Visual Classification with Multi-Task Joint Sparse Representation

Xiao-Tong Yuan, Shuicheng Yan
Department of Electrical and Computer Engineering
National University of Singapore, Singapore
{eleyuanx, eleyans}@nus.edu.sg

Abstract

We address the problem of computing joint sparse representation of visual signal across multiple kernel-based representations. Such a problem arises naturally in supervised visual recognition applications where one aims to reconstruct a test sample with multiple features from as few training subjects as possible. We cast the linear version of this problem into a multi-task joint covariate selection model [15], which can be very efficiently optimized via kernelizable accelerated proximal gradient method. Furthermore, two kernel-view extensions of this method are provided to handle the situations where descriptors and similarity functions are in the form of kernel matrices. We then investigate into two applications of our algorithm to feature combination: 1) fusing gray-level and LBP features for face recognition, and 2) combining multiple kernels for object categorization. Experimental results on challenging real-world datasets show that the feature combination capability of our proposed algorithm is competitive to the state-of-the-art multiple kernel learning methods.

1. Introduction

The problem of recovering sparse linear representation of a query datum with respect to a set of reference data has recently received wide interests in signal processing and computer vision literatures [3][22][23]. It has been discovered in neural science [17] that the human vision system seeks a sparse coding for the incoming image using a few words in a feature vocabulary. Olshausen et al. [16] introduced a Bayesian framework to simulate the sparse coding mechanism of human vision system. In the setting of face recognition, Wright et al. [22] demonstrated that the $\ell_1$-norm regularized reconstruction error minimization leads to robust face recognition. For semi-supervised learning, Yan and Wang [23] proposed the $\ell_1$-graph in which the graph adjacency structure and the graph weights are derived simultaneously via datum-wise linear sparse coding.

In the language of linear regression, by viewing training datum as observation of covariate and the query datum as response, the sparse representation problem can be casted into a problem of sparse covariate selection via linear regression. In this paper, motivated by the success of multi-task sparse linear regression models [10][15][25], we investigate into the problem of multi-task joint sparse representation along with its applications in visual recognition. The general problem of jointly estimating models from multiple related data sets is often referred to as “multi-view learning” or “multi-task learning” in the machine learning literature. We adopt the following terminology from this literature: a representation task is defined on a training set $X^k = [X^k_1, ..., X^k_J]$ with $J$ classes and a query data $y^k$ to be represented, where $k \in \{1, ..., K\}$ indexes tasks. We aim to find out across these $K$ tasks a very few common classes of training samples that are mostly useful for query data reconstruction. For object recognition, we may generate $K$ representation tasks from $K$ different modalities of features associated with the same visual input. In this setting, the constraint of joint sparsity across different tasks is valuable since different tasks may favor different sparse reconstruction coefficients, yet the joint sparsity may enforce the robustness in coefficient estimation. The goal of joint sparsity can be achieved by imposing $\ell_{1,2}$ mixed-norm
penalty on the reconstruction coefficients. The formulated objective is composed of a quadratic part and a non-smooth part, and thus we conventionally adopt the accelerated proximal gradient method [12] for optimization with fast convergence rate guaranteed. The classification is ruled in favor of the class that has the lowest total reconstruction error. We then naturally develop two kernel-view extensions of this method to deal with the situations where image descriptors and features are represented by kernel-based similarity matrices. Taking flower image classification as an example, Figure 1 depicts the working mechanism of our algorithm.

We applied the proposed joint sparse representation and classification algorithm to several visual recognition problems, which include:

- **Multi-cue face recognition:** The gray-level images and LBP [1] images from all subjects are used to develop two dictionary matrices during the training session. The gray-level image and LBP image from each test face are represented as linear combinations of the corresponding training images, thereby resulting in a two-task joint sparse representation and classification problem. We show that such a multi-task representation mechanism can elegantly fuse the advantages of gray-level and LBP features to improve recognition performance.

- **Multiple kernels based object categorization:** We apply the proposed algorithm to fuse the discriminative capabilities of different complementary descriptors for object categorization. Many hand-crafted descriptors have been proposed in the literature for measuring visual similarity [2][5][8][19]. These descriptors are typically encoded in the form of kernel matrices. We show that the kernel-view extensions of our algorithm are competitive to the state-of-the-art multiple kernel learning methods [7][14][21] for object recognition.

