Learning Component-Level Sparse Representation Using Histogram Information for Image Classification

Chen-Kuo Chiang, Chih-Hsueh Duan, Shang-Hong Lai
National Tsing Hua University,
Hsinchu, 300, Taiwan
{ckchiang, g9862568, lai}@cs.nthu.edu.tw

Shih-Fu Chang
Columbia University,
New York, New York, 10027, USA
sfchang@ee.columbia.edu

Abstract

A novel component-level dictionary learning framework which exploits image group characteristics within sparse coding is introduced in this work. Unlike previous methods, which select the dictionaries that best reconstruct the data, we present an energy minimization formulation that jointly optimizes the learning of both sparse dictionary and component level importance within one unified framework to give a discriminative representation for image groups. The importance measures how well each feature component represents the image group property with the dictionary by using histogram information. Then, dictionaries are updated iteratively to reduce the influence of unimportant components, thus refining the sparse representation for each image group. In the end, by keeping the top K important components, a compact representation is derived for the sparse coding dictionary. Experimental results on several public datasets are shown to demonstrate the superior performance of the proposed algorithm compared to the-state-of-the-art methods.

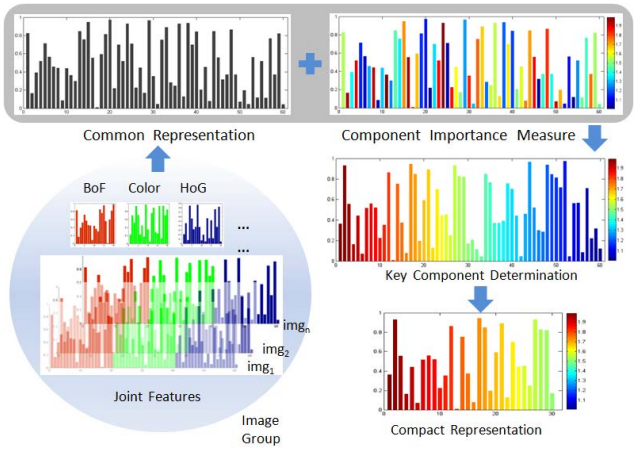

Figure 1. Learning component-level sparse representation for image groups. Histogram-based features, like BoF, color histogram or HoG, are extracted to form joint features. Then a common representation is obtained by learning the dictionary in a sparse coding manner. Component-level importance is determined with an optimization formulation according to image-group characteristics. After the key components are determined, a compact representation can be computed for classification.

1. Introduction

Image classification is one of the most important topics in computer vision. Recently, sparse coding technique attracts more and more attention because of its effectiveness in extracting global properties from signals. It recovers a sparse linear representation of a query datum with respect to a set of non-parametric basis set, known as *dictionary* [3, 24]. In image classification problem, methods based on sparse coding or its variants mainly collect a set of image patches to learn the dictionaries [22, 26].

By representing an image with a histogram of local features, Bag of Words (BoW) models [25] have shown excellent performance, especially its robustness to spatial variations. Considering spatial information with BoW, Lazebnik et al. [10] built a spatial pyramid and extended the BoW model by partitioning the image into sub-regions and computing histograms of local features. Yang et al. [29] further extended Spatial Pyramid Matching (SPM) by using sparse coding. They provided a generalized vector quantization for sparse coding followed by multi-scale spatial max pooling.

Although these previous methods are robust to spatial variations, the histogram-based representation is sensitive to noise. In Figure 2, taking an image group of yellow lily for example, all pictures share the same subjects or main objects (yellow lily). Irrelevant objects (tree, wall) or cluttered background (sky, ground) inside the image introduce lots of noises and decrease the discrimination capability of histogram feature. In this work, a component level sparse representation is proposed by using histogram information. Here, each dimension (bin) of the histogram is

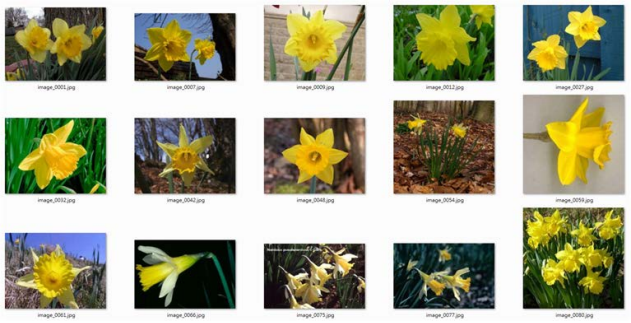

Figure 2. A sample image group of yellow lily.

referred to as a *component*. The importance of each component is measured via image group property. The common objects within the image group are considered to be important, while irrelevant objects and background should correspond to noisy information. The learning of dictionaries for image groups in this work is aimed at best reconstruction simultaneously for data representation and discrimination for one image group against all the others.

