [18], the dictionary training is still computationally challenging. Motivated by the work in [7] that sparse coding should be local with respect to the dictionary, we propose an efficient sparse coding scheme with

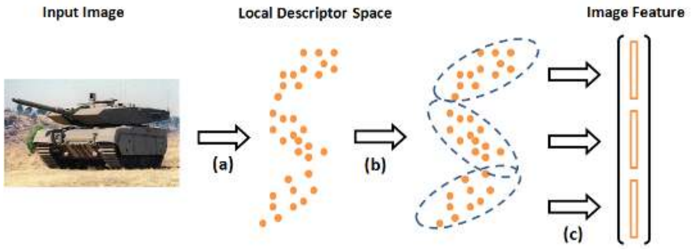


**Fig. 1.** A simplified schematic illustration of the image encoding process using the mixture sparse coding scheme. (a) local descriptor extraction; (b) mixture modeling in the descriptor space; (c) sparse coding and feature pooling. Within each mixture, a small dictionary for sparse coding can be applied, thus speeding up the coding process.

highly over-complete dictionaries using a mixture model. The model is derived via a variational approach, and the coding speed can be improved approximately at the rate of the mixture number. Fig. 1 illustrates the simplified version of the image encoding process. The mixture modeling allows a much smaller dictionary for describing each mixture well, and thus the sparse coding computation can be effectively boosted.

The reminder of this paper is organized as follows: Section 2 talks about two closely related works and the motivations; Section 3 presents the proposed model and a practical algorithm for learning the model parameters; in Section 4, classification results on PASCAL VOC 2007 and 2009 datasets are reported and compared with the existing systems; and finally Section 5 concludes our paper with discussions and future work.

## 2 Related Works and Motivations

### 2.1 Sparse Coding for Image Classification

We review the ScSPM system for image classification using sparse coding proposed in [11]. Given a large collection of local descriptors randomly extracted from training images $X = [x_1, x_2, ..., x_N]$, where $x_i \in R^{d\times 1}$ is the $i^th$ local descriptor in column manner and $N$ is the total number of local descriptors selected, the ScSPM approach first concerns learning an over-complete dictionary $B \in R^{d\times D}$ by

$$\begin{aligned} &\min_{B,\{\alpha_i\}_i^N} \sum_i^N \|\boldsymbol{x}_i - B\alpha_i\|_2^2 + \lambda\|\alpha_i\|_{\ell_1} \\ &s.t. \quad \|B(m)\|_2^2 \leq 1, m = 1, 2, ..., D, \end{aligned} \tag{1}$$

where $\ell_1$-norm is used for enforcing sparsity, $\lambda$ is to balance the representation fidelity and sparsity of the solution, and $B(m)$ is the $m^{th}$ column of $B$. Denote $A = [\alpha_1, \alpha_2, ..., \alpha_N]$, Eq. 1 is optimized by alternating between $B$ and $A$.

Fixing $B$, $A$ is found by linear programming; and fixing $A$, optimizing $B$ is a quadratically constrained quadratic programming.

Given a set of local descriptors extracted from an image or a sub-region of the image $S = [x_1, x_2, ..., x_s]$, we define the *set-level* feature over this collection of local descriptors in two steps:

1. *Sparse coding.* Convert each local descriptor into a sparse code with respect to the trained dictionary $B$:

$$\hat{A}_s = \min_A \|S - BA\|_2^2 + \lambda \|A\|_{\ell_1}, \tag{2}$$

2. *Max pooling.* The set-level feature is extracted by pooling the maximum absolute value of each row of $\hat{A}_s$:

$$\beta_s = \max(|\hat{A}_s|). \tag{3}$$

   Note that $\hat{A}_s$ contains the sparse codes as columns. Max pooling extracts the highest response in the collection of descriptors with respect to each dictionary atom, yielding a representation robust to translations within the image or its sub-regions.

To incorporate the spatial information of the local descriptors, spatial pyramid is employed to divide the image into different spatial sub-regions over different spatial scales [19]. Within each spatial sub-region, we collect its set of local descriptors and extract the corresponding *set-level* feature. The final *image-level* feature is constructed by concatenating all these *set-level* features.

