Pattern Recognition Letters 49 (2014) 146–154

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Chronological classification of ancient paintings using appearance and shape features ☆

Qin Zou [a,*], Yu Cao [b], Qingquan Li [c], Chuanhe Huang [a], Song Wang [b]

[a] School of Computer Science, Wuhan University, Wuhan 430072, PR China
[b] Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208, USA
[c] Shenzhen Key Laboratory of Spatial Information Smart Sensing and Service, Shenzhen University, Guangdong 518060, PR China

ARTICLE INFO

Article history:
Received 26 November 2013
Available online 23 July 2014

Keywords:
Painting classification
Painting style analysis
Deep learning
Image classification

ABSTRACT

Ancient paintings are valuable for historians and archeologists to study the humanities, customs and economy of the corresponding eras. For this purpose, it is important to first determine the era in which a painting was drawn. This problem can be very challenging when the paintings from different eras present a same topic and only show subtle difference in terms of the painting styles. In this paper, we propose a novel computational approach to address this problem by using the appearance and shape features extracted from the paintings. In this approach, we first extract the appearance and shape features using the SIFT and *k*AS descriptors, respectively. We then encode these features with deep learning in an unsupervised way. Finally, we combine all the features in the form of bag-of-visual-words and train a classifier in a supervised fashion. In the experiments, we collect 660 *Flying-Apsaras* paintings from Mogao Grottoes in Dunhuang, China and classify them into three different eras, with very promising results.

1. Introduction

Ancient paintings have provided valuable sources for historians and archeologist to study the history and humanity at the corresponding eras. Fig. 1 displays four painting images collected from Mogao Grottoes in Dunhuang, China. These four paintings were created in different eras of China, namely the Wudai dynasty, the Sui dynasty, and the peak Tang dynasty, respectively. From these paintings, we can find a lot of important information in the corresponding eras, e.g., the architecture style in the Wudai dynasty from Fig. 1(a), the musical instruments in the Sui dynasty from Fig. 1(b), the plowing manner of farmers in the peak Tang dynasty from Fig. 1(c), and the costumes in the peak Tang dynasty from Fig. 1(d).

Obviously, a very important problem is to correctly determine the era in which a painting was created. Usually it is unreliable to determine the painting era based only on the specific content of the painting, since one same topic may be presented in the paintings in different eras. As widely adopted in art appraisal, people usually determine the era of a painting by examining its painting style, which usually varies with time and shows subtle differences from one era to another. It is usually difficult, if no impossible, for the general people without special training on painting and painting history to identify such subtle variation of the painting style for correctly determining the era of a painting. In this paper, our goal is to develop an automatic, computational approach to address this problem by using both appearance and shape features. Together with a supervised learning, we expect that the proposed approach can implicitly capture the specific painting style in different eras for painting-image classification.

To better capture the painting style implied in the paintings, we focus on the paintings created in different eras, but presenting the same topic. An example is shown in Fig. 2, where all 12 painting images present the *Flying Apsaras*, an important theme of the paintings of Mogao Grottoes in Dunhuang, China. These 12 paintings were created in different periods of Dunhuang Art: the infancy period (Row 1), the creative period (Row 2), and the mature period (Row 3). From these sample images, we can see that the painting-style difference in these three periods are subtle and only experts on Chinese classic art may be able to distinguish them. In this paper, we develop our automatic approach to localize such subtle difference and correctly determine the eras for such paintings.

Besides the appearance features, one interesting observation is that, each flying fairy in the *Flying-Apsaras* painting wears scarves, and it seems that the shape of the scarves varies from one period to another. An example is shown in Fig. 3. In the infancy period of the

☆ This paper has been recommended for acceptance by Jie Zou.

* Corresponding author. Tel.: +86 27 68775721; fax: +86 27 68778035.
E-mail addresses: qinnzou@gmail.com (Q. Zou), cao@cec.sc.edu (Y. Cao), qqli@whu.edu.cn (Q. Li), huangch@whu.edu.cn (C. Huang), songwang@sc.edu (S. Wang).

http://dx.doi.org/10.1016/j.patrec.2014.07.002
0167-8655/

Fig. 1. Sample paintings from different eras. (a) A painting in Wudai dynasty (907–960) from Mogao Grottoes 61, (b) a painting in Sui dynasty (581–618) from Mogao Grottoes 285, (c) a painting in peak Tang dynasty (712–762) from Mogao Grottoes 23, (d) a painting in peak Tang dynasty (712–762) from Mogao Grottoes 45.

Fig. 2. Sample paintings with the same topic but created in different eras. Row 1: four paintings created at the infancy period of the *Flying-Apsaras* art (421–556), Row 2: four paintings created at the creative period of the *Flying-Apsaras* art (557–618), Row 3: four paintings created at the mature period of the *Flying-Apsaras* art (619–959).

Flying-Apsaras art, line and curve strokes sketching the scarves are relatively simple and flat, as shown in Fig. 3left). In the creative period, scarves are sketched with small waves, as shown in Fig. 3middle). While in the mature period, scarves as well as clouds are more wavy than in the early periods, as shown in Fig. 3right). In this work, the main hypothesis is that the painting style can be described by the local appearance and shape features extracted from the painting images. This way, the difference of painting styles in different eras can be captured by learning from a set of training image samples. Based on this hypothesis, the proposed approach consists of the following steps: (1) appearance and shape features are extracted using the Scale-Invariant Feature Transform (SIFT) [21] and *k*AS shape description [7], (2) appearance features are further encoded by a deep-learning algorithm to enhance the representation abstraction ability, (3) visual codebooks are constructed based on the encoded appearance features and shape features, (4) feature histograms are produced for each painting as the input of the classifier, (5) training a classifier in a supervised fashion to determine the era of a painting based on the above feature histogram. In the experiments, we collect 660 *Flying-Apsaras* paintings from Mogao Grottoes in Dunhuang, China and classify them into either the infancy period, the creative period, or the mature period as shown in Fig. 2.

