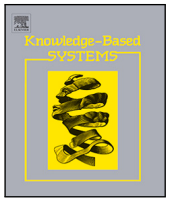
Knowledge-Based Systems 286 (2024) 111259

Contents lists available at ScienceDirect

Knowledge-Based Systems

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



A Sketch-texture Retrieval Framework using Perceptual Similarity

Yan Liu [a,1], Ying Gao [b,c,1], Nawaz Hafiza Sadia [c], Lin Qi [c], Junyu Dong [c,*]

[a] Department of Business, Qingdao Vocational and Technical College of Hotel Management, No. 599 Jiushuidong Road, Qingdao, China
[b] College of Data Science, Qingdao University of Science and Technology, No. 99 Songling Road, Qingdao, China
[c] College of Computer Science & Technology, Ocean University of China, No. 238 Songling Road, Qingdao, China

ARTICLE INFO

MSC:
0000
1111

Keywords:
Sketch retrieval
Perceptual similarity
Texture analysis

ABSTRACT

Sketch-based image retrieval is an important research topic in the field of image processing. Hand-drawn sketches consist only of contour lines, and lack detailed information such as color and textons. As a result, they differ significantly from color images in terms of image feature distribution, making sketch-based image retrieval a typical cross-domain retrieval problem. To solve this problem, we constructed a perceptual space consistent with both textures and sketches, and using perceptual similarity for sketch-based texture retrieval. To implement this approach, we first conduct a set of psychological experiments to analyze the similarity of visual perception of the textures, then we create a dataset of over a thousand hand-drawn sketches according to the textures. We proposed a layer-wise perceptual similarity learning method that integrates perceptual similarity, with which we trained a similarity prediction network to learn the perceptual similarity between hand-drawn sketches and natural texture images. The trained network can be used for perceptual similarity prediction and efficient retrieval. Our experimental results demonstrate the effectiveness of sketch-based texture retrieval using perceptual similarity.

1. Introduction

Images carry a wealth of data, and an increasing number of individuals are using them to represent semantic information. However, as the amount of image data grows, image retrieval has become a challenging problem, particularly for images that are difficult to describe precisely with words. In text-based image retrieval, users must use appropriate words to describe images in order to get satisfactory retrieval results. While they have knowledge of images and textures, it is often difficult to produces accurate descriptions of images. To address this problem, researchers have proposed using hand-drawn sketches as query data for retrieval. Sketches have the advantage of having a simple structure and few features, consisting entirely of contour lines, and are often utilized in a variety of scenarios. For example, an artist who enjoys painting can use their own hand-drawn sketches to develop new realistic and attractive images. Additionally, because hand-drawn sketches include rich semantic information, they can communicate a human's thoughts and emotions more effectively than words. Despite these advantages, finding features that can represent both sketches and textures simultaneously is challenging due to the substantial disparities between them.

Early hand-drawn sketch recognition methods typically followed traditional image classification models that extract manual features from hand-drawn images as input and send them to a classifier for classification. Common manual features include histogram of oriented gradient (HOG) [1] and scale-invariant feature transformation (SIFT) [2]. Building on this foundation, Hu et al. [3] proposed gradient field HOG (GF-HOG), an adapted form of the HOG descriptor suitable for sketch-based image retrieval. Eitz et al. [4] developed a bag-of-features sketch representation and used multi-class support vector machines for classification. These early methods have limitations in that they require manual feature extraction, which can be time-consuming and may not capture all relevant information.

Hand-drawn sketch recognition has seen significant improvements since 2012, thanks to advancements in deep learning. Sarvadevabhatla et al. [5] utilized two classical convolutional neural network structures for sketch recognition, while Yang et al. [6] trained a new convolutional neural network model and increased the receptive fields of the first convolutional layer to improve the model's generalization ability. Seddati et al. [7] used a deeper convolutional neural network to statistically characterize sketches and aid in the identification and retrieval of hand-drawn sketches. In 2016, Cai [8] proposed a sketch-based procedural texture retrieval method that maps from hand-drawn sketches to corresponding target textures using texture calculation features and enables the retrieval of hand-drawn sketches. However,

* Corresponding author.
E-mail addresses: qilin@ouc.edu.cn (L. Qi), dongjunyu@ouc.edu.cn (J. Dong).
[1] Contributed equally to this work and should be considered co-first authors.

https://doi.org/10.1016/j.knosys.2023.111259
Received 19 May 2022; Received in revised form 26 November 2023; Accepted 30 November 2023
Available online 4 January 2024
0950-7051/© 2023 Published by Elsevier B.V.

this method is only applicable to procedural textures and performs poorly on natural textures. Dong et al. [9] proposed a procedural texture generation framework based on semantic descriptions, which can generate procedural textures according to the semantic descriptions. However, this method has difficulty describing some complex textures that hand-drawn sketches can effectively capture.

To address these challenges, we propose a sketch-texture retrieval framework based on perceptual similarity. Given the significant differences between hand-drawn sketches and texture images, we construct a perceptual space consistent with both textures and sketches based on human perceptual similarity. We utilize a layer-wise perceptual similarity measurement method to learn image features for both texture images and hand-drawn sketches, extracting a robust feature representation that can effectively represent both types of images. We then train a perceptual similarity predicting network using the extracted paired image features, utilizing perceptual similarity values obtained from psychophysical experiments as training labels. The trained model can predict the perceptual similarity between the input hand-drawn sketch and the texture images, allowing for sorting of the texture images according to their similarity values and implementation of the sketch-based image retrieval framework.

The contributions are as follows:

1. We created a dataset of hand-drawn natural texture images. Currently, the commonly used datasets for sketch-based recognition and retrieval are object datasets, and there are no hand-drawn sketches available for texture images. To address this gap, we designed a sketch collection experiment in which we selected representative texture images from the natural texture dataset DTD [10] and used them as references for drawing sketches. We invited multiple subjects to draw sketches based on the given textures, resulting in the collection of over a thousand hand-drawn texture sketches.
2. We conducted a free-grouping psychophysical experiment to analyze the perceptual similarity of natural texture images, which allowed us to create a texture perceptual similarity matrix. We utilized the Isomap algorithm to construct a sketch-texture perceptual space, where textures and sketches that appear more similar in perceptual space are closer to each other, while textures and sketches that are significantly dissimilar are farther apart. This approach provides a new framework for sketch-based texture retrieval that is based on human perceptual similarity, allowing for more accurate and efficient retrieval of texture images based on their perceptual features.
3. We studied the feature extraction methods for both sketches and textures and utilized a combination of hand-drawn sketch textures and color images to obtain a new feature vector with human perception constraints. We then trained a fully connected similarity prediction network using these combined feature vectors, which can be used for sketch retrieval. This approach provides a more effective and accurate method for feature extraction and similarity prediction, allowing for improved performance in sketch-based texture retrieval.
4. Based on the perceptual similarity between natural texture images obtained from free-grouping psychophysical experiments, we propose a layer-wise perceptual similarity measurement method. This method enables end-to-end feature extraction and perceptual similarity prediction for paired images, which are then sorted according to their predicted similarity. We utilize this method to construct a sketch-based natural texture retrieval framework, allowing for efficient retrieval.