### 2.2 Local Coordinate Coding

Yu *et al.* [7] proposed a local coordinate coding (LCC) method for nonlinear manifold learning in high dimensional space. LCC concerns learning a nonlinear function $f(\boldsymbol{x})$ on a high dimensional sparse $\boldsymbol{x} \in R^d$. The idea is to approximate the nonlinear function by locally linear subspaces, to avoid the "curse of dimensionality". One main result of LCC is that the nonlinear function $f(\boldsymbol{x})$ can be learned in a locally linear fashion as stated in the following lemma:

**Lemma 1 (Linearization).** *Let $B \in R^{d \times D}$ be the set of anchor points on the manifold in $R^d$. Let $f$ be an $(a, b, p)$-Lipschitz smooth function. We have for all $\boldsymbol{x} \in R^d$:*

$$\left| f(\boldsymbol{x}) - \sum_{m=1}^{D} \alpha(m) f(B(m)) \right| \leq a\|\boldsymbol{x} - \gamma(\boldsymbol{x})\|^2 + b \sum_{m=1}^{D} |\alpha(m)| \|B(m) - \gamma(\boldsymbol{x})\|^{1+p}$$

where $B(m)$ is the $m^{th}$ anchor points in $B$, $\gamma(\boldsymbol{x}) = \sum_{m=1}^{D} \alpha(m) B(m)$ is the approximation of $\boldsymbol{x}$, and we assume $a, b \geq 0$ and $p \in (0, 1]$. Note that on the left hand side, a nonlinear function $f(\boldsymbol{x})$ is approximated by a linear function

$\sum_{m=1}^{D} \alpha(m) f(B(m))$ with respect to the coding $\alpha$, where $\{f(B(m))\}_{m=1}^{D}$ is the set of function values on the anchor points. The quality of this approximation is bounded by the right hand side, which has two terms: the first term $\|\boldsymbol{x} - \gamma(\boldsymbol{x})\|$ means $\boldsymbol{x}$ should be close to its physical approximation $\gamma(\boldsymbol{x})$, and the second term means that the coding should be local. Minimizing the right hand side will ensure good approximation for the nonlinear function. Note that this minimization differs from the standard sparse coding in the regularization term, where a weighted $\ell_1$ norm is employed to encourage localized coding. Nevertheless, as shown by the experiments in [7], in the high dimensional space with unit feature normalization, empirically the standard sparse coding well approximates the local coordinate coding for classification purposes.

### 2.3 Motivation

It should be easy to see that the ScSPM approach [11] works as an approximation to the LCC in modeling the manifold of the local descriptor space. If linear SVM is used, the nonlinear function values $\{f(B(m))\}_{m=1}^{D}$ are simply determined by the weights of the classifier. The final classification score is thus an aggregation of these function values. The ScSPM model shows promising classification results on generic images with linear classifiers. Nevertheless, there are two limitations with the ScSPM framework:

1. Standard sparse coding does not include locality constraints explicitly, and thus may be inaccurate in modeling the manifold, especially when the dictionary is not big enough;
2. The computation of sparse coding increases to be unaffordable when a large dictionary is necessary to fit the nonlinear manifold well.

To make a concrete argument, we show the ScSPM computation time for encoding one image as well as the performance (in Average Precision) for dictionaries of different sizes on PASCAL VOC 2007 dataset [20], where 30,000 local descriptors are extracted from each image. As shown, the performance keeps growing as the dictionary size increases, as well as the computation time, which increases approximately linearly. In our experiment, training dictionaries beyond size 8192 is almost infeasible. The local coordinate coding (LCC) work suggests that the sparse coding should be local and the bases far away from the current encoding point can be discarded. This motivates our local sparse coding scheme induced by a Mixture Model, where local sparse coding within each mixture can be very fast (Refer to Fig. 1). For comparison, using 1024 mixtures with dictionary size 256 for each mixture, the effective dictionary size is $1024 \times 256 = 262,144$, and our proposed approach can process one image (with 30,000 local descriptors) in about one minute.

## 3 Sparse Coding using a Mixture Model

The proposed approach partitions the descriptor space via a mixture model, where within each mixture a small over-complete dictionary is used to fit the

**Table 1.** The relationships between the dictionary size and the computation time as well as the performance on PASCAL VOC 2007 validation dataset. The computation time reported is an approximate time needed for encoding one image.

| Dictionary Size | 512 | 2048 | 8192 | 32,768 |
|---|---|---|---|---|
| Computation Time | 1.5 mins | 3.5 mins | 14 mins | N/A |
| Performance | 45.3% | 50.2% | 53.2% | N/A |

local sub-manifold. An variational EM approach is applied to learn the model parameters. Because of the descriptor space partition and dictionary sharing within each mixture, we can ensure that the sparse coding is local and similar descriptors have similar sparse codes. The image feature is finally constructed by pooling the sparse codes within each mixture.

