Beyond Explicit Codebook Generation: Visual Representation Using Implicitly Transferred Codebooks

Chunjie Zhang, Jian Cheng, Jing Liu, Junbiao Pang, Qingming Huang, and Qi Tian, *Senior Member, IEEE*

***Abstract*— The bag-of-visual-words model plays a very important role for visual applications. Local features are first extracted and then encoded to get the histogram-based image representation. To encode local features, a proper codebook is needed. Usually, the codebook has to be generated for each data set which means the codebook is data set dependent. Besides, the codebook may be biased when we only have a limited number of training images. Moreover, the codebook has to be pre-learned which cannot be updated quickly, especially when applied for online visual applications. To solve the problems mentioned above, in this paper, we propose a novel implicit codebook transfer method for visual representation. Instead of explicitly generating the codebook for the new data set, we try to make use of pre-learned codebooks using non-linear transfer. This is achieved by transferring the pre-learned codebooks with non-linear transformation and use them to reconstruct local features with sparsity constraints. The codebook does not need to be explicitly generated but can be implicitly transferred. In this way, we are able to make use of pre-learned codebooks for new visual applications by implicitly learning the codebook and the corresponding encoding parameters for image representation. We apply the proposed method for image classification and evaluate the performance on several public image data sets. Experimental results demonstrate the effectiveness and efficiency of the proposed method.**

***Index Terms*— Codebook transfer, image representation, classification, reconstruction, sparse constraint.**

I. Introduction

The BAG-OF-VISUAL-WORDS (BoW) model [1] is widely used for visual representation. Usually, local features are first extracted either by dense sampling or detection. A codebook is then learned (using k-means clustering [1] or sparse coding [2]) to encoded local features. The encoded parameters are used for image representation which is then applied for various visual applications, such as image classification [1], [2], image retrieval [3] and object detection [4]. The codebook and the corresponding feature encoding strategy are very important for image representation. Many works have been done to generate more discriminative codebooks, such as sparse coding [2], kernel codebook [5], locality-constrained sparse coding [6], laplacian sparse coding [7] etc. Although these methods have been proven very effective, they still have three drawbacks. First, the codebook generation process is dataset dependent which means we have to generate codebook for each dataset. Directly using the codebook generated by one particular dataset can not be able to perform as good as using the corresponding codebook. Second, when we only have a limited number of images, the generated codebook is probably biased and can not be able to represent the whole dataset's images well. Third, since the pre-learned codebook is fixed, it may not be able to cope with online visual applications. If we can transfer the pre-learned codebooks, we may be able to solve these problems to some extent.

To overcome the dataset dependency problem, researchers try to generate universal codebooks and then adapt them for particular visual applications [8]–[10]. This is often achieved by collecting a large number of images and use the extracted local features for universal codebook generation. As long as the collected images are ‘enough’, the generated universal codebook is able to cope with the dataset dependency problem. However, this strategy is time consuming both for image collection and for the codebook generation. Besides, how to judge the collected images are adequate is left unsolved. Moreover, the generated universal codebook is still dataset dependent after images are collected.

Manuscript received January 27, 2015; revised May 25, 2015, June 28, 2015, and July 28, 2015; accepted August 23, 2015. Date of publication October 1, 2015; date of current version October 27, 2015. This work was supported in part by the Open Project through the Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, in part by the National Basic Research Program of China (973 Program) under Grant 2012CB316400 and Grant 2015CB351802, and in part by the National Natural Science Foundation of China under Grant 61025011, Grant 61170127. Grant 61202234, Grant 61272329, Grant 61303154, Grant 61332016 and Grant 61429201. This work was supported in part to Dr. Qi Tian by ARO grants W911NF-15-1-0290 and W911NF-12-1-0057 and Faculty Research Awards by NEC Laboratories of America. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Lei Zhang.

C. Zhang is with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100864, China (e-mail: zhangcj@ucas.ac.cn).

J. Cheng and J. Liu are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: jcheng@nlpr.ia.ac.cn; jliu@nlpr.ia.ac.cn).

J. Pang is with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, College of Metropolitan Transportation, Beijing University of Technology, Beijing 100124, China (e-mail: junbiao_pang@bjut.edu.cn).

Q. Huang is with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China, also with the Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100864, China, and also with the Key Laboratory of Intell. Info. Process, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100864, China (e-mail: qmhuang@ucas.ac.cn).

Q. Tian is with the Department of Computer Sciences, University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: qitian@cs.utsa.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2485783

To make full use of images for codebook generation, the use of semi-supervised methods [11], [12] and transfer learning methods [13]–[17] are also proposed. By incorporating other information, we can represent the images better with more discriminative codebooks. Besides, the joint learning of ‘universal’ and ‘specific’ codebooks helps to represent images more properly. However, they are still unable to solve the dataset dependent problem and can only be updated by re-training. Although the use of linear transformation of codebooks [17] helps to alleviate this problem, it suffers three problems. First, only linear transformation may not be able to cope with the variations of visual features and non-linear transformation is needed. Second, the dimensions of encoded parameters increase with the number of transferred codebooks which costs a lot of computation power and storage, especially when a large number of codebooks are transferred. Third, linear transformation means we can only transfer codebooks generated by the same local features (e.g. SIFT [18]). However, different local features may be extracted from different image datasets (e.g. PCA-SIFT [19], HoG [20]) which can not be solved by [17] directly.

To solve the problems mentioned above, in this paper, we propose a novel implicit codebook transfer method for image representation which can be used for various visual applications. For one particular image dataset, instead of generating the corresponding codebook directly, we try to non-linearly transfer the pre-learned codebooks of other datasets. The optimal transfer parameters are learned by minimizing the summed reconstruction error of local features. Sparsity constraints is used for selection and to avoid over-fitting. The non-linear transformation goes one step beyond linear transformation and improves the discriminative power of image representation. To evaluate the performance of the proposed method, we conduct image classification experiments on several public image datasets. The results demonstrate the effectiveness of the proposed method.

The main contributions of this paper lie in four aspects. First, we make use of the pre-learned codebooks for new visual applications instead of generating the corresponding codebook. Second, by using non-linear transfer, we are able to cope with the variations of images and represent images better than linear transfer. Third, we can transfer codebooks generated with different types of local features while linear transfer can only deal with codebooks generated by the same type of local features. Fourth, we are able to improve the image representation efficiency over linear transfer which eventually helps the final visual applications.

The rest of this paper is organized as follows. Related work is given in Section 2. The details of the proposed implicit codebook transfer method are given in Section 3. To evaluate the effectiveness of the proposed method, we conduct experiments on several public image datasets for image classification in Section 4 and conclude in Section 5.

II. Related Work

The BoW model was widely used for various visual applications. Images are first extracted with local features either by dense sampling or detection. A codebook was then learned using these local features by k-means clustering and local features were assigned by nearest neighbor search [1]. This resulted in quantization loss which was alleviated by soft-assignment methods such as sparse coding [2] and kernel codebook [5]. Since the codebook generation and the local feature encoding are correlated, researchers have proposed various codebook generation and encoding methods [6], [7], [21]–[23]. Locality-constrained sparse coding [6] was proposed by Wang *et. al* to consider the locality information during the sparse coding process. This helped to reduce the information loss and speeded up the encoding of local features. Gao *et al.* [7] proposed laplacian sparse coding to ensure similar local features are encoded with similar parameters. To consider the spatial information during codebook generation, Zhang *et al.* [21] proposed to use spatial pyramid coding instead. Fisher vector was proposed to further reduce the information loss of encoded local features by Sánchez *et al.* [22] which improved the image classification performance. Kobayashi [23] used Dirichlet-based histogram feature transform with Dirichlet Fisher kernels for classification. Although very effective, the codebooks generated by the above methods were all dataset dependent.

To alleviate this problem, researchers tried to generate universal codebooks instead [8]–[10]. Perronnin *et al.* [8] tried to first construct an universal codebook and then adapt it for different classification tasks using the Gaussian Mixture Model (GMM). Zhang *et al.* [9] collected a large number of images from various sources and used them to generate the universal codebook with contextual information. This costed a lot of storage and computation time. Winn *et al.* [10] also generated universal visual codebooks for object categorization with good performance. However, the collection and computational costs of these methods are high which limits their applications. Since training images are often limited, the use of other images by semi-supervised methods becomes popular [11], [12]. Mũnoz-Marí *et al.* [11] proposed the semi-supervised one-class support vector machines and applied it for remote sensing data classification. Guillaumin *et al.* [12] made use of multi-modal information of images and combined it with semi-supervised learning for image classification with encouraging performance. However, these methods are still dataset dependent after the dataset is collected.

