International Journal of Innovative Computing, Information and Control
Volume **14**, Number **4**, August **2018**

ICIC International ©2018 ISSN 1349-4198
pp. 1267–1283

LEARNING REGULARIZED MULTI-VIEW STRUCTURED SPARSE REPRESENTATION FOR IMAGE ANNOTATION

Zhiqiang Xing, Miao Zang* and Yongmei Zhang

School of Electronics and Information Engineering
North China University of Technology
No. 5, Jinyuanzhang Road, Shijingshan District, Beijing 100144, P. R. China
{ speech; zhangym }@ncut.edu.cn; *Corresponding author: zangm@ncut.edu.cn

Received December 2017; revised March 2018

Abstract. *Automatic image annotation is an important problem in computer vision owing to its critical role in image retrieval. In order to exploit the diversities of different features in a sample as well as the similarities, we present a regularized multi-view structured sparse representation model for image annotation. In this model, handcrafted visual features, deep learning based features and label information are considered as different views. Each view is coded on its associated dictionary to allow flexibility of coding coefficients from different views, while the disagreement between each view and a soft-consensus regularization term is minimized to keep the similarity among multiple views. The weight for each view is learned in the coding stage, and a weighted label prediction and propagation method is also proposed. Experimental results on ESP Game and IAPR TC-12 datasets demonstrate the effectiveness of the proposed approach compared with other related approaches for image-annotation task.*

Keywords: Regularization term, Image annotation, Multi-view learning, Structured sparsity, Deep learning

1. Introduction.

Automatic image annotation aims at automatically assigning relevant text labels to a given image reflecting its semantic content. It has become an active research topic due to the great potentials in image retrieval field. The difficulty of this task lies in that a balance has to be kept between two conflicting goals: firstly, the selected image representation needs to be specific to be able to differentiate objects that are easily confounded. On the other hand, the image representation is required to be invariant to occlusions, deformation, and scale and view point variations, etc. This makes the automatic image annotation be an extremely challenging problem.

Existing image annotation algorithms can be roughly divided into four types of models: discriminative model, generative model, nearest neighbor model and deep learning based model. Discriminative models [1-3] learn a separate classifier for each label and use these classifiers to predict label classification for the test image. It does not take the correlation between different labels into account, which is often very important for image annotation. Generative models [4-6] attempt to predict the correlations or joint probabilities between semantic labels and visual features from unlabeled images. Many parameters' estimation is required in this type of model, which leads to heavy computation cost. Nearest neighbor (NN) models [7-11] solve image annotation problem as a retrieval problem, and attempt to represent the image to be annotated by its nearest neighbors. Because of its simplicity and efficiency, NN models attract more researching attention. Deep learning based models [12-15] learn the image features based on a deep neutral network structure such as

DOI: 10.24507/ijicic.14.04.1267

convolution neural network (CNN) and auto-encoder (AE), and output the score of each label by a mapping function, such as softmax or sigmoid function. It is a new branch in image annotation and attracts many research interests due to the ground-breaking results of deep learning on image classification [16,17]. However, the annotation performance of deep learning models lags behind the state of the art due to the lack of supervised learning. Some works [11,12] employ the deep learning based features directly into K-nearest neighbor (KNN) based annotation models and obtain promising results.

In recent decades, sparse coding has been widely used in computer vision and demonstrates good performance [18-20]. It is close to the NN-based method since it also represents the image to be annotated as the linear combination of training samples while forcing the representation coefficients being sparse. Many researchers have extended and improved sparse coding to solve image annotation problem [21-30]. Wang et al. [21] presented a multi-label sparse coding framework for feature extraction and image annotation. Gao et al. [22] proposed a tri-layer group sparse coding framework for concurrent single-label image classification and annotation. Cao et al. [23] utilized the group sparse reconstruction framework to learn the label correlation for dictionary and reconstructed the test image for label prediction under weakly supervised case. Lu et al. [24] presented a more descriptive and robust visual bag-of-word (BOW) representation by semantic sparse recoding method for image annotation and classification. Jing et al. [25] learned a label embedded dictionary as well as a sparse representation for image annotation. However, the above methods utilize single feature or concatenate multiple features to a long vector to represent images, which cannot efficiently explore the complementary of different features carrying different physical characteristics. Moran and Lavrenko [26] introduced a sparse kernel learning framework for the continuous relevance model, which adaptively selected a different kernel for a different feature, and combined multiple features by a greedy approach to get the performance maximization. However, the number of kernels they employed is limited, which may be not enough to discover the potential complementary among diverse features and that between visual features and labels. Liu et al. [27] introduced structured sparsity with multi-view learning and treated label information along with different features as different views of an image, which achieved good performance for image annotation. In our previous work [28], a mixed-norm sparsity was introduced into the multi-view learning framework to obtain a better image representation for annotation. However, both these methods assumed that the coding coefficients for different views are the same, and the weights for different views are equal, which is often not true in nature. To solve this problem, [29] presented a multi-view joint sparse coding model, in which each feature/label view corresponds to a specific sparse representation, and a joint sparse regularization term is introduced to ensure the similar sparse pattern across multiple views. [30] revised this framework by mapping each view to an implicit kernel space to find a set of optimal sparse representations and discriminative dictionaries jointly, which obtained a better performance. However, the joint sparse regularizer only enforced the similar sparsity of different views in the row direction, which is still limited in employing the similarity and diversity of multiple views, and the kernel mapping increases calculation cost greatly. Yang et al. [19] introduced a variance regularization term with adaptive weighting for different views to effectively exploit the similarity and diversity of different features for coding and classification.

Inspired by earlier work [19,27], we propose to learn a regularized multi-view structured sparse representation for image annotation. We account for both the similarity and distinctiveness of different views in coding and label prediction stages. In the coding stage, we introduce distinct coding coefficients for different views into the structured sparsity representation framework, which is also integrated with a weighted soft-consensus regularization term to explore both similarities and differences in all views. We treat the handcrafted features as well as the deep learning based features and label information as multiple separate views of images for multi-view learning to strengthen the discriminative power of the learned sparse representation. In the label transfer stage, we first predict the sparse coefficient vector for test image in label view by weighted average of the learned sparse coding in other views; then the product of learned dictionary in label view from training images and the predicted sparse coefficient vector is used directly to propagate the labels from the training images to the test image. Our extensive experiments on ESP Game and IAPR TC12 datasets demonstrate the effectiveness of our proposed method and the competitive performance compared with other related methods.

The contributions of this work can be summarized as follows. Firstly, we propose a soft-consensus regularized multi-view structured sparse representation (RmSSR) framework and successfully apply it into image-annotation task. Secondly, we present the optimization algorithm of the proposed method based on the accelerated proximal gradient (APG) method, and a weighted greedy label prediction scheme is also proposed. Finally, we incorporate deep learning based feature into multi-view learning besides the handcrafted visual features and label information, and provide the experimental comparison and analysis.

The rest of this paper is organized as follows. Section 2 briefly reviews some related works. Section 3 describes the details of our regularized multi-view structured sparse representation model, optimization method and the label prediction scheme. Section 4 demonstrates experimental results followed by the conclusion in Section 5.

2. Related Works.

This section briefly presents a review on existing image annotation methods using sparse coding.

2.1. Annotation based on sparse coding.

The sparse representation based image annotation methods were presented in [21,22] for multi-label image annotation. Denote by $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in \mathbf{R}^{P\times N}$ the feature vector matrix formed by original N training samples, where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{iP}]^T$, $(i = 1, 2, \ldots, N)$, is the feature vector of the ith sample image, and P is the feature dimension. Letting $\boldsymbol{x}_t \in \mathbf{R}^P$ be a test sample to be labeled, the sparse coding vector $\boldsymbol{\omega}$ of the test image over all training images is obtained by solving the following optimization problem:

$$\arg\min_{\boldsymbol{\omega}} \frac{1}{2} \|\boldsymbol{x}_t - \boldsymbol{X}\boldsymbol{\omega}\|_2^2 + \lambda\phi(\boldsymbol{\omega}) \tag{1}$$

where $\boldsymbol{\omega} = [\omega_1, \omega_2, \ldots, \omega_N] \in \mathbf{R}^N$ and ω_i is the coefficient associated with the ith training sample. $\phi(\boldsymbol{\omega})$ is a regularizer over $\boldsymbol{\omega}$ to encourage sparsity. λ is a scalar constant used to set the relative influence of both terms.