### 3.1 The Model

We describe the image local descriptor space using a $K$-mixture model, where the local distribution of each mixture is further governed by an over-complete dictionary. Let $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^N$ be the $N$ independent and identically distributed observation points, and $\mathbf{z} = \{z_n\}_{n=1}^N$ be the corresponding $N$ hidden variables, where $z_n \in \{1, 2, ..., K\}$ is a random variable indicating the mixture assignments. Denote the mixture model parameters as $\Theta = \{\mathbf{B}, \mathbf{w}\}$, where $\mathbf{B} = \{B_k\}_{k=1}^K$ is the set of over-complete dictionaries, where $B_k \in R^{d \times D}$, and $\mathbf{w} = \{w_k\}_{k=1}^K$ is the set of prior weights for the mixtures. We desire to learn the model by maximizing the likelihood

$$P(\boldsymbol{X}|\Theta) = \prod_{n=1}^N P(\boldsymbol{x}_n|\Theta) = \prod_{n=1}^N \sum_{z_n=1}^K w_{z_n} p(x_n|B_{z_n}) \tag{4}$$

where we let

$$p(\boldsymbol{x}_n|B_{z_n}) = \int p(\boldsymbol{x}_n|B_{z_n}, \alpha_n^{z_n}) p(\alpha_n^{z_n}|\sigma) d\alpha_n \tag{5}$$

be the marginal distribution of a latent-variable model with a Laplacian prior $p(\alpha_n^{z_n}|\sigma)$ on the latent variable $\alpha_n^{z_n}$, and $p(\boldsymbol{x}_n|B_{z_n}, \alpha_n^{z_n})$ is modeled as a zero-mean isotropic Gaussian distribution regarding the representation error $\boldsymbol{x}_n - B_{z_n}\alpha_n^{z_n}$.

Learning the above model requires to compute the posterior $P(\mathbf{z}|\boldsymbol{X}, \Theta)$. However, under this model, this distribution is infeasible compute in a close form. Note that approximation can be used for the marginal distribution $p(\boldsymbol{x}_n|B_{z_n})$ (introduced later in Eq. 9) in order to compute the posterior. This requires evaluating the mode of the posterior distribution of the latent variable for each data point, which, however, is computationally too slow. We thus develop a fast variational approach, where the posterior $p(\mathbf{z}_n|\boldsymbol{x}_n, \Theta)$ is approximated by

$$q(\mathbf{z}_n = k|\boldsymbol{x}_i, \Lambda) = \frac{\boldsymbol{x}_n^T A_k \boldsymbol{x}_n + b_k^T \boldsymbol{x}_n + c_k}{\sum_{k'} \boldsymbol{x}_n^T A_{k'} \boldsymbol{x}_n + b_{k'}^T \boldsymbol{x}_n + c_{k'}} \tag{6}$$

where $\mathit{\Lambda} = \{(A_k, b_k, c_k)\}$, $A_k$ is a positive definite matrix, $b_k$ is a vector, and $c_k$ is a scalar. For computational convenience, we assume $A_k$ to be diagonal. $\mathit{\Lambda}$ is a set of free parameters, determining the mixture partition in the descriptor space. Then the learning problem can be formulated as

$$\min_{\mathit{\Theta},\mathit{\Lambda}} \sum_{n=1}^{N} \sum_{z_n=1}^{K} \left[-q(z_n|\boldsymbol{x}_n, \mathit{\Lambda}) \log p(\boldsymbol{x}_n, z_n|\mathit{\Theta}) + q(z_n|\boldsymbol{x}_n, \mathit{\Lambda}) \log q(z_n|\boldsymbol{x}_n, \mathit{\Lambda})\right] \tag{7}$$

which minimizes an upper bound of the negative log-likelihood $-\sum_{i=1}^{N} \log p(\boldsymbol{x}_i|\mathit{\Theta})$ of the model [21].

