A Perception-Inspired Deep Learning Framework for Predicting Perceptual Texture Similarity

Ying Gao, Yanhai Gan, Lin Qi, Huiyu Zhou, Xinghui Dong, and Junyu Dong

***Abstract*— Similarity learning plays a fundamental role in the fields of multimedia retrieval and pattern recognition. Prediction of perceptual similarity is a challenging task as in most cases we lack human labeled ground-truth data and robust models to mimic human visual perception. Although in the literature, some studies have been dedicated to similarity learning, they mainly focus on the evaluation of whether or not two images are similar, rather than prediction of perceptual similarity which is consistent with human perception. Inspired by the human visual perception mechanism, we here propose a novel framework in order to predict perceptual similarity between two texture images. Our proposed framework is built on the top of Convolutional Neural Networks (CNNs). The proposed framework considers both powerful features and perceptual characteristics of contours extracted from the images. The similarity value is computed by aggregating resemblances between the corresponding convolutional layer activations of the two texture maps. Experimental results show that the predicted similarity values are consistent with the human-perceived similarity data.**

***Index Terms*— Similarity learning, perceptual similarity, texture similarity, convolutional neural networks.**

I. Introduction

As a widely studied visual element in computer vision, computer graphics and pattern recognition, texture can be found everywhere in the real world. Since texture provides a wealth and depth of visual information, such as coarseness, directionality and roughness, it becomes one of the most common vision cues in the human visual system (HVS). Similarity is the measurement of the likeness of two samples. It has been widely used in texture and material recognition [1], [2], semantic segmentation [3], aerial imagery classification [4], and person re-identification [5], and play important roles in object recognition and scene understanding. Accurate prediction of texture similarity can benefit many visual tasks, such as texture recognition and surface defect detection. When Unmanned Aerial Vehicles (UAV) are used to identify the ground, for example, the ground can be effectively analyzed based on the similarity between the images captured by the drone as they exhibit rich texture characteristics. Another example is the identification of objects with different materials, such as clothes, metals and plastics, by humans who are able to determine the category of different images by analyzing the similarity between two materials.

Although texture images contain rich visual characteristics, they are difficult to be described due to inadequate semantic information. In the vision science community, researchers have managed to reveal the mystery of the human visual mechanism for texture perception [6], [7]. Among these studies, texture similarity perception has received much attention, which can be used to evaluate the performance of automated systems for texture analysis. Research in texture similarity perception can be traced back to 1960s, when Julesz [8] conducted visual discrimination experiments using unfamiliar stimuli generated by a digital computer. In this experiment, he aimed to study how the subjects observe and understand the visual attributes of two images connected side by side. This was followed by the texture similarity studies using psychophysical experiments performed by human observers, including free-grouping, perceptual feature scoring and pair-wise comparison [9]–[11]. Using these experiments, human perceptual texture similarity data can be collected and further analyzed in order to discover the visual mechanism behind the human perception of texture similarity. Nevertheless, psychophysical experiments are time-consuming and are also expensive to conduct because their special requisition on subjects, experimental setup and environments and procedural issues. Alternatively, predicting texture similarity by accurately mimicking human perceptual similarity judgements can be a solution to the problem.

Perceptual texture properties have been used for estimating texture similarity [12]. However, Dong *et al.* demonstrated that there is no simple relationship between the perceptual attributes and the computational features of texture images [13]. Although computational features perform well in

Manuscript received June 12, 2019; revised September 10, 2019; accepted September 25, 2019. Date of publication September 30, 2019; date of current version October 2, 2020. The work of J. Dong was supported in part by the National Key Research and development Program of China under Grant 2018AAA0100602, and in part by the National Natural Science Foundation of China (NSFC) under Grant 41576011, Grant U1706218, and Grant 41927805. The work of H. Zhou was supported in part by the U.K. Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/N011074/1, in part by the Royal Society-Newton Advanced Fellowship under Grant NA160342, and in part by the European Unions Horizon 2020 Research and Innovation Program through the Marie-Sklodowska-Curie under Grant 720325. The work of L. Qi was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61501417. This article was recommended by Associate Editor H. Lu. *(Corresponding author: Junyu Dong.)*

Y. Gao, Y. Gan, L. Qi, and J. Dong are with the Department of Information Science and Technology, Ocean University of China, Qingdao 266100, China (e-mail: dongjunyu@ouc.edu.cn).

