Supervised Translation-Invariant Sparse Coding

Jianchao Yang[†], Kai Yu[‡], Thomas Huang[†]
[†]Beckman Institute, University of Illinois at Urbana-Champaign
[‡]NEC Laboratories America, Inc., Cupertino, California
[†]{jyang29, huang}@ifp.uiuc.edu, [‡]kyu@sv.nec-labs.com

Abstract

In this paper, we propose a novel supervised hierarchical sparse coding model based on local image descriptors for classification tasks. The supervised dictionary training is performed via back-projection, by minimizing the training error of classifying the image level features, which are extracted by max pooling over the sparse codes within a spatial pyramid. Such a max pooling procedure across multiple spatial scales offer the model translation invariant properties, similar to the Convolutional Neural Network (CNN). Experiments show that our supervised dictionary improves the performance of the proposed model significantly over the unsupervised dictionary, leading to state-of-the-art *performance on diverse image databases. Further more, our supervised model targets learning linear features, implying its great potential in handling large scale datasets in real applications.*

1. Introduction

Sparse coding approximates the input signal, $\mathbf{x}$, in terms of a sparse linear combination of the given overcomplete bases or dictionary $\mathbf{B}$. Such sparse representations are usually derived by linear programming as an ℓ^1 norm minimization problem [7]. Many efficient algorithms aiming to find such a sparse representation have been proposed in the past several years [1, 13, 8]. A number of empirical algorithms are also proposed to seek dictionaries which allow sparse representations of the signals [2, 13].

The sparse representation has been successfully applied to many inverse problems, e.g., image restoration [17, 24], and also well applied to classification tasks [23, 4, 25]. Wright *et al.* [23, 22] 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 achieves impressive results on public datasets, but fails to handle practical face variations such as alignment and pose. Both [23] and [22] utilize the training set as the dictionary for sparse coding, and the sparse representation is modeled directly as the classifier. Others tried to train a compact dictionary for sparse coding [4, 27], and the sparse representations of the signals are used as image features trained latter with some classifier, e.g., SVM. These holistical sparse coding algorithms on the entire image, on one hand, hold robustness to corruptions such as noise and occlusions, as shown in [23]. On the other hand, the underlying linear subspace assumption considerably limits the applications and performances of these approaches, e.g., face expression is known to be nonlinear.

Instead of sparse coding holistically on the entire image, learning sparse representations for local descriptors has also been explored for classification purposes. Raina *et al.* [19] described an approach using sparse coding to construct high-level features, showing that sparse representations perform much better than conventional representations, e.g., raw image patches. Yang *et al.* [25] proposed a stage structure where sparse coding model is applied over the hand crafted SIFT features, followed by spatial pyramid max pooling. Applied to general image classification tasks, the proposed approach achieves *state-of-the-art* performance on several benchmarks with a simple linear classifier. However, the method is based on dictionaries trained in a reconstructive manner, which is optimal for reconstruction, but not necessarily for classification. Different network structures were also proposed for fast inference for sparse coding algorithms [20, 10, 14]. However, these models are difficult to train and the supervised training can not guarantee the sparsity of the data representation.

Recent research on dictionary learning for sparse coding has been targeted at learning discriminant sparse models [16, 4] instead of the purely reconstructive ones. Mairal *et al.* [16] generalized the reconstructive sparse dictionary learning process by optimizing the sparse reconstruction jointly with a linear prediction model. Bradley and Bagnell [4] proposed a novel differentiable sparse prior rather than the conventional ℓ_1 norm, and employed a backpropagation procedure to train the dictionary for sparse coding in order to minimize the training error. These approaches need to

explicitly associate each data sample, either an image or image patch, with a label in order for the supervised training to proceed. In [25] and [19], the local descriptors can belong to multiple classes. How to learn a discriminant dictionary both for sparse data representation and image classification based on image local descriptors is still not addressed.

This paper introduces a novel supervised hierarchical sparse coding model, based on images represented by *bag-of-features*, where a local image descriptor may belong to multiple classes. We train the dictionary for local descriptors through back-propagation, by minimizing the training error of the image level features, which are extracted by max pooling over the sparse codes within a spatial pyramid [25]. The achieved dictionary is remarkably more effective than the unsupervised one in terms of classification. And the max pooling procedure over different spatial scales equips the proposed model with local translation-invariance similar to the convolutional network [14]. Our hierarchical structure can be trained with many classifiers, but in this paper we specifically take linear SVM as our prediction model. For linear SVM, the proposed framework has a computation complexity linear to the data size in training and a constant in testing, and thus will be of particular interest in real applications.

The rest of the paper is organized as follows. Sec. 2 introduces the notations used in our presentation. Sec. 3 presents the hierarchical translation-invariant sparse coding structure. And Sec. 4 talks about the supervised dictionary training. We further sketch a justification for our sparse coding model in Sec. 5. And experiments are evaluated in Sec. 6. Finally, Sec. 7 concludes our paper.