The uses of transfer based methods were also proposed by researchers [13]–[17]. Ramirez *et al.* [13] tried to generate a codebook with structured incoherence and shared features while Yang *et al.* [14] proposed to reuse structured knowledge for transfer learning. As to fine-grained image classification, Gao *et al.* [16] proposed to jointly learn the category-specific codebook and shared codebooks through optimization and achieved good performance. Zhang *et al.* [17] tried to undo the codebook bias by linearly transform pre-learned codebooks with sparsity and F-norm constraints. However, the linear transform may not be able to model codebooks well and non-linear transformation is necessary to improve the performance.

Using non-linear transformation has also been widely used by researchers, e.g. the neural network [24], random forest [25] and markov random field [26]. The kernel method [27], [28]

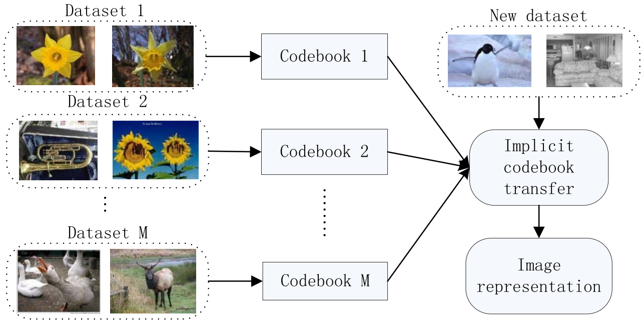


Fig. 1. Flowchart of the proposed implicit codebook transfer method for image representation.

was also used for non-linear analysis without explicitly computing the coordinates in the mapped space. It computed the inner products between the features to be mapped instead. Duan *et al.* [28] proposed to use multiple kernel learning for domain transferring and achieved superior performance over many methods. This is achieved by minimizing both the structural risk functional and the distribution mismatch between the labeled and unlabeled samples. This strategy was very useful for visual applications. For example, the support vector machine (SVM) [29] was often used as classifier for image class prediction.

To avoid generating the codebook, researchers also explored using local features for image classification directly [30]–[32]. Yang *et al.* [30] tried to unify codebook generation with classifier training for object categorization. Instead of training classifiers, Boiman *et al.* [31] made use of the nearest-neighbor information of local features by calculating the 'Image-to-Class' distances. Muja and Lowe [32] extended this for fast computation using scalable nearest neighbor information. Besides, the learning of features from images [33]–[35] directly has also been proposed with good performance. LeCun *et al.* [33] used convolutional networks while Zeiler *et al.* [34] proposed deconvolutional networks for visual applications. Shao *et. al* proposed a multi-objective genetic programming method for feature learning and applied it for image classification with encouraging performance. However, the computational cost of these methods are relatively high compared with the codebook based methods. The use of subspace techniques [36]–[39] were also widely used for different domain modeling with good performances.

III. Transfer Implicit Codebooks for Image Representation

In this section, we give the details of the proposed implicit codebook transfer method for image representation. We then apply it for image classification. Figure 1 shows the flowchart of the proposed method. We also give the symbols and their descriptions used in this paper in Table 1.

A. Linear Codebook Transfer

TABLE I
The Symbols Used in This Paper and Their Corresponding Descriptions

Symbol	Description
$\mathbf{x_n}$	extracted local feature
$\mathbf{B}$	codebook
$\alpha_\mathbf{n}$	encoded parameter
$\mathbf{D}$	basis vector set
$\mathbf{B_m}$	pre-learned codebooks
$\mathbf{A_m}$	linear transformation matrix
$\gamma_\mathbf{m}$	linear transformation parameter
$\mathbf{B_{M+1}}$	new codebook to be learned
$\mathbf{A_{M+1}}$	linear transformation matrix for $\mathbf{B_{M+1}}$
$\varphi_m(\mathbf{b_m}^j)$	non-linear transformation of $\mathbf{b_m}^j$
Q_m	number of visual words in m-th codebook
γ	non-linear transformation parameter
$\mathbf{b_g}$	anchor point for non-linear transform
$\mathbf{K}$	non-linear transform matrix
$\mathbf{h}$	max pooled image representation
y	image label
$\mathbf{w_c}$	linear classifier parameter
$\ell(.,.)$	quadratic hinge loss
ϵ	linear classifier regularization parameter

Recently, the use of sparse coding for codebook generation becomes popular. Formally, let $\mathbf{x_n}, n = 1, \ldots, N$ be the local features where N is the number of local features. The sparse coding technique tries to find the optimal codebook $\mathbf{B}$ and the encoding parameters $\alpha_\mathbf{n}$ by:

$$min_{\mathbf{B},\alpha_\mathbf{n}} \sum_{n=1}^{N} \parallel \mathbf{x_n} - \mathbf{B}\alpha_\mathbf{n} \parallel^2 + \lambda \parallel \alpha_\mathbf{n} \parallel_1 \tag{1}$$

This is often optimized by alternatively solving for the optimal $\mathbf{B}/\alpha_\mathbf{n}$ while fixing $\alpha_\mathbf{n}/\mathbf{B}$ [40].

However, the codebook generated in this way is dataset dependent. To take advantage of the pre-learned codebooks of other image datasets, a linear codebook transfer method was used by [17]. It is based on the assumption that each point can be represented by the basis vectors in the local feature space. In other words, each codebook can be represented using these basis vectors. Hence the linear transformation of codebooks is possible. Let $\mathbf{D} = [\mathbf{d_1}, \mathbf{d_2}, \ldots, \mathbf{d_P}]$ be the set of P basis vectors, $\mathbf{B_1}, \ldots, \mathbf{B_M}$ represent the M pre-learned codebooks which can be linearly represented by $\mathbf{D}$ as:

$$\mathbf{B_m} = \mathbf{D}\mathbf{A_m}^T, \quad m = 1, \ldots, M \tag{2}$$

Where $\mathbf{A_m}$ is the linear transformation matrix. Hence, the basis vector matrix can be computed as:

$$\mathbf{D} = \mathbf{B_m}(\mathbf{A_m}^T)^+, \quad m = 1, \ldots, M \tag{3}$$

Where $(\mathbf{A_m}^T)^+$ represents the psedoinverse of matrix $\mathbf{A_m}^T$. This can also be rewritten in a linear combination form as:

$$\mathbf{D} = \sum_{m=1}^{M} \gamma_m \mathbf{B_m}(\mathbf{A_m}^T)^+, \quad \sum_{m=1}^{M} \gamma_m = 1 \tag{4}$$

Where γ_m is the linear combination parameters.

Suppose we want to learn a codebook $\mathbf{B_{M+1}}$ for the new image dataset. It can also be linearly represented by $\mathbf{D}$ as:

$$\begin{aligned} \mathbf{B_{M+1}} &= \mathbf{D}\mathbf{A_{M+1}}^T \\ &= \sum_{m=1}^{M} \gamma_m \mathbf{B_m}(\mathbf{A_m}^T)^+ \mathbf{A_{M+1}}^T, \quad \sum_{m=1}^{M} \gamma_m = 1 \end{aligned} \tag{5}$$

Let $\hat{\mathbf{A}_\mathbf{m}}^T = (\mathbf{A_m}^T)^+ \mathbf{A_{M+1}}^T$, the above equation can be rewritten as:

$$\mathbf{B_{M+1}} = \mathbf{D}\mathbf{A_{M+1}}^T = \sum_{m=1}^{M} \gamma_m \mathbf{B_m} \hat{\mathbf{A}_\mathbf{m}}^T, \quad \sum_{m=1}^{M} \gamma_m = 1 \tag{6}$$

In this way, we are able to transfer the pre-learned codebooks for new image dataset. To encode local features with the new codebook $\mathbf{B_{M+1}}$, we can follow the sparse coding scheme and try to optimize:

$$min_{\mathbf{B_{M+1}},\alpha_\mathbf{n}} \sum_{n=1}^{N} \parallel \mathbf{x_n} - \mathbf{B}\alpha_\mathbf{n} \parallel^2 + \lambda \parallel \alpha_\mathbf{n} \parallel_1 \tag{7}$$

Which equals to:

$$\begin{aligned} min_{\gamma_m, \hat{\mathbf{A}_\mathbf{m}}, \alpha_\mathbf{n}} & \sum_{n=1}^{N} \parallel \mathbf{x_n} - \sum_{m=1}^{M} \gamma_m \mathbf{B_m} \hat{\mathbf{A}_\mathbf{m}}^T \alpha_\mathbf{n} \parallel^2 \\ & + \lambda \parallel \alpha_\mathbf{n} \parallel_1, \quad m = 1, \ldots, M \end{aligned} \tag{8}$$