In the proposed framework, several histogram-based features, like color histogram, BoW and HoG, are combined to represent one image within an image group. Dictionaries are learned to give a common representation for each image group. The importance measure can be used to determine if a feature component is useful information for common objects within the image group, noise from irrelevant objects, or background for each image. Then, a compact representation of dictionaries can be obtained by identifying key components and reducing the influence of the other irrelevant components. Figure 1 depicts the mechanism of the proposed component-level feature learning framework.

The main novelty of the proposed approach is to incorporate component-level importance into the sparse representation, which is accomplished by optimizing the relative reconstruction errors for the associated image groups. In contrast to the previous methods [8, 9] in dictionary learning, weight assignment is usually decided in feature-type level (i.e. all feature dimensions of the same type have the same weight) and heuristically based on logarithm penalty function of reconstruction error of different classes [15] to increase the discrimination capability. In the proposed algorithm, the importance measure is down to the component level (i.e. each feature dimension has its own weight) and determined according to individual image group properties. The importance measures are employed to obtain the refined sparse feature representations that are learned by enforcing image classification constraints, making the proposed sparse representations more discriminative.

We demonstrate the benefits of the proposed framework in the image classification task on several publicly available datasets, including Caltech101 [6], Oxford17 [18] and Oxford102 [19] datasets. Our experiments show that the proposed algorithm provides significant accuracy improvement over the state-of-the-art methods.

1.1. Related Works

Sparse coding, recently attracted a considerable amount of attention from researchers in various domains, is very effective in many signal processing applications. Given a signal x in R^m, a sparse approximation over a dictionary D in $R^{m\times k}$ is to find a linear combination of a few atoms from D that is close to the signal x, where the k columns selected from D are referred to as atoms. Originally, predefined dictionaries based on various types of wavelets have been used for this task [16]. Lately, learning the dictionary instead of using predefined bases has been shown to improve signal reconstruction significantly [15].

Dictionary learning of sparse representation is aimed to find the optimal dictionary that leads to the lowest reconstruction error with a set of sparse coefficients. Traditional approaches collect sets of overlapping training patches from different classes. Dictionaries are learned for each class. Each patch of the testing data is approximated by sets of sparse coefficients and sets of different dictionaries which give different residual errors. These are used as criteria to decide image classes [22, 26]. With the same concept, Wright et al. [28] exploited the sparse representation classification (SRC) method for robust face recognition. Mairal et al. [15] assumed a dictionary D_i associated to a class C_i should reconstruct this class better than the other classes. They introduced an additional term in the cost function for discrimination and showed good results in texture segmentation. Another variant is online dictionary learning [13]. This is due to the growing of very large training examples. Dictionary learning with iterative batch procedures which access the whole training set is not effective. Based on stochastic approximation, the online optimization algorithm for dictionary learning scales up to large datasets with millions of training samples.

Combining multiple discriminative features shows significant improvement for object recognition and image classification. The problem considers joint model along with multiple related data sets is often referred to as multi-task learning. It aims to find out a very few common classes of training samples across multiple tasks that are mostly useful to represent query data. Multi-task Joint Covariate Selection (MTJCS) [20] which penalizes the sum of L_2 norms of coefficients associated with each covariate group can be regarded as a combination group Lasso [30] and multi-task Lasso [32]. Another famous method is Multiple Kernel Learning (MKL). MKL can be seen to linearly combine kernel functions such that the classification performance can be improved by the combined global function [12] [27]. Dif-

ferent from MKL, the proposed method learns class specific dictionaries and weights for each dimension of the dictionaries. Although the kernel weights are sparse in MKL, it doesnt include any sparse coding techniques. Recently, Yuan and Yan [31] exploited such a joint sparse visual representation method to reconstruct a test sample with multiple features from just a few training samples for image classification.

