Neurocomputing 95 (2012) 105–116

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Nearest-neighbor method using multiple neighborhood similarities for social media data mining

Shuhui Wang [a,*], Qingming Huang [a,b], Shuqiang Jiang [a], Qi Tian [c], Lei Qin [a]

[a] *Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China*
[b] *Graduate University, Chinese Academy of Sciences, Beijing 100049, China*
[c] *Department of Computer Science, University of Texas at San Antonio, TX 78249, USA*

ARTICLE INFO

Available online 29 March 2012

Keywords:
Nearest neighbor method
Multiple neighborhood similarity
Visual categorization
Locality sensitive hashing

ABSTRACT

Currently, Nearest-Neighbor approaches (NN) have been applied to large scale real world image data mining. However, the following three disadvantages prevent them from wider application compared to other machine learning methods: (i) the performance is inferior on small datasets; (ii) the performance will degrade for data with high dimensions; (iii) they are heavily dependent on the chosen feature and distance measure. In this paper, we try to overcome the three mentioned intrinsic weaknesses by taking the abundant and diversified content of social media images into account. Firstly, we propose a novel neighborhood similarity measure which encodes both the local density information and semantic information, thus it has better generalization power than the original image-to-image similarity. Secondly, to enhance the scalability, we adopt kernelized Locality Sensitive Hashing (KLSH) to conduct approximated nearest neighbor search by utilizing a set of kernels calculated on several complementary image features. Finally, to enhance the robustness on diversified genres of images, we propose to fuse the discrimination power of different features by combining multiple neighborhood similarities calculated on different features/kernels with the entire retrieved nearest labeled and unlabeled image via the hashing systems. Experimental results on visual categorization on the Caltech-256 and two social media databases show the advantage of our method over traditional NN methods using the labeled data only.

1. Introduction

During the past decade, social media for social interaction have developed into a very important media on the web. According to the statistics in [1], social networking now accounts for 22% of all time spent online in the USA. For every second, there are millions of people acquiring and sharing various kinds of information on web sites such as Flickr [2], YouTube [3] and Twitter [4].

Among all kinds of information carrier, image and video are believed to convey more vivid life experiences than text. Due to the popularity of digital camera and the social networking, photographs and videos can be easily produced by ordinary users and shared online. However, facing with explosively growing web images and videos, we cannot effectively retrieve and utilize the web image database without effective data mining tools, including image categorization [9,11,17,28,34,37–39,40,46–48], image tagging [45], image re-ranking [36] and video enhancement [24].

As one of the most important image data mining research, automatic image categorization has drawn considerable attention during the past few decades. The significant endeavors made in the research community have resulted in many novel and effective approaches. For typical dataset such as Caltech-101, the classification accuracies of state-of-the-art methods have been improved from 20% to almost 90% during the past few years [9,38].

Among the existing approaches, a well-studied paradigm for image classification is offline learning based approaches. The classification model is generated offline using the whole set of training data, which requires an intensive training step (for example, SVM [17], Boosting [46] and distance metric learning [21,45]). Another is the lazy learning paradigm, which requires no training step on model parameters, and the prediction is not given until a query is made into the system. The most common lazy learning approach is k-Nearest Neighbor Classification (k-NN), which classifies an image by the class of its most similar images in the database.

Compared with the offline learning approaches, lazy learning approaches have several advantages: (i) no training and learning step is required; (ii) no over-fitting issues should be considered; (iii) they can naturally handle thousands of image classes and millions of images, which is more suitable for the application of

[*] Corresponding author.
E-mail addresses: shwang@jdl.ac.cn (S. Wang), qmhuang@jdl.ac.cn (Q. Huang), sqjiang@jdl.ac.cn (S. Jiang), qitian@cs.utsa.edu (Q. Tian), lqin@jdl.ac.cn (L. Qin).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2011.06.039

social media data mining with billions of data; (iv) the classification model can be easily updated by replacing the examples in the database instead of expensive model re-training. Furthermore, when the number of labeled images in the database is large enough, the error rate of Nearest-Neighbor approaches converges to the optimal Bayes error rate [7], which provides a theoretical foundation for NN methods. However, Nearest-Neighbor approaches usually achieve inferior performance than offline learning approaches in many scenarios [9]. To improve the robustness of NN methods so as to perform well on large scale real world image categorization, we address the weakness of them from three aspects in this paper.

Firstly, NN approaches are weak on small size database. When the data size is small, the true data distribution under certain feature representation could not be well approximated because of the sparseness in high dimensional space. In this situation, the image-to-image distance will be sensitive to various types of appearance and illumination variation, which would easily influence the local neighborhood structure. Therefore, given the query, the ratio of the retrieved nearest neighbor images with consistent semantic information will be reduced, which is the main issue that determines the model capacity of NN. Previously, there were mainly two major solutions for this problem. The first is increasing the "hit rate" of semantically consistent images in the retrieved subset by enlarging the size of database with labeled or weakly labeled images [19,9], and the second directly enhances the semantic consistency of the original feature representation by learning a new distance metric or space transformation using the side information in the image data [15,20,21,43,45].

Considering the fact that increasing the number of labeled image spends a lot more computation and annotation efforts compared with increasing the number of unlabeled image, we improve the performance of NN approaches under limited number of labeled images by designing a more robust similarity measures in this study. We address the problem from two aspects. To enhance the semantic consistency of the similarity measure, an offline distance metric learning is conducted to obtain a *Mahalanobis* matrix for calculating the pair-wise similarity. To improve the robustness to appearance variation and noise, the local data density information is encoded into the similarity measure by using the large scale unlabeled and untagged web images, since they could be used to approximate the distribution in the feature space. Specifically, the proposed neighborhood similarity between query images and labeled images in the database is calculated with both the image-to-image similarity between them and their neighboring unlabeled data. The neighborhood similarity provides more semantic consistent and noise-free description of the similarity among images than the original image-to-image similarity, so that the model capacity will be enhanced.

The second aspect that we consider in this paper is designing a scalable nearest neighbor search system on large scale real world image database. Generally, compared with precise nearest neighbor search which usually requires at least $O(N\log(N))$ time in complexity, conducting approximated nearest neighbor search is a better choice as it usually requires sub-linear time. Among all the approximated approaches, locality sensitive hashing (LSH) [12,14] is believed to perform well on high dimensional data by using random projection functions to project each data item into a set of binary codes. However, when doing hashing on data with tens of thousands of dimensions, like any other approximated nearest neighbor methods, LSH takes risk to degrade since the data structure with high dimensional representation is extremely sparse. Therefore, we adopt a kernelized locality sensitive hashing (KLSH) [27]. The hash function is constructed using the similarities among data items instead of the original feature, which avoids the "*curse-of-dimensionality*" problem by kernel trick. Another advantage of hashing with kernel endows ability to improve the approximated nearest neighbor searching by using arbitrary kernel representation, including those learned by any state-of-the-art distance metric or kernel learning approaches. The complexity of KLSH is almost the same as the original LSH, which implies that KLSH inherits the efficiency of LSH.

Finally, NN approaches are very sensitive to the chosen feature and distance measure. According to our observation, different features and distance measures provide different descriptive ability for different genres of images. For example, color feature is good at describing sports images, texture feature is descriptive for object images, and dense bag-of-word feature performs consistently well for discriminating scene images. The limitation of using one feature only constrains NN approaches to apply on general image categorization tasks. To overcome this weakness, we construct KLSH on a set of features and kernel representations instead of one, and the nearest neighbors returned by the system using different features are used to calculate the neighborhood similarity respectively on each feature channel. Then the final similarity measure between any two images is the weighed combination of the neighborhood similarity on different features. This is similar to the Multiple Kernel Learning [5,31,35] in spirit, where the outputs are formed by using different features/kernels. Therefore, more generalization power can be obtained, and the overall error would be reduced by combining the discrimination power of multiple neighborhood similarities. This statement will be proved by experimental evaluation and theoretical analysis in Section 2.7.

In this paper, we propose a new Nearest-Neighbor image classification method, which is based on the studies in our previous work [41]. We extend our work by incorporating the issue of semantic consistence, and provide some theoretic view to prove that the error rate of our combined method will be lower than traditional NN approaches with single feature. Moreover, we conduct experiment on social media image datasets to show the potential of our method on social media data mining.

In general, the key contributions are presented in four aspects: (1) we propose a neighborhood similarity measure for Nearest-Neighbor classification, which encodes the local density information by using unlabeled data and semantic consistence by incorporating distance metric learning; (2) we propose to combine the discrimination power of different features to form the final decision output of an unknown sample, which enhances the robustness for processing the real world data; (3) we provide theoretic analysis to demonstrate how k-NN using multiple neighborhood similarity outperforms k-NN using on single feature and image-to-image distance; (4) we construct a practical system that is able to perform real world social media image categorization.

The rest of this paper is organized as follows: In Section 2, we describe every details of our method. In Section 3, we conduct extensive experiment on three distinguished image database to evaluate our method. In Section 4, we provide a brief review of the related works. Section 5 provides conclusions and discussion on future works.