This problem can be alternatively solved by optimizing over $\gamma_m, \hat{\mathbf{A}_\mathbf{m}}, \alpha_\mathbf{n}$ while keeping the others fixed. When searching for the optimal γ_m while keeping $\hat{\mathbf{A}_\mathbf{m}}$ and $\alpha_\mathbf{n}$ fixed, Problem 8 can be rewritten as:

$$min_{\gamma_m} \sum_{n=1}^{N} \parallel \mathbf{x_n} - \sum_{m=1}^{M} \gamma_m \mathbf{B_m} \hat{\mathbf{A}_\mathbf{m}}^T \alpha_\mathbf{n} \parallel^2, \quad m = 1, \ldots, M \tag{9}$$

When optimizing over $\hat{\mathbf{A}_\mathbf{m}}$ while keeping γ_m and $\alpha_\mathbf{n}$ fixed, Problem 8 can be rewritten as:

$$min_{\hat{\mathbf{A}_\mathbf{m}}} \sum_{n=1}^{N} \parallel \mathbf{x_n} - \sum_{m=1}^{M} \gamma_m \mathbf{B_m} \hat{\mathbf{A}_\mathbf{m}}^T \alpha_\mathbf{n} \parallel^2, \quad m = 1, \ldots, M \tag{10}$$

When optimizing over $\alpha_\mathbf{n}$ while keeping γ_m and $\hat{\mathbf{A}_\mathbf{m}}$ fixed, Problem 8 can be rewritten as:

$$\begin{aligned} min_{\alpha_\mathbf{n}} & \sum_{n=1}^{N} \parallel \mathbf{x_n} - \sum_{m=1}^{M} \gamma_m \mathbf{B_m} \hat{\mathbf{A}_\mathbf{m}}^T \alpha_\mathbf{n} \parallel^2 \\ & + \lambda \parallel \alpha_\mathbf{n} \parallel_1, \quad m = 1, \ldots, M \end{aligned} \tag{11}$$

Although this linear codebook transfer method is able to transfer the pre-learned codebooks for new visual applications, it still suffers two problems. First, only using linear transformation may be not able to cope with the intra-class and inter-class variations of images. Non-linear transformation can solve this problem to some extent. Second, the codebooks have to be generated with the same type of local features for linear codebook transfer. This restricts the scalability of the linear codebook transfer method.

B. Implict Codebook Transfer

To avoid the drawbacks of linear codebook transfer method, we use non-linear codebook transfer instead to implicitly learn the codebook. Let $\mathbf{B_m} = \{\mathbf{b_m}^1, \ldots, \mathbf{b_m}^{Q_m}\}, m = 1, \ldots, M+1$, similarly, each $\mathbf{b_{M+1}}^i$ can be represented as a combination of non-linear transform of each pre-learned codebook $\mathbf{B_m}$ as:

$$\mathbf{b_{M+1}}^i = \sum_{j=1}^{Q_m} \gamma_m^{i,j} \varphi_m(\mathbf{b_m}^j), \quad m = 1, \ldots, M \tag{12}$$

Where Q_m represents the number of visual words of the m-th codebook. In this way, we can optimize for the codebook $\mathbf{B_{M+1}}$ by solving the problem as:

$$\begin{aligned} min_{\mathbf{B_{M+1}},\alpha_\mathbf{n}} & \sum_{n=1}^{N} \parallel \mathbf{x_n} - \mathbf{B_{M+1}}\alpha_\mathbf{n} \parallel^2 + \lambda \parallel \alpha_\mathbf{n} \parallel_1 \\ & = \sum_{n=1}^{N} \mathbf{x_n}^T \mathbf{x_n} - \mathbf{x_n}^T \mathbf{B_{M+1}} \alpha_\mathbf{n} - \alpha_\mathbf{n}^T \mathbf{B_{M+1}}^T \mathbf{x_n} \\ & + \alpha_\mathbf{n}^T \mathbf{B_{M+1}}^T \mathbf{B_{M+1}} \alpha_\mathbf{n} + \lambda \parallel \alpha_\mathbf{n} \parallel_1 \end{aligned} \tag{13}$$

Which equals to

$$\begin{aligned} min_{\mathbf{B_{M+1}},\alpha_\mathbf{n}} & \sum_{n=1}^{N} -\mathbf{x_n}^T \mathbf{B_{M+1}} \alpha_\mathbf{n} - \alpha_\mathbf{n}^T \mathbf{B_{M+1}}^T \mathbf{x_n} \\ & + \alpha_\mathbf{n}^T \mathbf{B_{M+1}}^T \mathbf{B_{M+1}} \alpha_\mathbf{n} + \lambda \parallel \alpha_\mathbf{n} \parallel_1 \end{aligned} \tag{14}$$

With $\mathbf{B_{M+1}} = \{\sum_{j=1}^{Q_m} \gamma_m^{1,j} \varphi_m(\mathbf{b_m^j}), \ldots, \sum_{j=1}^{Q_m} \gamma_m^{Q_{M+1},j} \varphi_m(\mathbf{b_m^j})\}$.

Besides, the $\mathbf{x_n}$ can also be represented as a combination of these non-linear transferred visual words $\varphi_m(\mathbf{b_m}^i)$ of each pre-learned codebook as:

$$\mathbf{x_n} = \sum_{i=1}^{Q_{\hat{m}}} \beta_{\hat{m}}^{n,i} \varphi_{\hat{m}}(b_{\hat{\mathbf{m}}}^i), \quad \hat{m} = 1, \ldots, M \tag{15}$$

With the optimal parameters $\beta_{\hat{m}}^{n,i}$ learned through reconstruction. Each dimension of the non-linear transferred $\varphi(\mathbf{b})$ is defined as:

$$\varphi(\mathbf{b})_g = \Gamma(\mathbf{b}, \mathbf{b_g}), \quad g = 1, \ldots, G \tag{16}$$

Where $\mathbf{x_n} \in \mathbb{R}^{G \times 1}$, $\mathbf{b_g}$ is the anchor point which is randomly chosen from the visual words of each codebook. We can use the kernel trick in machine learning as the non-linear transferred function $\Gamma(\mathbf{b}, \mathbf{b_g})$, such as the radial basis function kernel (RBF) and the histogram intersection kernel (HIK). In this way, we can avoid the manual design of non-linear transfer function and make use of the popular methods whose effectiveness have been proven by researchers. In this way, we can rewrite each part of Problem 14 as:

$$\begin{aligned} \mathbf{x_n^T} \mathbf{B_{M+1}} \alpha_\mathbf{n} &= \sum_{i=1}^{Q_{\hat{m}}} \beta_{\hat{m}}^{n,i} \varphi_{\hat{m}}^T(b_{\hat{\mathbf{m}}}^i) \times \{\sum_{j=1}^{Q_m} \gamma_m^{1,j} \varphi_m(\mathbf{b_m}^j), \ldots, \\ & \sum_{j=1}^{Q_m} \gamma_m^{Q_{M+1},j} \varphi_m(\mathbf{b_m}^j)\} \times \alpha_\mathbf{n} \\ &= \{\sum_{i=1}^{Q_1} \sum_{j=1}^{Q_1} \beta_1^{n,i} \gamma_1^{1,j} \varphi_1^T(\mathbf{b_1}^i) \varphi_1(\mathbf{b_1}^j), \ldots, \\ & \sum_{i=1}^{Q_M} \sum_{j=1}^{Q_M} \beta_M^{n,i} \gamma_M^{Q_{M+1},j} \varphi_M^T(\mathbf{b_M}^i) \varphi_M(\mathbf{b_M}^j)\} \times \alpha_\mathbf{n} \end{aligned} \tag{17}$$

with

$$\begin{aligned}
{\alpha_{\mathbf{n}}}^{T}{\mathbf{B_{M+1}}}^{T}\mathbf{x_n} &= \alpha_{\mathbf{n}}^{\mathbf{T}} \times \{\sum_{j=1}^{Q_m} {\gamma_m}^{1,j} \varphi_m^T({\mathbf{b_m}}^{j}); \ldots; \\
&\quad \sum_{j=1}^{Q_m} {\gamma_m}^{Q_{M+1},j} \varphi_m^T({\mathbf{b_m}}^{j})\} \times \sum_{i=1}^{Q_{\hat{m}}} \beta_{\hat{m}}^{n,i} \varphi_{\hat{m}}(b_{\hat{\mathbf{m}}}^{i}) \\
&= \{\sum_{i=1}^{Q_1}\sum_{j=1}^{Q_1} \beta_1^{n,i} {\gamma_1}^{1,j} \varphi_1^T({\mathbf{b_1}}^{i})\varphi_1({\mathbf{b_1}}^{j}); \ldots; \\
&\quad \sum_{i=1}^{Q_M}\sum_{j=1}^{Q_M} \beta_M^{n,i} {\gamma_M}^{Q_{M+1},j} \varphi_M^T({\mathbf{b_M}}^{i})\varphi_M({\mathbf{b_M}}^{j})\} \times \alpha_{\mathbf{n}}
\end{aligned} \tag{18}$$

and

$$\begin{aligned}
&{\alpha_{\mathbf{n}}}^{T}{\mathbf{B_{M+1}}}^{T}\mathbf{B_{M+1}}\alpha_{\mathbf{n}} \\
&\quad = {\alpha_{\mathbf{n}}}^{T} \times \{\sum_{j=1}^{Q_m} {\gamma_m}^{1,j} \varphi_m^T({\mathbf{b_m}}^{j}); \ldots; \\
&\qquad \sum_{j=1}^{Q_m} {\gamma_m}^{Q_{M+1},j} \varphi_m^T({\mathbf{b_m}}^{j})\} \times \{\sum_{j=1}^{Q_m} {\gamma_m}^{1,j} \varphi_m({\mathbf{b_m}}^{j}), \ldots, \\
&\qquad \sum_{j=1}^{Q_m} {\gamma_m}^{Q_{M+1},j} \varphi_m({\mathbf{b_m}}^{j})\} \times \alpha_{\mathbf{n}}
\end{aligned} \tag{19}$$