There are two major contributions in this paper. First, we developed a feature detection/combination method that can distinguish the subtle difference of the Dunhuang *Flying-Apsaras* paintings from different eras. We found that both appearance features and shape features are important for this classification task, which is consistent with the opinions of the art experts on Dunhuang paintings. Second, we proposed to use, in an original way, the combination of a typical set of image features (SIFT, an image-gradient based feature that are scale and rotation invariant) with one of the most effective feature refinement algorithms (deep learning, a specific type of Boltzmann machines). In the experiments, we compared the proposed method to a recent state-of-the-art painting classification method, with a clearly better performance.

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 presents our approach for extracting the appearance and shape features. Section 4 reports our experiment results on 660 *Flying-Apsaras* paintings and Section 5 concludes the paper.

2. Related work

As a branch of the image classification/retrieval research, painting classification has attracted more and more attention in the past two decades [26,17,28,29,15,8]. Existing painting-classification researches are usually focused on two applications, classifying

Fig. 3. Curve styles in the paintings from different eras. Top row from left to right are sample paintings from the infancy era, the creative era and the mature era, respectively. Bottom row are the detected curves on the corresponding painting and the red curves highlight the curve-style difference in these three paintings. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

paintings to specific painters [26,20,19,18,16], and classifying paintings to specific art movements [12,9,3,34,1,5,14,13].

Classifying Paintings to Painters To attribute a painting to a certain artist, a hierarchically structured classification scheme [26] was exploited by incorporating three different levels of information: the color, the shape of region, and the structure of brush strokes. It exploited the artist-specific and artist-independent characteristics of a painted portrait miniature. These characteristics are supposed to express the style of the brush stroke. The classification of painting styles was studied by using a palette description algorithm which describes the color content of paintings [20,19]. The classification of traditional Chinese paintings was explored by using wavelet decomposition based features, and 2-D multiresolution hidden Markov models were employed for classification [18]. A bag-of-words approach was adapted for classifying painters to different painters [16]. It employed the SIFT descriptor [21] and the Color Name descriptor [31] for creating feature vocabularies, respectively and showed that the combination of these two kinds of features leads to better classification performance.

Classifying Paintings to Art Movements. To classify the paintings to different art movements, e.g., classicism, impressionism, cubism, and romanticism, etc., six different features were extracted by applying statistic analysis to the image color, gradient, and intensity [12]. Extended research was performed by [9], which led to a prototype system. However, in these two papers, only a small set of real paintings were used for experiments and performance evaluation. By representing painting similarity using a Fisher-kernel metric [23], a classification system was built based on SIFT features and local color statistical features [3], which was reported to be capable of discovering non-obvious connections between the painters that belong to different art movements. Color distribution were exploited in the HSI space for several pre-defined groups of paintings and based on the color features, painting images were automatically classified for the application of image retrieval [14]. This work was further improved, where an MPEG-7 descriptor was adapted for extracting higher-level visual features, such as dominant colors, edginess, and textures, for painting image classification [13]. A 3-D color histogram and a Gabor filter energy were used for art description by [5].

Different from these related works, the goal of the proposed work is to classify the paintings to specific eras in which these paintings were created. Specifically, we focus on the paintings that present the same topic, such as *Flying-Apsaras* shown in Fig. 2, and have to identify the very subtle painting-style difference from one era to another for image classification.

3. Our approach

In this paper, we extract local appearance and shape features from a painting image and then use the bag-of-visual-words (BoV) technique to quantize and organize all these local features for classification. The flowchart of the proposed feature extraction approach is illustrated in Fig. 4. In the following, we elaborate on each of steps in detail.

3.1. Appearance features

One popular way to extract local appearance features is to uniformly partition the input image into a set of patches, as shown in Fig. 5(a), and then use simple intensity/color statistics in each patch as local appearance features. However, features extracted using such a patch-based method are usually sensitive to scaling and rotation transforms, which are very common for painting images. In this paper, we instead extract SIFT (Scale-Invariant Feature Transform) features [21] as the local appearance features. SIFT feature extraction consists of two steps: key-point detection and feature descriptor calculation.

Key-point detection. Let $I(x,y)$ be the input image, $L(x,y,\sigma_c), c = 1,2,\ldots,K$, be a sequence of smoothed images of I, by convolving I with 2D Gaussian filters as

$$L(x,y,\sigma_c) = \frac{1}{2\pi\sigma_c^2} exp\left(-\frac{x^2+y^2}{2\sigma_c^2}\right) * I(x,y), \quad (1)$$

where $\sigma_c, c = 1,2,\ldots,K$ is a sequence of monotonically increased scale parameters. Then the difference of Gaussian (DoG) image $D(x,y,\sigma_c)$ at scale c can be computed as

$$D(x,y,\sigma_c) = L(x,y,\sigma_{c+1}) - L(x,y,\sigma_c). \quad (2)$$

By stacking multiple DoG images, we actually obtain a 3D image (two spatial dimension and one scale dimension). We then check each pixel at each scale in its $3 \times 3 \times 3$ neighborhood. If it is a local extremum (by comparing with the other 26 pixels in the neighborhood), we take it as a key point [21]. In the key point