For different purposes, different forms of $\phi(\boldsymbol{\omega})$ can be used. Typically, L_1 norm is used, which yields:

$$\arg\min_{\boldsymbol{\omega}} \frac{1}{2} \|\boldsymbol{x}_t - \boldsymbol{X}\boldsymbol{\omega}\|_2^2 + \lambda \|\boldsymbol{\omega}\|_1 \tag{2}$$

Many annotation methods [9,22,23] introduce the group sparsity to enhance the annotation performance by exploring the structure information in the observed data, which can be generally formulated as structured sparse coding [31]:

$$\arg\min_{\boldsymbol{\omega}} \frac{1}{2} \|\boldsymbol{x}_t - \boldsymbol{X}\boldsymbol{\omega}\|_2^2 + \lambda \|\boldsymbol{\omega}\|_{1,p} \tag{3}$$

Typically, these methods rely on the notion of groups among the training examples to encourage members of the same group to rely on the same training examples. $\|\boldsymbol{\omega}\|_{1,p}$ means L_p norm is used on the sparse codes within each group, while L_1 norm is used to sum the results between groups. Generally $p = 2$ or ∞.

For label transfer, [21] propagates the labels from the training images to the test image directly by the product of label matrix of training images with the learned sparse vector. [22] predicts the test image labels based on the reconstruction error in the (sub)groups, and assigns labels from the (sub)groups with the minimum reconstruction error to the test image.

2.2. **Annotation based on dictionary learning.** Although sparse coding has been proven effective in many domains, all the training samples (called dictionary) used in it may introduce the noisy information and increase the coding complexity, and could not fully exploit the discriminative information hidden in the training samples. [23,25] introduce dictionary learning for image annotation aiming at learning the space where the given signal could be well represented for processing from the training samples. They follow the conventional dictionary learning framework [32]:

$$\begin{aligned}&\arg\min_{\boldsymbol{D},\boldsymbol{W}} \frac{1}{2N}\|\boldsymbol{X}-\boldsymbol{D}\boldsymbol{W}\|_F^2+\lambda\phi(\boldsymbol{W})\\ &\text{s.t.}\quad \|\boldsymbol{D}_i\|_2\le 1,\ 1\le i\le N_d\end{aligned} \tag{4}$$

where $\boldsymbol{D} \in \mathbf{R}^{P\times N_d}$ is the dictionary, $1 \le N_d \le N$, and $\boldsymbol{W} \in \mathbf{R}^{N_d\times N}$ is the sparse coefficient matrix. The label transfer scheme is similar to the scheme above except that it utilizes the learned dictionary to replace all the training samples.

2.3. **Annotation based on multi-view structured sparse coding.** While the dictionary learning framework is successful in many tasks, it has only been applied to the single-view case. With multiple types of input modalities, [27] proposes a multi-view learning model with structured sparsity to factorize multiple representations. Suppose each sample is represented by V different features of views. Denote by $\boldsymbol{X}^{(v)} = \left[\boldsymbol{X}_1^{(v)}, \boldsymbol{X}_2^{(v)}, \ldots, \boldsymbol{X}_i^{(v)}, \ldots, \boldsymbol{X}_N^{(v)}\right] \in \mathbf{R}^{P_v\times N}$ the feature vector matrix of the vth view for the training images, by $\boldsymbol{D}^{(v)} = \left[\boldsymbol{D}_1^{(v)}, \boldsymbol{D}_2^{(v)}, \ldots, \boldsymbol{D}_{N_d}^{(v)}\right] \in \mathbf{R}^{P_v\times N_d}$ $(N_d > P_v)$ the overcomplete dictionary of the vth view and by $\boldsymbol{W} \in \mathbf{R}^{N_d\times N}$ the coding coefficient matrix. Thus, the method of multi-view learning with structured sparsity is formulated as:

$$\begin{aligned}&\arg\min_{\boldsymbol{D}^{(v)},\boldsymbol{W}} \frac{1}{2N}\sum_{v=1}^{V}\left\|\boldsymbol{X}^{(v)}-\boldsymbol{D}^{(v)}\boldsymbol{W}\right\|_F^2+\lambda_1\|\boldsymbol{W}\|_{1,\infty}+\lambda_2\sum_{v=1}^{V}\left\|\left(\boldsymbol{D}^{(v)}\right)^T\right\|_{1,\infty}\\ &\text{s.t.}\quad \left\|\boldsymbol{D}_i^{(v)}\right\|_2\le 1,\ 1\le i\le N_d\end{aligned} \tag{5}$$

where $\|\boldsymbol{W}\|_{1,\infty}$ is a regularizer that controls the sparsity over $\boldsymbol{W}$. $\sum_{v=1}^{V}\left\|\left(\boldsymbol{D}^{(v)}\right)^T\right\|_{1,\infty}$ is a regularizer that controls the structure of dictionary, and λ_1 and λ_2 are parameters that balance the loss function and regularizations respectively. Here the structured sparsity represented by mixed norm takes each row vector of the matrix as a group. This formulation enforces structured sparsity on the dictionary entries as well as on the sparse coding coefficients, and expects to find a $\boldsymbol{W}$ that naturally extracts the information shared among different views from the information specific to each view.

In [27], labels are treated as an additional $(V+1)$th view, then $\boldsymbol{X}^{(V+1)} = \left[\boldsymbol{X}_1^{(V+1)}, \boldsymbol{X}_2^{(V+1)}, \ldots, \boldsymbol{X}_N^{(V+1)}\right] \in \mathbf{R}^{P_{V+1}\times N}$ represents the label view matrix of training samples,

P_{V+1} is the number of labels, $\boldsymbol{X}_i^{(V+1)} \in \mathbf{R}^{P_{V+1}}$ is the label vector of the ith image, and $\boldsymbol{D}^{(V+1)}$ is the associated label view dictionary which also can be learned by Equation (5).

For a new test sample $\boldsymbol{x}_t = \left\{\boldsymbol{x}_t^{(1)}, \boldsymbol{x}_t^{(2)}, \ldots, \boldsymbol{x}_t^{(V)}\right\}$, the corresponding sparse code matrix $\boldsymbol{\omega}$ can be obtained by solving the convex problem:

$$\arg\min_{\boldsymbol{\omega}} \frac{1}{2} \sum_{v=1}^{V} \left\| \boldsymbol{x}_t^{(v)} - \boldsymbol{D}^{(v)} \boldsymbol{\omega} \right\|_2^2 + \lambda_1 \left\| \boldsymbol{\omega} \right\|_1 \tag{6}$$

and the label view of the test sample $\boldsymbol{x}_t^{(V+1)}$ are then predicted by $\boldsymbol{x}_t^{(V+1)} = \boldsymbol{D}^{(V+1)} \boldsymbol{\omega}$ directly.

2.4. **Discussion.** Our approach is related to the work in [27], but differs in two aspects. First, [27] uses multi-view structured sparsity for semi-supervised image annotation, i.e., they assume part of testing data is available. Our work focuses on the supervised occasion since it is often the case that testing data are not known. Second, [27] uses the common coefficients matrix for different views in learning and label transfer stages, which omits the diversity between different views. In our formulation, we allow the coding coefficients to be flexible to some extent for different views, while considering their similarity. Thus, the learned coefficients and dictionary specific to label view can be effectively used to obtain labels of test image.

Our proposed use of multi-view learning with soft-consensus regularization term is also related to the work in [10], but rather than using a set of predefined weights for different views in the regularization term, we learn the weight for each view to achieve better representation.

3. **Proposed Method.**

In this section, we will describe the formulation of our method, its optimization and the labels prediction and propagation scheme.

3.1. **Multi-view regularized sparse representation model.** We propose a multi-view regularized structure sparse representation model for image annotation. Assume that we have a multi-view dataset of N training samples from V visual feature views and a label view, i.e., $\left\{\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \ldots, \boldsymbol{X}^{(V)}, \boldsymbol{X}^{(V+1)}\right\}$, where $\boldsymbol{X}^{(v)} \in \mathbf{R}^{P_v \times N}$. $\boldsymbol{X}^{(V+1)}$ is the label view matrix, in which each entry is either 1 or 0 representing whether the occurrence of a certain label is in the image or not. Our method aims to find the sparse coefficient matrix of each view $\boldsymbol{\alpha}^{(v)} \in \mathbf{R}^{N_d \times N}$ for the sample data $\boldsymbol{X}^{(v)}$ over associated dictionary $\boldsymbol{D}^{(v)} \in \mathbf{R}^{P_v \times N_d}$, where N_d is the number of dictionary atoms.