### 3.2 Learning Algorithm

The learning problem in Eq. 7 can be cast into a standard variational EM algorithm, where we optimize $\mathit{\Lambda}$ to push down the upper bound to approximate the negative log-likelihood at E-step, and then update $\mathit{\Theta}$ in the M-step to maximize the approximated likelihood. Let the first term in the object be formulated into

$$\begin{aligned} &\sum_{n=1}^{N} \sum_{z_n=1}^{K} g(z_n|\boldsymbol{x}_n, \mathit{\Lambda}) \log p(\boldsymbol{x}_n, z_n|\mathit{\Theta}) \\ =&\sum_{n=1}^{N} \sum_{z_n=1}^{K} g(z_n|\boldsymbol{x}_n, \mathit{\Lambda}) \log p(\boldsymbol{x}_n|B_{z_n}) + \sum_{n=1}^{N} \sum_{z_n=1}^{K} g(z_n|\boldsymbol{x}_n, \mathit{\Lambda}) \log w_{z_n} \end{aligned} \tag{8}$$

Note that the marginal distribution $p(\boldsymbol{x}_n|B_{z_n})$ is difficult to evaluate due to the integration. We then simplify it by using the mode of the posterior distribution of $\alpha_n$:

$$\begin{aligned} -\log p(\boldsymbol{x}_n|B_{z_n}) &\approx \min_{\alpha_n^{z_n}} \{-\log p(\boldsymbol{x}_n|B_{z_n}, \alpha_n^{z_n}) - \log p(\alpha_n^{z_n}|\sigma)\} \\ &= \min_{\alpha_n^{z_n}} \|\boldsymbol{x}_n - B_{z_n}\alpha_n^{z_n}\|_2^2 + \lambda\|\alpha_n^{z_n}\|_1 \end{aligned} \tag{9}$$

which turns the integration into a standard sparse coding (or LASSO) problem. We then have the following updates rules for learning the model

1. Optimize $\mathit{\Lambda}$

$$\min_{\mathit{\Lambda}} \sum_{n=1}^{N} \sum_{z_n=1}^{K} \{q(z_n|\boldsymbol{x}_n, \mathit{\Lambda}) \left[-\log p(\boldsymbol{x}_n|B_{z_n}) - \log w_{z_n} + \log q(z_n|\boldsymbol{x}_n, \mathit{\Lambda})\right]\} \tag{10}$$

2. Optimize $\mathbf{B}$

$$\min_{\mathbf{B}} -\sum_{n=1}^{N} \sum_{z_n=1}^{K} q(z_n|\boldsymbol{x}_n, \mathit{\Lambda}) \log p(\boldsymbol{x}_i|B_{z_n}) \tag{11}$$

where each column of the dictionaries $\{B_k\}_{k=1}^{K}$ is constrained to be of unit $\ell_2$ norm. The optimization is again a quadratically constrained quadratic programming problem, similar to the procedure of updating $B$ in Eq. 1.

3. Optimize $\mathbf{w}$

$$\begin{aligned} &\min_{\mathbf{w}} - \sum_{n=1}^{N} \sum_{z_n=1}^{K} q(z_n|\boldsymbol{x}_n, \Lambda) \log w_{z_n} \\ &s.t. \quad \sum_{z_n=1}^{K} w_{z_n} = 1 \end{aligned} \tag{12}$$

which always leads to $w_{z_n} = \frac{1}{N}\sum_{n=1}^{N} q(z_n|\boldsymbol{x}_n, \Lambda)$ using the Lagrange multiplier.

By alternatively optimizing over $\Lambda$, $\mathbf{B}$ and $\mathbf{w}$, we are guaranteed to find a local minimum for the problem of Eq. 7. Note that $\mathbf{B} = [B_1, B_2, ..., B_K] \in R^{d\times KD}$ is the effective highly over-complete dictionary ($KD \gg d$) to learn for sparse coding. The above mixture sparse coding model leverages the learning complexity by training $B_k$ ($k = 1, 2, ..., K$) separately and independently in Step 2 given the posteriors from Step 1. On the other hand, since we specify all the mixture dictionaries $B_k$ to be of the same size, their fitting abilities for each data mixture will affect the mixture model parameters in Step 1, and thus the mixture weights in Step 3. Therefore, the above training procedure will efficiently learn the highly over-complete dictionary $\mathbf{B}$, while ensuring that the mixture dictionaries can fit each data mixture equally well [3].

### 3.3 Practical Implementation

The above iterative optimization procedures can be very fast with proper initialization for $\Lambda$, $\mathbf{B}$, and $\mathbf{w}$. We propose to initialize the model parameters by the following:

1. Initialize $\Lambda$ and $\mathbf{w}$: fit the data $\boldsymbol{X}$ into a Gaussian Mixture Model (GMM) with $K$ mixtures. The covariance matrix of each mixture is constrained to be diagonal for computational convenience.

$$p(\boldsymbol{X}|\mathbf{v}, \mathbf{\Sigma}, \mathbf{w}) = \prod_{n=1}^{N} \sum_{k=1}^{K} v_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k). \tag{13}$$

The above Gaussian Mixture Model can be trained with standard EM algorithm. Initialize $A_k$, $b_k$, $c_k$ and $w_k$ with $\Sigma_k^{-1}$, $-2\Sigma_k^{-1}\mu_k$, $\mu_k^T \Sigma_k^{-1}\mu_k$ and $v_k$ respectively.