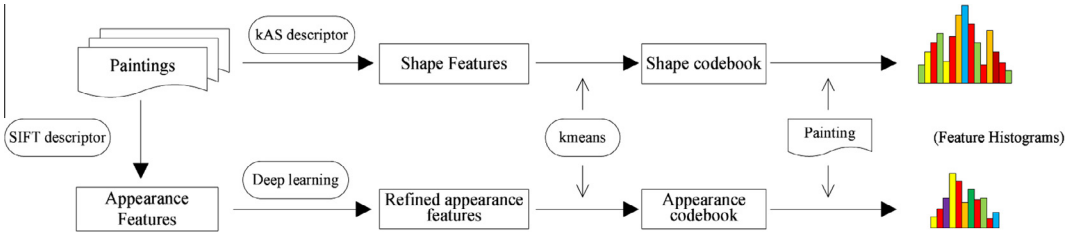

Fig. 4. The flowchart of the proposed approach.

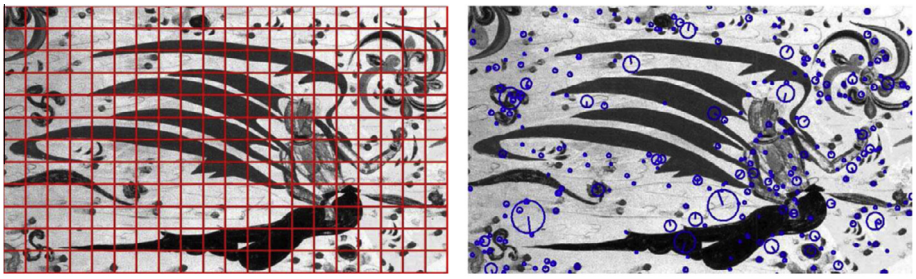

Fig. 5. An illustration of appearance feature extraction. (a) Dividing an image to patches for feature extraction, which is sensitive to scalings and rotation transforms. (b) Extracting SIFT features, which are invariant to scaling and rotation transforms.

detection, low contrast points and edge points are excluded since their features lack of discrimination powers.

SIFT feature descriptor At each detected point, SIFT descriptor is computed as the local appearance feature. For each key point (x,y), gradient magnitude $g(x,y)$ and orientation $\theta(x,y)$ is calculated at the scale where the key point is detected as

$$g(x,y) = \sqrt{f_x(x,y)^2 + f_y(x,y)^2}, \tag{3}$$

$$\theta(x,y) = tan^{-1}\frac{f_y(x,y)}{f_x(x,y)}, \tag{4}$$

where $f_x(x,y)$ and $f_y(x,y)$ are calculated by Eq. 5,

$$\begin{cases} f_x(x,y) = L(x+1,y,\sigma_c) - L(x-1,y,\sigma_c), \\ f_y(x,y) = L(x,y+1,\sigma_c) - L(x,y-1,\sigma_c), \end{cases} \tag{5}$$

with c being the scale in which the key point (x,y) is detected.

At each key point, a weighted histogram of 36 directions is constructed using the gradient magnitude and orientation in the region around the key point, and the peak that is 80% or more of the maximum value of the histogram is selected to be the principal orientation of the key point. After rotating the region around the key points to the principal orientation, the region is divided into blocks of 4×4, and the histogram of eight directions is computed at each block. Thus, a $4 \times 4 \times 8 = 128$ element feature vector is finally calculated as the SIFT descriptor at each key point. Fig. 5(b) shows the extracted SIFT features from a *Flying-Apsaras* painting image. Each SIFT key points is represented by a circle, where the scale is denoted by the radius of the circle and the principal direction is denoted by a line segment in the circle.

3.2. Shape features

For shape-feature extraction, we explore a local shape descriptor proposed by [7]. With this descriptor, the shape feature is derived from a chains k connected, roughly straight contour segments. In particular, we use $k = 3$, with which shape features are derived from Triple-Adjacent-Segments (TAS). A prototype TAS that represents a group of similar TASs is called a TAS codeword [32].

There are three steps to construct TASs [7,32]. First, edges are detected from the input painting image using the Berkeley Segmentation Engine (BSE) [22]. The BSE performs in a global way and produces soft edges, i.e., a probability map of the structural boundaries. Generally, BSE can obtain a better edge detection than the Canny edge detector. Second, small gaps along the contours are filled as follows: every edgel (a sequence of connected edge pixels) c_1 is linked to another edgel c_2, if c_2 passes a location that is near to the endpoint of c_1 and the ending of c_1 is directed towards c_2. Connected edges are then partitioned into roughly straight line segments. Finally, starting from each line segment, every triplet of the line segments is taken as a TAS (See Fig. 6).

Let s_i for $i = 1, 2, 3$ denote the three segments in a TAS P, and $r_i = (r_i^x, r_i^y)$ be the vector going from the midpoint of s_1 to the midpoint of s_i. Furthermore, let θ_i and l_i be the orientation and length of s_i respectively, then the descriptor of P is composed of 10 values [7]:

$$\left(\frac{r_2^x}{N_d}, \frac{r_2^y}{N_d}, \frac{r3^x}{N_d}, \frac{r_3^y}{N_d}, \theta_1, \theta_2, \theta_3, \frac{l_1}{N_d}, \frac{l_2}{N_d}, \frac{l_3}{N_d}\right). \tag{6}$$

The distance N_d between the two farthest midpoints is used as normalization factor, making the descriptor scale-invariant. In order to find TAS codewords, we need to define the distance between a pair of TASs. According to [7] and [32], the distance between two TASs, P^a and P^b, are defined by their locations, lengths, and orientations, that is

$$D(P^a, P^b) = w_\theta \sum_{i=1}^{3} D_\theta(\theta_i^a, \theta_i^b) + \sum_{i=1}^{3} |log(l_i^a / l_i^b)|, \tag{7}$$

where $D_\theta \in [0,1]$ is the difference between two segment orientations normalized by π. In this distance measure, the first term measures the difference in orientation and the second term measures the difference in length. A weight $w_\theta = 2$ is used to emphasize the difference between two TASs in orientation.

