Neurocomputing 145 (2014) 416–426

Contents lists available at ScienceDirect

Neurocomputing

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

Sparse representations based attribute learning for flower classification

Keyang Cheng [a,b,*], Xiaoyang Tan [a,*]

[a] School of Information Science & Technology, Nanjing University of aeronautics & astronautics, Nanjing, Jiangsu, 210016, China
[b] School of Computer Science & Telecommunications Engineering, Jiangsu University, Zhenjiang, Jiangsu 212013, China

ARTICLE INFO

Article history:
Received 6 July 2013
Received in revised form
5 March 2014
Accepted 10 May 2014
Communicated by Tao Mei
Available online 17 May 2014

Keywords:
Flower classification
Attribute learning
Sparse representation
Attribute reduce

ABSTRACT

Classification for flowers is a very difficult task. Traditional methods need to built a classifier for each flower category, and obtain large number of flower samples to train these classifiers. In practice, many different types of flowers make the job become very difficult and boring. In this work, we present an attribute based approach for flowers recognition. Particularly, instead of training for a specific category of flowers directly based on manually designed features such as SIFT and HoG, we extract a series of visual attributes from a given set of flower images and generalize these to new images with possibly unknown flowers. A recently proposed sparse representations classification scheme is employed to predict the attributes of a given flower image from any category. In addition, we use a genetic algorithm to find the most discriminative attributes among others for better performance during the stage of flower classification. The effectiveness of the proposed method is validated on a publicly available flower classification database with promising results.

1. Introduction

The task of classification for flowers is very difficult. There are large variations in scale and viewpoint in typical flower images, partial occlusions, illumination, multiple instances, etc. Perhaps the biggest challenge comes from the intra-class and the inter-class variability. For example, some images from different classes are with a smaller variation than from a class itself, and some minute differences determine their different classification [1–7]. In addition, as a plant with continuous growth and a non-rigid object, flowers can be deformed in many ways, so in the intra-class there is also a big change. Recent machine learning technique has demonstrated its power in classifying flowers. For example, one can consider kernel SVM [8] or boosting [9]. The performance of the state of the art method is obtained with the use of multiple features [10] and a multiple kernel classifier: each kernel is designed for different features (e.g. colour, texture), and an additional kernel is designed for the weighted combination of these feature kernels.

Recently, along with the emergence of camera-equipped mobile phones, new opportunity and challenge have generated in the computer vision field. One of these challenges is that in resource constrained environments, available memory, bandwidth and processing power become restricting factors in image classification [11]. For example, with a camera-equipped smart phone, a user captures a flower image and wants to learn more knowledge about the flower. In a traditional image identification system, it would either extract some type of features of the flower image and transmit it to a server or transmit the whole flower image to the server directly. Then, for recognizing the category of the flower image and feeding back the depicted information of the flower, the server needs to perform a series of classification tests.

Regarding the feasibility of flowers classifying, there are two important issues. The first one is that there should be enough labeled examples for training the classifier. Then we can use the classifier to identify the possible category of new test examples having the same distribution with the training examples. However, due to the large number of the possible image classes, collecting a large amount of examples for training may not be feasible even though the number of classes is not a large one [12]. In addition, training classifiers of a real-time application in the server may also be impractical, because along with the increase of visitors, the performance of system response will drop quickly.

The second one which leads to the failure of system design is the limited power availability, bandwidth and processing capabilities of mobile systems. For example, the available battery power will quickly drain when transmitting raw images. Besides, because the transmitted information load also increase congestion on the server and the network, the user satisfaction will be directly affected.

To cope with the challenges mentioned above, attribute learning proposed recently is attempted to learn attributes instead of

* Corresponding authors at: School of Information Science & Technology, Nanjing University of Aeronautics & Astronautics, Nanjing, Jiangsu 210016, China.
Tel.: +86 13094956326.
E-mail addresses: kycheng@ujs.edu.cn (K. Cheng), x.tan@nuaa.edu.cn (X. Tan).

http://dx.doi.org/10.1016/j.neucom.2014.05.011
0925-2312/

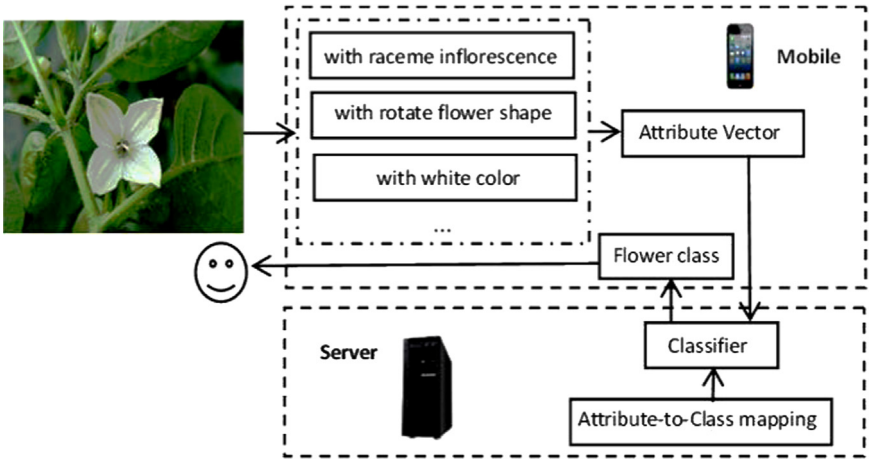

Fig. 1. Attribute based image classification on a mobile system.

traditional classes of image. In the camera-equipped smart phone system shown in Fig. 1, it could recognize the attributes e.g. "red" or "green" for the specific classes' classification. The processing pipeline includes two stages: the one is the attribute vector creating process which is accomplished by the mobile facility, and the other one is final classification taken place on the server. The classification stage is learned off-line from textual description, which gives consideration to the mapping of attributes to classes.

There are many advantages in employing attributes to be a bridge between image examples and classes. The greatest benefit from the use of attributes is that classifiers can be trained and tested using only text information without direct image features. This means that instead of using label sample as metadata, and in order to automatically identify test examples, large knowledge databases can be employed for extracting information. Besides, textual information is easier to transmit, process and store than the image information. Compared to raw images, the text of attributes with smaller dimensions can be easier employed to reduce the query time for a huge retrieval system. Considering that the visits to the server is very large at the same time, the advantage mentioned above becomes very vital for a system like the one shown in Fig. 1.

In recent years, as a kind of high-level image feature, the semantic attribute has gained more and more attention in the field of computer vision. Attributes' learning has been applied widespread in classifying objects [13] or images [14]. Farhadi et al. [15] first put forward a set of visual semantic attributes to describe objects. Later, Kumar et al. [16] proposed a novel method to predict the visual attribute vector through the related attribute classifiers, and using these attribute vectors to represent faces. Vogel and Schiele [17] employed visual attributes to express the semantics of the outdoor scenes. Vaquero et al. proposed an attribute based people search method [18]. Attributes learning also has many potential applications in transfer learning [19], multi-label learning [20], multi-instance learning [21], video annotation [22] and image retrieval [23]. Despite that the semantic attributes have been used in many kinds of image classifications, to our knowledge there is no system that identifies flowers using their attributes.

As far as the power availability and the communication bandwidth of mobile systems are concerned, when the raw images or image descriptors are limitedly transferred, attribute expression can still be transmitted. In addition, instead of integrating the whole system of images recognition, only two processing stages are needed in the mobile system. The first one is the feature extraction, and the second is the attribute prediction. In recent years, feature extraction and image classification have been applied in smart phones and other mobile devices [24–26] and have got very exciting results.