2. Notation

Bold uppercase letters, $\mathbf{X}$, denote matrices and bold lowercase letters, $\mathbf{x}$, denote column vectors. For both matrices and vectors, bold subscripts $\mathbf{X}_m$ and $\mathbf{x}_n$ are used for indexing, while plain subscripts, such as X_{ij} and x_k, denote the element respectively. When both matrix indexing and element indexing exist, we use the superscripts as matrix indexing and lowerscripts as element indexing. For example, X_{ij}^m denotes the matrix element at the i-th row and j-th column of the m-th matrix $\mathbf{X}^m$. The same case with vectors for mixed indexing.

3. Hierarchical translation-invariant sparse coding structure

This section introduces the hierarchical model based on sparse coding on local descriptors for image classification. Given the dictionary, the image level feature extracted is of translation-invariance property, and thus is robust to image misalignment [1].

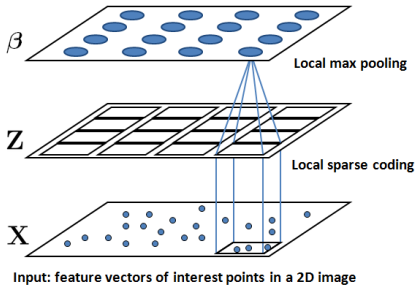

Figure 1. An example architecture of convolutional sparse coding.

3.1. Convolutional sparse coding

In our hierarchical model, an image is represented by a local descriptor set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 ..., \mathbf{x}_N] \in \mathbb{R}^{d \times N}$, where $\mathbf{x}_i$ denotes the i^{th} local descriptor of the image in column vector. Suppose we are given a dictionary $\mathbf{B} \in \mathbb{R}^{d \times K}$ that can sparsely represent these local descriptors, where K is the size of the dictionary and is typically greater than $2d$. The sparse representations of a descriptor set are computed as

$$\hat{\mathbf{Z}} = \arg\min_{\mathbf{Z}} \|\mathbf{X} - \mathbf{B}\mathbf{Z}\|_{\ell_2}^2 + \gamma \|\mathbf{Z}\|_{\ell_1}, \tag{1}$$

where $\hat{\mathbf{Z}} \in \mathbb{R}^{K \times N}$ contains the sparse representations in columns for the descriptors in X. In order for classification, where we need fixed length feature vectors, we define the image level feature over the sparse representation matrix $\hat{\mathbf{Z}}$ by *max pooling*

$$\boldsymbol{\beta} = \xi_{max}(\hat{\mathbf{Z}}), \tag{2}$$

where ξ_{max} is defined on each row of $\hat{\mathbf{Z}} \in \mathbb{R}^{K \times N}$, returning a vector $\boldsymbol{\beta} \in \mathbb{R}^K$ with the i-th element being

$$\beta_i = \max \left\{ |\hat{Z}_{i1}|, |\hat{Z}_{i2}|, ..., |\hat{Z}_{iN}| \right\}. \tag{3}$$

The feature vector $\boldsymbol{\beta}$ defined in such a way is called *global pooling feature*, because the pooling function is evaluated on the whole descriptor set, discarding all spatial information of the local descriptors. Max pooling has been widely used in neural network algorithms and is also shown to be biological plausible. As we show in Sec. 5, max pooling is critical for the success of our sparse coding model.

To consider the spatial information and also to achieve regional invariance of the local descriptors to translations, we do convolutional coding [2] by dividing the whole image into $m \times m$ non-overlapping spatial cells. The image level

[1] This is different from [11], where the sparse coding model is robust to feature misalignment, i.e., the sparse feature won't change much if the feature itself shifts a little.

[2] The convolutional coding is not done by weight sharing as in the conventional case , e.g., [14], but performed by sparse coding with the same dictionary on local descriptors shifting across the image.

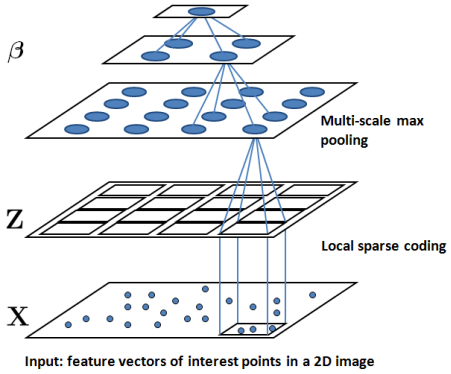

Figure 2. The architecture of the unsupervised hierarchical sparse coding on multiple spatial scales. Three scales are shown.

feature is then defined by a concatenation of the max pooling features defined on m^2 spatial cells:

$$\boldsymbol{\beta} = \bigcup_{c=1}^{m^2} \left[\xi_{max}(\hat{\mathbf{Z}}_{I_c}) \right] \tag{4}$$

where $\boldsymbol{\beta}$ is in $\mathbb{R}^{m^2 K}$, $\bigcup[\star]$ denotes the vector concatenation operator, and I_c is the index set for the local descriptors falling into the receptive field of c-th spatial cell. Fig. 1 illustrates the structure of our convolutional sparse coding scheme. In the figure, the image is divided into 4×4 spatial cells. The max pooling feature is invariant to translations of the local descriptors within each cell, while on the other hand still retains coarse spatial information as a whole image.