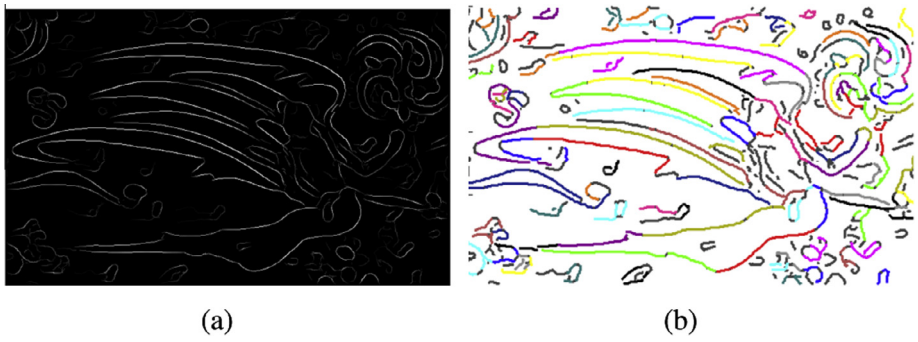

Fig. 6. An illustration of shape feature extraction. (a) Edge probability map computed using the Berkeley Segmentation Engine. (b) Detected TASs, where each TAS is represented by a connected curve segment with the same color.

3.3. SIFT feature refinement by deep learning

As a local appearance descriptor, SIFT is invariant to uniform scaling, and rotation. However, SIFT feature is sensitive to wide illumination variations and non-rigid transformations. In ancient paintings, worn-out regions are frequently observed and such regions can be viewed as strong illumination variations. Moreover, structural difference between two paintings, even for the two paintings in the same era (see Fig. 2), are usually non-rigid. In this paper, we use unsupervised deep learning to further refine the extracted SIFT features. This way, we utilize not only the scale/rotation invariance in the SIFT features, but also the representation abstraction ability of deep learning.

Deep learning [10,2,24] is motivated by the studies on visual cortex, which have revealed that the brain has a deep architecture and signals flow from one brain area (layer) to the next. Each layer of this feature hierarchy represents the input at a different level of abstraction. Deep learning is a computational feature abstraction strategy which simulates the function of the deep architecture of brain. In this paper, we use the popular deep belief networks (DBN) [10] for refining the SIFT features. Typically, the deep-learning procedure of DBN consists of two stages: 1) abstracting information layer by layer and 2) fine-tuning the whole deep network [10]. In the first stage, DBN tunes the weights between two adjacent layers by a family of Restricted Boltzmann Machines (RBMs) [25]. In the second stage, the weights in the whole deep network are fine-tuned using a contrastive version of the wake-sleep algorithm [10,33]. The first stage is unsupervised, which is also referred as *feature learning*, and the second stage is supervised, which requires the class labels for the input feature. In this paper, SIFT feature points are detected without any class labels. Therefore, we only use the first stage of the DBN deep-learning procedure, i.e., the unsupervised feature learning, to refine the detected SIFT features.

Without loss of generality, let us consider feature learning from a visible layer v to a hidden layer h, e.g., input feature layer H_0 to a hidden layer H_1 in Fig. 7(a). Each node represents a feature dimension in its respective layer. Assuming all nodes are binary random variables (0 or 1) and they satisfy the Boltzmann distribution, the bipartite graph formed by connecting the nodes across these two layers is a Restricted Boltzmann Machine (RBM).

In an RBM, a joint configuration of the visible and hidden units has an energy

$$E(v,h;\theta) = -\sum_{i,j} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j, \tag{8}$$

where θ denotes the parameters (W, a, b), W denotes the weights between visible and hidden units, a and b denote the bias of the hidden layer and the visible layer, respectively. Then the probability of the joint configuration is given by the Boltzmann distribution:

$$P_\theta(v,h) = \frac{1}{Z(\theta)} \exp(-E(v,h;\theta)), \tag{9}$$

where $Z(\theta) = \sum_{v,h}(\exp(-E(v,h;\theta))$ is the normalization factor. Combining Eq. (8) and Eq. (9) we have

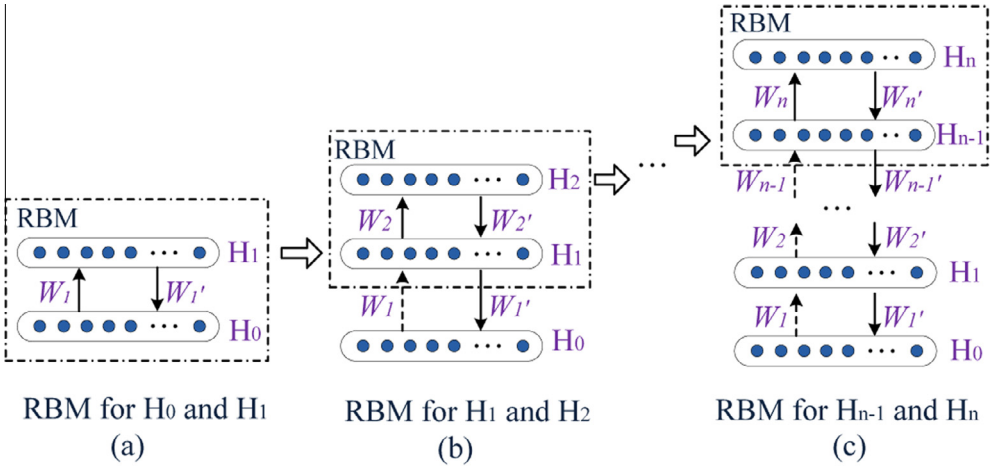

Fig. 7. An n-layer feature learning with RBMs. W and W' represent weights between the neighboring layers, down-up and up-down, respectively. H_0 is the input visible layer, and H_1 to H_n are the hidden layers.