Here we exploit the similarity as well as the distinctiveness of different views and introduce a regularization term $\left\|\boldsymbol{\alpha}^{(v)} - \widetilde{\boldsymbol{\alpha}}\right\|_F$ as a measure of disagreement between coefficient matrix $\boldsymbol{\alpha}^{(v)}$ and the consensus matrix $\widetilde{\boldsymbol{\alpha}}$, which is used to regularize the coding coefficients of different views over their associated dictionaries.

Thus, the objective function of our RmSSR is written as followed:

$$\begin{aligned} \arg\min_{\boldsymbol{D}^{(v)}, \omega^{(v)}, \boldsymbol{\alpha}^{(v)}} \sum_{v=1}^{V+1} \Bigg(& \left\| \boldsymbol{X}^{(v)} - \boldsymbol{D}^{(v)} \boldsymbol{\alpha}^{(v)} \right\|_F^2 + \lambda_1 \left\| \boldsymbol{\alpha}^{(v)} \right\|_{1,\infty} + \lambda_2 \left\| \left(\boldsymbol{D}^{(v)} \right)^T \right\|_{1,\infty} \\ & + \lambda_3 \omega^{(v)} \left\| \boldsymbol{\alpha}^{(v)} - \widetilde{\boldsymbol{\alpha}} \right\|_F^2 \Bigg) \\ \text{s.t.} \quad & - \sum_{v=1}^{V+1} \omega^{(v)} \ln \omega^{(v)} > \sigma \end{aligned} \tag{7}$$

where $\sum_{v=1}^{V+1}\left\|\boldsymbol{\alpha}^{(v)}\right\|_{1,\infty}$ is a regularizer that controls the sparsity over code coefficients of different views. $\sum_{v=1}^{V+1}\left\|\left(\boldsymbol{D}^{(v)}\right)^T\right\|_{1,\infty}$ is still a regularizer that controls the structure of dictionary. $\widetilde{\boldsymbol{\alpha}}$ is the consensus matrix, and $\omega^{(v)}\left\|\boldsymbol{\alpha}^{(v)}-\widetilde{\boldsymbol{\alpha}}\right\|_F^2$ is the soft-consensus regularizer introduced to enforce coefficient matrix corresponding to each view to be similar to a consensus matrix among all views. This also results in the dictionary to capture similar contents in their respective views. $\omega^{(v)}$ is a factor used to tune the relative weight among different views. The weights are constrained to the maximum entropy principle [19], and σ is a limit value of entropy, which makes the distribution of $\omega^{(v)}$ not concentrated in some individual views. Each weight $\omega^{(v)}$ is normalized in $[0,1]$. λ_1, λ_2 and λ_3 are positive constants that balance the loss function and the regularization terms, respectively.

3.2. **Objective optimization.** The objective function in Equation (7) can be solved by alternating optimization algorithm [33]. That is, we optimize with respect to $\omega^{(v)}$, $\boldsymbol{\alpha}^{(v)}$ and $\boldsymbol{D}^{(v)}$ respectively, keeping the other two fixed. In the following, we first provide a general description of our alternating optimization for RmSSR, and then present the optimization process for the subproblems in detail.

Firstly, by fixing $\boldsymbol{D}^{(v)}$ and $\omega^{(v)}$, Equation (7) can be simplified to:

$$\arg\min_{\boldsymbol{\alpha}^{(v)}}\sum_{v=1}^{V+1}\left(\left\|\boldsymbol{X}^{(v)}-\boldsymbol{D}^{(v)}\boldsymbol{\alpha}^{(v)}\right\|_F^2+\lambda_1\left\|\boldsymbol{\alpha}^{(v)}\right\|_{1,\infty}+\lambda_3\omega^{(v)}\left\|\boldsymbol{\alpha}^{(v)}-\widetilde{\boldsymbol{\alpha}}\right\|_F^2\right) \tag{8}$$

and $\boldsymbol{\alpha}^{(v)}$ can be optimized iterately by Equation (8).

Once $\boldsymbol{\alpha}^{(v)}$ has been updated for each view in a particular iteration, we can take derivative of Equation (8) w.r.t $\widetilde{\boldsymbol{\alpha}}$, and obtain the solution of $\widetilde{\boldsymbol{\alpha}}$ by setting the derivative to be 0:

$$\widetilde{\boldsymbol{\alpha}}=\sum_{v=1}^{V+1}\omega^{(v)}\boldsymbol{\alpha}^{(v)}\Bigg/\sum_{v=1}^{V+1}\omega^{(v)} \tag{9}$$

Next, by fixing $\boldsymbol{\alpha}^{(v)}$ and $\omega^{(v)}$, Equation (7) can be simplified to update dictionary $\boldsymbol{D}^{(v)}$ as follows:

$$\arg\min_{\boldsymbol{D}^{(v)}}\sum_{v=1}^{V+1}\left(\left\|\boldsymbol{X}^{(v)}-\boldsymbol{D}^{(v)}\boldsymbol{\alpha}^{(v)}\right\|_F^2+\lambda_2\left\|\left(\boldsymbol{D}^{(v)}\right)^T\right\|_{1,\infty}\right) \tag{10}$$

Finally, by fixing $\boldsymbol{D}^{(v)}$ and $\boldsymbol{\alpha}^{(v)}$, Equation (7) can be simplified to:

$$\arg\min_{\omega^{(v)}}\sum_{v=1}^{V+1}\lambda_3\omega^{(v)}\left\|\boldsymbol{\alpha}^{(v)}-\widetilde{\boldsymbol{\alpha}}\right\|_F^2+\xi\omega^{(v)}\ln\omega^{(v)} \tag{11}$$

here $\xi>0$ is the Lagrange multiplier, and the weights could be directly updated by setting the deviation equal to 0 as:

$$\omega^{(v)}=\exp\left\{-1-\lambda_3\left\|\boldsymbol{\alpha}^{(v)}-\widetilde{\boldsymbol{\alpha}}\right\|_F^2\Big/\xi\right\} \tag{12}$$

The overall procedure of the alternating optimization for our method is summarized in Algorithm 1.

Next, we optimize subproblem (8) and (10) to learn the sparse coding and dictionary respectively.

Learning sparse coding: Equation (8) can be separated to $V+1$ parts w.r.t each view, and each part is written as the following general form. For convenience, we omit

Algorithm 1 Alternating optimization algorithm for Equation (7)

```
Input: Feature view of training samples $\boldsymbol{X}^{(v)}$, $v = 1, 2, \dots, V$; Label view of training samples $\boldsymbol{X}^{(V+1)}$; Parameters $\lambda_1$, $\lambda_2$, $\lambda_3$, $\xi$
Output: Dictionary $\boldsymbol{D}^{(v)}$, sparse coefficients matrix $\boldsymbol{\alpha}^{(v)}$, and weight $\omega^{(v)}$, $v = 1, 2, \dots, V+1$
1:  Initialize $\boldsymbol{D}^{(v)}$, $\boldsymbol{\alpha}^{(v)}$ with random entries, initialize the weight $\omega^{(v)}$ with $\frac{1}{V+1}$
2:  While not convergence do
3:      Update coding coefficient $\boldsymbol{\alpha}^{(v)}$ via Equation (8) and get $\widetilde{\boldsymbol{\alpha}}$ via Equation (9)
4:      Update dictionary $\boldsymbol{D}^{(v)}$ via Equation (10)
5:      Update weights $\omega^{(v)}$ via Equation (12)
6:  End While
```

the superscript of the view index.

$$\arg\min_{\boldsymbol{\alpha}} \left(\|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{\alpha}\|_F^2 + \lambda_1 \|\boldsymbol{\alpha}\|_{1,\infty} + \lambda_3 \omega \|\boldsymbol{\alpha} - \widetilde{\boldsymbol{\alpha}}\|_F^2 \right) \tag{13}$$

By denoting $f(\boldsymbol{\alpha}) = \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{\alpha}\|_F^2 + \lambda_3 \omega \|\boldsymbol{\alpha} - \widetilde{\boldsymbol{\alpha}}\|_F^2$, Equation (13) is rewritten as:

$$\arg\min_{\boldsymbol{\alpha}} \left(f(\boldsymbol{\alpha}) + \lambda_1 \|\boldsymbol{\alpha}\|_{1,\infty} \right) \tag{14}$$

Equation (14) is a non-smooth convex function, in which $f(\boldsymbol{\alpha})$ is differentiable and its differential is Lipschitz continuous, while $\|\boldsymbol{\alpha}\|_{1,\infty}$ is non-differentiable. We employ the APG [34,35] method to solve this problem since it is an accelerated gradient descent algorithm with the convergence rate of $O(1/k^2)$ (k is the number of iterations) and the generalized gradient update in each iteration is solved analytically by a simple sorting procedure. Moreover, APG method only need first order information, which makes it suitable for large scale learning problems.