3.2. Multi-scale convolutional sparse coding

The max pooling feature from convolutional sparse coding achieves a trade off between translation invariance and the spatial information of the local image descriptor. In one extreme, if we use 1×1 cell, i.e., the global pooling, the max pooling feature is most invariant to local descriptor translations, but we lose the informative spatial information totally. In the other extreme, if we divide the image into so many cells that each cell contains only a single descriptor, the spatial information is totally retained, by we lose the translation invariance. Therefore, to achieve both translation invariance and spatial information of the max pooling feature, we can combine convolutional sparse coding on different scales of spatial cell structures, resulting in a hierarchical model similar to the Convolutional Neural Network (CNN) and the spatial pyramid [25]. Fig. 2 shows the multi-scale convolutional sparse coding structure we used in our paper. The higher level of the max pooling feature, the more translation invariant, while the lower level, the more spatial information retained. The final image level feature derived from such a structure is the concatenation of max pooling features from different spatial cells on different convolutional scales.

Suppose we model the image in R spatial scales. In each scale s, the image is divided into $2^{s-1} \times 2^{s-1}$ non-overlapping cells. The multi-scale convolutional max pooling feature $\boldsymbol{\beta}$ is expressed as

$$\boldsymbol{\beta} = \bigcup_{s=1}^{R} [\boldsymbol{\beta}^s] = \bigcup_{s=1}^{R} \left[\bigcup_{c=1}^{2^{s-1}} \boldsymbol{\beta}_c^s \right], \tag{5}$$

where $\boldsymbol{\beta}_c^s$ denotes the set-level max pooling feature for the c-th spatial cell on the s-th scale. Eqn. 5 is the final feature vector as the input to latter classifiers such as linear SVM.

4. Supervised dictionary learning for hierarchical sparse coding

In the previous section, the dictionary $\mathbf{B}$ for sparse coding is used as given. In practice, the dictionary $\mathbf{B}$ is usually trained over a large set of local descriptors $\mathbb{X} = \left[\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^m\right]$ [3] by an ℓ_1 minimization [13]

$$\min_{\{\mathbf{z}^i\}, \mathbf{B}} \sum_{i=1}^{m} \left\{ \|\mathbf{x}^i - \mathbf{B}\mathbf{z}^i\|_{\ell_2}^2 + \gamma \|\mathbf{z}^i\|_{\ell_1} \right\}, \tag{6}$$

which aims to learn a dictionary that can sparsely represent each local descriptor. The optimization problem in Eqn. 6 is not convex in $\mathbf{B}$ and $\{\mathbf{z}_i\}_{i=1}^m$ simultaneously, but is convex in one once the other fixed. Therefore, the optimization is done in an alternative coordinate descent fashion between $\mathbf{B}$ and $\{\mathbf{z}_i\}_{i=1}^m$, which guarantees to converge to a local minimum. Obviously, such a reconstructive learning process is not necessarily optimal for discriminant analysis we are interested in. The following discussions will introduce a discriminant dictionary training procedure based on back-propagation for our hierarchical sparse coding structure.

4.1. Back-propagation

In retrospect, the previous system in Sec. 3 defines implicit feature transformations from the descriptor-set represented image $\mathbf{X}_k$ to the hierarchical max pooling feature $\boldsymbol{\beta}_k$. The transformations are achieved by two steps.
Step 1:

$$\mathbf{Z}_k = \varphi(\mathbf{X}_k, \mathbf{B}) \tag{7}$$

denoting the transformation defined by Eqn. 1, and
Step 2:

$$\boldsymbol{\beta}_k = \xi_{max}^*(\mathbf{Z}_k), \tag{8}$$

where we use ξ_{max}^* to denote the multilevel max pooling function, to differentiate from ξ_{max}. Combining these two steps we have

$$\boldsymbol{\beta}_k = \phi(\mathbf{X}_k, \mathbf{B}). \tag{9}$$

[3] In our context, $\mathbb{X}$ contains local descriptors from many images.

For classification tasks, with a predictive model $f(\boldsymbol{\beta}_k, \mathbf{w}) \equiv f(\phi(\mathbf{X}_k, \mathbf{B}), \mathbf{w})$, a class label y_k of the image, and a classification loss $\ell(y_k, f(\phi(\mathbf{X}_k, \mathbf{B}), \mathbf{w}))$, we desire to train the whole system with respect to the predictive model parameters $\mathbf{w}$ and $\mathbf{B}$ given n training images:

$$\min_{\mathbf{w},\mathbf{B}} \left\{ \sum_{k=1}^{n} \ell\left(y_k, f\left(\phi(\mathbf{X_k}, \mathbf{B}), \mathbf{w}\right)\right) + \lambda \|\mathbf{w}\|_{\ell^2}^2 \right\}, \quad (10)$$

where λ is used to regularize the predictive model. For ease of presentation, denote

$$E\left(\mathbf{B}, \mathbf{w}, \{\mathbf{X_k}\}_{k=1}^n\right) = \sum_{k=1}^{n} \ell\left(y_k, f\left(\phi(\mathbf{X}_k, \mathbf{B}), \mathbf{w}\right)\right) + \lambda \|\mathbf{w}\|_{\ell_2}^2. \quad (11)$$