Let $\gamma = ({\gamma_1}^{1,1}; \ldots; {\gamma_M}^{Q_{M+1},Q_M})$, $\beta^n = ({\beta_1^{n,1}}; \ldots; {\beta_M^{n,Q_M}})$, $\hat{\beta^n} = (\beta_1^{n,1}{\gamma_1}^{1,1}; \ldots; \beta_M^{n,Q_M}{\gamma_M}^{Q_{M+1},Q_M})$.

Let

$$\mathbf{K} = \begin{pmatrix}
\varphi_1^T({\mathbf{b_1}}^{1})\varphi_1({\mathbf{b_1}}^{1}) & . & \varphi_1^T({\mathbf{b_1}}^{Q_1})\varphi_M({\mathbf{b_M}}^{Q_M}) \\
\varphi_1^T({\mathbf{b_1}}^{1})\varphi_1({\mathbf{b_1}}^{2}) & . & . \\
. & . & . \\
. & . & . \\
. & . & . \\
\varphi_1^T({\mathbf{b_1}}^{Q_1})\varphi_M({\mathbf{b_M}}^{Q_M}) & . & \varphi_M^T({\mathbf{b_M}}^{Q_M})\varphi_M({\mathbf{b_M}}^{Q_M})
\end{pmatrix} \tag{20}$$

Problem 14 then equals to:

$$min_{\gamma,\alpha_{\mathbf{n}}} \sum_{n=1}^{N} {\alpha_{\mathbf{n}}}^{T} \gamma \mathbf{K} \gamma^{T} \alpha_{\mathbf{n}} - 2 \times {\alpha_{\mathbf{n}}}^{T} \mathbf{K}\hat{\beta^{\mathbf{n}}} + \lambda \parallel \alpha_{\mathbf{n}} \parallel_1 \tag{21}$$

In this way, we are able to transform the searching for the codebook to the problem of finding the optimal transformation coefficients by making use of the pre-learned codebooks. In other words, we do not need to explicitly find the optimal codebook. Instead, we can use the learned encoding parameters $\alpha_{\mathbf{n}}$ for image representation directly. If we set all the $\varphi(*)$ to be linear functions, the proposed method will degenerate to the linear codebook transfer method [17]. Hence, it can be viewed as a special case of the proposed method in this paper.

It is hard to optimize over γ and $\alpha_{\mathbf{n}}$ jointly. Hence, we follow the alternative optimization strategy as [2], [17], [40] and try to find the optimal $\gamma/\alpha_{\mathbf{n}}$ while keeping $\alpha_{\mathbf{n}}/\gamma$ fixed.

Algorithm 1 The Proposed Feature-Sign-Search Algorithm for Solving Problem 23

```
Input:
   The local features $\mathbf{x}$, $\hat{\beta}$, $\mathbf{K}, \alpha$, $\rho := sign(\alpha)$,
   $activeset := (\alpha \neq 0)$, let $f(\alpha) = \alpha^T\gamma\mathbf{K}\gamma^T\alpha - 2\alpha^T\mathbf{K}\hat{\beta^{\mathbf{n}}}$
Output:
   The learned encoding parameters $\alpha$;
1: From zeros parameters of $\alpha$, select the $k$-th parameter with the largest $\mid \partial f(\alpha)/\partial\alpha_k \mid$
   If $\partial f(\alpha)/\partial\alpha_k > \lambda$, set $\rho_k = -1$, $activeset = activeset \cup \{k\}$
   If $\partial f(\alpha)/\partial\alpha_k < -\lambda$, set $\rho_k = 1$, $activeset = activeset \cup \{k\}$
2: Feature-sign-search:
   Compute the solution to the unconstrained problem as: $\alpha_{new} = argmin_\alpha f(\alpha) + \lambda\rho^T\alpha$ with only the parameters corresponding to the active set.
   Perform a discrete line search on the closed line segment from $\alpha$ to $\alpha_{new}$
   Check the objective value at $\alpha_{new}$ and all points where any parameter changes sign.
   Update $\alpha_{new}$ to the point with the lowest objective values and update the $activeset$ and $\rho$.
3: Check the optimality conditions:
     (a) Optimality condition for nonzero parameters:
   s: $f(\alpha) + \lambda sign(\alpha) = 0$.
       If unsatisfied, go to Step 2.
     (b) Optimality condition for zero parameters: $\mid f(\alpha) \mid < \lambda$
       If unsatisfied, go to Step 1.
   else stop;
4: return the learned $\alpha$;
```

When $\alpha_{\mathbf{n}}$ is fixed, Problem 21 equals to:

$$min_{\gamma} \sum_{n=1}^{N} {\alpha_{\mathbf{n}}}^{T} \gamma \mathbf{K} \gamma^{T} \alpha_{\mathbf{n}} - 2 \times {\alpha_{\mathbf{n}}}^{T} \mathbf{K}\hat{\beta^{\mathbf{n}}} \tag{22}$$

Which can be optimized by gradient descent. When γ is fixed, Problem 21 equals to:

$$min_{\alpha_{\mathbf{n}}} \sum_{n=1}^{N} {\alpha_{\mathbf{n}}}^{T} \gamma \mathbf{K} \gamma^{T} \alpha_{\mathbf{n}} - 2 \times {\alpha_{\mathbf{n}}}^{T} \mathbf{K}\hat{\beta^{\mathbf{n}}} + \lambda \parallel \alpha_{\mathbf{n}} \parallel_1 \tag{23}$$

This can be optimized using the feature-sign-search strategy. Algorithm 1 gives the details of the feature-sign-search algorithm for solving Problem 23.

We alternatively search for the optimal γ and $\alpha_{\mathbf{n}}$ until the summed reconstruction error is below a threshold or a pre-defined number of iterations has been reached. The encoded parameters can then be used with max pooling for image representation which can be eventually applied for the classification task. The local features are encoded one by one. Algorithm 2 shows the flowchart of the proposed implicit codebook transfer method for local feature encoding.

Algorithm 2 The Proposed Implicit Codebook Transfer Method for Local Feature Encoding

```
Input:
    The local features $\mathbf{x}$, pre-learned codebooks $\mathbf{B_m}$,
    the threshold parameter $\theta$ and max iteration number $maxiter$;
Output:
    The learned transfer parameter $\gamma$ and encoding
    parameters $\alpha$;
1: for $iter = 1, 2, ..., maxiter$
2:     Find the optimal $\gamma$ with encoding parameters
       $\alpha$ fixed by solving problem 22 with gradient descent;
3:     Find the optimal encoding parameters $\alpha$ with
       $\gamma$ fixed by solving problem 23 with the feature-sign-search in Algorithm 1;
4:     Check whether the decrease of objective function of problem 21 falls below the threshold $\theta$.
       If unsatisfied
         go to step 1
       Else
         stop;
5: return $\gamma$, $\alpha$;
```

C. Max Pooling Based Image Representation for Classification

After the encoding parameters of local features are learned, we can use them for image representation. We use the max pooling approach as it has been proven very effective when combined with sparsity constrained encoding. The max pooling strategy chooses the max absolute value of each dimension of encoding parameters as the image representation. Formally, let $\alpha_\mathbf{l}, l = 1, \ldots, L$ represent the parameters learned by implicit codebook transfer, where L is the number of local features within one image region. The max pooled image representation $\mathbf{h}$ is then calculated as:

$$h_i = max(| \alpha_{l,i} |, \ldots, | \alpha_{L,i} |) \tag{24}$$

Where i represents the i-th dimension of parameters. Usually, to combine the spatial information, spatial pyramid matching technique (SPM) [41] is used. We choose to use SPM with three pyramids (1×1, 2×2, 4×4) in this paper. This is achieved by concatenating the image representations of each region into a long vector.

