Image Annotation by Latent Community Detection and Multikernel Learning

Yun Gu, *Student Member, IEEE*, Xueming Qian, *Member, IEEE*, Qing Li, Meng Wang, *Member, IEEE*, Richang Hong, *Member, IEEE*, and Qi Tian, *Senior Member, IEEE*

***Abstract*— Automatic image annotation is an attractive service for users and administrators of online photo sharing websites. In this paper, we propose an image annotation approach that exploits latent semantic community of labels and multikernel learning (LCMKL). First, a concept graph is constructed for labels indicating the relationship between the concepts. Based on the concept graph, semantic communities are explored using an automatic community detection method. For an image to be annotated, a multikernel support vector machine is used to determine the image's latent community from its visual features. Then, a candidate label ranking based approach is determined by intracommunity and intercommunity ranking. Experiments on the NUS-WIDE database and IAPR TC-12 data set demonstrate that LCMKL outperforms some state-of-the-art approaches.**

***Index Terms*— Image annotation, multiple-kernel learning, concept graph, community detection.**

I. Introduction

With the explosive growth of web images, image annotation, which is beneficial to information management, has attracted considerable attention in recent years. Given an image, the goal of image annotation is to analyze its visual content and assign labels to it. Different from traditional classification problems, the number of labels is quite large and label co-occurrence is fairly common in image annotation. For example, it is highly likely that an image associated with the concept 'sea' will also contain the concept 'sky'.

In recent years, great research effort has been devoted to automatic image annotation [1]–[11]. In general, approaches for image annotation can be classified into two categories: learning-based and search-based annotation [10]. In search-based annotation, the labels are directly provided and annotated by utilizing images in the database. The k-nearest neighbor (KNN) search (including the extended algorithms) is widely used because of its simplicity and good performance with large scale data [6], [8], [10], [12], [13]. When using KNN to annotate images, we find the nearest images in the training set and label the target image according to the labels of its neighbors. However, two issues must be considered. One is ignorance of label co-occurrence, which leads to low precision. Previous works [14]–[18] have shown that co-occurrence plays a significant role in improving precision. Admittedly, if a picture contains a concept (i.e., labels/tags) like 'sunshine' or 'sea', it is very likely to include the concepts 'boat', 'sky', and so on. The other issue is the large size of the dataset, which leads to low efficiency of KNN.

For learning-based methods, the annotation problem can be considered a multi-class classification that predicts one label from a set of exclusive labels, or a binary classification that makes a binary decision on each label independently. In previous work, researchers applied machine learning methods such as the support vector machine (SVM) to the annotation problem [19]–[22] and showed its good performance with high dimensional data. In traditional image annotation problems, the number of classes or labels is always limited and samples of each class are often uniform. This can be considered as a classification problem. However, there are more than hundreds of labels (even millions) in an online image dataset like Flickr. Since each image can be tagged with many labels, this problem is no longer compatible with a traditional classification model.

Recent works have found that community detection achieves great success in social networks [23]–[26]. Papadopoulos *et al.* [26] presented an image clustering method on a hybrid image similarity graph exploiting both visual and textual features. We found that the connections between labels are similar to a social network. In the same way that Latent Dirichlet Allocation (LDA) [27], [28] finds topics in a bag of words, the community detection approach divides the labels into several communities. These communities reflect clustering coherence well. More importantly, the number of communities is much smaller than the number of labels thereby decreasing the complexity of learning. In this paper, community detection

Manuscript received November 5, 2014; revised April 1, 2015; accepted May 30, 2015. Date of publication June 10, 2015; date of current version July 7, 2015. This work was supported in part by the 973 Program under Grant 2012CB316400, in part by the National Natural Science Foundation of China under Grant 61173109, Grant 61202180, Grant 60903121, Grant 61332018, Grant 61429201, and ARO Grant W911NF-12-1-0057, and in part by Microsoft Research. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jianfei Cai. *(Corresponding author: Xueming Qian.)*

Y. Gu was with the SMILES Laboratory, Xi'an Jiaotong University, Xi'an 710049, China. He is now with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: geron762@sjtu.edu.cn).

X. Qian is with the Key Laboratory for Intelligent Networks and Network Security Ministry of Education, Xi'an Jiaotong University, Xi'an 710049, China, and also with the SMILES Laboratory, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: qianxm@mail.xjtu.edu.cn).

Q. Li was with the SMILES Laboratory, Xi'an Jiaotong University, Xi'an 710049, China. He is now with the University of Wisconsin–Madison, Madison, WI 53706 USA (e-mail: qli239@wisc.edu).

M. Wang and R. Hong are with the Hefei University of Technology, Hefei 230000, China (e-mail: eric.mengwang@gmail.com;hongrc.hfut@gmail.com).

Q. Tian is with The 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.2015.2443501

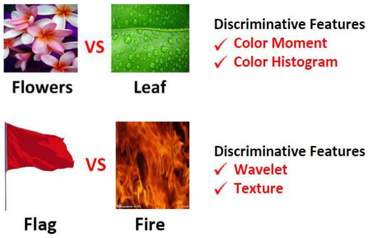

Fig. 1. For classification of 'leaf' and 'flowers', color moments and a color histogram are more discriminative features than the wavelet feature. However, since the colors of 'flag' and 'fire' are quite similar, texture features are more discriminative.

is adopted to explore the latent semantics between labels. When applying the community detection method to a multi-label annotation problem, we should classify each input image into only one most likely community (and sometimes two communities) instead of multiple concepts. In this way, community detection not only reduces the time complexity, but also allows machine learning methods to be applied to a multi-label annotation problem.

Another critical problem in image annotation today is the diversity of the visual content of images, which leads to poor performance of traditional visual representation. For example, color histogram features play a more significant role than the edge detection histogram and wavelet texture in discriminating the two classes 'flower' and 'leaf' as shown in Fig. 1. Therefore, adaptive selection of representative features is important. Sonnenburg *et al.* [29] proposed a multiple-kernel SVM (MKL-SVM) that assigns different weights to multiple features for classification.

In this paper, we propose an image annotation method using latent semantic community of labels and multi-kernel learning (LCMKL). It is a general framework composed with community detection, community classification and intra/inter-community annotation. Given training samples, a concept graph is first constructed with tagging information. Then, concept communities are detected from the concept graph. A community classifier is trained using a multiple-kernel SVM based on the concept communities. For an untagged image, the corresponding community is first determined by the community classifier. Then, intra-community annotation using a KNN is performed with training samples according to the result of the community classification. As discussed above, the MKL-SVM guarantees high accuracy classification utilizing various features. Compared with traditional KNN, the time complexity of the proposed method can be reduced through feature selection by the MKL-SVM. Inter-community annotation is finally carried out to provide complementary image annotation.

The main contributions of our work are as follows:

- We propose a general framework exploiting latent community detection for image annotation based on posterior probability and introduce the latent community concept.
- For community classification, we use an MKL-SVM in the image annotation problem instead of an SVM. Multiple features can be adaptively assigned weights for better representation. Higher classification accuracy guarantees that the following step can be implemented successfully. It can be altered with stronger feature representation and classifiers (e.g. Deep Learning features) for better performance.
- To infer more relevant labels and avoid the negative effect of hard community classification, we introduce inter-community annotation to assign additional labels. Thus, although some pictures may be classified into incorrect clusters, they could still be labeled with the correct tags after inter-community detection.

Compared with our preliminary work [30], several improvements are made: 1) detailed steps for the LCMKL are provided; 2) more experimental results and discussions are provided including evaluation on the NUS-WIDE and IAPR TC-12 datasets; and 3) systematic comparisons between the proposed method and other methods are given.

The rest of the paper is organized as follows. Section II reviews related work on image annotation, community detection, and MKL-SVMs. The framework adopted in this study is presented in Section III. Offline learning and online annotation are introduced in Sections IV and V, respectively. Experiments and discussions are given in Section VII. Finally, we give our conclusion in Section VIII.

II. Related Work

Recently, numerous approaches have been proposed for automatic image annotation. Makadia *et al.* [1] presented a baseline for image annotation. For search-based annotation methods, KNN is extensively used to annotate images with labels. Given an untagged image, Zhang and Zhou [12] proposed ML-KNN to annotate concepts. Based on statistical information acquired from neighboring instances, the labels of a given image can be determined. Besides, considering the dependencies between labels, Younes *et al.* [31] proposed a Bayesian version of KNN for multi-label classification. In addition, to achieve greater accuracy, many works have focused on ameliorating KNN [6], [10], [12], [13], [32], [33]. Tang *et al.* [6] proposed a KNN sparse-graph based semi-supervised learning approach, while Wang *et al.* [32] presented an image annotation approach based on weighted KNN. Based on Wang's work, Yu *et al.* [33] proposed a neighborhood rough set based multi-label classification. The aforementioned methods based on KNN fully consider the statistical information and express its simplification and efficiency. However, the accuracy precision is largely determined by the image set. Much training sample information has not been mined. Besides, these methods do not consider the latent community of concepts, which lacks the association function. As is commonly agreed, if we see the concept 'sea' in a part of a given image, we would surmise that the image also contains the concept 'sky', as determined by the associative function of our brain [34].

Apart from the search-based approaches mentioned above, several learning-based methods have also been proposed for image annotation. Considering the similarity of image annotation and classification, some researchers have adopted classification algorithms. However, compared with traditional single-label classification that assigns an object to exactly one class, the multi-label classification method should be able to assign an image to one or more classes. Thus, Elisseeff and Weston [35] proposed a multi-label SVM to handle this problem. Zhang *et al.* [36] proposed a multi-label naïve Bayes classification approach and gave the feature selection. Thabtah *et al.* [37] introduced a new associative classification approach called multi-class, multi-label associative classification, while Zhang [38] proposed the LIFT approach, which constructs features specific to each label. These methods usually consider each concept as a class [36]–[38] and label the given image according to a confidence value. Compared with KNN, these approaches take full advantage of the training samples. However, since the connection between each pair of concepts is ignored, more time is spent on the image annotation problem.