$$P_\theta(v,h) = \frac{1}{Z(\theta)} \exp\left(\sum_{i,j} W_{ij} v_i h_j + \sum_i b_i v_i + \sum_j a_j h_j\right). \quad (10)$$

Because of the bipartite structure of RBMs, the visible and hidden units are conditionally independent to each other. Thus the marginal distribution of v respect to h can be written as

$$P_\theta(v) = \frac{1}{Z(\theta)} \exp[v^T W h + a^T h + b^T v]. \quad (11)$$

Parameters θ are then obtained by maximizing the likelihood of $P_\theta(v)$, which is equal to maximizing $\log(P_\theta(v))$. With the parameters θ, the hidden layer, say H_1 in Fig. 7(a), becomes a visible layer, based on which we can repeat the same algorithm to learn a new hidden layer, say H_2 in Fig. 7(b), and make it a visible layer. This process can be repeated to learn a multiple layer deep Boltzmann machine, as shown in Fig. 7(c).

In this paper, we use the DBN implementation [11][1] for feature refinement. Specifically, we try up to 4 hidden layers in the DBN with decreasing number of nodes. In the experiment, we will examine the SIFT features refined at each hidden layer and explore their representative abilities based on the painting image classification results. We will also investigate the impact of the number of the nodes at each hidden layer to the final classification performance.

3.4. Feature quantization and image classification

We construct feature codebooks for the appearance features and the TAS shape features separately. Given the large number of feature samples (e.g., more than 400,000 SIFT features, or more than 360,000 *k*AS features in our experiments) for codebook construction, we use the classical *K*-means algorithm to cluster the feature samples into a smaller group of cluster. Each cluster center is taken to be a feature-based visual word in the codebook. This way, any new feature sample can be quantized to its nearest visual words for constructing a feature-words histogram, i.e., a bag of visual words, for each image. Finally, this histogram is used for image classification. In the experiments, we will examine the impact of the number of clusters, i.e., the number of visual words in the codebook, i.e,, on the image-classification performance.

For classification, a multiclass Support Vector Machine (SVM) classifier using libSVM tool[2] was used for both training and testing [4]. The RBF was selected as the SVM kernel. There were mainly two parameters to be tuned in the classifier – the soft-margin constant C, and the γ in the RBF kernel, which will be examined in the experiments.

4. Experimental results and discussion

In this section, we first introduce the real painting image dataset we collected for evaluating the proposed approach. After that, we describe the experiment setup. At last, we report and analyze the experiment results.

4.1. Dataset

With the assistance of Dunhuang art researchers, we collected a set of 660 *Flying-Apsaras* painting images from Mogao Grottoes in Dunhuang. These images were labeled into three categories according to the eras they were created – 220 images from the infancy period of the *Flying-Apsaras* art (421–556), 220 images in the creative period of the *Flying-Apsaras* art (557–618), and 220 images from the mature period of the *Flying-Apsaras* art (619–959). Samples of the collected images are shown in Fig. 2. For each of the three categories, half of the collected data (110 images) were taken for training, and the remaining half were taken for testing.

4.2. Experiment setup

To fully justify the proposed feature extraction, we tried the following five type of features for image classification.

1. **Subimage**: Only use the image-patch based appearance features, as illustrated in Fig. 5(a). We specifically set patch size to be 28×28 in our experiments.
2. **SIFT**: Only use the SIFT-based appearance features.
3. ***k*AS**: Only use the *k*AS-based shape features.
4. **SIFT + *k*AS**: Combine SIFT-based appearance features and the *k*AS-based shape features, without any deep-learning-based refinement to these features.
5. **SIFT**$\times w_i$**+*k*AS**: Combine SIFT-based appearance features and the *k*AS-based shape features, where SIFT features are refined by using *i*-layer output of the deep-learning network.

Here SIFT$\times w_i$ + *k*AS is our proposed method and we tried up to four layers of deep-learning refinement to the SIFT features. For each type of feature, we use the same bag-of-visual-words technique to group them into a fixed-dimensional feature histogram before it is fed to the classifier. Note that, the dimension is 128 for the original SIFT descriptor, and 10 for *k*AS. The dimension of Subimage, the image-patch based feature used for comparison, is 784. After the deep learning, the dimension of the feature vector of SIFT$\times w_i$ is the number of nodes in the *i*th layer of the deep-learning network. For both *k*AS and SIFT, the codebook all contains 1024 codes, resulting from the *K*-means clustering. The codebook of SIFT$\times w_i$ + *k*AS that is used in the proposed method is 512.

In our experiments, we used vlFeat[3] tool [30] to obtain the SIFT features and the *k*-Adjacent-Segments detector[4] to generate *k*AS shape features. As discussed in Section 3.2, a *k*AS is a shape structure made up of k line segments. If k is smaller, the *k*AS will be simpler and can be used to fit more detected curve segments. By using a smaller k, the extracted local shape structures are simpler and become more frequent in the feature histogram [32]. In our experiments, we set $k = 3$ to detect TASs.