To evaluate the effectiveness of the proposed method, we apply it for image classification using one-vs-all linear SVM classifiers. Suppose we have P training images of C classes $\{(\mathbf{h^p}, y^p)\}_{p=1}^{P}, y^p \in \{1, \ldots, C\}$. We want to learn C linear functions so that

$$y = max(\mathbf{w_c}^T \mathbf{h}) \tag{25}$$

By solving the following optimization problem as:

$$min_{\mathbf{w_c}} \parallel \mathbf{w_c} \parallel^2 + \epsilon \sum_{p=1}^{P} \ell(\mathbf{w_c}^T \mathbf{h^p}, y^p) \tag{26}$$

Using the quadratic hinge loss:

$$\ell(\mathbf{w_c}^T \mathbf{h^p}, y^p) = (max(0, \mathbf{w_c}^T \mathbf{h^p} y^p - 1))^2 \tag{27}$$

After the classifiers are trained we can use them to predict image categories using Eq. 24.

IV. Experiments

To evaluate the effectiveness of the proposed implicit codebook transfer for image representation method (ICT), we conduct classification of images on several public datasets, the Scene-15 dataset [41], the Caltech-101 dataset [42], the Caltech-256 dataset [43], the Flower-17 dataset [44], the Flower 102 dataset [45] and the MIT Indoor dataset [46]. The Scene-15 dataset has fifteen classes of 4,485 images. Each class has 200 to 400 images with the average image size of 300×250 pixels. The Caltech-101 dataset has 9,144 images of 101 classes. The number of images per class varies from 31 to 800. The Caltech-256 dataset has more classes of images than the Caltech-101 dataset. It has 29,780 images of 256 classes with varied poses. The flower-17 dataset has 17 classes of images while its extended version has 102 flower categories with 40 to 258 images per class. The MIT Indoor dataset has 15,620 images of 67 indoor scenes with the minimum image of 200 pixels. Figure 2 shows some example images of these datasets. We densely choose local regions with overlap. The local region size is set to be 16×16 pixels with an overlap of 6 pixels. The SIFT feature and PCA-SIFT feature are extracted from these regions respectively. We use sparse coding to learn the initial codebooks. The codebook sizes for the six datasets are all set to 1,000. Spatial pyramids with three scales (1×1, 2×2, 4×4) are used to combine the spatial information of local features. The max iteration number $maxiter$ is set to 50. We use the mean of per-class classification rate for performance evaluation.

A. Transfer Codebooks of Similar Datasets

We first transfer codebooks between similar datasets. We try to transfer the codebook generated on the Caltech-101/ Caltech-256 dataset for classification on the Caltech-256/ Caltech-101 dataset respectively. For the Caltech-101 dataset, we randomly select 1, 5, 15, 30 images per class for training and use the rest of images for testing. For the Caltech-256 dataset, we randomly select 1, 5, 15, 30, 45 images per class for training and use the rest of images for testing.

For fair comparison, we choose to compare with the performances reported by other methods. Table 2 gives the performance comparisons on the Caltech-101 dataset using SIFT features. This is achieved by transferring the codebook generated by the Caltech-256 dataset. We also give the performance of directly using the codebook generated by the Caltech-256 dataset (abbreviated as ScSPM (Caltech-256)). We can have four conclusions from Table 2. First, the proposed ICT method can help to transfer useful information for improving classification performance, especially when we only have a limited number of training images. The proposed ICT outperforms SVM-KNN by about 11 percent when only one training image per class is used. The codebook generated

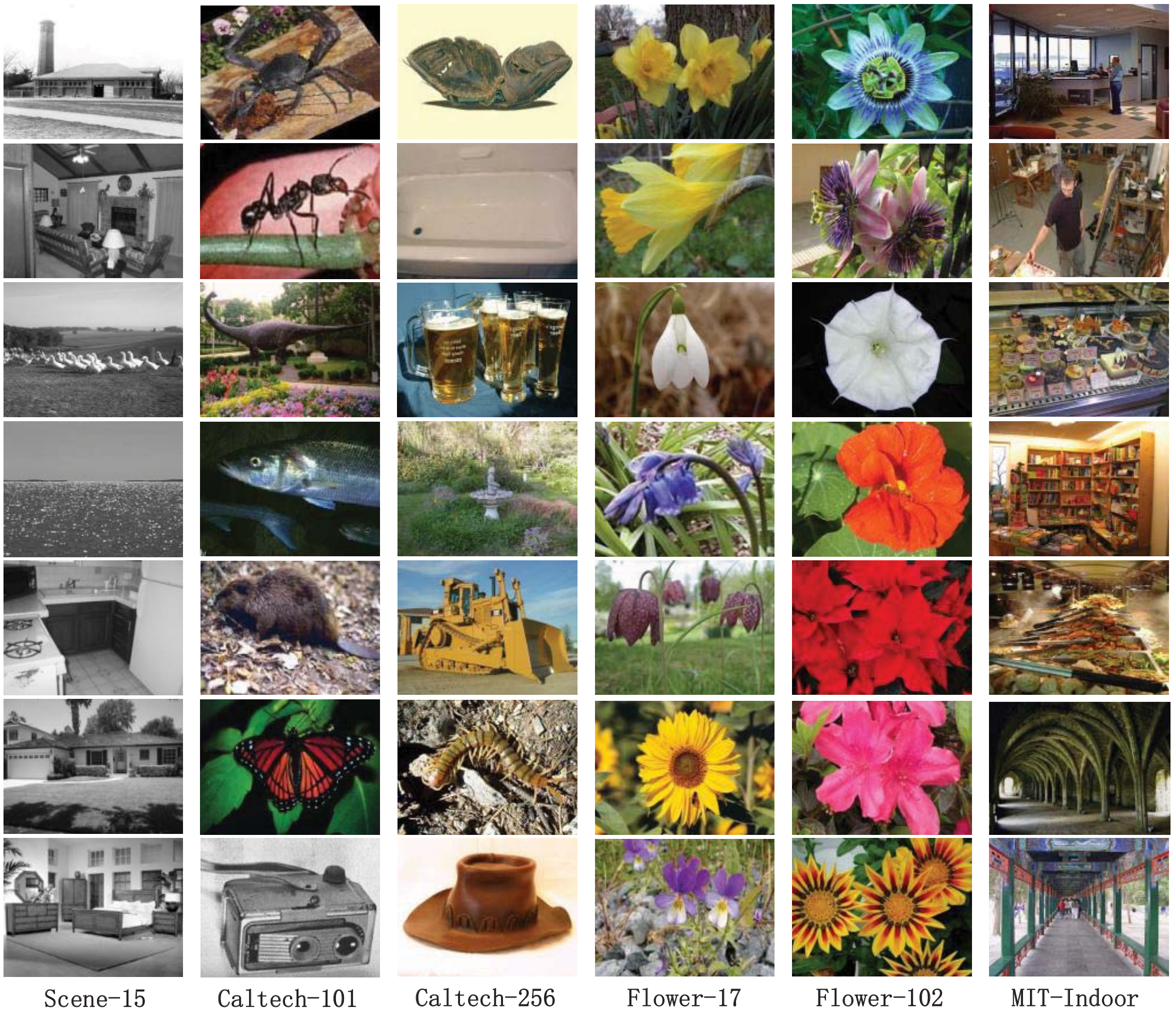


Fig. 2. Example images of the Scene-15 dataset, the Caltech-101 dataset, the Caltech-256 dataset, the Flower-17 dataset, the Flower 102 dataset and the MIT Indoor dataset.

using only one training image is probably biased and cannot be able to represent images well. However, by transferring the pre-learned codebook learned from similar datasets, we are able to alleviate this problem to some extent. Second, the usage of non-linear transfer can map the pre-learned codebook more effectively than linear codebook transfer, hence, is able to outperform linear transfer method. Third, by using soft-assignment based strategy, we are able to improve the representation efficiency over hard-assignment methods. The proposed ICT is able to outperform SPM [42] which uses k-means clustering and nearest neighbor assignment by about 9 percent when 30 images per class are used. Fourth, directly using the codebooks generated by other datasets is not able to represent new images well. Hence, transferring the pre-learned codebooks is necessary.

Besides, the proposed ICT can also cope with codebooks generated with different types of local features. We also give the re-implemented sparse coding based method and the implicit codebook transfer method on the Caltech-101 dataset using PCA-SIFT in Table 2. This is achieved by transferring the codebook generated with PCA-SIFT on the Caltech-256 dataset to encode the SIFT features of Caltech-101 dataset. We can see from Table 2 that because PCA-SIFT and SIFT features have different dimensions and reflect different properties of local regions, the transferring of PCA-SIFT for SIFT feature encoding does not perform as good as only using SIFT features. However, this can be alleviated with the increment of training images as more and more information can be used. This also means when applied for online codebook generation, the proposed method can gradually adapt to new applications with the arrival of images.