Several authors [39]–[41] focused on multi-label, multi-instance image annotation, which annotates a specific region of an image precisely using corresponding tags. In [39], correlations between textual concepts and visual features are exploited for both global and local features. A structural max-margin model was proposed to formulate the classifier. In [39], the former work was extended by deploying multiple-kernel learning approaches. Specific kernels are additionally learned for pairs of inter-correlated labels. For new data points, the similarity with training sets can be computed by learning the eigen functions of kernels in online mode. In [39], multi-label, multi-instance image annotation was studied in multiple modals. The image topic determined from the surrounding context is considered in combination with traditional visual features and textual features, while LDA is used to formulate the classification model.

The focus in [39] and [42] is on the tag relevance estimation problem. In [39], untagged images are assigned labels through voting by the visual and tag relevant neighbors. Li and Snoek [42] studied sample selection; positive samples were selected through tag relevance, while negative samples were selected by negative bootstrap instead of random selection. In [43], an image annotation approach is proposed to select some relevant tags with diverse semantics. More recently, to reduce the time required or increase accuracy, various researchers focused on feature selection or sample selection in image annotation [7], [9], [10]. Tang *et al.* presented semantic-gap-oriented active learning for multi-label image annotation and used an active learning method to create a sample selection [4]. Furthermore, Ma *et al.* [9] used subspace-sparsity collaborated feature selection to reduce noisy and redundant features. Liu *et al.* [10] introduced graph-based dimensionality reduction for KNN-based image annotation to solve the problems of high computational cost and difficulty in finding semantically similar images.

III. Main Framework

In this paper, we focus on the annotation problem in which an untagged image can be assigned multiple labels. Let $X = \{x_1, x_2, \ldots, x_n\}$ denote the image collection, where n is the size of the image set. For each image, several low-level features have been extracted. The features for image x_i are represented by vector $p_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,q}\}$, where $p_{i,j}$ is the j-th feature of image x_i, which is a vector with dimension related to the feature type. Some of the images have been annotated with tags from a concept (i.e., label or tag) set $C = \{c_1, c_2, \ldots, c_m\}$, where m is the number of concepts. The labels of an annotated image x_i can be represented by an m-dimensional binary vector $T_i = \{t_{i,1}, t_{i,2}, \ldots, t_{i,m}\}$. The elements of T_i represent the presence of tags in x_i. If image x_i is associated with concept c_j, the value of $t_{i,j}$ is 1, otherwise $t_{i,j} = 0$. The task of image annotation is to determine the binary vector T for untagged images based on the tagged ones.

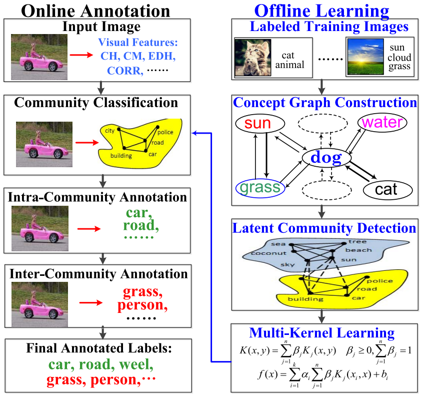
Online Annotation
Input Image
Visual Features: CH, CM, EDH, CORR, ……
Community Classification
city
police
road
building
car
Intra-Community Annotation
car, road, ……
Inter-Community Annotation
grass, person, ……
Final Annotated Labels:
car, road, weel, grass, person,⋯
Offline Learning
Labeled Training Images
cat animal
……
sun cloud grass
Concept Graph Construction
sun
water
dog
grass
cat
Latent Community Detection
sea
tree
beach
coconut
sky
sun
police
road
building
car
Multi-Kernel Learning
$K(x,y)=\sum_{j=1}^{n}\beta_j K_j(x,y)$ $\beta_j \geq 0, \sum_{j=1}^{n}\beta_j = 1$
$f(x)=\sum_{i=1}^{k}\alpha_i\sum_{j=1}^{n}\beta_j K_j(x_i,x)+b_i$

Fig. 2. Framework for the proposed LCMKL method consisting of two parts: offline learning and online annotation.

An image annotation method, called LCMKL, is proposed for image annotation by learning training samples based on latent community and multi-kernel learning. Fig. 2 illustrates our framework, which consists of two parts: offline learning and online annotation.

- **Offline Learning**: Given the labeled training samples, a concept graph is first created by exploiting the association between concepts. Then, concept communities are detected from the concept graph. Community classifiers are trained using a multiple-kernel SVM based on the visual features of training samples in each concept community.
- **Online Annotation**: First, the corresponding community of the untagged image is determined by the community classifier. Then, intra-community annotation is performed with training samples according to the result of the community classification. Inter-community annotation is finally carried out to provide complementary image annotation.

IV. Offline Learning

A. Concept Graph Construction

The first step in the proposed method is to construct a concept graph based on the tagged images. In multiple-labeling problems, co-occurrence of some concepts is common and notable. For example, the concepts of 'sky' and 'ocean' are simultaneously assigned to beach or seashore scenes. LCMKL was designed for multi-labeling annotation with sufficient concept co-occurrence in the dataset.

Therefore, a directed-weighted graph $G = \{V, E\}$ is constructed. The elements of vertex set V are tags from concept set $C = \{c_1, c_2, \ldots, c_m\}$. Concept c_i is connected with c_j by a directed edge e_{ij} if an image in the training set is tagged with c_i and c_j at the same time. Let w_{ij} denote the weight of e_{ij}. The weight of an edge implies the semantic correlation between the two concepts and is determined as follows, considering only their co-occurrence:

$$w_{i,j} = P(c_j|c_i) = \frac{N(c_i, c_j)}{N(c_i)} \tag{1}$$

where $P(c_j|c_i)$ is the conditional probability of concept c_j given c_i, $N(c_i)$ denotes the number of images tagged with concept c_i in the image collection, and $N(c_i, c_j)$ denotes the number of images tagged simultaneously with concepts c_i and c_j. It should be noted that $w_{i,j}$ and $w_{j,i}$ are usually not equivalent. This characteristic indicates the directionality of the concept co-occurrence. If $w_{i,j}$ and $w_{j,i}$ are similar and large enough, concepts c_i and c_j are likely to be annotated at the same time. If $w_{i,j}$ is much larger than $w_{j,i}$, the object represented by c_i may appear independently in different scenes, and not only together with c_j. For example, in the NUS-WIDE dataset [44], 4933 images are tagged with ‘grass’ and 19052 images are tagged with ‘sky’. The number of images tagged simultaneously with ‘sky’ and ‘grass’ is 3662. $P(‘sky’|‘grass’)$ and $P(‘grass’|‘sky’)$ are 0.733 and 0.193, respectively. The difference between the two probabilities is intuitive. In general, when an image is associated with ‘grass’, it is often related to an outdoor scene with blue sky and wide-open grassland. Conversely, the concept ‘sky’ may appear in other scenes like urban views or coastal landscapes, which are not necessarily associated with ‘grass’.

For simplicity, in this section, the concept graph is modeled only on the textual space. In our experiments, both visual and textual information is taken into account in the construction of the concept graph. Please refer to Section VII for the details.

B. Latent Community Detection

Traditional approaches for image annotation often adopt binary classifications, e.g., ‘SVM + features’ to make a binary decision independently for each label. However, this approach does not capture the complexity of semantic labels in the real world, where consistency between labels is ignored. In addition, the number of classifiers can be quite large since the semantic descriptions for images are rich and diverse. Therefore, we adopt community detection to solve this problem.

Concepts that often appear in the same scene or have similar semantic characteristics are likely to be grouped in the same community. The task of community detection in a concept graph involves decomposing the graph into several communities including a set of highly inter-connected nodes with sparse connections between different communities. The sparsely inter-connected and densely intra-clustered communities guarantee clear semantic differences between the communities and high correlation of intra-community concepts. If an untagged sample is allocated to a specific community, the concepts in that community are likely to be candidate labels for the image.

Since the concepts are grouped into different communities, we first find the community that is likely to describe the visual content of the image, and then assign labels in this community to the image.

The quality of community detection, which is critical, is often measured by the modularity of the partition [24]. The modularity of a community is a real number between -1 and $+1$ that measures the density of intra-community links compared with inter-community ones. Given a concept graph $G = \{V, E\}$ partitioned into M communities, denoted as $S = \{s_1, s_2, \ldots, s_M\}$, modularity Q is defined as the sum of the community allocation status between concepts given as:

$$Q = \frac{1}{g} \sum_{1 \le i,j \le |C|} \{[w_{i,j} - \frac{d_i d_j}{g}]\delta_1(c_i, c_j)\}, \quad g = \sum_{i,j} w_{i,j} \tag{2}$$

where $w_{i,j}$ denotes the directed weight of the links between concepts c_i and c_j, $d_i = \sum_j w_{i,j}$ is the sum of weights of the links attached to concept c_i, δ-function $\delta_1(c_i, c_j)$ is 1 if concepts c_i and c_j are assigned the same community and 0 otherwise, $g = \sum_{i,j} w_{i,j}$ is the sum of all weights, and $|C|$ represents the number of concepts (usually $|C| \ge M$, i.e., the number of concepts is greater than the number of communities). Higher modularity of communities leads to better partition quality, which is the objective function that needs to be optimized in community detection algorithms. In this paper, a fast unfolding algorithm [45] is applied to realize latent community detection. This algorithm has proved promising in generating proper communities with optimal time complexity.