4.3. Experiment results

In the following, we first report the overall performance based on different features and then we study the impact of several important parameters in the proposed SIFT$\times w_i$ + *k*AS method.

Overall performance. we use two metrics for performance evaluation, the classification *Accuracy* and the *AUC* (area under the ROC curve). *Accuracy* was calculated by

$$Accuracy = \frac{true\ positives + true\ negatives}{positives + negatives} \times 100\%. \quad (12)$$

For two-class classification, *AUC* is the area under the ROC curve [6], which can be obtained by applying varied threshold to the output of the classifier. In the case of a multiclass classification, an average ROC curve is plotted to calculate the *AUC*. For both *Accuracy* and *AUC*, the bigger the value, the better the classification performance.

Fig. 8(a) and (b) show the image classification performances using the five types of features described above. In Fig. 8(a), we show the classification *Accuracy* in terms of varied γ, the kernel

[1] http://www.cs.toronto.edu/~hinton/MatlabForSciencePaper.html.
[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm/.
[3] http://www.vlfeat.org/.
[4] http://www.vision.ee.ethz.ch/software/index.en.html.

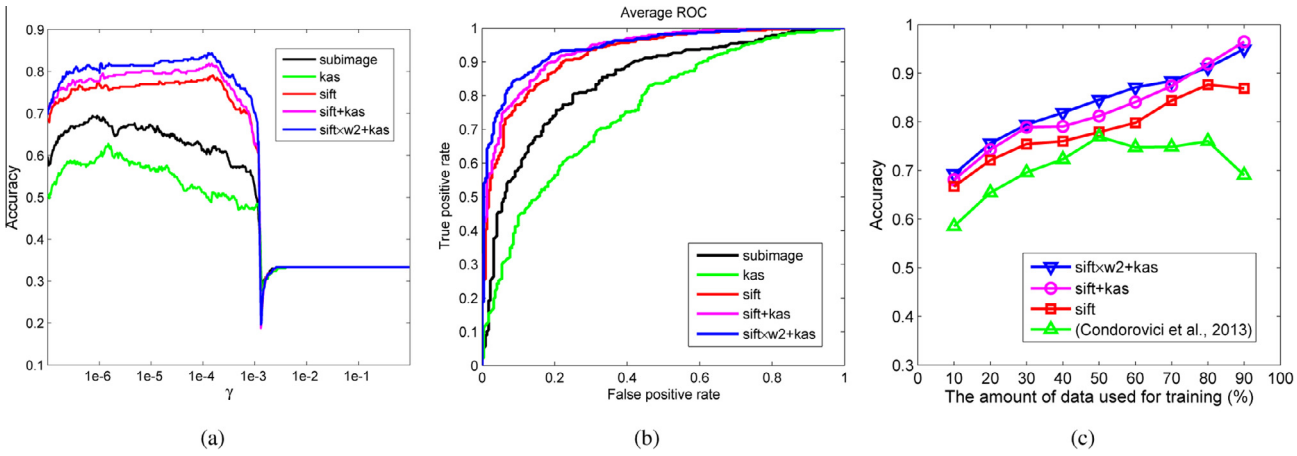

Fig. 8. Image classification performance when using different types of features. (a) Classification *Accuracy* in terms of varied γ, the kernel parameter in the SVM classifier, (b) ROC curves. For the above results, C is set 200 for the SVM classifier, (c) classification accuracy when using different amount of samples for training.

Table 1
Best *Accuracy* value and *AUC* using different types of features.

Metric	Features in use				
	Subimage	*k*AS	SIFT	SIFT + *k*AS	SIFT×w_2+*k*AS
Accuracy	69.39	62.73	77.88	82.12	84.24
AUC	0.843	0.757	0.922	0.931	0.936

Table 2
Classification accuracy of the proposed method (sift× w_2+kas) and a state-of-the-art comparison method [5].

Method	*Accuracy*			
	Infancy period	Creative period	Mature period	Overall
Proposed method	78.18	83.64	90.91	84.24
[5]	72.73	80.00	77.27	76.67

parameter in the SVM classifier. In Fig. 8(b), we show the average ROC curve for each type of feature. Table 1 shows the best *Accuracy* value and *AUC* obtained in each method. For the proposed method, we take the layer-2 output as the refined SIFT feature. We can see that the proposed SIFT×w_2 + *k*AS method produces the best classification accuracy and ROC performance against the comparison methods that use the other features. In particular, the combined SIFT + *k*AS features lead to better performance than using only SIFT-based appearance features or only *k*AS-based shape features. This indicates that the painting style in different eras is implied in both appearance and shape features. By refining the SIFT features using deep learning, the classification performance is further improved. This indicates that the deep learning does improve the representation abstraction ability of the SIFT features in this image-classification task.

To justify the proposed method, we compare its performance to one most recent method [5] that was developed for painting classification. This comparison method reported good accuracy in classifying 6 styles of paintings on a dataset of 3419 paintings. In addition, it was justified to be more favorable over several other painting-classification methods [9,34,27] in terms of the number of classes, the size of the painting datasets, and the reported classification accuracies. For this comparison method, we directly use its available source code[5], in which color-histogram and Gabor features are extracted for painting classification, and apply it to the Dunhuang *Flying-Apsaras* paintings. Similar as in the proposed method, we also use the same half of the paintings for training and the remaining half for testing in this comparison method. The results are shown in Table 2, where the accuracy is calculated for each class, as well as over all the test data. We can see that the proposed method outperforms this state-of-the-art comparison method in each of the class and over all the test data. Fig. 8(c) shows the accuracy of this comparison method when using different training samples. We believe the proposed method achieves a higher accuracy than the comparison method because the extracted SIFT features better describe the appearance of the paintings than the color-histogram and Gabor features. In addition, we extract shape features which are not considered in the comparison method.