2. Related work

2.1. Texture retrieval

Texture image retrieval involves selecting a target texture image from a texture database based on certain rules. In essence, texture image retrieval is a type of content-based image retrieval. Texture images differ from natural images in that they are characterized by surface attributes and properties of objects. The analysis of texture images primarily relies on their texture properties, such as the repeatability of primitives in the image, local periodicity, and global periodicity. Texture image retrieval typically comprises two main parts: feature extraction and retrieval. The retrieval step involves distance metrics and classification.

There are various methods for sketch-based retrieval. In 1992, Hirata et al. [11] proposed an image search method based on hand-drawn sketches. They compared user-entered sketches with the edge extraction maps of 205 color oil paintings in the database, normalized all the images, divided each image into multiple small squares, and calculated global correlation by accumulating local correlations. Lopresti et al. [12] treated hand-drawn sketches as special forms of handwritten fonts, transforming the sketch search process into string matching problems. Additionally, research has been conducted on sketch matching and elastic matching. For instance, Del Bimbo et al. [13] and Sclaroff et al. [14] deformed user-drawn sketches by bending and stretching them, and then matched them with the contour of the object.

In recent years, many sketch-based retrieval methods have relied on deep learning techniques. One of the most well-known methods is DeepSketch, proposed by Seddati et al. [7]. They used deep convolutional neural networks to extract ConvNets features and performed sketch-based retrieval using k-Nearest Neighbors. Building on this work, Seddati et al. proposed DeepSketch2 [15] and DeepSketch3 [16], which compared different ConvNet architectures, training paradigms, and data fusion schemes, resulting in improved accuracy for hand-drawn sketch retrieval. In 2021, Seddati et al. [17] proposed an automatic coloring method for ethnic costume sketches, which achieved better coloring performance. However, most current sketch-based retrieval methods are tested on images and rarely on textures.

2.2. Computational features

Image features are essential elements in the field of computer vision and image processing. Over the past few decades, researchers have proposed various feature extraction methods for texture images. Generally, the computational features of textures can be categorized as either hand-crafted features or features extracted by deep neural networks.

Hand-crafted features mainly include statistical texture features and filter-based features [18,19]. Statistical texture features are often used to describe the spatial distribution of the gray values in images. Gray scale mean and gray histograms [18] are the simplest forms of representing the first-order features of an image. Gray-level co-occurrence matrix (GLCM) [20] summarizes the relative frequency distribution in the image. And, texture features extracted by the gray-level co-occurrence matrix have better discriminating ability but have limitations for pixel-level texture recognition tasks. Besides, absolute grey level difference histograms and local covariance matrices based on image features [20] have also achieved good results in many tasks. The Robert Crossover Operator [21], Canny Operator [22], and Marr Operator [23] can be used to extract lines, edges, and points in texture images. Additionally, orthogonal filters [24], Gabor filters [25], and wavelet transforms [26] can be used to extract more information from images.

The methods mentioned above for feature extraction are all hand-crafted features, and some of them have achieved good experimental results in the fields of object recognition and image classification. However, the same feature can perform differently in different tasks, and different tasks require different features. Researchers aim to find a robust texture representation method that can be applied to multiple texture classification tasks.

Deep features refer to features extracted through a multi-layer convolutional network and are a high-level representation of image data. Deep learning typically employs a multi-layered network structure,

where each layer of the network consists of multiple non-linear modules responsible for transforming the input data into a higher-level and more abstract representation [27]. The output of the previous layer serves as the input for the next layer which performs a new transformation. Common models for deep learning include AutoEncoder, Sparse Coding, Restricted Boltzmann Machine (RBM), Deep Belief Networks (DBN), and Convolutional Neural Networks (ConvNet). Convolutional neural networks [28,29] are particularly important deep network structures. A typical convolution network consists of three parts: convolution layers, nonlinear transformation layers, and pooling layers. The deep features learned by convolutional neural networks have been shown to perform better in many computer vision tasks than hand-crafted features [30–34].

2.3. Retrieval methods

There are mainly two methods for image retrieval: distance measurement and classification. Different classification or distance measurement methods have different effects on the time required for target searching, which in turn affects the retrieval efficiency. Distance measurement is an effective retrieval method that can show similar image forms by calculating the distance between two features of two images [35]. It is also widely used in texture retrieval. To calculate the distance between two textures, the computational features of the textures must be extracted in advance. The distance between the computational features of the texture images represents the similarity between textures. A smaller distance implies greater similarity between textures, and vice versa. Commonly used distance measurement methods include Euclidean distance, discrete cosine distance, chi-square distance, Manhattan distance, and KLD distance. The distance measurement methods are relatively simple and fast and is applied in many retrieval methods. However, these distance measurement methods lack robustness and perform well in some applications but not in others.

When using the classification method for retrieval, the most common approach is to use support vector machines (SVM) or K-Nearest Neighbor (KNN) to establish the classification model, and then use the model to predict and judge the retrieval target from the obtained categories to realize the retrieval process. For example, in [36], the feature vector and target values of given training samples were respectively classified using SVM and KNN. The algorithm automatically learns the classification model and predicts the corresponding target value for a new feature vector. Wang et al. [37] proposed a new integrated SVM classifier for relevance feedback content-based image retrieval. They combined the asymmetric bagging SVM and the random subspace SVM using EM parameter estimation, and further improved the relevance feedback performance.

Sketch-based image retrieval is a typical cross-modal retrieval task, and many cross-modal retrieval methods have recently been proposed. Qiang et al. [38] proposed a discriminatory deep asymmetric supervised hashing method for cross-modal retrieval, which reduces training time. Dong et al. [39] proposed a cross-modal graph attention strategy to generate the graph attention representation for each sample from the local graph of its corresponding paired sample, which eliminated the heterogeneous gap between modifications. Lei et al. [40] proposed a semi-heterogeneous three-way joint embedding network (Semi3-Net) that integrates a sketch branch, a natural image branch, and an edgemap branch, and introduces joint semantic embedding to learn invariant cross-domain representations effectively. Xu et al. [41] proposed a deep hashing framework for sketch retrieval, which embeds the temporal order of sketch strokes and achieves success in sketch recognition under zero-shot settings, as well as good generalization performance. Wang et al. [42] proposed a new compact feature learning method to embed the underlying manifold information from a database and build a landmark graph as a database sketch. The proposed method directly uses the index of the most similar codeword node as a compact feature representation. Liu et al. [43] proposed on-the-fly FG-SBIR, which performs image retrieval after each stroke and learns a joint embedding space shared between the photo and its corresponding complete sketch. The proposed method successfully achieves the goal of using the minimum number of strokes to retrieve the target photo. Cao et al. [44] proposed a cross-domain translation network and an intra-domain adaptation network for face photo-sketch synthesis, achieving the best perceptual appearance with less deformation. Cross-domain synthesis is another commonly encountered challenge in the domain of hand-drawn sketches.