The latent community detection approach consists of two phases. The first phase (i.e., the detection phase) involves obtaining a local optimal modularity by maximizing the objective function given in Eq. (2). Based on the concept graph, each concept is first assigned a unique community. Thus, there are as many communities as concepts at this stage. For example, for NUS-WIDE with 81 concepts, there are 81 communities at this stage. Then, for concepts c_i and $c_{i,k}$ from the set of c_i's directly connected concepts $C_{i,neighbor}$, try to remove c_i to the community to which c_k belongs and recalculate modularity Q_k. The gain ΔQ_k compared with Q_{k-1}, which is the modularity before the move, can be obtained as:

$$\Delta Q_k = Q_k - Q_{k-1} \tag{3}$$

$$Q_k = \frac{1}{g} \sum_{1 \le i,j \le |C|} \{[w_{i,j} - \frac{d_i d_j}{g}\delta_1^k(c_i, c_j)]\} \tag{4}$$

where δ-function indicates whether the communities of c_i and c_j are the same for the community scheme. Obtain the maximum gain $\max\{\Delta Q_k\}$ for the trials. If $\max\{\Delta Q_k\} > 0$, which indicates an improvement in community modularity after the adjustment, the current concept is assigned the corresponding community with maximum gain of modularity. Traverse all the concepts and repeat the trials until there is no increase in modularity. After the community detection phase, a local optimal resolution is attained and no individual move of a simple concept can improve the modularity. Concepts are assigned the corresponding communities. In the second phase (i.e., the rebuilding phase) of the algorithm, a new graph $G^{'} = \{V^{'}, E^{'}\}$ is constructed. Nodes $V^{'}$ are now the communities found in the first phase. The weight between new nodes s_i and s_j is calculated as:

$$w_{i,j-new} = \sum_{a,b} w_{a,b}\delta_2(c_a, s_i)\delta_2(c_b, s_j) \quad 1 \le a, b \le |C| \tag{5}$$

where $w_{i,j-new}$ is the weight between new nodes s_i and s_j, $w_{a,b}$ is the weight between concept c_a and c_b in the old graph, the δ-function $\delta_2(c_a, s_i) = 1$ if c_a belongs to s_i in the first phase detection and 0 otherwise. Thus, $w_{i,j-new}$ is obtained by obtaining the sum of the weights of links between nodes in the corresponding two communities. Then, first phase detection is applied to the newly built graph for higher modularity. After a few rounds of the 'detection-rebuilding' iteration, the concept set C is clustered into M communities, with each concept belonging to a unique community.

According to the community detection results, it should be noted that each concept belongs to a unique community. Some concepts like 'sun' and 'sky' are likely to be associated with various scene co-occurrences from different communities. In another words, it is better to assign these concepts to multiple communities. Fortunately, the directionality of the concept co-occurrence can help solve this problem with the detailed solution given in Section V-C. The result of community detection on 81 concepts can be found in the Appendix.

C. Multiple-Kernel Learning

1) Training Sample Classification: Given a tagged image x_i whose labeling vector is denoted by T_i, the corresponding community is determined by a voting vector $N_s = \{N_{s,1}, N_{s,2}, \ldots, N_{s,m}\}$ and element $N_{s,k}$ is calculated as:

$$N_{s,k} = \sum_{1 \le j \le M} t_{i,j}\delta_2(c_j, s_k) \tag{6}$$

where $t_{i,j}$ indicates the presence of concept c_j in image x_i and δ-function $\delta_2(c_j, s_k)$ is the same as in Eq. (5). Thus, $N_{s,k}$ represents the number of concepts belonging to community s_k. The community with the maximum $N_{s,k}$ is assigned to training image x_i.

After community classification of the training samples has been completed, each labeled training image in the training set will have been associated with a unique community. The next task is to generate a mapping from the low-level image features to the community information, which is equivalent to a classification problem. In this paper, a MKL-SVM model is applied. For an untagged image, the MKL-SVM classifier first determines the most probable community for it.

2) Multi-Kernel SVM Training: Classifiers trained with a single visual feature are not robust or accurate in predicting the community label of untagged images. The traditional SVM model combines all visual features into a vector, which leads to the dimensionality curse. Besides, it is highly likely that features must be treated differently in specific classifying scenes. For example, color histogram features play a more significant role than edge detection histogram and wavelet texture features in discriminating two communities, where one includes 'sky', 'water', and 'ocean' and the other includes 'grass' and 'tree'. Hence, different visual features should have unique weights in classification. The multiple-kernel SVM model can be trained with adaptively weighted combined kernels where each kernel is associated with a specific type of visual feature. The combined kernel is given as:

$$K(p_a, p_b) = \sum_{j=1}^{q} \beta_j K_j(p_{a,j}, p_{b,j})$$
$$\text{s.t. } \beta_j \ge 0, \quad \sum_{j=1}^{q} \beta_j = 1, \tag{7}$$

where $K(\cdot)$ is the combined kernel, $K_j(\cdot)$ is the sub-kernel for the j-th visual feature, and β_j is the weight of $K_j(\cdot)$ to be learned. In addition, p_a, p_b are visual features of images x_a and x_b, respectively, with $p_a = \{p_{a,1}, p_{a,2}, \ldots, p_{a,q}\}$ and $p_b = \{p_{b,1}, p_{b,2}, \ldots, p_{b,q}\}$, and q is the number of features. The constraints on β_j are so-called '$L1$-norm' constraints that can generate a sparse solution for sub-kernel weights. This can be useful for feature selection in intra-community annotation as shown in Section V-B. The binary decision function is determined as:

$$\begin{aligned} f(p_l) &= \sum_{i=1}^{n_{train}} \alpha_i K(p_i, p_l) + b \\ &= \sum_{i=1}^{n_{train}} \alpha_i \sum_{j=1}^{q} \beta_j K_j(p_{i,j}, p_{l.j}) + b, \end{aligned} \tag{8}$$

where p_l is the feature of image x_l, α_i and b are support vector parameters for each training sample, and n_{train} is the number of images in the training set. The output of $f(\cdot)$ denotes the classification result. If $f(p_l) > 0$, x_l is classified as the positive class; otherwise x_l is classified as the negative class. The basic task of the training step is to obtain the optimal solution of sub-kernel β_j, support vector parameter α_i, and the bias b of each binary classifier. The optimization problem of binary classification is illustrated as follows:

$$\begin{aligned} &\min \ \frac{1}{2}\|f(p)\| + H\sum_{i=1}^{n_{train}} \xi_i \\ &s.t. \ \ f(p_l) = \sum_{i=1}^{n_{train}} \alpha_i K(p_i, p_l) + b \\ &\qquad K(p_i, p_l) = \sum_{j=1}^{q} \beta_j K_j(p_{i,j}, p_{l,j})\beta_j \ge 0, \quad \sum_{j=1}^{q} \beta_j = 1 \\ &\qquad \xi_i \ge 0, y_i, f(p_i) \ge 1 - \xi_i, i = 1, \ldots, n_{train}, \end{aligned} \tag{9}$$

where H is a penalty factor for classification ($H > 0$). Furthermore, the parameters including α_i, β_j, and b must be learned. SimpleMKL, proposed by Rakotomamonjy *et al.* [46], has been proved to be efficient for obtaining the optimal solution for multiple-kernel learning problems. Thus, we adopted it for training the classifier.

V. Online Annotation

The online annotation process, shown on the left of Fig. 2, consists of the following three steps: Given an untagged image, the corresponding community is first determined by the community classifier, which is trained, as explained in Section IV, based on labeled training images. Next, intra-community annotation is performed using training samples belonging to the community identified in the classification. Finally, inter-community annotation is carried out to provide complementary image annotation.

A. Community Classification

The corresponding community for an untagged image x_u is determined by the trained community classifier. The features of x_u are deployed in the MKL-SVM classifier. Voting for the most relevant class is based on the binary results from the SVM. Let $N_c = \{N_{c,1}, N_{c,2}, \ldots, N_{c,M}\}$ denote the voting vector for each community of untagged image x_u. The elements of N_c are determined by:

$$N_{c,k} = \sum_i \delta_3(f_{s_k-vs-s_i}(p_u)), \tag{10}$$

where $N_{c,k}$ denotes votes for community s_k, $f_{s_k-vs-s_i}(\cdot)$ is the binary function for classification between s_k and s_i, p_u is the visual feature of untagged image x_u, and δ-function $\delta_3(\cdot) = 1$ if $f_{s_k-vs-s_i}(p_u)$ indicates that x_u belongs to community s_k and 0 otherwise. The untagged image is assigned to the top $Z(Z \geq 1)$ possible relevant communities with the highest votes to cover more relevant labels. The proper selection of Z, the number of most relevant communities, is discussed in Section VII.

B. Intra-Community Annotation

The corresponding communities of an untagged image can be determined by trained community classifiers. A naïve KNN search is carried out to find the initial annotation in each community based on the Euclidean distance between the low-level features of the untagged image and those in the community.

If the untagged image x_u is classified into several communities $\{s_a, s_b, \ldots\}$ by the MKL-SVM and $X_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,n}\}$ denotes the initial labeled images belonging to community s_i in the training set, the distance between x_u and $x_{i,k}$ can be calculated by the visual feature as:

$$d(x_u, x_{i,k}) = \|p^*(x_u) - p^*(x_{i,k})\|_2, \tag{11}$$

where $p^*(x)$ is the combined vector of features of community s_i with larger weights. The weight $\bar{\beta}_{i,j}$ of the j-th feature in community s_i can be obtained from the binary functions trained by the MKL-SVM associated with community s_j. For each feature p_i, weight $\bar{\beta}_{i,j}$ is determined as:

$$\bar{\beta}_{i,j} = \frac{1}{M-1} \sum_{k \neq i} \beta_{k,j}, \tag{12}$$

where $\beta_{k,j}$ is the weight of the j-th feature of the binary function associated with community s_k. In the MKL-SVM, each binary classifier (1-vs-1) is trained using a different group of kernel weights, which implies different weights of the low-level features. Features with large weights can represent the community better. In this paper, a low-level feature with a weight greater than the average is selected as the distinguished feature of a specific community.