Attribute based identification as a new way for image classification can be used to deal with the flowers classifying based on camera-equipped mobile phones. However, for the existence of a large number of available attributes, collecting samples to train the attribute classifiers is a tedious task. In addition, the traditional attribute based classification framework assumes independence among attributes, which cannot be ensured by the attributes annotated by human. In this paper we propose an integrative framework that enables us to predict attribute automatically and to estimate the prior attribute–class probability relationship matrix. We use some image samples to compose the dictionary of the attribute classifier and employ the recently proposed sparse representations classification scheme to predict other samples attribute. Besides, in order to find the most discriminative attributes, a genetic algorithm is employed to reduce attributes for the flower classification. The benefits of the proposed extensions are validated through the attribute-to-class mapping experimental results.

2. The proposed method

In the traditional approach, given a signal such as a vectorized image $x \in R^n$, the signal x is called the k-sparse with respect to a dictionary $D \in R^{n \times n}$ if $x = Ds$ where $k = \|s\|_0$. If the class of the image x is $y \in Y$, there will be a mapping matrix W which makes $y = Wx = WDs$. The parameter W will be learned by the training samples in the training step and after that the label y of the testing sample will be predicted by the learned classifier [27–29].

In the classification scheme based on the attributes, however, we are given an attribute representation $a \in A$ for each class $y \in Y$ and the goal is to learn a non-trivial classifier $f : X \to Y$. This can be achieved with two subsequent parts. The first part is the sparse representation-based classifier for attribute prediction $f_a : X \to A$, and the second is the class mapping with class–attribute matrix $f_c : A \to Y$.

In particular, a classifier for attribute a, trained with a set of images labeled with $a = 1$ as positive otherwise as negative, can provide an estimate of the posterior probability $p(a|x)$ of that attribute being present in image x. To obtain the posterior probability of class y for a given image x, we marginalize over all

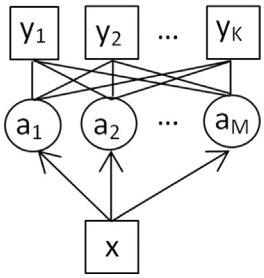

Fig. 2. Classification model based on attributes [13].

Table 1
Attribute learning based on sparse representations algorithm.

Attribute learning based on sparse representations algorithm
Input: The training data $X_{tr}=\{x_i\}_{i=1}^{N}$, its class label Y_{tr}, and its attribute label $L_{tr}=\{l_i^k\}_{i=1k=1}^{NM}$, $l_i^k\in\{0,1\}$, the testing data $X_{te}=\{x_t\}_{t=1}^{L}$, the class set $Y=\{y_j\}_{j=1}^{C}$, the attribute set $A=\{a_k\}_{k=1}^{M}$, the target threshold ϵ
Output: The class label of testing data.
1: With the training data, run the K-SVD algorithm to approximate the Dictionary D and the sparse representation of training samples' feature: $\{\hat{D},\hat{s_i}\}=\mathrm{argmin}_{D,s_i}\sum_{i=1}^{N}\|x_i-Ds_i\|_2$ s.t. $\|s_i\|_1\le\epsilon$.
2: Employ the training data to obtain the parameters w_k, b_k for every attributes' classifier: $\{\hat{w_k},\hat{b_k}\}=\mathrm{argmin}_{w_k,b_k}\|w_k\|_2$ s.t. $l_i^k(w_k^T s_i+b_k)\ge 1$
3: With the training data, class–attribute matrix can be obtained by the number ratio of the samples with category y_j in the attribute a_i to all the training samples of the attribute a_k: $p(y_j\|a_k)=\frac{\#N_{a_k}^{y_j}}{\#N_{a_k}}$.
4: With the testing data $\{x_t\}_{t=1}^{L}$, use the OMP algorithm to approximate the sparse representation of testing sample: $\hat{s}_t=\mathrm{argmin}_{s_t}\|s_t\|_1$ s.t. $\|x_t-Ds_t\|_2\le\epsilon$
5: In the testing stage, with the s_t of test sample calculated by OMP algorithm and the parameters of each attributes' classifier, estimate the attribute classification posterior probability of testing data x_t by a sigmoid function: $p(a_k\|x_t)=\frac{1}{1+e^{-(w_k s_t+b_k)}}$, where k indicate the kth attribute.
6: The posterior probability of class y_j the image x_t belonging to is then obtained by $p(y_j\|x_t)=\sum_{k=1}^{N}p(y_j\|a_k)p(a_k\|x_t)$.
7: The class label of testing sample x_t is given by $\hat{y}=\max_{y_j}p(y_j\|x_t)$.

possible attributes associated with this image, using Bayes rule [30]:

$$p(y|x)=\sum_{a\in\{0,1\}}p(y|a)p(a|x) \tag{1}$$

The role played by the attributes in the whole categorization framework is illustrated in Fig. 2 and the corresponding learning algorithm based on the sparse representations is given in Table 1.

2.1. Attribute prediction based on sparse representation

In this section, we describe our method for attribute prediction, whose goal is to identify the most prominent attributes of a given image. For this, we employ a sparse representation-based method. Sparse representation is intuitively appealing since a new test image with a series of semantic attributes is hoped to be represented by small amounts of training images. With such a sparse representation, detecting the particular attribute's presence or not can be efficiently evaluated by a simple binary classifier, such as SVM. Consequently, it offers high prediction accuracy with light computational cost and high scalability capabilities for its full consideration to the sparsity of flower image features.

In particular, to obtain the sparse representation s of a test sample x_t, we solve the following regularized l_1 minimization problem:

$$\hat{s}_t=\arg\min_{s_t}\|s_t\|_1 \quad \text{s.t. } \|x_t-Ds_t\|_2\le\epsilon \tag{2}$$

Many methods have been proposed to solve Eq. (2) such as orthogonal matching pursuit (OMP), basis pursuit (BP) and Least Absolute Shrinkage and Selection Operator (LASSO)[31]. We choose the OMP algorithm in this work due to its simplicity and efficiency. In the OMP, with the help of computing the orthogonal projection of the signal onto the set of atoms selected so far, all the coefficients extracted are updated after each step. However, before running the OMP algorithm, we have to prepare a dictionary D for it. In our implementation, the dictionary D is obtained by using an adaptive training process, i.e., K-SVD, which is an iterative method that alternates between updating the dictionary atoms to obtain an optimal value and computing sparse coefficients of the examples based on the current dictionary.