Minimizing $E\left(\mathbf{B}, \mathbf{w}, \{\mathbf{X_k}\}_{k=1}^n\right)$ over $\mathbf{B}$ and $\mathbf{w}$, the learned dictionary will be more closely tightened with the classification model, and therefore more effective for classification [4]. The problem can be approached by optimizing alternatively over $\mathbf{B}$ and $\mathbf{w}$. Given the dictionary $\mathbf{B}$ fixed, optimization over $\mathbf{w}$ is simply training the classifier with the multi-scale max pooling features. Given the classifier $\mathbf{w}$, to optimize E over $\mathbf{B}$, we have to compute the gradient of E with respect to $\mathbf{B}$ using the chain rule:

$$\begin{aligned} \frac{\partial E}{\partial \mathbf{B}} &= \sum_{k=1}^{n} \frac{\partial \ell}{\partial \mathbf{B}} = \sum_{k=1}^{n} \frac{\partial \ell}{\partial f} \frac{\partial f}{\partial \mathbf{B}} = \sum_{k=1}^{n} \frac{\partial \ell}{\partial f} \frac{\partial f}{\partial \boldsymbol{\beta}_k} \frac{\partial \boldsymbol{\beta}_k}{\partial \mathbf{B}} \\ &= \sum_{k=1}^{n} \frac{\partial \ell}{\partial f} \frac{\partial f}{\partial \boldsymbol{\beta}_k} \frac{\partial \boldsymbol{\beta}_k}{\partial \mathbf{Z}_k} \frac{\partial \mathbf{Z}_k}{\partial \mathbf{B}}. \end{aligned} \quad (12)$$

Therefore, the problem is reduced to computing the gradients of the sparse representation matrix $\mathbf{Z}_k$ with respect to the dictionary $\mathbf{B}$, which can be calculated by investigating the gradient of each individual sparse code $\mathbf{z}$ (a column of $\mathbf{Z}_k$) with respect to $\mathbf{B}$. The difficulty of computing such a gradient is that there is no analytical link between $\mathbf{z}$ and the dictionary $\mathbf{B}$, which we overcome by using implicit differentiation on the fixed point equations.

4.2. Implicit differentiation

In order to establish the relationship between a sparse code $\mathbf{z}$ and $\mathbf{B}$, we first find the fixed point equations by computing the gradient with respect to $\mathbf{z}$ on Eqn. 1 at its minimum $\hat{\mathbf{z}}$ as suggested by [4]:

$$\bigtriangledown(\|\mathbf{x} - \mathbf{B}\mathbf{z}\|_{\ell^2}^2)\big|_{\mathbf{z}=\hat{\mathbf{z}}} = -\bigtriangledown\left(\|\mathbf{z}\|_{\ell^1}\right)\big|_{\mathbf{z}=\hat{\mathbf{z}}}, \quad (13)$$

leading to

$$2(\mathbf{B}^T\mathbf{B}\mathbf{z} - \mathbf{B}^T\mathbf{x})\big|_{\mathbf{z}=\hat{\mathbf{z}}} = -\gamma \cdot \text{sign}(\mathbf{z})\big|_{\mathbf{z}=\hat{\mathbf{z}}}, \quad (14)$$

where $\text{sign}(\mathbf{z})$ is a vector functioning on each element of vector $\mathbf{z}$, and $\text{sign}(0) = 0$. Note that Eqn. 13 and 14 hold only under the condition of $\mathbf{z} = \hat{\mathbf{z}}$. In the following derivations, we will admit the condition without the symbol $|_{\mathbf{z}=\hat{\mathbf{z}}}$ unless otherwise mentioned.

In Eqn. 14, $\mathbf{z}$ is not linked with $\mathbf{B}$ explicitly. To calculate the gradient of $\mathbf{z}$ with respect to $\mathbf{B}$, we use implicit differentiation by taking derivative of $\mathbf{B}$ on both sides of Eqn. 14:

$$\frac{\partial\{2(\mathbf{B}^T\mathbf{B}\mathbf{z} - \mathbf{B}^T\mathbf{x})\}}{\partial B_{mn}} = \frac{\partial\{-\gamma \cdot \text{sign}(\mathbf{z})\}}{\partial B_{mn}}. \quad (15)$$