For the images in community i, the tagging status can be represented as an m-dimensional binary vector $T_j^i = \{t_{j,1}^i, t_{j,2}^i, \ldots, t_{j,m}^i\}(j = 1, 2, \ldots, k)$ where m is the number of concepts. It should be noted that intra-community annotation only assigns the untagged image concepts in the corresponding community. For untagged image x_u and its k-nearest neighborhood $\{x_{i,1}, \ldots, x_{i,k}\}$ with tagging status $\{T_1^i, T_2^i, \ldots, T_k^i\}$, the confidence of each tag is generated as vector T_p:

$$\begin{aligned} T_p &= \{t_{p,1}, t_{p,2}, \ldots, t_{p,m}\} \\ t_{p,q} &= \frac{1}{k} \sum_{j=1}^{k} t_{j,q}^i \quad (q = 1, 2, \ldots, m) \\ T_j^i &= \{t_{j,1}^i, t_{j,2}^i, \ldots, t_{j,m}^i\} \quad (j = 1 \ldots k), \end{aligned} \tag{13}$$

where each element of T_p is a real value between 0 and 1, denoting the confidence of each label. T_j^i is the label status of the k-nearest neighbors.

C. Inter-Community Annotation

Various problems may arise after images have been annotated with various concepts during intra-community annotation. Intra-community annotation is carried out separately on the top Z communities. In other words, potential tagged concepts can only come from these Z communities, and no others. Some concepts that are highly correlated with the tagged ones, but do not belong to the top two communities cannot be included. For example, certain concepts like ‘sun’ and ‘sky’ are associated with various scene co-occurrences from different communities. However, they will belong to a unique community after community detection, which may lead to missing annotations in some cases. Therefore, an inter-community annotation strategy is applied to compensate for this deficiency. The link between concepts, which implies co-occurrence, is characterized by directionality. For concepts c_i and c_j, there is a large difference between $w_{i,j}$ and $w_{j,i}$. Unidirectional co-occurrence helps to obtain extra concepts from a tagged one. If an image has been tagged with c_i and $w_{i,j}$ is much greater than $w_{j,i}$, c_j is likely to be the concept that is theoretically shared by multiple communities like ‘sun’ and ‘sky’. The inter-community strategy labels such an image with c_j.

In practice, for image x_u tagged with concept c_i after intra-community annotation, finding c_i’s directly connected concepts $\{c_{d,1}, c_{d,2}, \ldots, c_{d,t}\}$ is based on the concept graph. If $c_{d,j}$ is not a tag of this image and the conditional probability $P(c_{d,j}|c_i)$ exceeds the confidence threshold, say 0.6, (further discussion can be found in experiments), c_i is the support concept for $c_{d,j}$ and $c_{d,j}$ will be included in x_u’s tags. The confidence of the newly tagged label from concept c_i is calculated as:

$$t_{c_{d,j},c_i} = t_{c_i} \times P(c_{d,j} \,|c_i) \tag{14}$$

where t_{c_i} is the confidence of concept c_i of image x_u after intra-community annotation. If concept $c_{d,k}$ is supported by multiple concepts, the final confidence of $c_{d,k}$ is the sum of the support confidence of each concept:

$$t_{c_{d,k}} = \sum_i t_{c_{d,k},c_i}. \tag{15}$$

VI. Complexity Analysis

In this section, we analyze the computational complexity of LCMKL. In offline learning, we firstly detect the semantic communities on concept graph. The time complexity of constructing concept graph is $O(m^2)$ where m is the number of semantic concepts and the space complexity is $O(m^2)$. The complexity of community detection is $O(m^2 log_2 m)$ since the iteration of community detection is similar to a hierarchical clustering process. The space complexity of community detection is also $O(m^2)$. After community detection, all training samples are assigned to specific community whose time complexity is $O(nmlog_2 m)$ where n is the number of training samples and $n \gg m$. Therefore, it can be considered linear to the number of training samples. Then, we train the multiple kernel SVM based on the samples with multiple features. Since we adopt 1vs1 strategy for learning community classifiers, we have to train $n_c(n_c - 1)/2$ MKL-SVMs where n_c is the number of communities. However, it is hard to analyze the computational complexity of MKL-SVM training. It is related to the number of features and the number of training samples in each community. We provide the training time of MKL-SVM in Section VII-G.

In online annotation, the untagged samples are firstly assigned to the most relevant communities by MKL-SVM. After that, we will give the initial tags via intra-community annotation which is actually a K-NN process in top M relevant communities. If K-NN is boosted by KD-Tree, the time complexity of building for all communities is $O(n_c n_{cs} log_2 n_c)$ where n_c is the number of communities and n_{cs} is the number of training samples in each community. The time complexity of annotation is $O(Mn_t log_2 n_{cs})$ where M is the number of candidate communities and n_t is the number of untagged images. Therefore, the total time complexity of intra-annotation is $O(n_c n_{cs} log_2 n_c + Mn_t log_2 n_{cs})$. The annotation result is enhanced with inter-community annotation whose time complexity is $O(n_t m^2)$.

VII. Experiments and Discussion

In this section, we discuss various experiments conducted to validate the performance of the proposed method on the NUS-WIDE [44] and IAPR TC-12 [47] datasets. A comparison of LCMKL and state-of-the-art methods JEC [1], ML-KNN [12], ML-NB [36], RLVT [48], RANK [48], TagProp [49], and NBVT [50] is also presented. JEC is a neighbor-voting scheme where each feature contributes equally to the image distance. NBVT is another neighbor voting method for tag relevance estimation. TagProp is a label propagation method based on neighbors. RLVT takes the relevance between tags into consideration based on the Google distance combined with low-level visual features. RANK is an extension of RLVT using a random walk. MLNB is a learning-based method. Principal component analysis (PCA) or a genetic algorithm is adopted for feature selection, while naïve Bayesian inference is used for label allocation. ML-KNN is derived from the k-nearest method exploiting Bayesian rules.

According to the latest progress in feature learning, we also present various comparisons with stronger features including the vector of locally aggregated descriptors (VLAD) [51] and convolutional neural network (CNN) features [52]. The details can be found in Section VII-E.

All the experiments were executed on a PC with an Intel 2.4GHz CPU and 8GB RAM. Most of the algorithm is implemented with MATLAB except MKL-SVM is implemented with C++.

A. Datasets

NUS-WIDE dataset is a large-scaled real-world dataset crawled from Flickr and used in many research studies in recent years. Several low-level visual features are provided by the founder of NUS-WIDE including color histogram (CH-64D), color correlation histogram (CORR-73D), edge-detection histogram (EDH-73D), block-wise color moments (CM-256D), and wavelet textures (WT-128D). The Lite version of NUS-WIDE dataset is composed of two parts: the training part containing 27807 images, and the testing part containing 27808 images. All images are tagged with the concepts from 81 Ground Truth.

IAPR TC-12 dataset was used for the ImageClef Challenge from 2006 to 2008. It consists of still natural images taken from locations around the world and comprising an assorted cross-section of still natural images. In this paper, we use the features extracted by INRIA [49] including Gist (512D), DenseHue (100D), HarrisHue (100D), DenseSift (1000D), HarrisSift (1000D). The numbers of training and test samples are 17665 and 1962, respectively. The annotation model is trained using the training part while the evaluation of the model is based on the testing part. All visual features are deployed for the compared methods. We also use the pre-trained Deep Convolutional Neural Networks [53] and VLAD [51] to obtain stronger features for further comparison.

B. Evaluation Criteria

In this paper, the F1-score and average precision (AP) are used to measure the performance of the image annotation. The F1-score measure is calculated as:

$$\begin{aligned} F_1 - score(c_i) &= 2\frac{\text{Precision}(c_i) \times \text{Recall}(c_i)}{\text{Precision}(c_i) + \text{Recall}(c_i)} \\ \text{Precision}(c_i) &= \frac{N_{corr}}{N_{tagged}} \\ \text{Recall}(c_i) &= \frac{N_{corr}}{N_{all}} \end{aligned} \tag{16}$$

where N_{tagged} denotes the number of images tagged with a specific concept c_i in the testing part by image annotation,

TABLE I

PERFORMANCE COMPARISON ON NUS-WIDE 81-TAGS

Measures	MLKNN	MLNB	RLVT	RANK
F1-score (Top 5)	0.062	0.212	0.212	0.207
F1-score (Top 10)	0.079	0.202	0.204	0.202
AP	0.536	0.606	0.392	0.350
Measures	NBVT	JEC	TagProp	LCMKL
F1-score (Top 5)	0.230	0.200	0.193	0.324
F1-score (Top 10)	0.253	0.216	0.216	0.315
AP	0.663	0.438	0.419	0.668

N_{corr} denotes the number of images tagged correctly according to the original tagging information, and N_{all} denotes the number of images tagged with c_i in the training part. For fair comparison, for each untagged image, the top five and top ten relevant concepts are selected for annotation.

Average Precision (AP) [12], [36], which evaluates the average fraction of labels ranked above a particular label, is calculated as:

$$AP(f) = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \frac{1}{\left|GT_i\right|} \sum_{GT_{i,j} \neq 0} \frac{|\{k|T_{i,k} \geq T_{i,j}\}|}{pos(T_{i,j})}, \qquad (17)$$

where n_{test} is the number of training samples, GT_i is the ground truth for image x_i, $GT_{i,j}$ indicates the existence of the j-th concept in the ground truth, $T_{i,j}$ is the confidence of concept j in image x_i generated by LCMKL or the other methods, $pos(T_{i,j})$ is the ranking position of the j-th concept's confidence in descending order. The performance is optimal if $AP(f) = 1$. The greater the value of $AP(f)$ is, the better is the performance.