Fig. 1. The process of the psychophysical experiments.

In our work, we compared the results of different distance metrics and classification methods, combined with various computational features. We proposed a layer-wise perceptual measurement method that integrates perceptual similarity. The effective experimental results demonstrate the feasibility of our method.

3. Dataset

3.1. Natural texture

As the research moves along, several natural texture databases, including Brodatz [1], OuTex [3], and CUReT [4], have been established. These texture datasets typically include texture images of different materials, such as wood, cloth, rock, etc. Many material-based classification methods have achieved good experimental results on these datasets. However, most of these datasets are lacking in human perception and do not contain descriptions of “real world” patterns.

The database selected for this paper is the Describable Textures Dataset (DTD) proposed in [10]. It contains 47 classes, with 120 texture images per class, totaling 5,640 images that are jointly annotated with 47 different attributes. In our experiment, we selected 10 representative images from each class, resulting in a new experimental dataset named DTD-R, which consists of a total of 470 images, as shown in Fig. 2. This is because psychophysical experiments can only be conducted on a limited quantity of experimental data, and testing a large amount of data is unfeasible due to labor and time constraints. The DTD-R dataset was used in two psychophysical experiments: (1) free-grouping experiments and (2) group-combining experiments (see Fig. 1).

3.2. Free-grouping experiments

3.2.1. Observers

A total of 20 undergraduate students participated in the free-grouping experiments. They do not have the expertise of texture analysis and have normal vision or corrected-to-normal vision.

Fig. 2. Typical textures in DTD-R Dataset. From left to right, the attributes of these textures are polka-dotted, zigzagged, banded, checked, frilly and honeycombed in order.

3.2.2. Procedure

The experiment was conducted in 470 images. The experiment was divided into two parts, free-grouping experiments and group-combining experiments.

A. Free-grouping

(1) A total of 470 samples were printed using the same printer and randomly divided into 10 sets, with each set containing 47 samples.
(2) During the experiment, the 47 samples in each set were spread out on a desktop. Observers were asked to group the samples freely based on their similarity, with samples in the same group expected to exhibit high similarity, while samples in different groups were to show no similarity. The number of groups and the number of samples in a group were not limited, and observers were free to move samples, combine groups, and open groups during the experiment.
(3) After grouping the first 47 samples, observers were then asked to group another 47 samples, placing them into existing groups or creating new groups. During this process, observers were also free to remove samples, combine or create new groups.
(4) This grouping experiment was repeated until all 10 sets were completed. It should be noted that a single image could not be used as a group, and there was no time limit for the free-grouping experiments. Once the observers had made the groups, they were not allowed to change the arrangement of sample images within each group, which we refer to as a locked group. This indicated that once the groups were finished, the observer was not permitted to make any further changes.

B. Group-combining

(1) After the free-grouping experiment, the observers were asked to combine groups freely, with no limitation on the number of groups. However, they were not allowed to move samples out of locked groups. Group combinations only occurred when there were similarities between different groups, and the observers were required to record their degree of confidence in the combination. During the free-grouping experiment, observers were free to categorize the 47 sample images provided in various ways based on their similar or dissimilar attributes. However, once the observers had made these categorizations, they were unable to change the arrangement of sample images inside that category, which we refer to as a locked category. This indicated that after the categories had been created, the observer was not permitted to make any further changes.
(2) The combining step was repeated until the observers believed that no further group combinations were possible.

3.2.3. Experimental analysis

The perceptual similarity matrix of 470 natural textures, denoted as $S_{texture}$, and the perceptual similarity matrix of 47 different classes, denoted as S_{class}, were calculated based on the results of the psychological experiments.

Given an observer $i(i = 1, 2, \ldots, 20)$, the matrix $S_i^{free-grouping}$ with a size of 470×470 is created, and each row and column of the matrix represents a sample of texture. During the free-grouping experiments, for each observer, the sample m and the sample n are grouped in the same group, the corresponding element $S_{(m,n)}^{free-grouping}$ in the matrix is set to 1, and the samples that are not divided into the same group are set to 0, as a result of binary matrix $S_i^{free-grouping}$ is obtained. The experimental results of 20 observers can be counted into 20 binary matrixes. Adding the 20 binary matrixes together:

$$S_{all}^{free-grouping} = S_1^{free-grouping} + \cdots + S_{20}^{free-grouping} \tag{1}$$

the elements in the matrix represent the number of times when sample m and sample n are grouped by different observers. Dividing the obtained similarity matrix by the number of observers participating in the experiment, the element values in the similarity matrix are normalized between 0 and 1, which obtains the perceptual similarity matrix of the free-grouping experiments $S_{texture}^{free-grouping}$.

Next, we super impose the results from group-combining to $S_{texture}^{free-grouping}$. During each group-combining experiment, the new groups are obtained with self-confident degree, multiply the results of the combined groups with confidence and add them to the matrix of free grouping. The final similarity matrix can be written as

$$S_{texture} = \frac{1}{N} \times \sum_{N}^{i=1} (S_i^{free-grouping} + \alpha S_i^{group-combining})$$

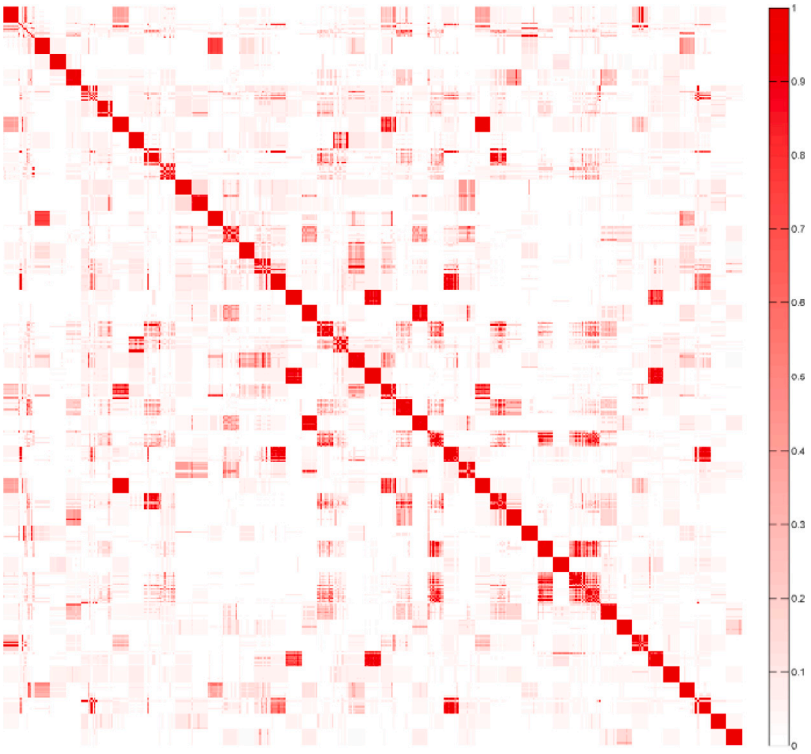


Fig. 3. The 470 × 470 texture perceptual similarity matrix obtained by free-grouping experiment. The abscissa and ordinate coordinates represent different texture samples, and the color bar on the right side of the graph indicates the similarity values of different colors.

$$= \frac{1}{N} \times \sum_{N}^{i=1} \left(\sum_{470}^{m,n=1} S_{m,n}^{free-grouping} + \alpha \sum_{470}^{m,n=1} S_{m,n}^{free-grouping} \right) \tag{2}$$

where α is the self-confident degree in group combining experiments. The element $S_{(m,n)}$ in $S_{texture}$ represents the similarity coefficients of sample m and sample n. The closer $S_{m,n}$ is to 1, the more observers divided the two samples into one group that is the sample pairs (m, n) are more similar to each other.