Impact of parameters The kernel parameter γ in SVM defines how far the influence of a single training sample reaches – the smaller the value of this parameter, the farther a sample can reach in the classification. The parameter C in SVM trades off the misclassification of the training sample and the simplicity of the decision surface. The smaller the value of C, the smoother the resulting decision surface and the lower the classification rate on the training samples. In our experiments, we use a grid-search strategy to find the optimal parameter pair at $(C, \gamma) = (200, 2.0\times10^{-4})$ for the proposed approach, i.e., SIFT×w_2 + *k*AS. We found the performance of the proposed approach is not very sensitive to the value selection of C, especially when C is varied near 200. Therefore, we set $C = 200$ consistently, and evaluate the performance by varying the value of γ. The results of using different γ's are shown in Fig. 8(a). Clearly, the selection of an appropriate γ is important – if we select an overly large γ, say larger than $1e^{-3}$, the classification rate will drop substantially.

Besides the parameters in the SVM classifier, we also evaluated the proposed approach by using refined SIFT features from different layers of deep learning. On each layer, we can also vary *numNodes*, the number of nodes used for deep learning. Furthermore, we can vary *numCenters*, the number of clusters in constructing the feature codebook. In our experiments, we tried the value of *numCenters* to be 128, 256, 512 and 1024, and the value of *numNodes* to be 256, 512 and 1024. Table 3 shows *Accuracy* of the proposed SIFT×w_i+*k*AS method under different settings of these three parameter. In these results, we consistently set the SVM parameters $C = 200$ and search the best γ in the range of $[0, 1]$. From these two tables we can see that the best classification results were obtained at *numCenters* = 512 and *numNodes* = 512. The performance of the proposed approach does not change much when choosing *numCenters* $\in$ {256, 512} and *numCenters* $\in$ {512, 1024}.

[5] http://www.imag.pub.ro/common/staff/rcondorovici/rc_paint.htm.

Table 3
Impact of parameters on the performance in terms of *Accuracy*

Layers	*numNodes*	*numCenters*			
		128	256	512	1024
1 Layer (w_1)	256	80.61	82.12	81.52	81.21
	512	82.42	83.33	82.12	81.52
	1024	80.30	82.42	81.82	80.61
3 Layers (w_3)	256	83.94	83.64	80.30	83.94
	512	82.42	82.42	**84.24**	82.73
	1024	80.91	83.64	82.42	81.82
3 Layers (w_3)	256	79.39	79.39	83.94	81.52
	512	82.12	81.82	82.12	82.42
	1024	80.61	83.94	83.33	82.42
4 Layers (w_4)	256	82.42	82.12	82.12	83.03
	512	79.09	81.52	83.33	78.79
	1024	80.00	82.12	83.03	80.00

To further justify the proposed method, we conducted experiments by using different amount of training samples. Specifically, We tried the use of 10% to 90% (of the 660 paintings), at an interval of 10%, as the training data, and the rest were taken as the testing data. The classification accuracies in terms of different amount of training samples are plotted in Fig. 8(c). We can see that, in general (other than the case where more than 80% of data are used for training) the performance rank is SIFT$\times w_i$+*k*AS > SIFT + *k*AS > SIFT > [5]. The results further verify that the shape features do augment the appearance features for classifying the paintings and the SIFT-feature refinement by deep learning also improves the classification accuracy.

4.4. Discussion

The above experimental results have validated the hypothesis that both the appearance and the shape features are useful for classifying the Dunhuang *Flying-Apsaras* paintings in chronology. Specifically, by using only *k*AS shape features, we achieve a classification accuracy of 62.73%, which is much higher than 33.33% from random guessing. This shows that the shape features are informative for this painting-classification task. We consulted the art experts on Dunhuang paintings and they also believe that the highly wavy curves in the painting not only reflect the mature of the *Flying-Apsaras* art, but also express the glorious history of the peak Tang Dynasty.

From the experiments, we can also see that the chronology classification of the Dunhuang *Flying-Apsaras* paintings is a very challenging problem. The major reason is that the painting contents are the same, i.e., *Flying-Apsaras* and the difference of painting styles in different periods is subtle. The recently published comparison method [5] achieves an accuracy of 76.67% and the proposed method achieves the accuracy of 84.24% when using the best settings. This is not very high accuracy given that we only have three classes. In the future, we plan to work closely with the art experts to extract richer features for further improving the classification accuracy.

5. Conclusion

In this paper, we developed a new supervised approach to determine the era when an ancient painting is created. In this approach, we use the bag-of-visual-words technique to extract and quantize both the appearance feature and the shape feature from the painting images. Specifically, we used the SIFT descriptor and *k*AS shape descriptor to extract the appearance feature and shape feature, respectively. Moreover, we found the multi-layer deep learning technique can be used to refine the SIFT-based appearance features for improving the classification performance. We tested the proposed approach on 660 real painting images from Mogao Grottoes in Dunhuang of China and found that the proposed method can effectively classify them in three different eras by capturing the subtle difference of the painting style in each era.

Acknowledgments

This research was supported by the National Basic Research Program of China under Grant No. 2012CB725303, the China Postdoctoral Science Foundation funded project (2012M521472), National Natural Science Foundation of China (61301277 and 41371431), Hubei Provincial Natural Science Foundation (2013CFB299), the fund of AFOSR FA9550-1-1-0327, and NSF IIS-1017199.