In contrast to previous methods which decide a few useful training samples for representation, the proposed framework finds out important components for each classification task. Back to the yellow lily example in Figure 2, the previous methods may choose a set of individual features that are best for reconstruction, but the proposed method picks important components from these features while reducing the impact of the other unimportant components.

1.2. Organization

The rest of this paper is organized as follows. In Section 2 we revisit the sparse coding formulation and dictionary learning method. The histogram-based component-level sparse representation is presented in Section 3 along with the optimization formulation and reweighted update algorithm. The experimental results are provided in Section 4. Finally, we conclude this paper in Section 5.

2. Sparse Coding and Dictionary Learning

Sparse coding is to represent a signal as a linear combination of a few atoms of a dictionary. Dictionary learning methods [11, 3, 21] consider a training set of signals $X = [x_1, \ldots, x_n]$ in $R^{m\times n}$ and optimize the cost function:

$$R(D) = \frac{1}{n}\sum_{i=1}^{n} l(x_i, D) \tag{1}$$

where D in $R^{m\times k}$ is the dictionary with each column representing a basis vector, and l is a loss function. The sparse representation problem can be formulated as:

$$R(x, D) = \min_{\alpha\in R^k} \frac{1}{2}\|x - D\alpha\|_2^2 + \lambda\|\alpha\|_1 \tag{2}$$

where λ is a parameter that balances the tradeoff between reconstruction error and sparsity. The L_1 constraint induces sparse solutions for the coefficient vectors α. Using the LARS-Lasso algorithm [5], this convex problem can be solved efficiently. In this problem, the number of training samples n is usually larger than the relatively small sample dimension m. L_1 solution has also been shown to be more stable than the L_0 approach since very different non-zero coefficients in α may be produced when small variation is introduced to the input set in L_0 formulation.

Instead of predefined dictionaries, the-state-of-the-art results have shown that dictionaries should be learned from data. To prevent D from being arbitrarily large, the norms of the atoms are restricted to be less than one [15], where a convex set Ψ of matrices with this constraint is given by

$$\Psi \triangleq \{D \in R^{m\times k} \mid \forall j = 1, \ldots, k, d_j^T d_j \leq 1\} \tag{3}$$

A joint optimization problem with respect to the dictionary D and the coefficient vector α for the sparse decomposition is given by

$$\min_{D\in\Psi,\alpha\in R^k} \frac{1}{n}\sum_{i=1}^{n}(\frac{1}{2}\|x_i - D\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1) \tag{4}$$

By using the sparse coding on a fixed D to solve α and updating the dictionary D with a fixed α alternatively, the optimization problem can be solved iteratively to find the optimal solution.

Ramirez et al. [23] suggested modeling the data as a union of low-dimensional subspaces. Then, the data points associated with the subspaces can be spanned by a few atoms of the same learned dictionary. Dictionaries associated with different classes are formulated to be as independent as possible. Assume $X^{(p)}, p = 1, \ldots, C$, be a collection of C classes of signals and $D^{(p)}$ be the corresponding dictionaries, the optimization problem is rewritten as:

$$\begin{aligned}\min_{\{D^{(p)},A^{(p)}\}_{p=1\ldots C}} \sum_{p=1}^{C}\{\|X^{(p)} - D^{(p)}A^{(p)}\|_2^2 + \lambda\sum_{j=1}^{m_i}\|\alpha_j^{(p)}\|_1\} \\ +\eta\sum_{p\neq q}\|(D^{(p)})^T D^{(q)}\|_2^2\end{aligned} \tag{5}$$

where $A^{(p)} = [\alpha_1^{(p)} \ldots \alpha_{m_i}^{(p)}] \in R^{k\times m_i}$, each column $\alpha_j^{(p)}$ is the sparse code corresponding to the signal $j \in [1 \ldots m_i]$ in class p. From our experimental results, we obtain dictionaries for one image group by selecting the entire training set from the correct group.

3. Histogram-Based Component-Level Sparse Representation

Histogram-based representations have been widely used with the feature descriptors, e.g., HOG [4], BoW [25], and GLOH [17]. It provides very compact representation and captures global frequency of low-level features. In this section, we present a framework that determines the component-level importance of histogram information and combines it with a sparse representation, which is referred to as Histogram-Based Component-Level Sparse Representation (HCLSP) for the rest of this paper.