2. Method

2.1. Overview

The framework proposed in this paper is described in Fig. 1. Firstly, a set of kernelized LSH systems are constructed both on labeled and unlabeled data based on the learned kernels for each feature channel as introduced in Sections 2.3 and 2.5. A query image is fed into the system and all the nearest neighbors found

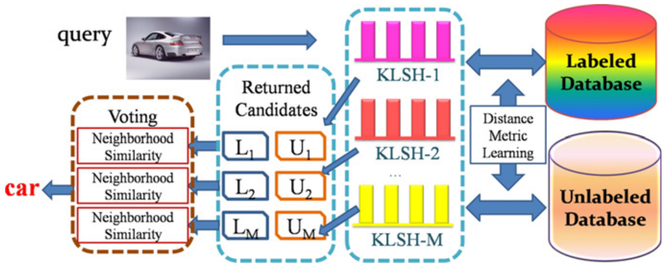

Fig. 1. System work flow of our method.

by different KLSH are returned. Then the similarity between query image and the returned nearest labeled data is calculated using the learning method introduced in Sections 2.2, 2.3 and 2.4. The final decision output is determined by the scheme introduced in Section 2.6. We describe each part in detail.

2.2. Neighborhood similarity

Generally, a better similarity measure should be able to encode the local density and manifold information. The motivation of the neighborhood similarity measure is to use the unlabeled data to approximate the true data distribution in the unknown kernel space. Our method is based on the following assumptions:

(a) Data points with similar local density are likely to be more similar than data points with different local density.
(b) The similarity among data points on dense manifolds tend to be larger than data points on sparse manifolds.

These two assumptions are consistent with the manifold assumption [50,32,33] and cluster assumption [44,50] used in many semi-supervised literature. Specifically, given a fixed feature representation which corresponds to a certain kernel K, we represent a sample $\mathbf{x}$ by the linear combination of its own implicit representation with respect to the kernel K and its average of neighboring representation as:

$$\Phi_N(\mathbf{x}) = \alpha_0 \Phi_O(\mathbf{x}) + \frac{(1-\alpha_0)}{|Nbd(\mathbf{x})|} \sum_{x' \in Nbd(\mathbf{x})} \Phi_O(\mathbf{x}') \tag{1}$$

where $Nbd(*)$ denotes the neighborhood unlabeled sample set, and $\Phi_O(\mathbf{x})$ denotes the feature representation in the high dimensional Hilbert space given the kernel.

Based on the expression in Eq. (1), the similarity of query $\mathbf{x}$ and the labeled examples $\mathbf{y}$ are calculated by the weighted averaging of the original kernel value $K_O(\mathbf{x},\mathbf{y})$ and the neighborhood kernel values in the following equivalent form:

$$K_N(\mathbf{x},\mathbf{y}) = \alpha K_O(\mathbf{x},\mathbf{y}) + (1-\alpha)\frac{\sum K_O(\mathbf{x}',\mathbf{y}')}{|Nbd(\mathbf{x})| \bullet |Nbd(\mathbf{y})|},$$
$$\mathbf{x}' \in Nbd(\mathbf{x}), \quad \mathbf{y}' \in Nbd(\mathbf{y}), \quad \mathbf{x}',\mathbf{y}' \in U \tag{2}$$

where $K_O(\mathbf{x},\mathbf{y}) = \langle \Phi_O(\mathbf{x}), \Phi_O(\mathbf{y}) \rangle$, U denotes the unlabeled data. α is the weight parameter. We set α=0.5 empirically. The modified feature representation is similar to the cluster center in the convex hull formed by the linear combination of feature representation from the data itself and the neighborhood unlabeled samples. Under this representation, different clusters rather than different samples become better separated from each other. Compared with the image-to-image similarity, the neighborhood similarity measure provides better discrimination power for a set of samples instead of only one sample. Therefore, it is more robust to noise and small image variations and noise.

2.3. Distance metric learning

To improve the semantic correspondence of the proposed neighborhood similarity measure, we conduct distance metric learning on each feature representation. The target of distance metric learning is to learn a matrix $\mathbf{A}=\mathbf{L}^T\mathbf{L}$ for the *Mahalanobis* distance as:

$$d_{\mathbf{A}}(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T \mathbf{L}^T \mathbf{L} (\mathbf{x}-\mathbf{y})} \tag{3}$$

The learned metric should satisfy that the distance between any pairs of data from the same class should be nearer than those pairs from different classes. To this end, we minimize the following objective function with large margin constraints as proposed in [43]:

$$\min \textstyle\sum_{j \mapsto i} \left[d^2_{\mathbf{A}}(\mathbf{x}_i,\mathbf{x}_j) + \mu \sum_l (1-y_{il}) \xi_{ijl} \right]$$
$$\text{s.t.} \quad d^2_{\mathbf{A}}(\mathbf{x}_i,\mathbf{x}_l) - d^2_{\mathbf{A}}(\mathbf{x}_i,\mathbf{x}_j) \geq 1 - \xi_{ijl}$$
$$\xi_{ijl} \geq 0, \quad \mathbf{A} \succeq 0 \tag{4}$$

The method maximizes the margin between the distance of local sample from the same class and the distance of samples from different class. In fact, the *LMNN* metric learning approach is only able to achieve acceptable training time on datasets with hundreds of feature dimension. When learning the distance metric on high dimensional feature representation, for example, the multi-level spatial pyramid feature with more than 20 K dimensions, the training time cost as well as the memory consumption become prohibitive. Therefore, for features with more than 1 K dimensions, we first conduct PCA to reduce the dimensions to 300.

Consequently, after we learn the matrix $\mathbf{A}$, the kernel value (similarity) between $\mathbf{x}$ and $\mathbf{y}$ is represented by:

$$K_{\mathbf{A}}(\mathbf{x},\mathbf{y}) = K(\mathbf{L}\mathbf{x},\mathbf{L}\mathbf{y}) \tag{5}$$

where $K(\bullet,\bullet)$ represents any kernel form, such as inner product, RBF kernel or Gaussian kernel. For notation convenience, we denote the learned kernel using distance metric learning in (5) for mth feature channel as $K_L^{(m)}$. Consequently, the neighborhood similarity $K_N^{(m)}$ of mth feature channel is:

$$K_N^{(m)}(\mathbf{x},\mathbf{y}) = \alpha K_L^{(m)}(\mathbf{x},\mathbf{y}) + (1-\alpha)\frac{\sum K_L^{(m)}(\mathbf{x}',\mathbf{y}')}{|Nbd^{(m)}(\mathbf{x})| \bullet |Nbd^{(m)}(\mathbf{y})|}$$
$$\mathbf{x}' \in Nbd^{(m)}(\mathbf{x}), \mathbf{y}' \in Nbd^{(m)}(\mathbf{y}), \mathbf{x}',\mathbf{y}' \in U \tag{6}$$

2.4. Multiple neighborhood similarity

As discussed in the previous section, the original NN approaches based on single feature is not capable of dealing with various genres of images from real world. Recent studies on machine learning have proved that employing multiple features/kernels can improve the discrimination power of the model [5,11,28,31,35,38,47]. Specifically, the multiple neighborhood similarity is described as the weighted average of the neighborhood similarity on single feature:

$$K_N(\mathbf{x},\mathbf{y}) = \sum_{m=1}^{M} w_m K_N^{(m)}(\mathbf{x},\mathbf{y}), \quad \text{s.t.} \quad w_m \geq 0, \sum_{m=1}^{M} w_m = 1 \tag{7}$$

where w_m is a group of pre-calculated weight coefficients. This similarity measure is similar to the form of kernel in multiple kernel learning [5,31,35], where the kernel $K(\mathbf{x}_i,\mathbf{x}_j)$ and model $f(\mathbf{x})$ are:

$$K(\mathbf{x}_i,\mathbf{x}_j) = \sum_{m=1}^{M} \beta_m K_m(\mathbf{x}_i,\mathbf{x}_j), \quad \text{s.t.} \quad w_m \geq 0, \sum_{m=1}^{M} w_m = 1$$
$$f(\mathbf{x}) = \sum_i \left(\alpha_i y_i \sum_{m=1}^{M} \beta_m K_m(\mathbf{x}_i,\mathbf{x}) \right) \tag{8}$$

Different from [5,31,35], we do not train the model offline to obtain the model parameter α and β with some optimization procedure. The most time consuming offline procedure in our method is to conduct distance metric learning to learn a better kernel $K_L^{(m)}$ for each feature channel. Then the final decision output is calculated directly by the voting scheme, which will be described in Section 2.6.