Fig. 3 shows the texture similarity matrix $S_{texture}$ obtained from the free-grouping experiments. The abscissa and ordinate coordinates represent different texture samples, and the similarity of sample pairs is shown by the color. The various shades of red at coordinate points (m, n) represent the similarity between sample m and sample n. The deeper the color, the higher the similarity between the samples. From the graph, we can observe that the color of most sub-blocks tends to be white, indicating that the similarity between texture samples of different categories is very low, while the similarity between textures of the same category is high.

To analyze the relationships between different categories in the dataset, a category similarity matrix is constructed based on the similarity between textures of different categories. If the similarity between the textures in two categories is higher, then the similarity between the two categories will be considered higher, and vice versa. The process of obtaining the category similarity matrix is as follows: a matrix with size 47 × 47 is created, where each row and column represents a category of texture. Each entry of the matrix is calculated by averaging the similarity coefficients between samples of the two compared categories. Based on the sample similarity matrix $S_{texture}$, the similarity coefficients between samples of different categories are averaged, i.e., the similarity coefficients between samples of category P and Q. As shown in Fig. 4, the resulting matrix S_{class} provides a rough estimation of the similarity between categories in the DTD-R database.

The texture category similarity matrix can provide a more intuitive representation of the similarities between different categories. For example, the sub-blocks corresponding to Category 1 and Category 3 are white, indicating that there is no similarity between the two types of texture samples. And the sub-blocks corresponding to Category 8 and Category 31 are relatively more red, indicating that the texture samples of these two categories are similar to each other. From the figure, we can observe that in the dataset, there is relatively low similarity between different categories in most cases.

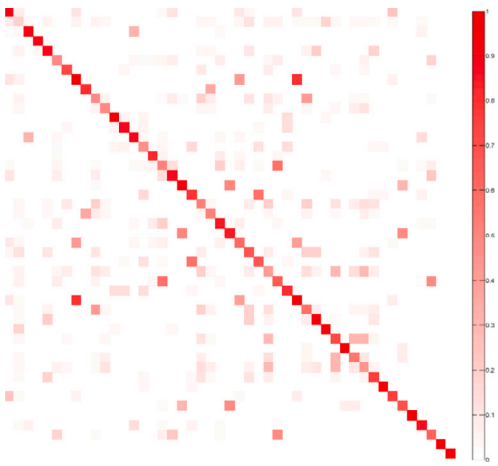


Fig. 4. The 47 × 47 texture category similarity matrix obtained by the experiment. The abscissa and ordinate coordinates represent different texture categories, and the color bar on the right side of the graph indicates the similarity values of different colors.

3.3. Sketch data collection

3.3.1. Observers

A total of 12 undergraduate students participated in the sketch collection experiments. The participants were not familiar with texture images, except for two who have painting skills.

Fig. 5. The original texture image and corresponding sketch. The images above are the original textures and below are the corresponding sketches.

3.3.2. Procedure

Participants were provided with natural texture images and pre-designed white paper with boxes. They were instructed to use pencils to draw sketches to fill the boxes. Ten of the participants each drew 20 sketches, while the other two participants with drawing experience drew 470 sketches. For each texture presented by the computer, the participants drew a picture based on the texture image and then recorded the texture number and their name. There was no time limit for drawing, and the participants were free to take breaks at any point during the process. After the drawing was completed, the participants' sketches were collected and scanned using an HP (M1552n) scanner with a resolution of 300 pixels.

3.3.3. Experimental analysis

A total of 1,140 hand-drawn sketches were collected in the experiment. By using MATLAB to crop and binarize the scanned images, we obtained the DTD-S Dataset of hand-drawn texture sketches. The original texture image and its corresponding hand-drawn image are shown in Fig. 5.

The participants took slightly different amounts of time to draw the sketches, with an average time of 5 min per texture. Sketching a set of 470 texture images took approximately 40 h. These data are highly valuable. Two participants who did not participate in the sketch collection experiments took part in the verification experiment. Both participants agreed that the sketches could represent natural textures. During the verification experiment, the sketches and the natural textures referenced during the sketch drawing were displayed on a computer screen, and the participants were asked to indicate whether the sketches were consistent with the original textures using a Y/N response. The results showed that the hand-drawn sketches are valid.

4. Sketch-texture retrieval framework

4.1. Sktech-texture perceptual space

The Isomap algorithm can construct a low-dimensional space while preserving the internal geometric structure of data points in the space by calculating the distance between data points on the global geometric

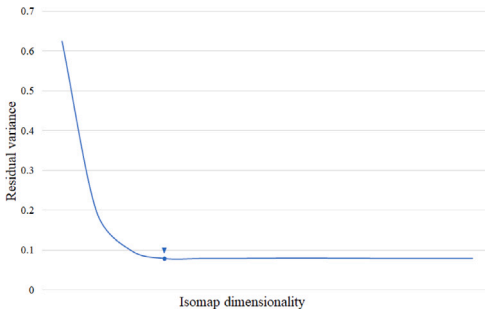


Fig. 6. The relationship between the reduced dimension and residual variance.

manifold. Let X represent the high-dimensional input space, $D_X(i,j)$ represent the original input distance between points i and j, Y represent the output d-dimensional Euclidean space, Y_i represent the output coordinate vector, D_G represent the geodesic distance matrix, $D_G(i,j)$ represent the geodesic distance between point i and j in G, and ϵ be a self-defined parameter. The steps of the Isomap algorithm are as follows:

Construct neighborhood graph. If the distance between i and j $d_X(i,j) < \epsilon$, there is a distance $d_X(i,j)$ between points i and j. For all points, G is constructed as described above.

Calculate the shortest path and estimate the geodesic distance. Initialize the geodesic distance matrix $d_G(i,j)$, if there is a distance between i and j, then $d_G(i,j) = d_X(i,j)$, otherwise, $d_G(i,j) = \infty$. For $k = 1, 2, \ldots, N$, calculate $mind_G(i,j), d_G(i,k) + d_G(k,j)$, use the minimum value to replace all elements in D, and the obtained element $D_G = d_G(i,j)$ in geodesic matrix contains the shortest path between all the points in G.