The "sign" function on the right side is not continuous at zero. However, since the left side of Eqn. 15 cannot be infinite, $\partial\{-\gamma \cdot \text{sign}(\mathbf{z})\}/\partial B_{mn} = 0$.[5] Since the gradient $\partial z_i / \partial B_{mn}$ is not well defined for $z_i = 0$, we set them to be zero and only care about the gradients where $z_i \neq 0$ [6]. Denote $\tilde{\mathbf{z}}$ as the nonzero coefficients of $\hat{z}$, and $\tilde{\mathbf{B}}$ being the corresponding bases (the supports selected by $\hat{z}$). From Eqn. 15, we have

$$\frac{\partial\{2(\tilde{\mathbf{B}}^T\tilde{\mathbf{B}}\tilde{\mathbf{z}} - \tilde{\mathbf{B}}^T\mathbf{x})\}}{\partial B_{mn}} = 0, \quad (16)$$

which gives

$$\frac{\partial\tilde{\mathbf{B}}^T\tilde{\mathbf{B}}}{\partial B_{mn}}\tilde{\mathbf{z}} + \tilde{\mathbf{B}}^T\tilde{\mathbf{B}}\frac{\partial\tilde{\mathbf{z}}}{\partial B_{mn}} - \frac{\partial\tilde{\mathbf{B}}^T\mathbf{x}}{\partial B_{mn}} = 0, \quad (17)$$

Therefore, the desired gradient can be solved by

$$\frac{\partial\tilde{\mathbf{z}}}{\partial B_{mn}} = \left(\tilde{\mathbf{B}}^T\tilde{\mathbf{B}}\right)^{-1}\left(\frac{\partial\tilde{\mathbf{B}}^T\mathbf{x}}{\partial B_{mn}} - \frac{\partial\tilde{\mathbf{B}}^T\tilde{\mathbf{B}}}{\partial B_{mn}}\tilde{\mathbf{z}}\right). \quad (18)$$

Since the number of non-zero coefficients is generally far smaller than the descriptor dimension d, the inverse $\left(\tilde{\mathbf{B}}^T\tilde{\mathbf{B}}\right)^{-1}$ is well-conditioned.

4.3. Multi-scale gradient

Now we are ready to compute the gradient of the multi-scale max pooling feature $\boldsymbol{\beta}$ in Eqn. 5 with respect to $\mathbf{B}$. To show the details of derivation, we first examine the simplest case where $\boldsymbol{\beta}_0 = \xi(\hat{\mathbf{Z}})$, the global pooling feature $\hat{\mathbf{Z}}$.

$$\frac{\partial\boldsymbol{\beta}_0}{\partial B_{mn}} = \text{sign}(\hat{\mathbf{z}}^{max}) \odot \frac{\partial\hat{\mathbf{z}}^{max}}{\partial B_{mn}} \quad (19)$$

where $\hat{\mathbf{z}}^{max}$ is a vector composed of the elements with the largest absolute values in each row of $\hat{\mathbf{Z}}$. Similarly, the gradient of the final multi-scale max pooling feature $\boldsymbol{\beta}$ with

[4] In theory, a reconstructive term should be added in Eq. 11 to ensure the learned $\mathbf{B}$ can represent the data well. In practice, we initialize $\mathbf{B}$ with its unsupervised version, and then perform supervised training to refine it, which runs much faster.

[5] Otherwise, if a small change of $\mathbf{B}$ causes sign change of $\mathbf{z}$, the right side of Eqn. 15 will be ∞.

[6] In practice, such a procedure works well

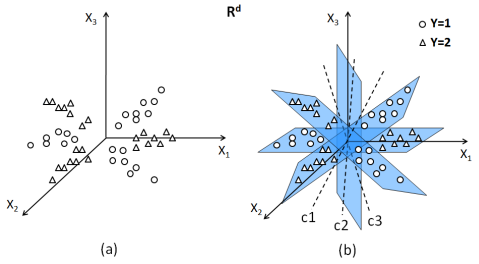

Figure 3. The rationale behind our hierarchical sparse coding model. Left: the local descriptors from two classes $Y = 1, 2$ in the original space; Right: sparse coding discovers the linear subspace structure of the local descriptors. Simple linear classifiers are sufficient to separate two classes within each linear subspace.

respect to $\mathbf{B}$ is computed as

$$\frac{\partial \boldsymbol{\beta}}{\partial B_{mn}} = \frac{\partial \bigcup_{s=1}^{R} \left[\bigcup_{c=1}^{2^{s-1}} \boldsymbol{\beta}_c^s \right]}{\partial B_{mn}} = \bigcup_{s=1}^{R} \left[\bigcup_{c=1}^{2^{s-1}} \left[\frac{\partial \boldsymbol{\beta}_c^s}{\partial B_{mn}} \right] \right] \tag{20}$$

where I_s^c again indicates the index set of the local descriptors in receptive field of c-th cell on the s-th scale. With $\partial\boldsymbol{\beta}/\partial B_{mn}$ calculated, the quantity of 12 can be evaluated through the chain rules, and the dictionary $\mathbf{B}$ can be updated iteratively for toward optimal for classification.