2.5. Kernelized locality sensitive hashing

To efficiently conduct approximated nearest neighbor search based on arbitrary kernel representation, we adopt the recent developed kernelized locality sensitive hashing method [27]. The intuitive of kernelized LSH is to perform LSH in an unknown high dimensional kernel space. With similar theorem used in Kernel PCA [27], the hashing function for mth feature channel is written as:

$$h^{(m)}(\phi(x)) = sign\left(\sum_{i=1}^{P} \mathbf{w}(i)K^{(m)}(x_i,x)\right) \tag{9}$$

where $\phi(x)$ denotes the unknown representation in high dimensional space. In the original LSH approaches [12,14], randomized projection is conducted to project the original feature into a set of binary codes by using a family of randomized functions. For KLSH, the randomized projection is simulated approximately by a subset of data items. Specifically, the weight vector $\mathbf{w}$ is calculated as:

$$\mathbf{w} = (K^{(m)})^{-1/2}\left(\frac{1}{T}\mathbf{e}_s - \frac{1}{P}\mathbf{e}\right),$$
$$K^{(m)} = V^{(m)}\Lambda^{(m)}(V^{(m)})^T, (K^{(m)})^{-1/2} = V^{(m)}(\Lambda^{(m)})^{-1/2}(V^{(m)})^T \tag{10}$$

where $K^{(m)}$ is the kernel matrix of the randomly chosen P items of the whole database, and each element can be calculated using (5). The eigenvector and eigenvalues of $K^{(m)}$ using SVD are represented by $V^{(m)}$ and $\Lambda^{(m)}$, respectively. P is usually very small (we set $P=300$ for our experiments unless special statement is given) compared with the whole data size. $\boldsymbol{e}$ is a vector with P ones and $\boldsymbol{e_s}$ is an indicator vector for a subset S from the P items:

$$\mathbf{e}_s(i) = \begin{cases} 1, & \text{if} \quad i \in S \\ 0 & \text{else} \end{cases}, \quad i = 1,..P \tag{11}$$

The size of S is T. In this paper we set $T=30$. A set of hash functions could be obtained by randomly choosing the subset S.

According to the analysis in [27], the effect of the randomized projection on unknown feature space can be guaranteed by large number theory. The larger P and T are, the better the randomized projection will be. However, for tradeoff of efficiency and effectiveness, we declare that the setting of $P=300$ and $T=30$ is a reasonable choice.

In this paper, to improve the recall of the nearest neighbor search, we construct 3 hash tables for each feature channel. The candidate nearest neighbor subset is the union of retrieved images using these 3 hash tables. Then the nearest neighbors are identified as the top items with largest kernel values from the sorted list. We describe how to conduct decision on unknown image in Section 2.6.

2.6. Decision output

Based on KLSH, we firstly identify those nearest neighbor samples from the database. Given a query $\boldsymbol{x}$, for each KLSH system, the candidate nearest neighbor samples are those examples whose hash codes are in the most similar buckets with the query samples. Since we must guarantee that both the nearest labeled and unlabeled data can be chosen, we respectively retrieve the labeled and unlabeled items in the Top 3 nearest buckets, considering the distribution difference of the hash code representation. The fact that the chosen labeled and unlabeled data are neighboring samples to each other can be guaranteed by triangle inequality:

$$d(\mathbf{y},\mathbf{z}) \le d(\mathbf{x},\mathbf{z}) + d(\mathbf{x},\mathbf{y}) \tag{12}$$

where $\mathbf{x}$ denotes the query, $\mathbf{y}$ and $\mathbf{z}$ denotes any retrieved nearest sample candidate from the labeled and unlabeled sample, respectively.

After we have obtained $(N_L^1,...,N_L^M)$ nearest labeled candidate sample and $(N_U^1,...,N_U^M)$ nearest unlabeled candidate samples given a query $\mathbf{x}$, the set of the nearest labeled samples are identified as the top B_L items with largest kernel values. Therefore, the final nearest labeled samples sets are $N_{LC} = unique(N_{LC}^1,...,N_{LC}^M)$, where $|N_{LC}^m| = \min(|N_L^m|,B_L)$. For calculating the neighborhood similarity measure, we should identify those nearest unlabeled samples for both the query $\boldsymbol{x}$ and the selected labeled subset N_{LC}. For finding neighborhood of $\mathbf{x}$, we can easily obtain nearest unlabeled samples for each feature channel by choosing the top B_U items with largest kernel values. For finding the neighborhood of a labeled sample $\boldsymbol{y}$ from N_{LC}, the most direct solution is by treating $\mathbf{y}$ as a query to find those unlabeled sample candidates. However, this is not only time consuming, but also requiring extra memory to store the retrieved sample subset for each $\boldsymbol{y}$ from N_{LC}, which usually contains more than one hundred labeled samples and 10 K unlabeled samples. Therefore, we select the top B_U items from N_U^m for each $\boldsymbol{y}$ as the neighborhood instead. The selected neighborhood can be a good approximation of the real neighborhood, since the triangle inequality ensure that the samples from N_U^m are similar with those from N_{LC} with high probability.

Since we have identified the neighborhood information for the given query and the retrieved nearest neighboring labeled images, we calculate the neighborhood similarities for each feature channel between query $\boldsymbol{x}$ and the labeled sample set N_{LC}. If all the returned labeled samples come from the same class, the decision output is directly assigned with this class index. Otherwise, the decision output of query $\boldsymbol{x}$ is calculated as follows:

$$\begin{aligned} C &= \arg\max_Q \frac{1}{|N_{LC(Q)}|} \sum_{j=1}^{|N_{LC(Q)}|} K_N(\mathbf{x},\mathbf{x}_{j,Q}) \\ &= \arg\max_Q \frac{1}{|N_{LC(Q)}|} \sum_{j=1}^{|N_{LC(Q)}|} \sum_{m=1}^{M} w_m K_N^{(m)}(\mathbf{x},\mathbf{x}_{j,Q}) \\ \mathbf{x}_{j,Q} &\in N_{LC(Q)}, \quad Q \in [1,N_C], \quad \sum_{m=1}^{M} w_m = 1 \end{aligned} \tag{13}$$

where $N_{LC(Q)}$ denotes the set of Qth class samples in N_{LC}. $N_{LC(Q)}$ is used to reduce the influence of imbalanced number of retrieved labeled images. The weight w_m of mth feature channel determined by experiment on the validation set, which is a subset of unlabeled data with human labeled ground truth. The whole procedure is described in Algorithm 1.

Algorithm 1. The proposed nearest neighbor classification procedure

```
Settings: A labeled dataset $L$ with $N_c$ classes of images; An
  unlabeled dataset $U$. $M$ features/kernels.
Procedure:
1. Build $M$ kernelized LSH systems on both labeled data and
   unlabeled data for each features/kernels.
2. For each test query sample $\boldsymbol{x}$:
     (a) Obtain $(N_L^1,...,N_L^M)$ labeled candidates and $(N_U^1,...,N_U^M)$
         unlabeled candidates from the database.
     (b) Obtain the labeled subset $(N_{LC}^1,...,N_{LC}^M)$ and identify the
         neighborhood of $\boldsymbol{x}$ and $(N_{LC}^1,...,N_{LC}^M)$.
     (c) If all the returned labeled samples come from
         one class.
```

```
            return Class index for $\boldsymbol{x}$
        Else Calculate the similarity between $q$ and the
        labeled data using Eq. (13).
        End
    End For
```

2.7. Theoretic analysis

In this section, we provide some theoretic analysis on the error bound of our method. Suppose the k-NN model on each distinguished feature $f_m(x)$ satisfies:

$$Y(x) = f_m(x) + \varepsilon_m,$$
$$\varepsilon_m \sim N(0,\sigma_m^2), \quad m = 1,\ldots,M \tag{14}$$

where the noise ε_m is independent with each other. Then the error of each $f_m(x)$ can be represented by:

$$E_m = \mathrm{E}_x[(f_m(x) - Y(x))^2] = \sigma_m^2, \quad m = 1,\ldots,M \tag{15}$$

The error of the combined k-NN model in our study can be written as:

$$\overline{E} = \min_{\mathbf{w}} \mathrm{E}_x \left(\sum_m w_m f_m(x) - Y(x) \right)^2 \quad \text{s.t.} \quad \sum_m w_m = 1 \tag{16}$$

Since each ε_m is independent, we have:

$$\mathrm{E}_x\left[\left(\sum_m w_m f_m(x) - Y(x) \right)^2 \right] = \sum_m w_m^2 \mathrm{E}_x (f_m(x) - Y(x))^2$$
$$+ 2 \sum_{m_1 \neq m_2} w_{m_1} w_{m_2} \mathrm{E}_x[(f_{m_1}(x) - Y(x))(f_{m_2}(x) - Y(x))] = \sum_m w_m^2 \sigma_m^2 \tag{17}$$

The Lagrange of Eq. (16) is:

$$L = \sum_m w_m^2 \sigma_m^2 + \lambda \left(\sum_m w_m - 1 \right)$$
$$\frac{\partial L}{\partial w_m} = 2 w_m \sigma_m^2 + \lambda = 0 \rightarrow w_m = -\frac{\lambda}{2\sigma_m^2} \tag{18}$$

We put w_m back to the Lagrange:

$$L = \sum_m w_m^2 \sigma_m^2 + \lambda \left(\sum_m w_m - 1 \right) = \lambda^2 \sum_m \frac{1}{4\sigma_m^2} - \lambda^2 \sum_m \frac{1}{2\sigma_m^2} - \lambda$$
$$= -\left(\lambda^2 \sum_m \frac{1}{4\sigma_m^2} + \lambda \right)$$

And then we have the following unconstrained dual problem:

$$\overline{E} = \min_{\lambda} -\left(\lambda^2 \sum_m \frac{1}{4\sigma_m^2} + \lambda \right) \tag{19}$$

We can easily conclude that the optimal value of $\overline{E}$ can be obtained as:

$$\lambda = -\frac{1}{\sqrt{\sum_m 1/\sigma_m^2}}, \quad \overline{E} = \frac{1}{\sum_m 1/\sigma_m^2} \tag{20}$$

We show that $\overline{E} < \min_m E_m$, because for any E_m:

$$\overline{E} = \frac{1}{\sum_m 1/\sigma_m^2} < E_m = \sigma_m^2 \Leftrightarrow 1 < \left(\sum_{m'} \frac{1}{\sigma_{m'}^2} \right) \sigma_m^2 = 1 + \sum_{m' \neq m} \frac{\sigma_m^2}{\sigma_{m'}^2} \tag{21}$$

From the analysis we can see that, the generalization error of the combined model in our study is less than the minimal error of any single model $f_m(x)$, if each feature channel is independent with each other. When more features are involved and their prediction error σ_m is small enough, the denominator of $\overline{E}$ on the left of the equal sign in Eq. (21) will become larger, and the error $\overline{E}$ will be smaller. This also provides the guidance on how we choose the features for constructing the system. Generally, more features are desirable when they are able to describe different aspects of the images so as to make the model $f_m(x)$ uncorrelated with each other. Moreover, the generalization error of the combined model is determined by the generalization error lower bound of each $f_m(x)$. Therefore, it is reasonable to improve the performance of each $f_m(x)$ by techniques such as distance metric learning in this paper.