3.1. Component-Level Importance

Suppose we have a training set of image groups. Each image group is defined as a class. For the training set, we

have C classes, for each class index $p = 1, \ldots, C$. Denote $X^{(p)} = [x_1^{(p)}, \ldots, x_n^{(p)}]$ in $R^{m \times n}$ to be a set of training samples from class p, with each individual sample $x_i^{(p)}$ in R^m. The dictionaries trained from class p are represented by $D^{(p)}$. Given the training data and the dictionaries, the sparse coefficient vector α can be obtained by solving equation (2) using LARS-Lasso or other standard algorithms. Denote the reconstruction error for the training set $X^{(p)}$ by using the dictionaries $D^{(p)}$ as:

$$R^{(p)}(X^{(p)}) = \sum_{i=1}^{n} |x_i^{(p)} - D^{(p)}\alpha_i^{(p)}| \tag{6}$$

where $R^{(p)}(X^{(p)}) \in R^m$. In contrast to the previous methods based on L_2-norm minimization, here the reconstruction error is represented by the sum of absolute values of the difference from the training data and its reconstruction. Similarly, the reconstruction error for the training set $X^{(p)}$ by using dictionary $D^{(q)}$ is represented by $R^{(q)}(X^{(p)})$.

For an image group of yellow lily, if we take color histogram as image feature, for example, and form our training set, every histogram in this group should have a certain nontrivial amount in the yellow-related components whereas other components containing background or irrelevant objects may have large or small values in the histogram. The basic idea of component-level importance is: the components representing common objects or subjects show lower reconstruction errors while cluttered background introducing large amount of reconstruction errors for the corresponding components. This is used as an indication to the importance measure of components.

Denote $\beta^{(p)}$ in R^m to be the importance for data of class p. The objective function can be formulated as:

$$\min_{\beta^{(1)} \ldots \beta^{(C)}} \sum_{p=1}^{C} ((\beta^{(p)})^T R^{(p)}(X^{(p)}) - (\beta^{(\hat{c})})^T R^{(\hat{c})}(X^{(p)}))$$
$$\textit{subject to} \quad 0 \le \beta_j^{(p)} < 1, \sum_{j=1}^{m} \beta_j^{(p)} = 1 \tag{7}$$
$$\textit{where} \quad \hat{c} = \arg \min_{\{q, q \neq p\}} R^{(q)}(X^{(p)})$$

Minimizing the objective function enforces the large reconstruction error in component j to have a smaller importance value $\beta_j^{(p)}$. Considering the importance measure with reconstruction from one image class against all the other classes over the entire training set, the second term in the objective function is a penalty term for the minimal reconstruction error computed from dictionaries of other classes. The component importance vectors $\beta^{(p)}$ for all classes are learned simultaneously in the above formulation. Thus, this global importance measure gives more discriminative power for the classification problem. The solution can be obtained by solving the optimization via linear programming.

Algorithm 1 : The Histogram-Based Component-Level Sparse Representation with reweighted update algorithm.

1. **Define:** For class p at iteration t, define the reconstruction error $R^{p,t}$ using dictionary $D^{p,t}$, importance $\beta^{p,t}$ and the updated weight $w^{p,t}$.
2. **Input:** Training set $X^p = \{x_i\}_{i=1}^n, x_i \in R^m$, collected from image group p. Testing data y. Regularization parameter λ. A threshold ρ. Error bound parameter ϵ and iteration bound T.
3. **Training:**
4. **Initialization:** Set $t \leftarrow 1$. Choose the whole training set X^p as dictionary $D^{p,1}$ for image group p. $w^{p,0} \in R^m$,$w_i^{p,0} = 1$, $R_i^{p,0}(X^{p,0}) = 0$, $\beta_j^{p,0} = 1$, for $j = 1 \ldots m$.
5. **repeat** { Main loop}:
6. $D^{p,t} \leftarrow D^{p,1} .* w^{p,t-1}$,
7. Solve $\alpha^{p,t}$ by X^p and $D^{p,t}$,
8. Calculate $R^{p,t}(X^p)$, $R^{q,t}(X^p)$, $\forall p, q \in \{1 \ldots C\}, q \neq p$,
9. Solve $\beta^{p,t}$ by equation (7),
10. $\Delta R = \sum_{p=1}^{C}((\beta^{p,t})^T R^{p,t}(X^p) - (\beta^{p,t-1})^T R^{p,t-1}(X^p))$
11. $\delta^{p,t} \leftarrow w^{p,t-1} .* \beta^{p,t}$
12. $w_j^{p,t} = \begin{cases} \beta_j^{p,t} & \textit{if} \quad \beta_j^{p,t} < \rho \\ 1 & \textit{otherwise} \end{cases}$
13. $w^{p,t} \leftarrow w^{p,t-1} .* w^{p,t}$,
14. $t \leftarrow t + 1$,
15. **until** $\Delta R < \varepsilon$ or $t < T$