2. Initialize $\mathbf{B}$: Sample the data $\boldsymbol{X}$ into $K$ clusters $\{\boldsymbol{X}_k\}_{k=1}^K$, according to the posteriors of the data points calculated from the above GMM. Train the corresponding over-complete dictionaries $\{B_k^0\}_{k=1}^K$ for those clusters using the procedure discussed for Eq. 1. Initialize $\mathbf{B}$ with this trained set of dictionaries.

[3] In [22], a Gaussian mixture model is proposed for image classification. Instead of using Gaussian to model each mixture, we use sparse coding, which can capture the local nonlinearity.

### 3.4 Image Encoding

The proposed model can be regarded as a good approximation to the LCC theory [7]: i) the mixture clustering ensures the locality of the sparse coding; ii) and the highly over-complete dictionary provides sufficient anchor points for well approximation of the nonlinear manifold. Similar to case in Sec. 2.1, suppose we have a set of local descriptors $S = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_S]$ extracted from an image or its sub-region, the *set-level* feature is defined on the latent variables (sparse codes) $\{\alpha_n^{z_n}\}$. Specifically, the local descriptors are first assigned to multiple mixtures according to the posteriors, and then the sparse codes are extracted with the corresponding dictionaries. We pool these sparse codes using a weighted average within each mixture and stack them into a super-vector:

$$f_s = [\sqrt{w_1}\mu_1^\alpha; \sqrt{w_2}\mu_2^\alpha; ...; \sqrt{w_K}\mu_K^\alpha] \tag{14}$$

where

$$\mu_k^\alpha = \frac{\sum_{n=1}^N q(z_n = k|\boldsymbol{x}_n, \Lambda)\alpha_n^{z_n}}{\sum_{n=1}^N q(z_n = k|\boldsymbol{x}_n, \Lambda)} \tag{15}$$

is the weighted average of the sparse codes with their posteriors for the $k^{th}$ mixture. The super-vector feature representation Eq. 14 has several characteristics that are not immediately obvious:

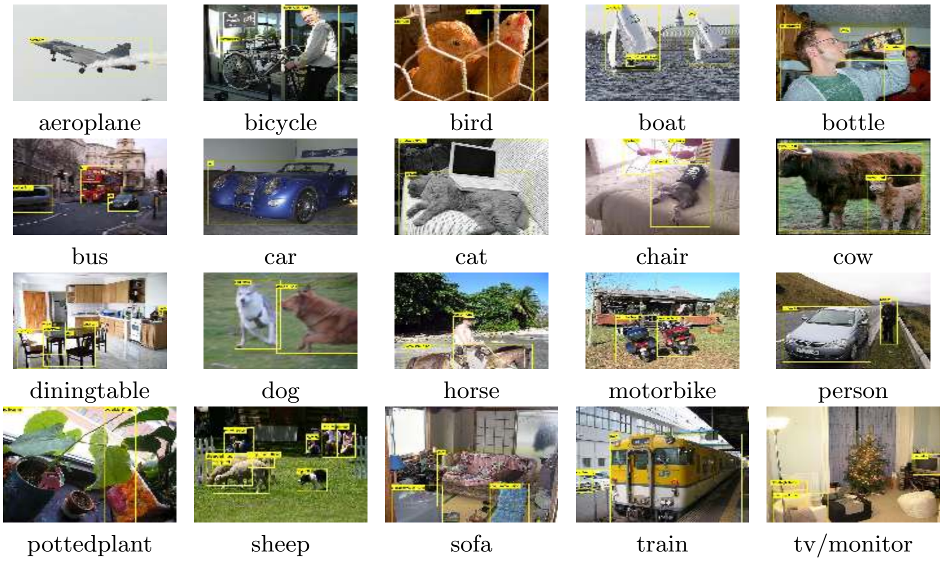
- The feature constructed in Eq. 14 is based on the locally linear model assumption, and thus is well fitted to linear kernels.
- The square root operator on each weight $w_k$ corresponds to the linearity of the feature.
- In practice, the posteriors $\{p(z_n = k|\boldsymbol{x}_n, \Lambda)\}_{k=1}^K$ are very sparse, i.e., each data point will be assigned to only one or two mixtures. Therefore, Eq. 15 is very fast to evaluate.
- The effective dictionary size of the sparse coding is $K \times D$. However, in our mixture sparse coding model, the nonlinear coding only involves dictionaries of size $D$, improving the computation approximately by $K$ times (typically we choose $K \geq 1024$).