H. Zhou is with the Department of Informatics, University of Leicester, Leicester LE1 7RH, U.K. (e-mail: hz143@leicester.ac.uk).

X. Dong is with the Centre for Imaging Sciences, The University of Manchester, Manchester M13 9PT, U.K. (e-mail: dongxinghui@gmail.com).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2019.2944569

many tasks, such as texture classification [14] and semantic segmentation [15], they still produced large deviation when used to estimate perceptual texture similarity [16]. On the other hand, researchers attempted to predict texture similarity by constructing a perceptual texture space using manifold learning [12], [17], [18]. Based on the perceptual data obtained from psychophysical experiments, a texture perceptual space can be constructed by subspace transformation algorithms. Then the computational features extracted from the textures are mapped to this space. It has been shown that the perceptual space can effectively describe the similarity between textures. However, the procedure of constructing the subspaces is cumbersome. When new samples are encountered, the effectiveness of the texture perceptual space needs to be further verified.

As a metric learning technique, similarity learning is important to the research of multimedia retrieval and pattern recognition. In particular, it has been widely applied to ranking, visual identity tracking, face verification and speaker verification [19]–[22]. Nevertheless, these studies normally concern the similarity or matching of two image patches rather than predicting the similarity of the images which is consistent with human labeled ground-truth. In contrast, fine-grained texture similarity has received less attention although this data is able to provide more precise likeness estimation. The possible reason attributes to the lack of human-derived fine-grained texture similarity data.

One of the exceptions is the study conducted by Dong and Chantler [23] who examined the ability of 51 different texture feature sets for estimating fine-grained human perceptual texture similarity. It was found that none of these feature sets produced comparable results to those of humans. Dong and Chantler [24] further introduced a set of perceptually motivated image features by exploiting contour cues. Nevertheless, this feature set is not good at encoding textures containing small structures.

To our knowledge, very few of the existing studies exploit similarity learning for estimating fine-grained texture similarity. To address this challenging problem, we are inspired to explore both the human visual mechanism on texture similarity perception and recent advances on Convolutional Neural Networks (CNN). In [25], Gatys *et al.* used Gram matrices computed from a pre-trained CNN for texture synthesis. Given a CNN, they combined the conceptual framework of spatial summary statistics on the feature responses with the powerful features generated by the CNN. The experiments indicated that the established CNNs may be used to generate suitable stimuli for perceptual or physiological studies on texture representation [26]. Zhang *et al.* demonstrated the effectiveness of deep features as the perceptual metric for determining the similarity of two pairs of image patches [27]. In addition, it has been highlighted in the literature that contour cues are important to the human visual system [28], [29]. In [24], Dong and Chantler observed that contour maps provide better texture representation than other types of local texture characteristics. They attributed this success to the long-range interactions between local image characteristics encoded by contours [23], [30].

Motivated by the aforementioned studies, we here introduce a new fine-grained texture similarity estimation approach, which mimics human visual perception for texture similarity. This approach not only explores the advantages of CNNs but also benefits from contour cues which are able to encode long-range interaction. First, contour maps of the textures are extracted because of its importance to texture representation [24], [30]. Second, the paired texture images and the corresponding contour maps are used to train a similarity learning network. Third, the network is used to predict the similarity between a pair of texture images. The network contains three parts: deep feature extraction, layer-wise similarity calculation and perceptual similarity prediction. In particular, we use a commonly used deep architecture for feature extraction. The features extracted at each convolutional layer are used to compute one layer-wise similarity. Using all the layer-wise similarity values, we create a fully-connected network, namely, the similarity network. The fully-connected layers are able to automatically assign appropriate weights to different layers.

In this paper, we propose a new method to predict the fine-grained perceptual similarity between two texture images. Both contour maps together with their corresponding original input textures are used inspired by visual perception studies, and our experiments show that contour information is indeed beneficial for estimating fine-grained perceptual texture similarity, which is consistent with that of human observers [24]. The proposed deep learning framework also shows that the layer-wise similarity between two CNNs is able to achieve better performances than higher level features (e.g. FC layer). The proposed network can be easily transferred to the estimation of the similarity between natural images by exploiting a publicly available dataset, and it also achieve good results in texture retrieval experiments.