### 1.1. Related Works

Methodologically, our algorithm is motivated by the recent advance in sparse learning called as Multi-task Joint Covariate Selection (MTJCS) [15]. The MTJCS can be regarded as a combinational model of group Lasso [24] and multi-task Lasso [25]. By penalizing the sum of $\ell_2$ norms of the blocks of coefficients associated with each covariate group (or factor) across different classification problems, similar sparsity patterns in all models are encouraged. In this paper, we introduce this powerful sparse learning model into computer vision as a joint sparse visual representation method. We further provide two kernel extensions of this model to fuse the discriminative power of different visual descriptor kernels in recognition problems. In our implementation, instead of the blockwise path-following scheme used in [15], we choose the accelerated proximal gradient method [12] as optimization tool for its simplicity, efficiency, and kernelization capability.

The sparse representation has received wide attentions in the practice of visual recognition. Wright et al. [22] exploited the sparse representation classification (SRC) method for robust face recognition. They assumed that the training samples of a particular class approximately form a linear basis set for any test sample belonging to this class. The Lasso regularization was proposed to select the representative training samples from the entire training set. Lasso however tends to select a single sample from a group of correlated training samples and thus does not promote the representation of the test sample in terms of all the training samples from the correct group. To overcome this problem, Majumdar and Ward [11] proposed two alternative regularization methods, Elastic Net [26] and group Lasso, to improve SRC. Both these regularization methods favor the selection of multiple correlated training samples to represent the test sample.

Combining multiple discriminative features for object recognition is a recent trend in class-level object recognition and image classification. One popular method in computer vision is Multiple Kernel Learning (MKL), which can be seen to linearly combine similarity functions between images such that the combined similarity function yields improved classification performance [9][21]. Several SVM ensemble methods inspired by linear programming Boosting have also been proposed for multi-kernel object classification [7]. Differently, our proposed algorithm considers SRC in the setting of multi-task learning and performs well for feature combination.

### 1.2. Organization

The remainder of this paper is organized as follows. In Section 2 we present the multi-task joint sparse representation algorithm. The optimization details along with the final classification rule are stated in Section 3. The kernel-view extensions of our method are provided in Section 4. The applications of our method to face recognition and object categorization are studied in Section 5. Finally, we conclude this work in Section 6.

### 2. Multi-task Joint Sparse Representation

Suppose we have a training set with $J$ classes in which each sample has $K$ different modalities of features (e.g., colour, texture and shape). For each modality (task) index $k = 1, \ldots, K$, denote $X^k = [X^k_1, \ldots, X^k_J]$ the training feature matrix in which $X^k_j \in \mathbb{R}^{m_k \times p_j}$ is associated with the $j$th class. Here $m_k$ is the dimensionality of the $k$th modality, $p_j$ is the number of training samples in class $j$, and $\sum_{j=1}^J p_j = p$ is the total number of training samples.
a test sample with the corresponding $K$ modalities of representations, $y^k \in \mathbb{R}^{m_k}$, we consider a supervised $K$-task linear representation problem as follows

$$y^k = \sum_{j=1}^{J} X^k_j w^k_j + \epsilon^k, k = 1, ..., K,$$

where $w^k_j \in \mathbb{R}^{m_j}$ is a reconstruction coefficient vector associated with the $j$th class, and $\epsilon^k \in \mathbb{R}^{m_k}$ is the residual term. Denote $w^k = [w^k_1^T, ..., w^k_J^T]^T$ the representation coefficients in task $k$ and $w_j = [w^k_j^1, ..., w^k_j^K]$ the representation coefficients from the $j$th class across different tasks. Let $W = [w^k_j]_{k,j}$. Our multi-task joint sparse representation is formulated as the solution to the following problem of multi-task least square regressions with $\ell_{1,2}$ mixed-norm regularization:

$$\min_W \frac{1}{2} \sum_{k=1}^{K} \left\| y^k - \sum_{j=1}^{J} X^k_j w^k_j \right\|^2_2 + \lambda \sum_{j=1}^{J} \left\| w^k_j \right\|_2. \quad (1)$$

This problem is known as multi-task joint covariate selection in Lasso related research [15].