The convolutional network can also be used to transfer information between image datasets for recognition. As Yosinski *et al.* [48] found, the performance is relatively unchanged if we fix the first two layers when transferring between datasets for classification. We follow this setup and try to transfer the Caltech-256 dataset for the recognition of the Caltech-101 dataset. The performance is 81.3% which is better than the proposed method. We believe this is for two reasons. First, by operating on the image pixels directly, the convolutional network can learn more complex and useful features than SIFT features which is extracted with gray images. Second, the transferred information is relatively more compared with codebook based method as back-propagation

TABLE II

IMAGE CLASSIFICATION ACCURACY COMPARISON ON THE CALTECH-101 DATASET, WE TRANSFER THE CODEBOOK GENERATED ON THE CALTECH-256 DATASET FOR CLASSIFICATION ON THE CALTECH-101 DATASET. (SPM: SPATIAL PYRAMID MATCHING; KC: KERNEL CODEBOOK; ScSPM: SPARSE CODING ALONG WITH SPATIAL PYRAMID MATCHING; NBNN: NAIVE-BAYES NEAREST-NEIGHBOR; SVM-KNN: SUPPORT VECTOR MACHINE WITH K NEAREST NEIGHBOR; LCT: LINEAR CODEBOOK TRANSFER; ICT: THE PROPOSED METHOD)

Algorithm	1 train	5 train	15 train	30 train
SPM [42]	–	–	56.4	64.4
KC [5]	–	–	–	64.2
ScSPM [2]	–	51.2	65.4	73.4
NBNN [31]	25.7	49.5	64.5	72.9
SVM-KNN [47]	22.1	45.9	58.8	66.3
LCT [17]	27.3	50.7	64.9	73.0
ScSPM (Caltech-256)	24.7	47.8	64.5	72.3
ICT	33.5	53.6	66.2	73.5
ScSPM (PCA-SIFT)	21.8	44.3	60.7	68.2
ICT (PCA-SIFT)	26.4	49.1	63.9	71.6

TABLE III

IMAGE CLASSIFICATION ACCURACY COMPARISON ON THE CALTECH-256 DATASET, WE TRANSFER THE CODEBOOK GENERATED ON THE CALTECH-101 DATASET FOR CLASSIFICATION ON THE CALTECH-256 DATASET

Algorithm	1 train	5 train	15 train	30 train	45 train
SPM [42]	–	18.6	28.3	34.1	–
KC [5]	–	–	–	27.2	–
ScSPM [2]	–	–	27.7	34.0	37.5
LLC [6]	–	–	34.4	41.2	45.3
LScSPM [7]	–	–	30.0	35.7	38.5
NBNN [31]	8.0	19.8	30.6	38.0	–
LCT [17]	12.7	20.8	27.3	34.5	37.4
FV [22]	–	–	38.5	47.4	52.1
KSRSPM-HIK [51]	–	–	33.6	40.6	44.4
ScSPM (Caltech-101)	8.3	14.5	21.8	28.5	33.6
ICT	15.6	22.5	31.2	36.9	39.3
ICT-combined	–	–	36.9	43.1	46.3
ICT-LLC	–	–	33.5	40.7	45.1
ICT-LSc	–	–	31.9	37.2	39.2
ICT-FV	–	–	36.7	46.1	51.2

can help spread information through layers. To fully explore the discriminative power of the proposed method, we also extract color SIFT features [49] and combine them by multiple kernel learning [50]. In this way, we can boost the performance to 78.5%.

To systematically evaluate the proposed ICT method, we give the performance comparison on the Caltech-256 dataset by transferring the codebook generated by the Caltech-101 dataset in Table 3 using SIFT features. We also give the performance of directly using the codebook generated by the Caltech-101 dataset (abbreviated as ScSPM (Caltech-101)). We can have similar conclusions as from Table 2. The proposed method is able to transfer codebooks generated by other datasets for new image representation which eventually helps the classification task. However, there are still differences between Table 3 and Table 2. The codebook generated with the Caltech-101 dataset cannot be able to represent images of the Caltech-256 dataset well. This can be seen from Table 3 that the relative improvements of ICT over LCT and other methods are not as significant as in Table 2. We need more training images to adjust the pre-learned codebooks for new images representation which are harder to classify than the original images. However, ICT can gradually adapt to the new application with the increase of training images as more and more information is used. Besides, directly using codebook generated by the Caltech-101 dataset for Caltech-256 dataset classification does not perform as good as using codebook generate by the Caltech-256 dataset for Caltech-101 dataset classification. This is because Caltech-256 dataset has more classes of images and is more difficult to represent and classify compared with the Caltech-101 dataset. To intuitively illustrate this, we give some example images misclassified using the codebook generated with the Caltech-101 dataset but correctly predicted using the proposed codebook transfer method on the Caltech-256 dataset in Figure 3. We can see from Figure 3

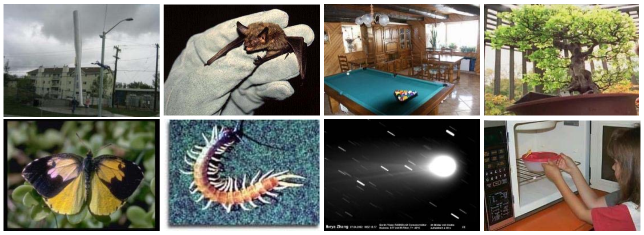


Fig. 3. Example images misclassified using the codebook generated on the Caltech-101 dataset but correctly predicted using the proposed method on the Caltech-256 dataset.

TABLE IV

PERFORMANCE COMPARISON ON THE FLOWER-17 DATASET, WE TRANSFER THE CODEBOOK GENERATED ON THE FLOWER-102 DATASET FOR CLASSIFICATION ON THE FLOWER-17 DATASET

Algorithm	Performance
Nilsback [44]	71.76 ± 1.76
Varma [52]	82.55 ± 0.34
Xie [53]	87.45 ± 1.13
KMTJSRC-CG [54]	88.90 ± 2.30
ScSPM (Flower-102)	82.53 ± 1.49
ICT	91.37 ± 0.72

that most of these images are from the classes which only exist in the Caltech-256 dataset. This proves the usefulness of codebook transfer for image representation and recognition.

The proposed method can also be combined with other codebook generation and local feature encoding techniques (e.g. LLC [6] and Laplacian Sparse coding [7]) to further improve the performance. We give the performances by combination of the transferred codebook with these techniques on the Caltech-256 dataset in Table 3 (ICT-LLC and ICT-LSc). By adapting the locality and smooth constraints, we can further improve the performance with the implicitly transferred codebook. Besides, with the increase of training images, the relative improvement decreases. This is because with more training images, we can learn more reliable classifiers. The proposed method is generic and can be combined with other codebook based methods for better image representation and help to improve the classification performance. The proposed codebook based method is not able to perform as well as Fisher Vector (FV) [22] based method which uses the first and second order information during local feature encoding process. If we only have pre-learned codebooks, we are not able to recover enough information as FV hence performs not as good as [22]. As a rough comparison, we use the pre-learned codebook as the initial mean values in the FV method and use the new training images to learn the corresponding Fisher Vector based image representation (ICT-FV). In this way, we can achieve comparable classification rate as [22]. The proposed method performs not as well as KSRSPM-HIK [51] which achieves 33.6/40.6/44.4 percent 15/30/45 training images respectively. KSRSPM-HIK [51] mapped local features by combining sparse representation with kernels which proves the effectiveness of using non-linear transformation over linear mapping. Although both KSRSPM and ICT use kernels for representation, they are targeted at different tasks. KSRSPM uses the kernels to map local features to high dimensional space where local features are encoded. KSRSPM has same source domain and target domain. ICT tries to adapt the learned codebooks of source domain for classification in the target domain. To fully explore the discriminative power of the proposed ICT method, we also extract color SIFT features (HueSIFT, HSV-SIFT, OpponentSIFT, RGB-SIFT and C-SIFT) as [49] did and transfer the codebooks for classification (ICT-combined) on the Caltech-256 dataset. This improves the classification performances to 36.9/43.1/46.3 percent for 15/30/45 training images respectively.