3.2. Reweighted Update

In this section, we present the algorithm that incorporates the learned component importance into a sparse representation. According to the previous section, the importance for all components can be derived for each class. It indicates how representative these components are for a specific image group. We prefer meaningful components rather than irrelevant ones for better reconstruction by using the dictionary from the right image group instead of those from other image groups. Choosing incorrect components leads to ex-

tremely large reconstruction error. Taking a worst case for example, it is like using red components from a red rose image group to reconstruct a yellow lily image.

This leads to a component weight. The component weight $w_j^{(p)} \in R^m$ with $j = 1, \ldots, m$ can be defined as:

$$w_j^{(p)} = \begin{cases} \beta_j^{(p)} & \textit{if} \quad \beta_j^{(p)} < \rho \\ 1 & \textit{otherwise} \end{cases} \tag{8}$$

where ρ is a threshold. Note that $\beta_j^{(p)}$ is a normalized value between 0 and 1. The more important the component is, the bigger value it is. We assign the weight of important components to 1 while reweighting the unimportant components to small values. This is used to reweight each dimension in the dictionary. The benefit of reweighting is that only the important components in the dictionary are used for data reconstruction. The reconstruction errors introduced by unimportant components are reduced if it is reconstructed from the dictionary of the correct class. This makes the dictionaries more discriminative for their own classes.

Intuitively, we assume most components are useful for reconstruction. Only a few unimportant components are selected based on their weight values. Thus, the proposed algorithm is carried out by an iterative reweighted update process. In each iteration, dictionary $D^{(p)}$ is adjusted by weight $w^{(p)}$. Then, new dictionaries are used to solve $\alpha^{(p)}$. The importance vector $\beta^{(p)}$ is obtained from the new sparse coding coefficients and used to determine the weight vector $w^{(p)}$. The purpose of weight vector $w^{(p)}$ is to reweight components in the dictionary. The value is accumulated and only a small portion of components will be chosen as unimportant ones in each iteration. The importance vector $\beta^{(p)}$ is used to decide the image class during the testing. This gives the decision criterion as follows:

$$c^* = \arg\min_p (\delta^{(p),t^*})^T |y - D^{(p)}\alpha^{(p)}| \tag{9}$$

where t^* is optimal iteration number, δ serves as the combination of dictionary adjusted weight vector $w^{(p),t^*-1}$ for image class p and the importance measure $\beta^{(p),t^*}$ for the decision. The details are given in Algorithm 1.

3.3. Compact Representation

Since the importance vector $\beta^{(p)}$ is used for the decision of the image class, we can prune the unimportant components and give a compact representation for the dictionaries:

$$\beta_j^{(p)} = \begin{cases} 0 & \textit{if} \quad \beta_j^{(p)} < \varphi \\ \beta_j^{(p)} & \textit{otherwise} \end{cases} \tag{10}$$

where φ is a threshold to prune a certain ratio of components. In our implementation, the importance vector $\beta^{(p)}$ is used to prune the dimension of dictionary $D^{(p)}$ after a given number of iterations. The pruned dictionary and the weight vector for each class are saved. In the testing phase, the important components are selected from the test data. Then, the reconstruction is performed by the compact dictionary and pruned data to decide the final classification results.

Table 1. Classification accuracy (%) on Oxford 17 Category Flower Dataset using different features and their combination.

Accuracy	Color	BoW	HoG	ALL
NN	36.47	44.63	35.96	39.56
SRC[28]	36.91	49.71	41.18	58.82
MCLP[7]	42.62	50.38	42.33	66.74
KMTJSRC[31]	44.80	51.72	44.51	69.95
HCLSP	45.15	52.34	43.38	63.15
HCLSP_ITR	50.15	55.68	46.76	67.06

Table 2. Classification accuracy (%) on Oxford 102 Category Flower Dataset using different features and their combination.