Specifically, we solve the following optimizing problem on the training data using the K-SVD algorithm:

$$\{\hat{D},\hat{s_i}\}=\arg\min_{D,s_i}\sum_{i=1}^{N}\|x_i-Ds_i\|_2 \quad \text{s.t. } \|s_i\|_1\le\epsilon,\ \forall i \tag{3}$$

where x_i is the i-th training image. In this way, we simultaneously obtain for the training set a dictionary $\hat{D}$, which will be used in the test stage (see Eq. (2)), and a sparse representation $\hat{s_i}$, which can be used to construct the attribute predictor. Here we adopt a linear model for each predictor. By denoting the k-th attribute of the i-th training image as l_i^k, the parameters w_k and b_k for the k-th attribute predictor can be obtained as

$$\{\hat{w_k},\hat{b_k}\}=\arg\min_{w_k,b_k}\|w_k\|_2 \quad \text{s.t. } l_i^k(w_k^T s_i+b_k)\ge 1,\ \forall i \tag{4}$$

In the testing stage, given the sparse representation s_t of a test image x_t calculated by OMP algorithm and the parameters of every attribute classifier, we can estimate the posterior probability of testing data having the attribute k using a sigmoid function:

$$p(a_k=1|x_t)=\frac{1}{1+e^{-(w_k s_t+b_k)}} \tag{5}$$

Before making the final flower classification, however, we need to estimate the mapping from attributes to class, i.e., $p(y_j|a_i)$, which we call the attribute–class matrix. A maximal likelihood method is adopted here as follows:

$$p(y_j|a_k)=\frac{\#N_{a_k}^{y_j}}{\#N_{a_k}} \tag{6}$$

where $\#N_{a_k}^{y_j}$ is the number of training images in the y_j class with attribute a_k, and $\#N_{a_k}$ is the total number of training images with attribute a_k. With this, the posterior probability that the test image x_t belongs to the category y_j can be estimated using $p(y_j|x_t)=\sum_{k=1}^{N}p(y_j|a_k)p(a_k|x_t)$, and hence the final class label is given by

$$\hat{y}=\max_{y_j}p(y_j|x_t) \tag{7}$$

2.2. Attribute reduction based on genetic algorithm

The idea of attribute reduction is to choose the core and important knowledge like a filter, and maintain the performance of the classification or decision-making of information system as best as possible. In the process of attribute reduction, a minimal subset of the original attribute set, which contains compulsory and important attributes, is looked for. The minimal subset should

represent the original attribute set without losing too much information. Finding this minimal subset is usually NP-hard and solved using approximate algorithms.

An approximate algorithm is an algorithm that is used to find an optimum solution, but there is no guarantee that the solution is the best one. Among many approximate algorithms, Genetic algorithm (GA) has been successfully and popularly used in optimization problems with its idea of Darwinian process of evolution. GA does not depend on any specific application areas, providing us with an effective way to deal with the search and optimization problems with limited cost. The motivation for applying GA to attribute reduction is that it can search the solution space with a global scale, and provides a robust and adaptive search performance. As an random search algorithm, GA can be considered as an "anytime" approach for learning, and could quickly give a good enough solution.

To perform attribute reduction using genetic algorithm, we employ a binary string to represent each candidate attribute reduction solution, which is usually called a chromosome. In other words, every chromosome is a 0–1 vector with the dimension same as the number of attributes, where "1" indicates the presence of corresponding attribute with a high probability, and "0" means that the corresponding attribute is absent or with a low probability of presence. Then the problem of attribute reduction boils down to find the best chromosome among the group of candidates generated with GA operations such as crossover and mutation, while the goodness of the solution is measured with a fitness function. Two factors are considered here in designing the fitness function, i.e., the compactness and the discriminative capability.

Specifically, a flower category decision table is defined as $S=(X,A,Y,V,f)$, where $X=\{x_1,x_2,\ldots,x_n\}$ is the set of flower samples, $A=\{a_1,a_2,\ldots,a_m\}$ is the set of attributes, $Y=\{y_1,y_2,\ldots,y_k\}$ is the set of class set, V is a set of the domain of the attribute, and $f: X\times A\cup Y\rightarrow V$ is an information function. Define $POS_A(Y)=\bigcup_{P_i\in X/Y}\{x|x\in X\wedge [x]_A\subseteq P_i\}$ be the positive region A with respect to Y, where $[x]_A$ means the equivalence class of x over attribute set A, and X/Y means a partition of the sample set X according to class set Y, with P_i as its i-th partition set. Hence $POS_A(Y)$ can be thought of as a measure of consistency between the current attribute sets A and the label set Y over the training images, and the dependence of the attribute set A with respect to class set Y can be defined as $r(A,Y)=|POS_A(Y)|/|X|$, where $|\bullet|$ means the number of elements in the collection [32,33].

The above basic concepts are helpful for us to define a good fitness function as follows:

$$F(L)=(M-\|L\|_0)/M+r(L,Y), \tag{8}$$

where L represents a chromosome, $\|L\|_0$ refers to the number of nonzero element in the chromosome L, M refers to the length of the initial chromosome, i.e. the number of initial attributes, $r(L,Y)$ means the dependence of the attribute set corresponding to the chromosome L, with respect to class set Y. In our implementation, an improved genetic algorithm [34] is adopted to solve the objective function (Eq. (8)). See Section 3.3 for more details.

3. Experiments

3.1. Experimental setting and arrangement

Data setting: In order to compare our methods' performance to the others, a public flower dataset called Oxford17 is chosen as the experimental dataset, http://www.robots.ox.ac.uk/vgg/data/flowers/index.html. The flowers dataset consists of 17 species of flowers with 80 images of each (Fig. 3). The samples of the Oxford17 are all natural images. In Oxford17, some categories of flowers have very distinctive visual appearance, e.g. tigerlilies and fritillaries, but some others have very similar appearance to each other, such as dandelions and colts feet. Besides these, there exist large viewpoint, scale, and illumination variations in this dataset. The intra-class diversity and tiny differences between categories make this dataset very challenging. Furthermore, it is difficult to distinguish the flower categories by one or two attributes of the flowers. For example, snowdrop cannot be discriminated from window flower only by color, and bluebell cannot be discriminated from cowslip only by shape [3].

We select 27 attributes (Fig. 4) to describe the flowers of the dataset. In order to validate the generalization performance of the proposed classification model based on the attribute, the dataset was divided into three sections. We randomly select the splits into 40 images per class for training, 10 images per class for verification of attribute-class probability relational matrix and 30 images per class for testing of attribute and class prediction. Because images of each class often contain multiple attributes, one image with multiple attributes will be reused to predict different attributes. So the training database contains 2720 pictures and some of them are repetitive. In this way, the number of training images for each attribute will reach around 100. The testing set has 510 images selected for testing the performance of attributes and classes prediction.

Features extracting: All images are aligned and resized to ensure the image size is 500 pixels. In order to represent each image, a series of features such as SIFT [36], PHOG [37] and local color histogram [38] were extracted. The final feature vector is 204 in dimension, including 100 dimensional SIFT features, 40 dimensional PHOG features and 64 dimensional RGB color histograms features.

Experimental settings: Dictionary construction in sparse representation should meet the requirement of over completeness,

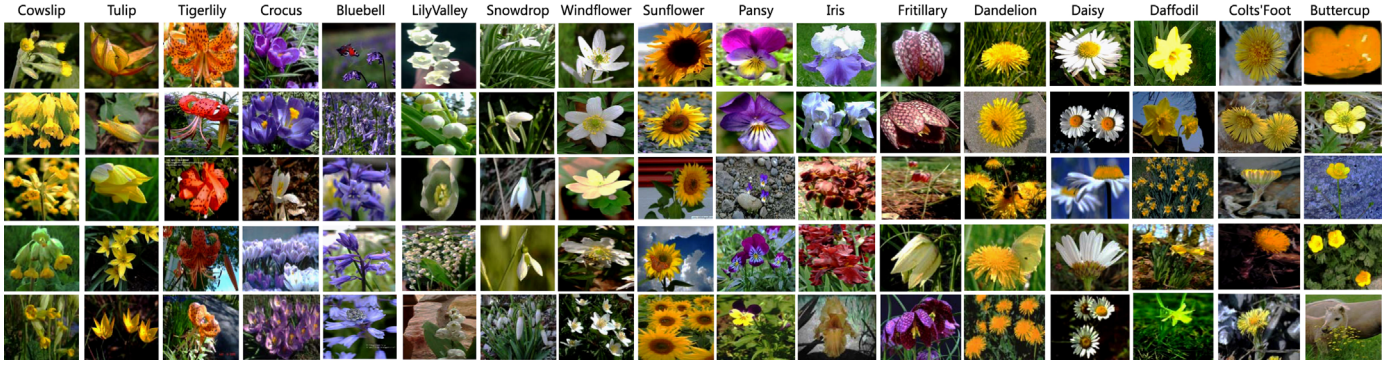

Fig. 3. Illustration of images in the Oxford 17 flower database.