In real world application scenario, we will show in the experiment that even when the independence condition cannot be guaranteed, the performance of the combined model still outperforms k-NN with single feature.

3. Experiments

We conduct experiments to evaluate the performance of our method. The details of overall experiment setup are shown in Table 1. We run all the experiments on a desktop computer, which does not need a lot of computational resources. We evaluate our method on three different real world image classification problems. Dataset 1 is a challenging visual object categorization dataset with 256 object classes. Dataset 2 is a large scale image dataset collected from Flickr, which involves classification or retrieval tasks on dozens of common semantic classes such as sky, water, building and person. Dataset 3 is a web image dataset with well labeled ground truth for 81 semantic concepts.

3.1. Experiment on Caltech-256 using social media unlabeled data

In this part of experiment, we use the Caltech-256 as the labeled dataset, and download more than 250 K unlabeled images from Flickr [2] as the unlabeled dataset. Five features and kernels describing different property of images are used, including texture [10], color, Bag-of-Words [10] and global feature [37] as described in Table 1.

We randomly select 5, 10, 15, 20, 25 and 30 labeled samples from each class of Caltech-256 as the labeled data, and randomly

Table 1
Experimental setup description.

Dataset 1:
L and *T*: Caltech-256: image dataset with 256 objects. Size: 30607.
U: Web image data: collected from flickr.com with 256 object categories. The dataset is downloaded using the same class names as Caltech-256. Size: 251921.

Dataset 2:
L and *T*: MIR-FLICKR-25 K: image dataset with 23 classes of images. Size: 25 K.
U: MIR-FLICKR-1 M: images which does not include the MIR-FLICKR-25 K subset. Size: 975000.

Dataset 3:
L: NUS-WIDE: a randomly chosen subset of the training data. Size: 80894 (50%).
U: NUS-WIDE: the rest of the training data. Size: 80895 (50%).
T: NUS-WIDE: test data. Size: 107859.
L: the labeled training data. *U*: the unlabeled data. *T*: the test data.

Environment:
OS: Windows XP; Computer: Lenovo Think Center M6000t desktop; CPU: Intel(R) Core2 Duo E7500 @3.00 GHz; Memory: 4.0 G RAM; Programming platform: Matlab R2009.

Features and similarity measures used for Dataset 1 and 2:
(1) 3 level PHOG-180 with Chi2+Gaussian kernel.
(2) 88 Color Moment with RBF kernels.
(3) Gist descriptor with chi2+Gaussian kernel.
(4) 3 level spatial pyramid kernel with dense SIFT feature (visual vocabulary size: 500)
(5) 3 level PHOG-360 with Chi2+Gaussian kernel.

select the other 25 samples for each class as the test data. We repeatedly choose the training and testing samples for ten times, and all the results reported in this paper are average and standard deviation of results on these ten different and independent data separations. All the web data is used as the unlabeled data.

When the unlabeled data is not used, our method is equal to the traditional Nearest-Neighbor approach, which is treated as the baseline method. We implement the baseline method in 3 versions, where NN-1, NN-3, NN-5 denotes the baseline methods using 1, 3 and 5 kernels, respectively. For NN-1, the kernel (1) in Table 1 is used, and for NN-3, the kernels (1)–(3) are used. We denote our approach with neighborhood similarity but without distance metric learning by UNN-1, UNN-3, and UNN-5, using the same 1, 3 and 5 kernels respectively as the baseline methods. The corresponding traditional NN methods with distance metric learning are denoted by D-NN-1, D-NN-3, and D-NN-5. And finally, we denote our methods with both distance metric learning and neighborhood similarity measure by D-UNN-1, D-UNN-3, and D-UNN-5, respectively. For all the methods, we use a 64-bit hash function for each kernel, and set $B_L=15$ and the unlabeled neighborhood size $B_U=10$ for both performance and efficiency consideration. The performances are demonstrated in Fig. 2 and Table 2.

The results in Fig. 2 and Table 2 are the average of the classification accuracy of ten runs. In Fig. 2(a), compared with NN-1, NN-3, and NN-5, the average classification accuracy is improved when the unlabeled data is incorporated, as can be seen from the performance curves of UNN-1, UNN-3, and UNN-5. In Fig. 2(b), the performance of NN is improved when distance metric learning is incorporated in D-NN. In Fig. 2(c), D-UNN significantly outperforms D-NN and UNN since both semantic consistence and neighborhood information are incorporated. Finally in Fig. 2(d), we compare the four approaches with 5 kernels, and the result shows great improvement of our method over other approaches.

From Table 2 we can see that, our methods UNN outperform the traditional NN approaches on the average accuracy and standard deviation, and they are also comparable with D-NN. Our methods D-UNN outperform other approaches as well as the state-of-the-art NN approach [9].

In the second experiment, we test how the setting of B_L will affect the performance and time required for testing. The results are shown in Fig. 3(a) illustrating the performance of 4 methods with 5 kernels and 30 labeled images per class, where the neighborhood B_U is set to 10 for UNN and D-UNN. Form the curve we can see that for NN-5 and D-NN-5, the best performance is achieved when $B_L=12$, and $B_L=15$ for UNN-5 and D-UNN-5. However, we also notice that the performance of NN-5 and D-NN-5 do not degrade very much when $B_L=15$. Therefore, we set $B_L=15$ for all the experiments in Section 3.1. Next, we conduct experiment to show how the setting of the unlabeled neighborhood B_U affects the performance of UNN-5 and D-UNN-5. The results are shown in Fig. 3(b). We can see that the highest accuracy is achieved when the neighboring sample number is about 10, so we set $B_U=10$. From the experiment result in Fig. 3 we see that the performance of NN based approaches is sensitive to the neighborhood size. Therefore, we have to carefully find the optimal neighborhood size, namely B_L and B_U in this paper on each image database.

Table 3 shows the average test time for each test sample for UNN-5 and D-UNN-5. We notice that the time increases significantly when the size of neighbors increases. If the neighborhood

Table 2
Accuracy with 30 images per class.

Methods	Performance	Methods	Performance
NN-1	33.0 ± 2.1%	D-NN-1	37.5 ± 1.8%
NN-3	36.5 ± 1.75%	D-NN-3	41.5 ± 1.6%
NN-5	40.1 ± 1.4%	D-NN-5	**43.6 ± 1.31%**
UNN-1	35.0 ± 1.1%	D-UNN-1	40.1 ± 1.0%
UNN-3	38.6 ± 0.76%	D-UNN-3	**44.9 ± 0.9%**
UNN-5	**44.4 ± 0.42%**	D-UNN-5	**47.1 ± 0.37%**
Boiman et al. [9]	≈ 42%		