C. Experiments on NUS-WIDE 81-Concepts

In this section, we present the performance of our method and the compared methods. Based on the tagging information of the training part, a concept graph is first constructed. Nine latent communities are detected over the concept graph using the community detection algorithm. The visual features are adopted in the training of community classifiers based on the MKL-SVM model. The optimal selection of sub-kernel weights and SVM parameters is solved using the shogun-toolbox 2.0 with the simple MKL algorithm [46]. According to the results of the community classification, the optimal sub-kernel weights are also obtained.

The most discriminative visual features are color moments and the color correlation histogram since their weights are, respectively, 0.62 and 0.22 on average in MKL. They are combined into a feature vector for intra-community annotation. In intra-community annotation, the number of nearest neighbors K is 50 and the confidence threshold for inter-community annotation is 0.6. For NBVT, the best performance is achieved with the number of neighbors set to 50. The annotation results are presented in TABLE I.

TABLE I shows the performance of LCMKL and other methods. According to the results, we see that the proposed method outperforms both the F1-scores with the top five (Top 5) and top ten (Top 10) relevant tags and AP values of the compared methods. The F1-score for LCMKL is 0.324 for the Top 5, which is much higher than the values for the other methods. LCMKL also yields the best AP value. We also test LCMKL on NUS-WIDE 81Concept for 10 times. The performance is stable that the AP is 0.668 ± 0.01 and F1-score is 0.324 ± 0.02.

With an increasing number of tags, the F1-score, of Since the average number of labels for images in the NUS-WIDE 81-Tag dataset is 4.2, annotation performance is stable with more than five tagged labels. The average F1-score of LCMKL remains at about 0.33, which is 30% higher than for NBVT and 50% higher than for RANK, MLNB, and RLVT.

In addition, we found that the AP for each concept using LCMKL suffers a slight loss after the inter-community process (dropping from 0.3185 to 0.2941), while recall increases by more than 18.8% (0.3032 to 0.3601). In general, the recall of each concept increases after inter-community annotation. For the most improved concepts 'sky' and 'clouds', the recall values are, respectively, 0.675 and 0.493 higher than the results without inter-community annotation. The improvement is associated with the number of relevant instances and training samples for classification. For NUS-WIDE with 81 concepts, there are 27807 images in the training part. The training images are re-annotated according to the results of community detection and their original tagging information. The number of images initially annotated with 'sky' is 19052. Only 5942 of these, however, are grouped in the specific community that includes 'sky', since 'sky' is very common in outdoor scenes. Images with tag 'sky' are not always densely semantically correlated, with several images scattered in other communities. According to the procedure for intra-community annotation, only test images classified as a specific community can be tagged with 'sky'. With the help of inter-community annotation, those concepts semantically correlated with 'sky' but not belonging to the specific community can lead to the recognition of 'sky'. For images in the NUS-WIDE 81-Tag dataset, the average number of tags is 4.2. If the number of tagged concepts is increased to ten, more true tags are likely to be covered. Inter-community annotation does not improve recall performance markedly.

D. Performance on IAPR TC-12 Dataset

In the evaluation using the IAPR TC-12 dataset, the visual features for LCMKL are grouped into two conditions: global features only (Gist + DenseHue + HarrisHue) and global-local mixed features (Gist + DenseHue + HarrisHue + HarrisSift + DenseSift). Since local features are not available for the Lite version of NUS-WIDE, we evaluated the performance thereof using the IAPR dataset. For the compared methods, Global-Local Mix features are used. For the F1-score, the number of relevant tags is set to ten because the average number of tags in a single image is 5.7 in the IAPR dataset. The detailed results are shown in TABLE II.

According to the F1-score, LCMKL outperforms the other methods irrespective of whether global features only or global-local mixed features are used. Nevertheless, the local features boost LCMKL to achieve a higher AP than the other methods. The results also demonstrate that local features

TABLE II
PERFORMANCE COMPARISON ON THE IAPR TC-12 DATASET

Measures	MLKNN	RLVT	TagProp	RANK
AP	0.294	0.071	0.233	0.071
F1-score	0.151	0.098	0.170	0.096
Measures	JEC	NBVT	LCMKL-Global	LCMKL-Mix
AP	0.233	0.280	0.279	0.321
F1-score	0.166	0.154	0.169	0.180

TABLE III
COMPARISON WITH STRONGER FEATURES

Measures	VLAD + LCMKL	CNN + LCMKL	VLAD + L-SVM	CNN + L-SVM	LCMKL
AP	0.302	0.322	0.297	0.213	0.321
F1-score	0.190	0.216	0.279	0.192	0.180

can improve the performance of LCMKL. Note that with the exception of LCMKL-Global, all the other approaches used mixed features like those in LCMKL-Mix. We also test LCMKL on IAPR TC-12 for 10 times. The performance is stable that the AP is 0.321±0.02 and F1-score is 0.180±0.01.

E. Comparison With Stronger Features on IAPR TC-12

According to the recent works of Razavian *et al.* [52], the feature extracted by deep CNNs shows good performance on classification, detection, and recognition tasks. In this section, we adopt stronger features including VLAD [51] and the CNN feature [52] for comparison using the IAPR TC-12 dataset. According to the default setting in the Vl-Feat Library that implements VLAD using MATLAB, the number of visual words is 64. Therefore, the dimension of the VLAD feature is 6400D. For the CNN feature, we refer to the structure established by Krizhevsky *et al.* [53], which is a multi-layer deep CNN. The network is pre-trained using 15 million images from ImageNet. The output of the fc-7 layer, which is 4096D, is adopted for the DeepNet feature. In this paper, the dimension of the CNN feature of an image is 4096D and that of VLAD is 6400D.

We designed two sets of experiments:

1) We replaced the MKL part for community classification with a single-kernel SVM and either the CNN feature or VLAD to demonstrate the performance of our main framework (denoted by 'CNN+LCMKL' and 'VLAD+LCMKL', respectively). According to Fig. 2, the community classifier is trained after latent community detection using the CNN feature or VLAD for classifying the untagged image as a specific community.
2) We trained binary classifiers for each concept using a Linear SVM and either the CNN feature or VLAD to make binary decisions on the existence of concepts (denoted by 'CNN+LSVM' and 'VLAD+LSVM', respectively).

The performance is given in TABLE III.

Since VLAD and the CNN feature are stronger features than previously used, they perform better than the original LCMKL. According to the report by Razavian *et al.* [52], 'Linear SVM + CNN feature' achieves good performance on classification and recognition. In fact, we found that the AP of 'Linear SVM + CNN feature' is almost the same as that of the original LCMKL, while the F1-score is greatly superior to that in the original method. However, the performance of 'VLAD + Linear SVM' is similar to the original LCMKL in terms of F1-score, but not in terms of AP. In addition, we found that 'Feature + SVM' does not perform well in terms of AP, but has a good F1-score. The reason for this is that the LSVM only produces a binary classification and not a ranking result; AP, however, considers ranking relevance.

TABLE IV
PERFORMANCE COMPARISON WITH A HYBRID GRAPH SCHEME AND TEXTUAL GRAPH SCHEME

Measures	0.6	0.4	0.2	0.1	0.05	0
AP	0.614	0.623	0.649	0.640	0.658	0.668
F1-score	0.266	0.293	0.325	0.321	0.321	0.324

F. Parameter Tuning and Discussion

1) Weight of the Concept Graph: As mentioned in Section IV-B, the weight of the concept graph is only dependent on the number of co-occurrences between the concepts, which considers only textual relevance. However, the features of concept co-occurrence can also be reflected by low-level image features tagged with a specific concept. In this section, we incorporate the image features into the weight of the concept graph and evaluate the performance thereof. In our method, a directed-weighted graph $G = \{V, E\}$ is constructed. The elements of vertex set V are tags from concept set $C = \{c_1, c_2, \ldots, c_m\}$. Two concepts c_i and c_j are connected by edge $e_{i,j}$ and the weight of $e_{i,j}$ is calculated as:

$$w_{i,j} = P(c_j|c_i) = \frac{N(c_i \wedge c_j)}{N(c_i)}. \tag{18}$$

To incorporate the low-level image features into the weight of the concept graph, the image distance between the directly-connected concepts is given by:

$$d_{img}(c_i, c_j) = \|f_{avg}^{c_i} - f_{avg}^{c_j}\| \tag{19}$$

$$f_{avg}^{c_i} = \frac{1}{\|x_{i,t_{j,i}=1}\|} \sum_{x_{i,t_{j,i}=1}} f_i \tag{20}$$

where f_{avg}^{*} is the mean of the low-level features of concepts used to tag the images, while the distance between the concepts is measured by the Euclidean distance between the means of the low-level features. Then, the image distance is normalized as a real number between 0 and 1 and combined with the textual distance:

$$w_{i,j} = \lambda d_{img}(c_i, c_j) + (1 - \lambda) d_{textual}(c_i, c_j), \tag{21}$$

where λ is the adjusting parameter between the textual distance and image distance. The annotation results on NUS-WIDE 81-Tags with varying values of λ are given in TABLE IV.

As shown in TABLE IV, when $\lambda = 0.2$, the final F1-score of LCMKL is 0.3251, which is a little higher than that for the simple textual scheme. As the weight of the image distance increases, the performance of LCMKL deteriorates.