3. Kernelizable Optimization and Classification

For model optimization, we chose to use the popularly applied Accelerated Proximal Gradient (APG) method [12][20][4] to efficiently solve problem (1). In our setting, the proposed APG method is characterized by inner product of features, thereby facilitates the kernel extension which shall be addressed later on.

3.1. The $\ell_{1,2}$ Mixed-norm APG Algorithm

The proposed APG algorithm comprises alternately updating a weight matrix sequence $\{W^t = [w^k_j]_{j \geq 1}\}$ and an aggregation matrix sequence $\{V^t = [v^k_{j,t}]_{j \geq 1}\}$. Each iteration consists of two steps: 1) a generalized gradient mapping step to update matrix $W^{t+1}$ with current aggregation matrix $V^t$, and 2) an aggregation forward step to update $V^{t+1}$ by linearly combining $W^{t+1}$ and $V^t$.

The generalized gradient mapping step. Given the current matrix $V^t$, we update $W^{t+1}$ according to the result in [18] as follows

$$\hat{w}^k_{j,t+1} = \hat{v}^k_{j,t} - \eta \nabla \hat{v}^k_{j,t}, \quad k = 1, ..., K,$$

$$\hat{w}^k_{j,t+1} = \left[ 1 - \frac{\lambda \eta \|\hat{w}^k_j\|_2}{\|\nabla \hat{v}^k_{j,t+1}\|_2} \right] \hat{v}^k_{j,t+1}, \quad j = 1, ..., J. \quad (2)$$

where $\nabla \hat{v}^k_{j,t} = -(X^k)^T y^k + (X^k)^T X^k \hat{v}^k_{j,t}$, $\eta$ is the step size parameter, and $[,]_+ = \max(\cdot, 0)$.

The aggregation step. We then construct a linear combination of $W^t$ and $W^{t+1}$ to update $V^{t+1}$ as follows:

$$\hat{V}^{t+1} = \frac{\alpha_t}{\alpha_t} W^{t+1} + \frac{\alpha_t}{\alpha_t} (W^{t+1} - W^t). \quad (3)$$

Here the sequence $\{\alpha_t\}_{t \geq 1}$ can be conventionally set to be $\alpha_t = 2/(t + 2)$ [20], as applied in our implementation.

3.2. Classification Rule

For each task $k$, using only the optimal coefficients $\hat{w}^k_j$ associated with the $j$th class, one can approximate the $k$th modality $y^k$ of a test sample as $\hat{y}^k = X^k_j \hat{w}^k_j$. The decision is ruled in favor of the class with the lowest total reconstruction error accumulated over all the $K$ tasks:

$$j^* = \arg \min_{j} \sum_{k=1}^{K} \| y^k - X^k_j \hat{w}^k_j \|^2_2. \quad (4)$$

where $\{\theta^k\}_{k=1}^{K}$ are the weights that measure the confidence of different tasks in final decision. We may simply pick $\theta^k = 1/K$, or as described below to optimize these parameters in a LPBoost way [7] on a validation set.

3.3. Boost the Task Weights on Validation Set

Suppose a validation set $Y_{val} = \{y^1_l, ..., y^K_{l_M}\}$ is available, here $l_i \in \{1, ..., J\}$ is the class label of the sample $y^j$. After multi-task joint sparse representation for each validation sample, we obtain a set of reconstruction errors $\{a^k_{i,j} = \|y^k_l - X^k_j \hat{w}^k_j\|_2|i = 1, ..., M, j = 1, ..., J, k = 1, ..., K\}$. According to the classification rule (4), the task weights should be chosen to enforce that the validation samples satisfy $\sum_{k=1}^{K} \theta^k a^k_{i,j} \leq \sum_{k=1}^{K} \theta^k a^k_{i,l_i} - \rho, \forall j \neq l_i$ with some margin parameter $\rho$. In practice, by introducing a slack variable $\xi_i$ for the $i$th validation sample, the optimal weight $\theta^k$ can be estimated by solving the following linear program

$$\min_{\rho, \xi, \theta} -\rho + \frac{1}{M} \sum_{i=1}^{M} \xi_i,$$

s.t.

$$\sum_{k=1}^{K} \theta^k a^{k}_{i,l_i} - \xi_i \leq \sum_{k=1}^{K} \theta^k a^{k}_{i,j} - \rho, \quad \forall i = 1, ..., M, \forall j = 1, ..., J, j \neq l_i,$$

$$K \theta^k = 1, \rho \geq 0, \xi_i \geq 0, \theta^k \geq 0, \forall i, k. \quad (5)$$

which can be efficiently optimized via standard linear programming solvers.