Construct d-dimensional European space. Perform eigenvalue decomposition on the inner product matrix $\tau(D_G)$ of geodesic matrix, λ_p is the pth eigenvalue of matrix $\tau(D_G)$, V_p^i is the ith component of the pth eigenvector, and the pth component of coordinate vector y_i in d-dimensional Euclidean space is $\sqrt{\lambda_p} v_p^i$.

Thus, the coordinate y_i in the low-dimensional space can represent the points in the original high-dimensional space. The dimension of the data can be estimated by reducing the error as the dimension of Y increases. Through the relationship between the reduced dimension and residual variance, we can select the appropriate dimension of the sketch-texture perceptual space. Fig. 6 illustrates the relationship between the reduced dimensions and residual variances. As shown in Fig. 6, the residual variance gradually decreases as the dimension increases. However, at a certain dimension, the residual variance will not decrease significantly. We can choose the dimension corresponding to the inflection point of the curve as the sketch-texture perceptual space dimension. From the figure, we can observe that the 48-dimensional perceptual space can better capture the similarity in the original high-dimensional space, with a corresponding residual variance of $r = 0.0782$.

4.2. Feature extraction and similarity measurement

Feature extraction is a crucial step in texture retrieval, as appropriate features can greatly improve the accuracy and efficiency of retrieval. In this paper, we evaluate four representative feature extraction methods, including Gabor [25], LBP [45], PCANet [46], and AlexNet [47]. The first two methods are manual feature extraction techniques, where the corresponding features are designed manually. The latter two methods are deep learning-based methods, where the corresponding features are referred to as deep features. These features have achieved excellent classification results on various texture datasets.

Table 1
Correlation coefficient between textures and sketches.

FEATURE	GABOR	LBP	PACNET	ALEXNET
CORRELATIONS	**0.5567**	0.2382	0.4434	**0.5396**

Distance measurement is widely used in texture retrieval and texture similarity estimation. To evaluate the effectiveness of the four computational features in representing both the hand-drawn sketches and the original texture images, we calculate the distance between the sketches and the distance between the corresponding natural textures using distance measures. We also conduct correlation analysis experiments using the calculated distances. The distance metric used in this paper is cosine similarity:

$$sim\langle F^{img1}, F^{img2}\rangle = \langle F^{img1}, F^{img2}\rangle / \|F^{img1}\| \|F^{img2}\| \quad (3)$$

$$sim\langle F^{skth1}, F^{skth2}\rangle = \langle F^{skth1}, F^{skth2}\rangle / \|F^{skth1}\| \|F^{skth2}\| \quad (4)$$

where F^{img1} and F^{img2} represents the feature vectors of texture images in the database and F^{skth1} and F^{skth2} represents the feature vectors of the corresponding sketches. According to the formula, the similarity between natural textures and the similarity between sketches can be calculated. The distance is calculated on the four feature spaces mentioned above, and the correlation coefficient between different textures and corresponding sketches are calculated.

Table 1 shows that there is a certain correlation between the original natural texture image and the corresponding hand-drawn sketch, and the correlation values calculated by different features vary significantly. We aim to identify a feature that can represent texture images and hand-drawn sketches as closely as possible, so that the same type of texture image and hand-drawn sketch have similar features.

To evaluate the performance of the different features in sketch-based retrieval, we conducted experiments using traditional distance metrics. Figs. 7 and 8 illustrate the retrieved results using LBP, Gabor, PCANet, AlexNet, LBP+PCANet, LBP+AlexNet, Gabor+PCANet, and Gabor+AlexNet features, respectively.

Based on the experimental results, we observed that the distance measurement method using these feature spaces cannot retrieve the desired texture based on the sketch. The retrieved results are significantly different from the ground truth values, which does not meet the requirements of accurate texture retrieval.

To further evaluate the expressive ability of features, we propose a perceptual similarity learning method based on stacked sparse auto-encoder. We combine hand-drawn sketch texture features with natural texture image features to construct a new feature vector. The combined features are then fed into the stacked sparse auto-encoder for training, with the training labels representing the perceptual similarity between natural texture images obtained through free-grouping experiments. We utilize different computational features as inputs to the stacked sparse auto-encoder, aiming to learn the similarity between hand-drawn sketches and natural texture images.

The stacked sparse auto-encoder consists of six layers, including an input layer, an output layer, and four hidden layers. The number of neurons in these layers is 192, 192, 96, 48, 48, and 1, respectively. The neurons in the first layer correspond to the paired sketch and texture features, so the number of neurons in the input layer changes depending on the input features. The neurons in the last layer correspond to the similarity values of the texture pairs. The paired texture features and sketch features are sent to the network for training, and the distance between each pair of images in the perceptual space can be obtained by mapping the features to the sketch-texture perceptual space. Given a query texture, extract the features of this hand-drawn texture and combine them with the features of all the texture images in the database. Then feed the combined features into the pre-trained stacked sparse autoencoder to compute the similarity between the hand-drawn texture and the natural texture images. We used Euclidean distance as

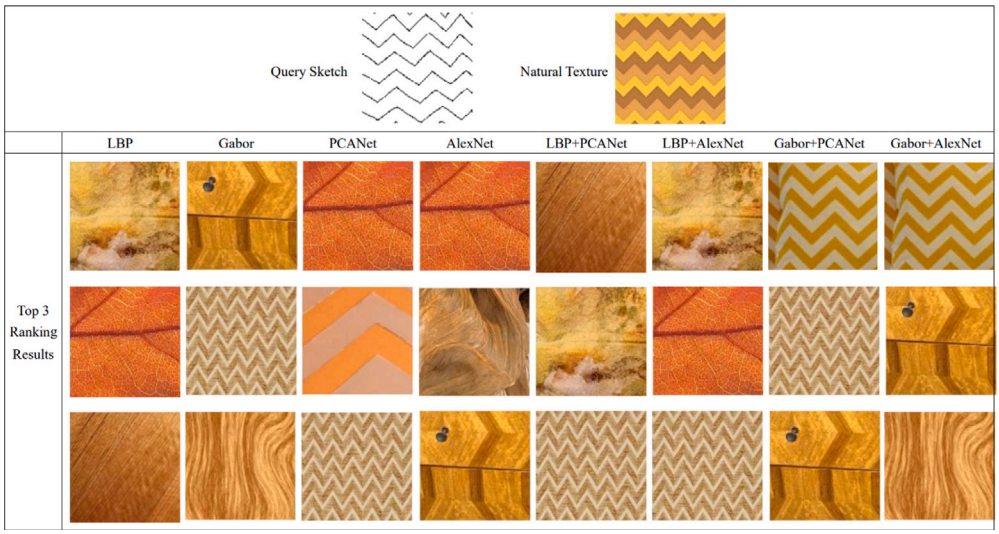


Fig. 7. Retrieval results based on distance measure with zig-zag sketch.