TABLE V

COMPARISON OF THE PERFORMANCE OF LCMKL WITH DIFFERENT COMMUNITY DETECTION ALGORITHMS

Measures	LCMKL-AF	LCMKL-RN	LCMKL-BV
F1-score	0.231	0.277	0.324
AP	0.654	0.636	0.668

2) Performance of Different Community Detection Methods: In this section, the selection of the community detection method is discussed. The community detection methods proposed by Ronhovde and Nussinov [54] (denoted as LCMKL-RN) and Arenas *et al.* [55] (denoted as LCMKL-AF) were chosen for comparison. The fast unfolding algorithm deployed in this paper is denoted as LCMKL-BV [45]. The performance on the NUS-WIDE 81-Tags is shown in TABLE V. We find that the LCMKL approaches achieve better performance.

As shown in TABLE V, the community detection method used in this paper helps LCMKL achieve the best performance compared with LCMKL-AF and LCMKL-BV. The reason for this is that LCMKL-BV generates the correct community detection for concepts at a semantic level and the number of training samples in each community is more uniform than for LCMKL-AF and LCMKL-RN. Since the performance of the SVM-like learning methods is largely influenced by the uniformity of the number of training samples in the positive and negative classes, the uniform training sample assignment by LCMKL-BV leads to better performance.

3) Latent Community Detection and Topic Models: LDA was initially proposed for document classification and later extended to image classification. Irrespective of whether it is used for document classification or image annotation, LDA requires fairly high-dimensional features. In [28], the image is pre-segmented into eight regions and 40 features are calculated for each region. In another words, the training data must be elaborately pre-processed to ensure each image contains a large number of 'words'. Another method for generating the words for an image is to use local features (e.g., SIFT and SURF). However, latent community detection can be carried out using only the concept graph containing information about concepts and links for tag co-occurrence. We do not have to pre-segment the image and tag each region with a specific label.

We tested the performance of LDA replacing latent community detection as discussed in Section IV-B on the NUS-WIDE dataset with 81 tags. Since the images are assigned some tags (with no region information or even local features available), the words for images are given as: '{Image1: cat: 1; grass: 1}; {Image2: people: 1}. Nine topics (each concept is allocated a max priori probability) are identified based on the tag information. The F1-score of the alternative version of LCMKL (where latent community detection is replaced by LDA) is 0.042, which differs greatly from that of community detection.

We also tested the performance of Labeled-LDA [56] and CorrLDA on the IAPR dataset since local features are available in this dataset. The codebook generated by DenseSift was used as words for the LDA. After 3,000 iterations, the F1-score and AP for Labeled-LDA were 0.0197 and 0.0445, respectively, while CorrLDA yielded 0.039 as the F1-score and 0.249 as the AP. Although CorrLDA does not reflect the most relevant tags, ranking of some weakly-relevant tags is performed well. Therefore, the AP for CorrLDA is not as bad as its F1-score. This result corresponds with that reported in [1] where CorrLDA does not perform well on the Corel5K dataset. We refer to the conclusion in [57]. Previous LDA-based topic models for image annotation all operate under the Dirichlet assumption, where each topic proportion is assigned independently, which leads to an unrealistic limitation that the presence of one topic is not correlated with the presence of others. This is the main problem with LDA-based approaches.

TABLE VI

COMPARISON OF THE PERFORMANCE OF LCMKL WITH A VARYING NUMBER OF RELEVANT COMMUNITIES

Measures	top1 Intra	top1 Inter	top2 Intra	top2 Inter	top3 Intra	top3 Inter
AP	0.532	0.664	0.538	0.668	0.532	0.664
F1-score	0.271	0.295	0.311	0.324	0.300	0.317

TABLE VII

PERFORMANCE COMPARISON WITH DIFFERENT PENALTY COEFFICIENTS

H	0.01	0.05	0.1	0.5	5	50
AP	0.654	0.656	0.657	0.668	0.655	0.652
F1-score	0.294	0.302	0.303	0.324	0.302	0.296

4) Parameters for Multiple-Kernel Learning: As mentioned in Section V-A, an untagged image is assigned the top Z relevant communities by the community classifier. Selection of the number of relevant communities is discussed in this section. The performance of LCMKL on NUS-WIDE 81-Tags with the top 1, top 2, and top 3 (i.e., $Z = 1$, 2 and 3) relevant communities is given in TABLE VI. Each untagged image is tagged with five recommended tags.

In TABLE VI, LCMKL with the top 2 relevant communities achieves the best performance in terms of F1-score. When only assigned the most relevant community, the untagged image is not associated with certain concepts that are shared by multiple communities at the semantic level. Instead, it is only assigned one community by the community detection algorithm. If these concepts were not complemented by inter-community annotation, the performance in terms of recall would suffer greatly. However, as the number of relevant communities increases, the performance is affected by the accumulated tagging noise in multiple communities. Extra communities with large numbers of training images are likely to take over the top position in tagging relevance. Therefore, $Z = 2$ is relatively optimal for the NUS-WIDE dataset. We also tested the performance of LCMKL with various values of the penalty coefficient H in the objective function of MKL-SVM given in Eq. (9). The results for $H = 0.05$, $0.5, 5, 50$, and 500 are given in TABLE VII.

The table shows that the performance of LCMKL does not change greatly with different values of H. With H set to 0.5, LCMKL achieves the best performance.

TABLE VIII
PERFORMANCE COMPARISON OF THE MKL-SVM, SKL-SVM, AND LIBLINEAR ON NUS-WIDE

Measures	MKL-SVM	LIBLINEAR	SKL-SVM
AP	0.668	0.571	0.622
F1-score	0.324	0.273	0.295

TABLE IX
PERFORMANCE COMPARISON BETWEEN MKL AND SKL

Measures	DS	DS+HS	All	MKL
AP	0.291	0.295	0.311	0.321
F1-score	0.131	0.135	0.173	0.180

TABLE X
THE PERFORMANCE GAIN FROM INTER-COMMUNITY ANNOTATION

	Intra-only	Intra+Inter	Intra-only	Intra+Inter
	@NUS-WIDE	@NUS-WIDE	@IAPR	@IAPR
AP	0.538	0.668	0.291	0.321
F1-score	0.311	0.324	0.163	0.180

5) Multiple-Kernel Learning vs. Single-Kernel Learning: In this section, we replace the MKL-SVM in LCMKL with LIBLINEAR [58] and a single-kernel SVM (SKL-SVM) [59] and test the performance on the NUS-WIDE 81-Tags dataset. For LIBLINEAR and SKL-SVM, image features are combined into a feature vector for each image and the remaining steps of the image annotation process are the same as those for LCMKL. The top 2 relevant communities (i.e., $Z = 2$) are selected according to the probability estimation of SKL-SVM and LIBLINEAR. The results are shown in TABLE VIII. Compared with the single-kernel SVM and LIBLINEAR, the MKL-SVM achieves a higher AP and F1-score on the NUS-WIDE dataset. Moreover, testing time is reduced since MKL does feature selection for intra-community annotation.

We used the IAPR TC-12 dataset with five visual features (Gist, DenseHue, DenseSift, HarrisHue, HarrisSift) for evaluation since it provides stronger visual representation. Three different combinations were adopted as follows:

a. DenseSift only (denoted by ‘DS’)
b. DenseSift+ DenseHue as a long vector (denoted by ‘DS+HS’)
c. All five visual features as a long vector (denoted by ‘All’)

The performance on the IAPR TC-12 dataset is given in TABLE IX. We can see that the single-kernel SVM with all five visual features achieves almost the same performance as MKL. However, MKL is used not only to achieve better performance in community classification, but also to boost the intra-community annotation process. We use the optimal selection of visual features in each community to find the nearest neighbors.

6) Intra-Community vs. Inter-Community: As mentioned in Section V-C, the inter-community annotation strategy is applied to compensate for this deficiency. We find that the performance gains from the inter-community annotation both on NUS-WIDE and IAPR TC-12 as shown in TABLE X.

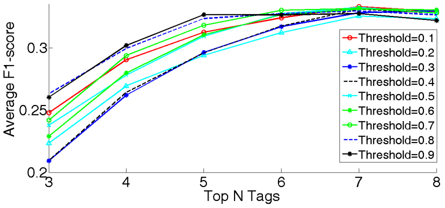

Fig. 3. Average F1-scores for LCMKL with different inter-community annotation thresholds.

In TABLE X, ‘Intra-only’ denotes that LCMKL is conducted without inter-community annotation while ‘Intra+Inter’ is the result after both intra and inter community annotation. The performance of LCMKL on NUS-WIDE and IAPR TC-12 is boosted by inter-community annotation.

7) Threshold of Inter-Community Annotation: As mentioned in Section V-A, for concept c_i tagged after intra-community annotation, LCMKL finds its directly connected concepts $\{c_{d,1}, c_{d,2}, \ldots, c_{d,t}\}$ based on the concept graph. If $c_{d,j}$ is not tagged to this image and the conditional probability $P(c_{d,j}|c_i)$ exceeds the confidence threshold (e.g., 0.6), this concept will be included. Experiments on NUS-WIDE 81-Tags have demonstrated that inter-community annotation helps improve both recall and the F1-score of the final annotation. However, the value of the confidence threshold is pre-defined and it is likely to influence the performance of inter-community annotation. In this section, the impact of the confidence threshold is discussed.

Varying the confidence threshold between 0.1 and 0.9, we tested the performance with different numbers of tagged relevant concepts (top 3 to top 8). The results of the experiments are illustrated in Fig. 3.

When tagged with five concepts or fewer, the performance tends to improve with an increase in the confidence threshold, which satisfies the purpose of inter-community annotation. Inter-community annotation is applied to solve the problem of the omission of certain concepts like ‘sun’ and ‘sky’, which are assigned only one community, but are semantically associated with multiple communities. A basic feature of unidirectional concept co-occurrence is the large difference between $w_{i,j}$ and $w_{j,i}$. A lower confidence threshold leads to falsely tagged images. If the number of tagged labels is greater than five, the confidence threshold does not obviously affect performance. As mentioned above, for images in the NUS-WIDE 81-Tags dataset, the average number of tags is 4.2. If the number of tagged concepts is increased to ten, it is more likely to cover more true tags. Inter-community annotation does not significantly improve the performance in terms of recall.