Algorithm 1 summaries the details of the optimization and classification procedure of our multi-task joint sparse representation and classification (MTJSRC) model. Note that when $K = 1$, MTJSRC reduces to the SRC method.
that the gradient mapping step (2) only involves the inner product of features, and thus can be straightforwardly extended to solve problem (6). Let \( C^k = \phi^k(X^k)^T \phi^k(X^k) \) with \( \phi^k(X^k) = [\phi^k(X^k_{j,1}), ..., \phi^k(X^k_{j,p_j})] \) be the training kernel matrix associated with the \( k \)th modality of feature, and \( h^k = \phi^k(X^k)^T \phi^k(y^k) \) be the test kernel vector associated with the \( k \)th modality, we have that \( \nabla h^k = -h^k + G^k \tilde{v}^k \) in (2). When the reconstruction coefficients are optimized, the classification decision is made by the following rule

\[
j^* = \arg \min_j \sum_{k=1}^{K} \theta^k \|y^k - \phi^k(X^k_j) \tilde{u}^k_j \|^2_2
\]

\[
= \arg \min_j \sum_{k=1}^{K} \theta^k (-2h^k \tilde{w}^k_j + (\tilde{w}^k_j)^T G^k_j \tilde{w}^k_j)
\]
5. Experiments

To evaluate the effectiveness of our proposed method for object classification by feature combination, we systematically apply it to several face recognition and multi-class object categorization data sets.

5.1. An Illustrative Example: Face Recognition

We first give a face recognition experiment to illustrate the feature combination capability of MTJSRC. We use the Extended Yale Face Database B\(^1\) for performance evaluation. The extended Yale B database contains 16128 images of 38 human subjects under 9 poses and 64 illumination conditions. We use 64 near frontal cropped face images (32 × 32 pixels, gray-level) for each individual in our experiment. We simply choose the gray-level and the LBP\(_{8,1}\) features for combination test. Therefore we have two sparse representation tasks. For each subject \(j\), we randomly select \(p_j \in \{5, 10, 20, 30, 40, 50\}\) images for training, and the rest for test. The reported mean and standard variance of recognition accuracy are estimated over 50 random splits.

Figure 2 shows the accuracy curves of single features and their combination. From this figure we can see that the multi-task joint sparse representation can well combine individual features to improve the performance. Some example classifications (with \(p_j = 40\)) are provided in Figure 3(a). As expected, the failing cases by gray-level feature are mainly due to poor illumination condition (see the first five columns in Figure 3(a)). The LBP feature, on the other hand, is robust to illumination variations and thus can classify these five samples correctly. However, LBP still fails in some other cases (see the last two columns in Figure 3(a)) that gray-level recognize correctly. The combination of two complementary features can fuse their benefits and achieve better classification performance. In detail, Figure 3(b) & 3(c) show the sparse reconstruction coefficients and residuals for the test images of the 1st and 7th column in Figure 3(a).

5.2. Object Categorization

On multiple kernels based object classification, we perform two sets of experiment on challenging databases which include two Oxford flower data sets and the Caltech101 object categorization data set.

5.2.1 Comparing Algorithms

We compare the overall recognition performances of our proposed algorithms with the following methods

- Baseline: Feature combination based on nearest subspace (NS). In this baseline, the column generation strategy is applied to handle kernel matrices, and the coefficients \(\hat{u}_j^k\) in (7) are independently learnt through regression for each task \(k\) and each class \(j\);
- Feature combination based on independent SRC. This method can be viewed as a simplification of our method without enforcing the joint sparsity across

\(^1\)http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html
tasks. The column generation strategy is applied to handle kernel matrices, and the coefficients $w^k_j$ in (7) are independently learnt by SRC for each task $k$.