Accuracy	Color	BoW	HoG	ALL
NN	33.52	22.76	19.39	36.27
SRC[28]	24.43	18.05	19.58	37.85
MCLP[7]	36.74	29.49	30.96	58.68
KMTJSRC[31]	36.67	30.16	29.14	57.00
HCLSP	37.37	29.73	31.58	51.77
HCLSP_ITR	44.53	32.35	39.01	60.14

Table 3. Classification accuracy (%) on Caltech101 Dataset using different features and their combination.

Accuracy	Color	BoW	HoG	ALL
NN	29.65	51.26	44.85	38.83
SRC[28]	14.06	53.39	47.61	52.64
MCLP[7]	33.68	57.15	55.34	65.77
KMTJSRC[31]	18.14	48.93	46.25	53.21
HCLSP	35.12	56.03	58.12	60.49
HCLSP_ITR	35.43	58.24	59.94	68.41

4. Experimental Results

We apply the proposed Histogram-based Component-Level Sparse Representation (HCLSP) to the task of image classification in our experiments. The extension of HCLSP with iterative reweighting is referred to as HCLSP_ITR in this section. Our experiments were performed on the following datasets: Oxford 17 category, Oxford 102 category and Caltech 101 datasets.

4.1. Experimental Settings

Three types of features are employed in the proposed framework: color histogram, BoW and HoG. The combination of these three features is also tested in the experiment. The color histogram with R, G, B channels is quantized to 1331 bins. For the BoW features, we use the Matlab code provided by [2]. The BoW is constructed via hierarchical K-means which provides 1555-dimensional BoW features. We also extract HoG features [1] from 3 levels of 8 bins and 360 degrees, which give a 680-dimensional feature vector. Then, we concatenate the above features to a combination of 3566-dimensional feature vector. In our implementation, the regularization parameter f of sparse coding is set to 0.15. The threshold l is set to adjust 10% unimportant components in each iteration. Iteration bound T is empirically set to 10 in our experiments. The threshold φ in eq. (10) for compact representation is set to cut 20% least important components. We used the tool developed by Mairal [14] to solve the sparse coding coefficients. The accuracies shown for HCLSP_ITR also include the contribution from the compact representation described in Section 3.3. All experiments were repeated three times to obtain the average accuracies. We compare the performance of the proposed HCLSP and HCLSP_ITR with the following methods:

- Nearest neighbor search (NN): In the baseline method, the image is classified to the class label of its nearest neighbor in the feature space.
- Sparse representation classification (SRC) [28]. Each feature vector is approximated by using the C sets of different dictionaries and the sparse coefficients α. The class label is decided from the class with minimal residue.
- Visual classification with multi-task joint sparse representation (KMTJSRC) [31]: Each feature is represented as a linear combination of the corresponding training features. The classification decision is made from the overall reconstruction error of an individual class.
- Multiclass LPboost (MCLP) [7]: It is the representative of the multiple kernel learning methods from literature [7].

4.2. Oxford 17 Category Flower Dataset

This dataset contains 17 categories of flower images. 80 images in each class make the total 1360 images. 40 images are randomly selected as training samples and the rest is used as the test set. Table 1 lists the accuracies of the proposed method and the above four methods for comparison. For experiments with single feature types, the proposed HCLSP is very competitive compared to KMTJSRC and MCLP in terms of accuracy. Basically, both KMTJSRC and the proposed HCLSP are extensions of SRC and outperform the previous method. However, KMTJSRC shows significant accuracy improvement with feature combination. On the other hand, NN is comparable to SRC and feature combination provides little help to NN. Note that the accuracy of KMTJSRC is much higher (about 88.1% reported in [31]) than the accuracy reported here mainly because it combines 7 different features in their implementation. For a fair comparison among different methods, we use the same three features for all the methods in our experiments.