The gradient of $f(\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$ can be written as:

$$\nabla f_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) = 2 \left[-\boldsymbol{D}^T \boldsymbol{X} + \boldsymbol{D}^T \boldsymbol{D} \boldsymbol{\alpha} + \lambda_3 \omega \left(\boldsymbol{\alpha} - \widetilde{\boldsymbol{\alpha}} \right) \right] \tag{15}$$

Then the optimization procedure of APG algorithm consists of alternately updating the coefficient matrix $\boldsymbol{\alpha}_m$ and an aggregation matrix $\boldsymbol{Q}_m$, where m indexes an iteration. For the first step, given the current matrix $\boldsymbol{Q}_m$, we update $\boldsymbol{\alpha}_{m+1}$ by the following formula:

$$\boldsymbol{Z} = \boldsymbol{Q}_m - \frac{1}{L} \nabla f_Q(\boldsymbol{Q}_m) \tag{16}$$

$$\boldsymbol{\alpha}_{m+1} = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2} \|\boldsymbol{\alpha} - \boldsymbol{Z}\|_F^2 + \frac{\lambda_1}{L} \|\boldsymbol{\alpha}\|_{1,\infty} \tag{17}$$

where L is a parameter controlling the step penalty. Equation (17) can be decomposed into N_d separate subproblems of dimension N, and each is solved with the primal dual relationship according to [34] (the detailed steps are listed in Step 4 of Algorithm 2).

For the second step, we update the aggregation matrix $\boldsymbol{Q}_{m+1}$ as a linear combination of $\boldsymbol{\alpha}_{m+1}$ and $\boldsymbol{\alpha}_m$ as follows:

$$\boldsymbol{Q}_{m+1} = \boldsymbol{\alpha}_{m+1} + \frac{\theta_{m+1}(1 - \theta_m)}{\theta_m} (\boldsymbol{\alpha}_{m+1} - \boldsymbol{\alpha}_m) \tag{18}$$

Here we set $\theta_{m+1} = \frac{2}{m+3}$ by convention. Algorithm 2 summarizes the optimization procedure of subproblem (8) by APG method, where for a matrix $\boldsymbol{M}$, $\boldsymbol{M}_{i,\cdot}$ represents the ith row vector of $\boldsymbol{M}$, and $\boldsymbol{M}_{ij}$ is the entry of the ith row and jth column of $\boldsymbol{M}$.

Learning dictionary: Subproblem (10) could be solved similar to subproblem (8), and the optimization procedures are summarized in Algorithm 3.

Algorithm 2: Optimizing algorithm for subproblem (8)

Input: Training image data $\boldsymbol{X}^{(v)}$, corresponding dictionary $\boldsymbol{D}^{(v)}$, the weight coefficient $\omega^{(v)}$, $v = 1, 2, \ldots, V+1$, consensus matrix $\widetilde{\boldsymbol{\alpha}}$, regularization parameter λ_1, λ_3 and parameter L.

Output: Sparse coefficient $\boldsymbol{\alpha}^{(v)}$, $v = 1, 2, \ldots, V+1$

For each view, denote $\boldsymbol{X} = \boldsymbol{X}^{(v)}$, $\boldsymbol{D} = \boldsymbol{D}^{(v)}$, $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{(v)}$, $\omega = \omega^{(v)}$ for convenience.

```
Initialization: Initialize $\boldsymbol{Q}_0$ and $\boldsymbol{\alpha}_0$ to be zero matrix. Set $\theta_0 = 1$ and $m = 0$.
Repeat {Main loop}
    Calculate $\boldsymbol{Z} = \boldsymbol{Q}_m - \frac{1}{L}\nabla f_Q(\boldsymbol{Q}_m)$ via Equation (15)
    Calculate $\boldsymbol{\alpha}_{m+1}$ by Equation (17) as followed:
    For $i$th row of $\boldsymbol{\alpha}_{m+1}$, $i = 1, \ldots, N_d$
        If $\|\boldsymbol{Z}_{i,\cdot}\|_1 \le \frac{\lambda_1}{L}$, set $(\boldsymbol{\alpha}_{m+1})_{i,\cdot} = 0$. Continue.
        Let $u_j = |\boldsymbol{Z}_{ij}|$, $j = 1, 2, \ldots, N$, sort vector $\boldsymbol{u}$ in the decreasing order: $u_1 \ge u_2 \ge \cdots \ge u_N$
        Find $q = \max\left\{q : \frac{\lambda_1}{L} - \sum_{r=1}^{q}(u_r - u_q) > 0\right\}$
        $(\boldsymbol{\alpha}_{m+1})_{ij} = sign(\boldsymbol{Z}_{ij}) \min\left(|\boldsymbol{Z}_{ij}|, \left(\sum_{r=1}^{q} u_r - \frac{\lambda_1}{L}\right)\Big/ q\right)$,
        $j = 1, 2, \ldots, N$
    End For
    $\theta_{m+1} = \frac{2}{m+3}$
    $\boldsymbol{Q}_{m+1} = \boldsymbol{\alpha}_{m+1} + \frac{(1-\theta_m)\theta_{m+1}}{\theta_m}(\boldsymbol{\alpha}_{m+1} - \boldsymbol{\alpha}_m)$
    $m = m + 1$
Until convergence is attained.
```

Algorithm 3: Optimizing algorithm for subproblem (10)

Input: Training image data $\boldsymbol{X}^{(v)}$, sparse coefficient $\boldsymbol{\alpha}^{(v)}$, the weight coefficient $\omega^{(v)}$, $v = 1, 2, \ldots, V+1$, consensus matrix $\widetilde{\boldsymbol{\alpha}}$, regularization parameter λ_2 and parameter L

Output: $\boldsymbol{D}^{(v)}$, $v = 1, 2, \ldots, V+1$

For each view, denote $\boldsymbol{X} = \boldsymbol{X}^{(v)}$, dictionary $\boldsymbol{B} = \left(\boldsymbol{D}^{(v)}\right)^T$, $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{(v)}$, $\omega = \omega^{(v)}$, $P = P_v$

```
Initialization: Initialize $\boldsymbol{Q}_0$ and $\boldsymbol{B}_0$ to be zero matrix. Set $\theta_0 = 1$ and $m = 0$.
Repeat {Main loop}
    Calculate $\boldsymbol{Z} = \boldsymbol{Q}_m - \frac{2}{L}\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^T\boldsymbol{Q}_m - \boldsymbol{\alpha}\boldsymbol{X}^T\right)$
    For $i$th row of $\boldsymbol{B}_{m+1}$, $i = 1, \ldots, N_d$
        If $\|\boldsymbol{Z}_{i,\cdot}\|_1 \le \frac{\lambda_2}{L}$, set $(\boldsymbol{B}_{m+1})_{i,\cdot} = 0$. Continue.
        Let $u_j = |\boldsymbol{Z}_{ij}|$, $j = 1, 2, \ldots, P$, sort vector $\boldsymbol{u}$ in the decreasing order: $u_1 \ge u_2 \ge \cdots \ge u_P$
        Find $q = \max\left\{q : \frac{\lambda_2}{L} - \sum_{r=1}^{q}(u_r - u_q) > 0\right\}$
        $(\boldsymbol{B}_{m+1})_{ij} = sign(\boldsymbol{Z}_{ij}) \min\left(|\boldsymbol{Z}_{ij}|, \left(\sum_{r=1}^{q} u_r - \frac{\lambda_2}{L}\right)\Big/ q\right)$,
        $j = 1, 2, \ldots, P$
    End for
    $\theta_{m+1} = \frac{2}{m+3}$
    $\boldsymbol{Q}_{m+1} = \boldsymbol{B}_{m+1} + \frac{(1-\theta_m)\theta_{m+1}}{\theta_m}(\boldsymbol{B}_{m+1} - \boldsymbol{B}_m)$
    $m = m + 1$
Until convergence is attained
$\boldsymbol{D}^{(v)} = (\boldsymbol{B})^T$
```

3.3. **Label prediction and propagation scheme.** Since we use different sparse representations for different views, we cannot use the learnt sparse coefficients from visual features to predict label directly like [27], but we still can infer the label information from these sparse coefficients. In particular, given a test image represented by multi-view features $\left\{\boldsymbol{x}_t^{(1)}, \boldsymbol{x}_t^{(2)}, \ldots, \boldsymbol{x}_t^{(V)}\right\}$, the learned dictionary $\boldsymbol{D}^{(v)}$ and weight $\omega^{(v)}$, $(v = 1, 2, \ldots, V+1)$, from the features and labels of the training images, we obtain the sparse coefficients $\boldsymbol{\alpha}_t^{(v)}$ of visual features for the test image in terms of learned dictionary and weight by solving the following convex problem:

$$\arg\min_{\boldsymbol{\alpha}_t^{(v)}} \sum_{v=1}^{V} \left(\left\| \boldsymbol{x}_t^{(v)} - \boldsymbol{D}^{(v)} \boldsymbol{\alpha}_t^{(v)} \right\|_2^2 + \lambda_1 \left\| \boldsymbol{\alpha}_t^{(v)} \right\|_1 + \lambda_3 \omega^{(v)} \left\| \boldsymbol{\alpha}_t^{(v)} - \widetilde{\boldsymbol{\alpha}}_t \right\|_2^2 \right) \tag{19}$$

where $\widetilde{\boldsymbol{\alpha}}_t$ is the mean vector of all the sparse coefficients from feature views for the test image.