- The representatives of multiple kernel learning methods from literatures [7][14][21].

Meanwhile, for single features, the kernel-views of our method reduce to the kernelizations of SRC, whose performance shall be compared with NS and SVM.

### 5.2.2 Oxford Flowers Data Sets

In this subsection, we present results on two Oxford flower data sets with 17 classes [13] and 102 classes [14] respectively. For both data sets, the background of each image is removed using segmentation so as to extract features from the flowers alone and not from the surrounding vegetation.

Some details on data set description and experiment setup are given below:

**17 category data set** This data set consists of 17 species of flowers with 80 images per class, totalling 1360 images. The classification is carried out on the basis of $\chi^2$ distance matrices of clustered HSV, HOG, SIFT on the foreground internal region (SIFTint), SIFT on the foreground boundary (SIFTbdy) and three matrices derived from colour, shape and texture vocabularies. The authors of [14] provide the $\chi^2$ distance matrices of these seven features along with the three predefined training (17 $\times$ 40 images), validation (17 $\times$ 20 images) and test (17 $\times$ 20 images) splits on the database website.

**102 category data set** This larger data set consists of 8189 images divided into 102 flower classes. Each class consists of 40-250 images. The classification is carried out on the basis of $\chi^2$ distance matrices of clustered HSV, HOG, SIFTint and SIFTbdy. The data set is divided into a training set, a validation set and a test set. The training set and validation set each consist of 10 images per class. The test set consists of the remaining 6149 images (minimum 20 per class). The $\chi^2$ distance matrices of these four features along with a predefined training/validation/test split are publicly available on the database website.

On both data sets, we use the predefined splits as aforementioned for training and parameter selection. Kernel matrices are computed as $\exp(-\chi^2(x,x')/\mu)$ where $\mu$ is set to be the mean value of the pairwise $\chi^2$ distance on the training set. The parameter selection and task weights optimization are conducted on the validation set. For the test set, accuracy performance is measured per class, i.e., the final performance is the classification accuracy averaged over all classes.

The accuracies from our proposed algorithms along with baselines and several state-of-the-art results from literatures on the 17 category data set are listed in Table 1. Table 1(a) shows the results on single feature kernel matrices. It is interesting to observe from this group of results that KMTJSRC methods perform better than SVM on single features. The baseline NS method is also comparable with SVM on these kernels. The feature combination results are listed in Table 1(b), from which we can see that all feature combination methods dramatically improve the classification performance and our two methods are better than the MKL, CG-Boost and LPBoost methods presented in [7]. Figure 4 shows some example classifications based on the seven single features and their combination. To show the effectiveness of task weights learning scheme for classification, we simultaneously provide the results of our methods under the learnt task weights and under the equal task weights in the bottom two rows of Table 1(b). As expected that task weights learning scheme is helpful to achieve better recognition performance. In detail, for KMTJSRC-RKHS, the learnt task weights on the split#1 are: Colour (0.12), Shape (0.08), Texture (0.00), HSV (0.19), HOG (0.03), SIFTint (0.36) and SIFTbdy (0.22). The SIFTint is automatically assigned with the highest weight and this is consistent with the observation from Table 1(a) that SIFTint is the most “reliable” feature. The color related features (HSV and Colour) also get relatively high weights since they own strong complementarity to SIFT features.

Table 2 lists the accuracies of our methods along with the results from the data set authors [14] on the 102 category data set. Table 2(a) shows the results on single feature kernel matrices, from which we can see that KMTJSRC methods are competitive to SVM for single features on this data set. Table 1(b) lists the feature combination results, from which we can see that our methods perform comparably to the MKL method used in [14]. As a simplification of our methods, the independent SRC combination is also competitive to the MKL, but slightly inferior to our methods that take into account the joint sparsity across different tasks.