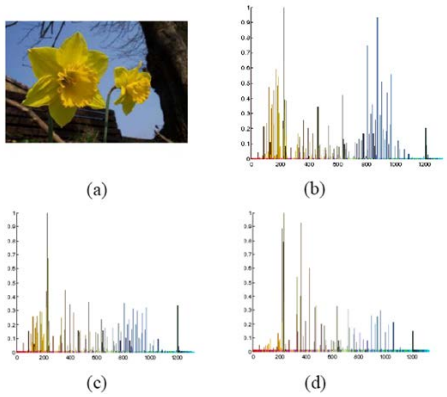

Figure 3. (a) A sample image from yellow lily image group. (b) The original histogram of the image. (c) The histogram after the first iteration. (d) The histogram adjusted by weights after the best number of iterations.

4.3. Oxford 102 Category Flower Dataset

This dataset contains 102 categories of flower images. The total number of images is 8189. The image number in each category is ranged from 40 to 250. 20 images are randomly selected as the training samples and the rest forms the test set. The accuracies of the proposed algorithms and other methods on the 102 category dataset are listed in Table 2. Comparing the accuracies with single feature types, we can see the proposed HCLSP is slightly better than KMTJSRC and MCLP with color and HoG features. After iteratively reweighting, our HCLSP_ITR shows improvement over HCLSP and outperforms all the other methods for all four feature options. Interestingly, some accuracies of SRC are a little worse than those of NN. In addition, when the number of category grows larger, color histogram shows better results than BoW. In overall, feature combination shows great improvement over single-feature results.

4.4. Caltech-101 Dataset

Caltech101 is a very popular, yet challenging, test set for object recognition. It contains 101 categories of objects. For this dataset, we randomly select 15 training samples and use the rest as the test set. Table 3 lists the classifi-

cation accuracies of different algorithms with different feature options. We can observe that HCLSP is very competitive compared to MCLP. Both HCLSP and MCLP outperform KMTJSRC in this dataset. HCLSP_ITR show superior performance over the other methods, including MCLP, KMTJSRC, SRC and NN. It is evident that HCLSP_ITR improves HCLSP with all different feature options listed in the experiment. The performance of SRC is comparable to that of KMTJSRC for the experiments on this dataset.

4.5. Results of Iterative Reweighting

In this section, we show the effectiveness of the proposed iterative reweighting approach. In each iteration, the proposed algorithm uses the importance as the weight to adjust unimportant components from the dictionary. This leads to the reduction of the reconstruction error. During training, the vector $\delta^{(p),t^*}$ from the best iteration t^* is saved for class p. Then, the decision can be made according to $\delta^{(p),t^*}$ and $D^{(p)}$. We follow the experimental settings and randomly select the training and test data. The mean accuracy is obtained by repeating three experiments. Note that, in the proposed algorithm, the accuracy of HCLSP represents the results of the first iteration. The accuracy of HCLSP_ITR is obtained from the best iteration. In detail, the best numbers of iterations for color, BoW, HoG and combination of all features are 2, 5, 4 and 3, respectively. From Table 1 to Table 3, the accuracy of HCLSP is improved significantly by HCLSP_ITR mainly due to the proposed reweighted update algorithm.

To demonstrate how the dictionary reweights during the iterations, we apply the component weights obtained from HCLSP and HCLSP_ITR to a histogram of a sample image. For better visualization, the largest component of the histogram is scaled to 1 to better show the difference among all components. Figure 3(a) depicts a sample image selected from the first class of Oxford 17 category dataset. It has a main object of big yellow lily along with the sky and other background. Figure 3(b) depicts the histogram which mainly contains two groups of color, yellow and blue. In Figure 3(c) and 3(d), the components of various blue colors are suppressed by the proposed algorithm while important components from yellow color remain in the high peak. It is interesting to observe that only a few important components are left after the iterations that make the proposed method like a two-dimensional sparse coding. In the first dimension the training samples are selected for best reconstruction and in the second dimension only a few important components are selected to reduce the reconstruction errors from unimportant components.

4.6. Dimension Reduction

In the proposed algorithm, dimension reduction is performed as the last step in the end of the proposed reweighted update algorithm to remove the least important components. In other words, the more iterations during the training, the more dimension reduction obtained for compact representation. In our implementation, we select an appropriate threshold φ to cut 20% unimportant components. Table 4 shows the dimension of dictionary in different numbers of iterations. As we mentioned in Section 4.5, the best number of iterations usually occurs within 5 iterations and in average around 2 or 3 iterations. We list the dimension reduction results for iteration 0, 1, 3, 5. Iteration 0 means the initial dimension of the original feature. It shows that the testing is performed on almost half of the components from the original feature. Thus, the proposed method is more efficient in term of execution time than the previous methods.