TABLE V

PERFORMANCE COMPARISON ON THE FLOWER-102 DATASET, WE TRANSFER THE CODEBOOK GENERATED ON THE FLOWER-17 DATASET FOR CLASSIFICATION ON THE FLOWER-102 DATASET

Algorithm	Performance
Nilsback [44]	72.8
KMTJSRC-CG [54]	74.1
ScSPM (Flower-17)	68.5
ICT	77.3

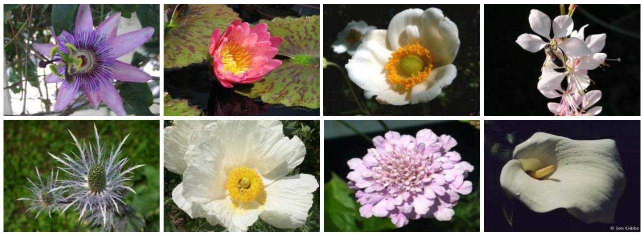


Fig. 4. Example images misclassified using the codebook generated on the Flower-17 dataset but correctly predicted using the proposed method on the Flower-102 dataset.

TABLE VI

PERFORMANCE COMPARISON ON THE SCENE-15 DATASET, WE TRANSFER THE CODEBOOKS GENERATED ON THE CALTECH-256 DATASET AND THE MIT INDOOR DATASET FOR CLASSIFICATION ON THE SCENE-15 DATASET

Algorithm	Performance
SPM [41]	81.40 ± 0.50
KC [5]	76.67 ± 0.39
ScSPM [2]	80.28 ± 0.93
LCT (Caltech-256) [17]	76.46 ± 0.77
LCT (MIT Indoor) [17]	77.50 ± 0.83
ScSPM (Caltech-256)	73.64 ± 0.82
ScSPM (MIT Indoor)	72.48 ± 0.77
ICT (Caltech-256)	80.15 ± 0.64
ICT (MIT Indoor)	79.86 ± 0.55

B. Transfer Codebooks of Fine-Grained Datasets

To test the effectiveness of ICT for fine-grained images, we also transfer the codebooks between the Flower-17 dataset and the Flower-102 dataset. The images of the two datasets are of large inter-class variation which makes them hard to classify. Since color information plays an important role for flower image representation, we first transform the images to different color spaces as [49] did and then extract the corresponding local features. For the Flower-17 dataset, we follow the image splits provided by [44] and use 40, 20, 20 images for training, validation and testing respectively. For the Flower-102 dataset, we use 10, 10 images for training and validation with the rest of images used for testing, as [45] did.

We give the classification comparison on the Flower-17 dataset in Table 4 with other methods [44], [52]–[54]. Varma and Ray [52] tried to learn the discriminative features while Xie *et al.* [53] used the χ^2 kernel for similarity measurement. Yuan and Yan [54] used sparse reconstruction method with multi-type of features. We also give the performance of directly using the codebook generated by

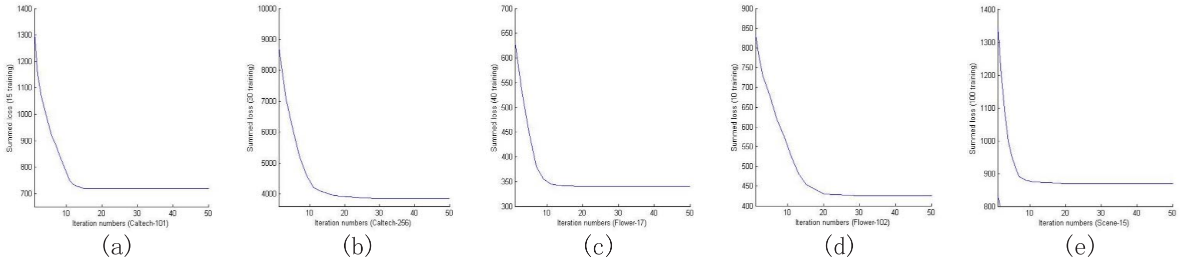


Fig. 5. The objective value changes of the above five experiments with the increase of iterations. (a) Caltech-101. (b) Caltech-256. (c) Flower-17. (d) Flower-102. (e) Scene-15.

the Flower-102 dataset (abbreviated as ScSPM (Flower-102)). From Table 4 we can see that directly using the codebok generated with the Flower-102 dataset cannot represent images of the Flower-17 very well while transferring the codebook by minimizing reconstruction error with sparsity constraints on the Flower-17 dataset helps to alleviate this problem. Besides, ICT outperforms [44] which proves the effectiveness of softly encode local features instead of hard assignment. Moreover, the proposed method also outperforms [52]–[54] to some extent. This again shows the feasibility of transferring pre-learned codebooks for fine-grained visual applications.

Similar as on the Caltech datasets, we also give the classification performance comparison on the Flower-102 dataset in Table 5. We can see that directly using the codebook of the Flower-17 dataset performs worse than using the Flower-102 dataset's codebook [44]. The performance is improved by more than 8 percent through implicit codebook transfer. We can see from Table 4 and Table 5 that the proposed implicit codebook transfer method can also cope with fine-grained image representation. We also give some example images misclassified using the codebook generated on the Flower-17 dataset but correctly predicted using the proposed codebook transfer method on the Flower-102 dataset in Figure 4. Since Flower-102 dataset has more images of different classes than the Flower-17 dataset, directly using the codebook generated with the Flower-17 dataset is not able to represent images well. However, by transferring the codebook with new images, we are able to improve the recognition performance.

C. Transfer Codebooks of Dissimilar Datasets

To fully evaluate the effectiveness of the proposed method, we apply it for transferring the codebooks among dissimilar datasets. Specifically, we transfer the codebooks generated by the Caltech-256 dataset and the MIT Indoor dataset for image classification on the Scene-15 dataset. We follow the same experimental setup as [41] did and randomly select 100 images per class for training and use the rest of images for testing. This process is repeated ten times to get reliable results.

Table 6 gives the performance comparisons. We also give the classification accuracies of using the codebooks of the Caltech-256 dataset and the MIT Indoor dataset respectively. We can have three conclusions from Table 6. First, directly using codebooks of other datasets is not enough for accurate classification. Second, both linear and non-linear transfer of codebooks can help to alleviate the codebook and image dataset discrepancy to some extent. Third, the use of non-linear transfer can make use of the pre-learned codebooks more efficiently than linear transfer.

The combination of ICT with other local feature encoding methods can not outperform the corresponding methods (e.g. LLC) for three reasons. First, the aim of the proposed ICT method is to transfer pre-learned codebooks for new datasets without explicitly generating the codebook. For a particular image dataset, the codebook generated with the codebook's images is more representative than codebooks generated by other datasets. By using the ICT method, we can obtain superior performance than directly using codebooks of other datasets. This can be proven by the experimental results. Besides, although we use sparse coding in the paper, the proposed method can also be combined with other more efficient coding methods such as LLC. In this way, we can improve the performance accordingly. Another reason is the differences among the source datasets and the target datasets. The visual similarities among different datasets also influence the effectiveness of the transferring. As shown in the experimental section, the performance is relatively better when we transfer among similar datasets than among dissimilar datasets. It requires relatively more training samples to transfer among dissimilar datasets for reliable classification. Third, it is often very hard to re-implement other researcher's work with the same performances because of the re-implementation differences, such as the local feature extracting and normalization.

We can see from Table 2-6 that the proposed codebook transfer method is able to transfer useful information from pre-learned codebooks for image representation. Besides, to make better use of the codebook transfer technique, we should transfer the codebooks among similar datasets instead of dissimilar datasets. Transferring information among similar datasets means we can share the representation power more efficiently and help to separate images. Although the transfer of codebooks among dissimilar datasets is also plausible, it requires relatively more computational time and training images. Moreover, when we only have limited images, we should explore various types of information to improve the classification accuracy.

D. Convergence Analysis

Problem 21 is not convex with γ and α. It is convex when optimizing over γ/α while keeping α/γ fixed. Since we alternatively optimize over γ and α, the summed reconstruction error and sparsity constraints decreases for each iteration. Besides, the objective values of Problem 21 are always non-negative. Hence, Algorithm 2 converges. To intuitively show the convergence of the proposed method, we plot the summed exponential losses of the above five experiments in Figure 5. We can see from Figure 5 that the objective values gradually stop to decrease with the increase of iterations. Besides, The transfer of dissimilar datasets' codebooks takes relatively more iterations than similar datasets.

V. Conclusion

In this paper, we propose a novel implicit codebook transfer method for image representation and apply it for image classification task with good performance. By transferring the codebook with non-linear transformation, we are able to make better use of the pre-learned codebooks than linear transfer. Besides, it can also be applied to transfer codebooks generated with different types of local features. Moreover, the proposed implicit codebook transfer method can also be used in an online setting instead of re-training the codebooks with new images. Finally, we validate the proposed method on several public datasets for image classification task, experimental results prove the effectiveness and efficiency of the proposed method.

References

[1] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE ICCV*, Oct. 2003, pp. 1470–1477.

[2] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 1794–1801.

[3] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Comput. Surv.*, vol. 40, no. 2, 2008, Art. ID 5.