### 5.2.3 Caltech101 Data Set

We give in this subsection some results on the Caltech101 database [6] which is a challenging object classification data set containing images of 101 categories of objects as well as a background class. We follow the experimental protocol

---

2http://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html
3http://www.robots.ox.ac.uk/~vgg/data/flowers/102/index.html
stated by the designers of the data sets. We select 15 training images per category and test on 15 images per category. Evaluation includes all 102 classes averaged over three random training/test splits, and the performance is measured as the mean accuracy per class. In our experiment, we use four image features, Geometric Blur (GB) [2], Phow-gray ($L = 2$) / color ($L = 2$) [8] and SSIM ($L = 2$) [19], among which three are represented in spatial pyramid with level $L$. These features are extracted using the MKL code package from [21]. Table 3 lists the accuracies of our methods along with the results from [7][21]. Once again we observe that our methods are quite competitive to the MKL and LPBoost methods for visual feature combination. The task weights in our methods are optimized on a validation set formed by the 10-training / 5-validation splits of the 15 training samples per class. The SRC combination method, which as aforementioned is a simplification of our method, also performs comparably to the state-of-the-art results.

6. Conclusion and Discussion

In this paper we developed the MTJSRC algorithm along with its kernel extensions for visual feature combination. We found that the multi-task joint sparse representation is an effective and efficient way to fuse the complementary features for improving the overall classification performance. Experiments on challenging multi-class object recognition data sets show that our method is quite competitive to the state-of-the-art results achieved by multiple kernel learning methods. It is interesting to note that when $K = 1$, our KMTJSRC-CG and KMTJSRC-RKHS methods reduce to two kernel-views of SRC, which both perform better than or comparably to SVM. Similar to SRC, one appealing aspect of our method lies in that it is generally training free (except for the boosting of task weights on validation set). In summary, we can conclude with observations that kernelized multi-task joint sparse representation is a simple yet effective method for feature combination when features are represented in the form of kernel matrices. One future research on this topic is to systematically evaluate the performance of kernelized linear representation and recognition method on more computer vision and machine learning problems with comparison to SVM.

Acknowledgment

This research was supported in part by the IDMPO-MDA project number NRF20071DM-1DM002-069 on Live Spaces - Place Oriented Embodied Media (POEM), and in part by the AcRF Tier-1 Grant of R-263-000-464-112, Singapore.

References


Table 1. The accuracy (mean ± std %) performance on the 17 category Oxford Flowers data set. The results in bracket are obtained under equal task weights.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Colour</td>
<td>61.7 ± 3.3</td>
<td>60.9 ± 2.1</td>
<td>64.0 ± 2.1</td>
<td>64.0 ± 3.3</td>
</tr>
<tr>
<td>Shape</td>
<td>69.9 ± 3.2</td>
<td>70.2 ± 1.3</td>
<td>72.7 ± 0.3</td>
<td>71.5 ± 0.8</td>
</tr>
<tr>
<td>Texture</td>
<td>55.8 ± 1.4</td>
<td>63.7 ± 2.7</td>
<td>67.6 ± 2.4</td>
<td>67.6 ± 2.2</td>
</tr>
<tr>
<td>HSV</td>
<td>61.3 ± 0.7</td>
<td>62.9 ± 2.3</td>
<td>64.7 ± 4.1</td>
<td>65.0 ± 3.9</td>
</tr>
<tr>
<td>HOG</td>
<td>57.4 ± 3.0</td>
<td>58.5 ± 4.5</td>
<td>61.9 ± 3.6</td>
<td>62.6 ± 2.7</td>
</tr>
<tr>
<td>SIFTint</td>
<td>70.7 ± 0.7</td>
<td>70.6 ± 1.6</td>
<td>74.0 ± 2.2</td>
<td>74.0 ± 2.0</td>
</tr>
<tr>
<td>SIFTbdy</td>
<td>61.9 ± 4.2</td>
<td>59.4 ± 3.3</td>
<td>62.4 ± 3.2</td>
<td>63.2 ± 3.3</td>
</tr>
</tbody>
</table>