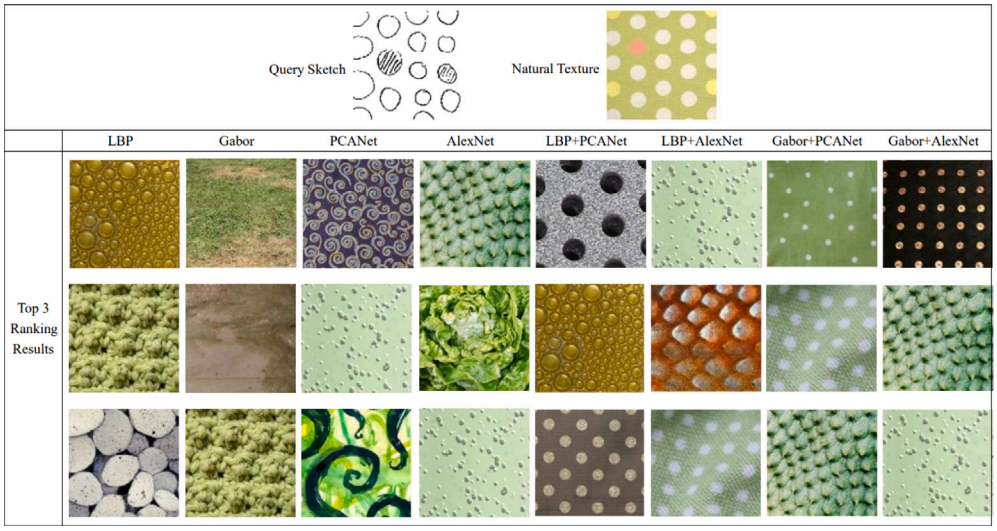


Fig. 8. Retrieval results based on distance measure with bubble sketch.

the measurement standard for distance calculation. Euclidean distance is defined as follows:

$$D(x, y) = \sqrt{\sum (x_i - y_i)^2}, i = 1, 2, 3 \ldots, n \tag{5}$$

Where x and y represents the features of the input sketch and texture respectively, $D(x, y)$ is the distance between vector x and vector y, and the dimension of the feature is n. As the distance between the two features decreases, the similarity between two textures gradually increases.

Taking into account the different attributes of the manual design features and the deep features, we combined them and fed them into an auto-encoder for encoding, and then reconstructed the input features. The experimental configuration used was $lamda(1e{-}11)$ and $beta(0.001)$, and the sparsity of each layer was set to 0.1. By reducing the dimension of the input features, we obtained a low-dimensional feature that had the same representation as the original feature. We tested different feature combinations and assessed the strength of different dimensions using correlation analysis. The experimental results are presented in Table 2.

As shown in the table above, the combined features encoded by the auto-encoder exhibit stronger expressive ability, which enables them to better represent natural textures and hand-drawn sketches. Among these features, the combination of Gabor and PCANet features achieves the best experimental results.

Table 2
Correlation coefficient with feature combination.

FEATURE	LBP+PCA	LBP+Alex	Gabor+PCA	Gabor+Alex
D = 1024	0.2778	0.3523	**0.5668**	0.5282
D = 512	0.2816	0.3387	**0.5703**	**0.5608**
D = 128	0.2723	0.3225	0.5290	0.5426
D = 48	0.2479	0.3284	0.5027	**0.5646**

4.3. Layer-wise sketch-texture retrieval method

Sketches and natural textures differ significantly from one another, and a simple distance measurement parameter is insufficient to retrieve the texture described by the sketch. Sketch retrieval based solely on distance measurement is not applicable to natural textures, representing a weak constraint in establishing a relationship between sketches and natural textures. In order to obtain effective retrieval results, we propose a layer-wise sketch-texture retrieval framework based on perceptual similarity. Pairs of hand-drawn sketches and texture images

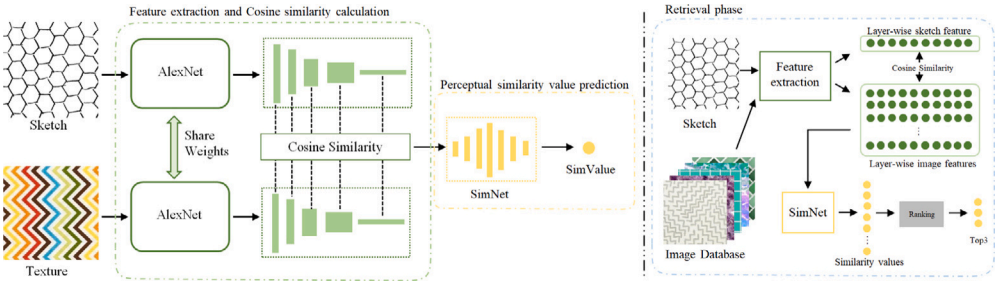


Fig. 9. The layer-wise sketch-texture retrieval framework integrating perceptual similarity. The predicted perceptual similarity is obtained by layer-wise perceptual prediction network. Sort according to the prediction results, and the top three outputs are the retrieval results.

are fed into dual-channel convolutional neural networks for training to learn the layer-wise perceptual similarity. The predicted perceptual similarity values are sorted, and the top three images are output as the retrieval results of hand-drawn sketches. This framework enables efficient retrieval based on the perceptual similarity between texture images.

In the sketch-texture perceptual space, the psychophysical similarity values between any two textures range from 0 to 1. A similarity value close to 1 indicates that the two textures are very similar, while a similarity value close to 0 suggests that the two textures are significantly different from each other. By leveraging psychophysical similarity values and features extracted from samples using different methods, we can train a similarity prediction model based on similarity prediction networks.

Inspired by the layer-wise perceptual similarity calculation method in the task of texture similarity prediction [48], we propose the hand-drawn sketch-texture layer-wise perceptual similarity prediction method, as depicted in Fig. 8. The proposed method encompasses three main components: feature extraction, cosine similarity calculation, and prediction of perceptual similarity values.

Feature extraction. We use AlexNet for feature extraction. Each input to the network consists of paired hand-drawn sketches and texture images. As AlexNet is trained on color images, we expand the original hand-drawn sketches into RGB images and input them into the AlexNet network. We perform standard forward propagation on each input and choose the features from the convolutional layers of the first five layers for similarity calculation. It should be noted that the network parameters are fixed when different channels of the network are used for feature extraction.

Cosine similarity calculation. After the feature extraction stage, we obtain pairs of convolutional features and compute the cosine similarity between each pair of features to obtain inputs for the subsequent similarity prediction network. When computing the cosine similarity, for each spatial position in the feature maps of the lth convolutional layer, there exists a vector of length equal to the number of channels in the lth layer. For each pair of convolutional features at the same spatial position, we compute the cosine similarity between the two vectors. We then calculate the final similarity value in the lth layer by averaging the similarity values across spatial positions. As AlexNet has five convolutional layers, the constructed similarity vectors have five dimensions, with each dimension representing the cosine similarity calculated in the feature space derived from a specific convolutional layer.