The rest of this paper is organized as follows. In Section II, we review the related work in the area of acquiring perceptual descriptions and estimating perceptual similarity using computational features. We introduce the proposed fine-grained perceptual texture similarity prediction method and implementation details in Section III. In Section IV, texture similarity prediction experiments are reported and the generalization of our method to natural image similarity prediction is investigated. The effect of different loss functions, the use of contour information, the fusion of perceptual information and retrieval-based evaluation experiments are further analyzed in Section V. Finally, in Section VI, we give conclusions and discuss the future work.

II. Related Work

A. Acquiring Perceptual Descriptions

As a branch of psychology, Psychophysics [31] studies human quantification of physical stimuli and the sensations caused by the stimuli. Psychophysicists believe that all human senses, including sight, hearing, etc., can be described by the relationship between sensory and stimulus intensities. Accordingly, texture perception is the study of stimuli and effects of different texture images on human vision, that is, how the human visual system describes and perceives texture images. Texture perception is of great importance in scene understanding and data visualization. Research in this area was initiated by Julesz [8], who displayed visual images randomly on digital computers to conduct visual discrimination experiments whilst studying how the subjects observe and understand two textures with the same visual attributes.

Along with continuous research on texture perception, many natural texture databases including Brodatz [32], OuTex [33], and CUReT [34] have been established. Researchers began to implement visual perception mechanisms for specific texture images. Tamura *et al.* [9] designed a psychophysical experiment based on the Brodatz texture dataset and asked subjects for rating texture images using six different perceptual attributes. Amadasun and King [35] also performed a similar experiment on this database, asking subjects to rate the texture attributes, and then defining an approximate calculation form for each texture attribute and studying the visual perception of humans according to the results of texture similarity measurement. This research had achieved certain results, but it was not as good as the desired because human visual perception is a very complicated process. In subsequent studies [36]–[40], researchers used many similar methods to quantify the perceptual features of textures, expecting to find texture features that are consistent with human perception. Due to different interpretations and quantitative criteria, there are certain deviations when a texture image is perceptually rated. Accordingly, it is difficult to comprehensively and reasonably analyze the experimental results of each subject. Consequently, these experimental results were not able to provide texture similarity values, which is the main objective of our study.

One way to obtain perceptual similarity is through a free-grouping experiment [11], which was first performed on the Brodatz database. The experiment required the subjects to group 56 images stored in the database according to their own understanding without any hint. The number of the groups is not limited, and the number of the images allocated in each group is not limit either. Images that are grouped by the majority of people in the same group are considered to have a high degree of similarity, while images that are not grouped together are treated as no similarity. Through the analysis of the grouping results of all subjects participating in the experiment, the similarity values between two images can be obtained and a 56 × 56 similarity matrix is constructed. Although the free-grouping experiments can obtain very reliable texture perceptual similarity values, the experiment takes a long time and cannot be performed on a large number of samples. Subsequently, the small number of human labeled similarity values provided in [11] limited further analysis or accurate prediction of perceptual texture similarity. Later, in [41], Halley *et al.* conducted a pairwise comparison experiment on the Pertex dataset containing 334 texture samples. The experiment was undertaken by thirty subjects. However, due to the insufficient number of experiments, the result similarity matrix contains many zero values and has a sparse matrix structure. Based on these experiments, Clark *et al.* [42] further proposed an equidistance mapping [43] algorithm to obtain a full similarity matrix. Liu [12] further performed free-grouping and merging experiments in the Procedural Texture Dataset (PTD). Twenty subjects were required to group 450 texture images according to their own understanding. After the first grouping had been completed, the subjects were asked to merge well-grouped textures and give each combination a confidence figure. The group-merging procedure was performed iteratively until the subjects felt that the remaining groups could not be merged anymore. By multiplying each grouping result with the confidence of the subjects and accumulating the results to obtain perceptual similarity values, the experiment successfully avoids the problem of sparsity in the similarity matrix as occurred in [41]. In material recognition studies, [2] derives a framework to discover locally-recognizable material attributes automatically. The authors measure perceptual distances between materials and defined an attribute space based on perceptual distances. However, its goal is to improve the material recognition accuracy and they mainly use for material dataset, which is different from our texture. In contrast, our work focuses on perceptual similarity prediction directly from two input texture images.