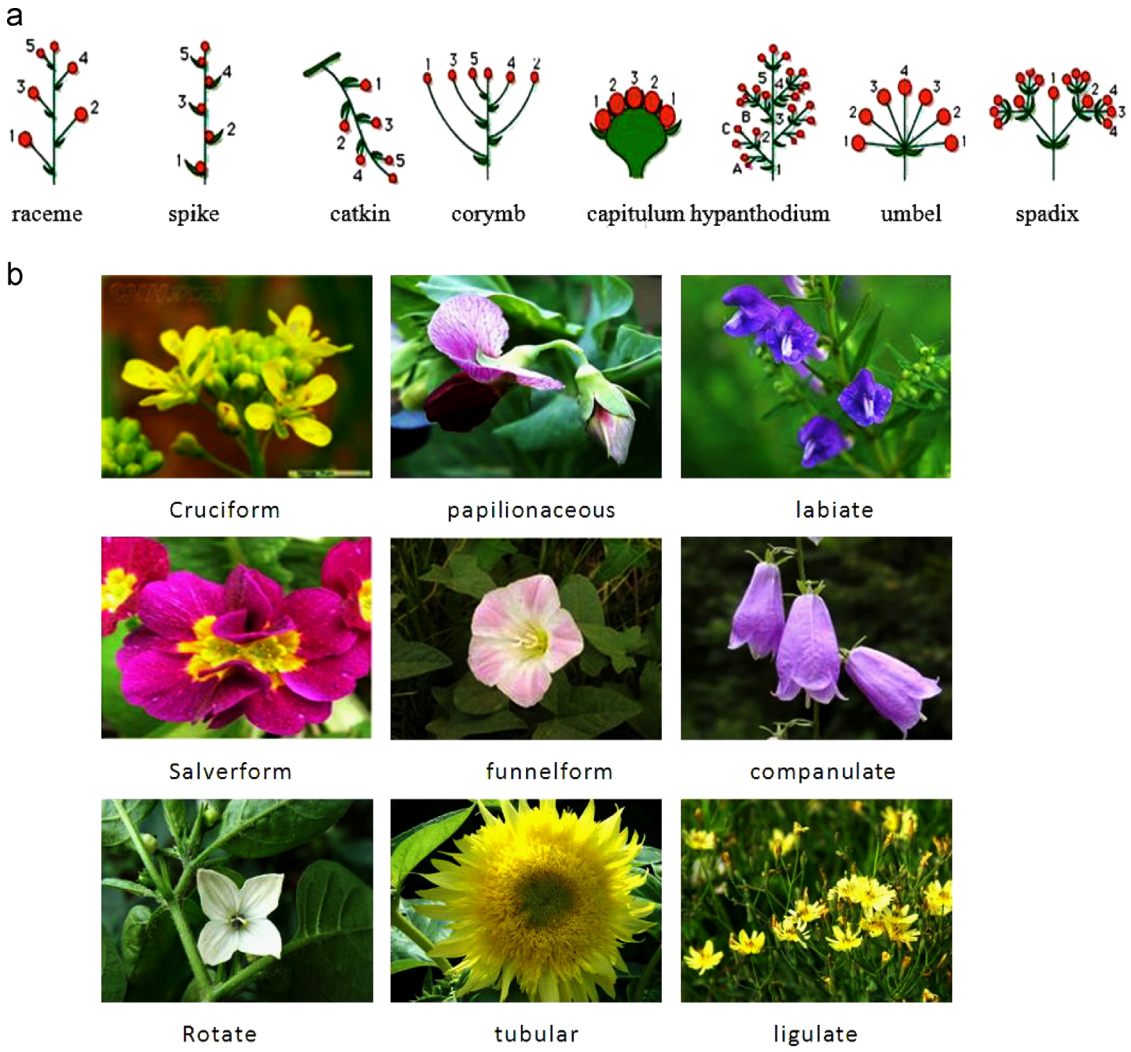

Fig. 4. Some attributes of flowers [35]. Attributes of (a) inflorescence and (b) flower shape.

namely the dimension of dictionary columns should be larger than the dimension of the feature. In our experiments, the dimension of the feature is set to be 204 and the number of dictionary columns is set to be 2720. In addition, the target threshold ϵ is set to be 1e-7 and the number of iterations is set to be 30.

Originally we have 27 attributes to describe flowers. After the attribute reduction, only 10 attributes remain. We use the approach introduced in Section 2.2 to predict attributes of each image and the class–attribute matrix. In the first step, with training data, the K-SVD algorithm is used to approximate the solution of Dictionary D. Then, use the training data's sparse representation to approximate the parameters of attributes' classifier. In the next step, with the testing data, use the OMP algorithm to obtain the sparse representation coefficient s. In the testing stage, with the representation of testing sample and the parameters of every attributes classifier, we can get the attribute classification posterior probability of the test data. Finally, we can get the label of test data with the Bayes rule.

3.2. Overall performance and comparison

Previously, several methods have reported their performance in the same Oxford 17 flower database. In [1], Nilsback and Zisserman developed and optimized a nearest neighbor classifier architecture on the segmented version of the dataset. They obtained 72.8% accuracy with a combination of HSV, SIFT and HOG descriptors. Later the same authors employed the "bag of visual words (BoW)" method to classify flowers, which improves the performance to 81.3%. With the same features, multiple kernel classifier [39] obtains a recognition rate of 85.2%. Ref. [40] shows the performance of column generation boosting for mixtures of kernels. It can give a recognition performance of 84.8%. LPBoost is another boosting approach which achieves 77.5% accuracy in flower classification [41]. These results besides ours are summarized in Fig. 5. Although our method does not provide a significant gain compared to others, it gives a good semantic explanation which cannot be given by other methods.

3.3. Critical steps of sparse representations based attribute learning

In this section, we investigate several factors that may have influence on the performance of our method, including attribute reduction, the effect of various amount of training images, and the performance of individual attribute learner.

In the experiment of attribute reduction based on the genetic algorithm, we first denote a chromosome as a binary string whose length is the number of the initial attributes. The parameters of the algorithm are set to as follows: the maximum evolution algebra is 20, the crossover probability is 0.7, and the mutation probability is 0.01. In order to transform the original problem of attributes reduction into a problem searching for minimum value, we change the fitness value to be opposite value of original fitness

value, i.e. $F(L) = -[(M - \|L\|_0)/M + r(A,Y)]$, where the parameters have been explained in Section 2. The target chromosome L, i.e. the reduced attributes set, is $\arg\min_L F(L)$. The the iteration process for objective optimization is illustrated in Fig. 6 and the best chromosome L is [00110111000110001000100010], which means that the set of attributes selected is $\{a3, a4, a6, a7, a8, a12, a13, a17, a21, a26\}$.

To investigate how many training images forming the dictionary are necessary for accurate class prediction, we run the experiments with various numbers of training examples used for attribute learning and test the final flower classification performance. Fig. 7 gives the accuracy corresponding to a specific number of training examples. It can be seen that with the increasing number of training examples, the flower classification tends to be more accurate. In addition, the figure also shows the benefit of attribute reduction. Actually, the discriminative attribute reduction helps us to reduce the overlapping regions between different classes, which in turn improves the final prediction performance.