5. Interpretation as Sparse Subspace Modeling

Assume the local image descriptors are sparse signals with respect to a dictionary $\mathbf{B}$. To simply the analysis, suppose these local descriptors reside in a union of M linear subspaces with non-sharing bases. Sparse coding clusters each local image descriptor into a linear subspace and projects it into the corresponding sub-coordination system, in which the nonzero coefficients of its sparse codes represent its coordinates. Within each sub-coordinate system, a simple linear classifier is likely to separate the projected local descriptors from the two classes. Concatenating these linear classifiers produces a linear classifier on the sparse codes of these local descriptors. Equivalently, training a linear classifier in the sparse feature space produces linear classifiers within each sub-coordinate systems. Fig. 3 illustrates this idea in the original descriptor space, where good linear separation is possible for descriptors from two classes within each subspace discovered by sparse coding, and overall a good nonlinear separation in the descriptor space is achieved.

Since sparse coding functions as a means of efficient descriptor space partition, which is determined by the dictionary, the supervised dictionary training can be explained as seeking the descriptor space partitions where the local image descriptors from two classes are most separable given the classifier.

6. Experiment Results

We verify the performance of our proposed algorithm on several benchmark tasks including face recognition (CMU PIE [21], CMU Multi-PIE [9]), handwritten digit recognition (MNIST) [12] and gender recognition (FRGC [18]). In each task, we report the prediction accuracies for our model with unsupervised and supervised dictionaries to show the benefits from supervised training. We also compare our algorithm with *state-of-the-art* algorithms specifical for each dataset under the same experiment settings.

6.1. Implementation details

6.1.1 Parameter Settings

For all the tasks we test, the local descriptors are simply the raw image patches, densely sampled from the image on a regular grid with step size of 1 pixel. These raw image patches are pre-normalized to be unit vectors before sparse coding. Tab. 1 summaries the experiment settings for all the datasets. In our experiments, image size, patch size, the sparsity regularization γ, and dictionary size all affects the performance of our hierarchical sparse coding model. These parameters are set empirically, without searching for optimal settings. The dictionaries $\mathbf{B}$'s are initialized by the corresponding unsupervised ones for supervised training.

Table 1. Parameter settings for the datasets we use. We set these parameters empirically, without testing for optimal settings.

dataset	image size	patch size	γ	dictionary size
CMU PIE	32×32	8×8	0.1	128
Multi-PIE	30×40	8×8	0.1	128
MNIST	28×28	12×12	0.2	256
FRGC	32×32	8×8	0.1	128

6.1.2 Supervised dictionary training

The optimization usually converges within 10 iterations. The following depicts some details regarding the proposed supervised training.

1. **Predictive model**: the predictive model we use in Eqn. 10 for discriminant dictionary training is linear SVM. But we modify the traditional hinge loss function to a differentiable squared hinge loss. The model regularization parameter λ is set to be 0.1 for all datasets.

2. **Feature**: instead of the multi-scale max pooling feature, we use the *global max pooling feature* for the dictionary training process. As we will see in the later experiment results, the performance of our hierarchical model with unsupervised dictionary is already fairly

decent. Therefore, minimizing the training error with the max pooling feature is more effective than that of the multi-scale pooling feature.

3. **Optimization**: the optimization in Eqn. 10 is implemented in a stochastic way with gradient descent. The initial learning rate is set as 0.001, which decays in a manner similar to the neural network:

$$r = \frac{r_0}{\sqrt{n/N + 1}} \tag{21}$$

where r_0 is the initial learning rate, r is the current learning rate, n is the incremental count of the current example and N is the size of the dataset.

To show the effectiveness of the supervised training procedure proposed in this work, we report the validation performance on each test set for both unsupervised dictionary and supervised dictionary. Note that the feature we use here is only the global max pooling feature, to be consistent with the optimization process. For datasets CMU PIE and Multi-PIE, where we have multiple testing settings, we only report one as an example. Tab. 2 shows the performance comparisons for the two dictionaries on all the datasets. In terms of error reduction, the supervised model improves the performance significantly. Without further notification, in the latter experiments, the performances reported are all obtained with the multi-scale max pooling feature for both unsupervised and supervised dictionaries. To make notation uncluttered, we specifically use "U-SC" and "S-SC" to denote the hierarchical model using unsupervised dictionary and supervised dictionary respectively.

Table 2. Error rate for both unsupervised and supervised dictionary on different datasets. Supervised training improvements are reported in terms of error reduction. Note that the feature used is the global pooling feature.

Dataset	Unsupervised	Supervised	Improvements
CMU PIE	11.5	2.3	80.0%
Multi-PIE	32.3	21.9	32.2%
MNIST	2.8	1.1	60.7%
FRGC	11.9	6.4	46.2%

6.2. Face recognition

We first apply the proposed algorithm to the face recognition problem. We evaluate the algorithm's performance on two database: CMU PIE[21] and CMU Multi-PIE[9].