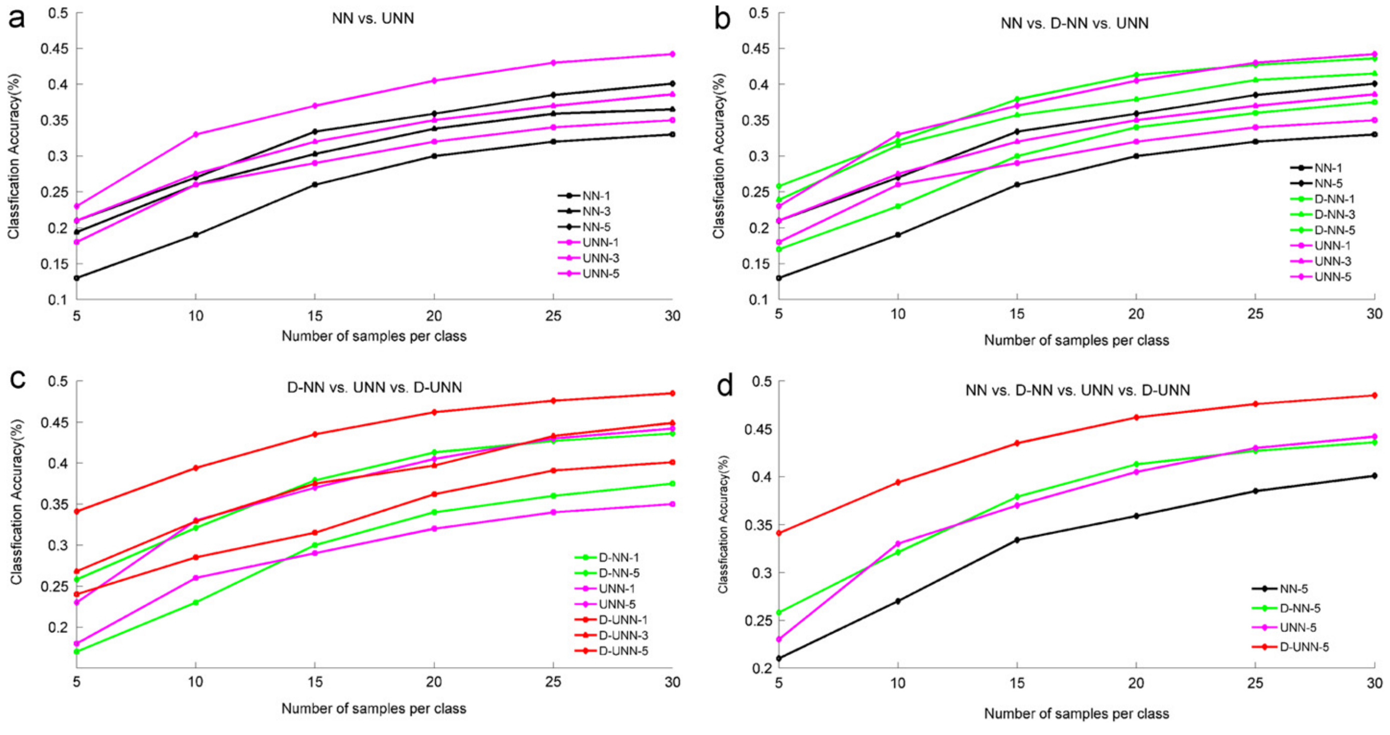

Fig. 2. (a) NN vs. UNN, (b) NN vs. D-NN vs. UNN, (c) D-NN vs. UNN vs. D-UNN and (d) NN vs. D-NN vs. UNN vs. D-UNN.

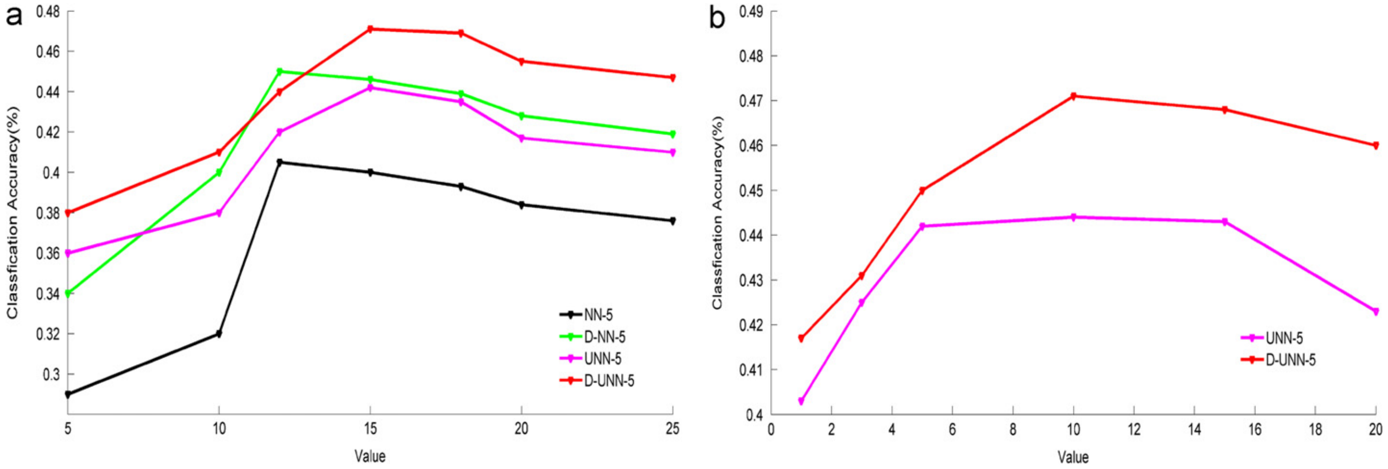

Fig. 3. (a) Performance on different B_L (B_U=10). (b) Performance on different B_U (B_L=15).

Table 3
Average time for different neighborhood size (in s).

#Neighbors	1	3	5	10	15	20
UNN-5	1.2	1.8	2.6	3.7	5.3	8.8
D-UNN-5	1.3	2.1	2.8	3.9	5.7	9.2

size is very large, say 100, the computational cost will be much larger. Meanwhile, the neighborhood similarity will be over-smoothed, and it will fail to describe the local data distribution. Moreover, the average test time for D-UNN-5 is a little bit longer than UNN-5, because the number of the returned labeled and unlabeled candidates of D-UNN-5 is usually large than UNN-5.

3.2. Experiment on MIR-FLICKR dataset

In this section, we conduct experiment on the social media data MIR-FLICKR [25,26]. It contains two datasets, the MIR-FLICKR-25 K [25] and MIR-FLICKR-1 M [26]. Specifically, MIR-FLICKR-25 K is included as a subset in MIR-FLICKR-1 M. Since the ground truth is only provided for images in MIR-FLICKR-25 K, for the experiment of visual categorization, we use MIRFLICKR-25 K data as the labeled data and test data by the way introduced in [25], where the first 3 files in each 5 images are labeled data and the rest are test data. We treat the rest of images in MIR-FLICKR-1 M which contains 975 K as the unlabeled data. We calculate feature (1), (2), (3) and (5) for each image and construct 4 KLSH systems with respect to each feature. The setting of B_L and B_U is set to 21 and 18 respectively using the same tuning method as Section 3.1. The weight w_m for each feature channel is also identified by using a small random subset from U as the validation set.

We conduct the wide sense visual categorization as described in [25], where the human annotators provide annotation with 23 classes, such as people, car, tree, and animal. By observing the images in the dataset, we find that for each image topic there may be dozens of subtopics. Given a query image, those images of the same topic with very different appearance are not likely to provide strong and reliable cues for determining the class of the query, as demonstrated in Fig. 4, where each row of images come from the same topic but with very different subtopics. Therefore, our nearest neighbor system can be a natural choice which can reduce the risk of predicting error using those images with very different appearances.

Before evaluating the performance, we first show what kinds of nearest neighbors are identified by our system given a query image. Some examples are show in Fig. 5. For each given image query on the left in Fig. 5, the images on the first rows on the right denotes the retrieved images using the weighted-combined learned similarity as described in Sections 2.3 and 2.4, and the images on the second rows on the right denotes the retrieved images using the original average Euclidean distance. The yellow crosses denote those images definitely from different classes. It is obvious that prediction made using the retrieved samples by our method rather than those images in Fig. 4 will be much more reliable, although there are a certain number of images from other classes are falsely detected as the nearest neighbors. It can also be seen that compared with the original Euclidean distance metric, our proposed weighted-combined learned similarity provides better semantic consistency so that the prediction made by k-NN using the examples on the first rows in Fig. 5 will be much more reliable.

For experimental evaluation using Mean Average Precision (MAP), we calculate the score of one image with respect to the Qth class as:

$$S(\mathbf{x},Q)=\frac{1}{|N_{LC(Q)}|}\sum_{j=1}^{|N_{LC(Q)}|}\sum_{m=1}^{M} w_m K_N^{(m)}(\mathbf{x},\mathbf{x}_{j,Q})$$
$$\mathbf{x}_{j,Q}\in N_{LC(Q)},\quad Q\in[1,N_C],\quad \sum_{m=1}^{M} w_m=1 \tag{22}$$

where $S(\mathbf{x}, Q)$ denotes the score of $\boldsymbol{x}$ for the Qth class. Then for each class Q, all the test images are ranked by their score $S(\mathbf{x}, Q)$. The Average Precision is calculated based on the ranked list, and finally the Mean Average Precision is calculated for all the classes. Using (22) for decision, the confidential scores rather than the class labels are needed, therefore it is a continuous version of (13). We record the MAP for NN-4, D-NN-4, UNN-4, and D-UNN-4 in Table 4. We see that our method achieves promising improvement. Although the performance gain of D-NN-4 over the baseline NN-4 by using distance metric learning is larger than the performance gain of UNN-4, combining both distance metric learning and neighborhood similarity achieves 29.4% improvement in MAP, which shows that both distance metric learning and neighborhood similarity play non-substitutable roles in our framework.