To enable attribute-to-class mapping, the accuracy of the attribute prediction for a specific image is important. Fig. 8 illustrates some flowers and their corresponding attributes automatically learned with our system. We also investigated the role of individual attribute with respect to flower classification accuracy. First, Fig. 9 gives the test accuracy of our attribute learner. It can be seen that some attributes are more difficult to predict than others. The effective identification of an attribute is depending on whether it can be described in detail. Better defined attributes are easier to be discriminated. Second, Fig. 10 visualizes the attribute–class matrix which reveals the correlation between each attribute and each category.

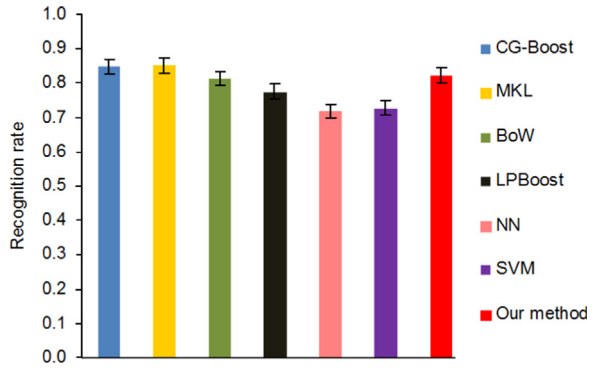

Fig. 5. Performance comparison between our method and previous methods [1,39–41].

3.4. Zero-shot learning based on attributes of flowers

To verify the performance of our method in zero-shot learning, the Oxford17 flower set is divided again into two new sets, for training and testing. Particularly, 12 categories and 50 samples of each class are chosen to be the source categories and are served to be the training set of attribute classifiers as well. The other five categories and 50 samples of each class are selected as the zero-shot learning classes and as the testing set. Altogether three different splits of classes for training and testing are designed. These three cross validation experiments are independent of each other, and our division of dataset is set in such a way that there are enough samples to train all the attribute classifiers. Finally, 30 samples of each class are randomly selected to test the performance of the attribute classifiers. Note that because all zero-shot learning classifiers base their decisions on the same learned attribute classifiers, the performance of category classifiers is determined solely by the performance of these attribute classifiers.

Figs. 11 and 12 respectively gives the accuracy of each individual attribute predictor and the corresponding ROC curve (AUC). One can see that the performance of learned attribute classifiers is significantly higher than the chance level of 0.5 in terms of AUC value.

The baseline methods we chosen to compare against our zero-shot classifier are the conventional 1NN and one vs. all SVMs. Similar to our zero-shot classifier, 12 categories with 50 samples of each class are chosen as the training set, and another five categories with 10 samples of each class are selected to train the classes classifiers for testing. At the testing stage, the five categories with 30 samples of each class are selected to evaluate the performance of these multi-class classifiers. Fig. 13 gives the comparative results between these methods and our method over 15 classes (five classes for each cross validation testing).

3.5. Computational complexity analysis

Flower classification is a multiclass classification problem. The multiclass classification problem can be decomposed into several binary classification tasks which are solved with a one-versus-rest or one-versus-one scheme using binary classifiers. There are many ways to deal with the multiclass classification problem, such as the

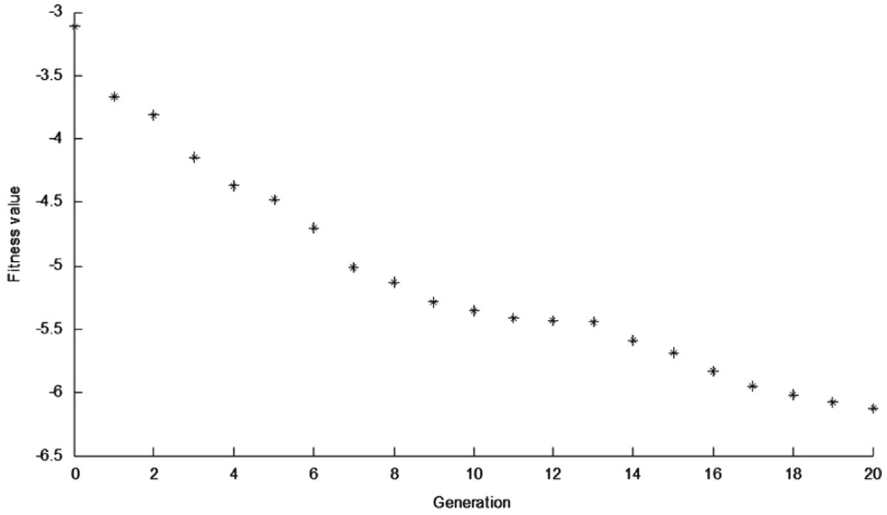

Fig. 6. Illustration of iterative process for attribute reduction with genetic algorithm.

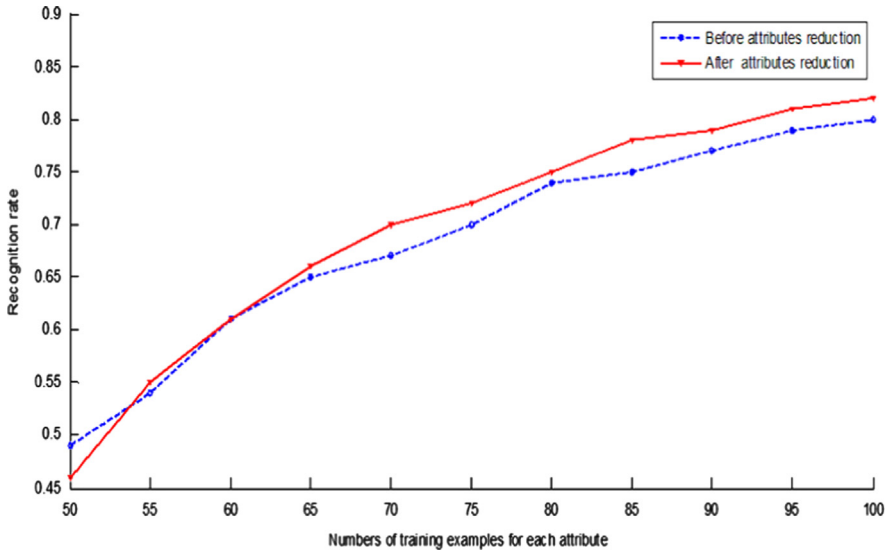

Fig. 7. Performance comparison before and after attributes reduction under different numbers of training images.