8) Parameter Settings for Intra-Community Annotation: For intra-community annotation, we varied the number of nearest neighbors K in KNN. The performance on NUS-WIDE 81-Tags with varying values of K ($K = 10, 30, 50, 100, 150, 200$) is shown in TABLE XI. As the value of K increases, the performance deteriorates slightly. However, different numbers of nearest neighbors do not obviously affect the final performance of LCMKL. This is because the supporting confidence $t_{p,q}$ of each concept in Eq. (13) is averaged by

TABLE XI
PERFORMANCE OF LCMKL WITH VARYING NUMBERS OF K-NEAREST NEIGHBORS

Measures	K=10	K=30	K=50	K=100	K=150	K=200
AP	0.647	0.663	0.668	0.666	0.665	0.664
F1-score	0.315	0.323	0.328	0.326	0.324	0.3185

TABLE XII
COMPARISON OF THE PERFORMANCE OF THE FULL FEATURE SCHEME AND OPTIMAL FEATURE SCHEME FOR INTRA-COMMUNITY ANNOTATION

Measures	AP	F1-score	Time consumption (sec)
Optimal Feature	0.668	0.328	0.1396
Full Feature	0.669	0.334	0.3466

TABLE XIII
RUNNING TIME OF EACH STEP IN LCMKL

Train	Graph Construction	Community Detection	MKL-SVM Training
Time (training part)	3.04s	0.80s	3132s
Code	MATLAB	MATLAB	C++
Test	MKL-SVM Classification	Intra-community Annotation	Inter-community Annotation
Time (per image)	0.01s	0.14s	2×10^{-4}s
Code	C++	MATLAB	MATLAB

the number of nearest neighbors. Therefore, an increase in nearest neighbors does not affect performance.

As mentioned in Section V, low-level features with a weight greater than the average are selected as distinguished features of a specific community. Then, naïve KNN annotation is performed in each community with the most distinguished features. The performance of the optimal feature selection and the full feature selection on NUS-WIDE 81-Tags is given in TABLE XII. For the full feature scheme, five low-level features are combined into a vector as a visual feature of the image.

As shown in TABLE XII, the optimal selection of visual features achieves similar performance in terms of F1-score to the full feature scheme. However, the time required for the testing part on NUS-WIDE using the optimal feature scheme for intra-community annotation is only 0.1396 sec, which is much quicker than using the full feature scheme with a computation time of 0.3466 sec.

G. Running Time

The theoretical analysis on the complexity of LCMKL is presented in Section VI. We provide the running time of LCMKL on NUS-WIDE-Lite dataset in TABLE XIII. We use 27,808 samples for training and 27,807 for testing. The training time in TABLE XIII is calculated over the whole training procedure on 27,808 samples while the testing time is the average value on single sample. The most time-consuming steps are the training of LCMKL and Intra-community detection. However, they are implemented with different type of code. We believe that the efficiency of LCMKL can be enhanced with more optimization on code.

TABLE XIV
COMMUNITY DETECTION ON 81 CONCEPTS FROM NUS-WIDE

COM1(15)	beach	boats	bridge	coral	fish
harbor	lake	ocean	reflection	sand	sun
sunset	surf	water	whales		
COM2(15)	buildings	cars	castle	cityscape	house
nighttime	police	railroad	road	street	tower
town	train	vehicle	window		
COM3(13)	airport	clouds	earthquake	fire	flags
military	moon	plane	rainbow	sign	sky
statue	temple				
COM4(13)	animal	bear	birds	cat	cow
dog	elk	fox	grass	horses	tiger
running	zebra				
COM5(8)	dancing	person	protest	soccer	sports
tattoo	swimmers	wedding			
COM6(7)	frost	glacier	mountain	rocks	snow
valley	waterfall				
COM7(6)	flowers	food	garden	leaf	plants
tree					
COM8(3)	book	computer	toy		
COM9(1)	map				

VIII. CONCLUSION

In this paper, we proposed the LCMKL framework for automatic image annotation. Our work integrates community features of multi-labeled images with multiple-kernel learning. A concept graph is constructed, which implies a dense semantic intra-community correlation of concepts. A robust multiple-kernel SVM is applied for community classification. Intra-community annotation makes initial decisions for label assignment. The final tagging performance is improved by inter-community annotation.

To evaluate the performance of our method, we applied our method to various experiments on the NUS-WIDE and IAPR TC-12 datasets. From the results of the experiments, it can be seen that our method outperforms classical and state-of-art methods for image annotation which is boosted by semantic communities. By introducing stronger visual representation (e.g. VLAD and DeepFeat), LCMKL is also proved a general framework which can be adapted with different features and classifiers flexibly.

APPENDIX
RESULT OF COMMUNITY DETECTION

We provide the result of community detection on NUS-WIDE dataset with 81 concept as shown in TABLE XIV. "COM i" ($i = 1, \dots, 9$) is short for ith community. The number after "COM i" is the number of concepts in the community. Since the number of images tagged with "map" is lower than 10 in NUS-WIDE dataset, there are only one concept in COM 9. We can find that the intra-community coherence is quite promising. Since the intra-community annotation step assigns the tags to images only from the community, the precision can be largely enhanced.

REFERENCES

[1] A. Makadia, V. Pavlovic, and S. Kumar, "Baselines for image annotation," *Int. J. Comput. Vis.*, vol. 90, no. 1, pp. 88–105, 2010.

[2] D. Zhang, M. M. Islam, and G. Lu, "A review on automatic image annotation techniques," *Pattern Recognit.*, vol. 45, no. 1, pp. 346–362, 2012.

[3] Y. Yang, F. Wu, F. Nie, H. T. Shen, Y. Zhuang, and A. G. Hauptmann, "Web and personal image annotation by mining label correlation with relaxed visual graph embedding," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1339–1351, Mar. 2012.

[4] J. Tang, Z.-J. Zha, D. Tao, and T.-S. Chua, "Semantic-gap-oriented active learning for multilabel image annotation," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2354–2360, Apr. 2012.

[5] J. Tang, S. Yan, C. Zhao, T.-S. Chua, and R. Jain, "Label-specific training set construction from Web resource for image annotation," *Signal Process.*, vol. 93, no. 8, pp. 2199–2204, 2013.

[6] J. Tang, R. Hong, S. Yan, T.-S. Chua, G.-J. Qi, and R. Jain, "Image annotation by *k*NN-sparse graph-based label propagation over noisily tagged Web images," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 2, 2011, Art. ID 14.

[7] P. T. M. Saito, P. J. de Rezende, A. X. Falcão, C. T. N. Suzuki, and J. F. Gomes, "A data reduction and organization approach for efficient image annotation," in *Proc. 28th Annu. ACM Symp. Appl. Comput.*, 2013, pp. 53–57.

[8] R. Yan, A. Natsev, and M. Campbell, "A learning-based hybrid tagging and browsing approach for efficient manual image annotation," in *Proc. IEEE CVPR*, Jun. 2008, pp. 1–8.

[9] Z. Ma, F. Nie, Y. Yang, J. R. R. Uijlings, and N. Sebe, "Web image annotation via subspace-sparsity collaborated feature selection," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 1021–1030, Aug. 2012.

[10] X. Liu, R. Liu, F. Li, and Q. Cao, "Graph-based dimensionality reduction for KNN-based image annotation," in *Proc. 21st ICPR*, Nov. 2012, pp. 1253–1256.

[11] Y. Han, F. Wu, Q. Tian, and Y. Zhuang, "Image annotation by input–output structural grouping sparsity," *IEEE Trans. Image Process.*, vol. 21, no. 6, pp. 3066–3079, Jun. 2012.

[12] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, 2007.

[13] X. Qian, X. Liu, C. Zheng, Y. Du, and X. Hou, "Tagging photos using users' vocabularies," *Neurocomputing*, vol. 111, pp. 144–153, Jul. 2013.

[14] J. Fan, Y. Gao, and H. Luo, "Hierarchical classification for automatic image annotation," in *Proc. 30th ACM SIGIR*, 2007, pp. 111–118.

[15] X. Zhou, M. Wang, Q. Zhang, J. Zhang, and B. Shi, "Automatic image annotation by an iterative approach: Incorporating keyword correlations and region matching," in *Proc. 6th ACM CIVR*, 2007, pp. 25–32.

[16] L. Wu, X.-S. Hua, N. Yu, W.-Y. Ma, and S. Li, "Flickr distance," in *Proc. ACM Multimedia*, 2008, pp. 31–40.

[17] J. Liu, M. Li, W.-Y. Ma, Q. Liu, and H. Lu, "An adaptive graph model for automatic image annotation," in *Proc. 8th ACM Int. Workshop Multimedia Inf. Retr.*, 2006, pp. 61–70.

[18] Y. Gao, J. Fan, X. Xue, and R. Jain, "Automatic image annotation by incorporating feature hierarchy and boosting to scale up SVM classifiers," in *Proc. ACM Multimedia*, 2006, pp. 901–910.

[19] C. Cusano, G. Ciocca, and R. Schettini, "Image annotation using SVM," in *Proc. Int. Soc. Opt. Photon. Electron. Imag.*, vol. 5304. Dec. 2003, pp. 330–338.

[20] O. Chapelle, P. Haffner, and V. N. Vapnik, "Support vector machines for histogram-based image classification," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1055–1064, Sep. 1999.

[21] K.-S. Goh, E. Y. Chang, and B. Li, "Using one-class and two-class SVMs for multiclass image annotation," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 10, pp. 1333–1346, Oct. 2005.