Again, to incorporate the spatial information, we make use of the philosophy of spatial pyramid [19] to divide the image into multiple sub-regions over different configurations. The final image feature is then built by concatenating all the super-vectors extracted from these spatial sub-regions.

## 4 Experimental Validation

### 4.1 PASCAL datasets

We evaluate the proposed model on the PASCAL Visual Object Classes Challenge (VOC) datasets. The goal of this challenge is to recognize objects from a number of visual object classes in realistic scenes (i.e., not pre-segmented objects). It is fundamentally a supervised learning problem in that a training set of labeled images is provided. Totally there are twenty object classes collected:



aeroplane bicycle bird boat bottle

bus car cat chair cow

diningtable dog horse motorbike person

pottedplant sheep sofa train tv/monitor

**Fig. 2.** Example images from Pascal VOC 2007 dataset.

- **Person**: person
- **Animal**: bird, cat, cow, dog, horse, and sheep
- **Vehicle**: aeroplane, bicycle, boat, bus, car, motorbike, and train
- **Indoor**: bottle, chair, dining table, potted plant, sofa, and tv/monitor

Two main competitions for the PASCAL VOC challenge are organized:

- **Classification**: for each of the twenty classes, predicting presence/absence of an example of that class in the test image.
- **Detection**: predicting the bounding box and label of each object from the twenty target classes in the test image.

In this paper, we apply our model for the classification task to both PASCAL VOC Challenge 2007 and 2009 datasets.

The PASCAL VOC 2007 dataset [20] consists of 9,963 images, and PASCAL VOC 2009 [23] collects even more, 14,743 images in total. Both datasets are split into 50% for training/validation and 50% for testing. The distributions of images and objects by class are approximately equal across the training/validation and test sets. These images range between indoor and outdoor scenes, close-ups and landscapes, and strange viewpoints. These datasets are extremely challenging because all the images are daily photos obtained from Flickr where the size, viewing angle, illumination, etc appearances of the objects and their poses vary significantly, with frequent occlusions. Fig. 4.1 shows some example images for the twenty classes from PASCAL VOC 2007 dataset.

The classification performance is evaluated using the Average Precision (AP) measure, the standard metric used by PASCAL challenge, which computes the area under the Precision/Recall curve. The higher the score, the better the performance.

### 4.2 Implementation Details

**Local descriptor.** In our experiments, we only use single descriptor type HoG as the local descriptors, due to its computational advantage over SIFT via integral image. These descriptors are extracted from a regular grid with step size 4 pixels on the image plane. At each location, three scales of patches are used for calculating the HoG descriptor: 16 × 16, 24 × 24 and 32 × 32. As a result, approximately 30,000 local descriptors are extracted from each image. We then reduce the descriptor dimension from 128 to 80 with PCA.

**Mixture modeling.** For the VOC 07 dataset, $K = 1024$ mixtures are used and the size of the dictionary $D$ for each mixture is fixed to be 256. Therefore, the effective dictionary size is $1024 \times 256 = 262144$. Recall from Tab. 1 that working directly on a dictionary of this size is impossible. Using our mixture model, we only need to perform sparse coding on dictionaries of size 256, with little extra efforts of computing the posteriors for each descriptor, leveraging the computation time for encoding one image below a minute. For the VOC 09 dataset, we increase the mixture number to 2048. $K$ and $D$ are chosen empirically, balancing the performance and computational complexity.

**Spatial pyramid structure.** Spatial pyramid is employed to encode the spatial information of the local descriptors. As suggested by the winner system of VOC 2007 [24], we use the spatial pyramid structure shown in Fig. 4.2 for both datasets. Totally 8 spatial blocks are defined, and we extract a super-vector by Eq. 14 from each spatial block and concatenate them with equal weights.