3.3. Experiment on NUS-WIDE dataset

To further evaluate the performance of our method on social media data, we conduct experiments on the web image database NUS-WIDE [13]. The dataset contains 81 semantic classes. We use all the six types of features provided by [13], which includes two color features (CH, CORR and CM55), texture features (EDH and

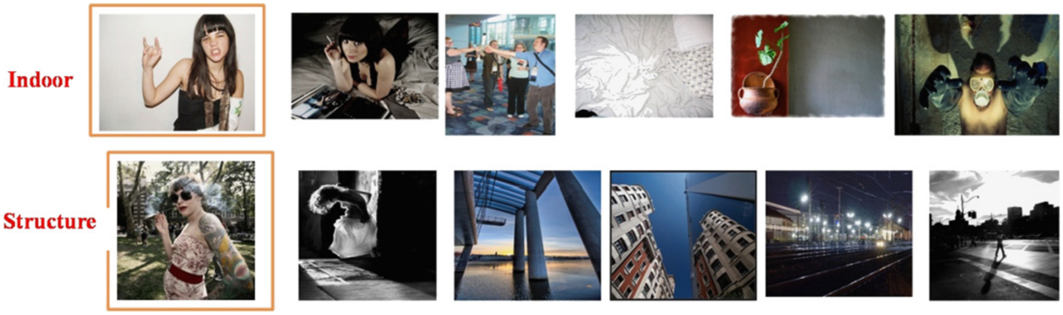

Fig. 4. Some examples from the same category.

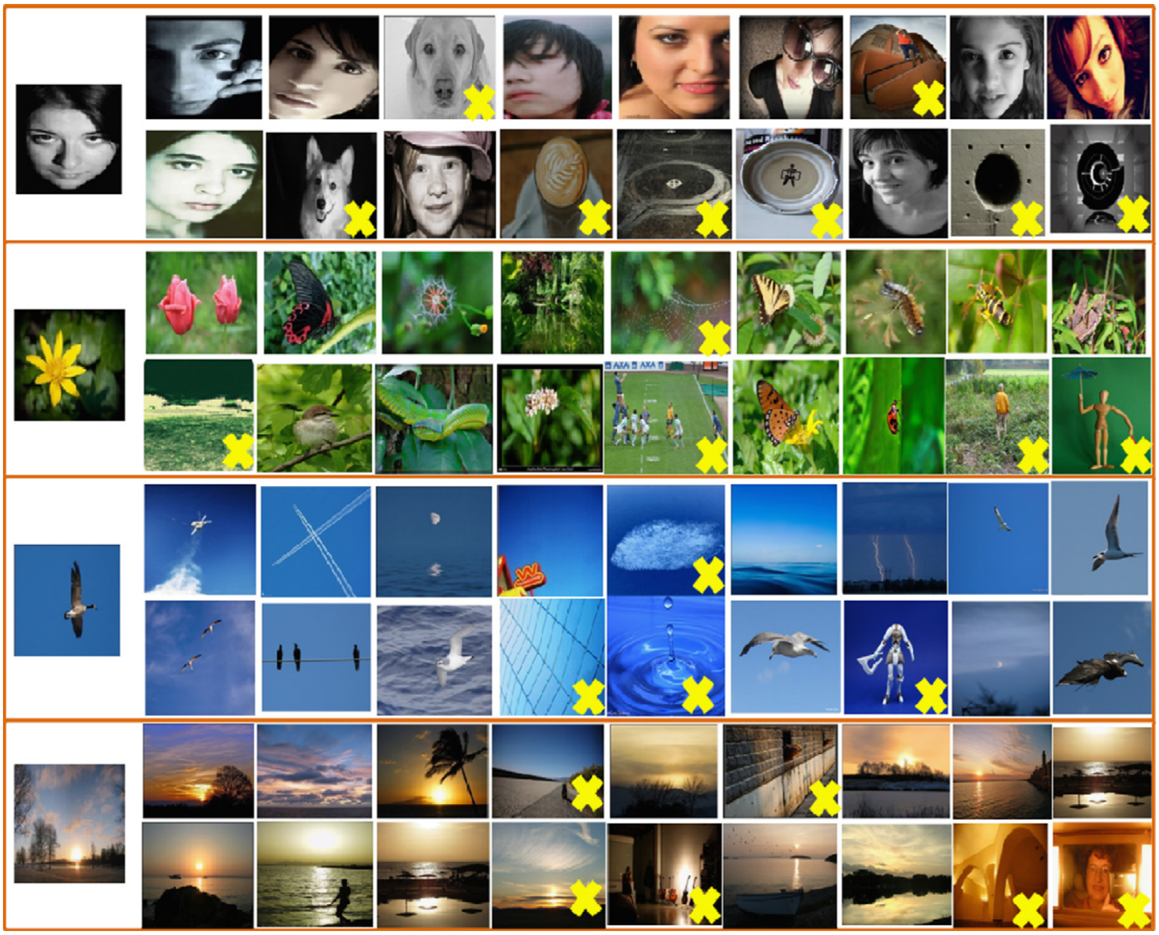

Fig. 5. Examples of retrieved images given the query. Left images are the queries. The first rows and the second rows denote retrieved examples using weighted combined learned distance and the original average distance. Images marked by yellow cross are irrelevant.

Table 4
Mean average precision on MIR-FLICKR dataset.

Method	MAP
NN-4	0.323
D-NN-4	0.361
UNN-4	0.358
D-UNN-4	**0.418**

WT) and bag-of-words feature. For evaluation of our method, we equally divide the training data into two disjoint subsets L and U. We test three methods on this dataset. The first (DML-half) is k-NN classification using the distance metric learning by *LMNN* on subset L. The second (DML-all) is k-NN classification using *LMNN* on both L and U, and we use U as the labeled set. The third (MNS) is k-NN using our multiple neighborhood similarity trained on L and U, but we treat L and U as the labeled set and unlabeled set respectively. For evaluation of our method, we set $B_L=25$ and $B_U=12$. To compute the Mean Average Precision (MAP) for each semantic concept, we calculate the decision output by (22). We repeat the training and testing procedure by 10 times, and we average the results for each semantic concept and the overall MAP and show them in Fig. 6.

In this part of evaluation, the first method using half of the training data as the labeled data is treated as the performance lower bound of our method, and the second method using all the training data as the labeled data is treated as the performance upper bound. What is interesting in this experiment is that how much the performance can be boosted by our multiple neighborhood similarity. From Fig. 6 we see that, the performance lower

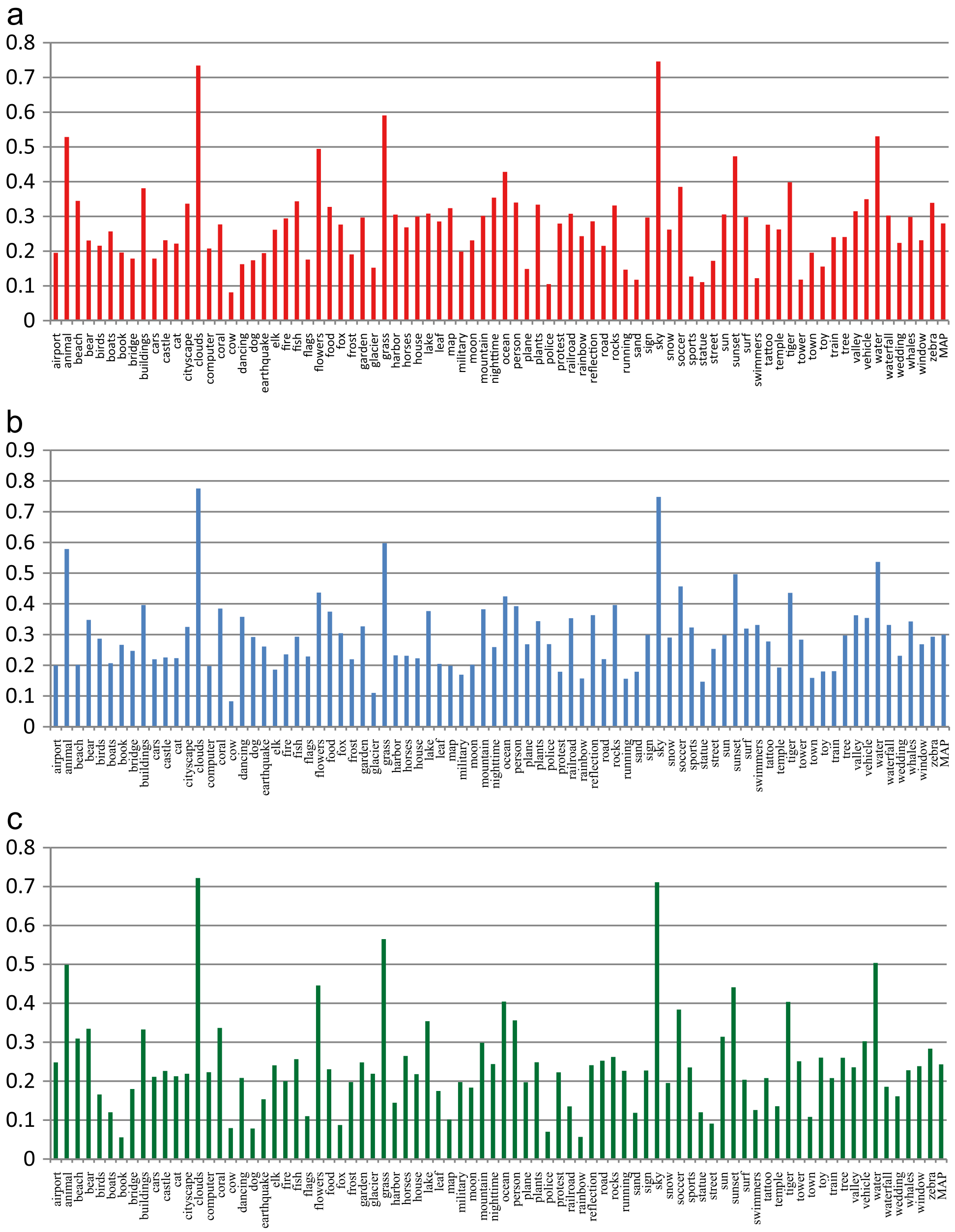

Fig. 6. (a) MAP of our MNS using L and U for labeled and unlabeled data respectively. (b) k-NN using DML-half with L and U as labeled training data. (c) k-NN using DML-all with L as the labeled training data.

bound and upper bound in MAP is 0.2434 for DML-half and 0.2995 for DML-all respectively, while our approach achieves 0.2797. Our method and the other two significantly outperform the reported MAP (0.1569) for the baseline k-NN in [13].