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and Z. Zisserman, "The PASCAL visual object classes (VOC) Challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[5] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J.-M. Geusebroek, "Visual word ambiguity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1271–1283, Jul. 2010.

[6] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 3360–3367.

[7] S. Gao, I. W.-H. Tsang, and L.-T. Chia, "Laplacian sparse coding, hypergraph Laplacian sparse coding, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 92–104, Jan. 2013.

[8] F. Perronnin, C. Dance, G. Csurka, and M. Bressan, "Adapted vocabularies for generic visual categorization," in *Proc. 9th ECCV*, 2006, pp. 464–475.

[9] S. Zhang, Q. Huang, G. Hua, S. Jiang, W. Gao, and Q. Tian, "Building contextual visual vocabulary for large-scale image applications," in *Proc. Int. Conf. Multimedia*, 2010, pp. 501–510.

[10] K. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *Proc. 10th IEEE ICCV*, Oct. 2005, pp. 1800–1807.

[11] J. Mũnoz-Marí, F. Bovolo, L. Gomez-Chova, L. Bruzzone, and G. Camp-Valls, "Semisupervised one-class support vector machines for classification of remote sensing data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 8, pp. 3188–3197, Aug. 2010.

[12] M. Guillaumin, J. Verbeek, and C. Schmid, "Multimodal semi-supervised learning for image classification," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 902–909.

[13] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 3501–3508.

[14] Q. Yang, V. W. Zheng, B. Li, and H. H. Zhou, "Transfer learning by reusing structured knowledge," *AI Mag.*, vol. 32, no. 2, pp. 95–106, 2011.

[15] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 10, pp. 1345–1359, Oct. 2010.

[16] S. Gao, I. W.-H. Tsang, and Y. Ma, "Learning category-specific dictionary and shared dictionary for fine-grained image categorization," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 623–634, Feb. 2014.

[17] C. Zhang *et al.*, "Undoing the codebook bias by linear transformation with sparsity and F-norm constraints for image classification," *Pattern Recognit. Lett.*, vol. 45, pp. 197–204, Aug. 2014.

[18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[19] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. IEEE Comput. Soc. Conf. CVPR*, Jun./Jul. 2004, pp. II-506–II-513.

[20] N. Dalal and W. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. CVPR*, Jun. 2005, pp. 886–893.

[21] C. Zhang, S. Wang, Q. Huang, J. Liu, C. Liang, and Q. Tian, "Image classification using spatial pyramid robust sparse coding," *Pattern Recognit. Lett.*, vol. 34, no. 9, pp. 1046–1052, 2013.

[22] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *Int. J. Comput. Vis.*, vol. 105, no. 3, pp. 222–245, 2013.

[23] T. Kobayashi, "Dirichlet-based histogram feature transform for image classification," in *Proc. IEEE Conf. CVPR*, Jun. 2014, pp. 3278–3285.

[24] C. M. Bishop, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford Univ. Press, 1995.

[25] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[26] S. Z. Li, *Markov Random Field Modeling in Image Analysis*. London, U.K.: Springer-Verlag, 2009.

[27] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *Ann. Statist.*, vol. 36, no. 3, pp. 1171–1220, 2008.

[28] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 465–479, Mar. 2012.

[29] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. ID 27.

[30] L. Yang, R. Jin, R. Sukthankar, and F. Jurie, "Unifying discriminative visual codebook generation with classifier training for object category recognition," in *Proc. IEEE Conf. CVPR*, Jun. 2008, pp. 1–8.

[31] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Proc. IEEE Conf. CVPR*, Jun. 2008, pp. 1–8.

[32] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2014.2321376.

[33] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proc. IEEE Int. Symp. Circuits Syst.*, May/Jun. 2010, pp. 253–256.

[34] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 2528–2535.

[35] L. Shao, L. Liu, and X. Li, "Feature learning for image classification via multiobjective genetic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 7, pp. 1359–1371, Jul. 2014.

[36] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 2066–2073.

[37] J. Ni, Q. Qiu, and R. Chellappa, "Subspace interpolation via dictionary learning for unsupervised domain adaptation," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 692–699.

[38] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proc. IEEE ICCV*, Dec. 2013, pp. 2960–2967.

[39] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 361–368.

[40] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," in *Proc. NIPS*, 2007, pp. 801–808.

[41] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. CVPR*, Jun. 2006, pp. 2169–2178.

[42] L.-J. Li and L. Fei-Fei, "what, where and who? Classifying events by scene and object recognition," in *Proc. IEEE 11th ICCV*, Oct. 2007, pp. 1–8.

[43] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," Dept. Comput. Sci., CalTech, Pasadena, CA, USA, Tech. Rep. no. 1, 2007.

[44] M. E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *Proc. IEEE Conf. CVPR*, Jun. 2006, pp. 1447–1454.

[45] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proc. 6th ICVGIP*, 2008, pp. 722–729.

[46] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proc. IEEE Conf. CVPR*, Miami, FL, USA, Jun. 2009, pp. 413–420.

[47] H. Zhang, A. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition," in *Proc. IEEE Conf. CVPR*, Jun. 2006, pp. 2126–2136.

[48] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. NIPS*, 2014, pp. 1–9.

[49] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1582–1596, Sep. 2010.

[50] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proc. 21st Int. Conf. ICML*, 2004, p. 6.

[51] S. Gao, I. W. Tsang, and L.-T. Chia, "Sparse representation with kernels," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 423–434, Feb. 2013.

[52] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in *Proc. IEEE 11th ICCV*, Oct. 2007, pp. 1–8.

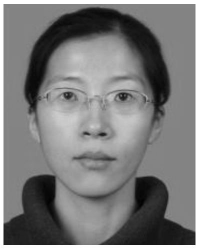
[53] N. Xie, H. Ling, W. Hu, and X. Zhang, "Use bin-ratio information for category and scene classification," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 2313–2319.

[54] X.-T. Yuan and S. Yan, "Visual classification with multi-task joint sparse representation," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 3493–3500.

Chunjie Zhang received the B.E. degree from the Nanjing University of Posts and Telecommunications, Jiangsu, China, in 2006, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, China, in 2011. He was an Engineer with the Henan Electric Power Research Institute from 2011 to 2012. He was a Post-Doctoral Fellow with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China. He is currently an Assistant Professor with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China.

His current research interests include image processing, machine learning, cross media content analysis, pattern recognition, and computer vision.

Jian Cheng received the B.S. and M.S. degrees from Wuhan University, in 1998 and 2001, respectively, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2004. From 2004 to 2006, he was a Post-Doctoral Fellow with the Nokia Research Center, China. He has been with the National Laboratory of Pattern Recognition since 2006. His current research interests include machine learning methods and their applications for image processing and social network analysis.



Jing Liu received the B.E. and M.E. degrees from Shandong University, Shandong, in 2001 and 2004, respectively, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2008. She is currently an Associate Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. Her current research interests include multimedia analysis, understanding, and retrieval.

Junbiao Pang received the B.S. and M.S. degrees in computational fluid dynamics and computer science from the Harbin Institute of Technology, Harbin, China, in 2002 and 2004, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2011. From 2004 to 2005, he was a Software Engineering with Huawei Technologies Company, Ltd.

He is currently a Faculty Member with the College of Computer Sciences, Beijing University of Technology, Beijing. His research areas include computer vision, multimedia, and machine learning.

Qingming Huang received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1994. He was a Post-Doctoral Fellow with the National University of Singapore from 1995 to 1996, and a member of the Research Staff with the Institute for Infocomm Research, Singapore, from 1996 to 2002. He joined the Chinese Academy of Sciences, Beijing, China, in 2003, and is currently a Professor with the University of Chinese Academy of Sciences. His current research areas are image and video analysis, video coding, pattern recognition, and computer vision.

Qi Tian (SM'03) is currently a Full Professor with the Department of Computer Science, The University of Texas at San Antonio (UTSA). He was a tenured Associate Professor from 2008-2012 and a tenure-track Assistant Professor from 2002-2008. During 2008-2009, he took one-year Faculty Leave at Microsoft Research Asia (MSRA) as a Lead Researcher with the Media Computing Group. He received his Ph.D. degree in ECE from University of Illinois at Urbana-Champaign (UIUC) in 2002, and B.E. degree in Electronic Engineering from Tsinghua University in 1992, and M.S. in ECE from Drexel University in 1996, respectively. His research interests include multimedia information retrieval, computer vision, pattern recognition, and bioinformatics.

Dr. Tian research projects are funded by ARO, NSF, DHS, Google, FXPAL, NEC, SALSI, CIAS, Akiira Media Systems, HP, and UTSA. He received 2014 Research Achievement Awards from the College of Science, UTSA. He received the 2010 ACM Service Award. He is a Member of ACM (2004).