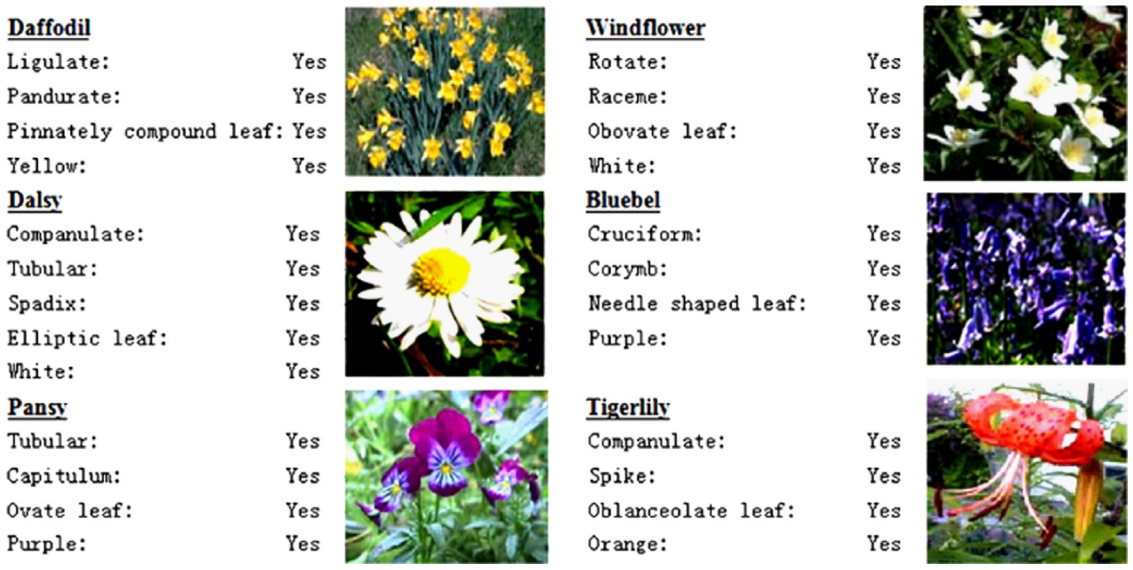

Fig. 8. Illustration of some flowers and the predicted attributes with our method.

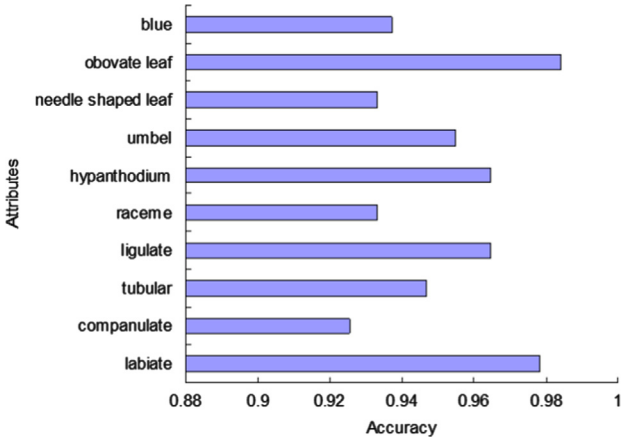

Fig. 9. The performance of individual attribute predictor on the flowers datasets.

KNN, MKL, CG-Boost. Owing to the large number of classes, the number of binary classifiers employed to deal with multiclass classification problem will increase enormously. By contrast, our method based on attribute learning with sparse representations is much efficient. Particularly, denote the number of classes as C, the number of training samples N, the number of attributes M, the dimension of test data's original feature d and the sparse representation's dimension l, the time complexity of traditional multi-class classification is $O(C^2N^2d)$, while ours is $O(MN^2l)$. Due to $M < N$ and $l < d$, the total computational complexity of the method based on attribute learning with sparse representation is smaller.

3.6. Discussion

Attribute learning method vs. non-attribute learning method: In the experiment, we compare our attribute learning method to some non-attribute learning methods such as CG-Boost and SVM. As one can see in Fig. 7, although our method does not provide a significant gain, it delivers a good semantic explanation and an ability to recognize new categories from purely textual descriptions. We can obtain three observations based on attribute contributions. First, the importance of the attributes is not same. This is why we can use attributes reduction method to search the core attributes for flower classification. In addition, better defined

attributes are easier to identify than more fuzzy ones. Second, the number of attributes is usually less than the number of classes, and the attribute classification is only a binary classifier and can be reused in the same categorization context. For this reason, the attribute learning method is very suitable for classification with a large number of category. The third observation is that attributes play a similar role not only in the sample learning prediction but also in the case of the zero shot learning. In other words, attributes can be learned without knowing the class labels of flowers. This observation further supports the argument of why attributes can be reliably employed in transfer learning.

Direct vs. indirect attribute prediction: A direct attribute prediction (DAP) learning method employed in our attribute learning frame adopts an attribute layer to separate the flower samples from the layer of their category tags. During the training stage, the output category label of each sample induces a deterministic labeling of the attribute layer. In addition, we can employ any supervised learning algorithm to obtain the parameter of attributes predictor. At the testing stage, we can use these trained attribute predictors to output the attribute labels of each test sample. Then, the class tag will be inferred from the attribute labels. Indirect attribute prediction (IAP) also employs the attributes to transfer knowledge from some classes to others. But the attribute layer is between two class layers, where the class labels of one layer are known at training time while the other one not. IAP can be used as a multi-class classifier. During the training stage of IAP, the assigned probability of each class is calculated using the training sample. Then, with the help of the class–attribute relation, the probability of each attribute can be obtained. At the testing phase, the normalized probability of each training class for the test image is predicted firstly. According to this posterior distribution over the training classes, the probability of each attribute for the test image is obtained by means of class–attribute relation. Then,

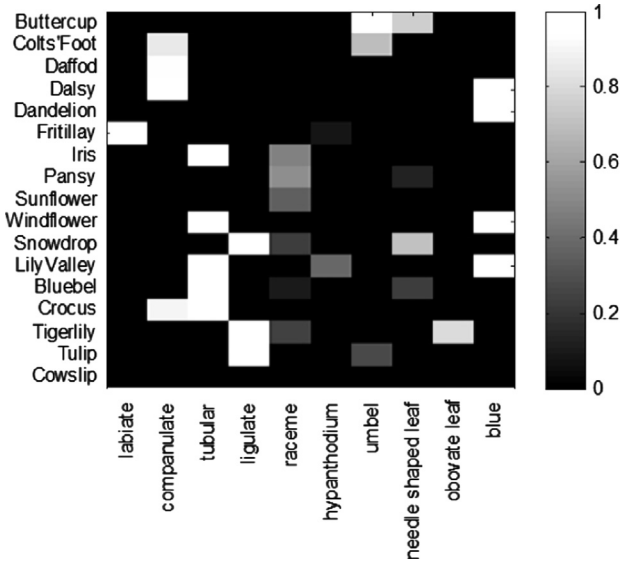

Fig. 10. Visualization of attribute–class matrix.

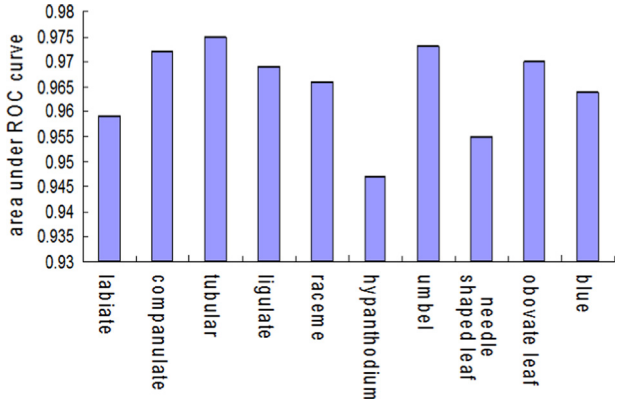

Fig. 12. Performance of each individual attribute predictor measured by the area under the ROC curve.

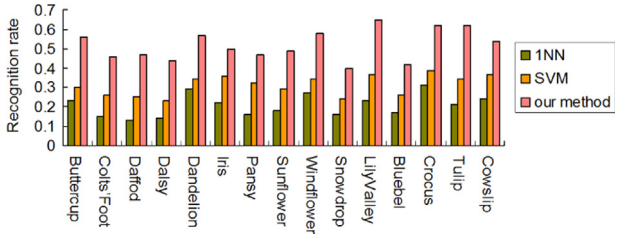

Fig. 13. Classification accuracy of our zero-shot learning method based on attributes, compared against 1NN and SVM.