Moreover, among all the 81 semantic concepts, our method outperforms DML-half on 56 concepts, and outperforms DML-all on 28 concepts. Our method achieves the highest Average Precision on 25 concepts including scene concepts such as *beach* and *nighttime*, object concepts such as *boats*, *moon*, *vehicle* and *zebra*, higher level semantic concepts such as *military* and event concepts *protest* and *fire*. The experiment again provides strong support for our proposed multiple neighborhood similarity to show that it is capable of take advantage of the unlabeled data to improve the performance for k-NN classification. Another interesting issue observed in Fig. 6 is that DML-all does not outperforms on all the semantic concepts, as we observe that it only outperforms others on 50 out of 81. And correspondingly, DML-half does not underperform on all the semantic concepts, as we observe that it still outperforms on 6 semantic concepts.

4. Related work

The method we are studying in this paper is closely related to three different research areas, the nearest-neighbor based method for image classification, learning with multiple features and nearest neighbor search. We briefly review some of the recent developments in these areas because it is hard to cover every piece.

4.1. Image classification using nearest-neighbor methods

As discussed in the introduction, NN approaches usually show inferior performances than offline learning approaches because they are generally dependent on the database size, the features and the distance metric. To bridge the performance gap, a lot of studies have been conducted, which try to overcome the intrinsic weakness of NN from different aspects. Firstly, from designing suitable distance metric aspect, Boiman et al. [9] claimed that the previously used image-to-image distance will lead to the degradation of NN approaches and proposed an image-to-class distance measure. Friedman [18] proposed a new local similarity measure based on kernel methods and recursive partitioning techniques. Another similarity measure was proposed in [37] to incorporate the invariance of translations and scaling. As an effective solution that complements NN approaches, *Distance Metric Learning* aims to learn a new distance metric better suited to the specific classification tasks given the side information of the training samples. Many state-of-the-art approaches were proposed during the past few years, such as *NCA* [20], *ITML* [15], *MLCC* [21] and *LMNN* [43]. In this paper, we use *LMNN* which demonstrates good performances on various classification tasks.

Secondly, from database size aspect, Torralba et al. [37] found that with extremely large tiny image database, i.e., 80 millions, NN methods could work well for image annotation, although the tags of the images are very noisy. Deng et al. [16] constructed a large scale database with human labeled ground truth on more than 10 K classes of images. Many efforts have also been devoted by other researchers to construct large scale image databases for benchmark testing and performance evaluation, such as NUS-WIDE [13], and MIR-FLICKR [25,26]. NN approach is more likely to achieve good performance on these large scale databases.

Next, from image feature aspect, Boiman et al. [9] showed that feature quantization will reduce the discrimination power of local features, since NN methods are already very sensitive to the variation of the local manifold structures. Zhang et al. [49] proposed to construct a large scale set of visual words and phrases vocabulary by considering the spatial context information, and good results were reported by Nearest-Neighbor classification using the histogram intersection based on the descriptive visual words and phrases representation.

Finally, to enhance to robustness of NN methods, Zhang et al. [48] combined the efficiency of NN and the effectiveness of SVM. Local kernel alignment of nearest neighbors was proposed by Lin et al. [28] to combine the discrimination power of multiple kernel representation of the neighboring samples given the query images. Promising performance was achieved by [48,28] on some small and medium scale image dataset with significant reduction of computation cost compared with the traditional global learning approaches.

4.2. Learning with multiple features

A lot of studies have been conducted on efficiently combining the discrimination power of different features. Two simple schemes are early fusion where the model is learned on a concatenated feature representation and late fusion where the decision outputs of individual classifier are ensemble to form the final output [34]. A canonical semi-supervised learning model for using multiple feature representation is co-training [8], or multi-view learning [32,33]. Wang et al. [42] proposed a semi-supervised learning approach for real world image applications.

As one of the most promising feature fusion methods, Multiple Kernel Learning (MKL) [5,31,35] combines multiple features in the way of linear combination of kernels. It has also been used for object recognition [38], and promising results have been reported on many challenging datasets. Later, several MKL approaches [11,22,47,51] modeling the nonlinear combination of kernels were developed, and better results are achieved especially on image classification task compared with the original MKL approaches [5,31,35].

The method in this paper is inspired by the idea of MKL. We learn the kernel coefficients by minimizing the empirical loss instead of the structural risk minimization and max-margin framework in MKL which calls for complicated convex programming procedures.

4.3. Nearest neighbor search

Nearest neighbor search is one of the key components in modern information retrieval on large scale database. Among the relevant researches, the most naïve approach is the precise linear search which could not scale well on large scale data. Instead, many methods were developed to conduct approximated nearest neighbor search by the idea of space partitioning, such as *KD-Tree* [6], *R-Tree* [23], *BSP-Tree* [30], and *Ball-Tree* [29]. However, these approaches are likely to fall prey to the “curse of dimensionality” problem. Locality Sensitive Hashing [12,14] is one well-known method which performs probabilistic dimension reduction and approximated nearest neighbor search for high-dimensional data. The basic idea is to hash the input items so that similar items are mapped to the same buckets with high probability. Compared with the space partitioning methods, the robustness and efficiency of LSH on high dimensional data has been proved in many studies such as [12,14]. Since LSH is only able to perform in Euclidean space, Kulis et al. [27] proposed to perform LSH in the unknown high dimensional space with respect to any given kernel, which endows LSH with the power of using many of the existing image kernels as well as learned distance metric/similarity. We use KLSH in this paper because of its advantage over space partition methods and traditional LSH.

5. Conclusion

We propose a new Nearest-Neighbor classification method in this paper. Our contribution includes four aspects: (1) we propose a neighborhood similarity measure which encodes the local density information by using unlabeled data and semantic consistence by incorporating distance metric learning; (2) we propose a method to combine the discrimination power of different features to form the final decision output of an unknown sample, which enhances the robustness for processing the real world data; (3) we provide theoretic analysis to demonstrate how k-NN using multiple neighborhood similarity outperforms k-NN using single feature and image-to-image distance; (4) we construct a practical system that is able to perform real world social media image categorization. Our method provides promising classification result on benchmark dataset Caltech-256 as well as social media image database MIR-FLICKR and NUS-WIDE compared with the traditional Nearest-Neighbor approaches. Future study will be focused on studying and developing more robust local learning model that can boost the performance of our proposed approximated nearest neighbor search system on multiple feature representation.

Acknowledgment

This work was supported in part by National Basic Research Program of China (973 Program): 2012CB316400, in part by National Natural Science Foundation of China: 61025011, 61035001, 61070108, 61133003 and 61003165. This work was also supported in part to Dr. Qi Tian by NSF IIS 1052851, Faculty Research Awards by Google, FXPAL, NEC Laboratories of America, and ARO grant W911BF-12-1-0057 respectively.

References

[1] ⟨http://blog.nielsen.com/nielsenwire/global/social-media-accounts-for-22-percent-of-time-online/⟩.

[2] ⟨http://www.flickr.com⟩.

[3] ⟨http://www.youtube.com⟩.

[4] ⟨http://www.twitter.com⟩.

[5] F. Bach, G. Lanckriet, M. Jordan, Multiple Kernel Learning, Conic Duality, and the SMO Algorithm, ICML, 2004.

[6] J.L. Bentley, Multidimensional binary search trees used for associative searching, Commun. ACM 18 (9) (1975) 509–517.

[7] C.M. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag New York, Inc., Secaucus, NJ, 2006.

[8] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceeding of 11th Annual Conference on Computational Learning Theory, 1998, pp. 92–100.

[9] O. Boiman, E. Shechtman, M. Irani, In Defense of Nearest-Neighbor Based Image Classification, CVPR, 2008.

[10] A. Bosch, A. Zisserman, and X. Munoz. Representing Shape with a Spatial Pyramid Kernel. CIVR, 2007.

[11] L. Cao, J. Luo, F. Liang, T.S. Huang, Heterogeneous Feature Machine for Visual Recognition, ICCV, 2009.

[12] M. Charikar, Similarity Estimation Techniques from Rounding Algorithms. ACM Symposium on Theory of Computing, 2002.

[13] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y. Zheng, NUS-WIDE: A Real-World Web Image Database from National University of Singapore, CIVR, 2009.

[14] M. Datar, N. Immorlica, P. Indyk, V.S. Mirrokni, Locality-sensitive hashing scheme based on p-stable distributions, in: Proceedings of the 20th Annual Symposium on Computational Geometry, 2004, pp. 253–262.