Then, estimate the coefficient vector of the label view, i.e., $\hat{\boldsymbol{\alpha}}_t^{(V+1)}$ by weighted average of coefficient vectors from all feature views, which is similar to [10]:

$$\hat{\boldsymbol{\alpha}}_t^{(V+1)} = \sum_{v=1}^{V} \omega^{(v)} \boldsymbol{\alpha}_t^{(v)} \tag{20}$$

Finally, the scores for predicted labels of the test image can be obtained by

$$\hat{\boldsymbol{x}}_t^{(V+1)} = \boldsymbol{D}^{(V+1)} \hat{\boldsymbol{\alpha}}_t^{(V+1)} \tag{21}$$

in which the elements of label view can be considered as the score of each label. The desired number of labels can be obtained by sorting the labels according to their obtained scores. The label prediction and propagation scheme for our method is summarized in Algorithm 4.

Algorithm 4. Our label prediction and propagation scheme

```
Input: Learned dictionary $\boldsymbol{D}^{(v)}$ and weight $\omega^{(v)}$, $v = 1, 2, \ldots, V+1$; Feature vector
of a test image: $\boldsymbol{x}_t^{(v)}$, $v = 1, 2, \ldots, V$; Parameters $\lambda_1$, $\lambda_3$
Output: Predicted labels for the test image: $\hat{\boldsymbol{x}}_t^{(V+1)}$
1. Obtain sparse coefficient of feature view $\boldsymbol{\alpha}_t^{(v)}$, $(v = 1, 2, \ldots, V)$ via Equation (19)
2. Estimate sparse coefficient of label view $\boldsymbol{\alpha}_t^{(V+1)}$ via Equation (20)
3. Calculate the scores for predicted labels $\hat{\boldsymbol{x}}_t^{(V+1)}$ via Equation (21)
4. Transfer the required number of labels to the test image according to their scores.
```

4. **Experimental Results and Analysis.** In this section, we will experimentally evaluate the proposed RmSSR for image annotation. We divide the methods to be compared into three classifications. One is the spare coding based annotation frameworks including multi-label sparse coding (MSC) [21], multi-view Hessian discriminative sparse coding (mHDSC) [27], multi-view joint sparse coding (MvJSC) [29], and kernel based multi-view joint sparse coding (KMvJSC) [30]. Our RmSSR falls into this classification. To evaluate the effectiveness of regularization term, we also test our method without regularization term (mSSR). The second category is the KNN based annotation model since our sparse reconstruction method is close to it in that they both represent the test image as the linear combination of training samples. We compare joint equal contribution (JEC) [7], tag propagation (TagProp) [8] and two-pass KNN (2PKNN) [11] in this classification. The

third category lists some deep learning based methods including convolutional neural network regressor (CNN-R) [12], multi-view stacked auto-encoder (MVSAE) [13], canonical correlation analysis with K-nearest neighbor (CCA-KNN) [12], as well as JEC, 2PKNN and our RmSSR using deep learning based features.

4.1. **Experimental settings.**

The proposed method is experimentally evaluated using two popular and publicly available datasets benchmarks ESP Game and IAPR TC-12.

ESP Game dataset [36] consists of 20,770 images with a wide variety of topics, such as logos, drawings, and personal photos. Each image is manually annotated with up to 15 labels from a dictionary of 268 keywords, and with 4.7 labels on average. The set is split into 18,689 training images and 2,081 test images.

IAPR TC-12 dataset [37] consists of 19,627 images of natural scenes including sports, actions, people, animals, cities, landscapes and so on. Each image is manually annotated with up to 23 labels from a dictionary of 291 keywords, and with 4.7 labels on average. In the dataset, 17,665 images are selected for training, and the remaining 1,962 images for testing.

There are four parameters in our RmSSR method, which are λ_1, λ_2, λ_3 and ξ. Parameter ξ is set as 0.001 empirically and is fixed for all experiments. Parameters λ_1, λ_2 and λ_3 are tuned by 5-fold cross validation on the training image set and are selected in the range $\{1 \times 10^e \,|e = -5, -4, -3, -2, -1, 0, 1\}$. Concretely, λ_1 and λ_2 are both set as 0.01 on ESP Game dataset, and are separately set as 0.01 and 0.1 on IAPR TC-12 dataset. λ_3 is set as 0.1 and 0.01 on ESP Game and IAPR TC-12 datasets respectively. Due to the random entries in initialization, we repeat all the experiments 5 times separately and report the average result.

Following existing works [7,8,11,12,21], the image annotation performance is evaluated by comparing the results with the manually labeled ground truth. Each testing image is automatically annotated with five labels, and we calculate the precision (P) and recall (R) for each label, and get F1 measure by F1 = 2×R×P/(R+P). We report the mean value over all labels for each metric. Besides, the number of labels with non-zero recall (N+) is also used.

4.2. **Features.**

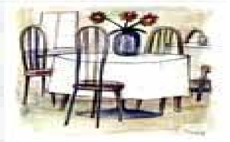
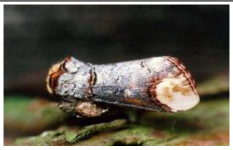
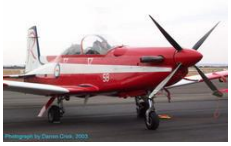
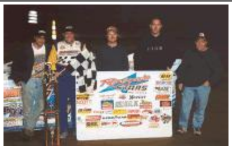
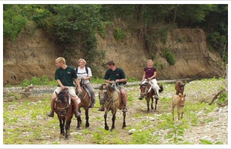
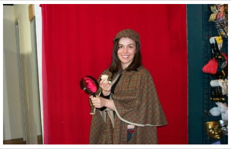
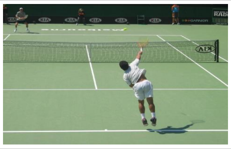
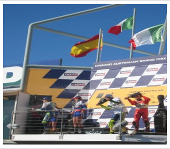
We consider two sets of features for our method evaluation. The first set of features denoted by T is the handcrafted features provided by [8], which consists of 15 features representing each image, including 3 color histograms features (RGB, LAB and HSV), 2 Hue and 2 SIFT features (extracted on dense grids and Harris-Laplacian interest points respectively, represented as DenseHue, HarrisHue, DenseSIFT, HarrisSIFT), 7 above histogram features with layout information (computed over a 3 × 1 horizontal decomposition of the image, represented as DenseSIFTV3H1, HarrisSIFTV3H1, DenseHueV3H1, HarrisHueV3H1, RGBV3H1, LABV3H1 and HSVV3H1) and a GIST feature.

The second set of feature denoted by VGG is the deep learning based feature extracted by employing the pre-trained CNN model on the ILSVRC-2012 dataset described in [17]. Following [12], we resize all the images to 224 × 224 irrespective of their aspect ratio and subtract the mean RGB value computed on the training samples from each image pixel. We use the VGG-16-D layered architecture, which outputs a 4096-dimensional feature vector.

4.3. **Image annotation results.**

Some examples of the predicted annotations produced on ESP Game and IAPR TC-12 datasets by our method are presented in Table 1. It contains at least one mismatched label compared with ground truth labels (perfectly matched annotations are not listed here). The differences in predicted labels are marked in italic font. The results in Table 1 demonstrate that, some predicted labels not contained in the

Table 1. Comparison of predicted labels with human annotations for images from ESP Game and IAPR TC-12 datasets. Italic labels are the differences between predicted and ground truth labels while some of them are also relevant to the image.