B. Estimating Perceptual Similarity Using Computational Features

Texture feature extraction is the most critical procedure in many visual tasks such as semantic segmentation, texture classification and retrieval. Whether the extracted features are good or not directly affects the accuracy of subsequent tasks. Researchers tried to find a characteristic pattern that can best represent a texture sample [44]–[46]. To achieve better results, hand-crafted features need to be further adjusted and different parameter settings have to be tested. Researchers hope to find a robust texture representation, which can make the extracted features be simultaneously applied to multiple texture classification tasks. In recent years, it is well known that deep learning based features demonstrated state-of-the-art performance. Generally, deep features refer to those extractions through a multi-layer convolutional neural network and can be treated as the discriminative representation of the sample data. Deep learning methods attempt to establish and simulate the human brain for analysis and learning to mimick the behavior of neurons in the human brain and to analyze and understand various data including images, sounds, and texts [47]–[50]. Reference [51] proposes a deep texture encoding network, the representation obtained in the network is particularly useful for material and texture recognition and produced state-of-the-art performance. However, the goal is different from ours, as we focus on perceptual similarity prediction directly from two input texture images. It is essentially a regression task, whereas the Deep-TEN focus on texture encoding.

Although there are little work on the estimation of perceptual texture similarity, recent advances based on convolutional neural networks for approximating natural image similarity produced promising results. Zagoruyko [52] proposed a general similarity function based on CNN and decision networks to compare image patches directly from image data to determine if the two patches match, i.e. the same scene region under different views. Han *et al.* [53] proposed a unified approach for combining feature and metric learning for patch-based matching, i.e. to determine whether the two patches are similar to each other. Zhang *et al.* [27] proposed a method to compute the distance between two patches, and then train a small network to predict perceptual ranking from two distance pairs. Their work shows the effectiveness of deep features as a perceptual metric for evaluating whether or not each of the two patches is similar to the original one. He and Sclaroff [54] proposed a learned local feature descriptors which achieved good results in ranking and matching experiments. Mots of these methods mentioned above focus on patch similarity evaluation, i.e. to determine whether one sample is similar to the other, whereas our method concentrates on determining the degree of similarity between textures. The proposed framework can output a quantative figure that represents the degree of similarity between two texture samples.

The most related work to ours is by Lou *et al.* [16], in which Random Forests were employed to regress the similarity data and produced promising results. In their work, after computational features have been extracted from the sample textures, the paired texture features were combined together and then sent to the random forests classifier for the prediction of fine-grained similarity values. In [55], the combined features obtained by the same feature processing method are sent to the trained auto-encoder, which obtained accurate prediction results. In this situation, a large number of feature sets need to be tested so that random forests can produce better results. In contrast, the method proposed in this paper can automatically learn effective features for accurately predicting perceptual texture similarity, as our deep model is an end-to-end architecture and can be easily trained. Our method is more robust and has better transferability in the task of predicting the similarity of natural image patches.

III. Method

In deep learning algorithms, the inputs and targets of the models are vectorized, forming the input vector space and target vector space. Deep learning usually uses a multi-layered network structure to compute a multi-layered abstract representation of the learned data [56]. Each layer in the deep learning model consists of a number of simple but non-linear modules, and undertakes a simple geometric transformation for the data passing through it. Each module is responsible for transforming the input of the layer into a higher-level, more abstract representation and outputting it to the next layer. After the original data is input into the network, it is processed layer by layer, continuously transformed, and output as a higher-level abstract feature. The deep features learned by convolutional neural networks show comparable or even stronger performance in many computer vision tasks than those extracted by traditional methods [49], [50], for example, in image classification tasks, higher-level features can highlight the useful information of the original input that play an important role in the final classification discrimination, while suppressing insignificant differences between the data. However, in similarity learning, the distance metric in the computational feature space cannot be well correlated with the perceptual similarity space [13]. Therefore, we propose an image-wise layer-wise similarity learning, which treats each network layer as a feature space transformation, and calculates multiple feature spaces in the process of mapping the input vector space to the target vector space. The similarity between the two images is obtained by considering the similarities between each feature spaces of the corresponding layers.

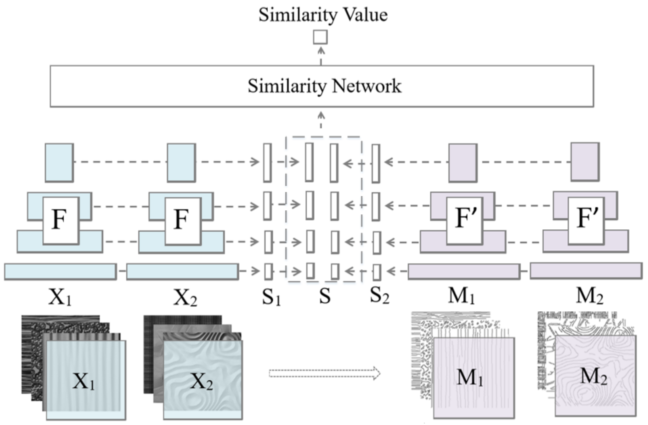

Fig. 1. The proposed framework for perceptual texture similarity learning. The framework contains feature extraction, layer-wise similarity calculation and perceptual similarity prediction. The input of the framework consists of paired texture images and corresponding paired contour maps, and the output is the similarity value between the paired images.