6.2.1 CMU PIE

The database consists of 41,368 images of 68 people, each person under 13 poses, 43 different illumination conditions and with 4 different expressions. We use a subset of the database same as in [6, 5] for fair comparison.The subset only contains five near frontal poses (C05, C07, C09, C27, C29) and all the images under different illuminations and expressions. Therefore, there are 170 images for each individual. A random subset of p ($p = 30, 50, 70, 90, 130$) images per person are selected as the training set and the rest of the database is considered as the testing set.

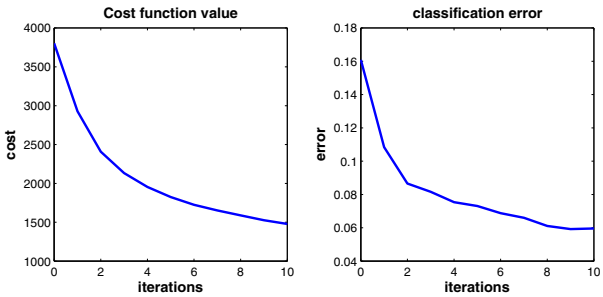

Figure 4. The supervised optimization process on CMU PIE for 10 iterations.

Fig. 4 shows the optimization process of supervised training for 10 iterations for $p = 50$ experiment setting. For each iteration, we record its cost function value, and also evaluate the performance with current learned dictionary on the test set. As expected, the classification error decreases as the cost function value decreases. Tab.3 shows the performance comparisons with the literature on this dataset. "Improvements" shows the improvement from unsupervised dictionary to supervised dictionary. As shown, both our unsupervised and supervised sparse coding algorithm significantly outperform S-LDA [5], reported as *state-of-the-art* performance algorithm on this database, reducing the error rate by more than 10 times.

Table 3. Classification error (%) on CMU PIE database for different algorithms.'Improve' shows the improvements from unsupervised sparse coding (U-SC) to supervised sparse coding (S-SC) in terms of reducing error rate.

Training	30	50	70	90	130
LDA	7.9	4.8	4.0	3.4	2.9
R-LDA [6]	5.0	3.9	3.5	3.2	3.0
S-LDA [5]	3.6	2.5	2.1	1.8	1.6
U-SC	0.81	0.26	0.22	0.11	0.037
S-SC	**0.49**	**0.15**	**0.12**	**0.037**	**0**
Improvements	39.5%	42.3%	45.5%	66.4%	100%

6.2.2 CMU Multi-PIE

The second experiment on face recognition is conducted on the large scale CMU Multi-PIE database[9]. The database contains 337 subjects across simultaneous variations in pose, expression, and illumination. In order to compare

with [22] fairly, we use the same experiment settings for face recognition. Of these 337 subjects, 249 subjects present in Session 1 are used as the training set. Session 2, 3 and 4 are used as testing. The remaining 88 subjects are considered "outliers" or invalid images in [22] for face verification, and in this work we neglect them and only care about face recognition. For the training set, [22] only included 7 frontal extreme illuminations, taken with neutral expression. We use exactly the same training set. For the test set, all 20 illuminations from Sessions 2-4 are used, which were recorded at distinct times over a period of several months. The dataset is challenging due to the large number of subjects, and due to natural variation in subject appearance over time.

Table 4. Face recognition error (%) on large-scale Multi-PIE. 'Improvements' row shows the improvements due to supervised training.

Rec. Rates	Session 2	Session 3	Session 4
LDA	50.6	55.7	52.1
NN	32.7	33.8	37.2
NS	22.4	25.7	26.6
SR	8.6	9.7	9.8
U-SC	5.4	9.0	7.5
S-SC	**4.8**	**6.6**	**4.9**
Improvements	11.1%	26.7%	34.7%

Tab. 4 shows our results compared with those reported in the [22] for Linear Discriminant Analysis (LDA)[3], Nearest Neighbor (NN), Nearest Subspace (NS)[15], and Sparse Representation (SR). LDA, NN and NS are used as the baseline algorithms in [22]. The SR algorithm, unifying face alignment and face recognition in the same framework, performs much better compared to these baseline algorithms, reporting the top classification accuracy on this dataset. To compare with the SR algorithm, we make two noteworthy comments:

1. The linear combination model of SR is known to be good at handling illuminations. The training set is chosen as to minimize its size.

2. The SR algorithm models the sparse representation as the classifier directly, which is highly nonlinear. Our model simply uses a linear SVM trained by *one-vs-all*, dividing the feature space into 249 parts.

And yet, our supervised sparse coding strategy significantly reduce the error rates of SR by 41.9%, 32.0% and 50.0% for session 2, 3 and 4 respectively.

6.3. Handwritten digit recognition

We also test our algorithm on the benchmark MNIST handwritten digits dataset [12]. The database consists of 70,000 handwritten digits, of which 60,000 digits are modeled as training and 10,000 as testing. The digits have been size-normalized and centered in a fixed-size image. The supervised training optimization process converges quickly and we stop by 5 iterations. Tab. 5 shows the performance comparisons with other methods reported on the dataset. "L1 sparse coding" and "Local coordinate coding" methods denote the holistical sparse coding scheme on the entire image with trained compact dictionaries, with the latter enforcing locality constraints. Our patch-based hierarchical model performs much better than the above holistical methods. The supervised training reduces the error of the unsupervised model by 14.3% and we achieve similar performance as CNN under the same condition, which is known as the best algorithm on the MNIST dataset.