Images from ESP Game				
Ground truth labels	chair, flower, red, room, table	bug, green, insect, tree, wood	fly, plane, red, sky	black, car, flag, group, man, people
Predicted labels	chair, flower, *painting*, red, table	bug, green, insect, *leaf*, tree	*airplane*, fly, plane, red, sky	car, flag, man, people, *star*
Images from IAPR TC-12				
Ground truth labels	creek, face, forest, horse, people, rock	cap, curtain, front, glass, hand, shop, woman	court, man, player, tennis	flag, man, sky, wall, woman
Predicted labels	creek, face, forest, horse, people	cap, curtain, front, glass, hand	court, man, player, tennis, *net*	flag, man, *side*, sky, wall

ground truth label set can still describe the image well in many cases, such as "painting" in the first image, which shows the potential effectiveness of our proposed method for automatic image-annotation task. We also notice that some labels are inevitably missed in the predicted keywords when the ground truth provides more words than five since we are restricted to annotating each image with only five words.

4.4. **Method comparison.** Table 2 compares the performance of our proposed method to the related approaches on both datasets. Methods with suffix "_T" represent the implementations using T features, with "_VGG" using VGG feature, and with "_T+VGG" using both of them. MSC* and MHDSC* refer to our implementation using T features. The results of JEC and 2PKNN implemented using VGG feature are obtained from [12]. For other methods, the results of related work are directly copied from their original papers.

From Table 2, we can see that for sparse code based annotation frameworks, the three types of mSSR based methods (mHDSC, mSSR, RmSSR) and two types of joint sparsity based methods are clearly better than MSC in all the evaluation measures. This shows that the integration of multi-view learning with the sparse coding could represent the images more robustly.

Compared with mHDSC, which utilizes different dictionaries but the same sparse coefficients for all the views, and mSSR, which allows diversity of sparse coefficients among different views while neglecting the correlation between multiple views without any regularization constraint, RmSSR outperforms both of them in all the measures. MvJSC gets a little higher performance on ESP Game dataset than mSSR, but much lower on

Table 2. Annotation performance evaluation on both datasets. The top portion displays published results of some KNN-based state-of-the-art methods. The middle part shows the results of the sparse code based framework methods, and the bottom rows list results of those methods dealing with deep learning. The results in bracket are obtained under equal view weights.

Method	ESP Game				IAPR TC12			
	AP	AR	F1	N+	AP	AR	F1	N+
JEC [7]	0.22	0.25	0.23	224	0.28	0.29	0.29	250
TagProp [8]	0.39	0.27	0.32	239	0.46	0.35	0.40	266
2PKNN_ML [11]	**0.53**	0.27	0.36	252	0.53	0.32	0.40	277
MSC* [21]	0.35	0.23	0.28	236	0.35	0.28	0.31	252
mHDSC* [27]	0.43	0.27	0.33	248	0.45	0.32	0.37	260
MvJSC [29]	0.41	0.28	0.33	245	0.42	0.32	0.36	261
KMvJSC [30]	0.44	0.29	0.35	255	0.46	0.34	0.39	268
mSSR_T	0.41	0.26	0.32	239	0.48	0.34	0.40	264
RmSSR_T	0.46 (0.44)	0.31 (0.30)	0.37 (0.35)	254 (250)	0.52 (0.50)	0.37 (0.37)	0.43 (0.43)	281 (277)
CNN-R [12]	0.45	0.29	0.35	248	0.49	0.31	0.38	272
MVSAE [13]	0.47	0.28	0.34	246	0.43	0.38	0.40	283
CCA-KNN_BV [12]	0.44	**0.32**	0.37	254	0.41	0.34	0.37	273
JEC_VGG [12]	0.26	0.22	0.24	234	0.28	0.21	0.24	237
2PKNN_VGG [12]	0.40	0.23	0.29	250	0.38	0.23	0.29	261
RmSSR_VGG	0.44 (0.42)	0.27 (0.26)	0.33 (0.32)	251 (244)	0.53 (0.47)	0.33 (0.31)	0.40 (0.37)	266 (260)
RmSSR_T+VGG	0.49 (0.46)	**0.32** (0.28)	**0.39** (0.35)	**256** (247)	**0.54** (0.53)	**0.38** (0.37)	**0.45** (0.44)	**286** (278)

IAPR TC12 dataset, which shows that joint sparsity constraint is still limited to employ the similarity and diversity of multiple views. Although KMvJSC improves the performance by kernel space mapping greatly, it is still less than RmSSR in F1 measure on ESP Game dataset, and in all measures on IAPR TC12 dataset. These results verify that the flexibility and the similarity of sparse coefficients in different views are both important for discrimination, and the tradeoff between them can be reached effectively by the consensus matrix regularization, which automatically learns weights to distinguish the importance of different views and minimizes the difference between coding coefficient matrix and consensus matrix to enforce similarity.

For other state-of-the-arts, we see that our proposed RmSSR model using only T features is better than or at least equal to all the previous works in F1 measure and N+ on both datasets, which verifies that our method is competitive.

Our RmSSR using VGG feature yields slightly worse results than that using T features, which is generally in accordance with the cases for JEC and 2PKNN using VGG feature compared with those using T features. That may because the feature is learned using a pretraining CNN specific for single-label image classification, it may not be optimal for image annotation, which is a multi-label classification problem.

Compared with RmSSR_T and RmSSR_VGG, RmSSR_T+VGG integrates handcrafted features and deep learning based feature together, and improves the performance of

RmSSR further on both datasets, which demonstrates that both sets of features have the complementary information and can be employed effectively by multi-view learning.

From Table 2, we can see that the learned view weights are beneficial to the image annotation on both datasets. We notice that in the case of equal view weights, RmSSR_T+VGG is even inferior to RmSSR_T in terms of R, F1, and N+ on ESP Game dataset. This demonstrates that more views may not lead to higher performance without weight learning for each view. By integrating multi-view learning with weight learning, our method can utilize the complementary among different views more effectively.

Figure 1 presents the learned weights for ESP Game and IAPR TC-12 datasets, which shows that the views of color features with layout information are the most important on ESP Game dataset while the views of SIFT features take more important effect on IAPR TC-12 dataset.

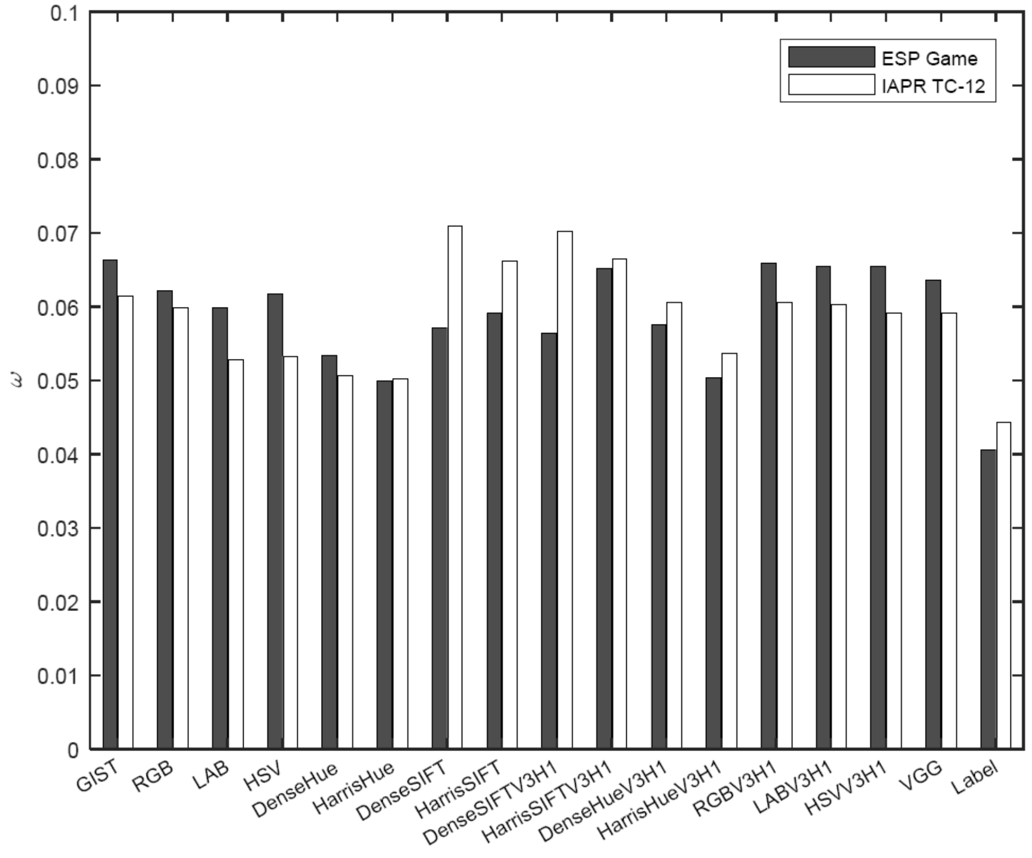


FIGURE 1. Learned weights for different views on ESP Game and IAPR TC-12 datasets

4.5. **Complexity analysis.** In this part, we compare the computation complexity of our RmSSR approach with those of related sparse coding based image annotation methods including MSC [21], mHDSC [27], MvJSC [29], and KMvJSC [30].