The proposed framework is shown in Fig. 1 and consists of three components: deep feature extraction via a convolutional neural network, layer-wise similarity calculation and perceptual similarity value prediction. For clarity, we use X_1 and X_2 to denote two texture images, and use y to denote their perceptual similarity values. As we aim to predict y as accurately as possible, we use MSE as the metric to evaluate our prediction, which is defined as:

$$\delta(D_v) = \frac{1}{n}\sum_{y\in D_v}(y-\hat{y})^2 \tag{1}$$

Here, we use D_v to denote the validation set, $\hat{y}$ to denote our predicted similarity value, and n to denote the number of valid similarity values in the validation set. As $\hat{y}$ is our predicted value, which can be written as:

$$\hat{y} = H(X_1, X_2) \tag{2}$$

in which, H denotes the prediction function. If we resolve H, the formulation can be rewritten as:

$$\begin{aligned}\hat{y} = {} & P(sim(<\Phi(X_1), \Phi(X_2)>),\\ & sim(<\Phi(contour(X_1)), \Phi(contour(X_2))>))\end{aligned} \tag{3}$$

In the formulation, P denotes the function in charge of converting the calculated similarity values to the final prediction result. *sim* denotes the similarity calculation procedure, which calculates similarity values in feature space set, Φ denotes a collection of feature spaces obtained by mapping of input X_1 and X_2, and *contour* denotes the hard wired method to obtain the contour maps of the original texture images.

To give the formulation for clarity, the details of the preprocessing, feature extraction, cosine similarity calculation and perceptual similarity predicting stage will be discussed in the following sections. In addition, we use quadratic loss and cross entropy loss as objective functions in our experiments. The quadratic loss is defined as:

$$L_q = \frac{1}{2m}\sum_{y\in D_t}(y-\hat{y})^2 = \frac{1}{2m}\sum_{y\in D_t}(y-H(X_1,X_2))^2 \qquad (4)$$

in which, D_t denotes the training set, and m denotes the number of valid similarity values in the training set. The cross entropy loss is defined as:

$$L_c = -\frac{1}{2m}\sum_{y\in D_t}[y\ln(\hat{y})+(1-y)\ln(1-\hat{y})] = -\frac{1}{2m}\sum_{y\in D_t}[y\ln(H(X_1,X_2))+(1-y)\ln(1-H(X_1,X_2)].$$

$$(5)$$

A. Extracting Contour Maps

Previous study on visual perception [23], [24], [30] has shown that edge information plays an important role in the estimation of texture similarity. In order to learn the perceptual similarity between the two texture images, we therefore propose to use the contour maps of the texture images as auxiliary inputs for prediction. This step can be viewed as the incorporation of prior knowledge into the proposed architecture. Thus, we first use the method proposed in [24] to compute a contour map for each texture image. As shown in Fig. 1, X_1 and X_2 represent the original texture images, and M_1 and M_2 represent the contour maps of the original texture images. The process of extracting contour maps can be written as follows:

$$< M_1, M_2 > = contour < X_1, X_2 > \qquad (6)$$

The paired texture images $< X_1, X_2 >$ and the corresponding contour maps $< M_1, M_2 >$ will be send to the network together during the training process.

B. Deep Features Extraction

Recently, researchers have been devoting themselves to developing variants of deep learning methods for complex image tasks, including conventional image classification, regression, multi-modal processing, semi-supervised learning, texture synthesis, and even image generation [57]–[59]. Theoretically, features extracted from commonly used convolutional networks can be used in the proposed framework for the prediction of perceptual texture similarity. Meanwhile, in [25], Gatys *et al.* designed a texture model for texture synthesis, in which the use of VGG-19 for feature extraction in each convolutional layer indicates that the statistical structure of natural images can be matched at an increasing scale, as the number of layers used for texture generation increases. On the other hand, it is intuitive that low layers of a convolutional network capture finer spatial information of images, while layers in the high hierarchies capture global statistical summary of images. Zhang etc. demonstrate the effectiveness of deep features used as perceptual metric [27]. Their experiments on rank prediction illustrate that the pretrained parameters of a network are essential for feature extraction. In addition, VGG-Net has become the de facto standard for image generation tasks [60].