References

[1] R.S. Arora, A. Elgammal, Towards automated classification of fine-art painting style: a comparative study, in: ICPR, 2012, pp. 3541–3544.
[2] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, in: NIPS, 2006, pp. 153–160.
[3] M. Bressan, C. Cifarelli, F. Perronnin, An analysis of the relationship between painters based on their work, in: ICIP, 2008, pp. 113–116.
[4] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27.
[5] R. Condorovici, C. Florea, R. Vranceanu, C. Vertan, Perceptually-inspired artistic genre identification system in digitized painting collections, Image Analysis, LNCS, vol. 7944, Springer, 2013. 687–696.
[6] T. Fawcett, An introduction to ROC analysis, Pattern Recogn. Lett. 27 (2006) 861–874.
[7] V. Ferrari, L. Fevrier, F. Jurie, C. Schmid, Groups of adjacent contour segments for object detection, IEEE TPAMI 30 (2008) 36–51.
[8] D.J. Graham, J.M. Hughes, H. Leder, D.N. Rockmore, Statistics, vision, and the analysis of artistic style, Wiley Interdisciplinary Rev. Comput. Stat. 4 (2012) 115–123.
[9] B. Günsel, S. Sariel, O. Icoglu, Content-based access to art paintings, in: ICIP, 2005, pp. 558–561.
[10] G.E. Hinton, S. Osindero, Y. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (2006) 1527–1554.
[11] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (2006) 504–507.
[12] O. Icoglu, B. Gunsel, S. Sariel, Classification and indexing of paintings based on art movements, in: Eusipco, 2004, pp. 749–752.
[13] I. Mitov, K. Markov, Features for art painting classification based on vector quantization of mpeg-7 descriptors, LNCS, vol. 6411, 2012, pp. 146–153.
[14] K. Ivanova, P.L. Stanchev, B. Dimitrov, Analysis of the distributions of color characteristics in art painting images, Serdica J. Comput. 2 (2008) 111–136.
[15] C. Jacobsen, M. Nielsen, Stylometry of paintings using hidden markov modelling of contourlet transforms, Signal Process. 93 (2013) 579–591.
[16] F.S. Khan, J. Weijer, M. Vanrell, Who painted this painting?, in: 2010 CREATE Conference, 2010, pp. 329–333.
[17] P.H. Lewis, K. Martinez, F.S. Abas, M.F.A. Fauzi, S.C.Y. Chan, M.J. Addis, M.J. Boniface, P. Grimwood, A. Stevenson, C. Lahanier, J. Stevenson, An integrated content and metadata based retrieval system for art, IEEE Trans. Image Process. 13 (2004) 302–313.
[18] J. Li, J. Wang, Studying digital imagery of ancient paintings by mixtures of stochastic models, IEEE Trans. Image Process 13 (2004) 340–353.
[19] T. Lombardi, The classification of style in fine-art painting (Dissertation), Pace University, 2005.
[20] T. Lombardi, S.H. Cha, C. Tappert, A graphical user interface for a fine-art painting image retrieval system, in: ACM SIGMM MIR, 2004, pp. 107–112.
[21] D.G. Lowe, Object recognition from local scale-invariant features, in: ICCV, 1999, pp. 1150–1157.
[22] D. Martin, C. Fowlkes, J. Malik, Learning to detect natural image boundaries using local brightness, color, and texture cues, IEEE TPAMI 26 (2004) 530–549.
[23] F. Perronnin, C. Dance, Fisher kernels on visual vocabularies for image categorization, in: CVPR, 2007, pp. 1–8.
[24] M. Ranzato, C. Poultney, S. Chopra, Y. LeCun, Efficient learning of sparse representations with an energy-based model, in: NIPS, 2006, pp. 1137–1144.
[25] D.E. Rumelhart, J.L. McClelland, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations, MIT Press, 1986.
[26] R. Sablatnig, P. Kammerer, E. Zolda, Hierarchical classification of paintings using face- and brush stroke models, in: ICPR, 1998, pp. 172–174.
[27] L. Shamir, T. Macura, N. Orlov, D. Eckley, I. Goldberg, Impressionism, expressionism, surrealism: Automated recognition of painters and schools of art, ACM Trans. Appl. Percept. 7 (2010) 1–17.
[28] D.G. Stork, Computer vision and computer graphics analysis of paintings and drawings: an introduction to the literature, in: CAIP, 2009, pp. 9–24.

[29] B. Temel, N. Kilic, B. Ozgultekin, O.N. Ucan, Separation of original paintings of matisse and his fakes using wavelet and artificial neural networks, J. Electr. Electron. Eng. 9 (2009) 791–796.
[30] A. Vedaldi, B. Fulkerson, VLFeat: an open and portable library of computer vision algorithms, 2008. <http://www.vlfeat.org/>.
[31] J. Weijer, C. Schmid, J. Verbeek, Learning color names from real-world images, in: CVPR, 2007, pp. 1–8.
[32] X. Yu, L. Yi, C. Fermuller, D. Doermann, Object detection using a shape codebook, in: BMVC, 2007, pp. 1–10.
[33] S. Zhong, Y. Liu, Y. Liu, Bilinear deep learning for image classification, ACM Multimedia (2011) 343–352.
[34] J. Zujovic, L. Gandy, S. Friedman, B. Pardo, T.N. Pappas, Classifying paintings by artistic genre: an analysis of features & classifiers, in: IEEE Int. Workshop on Multimedia Signal Processing, 2009, pp. 1–5.