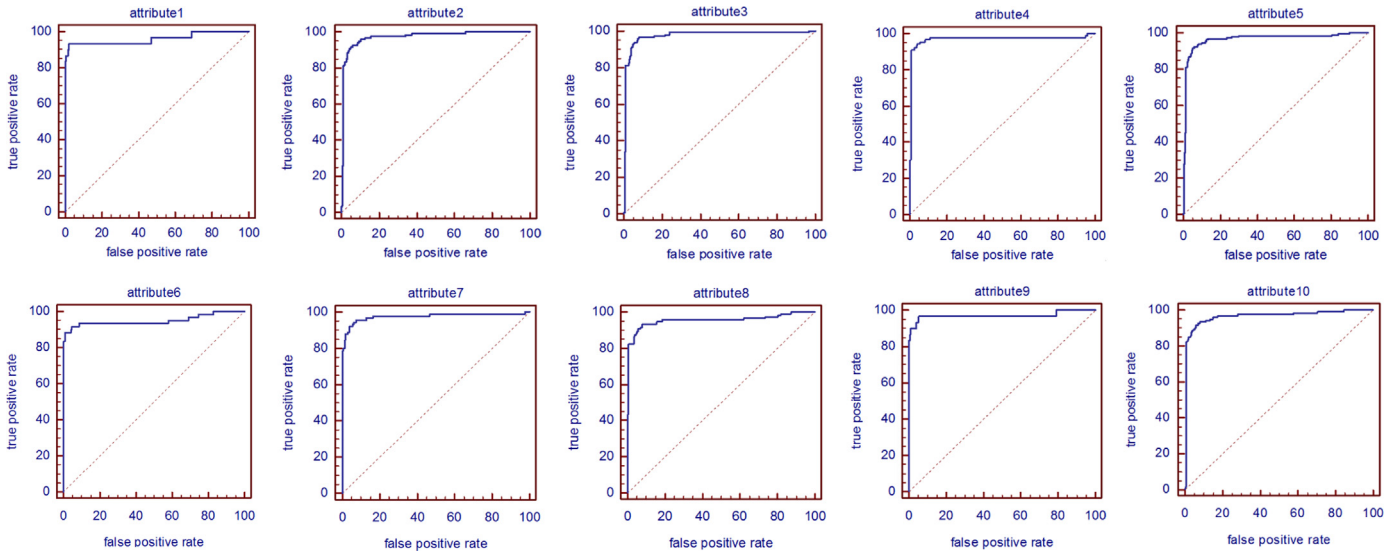

Fig. 11. The ROC curves of each attribute predictor.

Fig. 14. Example images that are correctly classified by the DAP scheme but are misclassified by the method of IAP.

the category tag of the test sample can be inferred from the labeling of the attribute layer. The major difference of the two methods lies on the relationship between training and test categories. In the DAP, all the classes are treated equally. In addition, the training and test classes are not disjoint. In the test stage, the class identification is based only on the attribute layer. In contrast, in the IAP, the training classes also occur in the test time as an intermediate layer. This leads to the following problem: in the zero-shot learning scenario, if there is a class of training samples which is relatively sensitive to the test sample (i.e., they are similar to each other in appearance but different in label), it will likely cause an error classification. This is why we chose the DAP scheme rather than the IAP in this work. Fig. 14 shows some example images of misclassified by the IAP method but are correctly classified by DAP.

Sparse representations vs. non-sparse representations. Employing sparse representations to the flower classification based on attributes learning has two advantages. For one thing, in the scenario of flower recognition, a high dimensional flower picture x can be exhibited by a vector with much lower dimensionality. That is why

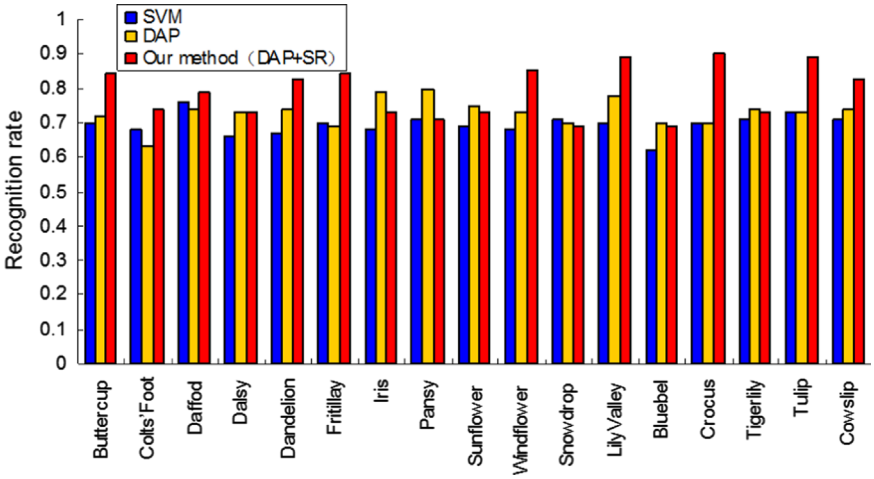

Fig. 15. Comparison of attribute classifiers using SVM, DAP and our method(DAP+SR).

we use the l1-norm regularization to describe flower images for improving the identification speed and reducing the storage cost. For another, in our experiment, owing to the limited training samples, we only use a general dictionary composed of samples with different attributes to describe the test samples. Assume that there are enough training samples for each attribute so that we can employ one set of training samples with attribute a_i, i.e., X_i, as the dictionary for this class, then a testing sample x with attribute a_i can be more sparsely represented over dictionary X_i than the general dictionary. The reason is as follows. If x is with attribute a_i, it is more likely that we can use only a few samples in X_i to represent x with a good accuracy. In contrast, we may need more samples with other attributes to represent x with nearly the same accuracy. With the sparsity constraint, the representation error of x by X_i will be visibly lower than that by the general dictionary, which will be helpful to the attribute prediction of x. For this reason, our sparse representation method not only uses the l1-norm regularization to describe flower images, but also could employ a certain sparsity constraint to control the representation error, making the classification of x easier (c.f., Fig. 15).

4. Conclusion

In this paper, a novel approach for flowers recognition is proposed based on the attribute learning. Instead of training for the recognition of a specific category of flowers directly based on the manually designed feature sets, a series of visual attributes are extracted from a given set of flower images, which are then generalized to new images from possibly unknown category. To automate the attribute extraction from a given image, a generative dictionary is learned from the training set, which facilitates a sparse representation based classification scheme for attribute prediction. Furthermore, the genetic algorithm is adopted to find the most compact and discriminative set of attributes for flower categorization. Extensive experiments on the publicly available Oxford flower database demonstrate the effectiveness of the proposed method.

Acknowledgment

This research is supported by the National Science Foundation of China (NFSC) Nos. 61170126, 61203246, 61003183, 61373060 and the Science Foundation of Jiangsu Province No. BK2011521.