Therefore, we turn to use VGG-19 for feature extraction in the current study. For each pair of texture samples, we actually obtain four input images, in which two contour maps are computed during the preprocessing procedure. As VGG-19 is pretrained on the ILSVRC2012 dataset, in which each sample is an RGB color image, we extend our original gray texture images and contour maps to RGB images, and then subtract the mean value in each channel, which are then fed to the VGG-19 network. Finally, standard forward propagation is performed for each input. The expressive ability of different features from convolutional layers and pooling layers are tested and the features of the convolutional layers are selected for the similarity calculation. We collect features in each of 16 convolutional layers as the input. It should be noted that the parameters of the network are fixed when the network is used for feature extraction.

As shown in Formulation 3, we denote a paired texture images as $< X_1, X_2 >$, and denote their contour maps as $< M_1, M_2 >$. In each convolutional layer, which corresponds to a feature space, we construct a tensor pair:

$$F^l = < F_1^l, F_2^l > \qquad (7)$$

Here, Fl^1 represents the calculated feature maps in the l_{th} layer for image X_1, and Fl^2 represents the calculated feature maps in the l_{th} layer for image X_2. The feature of l_{th} layer for $< M_1, M_2 >$ can be written as:

$$F'^l = < F'^l_1, F'^l_2 > \qquad (8)$$

All the feature pairs constitute a set:

$$F = < F^1, F^2, \ldots, F^{16} > \qquad (9)$$

For M_1 and M_2, the same set is constructed as:

$$F' = < F'^1, F'^2, \ldots, F'^{16} > \qquad (10)$$

Here F'^l is a tensor pair $< F'^l_1, F'^l_2 >$, in which F'^l_1 represents the calculated feature maps in the l_{th} layer for M_1, and F'^l_2 represents the calculated feature maps in the l_{th} layer for M_2.

C. Cosine Similarity Calculation

After the feature extraction stage, we have two sets F and F' for a paired texture image input. For each element of F, we calculate a cosine similarity value. As F_1^l and F_2^l are the feature maps in the l_{th} layer of the convolutional network, for each spatial position of F_1^l and F_2^l, there exists a vector with the length equaling to the number of the channels in the l_{th} layer. For the same spatial position of F_1^l and F_2^l, we calculate the cosine similarity value of the two vectors. Afterwards, the final similarity value in the l_{th} layer is calculated by averaging the similarity values across the spatial positions. Finally, a similarity vector is derived from F, which is denoted as S_1. The same operation is performed on the set F', correspondingly, and the derived similarity vector is denoted as S_2. The method to calculate similarity in each convolutional layer is inspired by Zhang *et al.* [27] who proposed to calculate distance in each layer and then summary all the distance values to get a final distance metric for a pair of texture images.

Here, we instead calculate similarity values in every layer, and then concatenate them together as a similarity vector. Equation 3 illustrates the procedure to calculate the similarity between the paired tensors stored in $< F_1^l, F_2^l >$. We use F_{imn}^l to denote the vector at spatial position (m, n) of feature maps F_i^l, which is calculated in the l_{th} convolutional layer from texture image X_i, and we use S_1^l to denote the l_{th} element of vector S_1. In other words, S_1^l represents the similarity value calculated in the l_{th} layer from texture pair $< X_1, X_2 >$:

$$\begin{aligned} S_1^l = sim(F_1^l, F_2^l) &= \frac{F_1^l \cdot F_2^l}{\|F_1^l\|\|F_2^l\|} \\ &= \frac{\sum_{m,n=1}^{N} F_{1mn}^l \times F_{2mn}^l}{\sqrt{\sum_{m,n=1}^{N} {F_{1mn}^l}^2} \times \sqrt{\sum_{m,n=1}^{N} {F_{2mn}^l}^2}} \end{aligned} \tag{11}$$