Table 4. The dimension reduction results from each iteration using a cutting threshold φ to cut 20% components from the least important components.

Iteration	0	1	3	5
Color	1331	1065	682	437
BoW	1555	1244	797	510
HoG	680	544	349	223
ALL	3566	2853	1826	1169

5. Conclusion

We presented a novel framework of learning component-level sparse representation for image groups to achieve optimal classification. This new representation is characterized by learning feature properties for each individual image group. The common objects or cluttered background are modeled by importance measure in a joint energy minimization with both sparse dictionary and component-level importance measure. This gives a more discriminative representation for image groups. In addition, the proposed algorithm was further improved with the proposed iterative reweighting scheme. In the end, by keeping important components, a compact representation is computed for the sparse coding dictionary.

Experimental comparisons among the proposed method and some representative methods were performed on several well-known datasets. The results show that the proposed method in general can provide superior or comparable performance in image classification compared to the-state-of-the-art methods. In addition, the performance of our iterative component reweighting approach showed significant improvement through our experiments. In the future, we would like to investigate a more robust measure for image group characteristics and incorporates the measure into a two-dimensional sparse coding representation.

Acknowledgements

This work was supported in part by National Science Council of Taiwan under the grant NSC 99-2220-E-007-016 and the SOC Joint Research Lab project sponsored by NOVATEK.

References

[1] http://www.robots.ox.ac.uk/ vgg/research/caltech/phog.htm.

[2] http://www.vlfeat.org/ vedaldi/code/bag/bag.html.

[3] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311 –4322, 2006.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886 –893 vol. 1, 2005.

[5] B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[6] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on*, page 178, 2004.

[7] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *Computer Vision, 2009 IEEE 12th International Conference on*, 29 2009.

[8] S. Hasler, H. Wersing, and E. Korner. Class-specific sparse coding for learning of object representations. In *Artificial Neural Networks: Biological Inspirations V ICANN 2005*. 2005.

[9] S. Hasler, H. Wersing, and E. Körner. Combining reconstruction and discrimination with class-specific sparse coding. *Neural Comput.*, 19:1897–1918, July 2007.

[10] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006.

[11] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *In NIPS*, pages 801–808. NIPS, 2007.

[12] Y.-Y. Lin, T.-L. Liu, and C.-S. Fuh. Local ensemble kernel learning for object category recognition. In *CVPR*, 2007.

[13] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 689–696, New York, NY, USA, 2009. ACM.

[14] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, 11:19–60, March 2010.

[15] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, 2008.

[16] S. Mallat. *A wavelet tour of signal processing*.

[17] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615 –1630, 2005.

[18] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006.

[19] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Computer Vision, Graphics Image Processing, 2008. ICVGIP '08. Sixth Indian Conference on*, pages 722 –729, 2008.

[20] G. Obozinski, B. Taskar, and M. I. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20:231–252, April 2010.

[21] B. A. Olshausen and D. J. Fieldt. Sparse coding with an overcomplete basis set: a strategy employed by v1. *Vision Research*, 37:3311–3325, 1997.

[22] G. Peyré. Sparse modeling of textures. *J. Math. Imaging Vis.*, 34:17–31, May 2009.

[23] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 3501–3508. IEEE, 2010.

[24] P. Sallee and B. A. Olshausen. Learning Sparse Multiscale Image Representations. *NIPS*, 15:1327–1334, 2003.

[25] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470 –1477 vol.2, 2003.

[26] K. Skretting and J. H. Husøy. Texture classification using sparse frame-based representations. *EURASIP J. Appl. Signal Process.*, 2006:102–102, January 2006.

[27] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *Proceedings of the IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil*, October 2007.

[28] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210 –227, 2009.

[29] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794 –1801, 2009.

[30] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal Of The Royal Statistical Society Series B*, 68(1):49–67, 2006.

[31] X.-T. Yuan and S. Yan. Visual classification with multi-task joint sparse representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3493 –3500, 2010.

[32] J. Zhang. A probabilistic framework for multi-task learning. Technical Report CMU-LTI-06-006, Language Technologies Institute, CMU, 2006.