Perceptual similarity value prediction. Deep convolutional neural networks can generate perceptually more realistic results. When using deep convolutional neural networks for image feature extraction, low-level layers can capture fine texton information, while high-level layers capture global statistical information. Therefore, each convolutional layer of the network reflects the similarity between paired hand-drawn sketches and texture images at different scales. To achieve this transformation, we implement a fully-connected network, named SimNet, to compute the perceptual similarity between paired hand-drawn sketches and texture images. The network structure of SimNet is defined as $\{5, 16, 64, 128, 64, 16, 1\}$, where each number represents the number of neurons used in each layer. SimNet has a total of 7 layers, including the input and output layers. The activation function used in SimNet is ReLU. The training objective is to minimize the Euclidean distance between the predicted similarity values and the ground truth.

The input hand-drawn sketch and texture image are fed into two twin networks for feature extraction. For each layer in the feature extraction process, we calculate the cosine similarity value between the hand-drawn sketch feature map and the natural texture image feature map at the same spatial position. The cosine similarity value of each layer is then sent to the perceptual similarity prediction network to predict the perceptual similarity between the hand-drawn sketches and texture images. The training objective is to minimize the distance between the predicted values and the perceptual similarity values obtained from psychophysical experiments (see Fig. 9).

4.4. Experimental results

In the layer-wise sketch-texture retrieval method, we use the perceptual similarity prediction network to directly predict the perceptual similarity between hand-drawn sketches and natural texture images. During the training process, pairs of hand-drawn sketches and natural texture images are fed into the similarity prediction network, with the perceptual similarity values obtained from psychophysical experiments used as labels for similarity prediction. For training, we randomly selected 2,820 images $(60images/category \times 47categories)$ and 1,000 sketches as the training data, while the remaining 2,820 images $(60images/category \times 47categories)$ and 140 sketches were used as the test data. During the testing phase, we input the hand-drawn sketch and natural texture images in the test set into the trained model to predict the perceptual similarity value between them. The similarity of all texture pairs is then sorted, and the top three texture images with the highest similarity are output as the retrieval results. The experimental results are presented in Fig. 10. The first row displays the input hand-drawn sketch, the second row shows the corresponding texture image, and the third to fifth rows present the top-3 ranking results.

The results demonstrate that the retrieval results obtained from predicted similarity values with models trained by similarity prediction networks conform well to the appearances of the given textures. These findings indicate that the proposed retrieval method can effectively accomplish texture retrieval and the results are consistent with human visual perception.

We also conducted a classification retrieval experiment. After completing the retrieval experiment, we evaluated whether the texture of the output TOP20 and TOP40 belonged to the same category as the query texture. If the top 20 retrieved texture images sorted by similarity all belong to the same category as the input hand-drawn sketch, the accuracy rate of TOP20 is 100%. If none of the top 20 retrieved texture images belong to the same category as the input hand-drawn sketch, the accuracy rate of TOP20 is 0%. The accuracy rates are presented in

Fig. 10. Retrieval results. The first row shows the input hand-drawn sketch, the second row shows the corresponding texture image, and the third to fifth rows show the top-3 ranking results.

Table 3
Classification accuracy.

FEATURE	LBP+PCA	LBP+Alex	Gabor+PCA	Gabor+Alex	Ours
TOP20	0.7443	0.7874	0.8021	**0.8231**	**0.9152**
TOP40	0.6224	0.6642	0.7370	**0.7846**	**0.8597**

Table 4
Comparative results on sketch recognition.

Models	Accuracy (%)
SIFT-SVM [49]	25.53
HOG-SVM [50]	38.29
LeNet [31]	46.80
AlexNet-SVM [47]	53.19
Ours	**59.57**

Table 3. Our experiments demonstrate that our method can effectively classify and retrieve hand-drawn sketches, with the best classification results obtained by using the layer-wise perceptual similarity prediction network.

We further compared our perceptual sketch-texture retrieval model with a variety of alternatives on the DTD-S Dataset for sketch recognition. These methods include some traditional hand-crafted feature methods: SIFT-SVM [49] and HOG-SVM [50], as well as some deep learning-based methods: LeNet [31] and AlexNet-SVM [47]. The experimental results, as shown in Table 4, demonstrate that our proposed method outperforms the other methods on the hand-drawn texture dataset.

5. Conclusion

In this paper, we propose a novel texture retrieval method based on perceptual texture similarity prediction using layer-wise similarity prediction networks. The proposed method effectively predicts perceptual similarity between hand-drawn sketches and natural texture images, and the retrieval results are consistent with human visual perception. However, it should be noted that although the similarity prediction networks achieve better results, the time required for model training remains a main bottleneck that needs to be improved.

Furthermore, several retrieval tests did not yield expected findings due to a lack of critical information in the hand-drawn designs. Additionally, the number of textures carrying psychological data is limited. To improve retrieval results, more textures with similar properties should be included in the model training phase.

CRediT authorship contribution statement

Yan Liu: Validation, Investigation, Data curation. **Ying Gao:** Writing – original draft, Validation, Methodology. **Nawaz Hafiza Sadia:** Writing – review & editing. **Lin Qi:** Writing – review & editing, Methodology. **Junyu Dong:** Writing – review & editing, Project administration, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The work of J. Dong was supported by National Key R and D Program of China (Grant No. 2018AAA0100602), National Natural Science Foundation of China (Grant No. U1706218) and the Natural Science Foundation of Shandong Province, China (Grant No. ZR2018ZB0852). The work of L. Qi was supported by National Natural Science Foundation of China (Grant No. 61501417).