Table 5. Classification error (%) comparison with *state-of-the-art* algorithms in the literature on MNIST.

Algorithms	Error Rate
SVM (RBF)	1.41
L1 sparse coding (linear SVM)	2.02
Local coordinate coding (linear SVM) [27]	1.90
Deep belief network	1.20
CNN [26]	0.82
U-SC (linear SVM)	0.98
S-SC (linear SVM)	**0.84**
Improvements	14.3%

6.4. Gender recognition

Our gender recognition experiment is conducted on the FRGC 2.0 dataset [18]. This dataset contains 568 individuals, totally 14714 face images under various lighting conditions and backgrounds. Besides person identities, each image is annotated with gender and ethnicity. For gender recognition, we fix 114 persons' 3014 images (randomly chosen) as the testing set, and the rest 451 individuals' 11700 images as our training images. Comparisons are performed with the *state-of-the-art* algorithms on FRGC in the same experiment setting as reported in Tab. 6. The supervised sparse coding strategy boosts the performance by 22.1% error reduction compared with the unsupervised version, outperforming the top performance algorithm CNN.

Table 6. Classification error (%) comparison with *state-of-the-art* gender recognition algorithms in the literature on FRGC.

Algorithms	Error Rate
SVM (RBF)	8.6
CNN [26]	5.9
Unsupervised Sparse Coding	6.8
Supervised Sparse Coding	**5.3**
Improvements	22.1%

7. Conclusion and Future works

This paper presents a novel supervised hierarchical sparse coding model for image classification. The hierarchical model is constructed by max pooling over the sparse codes of local image descriptors within a spatial pyramid, and thus the feature is robust to local descriptor translations. The supervised training of the dictionary is done via back-projection where implicit differentiation is used to relate the sparse codes with the dictionary analytically. Experiments on various datasets demonstrate the promising power of the supervised model. Future works along this line should address the following issues. First, current stochastic optimization based on back-propagation is slow. Faster methods should be investigated. Second, max pooling loses much information. More pooling methods coupled with the sparse coding model should be studied.

Acknowledgement

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

References

[1] Sparselab http://sparselab.stanford.edu/.

[2] M. Aharon, M. Elad, and A. Bruckstein. K-svd: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 2006.

[3] P. Belhumeur, J. Hespanda, and D. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projectoin. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1997.

[4] D. M. Bradley and J. A. Bagnell. Differential sparse coding. In *NIPS*, 2008.

[5] D. Cai, X. He, and J. Han. Semi-supervised discriminant analysis. In *IEEE International Conference on Computer Vison (ICCV), 2007*, 2008.

[6] D. Cai, X. He, Y. Hu, J. Han, and T. Huang. Learning a spatially smooth subspace for face recognition. In *CVPR*, 2007.

[7] D. L. Donoho. For most large underdetermined systems of linear equations, the minimal ℓ^1-nomr solution is also the sparest solution. *Comm. on Pure and Applied Math*, 2006.

[8] J. Friedman, T. Hastie, and R. Tibshirani. Regularized paths for generalized linear models via coordinate descent. 2008.

[9] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.

[10] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *IEEE International Conference on Computer Vision*, 2009.

[11] K. Kavukcuoglu, M. A. Ranzato, R. Fergus, and Y. LeCun. Learning invariant features through topographic filter maps. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of IEEE, vol.86, no.11, pp.2278-2324*, 1998.

[13] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *NIPS*, 2006.

[14] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *International Conference on Machine Learning*.

[15] K. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under varible lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2005.

[16] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Supervised dictionary learning. In *NIPS*, 2008.

[17] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transaction on Image Processing*, 2008.

[18] P. J. Philips, P. J. Flynn, T. Scruggs, K. W. Bower, and W. Worek. Preliminary face recognition grand challenge results. In *IEEE Conference and Automatic Face and Gesture Recognition*, 2006.

[19] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: Transfer learning from unlabeled data.

[20] M. A. Ranzato, Y.-L. Boureau, and Y. LeCun. In *NIPS*, 2007.

[21] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression (pie) database of human faces. Technical Report CMU-RI-TR-01-02, Robotics Institute, Pittsburgh, PA, January 2001.

[22] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, and Y. Ma. Towards a practical face recognition system:robust registration and illumination by sparse representation. In *CVPR*, 2009.

[23] J. Wright, A. Yang, A. Ganesh, S. Satry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Feburay 2009.

[24] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *CVPR*, 2008.

[25] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[26] K. Yu, W. Xu, and Y. Gong. Deep learning with kernel regularization for visual recognition. In *In Advances in Neural Information Processing Systems 21*, 2008.

[27] K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. In *In Advances in Neural Information Processing Systems 22*, 2009.