Now, we have described how to construct the similarity vector S_1^l. It is the same procedure to construct S_2^l, except that all the calculations are based on the contour pair $< M_1, M_2 >$ rather than the texture images:

$$\begin{aligned} S_2^l = sim({F'}_1^l, {F'}_2^l) &= \frac{{F'}_1^l \cdot {F'}_2^l}{\|{F'}_1^l\|\|{F'}_2^l\|} \\ &= \frac{\sum_{m,n=1}^{N} {F'}_{1mn}^l \times {F'}_{2mn}^l}{\sqrt{\sum_{m,n=1}^{N} {{F'}_{1mn}^l}^2} \times \sqrt{\sum_{m,n=1}^{N} {{F'}_{2mn}^l}^2}} \end{aligned} \tag{12}$$

As there are 16 convolutional layers in VGG-19, the constructed similarity vector S_1 has 16 dimensions, each of which represents the cosine similarity calculated in the feature space derived from a certain layer:

$$S_1 = < S_1^1, S_1^2, \ldots, S_1^{16} > \tag{13}$$

And S_2 is of the same dimensionality as S_1:

$$S_2 = < S_2^1, S_2^2, \ldots, S_2^{16} > \tag{14}$$

Finally, the two similarity vectors S_1 and S_2 are concatenated as:

$$S =< S_1, S_2 > \tag{15}$$

Thus, S is a 32-dimensional similarity vector, which is the final representation of a pair of texture images.

D. Perceptual Similarity Value predicting

Deep convolutional neural networks can produce results more consistent with human perception than traditional methods in computer vision areas. It seems that low layers in the network tend to detect simple edge information. When we go higher in the network, the neurons tend to learn something more sophisticated, which may be object parts (combination of edges) in the media layers and complicated objects in the top layers. This observation can be explained intuitively that complicated concepts, e.g. dogs, are composed of low-level concepts, such as leg, arm, head, and body. It is natural that the outputs of low-level layers serve as the components of high-level layers. On the other hand, low-level layers own small receptive fields due to the convolutional operation, while layers in the high hierarchies own larger and even global perspectives indeed.

When using deep convolutional neural network for image feature extraction, low-level layers can be used to capture fine texton information, while high-level layers capture global statistical information. This has also been demonstrated in [25] by Gatys etc. Therefore, the calculated similarity in each convolutional layer of the network reflects the similarity of the paired texture images in different scales. However, the perceptual similarity in PTD and Pertex is given as a scalar, which reflects the holistic sense of the subjects about similarity. Thus, we need to come up with a means to convert the calculated similarity vector to a figure consistent with human judgment. Even extracting features by deep convolutional neural networks remains a challenging task, as this is not a simple linear transformation. In addition, the similarity vector S_2 serving as auxiliary information, which has been proven to be effective complementary of S_1 in our experiments, needs to be handled properly.

To achieve this transformation, we implement a fully-connected network, which is shown at the top of Fig.1. We call this network as similarity network, and the prediction of similarity value can be written as:

$$\hat{y} = P(S) \tag{16}$$

The architecture of the similarity network P is depicted as $\{32, 64, 128, 256, 256, 128, 64, 32, 16, 8, 1\}$. Each number here represents the neurons used in each layer, and there are 11 layers totally, including the input and output layers.

It should be noted that we use relu as the activation function in the similarity network, except for the output layer. Because the perceptual similarity values range from 0 to 1, we use the sigmoid function in the last layer. The training objective is to minimise the Euclidean distance between the predicted values and the ground truth. Cross-entropy loss can also be used as the objective function, and we use quadratic loss to achieve better results, which are explained in our ablation experiments.

IV. Experimental Results

In this section, we performed experimental results on three different datasets with the proposed method. The test datasets include Procdeural Texture Dataset (PTD), Pertex Dataset, and Berkeley-Adobe Perceptual Patch Similarity (BAPPS) Dataset. The experimental results are as follows.

A. Results on Procdeural Texture Dataset (PTD)

PTD contains 450 textures of resolution 512 × 512, which are generated by 23 different procedural texture generation models. The texture generation model generates a texture height map with realistic procedural textures, which help us to accurately obtain humans visual perception of textures. Fig. 2 shows four pairs of textures in the dataset with their perceptual similarity values obtained by free-grouping experiments. For the comparison purpose, we also list the similarity values predicted by our methods. Each pair of textures in this dataset has a perceptual similarity value. The fine-grained perceptual