The main computation cost of RmSSR comes from the gradient calculation for subproblem (8) and (10). Suppose the number of views as v, the average dimension of all views as p, and the number of iteration as k for subproblem (8) and (10), we optimize the sparse codes and the dictionaries with the same complexity $O\big[kv\big(N_dNp+N_d^2N+N_d^2p\big)\big]$. Denote the number of alternating iterations as η, and the number of candidate parameters that need the δ-fold cross-validation as τ. Therefore, the total computation cost of RmSSR is $O\big[2k\tau\delta\eta v\big(N_dNp+N_d^2N+N_d^2p\big)\big]$.

Table 3 lists the computation complexities of our RmSSR and related sparse coding based image annotation methods. In the complexity of MSC, γ denotes the number of nonzero entries in the sparse coefficient. Usually, N_d and p are much smaller than N. Therefore, our RmSSR has a comparatively smaller computational complexity than all other methods except MvJSC, which demonstrates the competitiveness of our method.

Table 3. Computation complexity of RmSSR and related sparse coding based image annotation methods

Method	Computation Complexity
MSC [21]	$O\left[(vp)^3 + N^2\gamma^2\right]$
mHDSC [27]	$O\left[k\tau\delta\eta v\left(2N_dNp + 2N_d^2N + 2N_d^2p + N_dN^2\right)\right]$
MvJSC [29]	$O[2k\tau\delta\eta vpN_dN)]$
KMvJSC [30]	$O\left[\tau\delta\eta v\left(kN^3 + 2kN^2N_d + pN^2 + pN\right)\right]$
RmSSR	$O\left[2k\tau\delta\eta v\left(N_dNp + N_d^2N + N_d^2p\right)\right]$

4.6. **Parameter sensitivity.** We investigate the sensitivities of the parameters λ_1, λ_2 and λ_3 in our approach using the IAPR-TC12 dataset as an example. Figure 2 demonstrates the F1 measure variation versus different combinations of λ_1 and λ_2 with fixing $\lambda_3 = 0.01$. We can see that the best result is obtained when $\lambda_1 = 0.01$ and $\lambda_2 = 0.1$. This means the regularization terms of sparse coefficient and sparse dictionary are effective when both of them are not too small and too large. Figure 3 demonstrates the F1 measure variation versus different λ_3 with fixing $\lambda_1 = 0.01$ and $\lambda_2 = 0.1$. We can see the consensus matrix regularization term taking the best effect when $\lambda_3 = 0.01$. If it is too

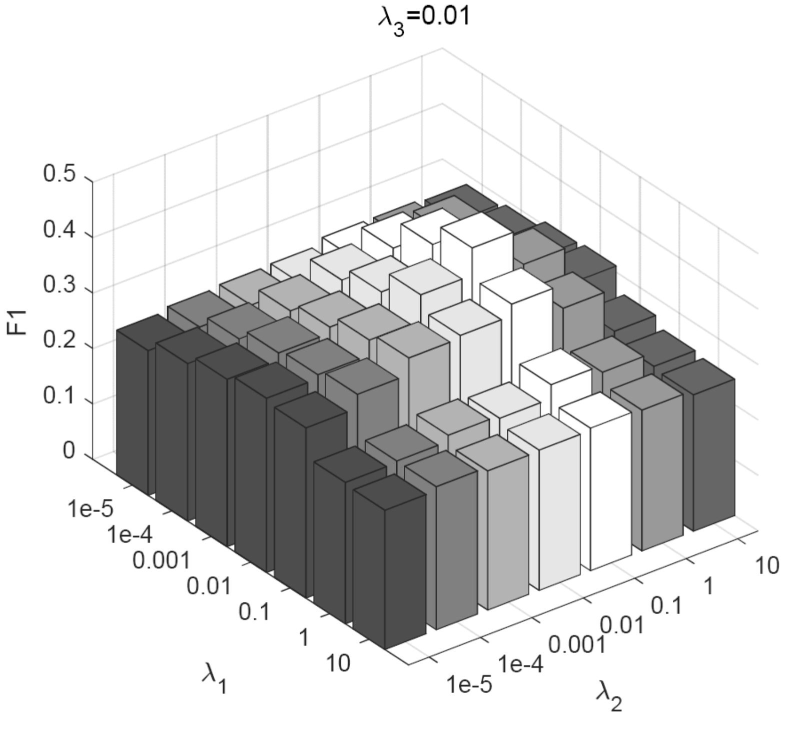


Figure 2. The F1 measure of RmSSR versus different combinations of λ_1 and λ_2 on IAPR TC-12 dataset

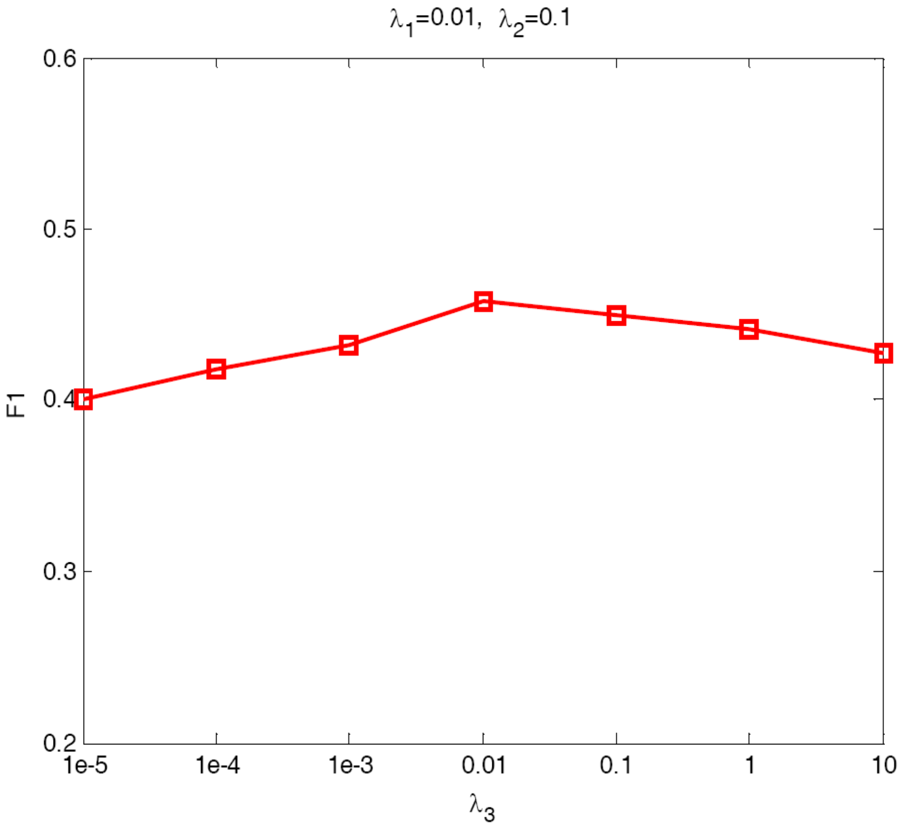


Figure 3. The F1 measure of RmSSR versus different λ_3 on IAPR TC-12 dataset

small, the correlations among multiple views are lost while it limits the diversity of all the variety of views with large values.