References

[1] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, 20-26 June 2005, San Diego, CA, USA, IEEE Computer Society, 2005, pp. 886–893.
[2] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[3] R. Hu, J.P. Collomosse, A performance evaluation of gradient field HOG descriptor for sketch based image retrieval, Comput. Vis. Image Underst. 117 (7) (2013) 790–806.
[4] M. Eitz, J. Hays, M. Alexa, How do humans sketch objects? ACM Trans. Graph. 31 (4) (2012) 44:1–44:10.
[5] R.K. Sarvadevabhatla, R.V. Babu, Freehand sketch recognition using deep features, 2015, CoRR abs/1502.00254.
[6] Y. Yang, T.M. Hospedales, Deep neural networks for sketch recognition, 2015, CoRR abs/1501.07873.
[7] O. Seddati, S. Dupont, S. Mahmoudi, DeepSketch: Deep convolutional neural networks for sketch recognition and similarity search, in: 13th International Workshop on Content-Based Multimedia Indexing, CBMI 2015, Prague, Czech Republic, June 10-12, 2015, IEEE, 2015, pp. 1–6.
[8] X. Cai, Sketch-Based Procedural Texture Retrieval, Ocean University of China, 2016.
[9] J. Dong, L. Wang, J. Liu, Y. Gao, L. Qi, X. Sun, A procedural texture generation framework based on semantic descriptions, Knowl.-Based Syst. 163 (2019) 898–906.
[10] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014, IEEE Computer Society, 2014, pp. 3606–3613.
[11] K. Hirata, T. Kato, Query by visual example - content based image retrieval, in: A. Pirotte, C. Delobel, G. Gottlob (Eds.), Advances in Database Technology - EDBT 92, 3rd International Conference on Extending Database Technology, Vienna, Austria, March 23-27, 1992, Proceedings, in: Lecture Notes in Computer Science, vol. 580, Springer, 1992, pp. 56–71.
[12] D. Lopresti, A. Tomkins, Temporal domain matching of hand-drawn pictorial queries, in: Proc. of the Seventh Conf. of the Intl. Graphonomics Society, Citeseer, 1995, pp. 98–99.
[13] A.D. Bimbo, P. Pala, Visual image retrieval by elastic matching of user sketches, IEEE Trans. Pattern Anal. Mach. Intell. 19 (2) (1997) 121–132.
[14] S. Sclaroff, Deformable prototypes for encoding shape categories in image databases, Pattern Recognit. 30 (4) (1997) 627–641.
[15] S. Dupont, O. Seddati, S. Mahmoudi, DeepSketch 2: Deep convolutional neural networks for partial sketch recognition, in: 14th International Workshop on Content-Based Multimedia Indexing, CBMI 2016, Bucharest, Romania, June 15-17, 2016, IEEE, 2016, pp. 1–6.
[16] O. Seddati, S. Dupont, S. Mahmoudi, DeepSketch 3 - analyzing deep neural networks features for better sketch recognition and sketch-based image retrieval, Multim. Tools Appl. 76 (21) (2017) 22333–22359.
[17] B. Liu, J. Gan, B. Wen, Y. LiuFu, W. Gao, An automatic coloring method for ethnic costume sketches based on generative adversarial networks, Appl. Soft Comput. 98 (2021) 106786.
[18] M. Mirmehdi, Handbook of Texture Analysis, Imperial College Press, 2008.
[19] M. Tüceryan, A.K. Jain, Texture analysis, in: C.H. Chen, L.F. Pau, P.S.P. Wang (Eds.), Handbook of Pattern Recognition and Computer Vision, World Scientific, 1993, pp. 235–276.
[20] R.M. Haralick, K.S. Shanmugam, I. Dinstein, Textural features for image classification, IEEE Trans. Syst. Man Cybern. 3 (6) (1973) 610–621.
[21] L.G. Roberts, Machine Perception of Three-Dimensional Solids, in: Outstanding Dissertations in the Computer Sciences, Garland Publishing, New York, 1963.
[22] J.F. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6) (1986) 679–698.
[23] D. Marr, E. Hildreth, Theory of edge detection, Proc. R. Soc. Lond. Ser. B Biol. Sci. 207 (1167) (1980) 187–217.
[24] F. Ade, Characterization of textures by eigenfilters, Signal Process. 5 (5) (1983) 451–457.
[25] I. Fogel, D. Sagi, Gabor filters as texture discriminator, Biol. Cybernet. 61 (2) (1989) 103–113.
[26] J. Chen, A. Kundu, Rotation and gray scale transform invariant texture identification using wavelet decomposition and hidden Markov model, IEEE Trans. Pattern Anal. Mach. Intell. 16 (2) (1994) 208–214.
[27] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[28] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: P.L. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a Meeting Held December 3-6, 2012, Lake Tahoe, Nevada, United States, 2012, pp. 1106–1114.
[29] K. Kavukcuoglu, P. Sermanet, Y. Boureau, K. Gregor, M. Mathieu, Y. LeCun, Learning convolutional feature hierarchies for visual recognition, in: J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, A. Culotta (Eds.), Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a Meeting Held 6-9 December 2010, Vancouver, British Columbia, Canada, Curran Associates, Inc., 2010, pp. 1090–1098.
[30] Q. Zhao, J. Dong, H. Yu, S. Chen, Distilling ordinal relation and dark knowledge for facial age estimation, IEEE Trans. Neural Networks Learn. Syst. 32 (7) (2021) 3108–3121.
[31] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[32] L. Sifre, S. Mallat, Rotation, scaling and deformation invariant scattering for texture discrimination, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Portland, OR, USA, June 23-28, 2013, 2013, pp. 1233–1240.
[33] G.E. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[34] Y. Bengio, A.C. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
[35] X. Dong, Perceptual Texture Similarity Estimation (Ph.D. thesis), Heriot-Watt University, Edinburgh, UK, 2014.
[36] M. Eitz, K. Hildebrand, T. Boubekeur, M. Alexa, Sketch-based image retrieval: Benchmark and bag-of-features descriptors, IEEE Trans. Vis. Comput. Graph. 17 (11) (2011) 1624–1636.
[37] X. Wang, J. Chen, H. Yang, A new integrated SVM classifiers for relevance feedback content-based image retrieval using EM parameter estimation, Appl. Soft Comput. 11 (2) (2011) 2787–2804.
[38] H. Qiang, Y. Wan, Z. Liu, L. Xiang, X. Meng, Discriminative deep asymmetric supervised hashing for cross-modal retrieval, Knowl.-Based Syst. 204 (2020) 106188.
[39] X. Dong, H. Zhang, X. Dong, X. Lu, Iterative graph attention memory network for cross-modal retrieval, Knowl.-Based Syst. 226 (6) (2021) 107138.
[40] J. Lei, Y. Song, B. Peng, Z. Ma, L. Shao, Y.-Z. Song, Semi-heterogeneous three-way joint embedding network for sketch-based image retrieval, IEEE Trans. Circuits Syst. Video Technol. 30 (9) (2019) 3226–3237.
[41] P. Xu, Y. Huang, T. Yuan, K. Pang, Y.-Z. Song, T. Xiang, T.M. Hospedales, Z. Ma, J. Guo, Sketchmate: Deep hashing for million-scale human sketch retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8090–8098.
[42] M. Wang, W. Zhou, Q. Tian, H. Li, Deep graph convolutional quantization networks for image retrieval, IEEE Trans. Multimed. (2022).
[43] Y. Liu, D. Dai, X. Tang, S. Xia, G. Wang, Bi-lstm sequence modeling for on-the-fly fine-grained sketch-based image retrieval, IEEE Trans. Artif. Intell. (2022).
[44] B. Cao, N. Wang, J. Li, Q. Hu, X. Gao, Face photo-sketch synthesis via full-scale identity supervision, Pattern Recognit. 124 (2022) 108446.
[45] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.
[46] T. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, Y. Ma, Pcanet: A simple deep learning baseline for image classification? IEEE Trans. Image Process. 24 (12) (2015) 5017–5032.
[47] Q. Dong, H. Wang, Z. Hu, Commentary: Using goal-driven deep learning models to understand sensory cortex, Front. Comput. Neurosci. 12 (2018) 4.
[48] Y. Gao, Y. Gan, L. Qi, H. Zhou, X. Dong, J. Dong, A perception-inspired deep learning framework for predicting perceptual texture similarity, IEEE Trans. Circuits Syst. Video Technol. 30 (10) (2020) 3714–3726.
[49] A. Oliva, A. Torralba, Building the gist of a scene: The role of global image features in recognition, Progress Brain Res. 155 (2006) 23–36.
[50] M. Eitz, J. Hays, M. Alexa, How do humans sketch objects? ACM Trans. Graph. (TOG) 31 (4) (2012) 1–10.