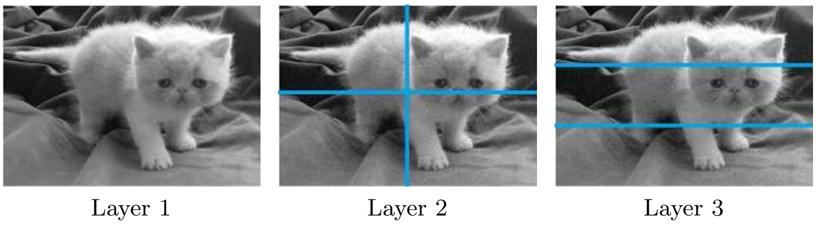


**Fig. 3.** Spatial pyramid structure used in both PASCAL VOC 2007 and 2009 datasets.

**Feature normalization.** Since our feature is based on the linear model assumption, we use Linear Discriminant Analysis (LDA) to sphere the features, and then linear SVM or Nearest Centroid is applied for classification. In practice, we always observe some improvements from this normalization step.

### 4.3 Classification Results

We present the classification results on the two datasets in this section. The precisions for each object class and the Average Precision (AP) are given by comprehensive comparisons.

**PASCAL VOC 2007 dataset.** For VOC 2007 dataset, the results we have are obtained by training on the training set and testing on the validation set. We report our results in Tab. 2, where the results of the winner system of VOC 2007 [24] and a recent proposed algorithm LLC [25] on validation set are also provided as reference. As the detailed results for Winner’07 and LLC are not available, we only cite their APs. Note that the Winner’07 system uses multiple descriptors beside dense SIFT, and the multiple kernel weights are also optimized for best performance. The LLC algorithm, similar to our system, only employs single kernel based on single descriptor. In both cases, our algorithm outperforms Winner’07 and LLC by a significant margin of about 5% in terms of AP.

**Table 2.** Image classification results on PASCAL VOC 2007 validation dataset.

| Obj. Class | aero | bicyc | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv | AP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Winner’07 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | 54.2 |
| LLC [25] | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | 55.1 |
| Ours | 78.5 | 61.6 | 53.0 | 69.8 | 31.69 | 62.2 | 81.0 | 60.5 | 55.9 | 41.8 | 59.3 | 50.3 | 75.4 | 72.9 | 82.1 | 26.1 | 36.1 | 55.7 | 81.6 | 56.3 | 59.6 |

**PASCAL VOC 2009 dataset.** Tab. 3 shows our results and comparisons with the top systems in VOC 2009. In this table, we compare with Winner’09 system (from NEC-UIUC team), and two honorable mention systems UVAS (from University of Amsterdam and University of Surrey) and CVC (from Computer Vision Centre Barcelona ). The Winner’09 system obtains its results by combining the detection scores from object detector. The UVAS system employs multiple kernel learning over multiple descriptors. The CVC system not only makes use of the detection results, but also unites multiple descriptors. Yet, our algorithm performs close to the Winner’09 system, and improves by a notable margin over the honorable mention systems.

## 5 Conclusion and Future Work

This paper presents an efficient sparse coding algorithm with a mixture model, which can work with much larger dictionaries that often offer superior classification performances. The mixture model softly partitions the descriptor space

**Table 3.** Image classification results on PASCAL VOC 2009 dataset. Our results are obtained based on single local descriptor without combining detection results.

| Obj. Class | aero | bicyc | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv | AP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Winner'09 | 88.0 | 68.6 | 67.9 | 72.9 | 44.2 | 79.5 | 72.5 | 70.8 | 59.5 | 53.6 | 57.5 | 59.0 | 72.6 | 72.3 | 85.3 | 36.6 | 56.9 | 57.9 | 85.9 | 68.0 | 66.5 |
| UVAS | 84.7 | 63.9 | 66.1 | 67.3 | 37.9 | 74.1 | 63.2 | 64.0 | 57.1 | 46.2 | 54.7 | 53.5 | 68.1 | 70.6 | 85.2 | 38.5 | 47.2 | 49.3 | 83.2 | 68.1 | 62.1 |
| CVC | 83.3 | 57.4 | 67.2 | 68.8 | 39.9 | 55.6 | 66.9 | 63.7 | 50.8 | 34.9 | 47.2 | 47.3 | 67.7 | 66.8 | 88.8 | 40.2 | 46.6 | 49.4 | 79.4 | 71.5 | 59.7 |
| Ours | 87.7 | 67.8 | 68.1 | 71.1 | 39.1 | 78.5 | 70.6 | 70.7 | 57.4 | 51.7 | 53.3 | 59.2 | 71.6 | 70.6 | 84.0 | 30.9 | 51.7 | 55.9 | 85.9 | 66.7 | 64.6 |