[22] X. Qi and Y. Han, "Incorporating multiple SVMs for automatic image annotation," *Pattern Recognit.*, vol. 40, no. 2, pp. 728–741, 2007.

[23] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, 2010.

[24] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, 2006.

[25] N. Du, B. Wu, X. Pei, B. Wang, and L. Xu, "Community detection in large-scale social networks," in *Proc. 9th WebKDD, 1st SNA-KDD Workshop Web Mining Soc. Netw. Anal.*, 2007, pp. 16–25.

[26] S. Papadopoulos *et al.*, "Image clustering through community detection on hybrid image similarity graphs," in *Proc. 17th IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2010, pp. 2353–2356.

[27] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Jan. 2003.

[28] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan, "Matching words and pictures," *J. Mach. Learn. Res.*, vol. 3, pp. 1107–1135, Mar. 2003.

[29] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *J. Mach. Learn. Res.*, vol. 7, pp. 1531–1565, Jul. 2006.

[30] Q. Li, Y. Gu, and X. Qian, "LCMKL: Latent-community and multikernel learning based image annotation," in *Proc. 22nd ACM CIKM*, 2013, pp. 1469–1472.

[31] Z. Younes, F. Abdallah, and T. Denœux, "Multi-label classification algorithm derived from K-nearest neighbor rule with label dependencies," in *Proc. 16th Eur. Signal Process. Conf.*, Aug. 2008, pp. 1–5.

[32] M. Wang, X. Zhou, and T.-S. Chua, "Automatic image annotation via local multi-label classification," in *Proc. Int. Conf. Content-Based Image Video Retr.*, 2008, pp. 17–26.

[33] Y. Yu, W. Pedrycz, and D. Miao, "Neighborhood rough sets based multi-label classification for automatic image annotation," *Int. J. Approx. Reason.*, vol. 54, no. 9, pp. 1373–1387, 2013.

[34] M. Kutas and S. A. Hillyard, "Brain potentials during reading reflect word expectancy and semantic association," *Nature*, vol. 307, no. 5947, pp. 161–163, 1984.

[35] A. E. Elisseeff and J. Weston, "A kernel method for multilabelled classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 681–687.

[36] M.-L. Zhang, J. M. Peña, and V. Robles, "Feature selection for multi-label naive Bayes classification," *Inf. Sci.*, vol. 179, no. 19, pp. 3218–3229, 2009.

[37] F. A. Thabtah, P. Cowling, and Y. Peng, "MMAC: A new multiclass, multi-label associative classification approach," in *Proc. 4th IEEE ICDM*, Nov. 2004, pp. 217–224.

[38] M.-L. Zhang, "Lift: Multi-label learning with label-specific features," in *Proc. IJCAI*, 2011, pp. 1609–1614.

[39] X. Xue, W. Zhang, J. Zhang, B. Wu, J. Fan, and Y. Lu, "Correlative multi-label multi-instance image annotation," in *Proc. IEEE ICCV*, Nov. 2011, pp. 651–658.

[40] W. Zhang, X. Xue, J. Fan, X. Huang, B. Wu, and M. Liu, "Multi-kernel multi-label learning with max-margin concept network," in *Proc. 22nd Int. Joint Conf. Artif. Intell. (IJCAI)*, vol. 22. 2011, pp. 1615–1620.

[41] C.-T. Nguyen, D.-C. Zhan, and Z.-H. Zhou, "Multi-modal image annotation with multi-instance multi-label LDA," in *Proc. IJCAI*, 2013, pp. 1558–1564.

[42] X. Li and C. G. M. Snoek, "Classifying tag relevance with relevant positive and negative examples," in *Proc. 21st ACM Multimedia*, 2013, pp. 485–488.

[43] X. Qian, X.-S. Hua, Y. Y. Tang, and T. Mei, "Social image tagging with diverse semantics," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2493–2508, Dec. 2014.

[44] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world Web image database from National University of Singapore," in *Proc. ACM CIVR*, 2009, Art. ID 48.

[45] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Statist. Mech., Theory Experim.*, vol. 2008, no. 10, p. P10008, 2008.

[46] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *J. Mach. Learn. Res.*, vol. 9, pp. 2491–2521, Nov. 2008.

[47] M. Grubinger, P. Clough, H. Müller, and T. Deselaers, "The IAPR TC-12 benchmark: A new evaluation resource for visual information systems," in *Proc. Int. Workshop OntoImage*, 2006, pp. 13–23.

[48] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang, "Tag ranking," in *Proc. WWW*, 2009, pp. 351–360.

[49] M. Guillaumin, "Exploiting multimodal data for image understanding," Ph.D. dissertation, Inst. Nat. Polytechn. Grenoble, Grenoble, France, 2010.

[50] X. Li, C. G. M. Snoek, and M. Worring, "Learning social tag relevance by neighbor voting," *IEEE Trans. Multimedia*, vol. 11, no. 7, pp. 1310–1322, Nov. 2009.

[51] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 3304–3311.

[52] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. (2014). "CNN features off-the-shelf: An astounding baseline for recognition." [Online]. Available: http://arxiv.org/abs/1403.6382

[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*. Red Hook, NY, USA: Curran Associates, Inc., 2012, pp. 1106–1114.

[54] P. Ronhovde and Z. Nussinov, "Local resolution-limit-free Potts model for community detection," *Phys. Rev. E*, vol. 81, no. 4, p. 046114, 2010.

[55] A. Arenas, A. Fernandez, and S. Gomez, "Analysis of the structure of complex networks at different resolution levels," *New J. Phys.*, vol. 10, no. 5, p. 053039, 2008.

[56] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, vol. 1. 2009, pp. 248–256.

[57] X. Xu, A. Shimada, and R.-I. Taniguchi, "Latent topic model for image annotation by modeling topic correlation," in *Proc. IEEE ICME*, Jul. 2013, pp. 1–6.

[58] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.

[59] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *J. Mach. Learn. Res.*, vol. 5, pp. 975–1005, Dec. 2004.

Yun Gu (S'12) received the B.S. degree from Xi'an Jiaotong University, in 2013. He is currently pursuing the M.S. degree with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University. He was a Visiting Student with the SMILES Laboratory from 2012 to 2013.

His research interests include image annotation, cross-modal hashing, and medical image analysis.

Xueming Qian (M'10) received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the Ph.D. degree from the School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, in 2008. He was an Assistant Professor. He was an Associate Professor from 2011 to 2014, and is currently a Full Professor. He was a Visiting Scholar with Microsoft Research Asia from 2010 to 2011. He is the Director of the SMILES Laboratory. His research is supported by NSFC, Microsoft Research, and MOST. His research interests include social media big data mining and search. He received the Microsoft Fellowship in 2006. He received the Outstanding Doctoral Dissertations of Xi'an Jiaotong University and Shaanxi Province in 2010 and 2011, respectively.

Qing Li received the B.S. degree in automation from Xi'an Jiaotong University, China, in 2013. He is currently pursuing the Ph.D. degree in urban computing with the University of Wisconsin, Madison.

His research interests include machine learning, pattern recognition, urban computing, and transportation big data analytics, e.g., mining cellular and taxi probe data.

Meng Wang (M'09) received the B.E. degree from the Special Class for the Gifted Young and the Ph.D. degree from the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China. He is currently a Professor with the Hefei University of Technology, China. His current research interests include multimedia content analysis, search, mining, recommendation, and large-scale computing. He received the best paper awards successively from the 17th and 18th ACM International Conference on Multimedia, the best paper award from the 16th International Multimedia Modeling Conference, the best paper award from the 4th International Conference on Internet Multimedia Computing and Service, and the Best Demo Award from the 20th ACM International Conference on Multimedia.

Richang Hong (M'12) received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2008. He was a Research Fellow with the School of Computing, National University of Singapore, from 2008 to 2010. He is currently a Professor with the Hefei University of Technology, Hefei. He has co-authored more than 60 publications in the areas of his research interests, which include multimedia question answering, video content analysis, and pattern recognition. He is a member of the Association for Computing Machinery. He was a recipient of the best paper award in the ACM Multimedia 2010.

Qi Tian (M'96–SM'03) received the B.E. degree in electronics engineering from Tsinghua University, China, in 1992, the M.S. degree in electrical and computer engineering from Drexel University, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign, in 2002. He is currently a Professor with the Department of Computer Science, University of Texas at San Antonio (UTSA).

He took a one-year faculty leave with Microsoft Research Asia from 2008 to 2009. His research projects were funded by NSF, ARO, DHS, SALSI, CIAS, and UTSA. He has authored over 290 refereed journal and conference papers. His research interests include multimedia information retrieval and computer vision. He received faculty research awards from Google, the NEC Laboratories of America, FXPAL, Akiira Media Systems, and HP Labs. He was a co-author of the ACM ICMCS 2012 Best Paper, the MMM 2013 Best Paper, the PCM 2013 Best paper, the Top 10% Paper Award in MMSP 2011, the Best Student Paper in ICASSP 2006, and the Best Paper Candidate in PCM 2007. He received the 2010 ACM Service Award.

Dr. Tian has served as the Program Chair, an Organization Committee Member, and a TPC for numerous IEEE and ACM conferences, including the ACM Multimedia, SIGIR, ICCV, and ICME. He is on the Editorial Board of the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON CIRCUIT AND SYSTEMS FOR VIDEO TECHNOLOGY, the *Multimedia System Journal*, the *Journal of Multimedia*, and the *Journal of Machine Visions and Applications*. He is also a Guest Editor of the IEEE TRANSACTIONS ON MULTIMEDIA, the *Journal of Computer Vision and Image Understanding*, *Pattern Recognition Letter*, the *EURASIP Journal on Advances in Signal Processing*, and the *Journal of Visual Communication and Image Representation*.