[15] J.V. Davis, B. Kulis, P. Jain, S. Sra, I.S. Dhillon., Information-Theoretic Metric Learning, ICML, 2007.

[16] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, CVPR, 2009.

[17] J. Fan, Y. Gao, H. Luo., Multi-level Annotation of Natural Scenes Using Dominant Image Components and Semantic Concepts, ACM Multimedia, 2004.

[18] J.H. Friedman, Flexible Metric Nearest Neighbor Classification. Technical Report, 1994.

[19] A. Gionis, P. Indyk, R. Motwani, Similarity Search in High Dimensions via Hashing, VLDB, 1999.

[20] J. Goldberger, S. Roweis, G. Hinton, R. Salakhutdinov, Neighborhood Component Analysis, NIPS, 2005.

[21] A. Globerson, S. Roweis, Metric Learning by Collapsing Classes, NIPS, 2006.

[22] M. Gonen, E. Alpaydin, Localized Multiple Kernel Learning, ICML, 2008.

[23] A. Guttman, R-Trees: A Dynamic Index Structure for Spatial Searching. Proc. ACM SIGMOD International Conference on Management of Data, 1984, pp. 47–57.

[24] R. Hong, M. Wang, X.-T. Yuan, M. Xu, J. Jiang, S. Yan, T.-S. Chua., Video Accessibility Enhancement for Hearing Impaired Users, ACM Trans. Multimedia Comput. Commun. App. 7S (24) (2011).

[25] M.J. Huiskes, M.S. Lew, The MIR Flickr Retrieval Evaluation, ACM MIR, Vancouver, Canada, 2008.

[26] M.J. Huiskes, B. Thomee, M.S. Lew, New Trends and Ideas in Visual Concept Detection, ACM MIR, Philadelphia, USA, 2010.

[27] B. Kulis, K. Grauman., Kernelized Locality Sensitive Hashing for Scalable Image Search, ICCV, 2009.

[28] Y. Lin, T. Liu, C. Fuh., Local Ensemble Kernel Learning for Object Category Recognition, CVPR, 2007.

[29] T. Liu, A.W. Moore, A. Gray, K. Yang., An Investigation of Practical Approximate Nearest Neighbor Algorithms, NIPS, 2005.

[30] H. Radha, R. Leonardi, M. Vetterli, B. Naylor, Binary Space Partitioning Tree Representation of Images, J. Visual Commun. Image Process. 2 (3) (1991).

[31] A. Rakotomamonjy, F. Bach, S. Canu, Y. Grandvalet, SimpleMKL, JMLR 9 (2008) 2491–2521.

[32] V. Sindhwani, P. Niyogi, M. Belkin., Beyond the Point Cloud: from Transductive to Semi-Supervised Learning, ICML, 2005.

[33] V. Sindhwani, D.S. Rosenberg., An RKHS for Multi-view Learning and Manifold Co-regularization, ICML, 2008.

[34] C.G.M. Snoek, M. Worring, Early Versus Late Fusion in Semantic Video Analysis, ACM Multimedia, 2005.

[35] S. Sonnenburg, G. Ratsch, C. Schafer, B. Scholkopf., Large Scale Multiple Kernel Learning, 7, JMLR, 2006, pp. 1531–1565.

[36] X. Tian, L. Yang, J. Wang, Y. Yang, X. Wu, X. Hua., Bayesian Video Search Re-ranking, ACM Multimedia, 2008.

[37] A. Torralba, R. Fergus, W.T. Freeman, 80 million tiny images: a large data set for nonparametric object and scene recognition, PAMI 30 (11) (2008) 1958–1970.

[38] M. Varma, D. Ray, Learning the Discriminative Power-Invariance Trade-off, ICCV, 2007.

[39] M. Wang, X.-S. Hua, J. Tang, R. Hong, Beyond distance measurement: constructing neighborhood similarity for video annotation, IEEE Trans. Multimedia 11 (3) (2009) 465–476.

[40] M. Wang, X. Hua, R. Hong, J. Tang, G. Qi, Y. Song, Unified video annotation via multi-graph learning, IEEE Trans. Circuit Syst. Video Technol. 19 (5) (2009).

[41] S. Wang, Q. Huang, S. Jiang, Q. Tian., Nearest-Neighbor Classification Using Unlabeled Data for Real World Image Application, ACM Multimedia, 2010.

[42] S. Wang, S. Jiang, Q. Huang and Q. Tian. S^3MKL: Scalable Semi-Supervised Multiple Kernel Learning for Image Data MiningACM Multimedia, 2010.

[43] K.Q. Weinberger, L.K. Saul, Distance Metric Learning for Large Margin Nearest-Neighbor Classification, 10, JMLR, 2009 207–244.

[44] J. Weston, C. Leslie, E. Le, D. Zhou, A. Alisseeff, W.S. Noble., Semi-Supervised protein classification using cluster kernels, Bioinformatics 21 (15) (2005) 3241–3247.

[45] L. Wu, S.C.H. Hoi, R. Jin, J. Zhu, N. Yu, Distance Metric Learning from Uncertain Side Information with Application to Automated Photo Tagging, ACM Multimedia, 2009.

[46] R. Yan, J. Tesic, J.R. Smith., Model-shared Subspace Boosting for Multi-label Classification, ACM SIGKDD, 2007.

[47] J. Yang, Y. Li, Y. Tian, L. Duan, W. Gao, Group Sensitive Multiple Kernel Learning for Object Categorization, ICCV, 2009.

[48] H. Zhang, A.C. Berg, M. Maire, J. Malik., SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition, CVPR, 2006.

[49] S. Zhang, Q. Tian, G. Hua, Q. Huang, S. Li, Descriptive Visual Words and Visual Phrases for Image Application, ACM Multimedia, 2009.

[50] X. Zhu, Semi-supervised learning literature survey. Technical Report 1530, University of Wisconsin—Madison, 2006.

[51] S. Wang, S. Jiang, Q. Huang, Q. Tian, Multiple Kernel Learning with High Order Kernels, ICPR, 2010.

Shuhui Wang received the B.S. degree in Electronic Engineering from Tsinghua University, Beijing, China, in 2006. He is currently pursuing the Ph.D. degree in Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences. His research interests include semantic image analysis, image and video retrieval and large-scale web multimedia data mining.

Qingming Huang received the B.S. degree in computer science and Ph.D. degree in computer engineering from Harbin Institute of Technology, Harbin, China, in 1988 and 1994, respectively. He is currently a Professor with the Graduate University of the Chinese Academy of Sciences (CAS), Beijing, China, and an Adjunct Research Professor with the Institute of Computing Technology, CAS. He has authored or coauthored nearly 200 academic papers in prestigious international journals and conferences. His research areas include multimedia video analysis, video adaptation, image processing, computer vision, and pattern recognition Dr. Huang is a reviewer for IEEE Trans. on Multimedia, IEEE Trans. on Circuits and Systems for Video Technology, and IEEE Trans. on Communications. He has served as program chair, track chair and TPC member for various conferences, including ACM Multimedia, CVPR, ICCV, ICME, PSIVT, etc.

Shuqiang Jiang received the M.S. degree from the College of Information Science and Engineering, Shandong University of Science and Technology, Shandong, China, in 2000, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005. He is currently an associate professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing. He is also with the Key Laboratory of Intelligent Information Processing, Chinese Academy of Sciences. His research interests include multimedia processing and semantic understanding, pattern recognition, and computer vision. He has authored or coauthored more than 90 papers on the related research topics.

Qi Tian received the B.E. degree in electronic engineering from Tsinghua University, China, in 1992 and the Ph.D. degree in electrical and computer engineering from the University of Illinois, Urbana–Champaign in 2002. He is currently an Associate Professor in the Department of Computer Science at the University of Texas at San Antonio (UTSA). His research interests include multimedia information retrieval and computer vision. He has published over 150 refereed journal and conference papers. His research projects were funded by NSF, ARO, DHS, SALSI, CIAS, and UTSA and he also received faculty research awards from Google, NEC Laboratories of America, FXPAL, Akiira Media Systems, and HP Labs. He took a one-year faculty leave at Microsoft Research Asia (MSRA) during 2008–2009. He was the author of a Top 10% Best Paper Award in MMSP 2011, a Best Student Paper in ICASSP 2006, and a Best Paper Candidate in PCM 2007. He received 2010 ACM Service Award. He has been serving as Program Chairs, Organization Committee Members and TPCs for numerous IEEE and ACM Conferences including ACM Multimedia, SIGIR, ICCV, ICME, etc. He is the Guest Editors of IEEE Transactions on Multimedia, Journal of Computer Vision and Image Understanding, Pattern Recognition Letter, EURASIP Journal on Advances in Signal Processing, Journal of Visual Communication and Image Representation, and is in the Editorial Board of IEEE Transactions on Circuit and Systems for Video Technology (TCSVT), Journal of Multimedia (JMM) and Journal of Machine Visions and Applications (MVA). He is a Senior Member of IEEE and a Member of ACM.

Lei Qin received the B.S. and M.S. degrees in mathematics from the Dalian University of Technology, Dalian, China, in 1999 and 2002,respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2008. He is currently a Faculty Member with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. His research interests include image/video processing, computer vision, and pattern recognition. He has authored or coauthored over 20 technical papers in the area of computer vision.