into local sub-manifolds, where sparse coding with a much smaller dictionary can fast fit the data. Using 2048 mixtures, each with a dictionary of size 256, i.e, effective dictionary size is $2048 \times 256 = 524,288$, our model can process one image containing 30,000 descriptor in about 1 minutes, which is completely impossible for traditional sparse coding. Experiments on PASCAL VOC datasets demonstrate the effectiveness of the proposed approach. One interesting finding we have is that although our method maps each image into an exceptionally high dimension space, e.g., the image from VOC 2009 dataset is mapped to a $2048 \times 256 \times 8 = 4,194,304$ dimensional space (spatial pyramid considered), we haven't observe any evidence of overfitting. This is possibly owing to the locally linear model assumption from LCC. Tighter connections with LCC will be investigated in the future, regarding the descriptor mixture modeling and the sparse codes pooling.

**Acknowledgments.** The main part of this work was done when the first author was a summer intern at NEC Laboratories America, Cupertino, CA. The work is also supported in part by the U.S. Army Research Laboratory and U.S. Army Research Office under grand number W911NF-09-1-0383.

## References

1. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature **381** (1996)
2. Olshausen, B., Field, D.: Sparse coding with an overcomplete basis set: A strategy emplyed by v1? Vision Research (1997)
3. Donoho, D.L.: For most large underdetermined systems of linear equations, the minimal $\ell^1$-nomr solution is also the sparest solution. Comm. on Pure and Applied Math (2006)
4. Aharon, M., Elad, M., Bruckstein, A.: K-svd: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing (2006)

5. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution as sparse representation of raw image patches. In: CVPR. (2008)
6. Bradley, D.M., Bagnell, J.A.: Differential sparse coding. In: NIPS. (2008)
7. Yu, K., Zhang, T., Gong, Y.: Nonlinear learning using local coordinate coding. In: Advances in Neural Information Processing Systems 22. (2009)
8. Cheng, B., Yang, J., Yan, S., Huang, T.: Learning with $\ell_1$ graph for image analysis. IEEE Transactions on Image Processing (2010)
9. Cevher, V., SanKaranarayanan, A., Duarte, M.F., Reddy, D., Baraniuk, R.G., Chellappa, R.: Compressive sensing for background subtraction. In: Europen Conference on Computer Vision. (2008)
10. Wright, J., Yang, A., Ganesh, A., Satry, S., Ma, Y.: Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) (Feburay 2009)
11. 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 (CVPR). (2009)
12. Yang, J., Yu, K., Huang, T.: Supervised translation-invariant sparse coding. In: IEEE Conference on Computer Vision and Pattern Recognition. (2010)
13. Lee, H., Battle, A., Raina, R., Ng, A.Y.: Efficient sparse coding algorithms. In: NIPS. (2006)
14. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. Journal of Machine Learning Research **11** (2010) 19–60
15. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Supervised dictionary learning. In: NIPS. (2008)
16. Ranzato, M.A., Boureau, Y.L., LeCun, Y. In: NIPS. (2007)
17. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.Y.: (Self-taught learning: Transfer learning from unlabeled data.)
18. Kavukcuoglu, K., Ranzato, M., Fergus, R., LeCun, Y.: Learing invariant features through topographic filter maps. In: IEEE Conference on Computer Vison and Pattern Recognition. (2009)
19. Lazebnik, S., Schmid, C., Ponce, J.: Beyonad bags of features: spatial pyramid matching for recognizing natural scene categories. In: IEEE Conference on Computer Vision and Pattern Recognition. (2006)
20. Everingham, M., Gool, L.V., Williams, C., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. (http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html)
21. Bishop, C.M.: Pattern recognition and machine learning. Springer (2006)
22. Zhou, X., Cui, N., Li, Z., Liang, F., Huang, T.: Hierarchical gaussianization for image classification. In: IEEE International Conference on Computer Vision (ICCV). (2009)
23. Everingham, M., Gool, L.V., Williams, C., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. (http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html)
24. Marszalek, M., Schmid, C., Harzallah, H., van de Weijer, J.: Learning representations for visual object class recognition. In: PASCAL Visual Object Class Challenge (VOC) workshop. (2007)
25. Jinjun Wang, Jianchao Yang, K.Y., Lv, F.: Locality-constrained linear coding for image classification. In: IEEE Conference on Computer Vision and Pattern Classificatoin. (2010)