References

[1] M. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: Proceedings of the Sixth Indian Conference on Computer Vision, 2008.
[2] D. Guru, Y. Sharath Kumar, S. Manjunath, Textural features in flower classification, Math. Comput. Model. Math. Comput. 54 (2011) 3–4.
[3] M. Nilsback, A. Zisserman, A visual vocabulary for flower classification, in: The Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2008.
[4] S. Takeshi, K. Toyohisa, Automatic recognition of wild flowers, in: The Proceedings of International Conference on Pattern Recognition, vol. 2, 2000, pp. 507–510.
[5] M. Das, R. Manmatha, M. Riseman, Indexing flower patent images using domain knowledge, IEEE Intell. Syst. 14 (5) (1999) 24–33.
[6] N. Shingo, S. Mie, A. Yoshimitsu, H. Shuji, Flower image database construction and its retrieval, in: The Proceedings of Korea–Japan Joint Workshop on Frontiers of Computer Vision, 2001, pp. 37–43.
[7] T. Saitoh, K. Aoki, T. Kaneko, Automatic recognition of blooming flowers, in: The Proceedings of International Conference on Pattern Recognition, vol. 1, 2004, pp. 27–30.
[8] E. Chang, B.T. Li, G. Wu, K. Goh, Statistical learning for effective visual information retrieval, in: The Proceedings of International Conference on Image Processing, 2003, pp. 609–612.
[9] Y. Freund, R. Yyer, R. Schapire, Y. Singer, An efficient boosting algorithm for combining preferences, J. Mach. Learn. Res. 4 (2003) 933–969.
[10] J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: a comprehensive study, Int. J. Comput. Vis. 73 (2) (2007) 213–238.
[11] T. Abe, T. Takada, H. Kawamura, T. Yasuno, N. Sonehara, Image-identification methods for camera-equipped mobile phones, in: The Proceedings of International Conference on Mobile Data Management, 2007.
[12] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, in: The Proceedings of International Conference on Computer Vision, vol. 88, 2010.
[13] G. Wang, D. Forsyth, Joint learning of visual attributes, object classes and visual saliency, in: The Proceedings of International Conference on Computer Vision, 2009.
[14] Y. Su, F. Jurie, Improving image classification using semantic attributes, J. Comput. Vis. 100(1) (2012) 59–77.
[15] A. Farhadi, I. Endres, D. Hoiem, D. Forsyth, Describing objects by their attributes, in: The Proceedings of International Conference on Computer Vision and Pattern Recognition, 2009.
[16] W. Kumar, A.C. Berg, P.N. Belhumeur, S.K. Nayar, Attribute and simile classifiers for face verification, in: The Proceedings of International Conference on Computer Vision, 2009.
[17] J. Vogel, B. Schiele, Semantic modeling of natural scenes for content-based image retrieval, Int. J. Comput. Vis. 72 (2) (2007) 133–157.
[18] D.A. Vaquero, R.S. Feris, D. Tran, L. Brown, A. Hampapur, Attribute-based people search in surveillance environments, in: The Proceedings of the Workshop on the Applications of Computer Vision, 2009.
[19] C.H. Lampert, H. Nickisch, S. Harmeling, Learning to detect unseen object classes by between-class attribute transfer, in: The Proceedings of International Conference on Computer Vision and Pattern Recognition, 2009.
[20] Z. Zha, T. Mei, J. Wang, Z. Wang, X. Hua, Graph based semi-supervised learning with multiple labels, J. Vis. Commun. Image Represent. 20 (2) (2009) 97–103.

[21] Z. Zha, X. Hua, T. Mei, J. Wang, Joint multi-label multi-instance learning for image classification, in: The Proceedings of Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[22] Z. Zha, Y. Zhang, Y. Yang, M. Wang, Interactive video indexing with statistical active learning, Multimedia 14 (1) (2012) 17–27.
[23] Z. Zha, L. Yang, T. Mao, M. Wang, Z. Wang, Visual query suggestion, in: The Proceedings of the 17th ACM international conference on Multimedia, 2009, pp. 15–24.
[24] D. Ta, W. Chen, N. Gelfand, K. Pulli, Efficient tracking and continuous object recognition using local feature descriptors, in: The Proceedings of International Conference on Computer Vision and Pattern Recognition, 2009.
[25] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, D. Schmalstieg, Pose tracking from natural features on mobile phones, in: The Proceedings of International Symposium on Mixed and Augmented Reality, 2008.
[26] D. Wagner, D. Schmalstieg, H. Bischof, Multiple target detection and tracking with guaranteed frame rates on mobile phones, in: The Proceedings of International Symposium on Mixed and Augmented Reality, 2009.
[27] R. Baraniuk, M. Davenport, R. DeVore, M. Wakin, Simple proof of the restricted isometry property for random matrices, Constr. Approx. 28 (3) (2008).
[28] E. Candes, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, IEEE Trans. Inf. Theory 52 (2) (2006).
[29] D. Donoho, Compressed sensing, IEEE Trans. Info. Theory 52 (4) (2004).
[30] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, B. Schiele, What helps where – and why? semantic relatedness for knowledge transfer, in: The Proceedings of International Conference on Computer Vision and Pattern Recognition, 2010.
[31] A. Yang, A. Ganesh, S. Sastry, Y. Ma, Fast l1-minimization algorithms and an application in robust face recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2010).
[32] Z. Pawlak, Rough set theory and its applications, J. Telecommun. Inf. Technol (2002) 7–10.
[33] D. Miao, G. Hu, heuristic algorithm for reduction of knowledge, J. Comput. Res. Dev 36 (6) (1999) 681–684.
[34] C. Zhang, J. Ruan, H. Zou, An improved genetic algorithm for attribute reduction in rough set theory, Int. J. Adv. Comput. Technol. 3 (8) (2011) 103–109.
[35] W.S. Judd, C.S. Campbell, E.A. Kellogg, P.F. Stevens, M.J. Donoghue, Plant Systematics: a phylogenetic approach, Sinauer Associates Inc., Sunderland, 2008.
[36] D. Lowe, Distinctive image features from scale-invariant keypoints, in: The Proceedings of International Conference on Computer Vision, 2004.
[37] K. Van de Sande, T. Gevers, C. Snoek, Evaluation of color descriptors for object and scene recognition, in: The Proceedings of International Conference on Computer Vision and Pattern Recognition, 2008.
[38] E. Shechtman, M. Irani, Matching local self-similarities across images and videos, in: The Proceedings of International Conference on Computer Vision and Pattern Recognition, 2007.
[39] M. Varma, D. Ray, Learning the discriminative power invariance trade-off, in: The Proceedings of International Conference on Computer Vision, 2007, pp. 1–8.
[40] K. Mikolajczyk, M. Awais, F. Yan, J. Kittler, Augmented kernel matrix vs classifier fusion for object recognition, in: The Proceedings of the British Machine Vision Conference, 2011, pp. 60.1–60.11.
[41] P.V. Gehler, S. Nowozin, On feature combination for multiclass object classification, in: The Proceedings of International Conference on Computer Vision, 2009, pp. 221–228.

Keyang Cheng is a member of CCF. He received the M.S. degree from the School of Computer Science and Telecommunication Engineering of Jiangsu University, in 2008. Now he is currently a Ph.D. student at the Department of Computer Science and Engineering, Nanjing University of Aeronautics & Astronautics. He has co-authored more than 20 journal and conference papers. He is currently a researcher and teaching assistant in the School of Computer Science and Telecommunications Engineering of Jiangsu University. His current research interests lie in the areas of pattern recognition, computational intelligence and computer vision.

Xiaoyang Tan received his B.Sc. and M.Sc. degrees in computer applications from Nanjing University of Aeronautics and Astronautics (NUAA), in 1993 and 1996, respectively. Then he worked at NUAA in June 1996 as an assistant lecturer. He received a Ph.D. degree from Department of Computer Science and Technology of Nanjing University, China, in 2005. From September 2006 to October 2007, he worked as a postdoctoral researcher in the LEAR (Learning and Recognition in Vision) team at INRIA Rhone-Alpesin Grenoble, France. His research interests are in face recognition, machine learning, pattern recognition, and computer vision. In these fields, he has authored or coauthored over 20 scientific papers.