Table 2. The accuracy (%) performance on the 102 category Oxford Flowers data set. The results in bracket are obtained under equal task weights.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HSV</td>
<td>39.8 ± 2.0</td>
<td>43.0 ± 4.6</td>
<td>43.6 ± 4.1</td>
<td>42.5 ± 2.6</td>
</tr>
<tr>
<td>HOG</td>
<td>34.9 ± 2.0</td>
<td>49.6 ± 4.6</td>
<td>46.7 ± 4.1</td>
<td>48.1 ± 2.6</td>
</tr>
<tr>
<td>SIFTint</td>
<td>46.6 ± 2.0</td>
<td>55.1 ± 4.6</td>
<td>54.7 ± 4.1</td>
<td>55.2 ± 2.6</td>
</tr>
<tr>
<td>SIFTbdy</td>
<td>34.1 ± 2.0</td>
<td>32.0 ± 4.6</td>
<td>33.0 ± 4.1</td>
<td>31.6 ± 2.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>NS Combination</td>
<td>85.2 ± 2.1</td>
</tr>
<tr>
<td>SRC Combination</td>
<td>85.9 ± 2.2</td>
</tr>
<tr>
<td>MKL [7]</td>
<td>85.2 ± 1.5</td>
</tr>
<tr>
<td>CG-Boost [7]</td>
<td>84.8 ± 2.2</td>
</tr>
<tr>
<td>LPBoost [7]</td>
<td>85.4 ± 2.4</td>
</tr>
<tr>
<td>KMTJSRC-RKHS</td>
<td>88.1 ± 2.3 (86.8 ± 1.8)</td>
</tr>
<tr>
<td>KMTJSRC-CG</td>
<td>88.9 ± 2.9 (88.2 ± 2.3)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>NS Combination</td>
<td>59.2 ± 2.1</td>
</tr>
<tr>
<td>SRC Combination</td>
<td>70.0 ± 2.1</td>
</tr>
<tr>
<td>MKL [14]</td>
<td>72.8 ± 2.1</td>
</tr>
<tr>
<td>KMTJSRC-RKHS</td>
<td>73.8 (71.5)</td>
</tr>
<tr>
<td>KMTJSRC-CG</td>
<td>74.1 (71.2)</td>
</tr>
</tbody>
</table>
Table 3. The accuracy (mean ± std%) performance on the Caltech101 data set (15 training/15 test). The results in bracket are obtained under equal task weights.

(a) Single features

<table>
<thead>
<tr>
<th>Features</th>
<th>NS</th>
<th>MKL [21]</th>
<th>KMTJSRC-RKHS</th>
<th>KMTJSRC-CG</th>
</tr>
</thead>
<tbody>
<tr>
<td>GB</td>
<td>40.8 ± 0.6</td>
<td>62.6 ± 1.2</td>
<td>58.3 ± 0.4</td>
<td>58.5 ± 0.3</td>
</tr>
<tr>
<td>PHOW-g</td>
<td>45.4 ± 0.9</td>
<td>63.9 ± 0.8</td>
<td>65.0 ± 0.7</td>
<td>64.5 ± 0.5</td>
</tr>
<tr>
<td>PHOW-c</td>
<td>37.3 ± 0.5</td>
<td>54.5 ± 0.6</td>
<td>56.1 ± 0.5</td>
<td>54.4 ± 0.7</td>
</tr>
<tr>
<td>SSIM</td>
<td>39.8 ± 0.8</td>
<td>54.3 ± 0.6</td>
<td>61.8 ± 0.6</td>
<td>59.7 ± 0.4</td>
</tr>
</tbody>
</table>

(b) Feature combination methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>NS Combination</td>
<td>51.7 ± 0.8</td>
</tr>
<tr>
<td>SRC Combination</td>
<td>69.2 ± 0.7</td>
</tr>
<tr>
<td>MKL [21]</td>
<td>70.0 ± 1.0</td>
</tr>
<tr>
<td>LPBoost [7]</td>
<td>70.7 ± 0.4</td>
</tr>
<tr>
<td>KMTJSRC-RKHS</td>
<td>71.0 ± 0.3 (69.5 ± 0.6)</td>
</tr>
<tr>
<td>KMTJSRC-CG</td>
<td>71.4 ± 0.4 (70.2 ± 0.7)</td>
</tr>
</tbody>
</table>

Figure 4. Example classifications from 17 category Oxford Flowers. (a) Classification results for single features and their combination. (b) Some detailed results on sparse representation coefficients. (a) Classification results for single features and their combination with simple constraints: a limited-memory projected quasi-newton algorithm. (b) Feature combination methods with simple constraints: a limited-memory projected quasi-newton algorithm.