5. **Conclusion.**

In this paper, we present a regularized multi-view structured sparse representation model for image annotation. The contributions of our work are mainly embodied in three aspects. Firstly, we introduce a weighted soft-consensus regularization term into multi-view structured sparsity framework to effectively exploit the similarity and distinctiveness of all views for coding and annotation. This framework enforces the coefficient matrix corresponding to each view to be similar to a consensus matrix across all views. Meanwhile, it distinguishes the distinctiveness of the coefficients from different views over the associated dictionaries by adaptive weighting. Secondly, we propose the optimization method for our formulation based on APG method. The corresponding label prediction and propagation algorithm is presented to annotate the test image. Thirdly, we conduct experiments using the conventional handcraft features (including label information), deep learning based feature and both respectively on two datasets. The experimental results demonstrate that exploiting similarity and distinctiveness of multiple views simultaneously helps to improve the annotation performance greatly, and the integration of complementary information of hand-crafted features and deep learning based features is useful for discrimination. For future work, following [10,11], we intend to introduce label weight matrices to our framework to solve the class imbalance problem by increasing the recall of rare labels.

Acknowledgment. This work is supported by National Natural Science Foundation of China (61371143) and the excellent youthful teacher project of North China University of Technology (XN019006).

REFERENCES

[1] Y. Verma and C. V. Jawahar, Exploring SVM for image annotation in presence of confusing labels, *Proc. of the 24th British Machine Vision Conf.*, Bristol, UK, pp.1-11, 2013.

[2] G. Carneiro, A. B. Chan, P. J. Moreno and N. Vasconcelos, Supervised learning of semantic classes for image annotation and retrieval, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.29, no.3, pp.394-410, 2007.

[3] M. L. Zhang and L. Wu, LIFT: Multi-label learning with label-specific features, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.37, no.1, pp.107-120, 2015.

[4] O. Yakhnenko and V. Honavar, Annotating images and image objects using a hierarchical Dirichlet process model, *Proc. of the 9th International Workshop on Multimedia Data Mining*, Las Vegas, NV, USA, pp.1-7, 2008.

[5] D. Putthividhya, H. T. Attias and S. S. Nagarajan, Topic regression multi-modal latent Dirichlet allocation for image annotation, *Proc. of the 23rd IEEE International Conf. on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, pp.3408-3415, 2010.

[6] Z. Li, Z. Shi, Z. Li and Z. Shi, Modeling latent aspects for automatic image annotation, *Proc. of IEEE the 16th International Conf. on Image Processing*, pp.1857-1860, 2009.

[7] A. Makadia, V. Pavlovic and S. Kumar, A new baseline for image annotation, *Proc. of the 10th European Conf. on Computer Vision*, Marseille, France, pp.316-329, 2008.

[8] M. Guillaumin, T. Mensink, J. Verbeek and C. Schmid, TagProp: Discriminative metric learning in nearest neighbor models for image auto-annotation, *Proc. of the 12th IEEE International Conf. on Computer Vision*, Kyoto, Japan, pp.309-316, 2009.

[9] S. Zhang, J. Huang, H. Li and D. N. Metaxas, Automatic image annotation and retrieval using group sparsity, *IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol.42, no.3, pp.838-849, 2012.

[10] M. M. Kalayeh, H. Idrees and M. Shah, NMF-KNN: Image annotation using weighted multi-view non-negative matrix factorization, *Proc. of the 27th IEEE International Conf. on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp.184-191, 2014.

[11] Y. Verma and C. V. Jawahar, Image annotation by propagating labels from semantic neighbourhoods, *International Journal of Computer Vision*, vol.121, no.1, pp.126-148, 2017.

[12] V. N. Murthy, S. Maji and R. Manmatha, Automatic image annotation using deep learning representations, *Proc. of the 5th ACM on International Conf. on Multimedia Retrieval*, pp.603-606, 2015.

[13] Y. Yang, W. Zhang and Y. Xie, Image automatic annotation via multi-view deep representation, *Journal of Visual Communication and Image Representation*, vol.33, pp.368-377, 2015.

[14] F. Wu, Z. Wang, Z. Zhang and Y. Yang, Weakly semi-supervised deep learning for multi-label image annotation, *IEEE Trans. Big Data*, vol.1, no.3, pp.109-122, 2015.

[15] J. Wu, Y. Yu, C. Huang and K. Yu, Deep multiple instance learning for image classification and auto-annotation, *Proc. of the 28th IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp.3460-3469, 2015.

[16] A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Proc. of the 26th International Conf. on Neural Information Processing Systems*, Lake Tahoe, NV, USA, pp.1097-1105, 2012.

[17] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, *Proc. of the 3rd International Conf. on Learning Representations*, San Diego, CA, USA, pp.1-14, 2015.

[18] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma, Robust face recognition via sparse representation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.31, no.2, pp.210-227, 2009.

[19] M. Yang, L. Zhang, D. Zhang and S. Wang, Relaxed collaborative representation for pattern classification, *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp.2224-2231, 2012.

[20] J. He, T. Zuo, B. Sun, X. Wu, Y. Xiao and X. Zhu, Robust face recognition framework with block weighted sparse representation based classification, *International Journal of Innovative Computing, Information and Control*, vol.11, no.5, pp.1551-1562, 2015.

[21] C. Wang, S. Yan, L. Zhang and H.-J. Zhang, Multi-label sparse coding for automatic image annotation, *Proc. of the 22nd IEEE International Conf. on Computer Vision and Pattern Recognition*, Miami, FL, USA, pp.1643-1650, 2009.

[22] S. H. Gao, L. T. Chia, I. W. Tsang and Z. Ren, Concurrent single-label image classification and annotation via efficient multi-layer group sparse coding, *IEEE Trans. Multimedia*, vol.16, no.3, pp.762-771, 2014.

[23] X. Cao, H. Zhang, X. Guo, S. Liu and D. Meng, SLED: Semantic label embedding dictionary representation for multilabel image annotation, *IEEE Trans. Image Processing*, vol.24, no.9, pp.2746-2759, 2015.

[24] Z. Lu, P. Han, L. Wang and J.-R. Wen, Semantic sparse recoding of visual content for image applications, *IEEE Trans. Image Processing*, vol.24, no.1, pp.176-188, 2015.

[25] X.-Y. Jing, F. Wu, Z. Li, R. Hu and D. Zhang, Multi-label dictionary learning for image annotation, *IEEE Trans. Image Processing*, vol.25, no.6, pp.2712-2725, 2016.

[26] S. Moran and V. Lavrenko, Sparse kernel learning for image annotation, *Proc. of the ACM the 4th International Conf. on Multimedia Retrieval*, Glasgow, UK, pp.113-120, 2014.

[27] W. Liu, D. Tao, J. Cheng and Y. Tang, Multiview Hessian discriminative sparse coding for image annotation, *Computer Vision and Image Understanding*, vol.118, pp.50-60, 2014.

[28] M. Zang, H. Xu and Y. Zhang, Multi-view mixed-norm sparse coding for image annotation, *ICIC Express Letters, Part B: Applications*, vol.7, no.11, pp.2483-2490, 2016.

[29] M. Zang and H. Xu, Multi-view joint sparse coding for image annotation, *International Journal of Innovative Computing, Information and Control*, vol.13, no.4, pp.1407-1414, 2017.

[30] M. Zang, H. Xu and Y. Zhang, Kernel-based multiview joint sparse coding for image annotation, *Mathematical Problems in Engineering*, vol.2017, no.4, pp.1-11, 2017.

[31] Y. Jia, T. Darrell and M. Salzmann, Factorized latent spaces with structured sparsity, *Proc. of the 24th International Conf. on Neural Information Processing Systems*, Vancouver, BC, Canada, pp.982-990, 2010.

[32] M. Aharon, M. Elad and A. Bruckstein, K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation, *IEEE Trans. Signal Processing*, vol.54, no.11, pp.4311-4322, 2006.

[33] J. C. Bezdek and R. J. Hathaway, Convergence of alternating optimization, *Neural Parallel and Scientific Computations*, vol.11, no.4, pp.351-368, 2003.

[34] X. Chen, W. Pan, J. T. Kwok and J. G. Carbonell, Accelerated gradient method for multi-task sparse learning problem, *Proc. of the 5th IEEE International Conf. on Data Mining*, Les Vegas, NV, USA, pp.746-751, 2009.

[35] P. Tseng, On accelerated proximal gradient methods for convex-concave optimization, *SIAM Journal on Optimization*, pp.1-20, 2008.

[36] L. V. Ahn and L. Dabbish, Labeling images with a computer game, *Proc. of SIGCHI Conf. on Human Factors in Computing Systems*, pp.319-326, 2004.

[37] M. Grubinger, P. Clough, H. Müller and T. Deselaers, The IAPR TC-12 benchmark: A new evaluation resource for visual information systems, *Proc. of the 5th International Conf. on Language Resources and Evaluation*, Genoa, Italy, pp.13-23, 2006.