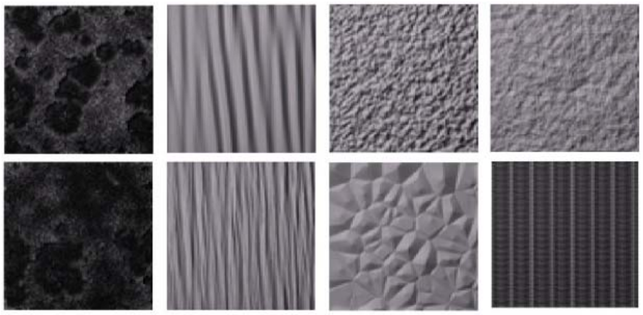

Fig. 2. Paired texture images in PTD. From left to right, the perceptual similarity values of the textures set are 0.9700, 0.8617, 0.3508 and 0.3175, whereas the predicted similarity values are 0.9825, 0.8646, 0.3494 and 0.3180.

similarity values of PTD are in a range of 0 to 1. A value close to 1 means that the two texture images are treated to be similar, whereas a value of 0 represents that the two textures are not similar at all.

As VGG-19 receives the inputs of size 224 × 224, we resize the textures in PTD to the size of 224 × 224 for feature extraction. It should be noted that we only use convolutional layers of VGG-19, so there are actually no resolution limits for the input. However, if we do not resize texture images to 224 × 224, the relative receptive field in each convolutional layer may be changed compared to the original VGG-19. In other words, each neuron in the convolutional layers can only see a smaller ratio of the whole image region. Whereas, the granularity of features is very important for texture similarity perception, and human observers give their opinions about similarity by reviewing texture images from regional textons to global distribution forms for holistic purpose. Therefore, we do resize the high-resolution texture images in PTD to 224 × 224.

PTD contains 450 texture images, and any pair of texture images in the dataset owns a similarity value. The 450 experimental data is randomly divided into training set and testing data with 400 and 50 images respectively. There are totally 80,200 pairs of texture images with fine-grained similarity values in the training set, and 1,275 pairs of texture images in the test set inputting into model for training and simulation. We use Adam method for gradient descent, and the initial learning rate is 0.0002. We use a mini-batch of size 40 samples for training. In the training process, we optimize the quadratic loss; meanwhile, the cross entropy loss is also calculated. The training process is illustrated in Fig. 3. Fig. 3(b) illustrates the varying curve of the quadratic loss, which is optimized during training. Fig. 3(a) illustrates the cross entropy loss curve, where the perturbation indicates severe noise in the process. The optimization is performed with 680,000 iterations, and the MSE on the test set is 0.004.

The predicted perceptual texture similarity matrix and the texture similarity matrix obtained by the psychophysical experiment are shown in Fig. 4. The size of the predicted perceptual similarity matrix of PTD is 50 × 50 (the number of texture samples in the test set is 50). The color of each element of the matrix represents the perceptual similarity value of the texture pair represented by the corresponding coordinates.

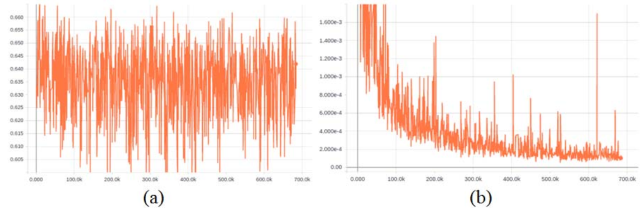

Fig. 3. Training process on PTD. Optimize the network with quadratic loss. The cross-entropy loss and quadratic loss are calculated. Figure (a) illustrates the curve of the cross-entropy loss, and Figure (b) illustrated the curve of the quadratic loss.

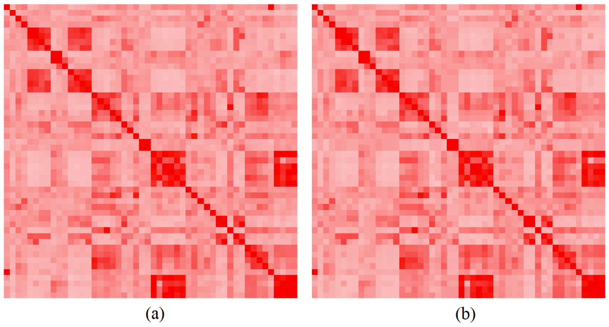

Fig. 4. Visualization of perceptual texture similarity matrix. Figure (a) shows the perceptual similarity matrix obtained by psychophysical experiments, and Figure (b) shows the predicted texture perceptual similarity matrix.

TABLE I

EXPERIMENTAL RESULTS ON PROCEDURAL TEXTURE DATASET

Method	Features	Deviations	MSE	ρ^*
Distance	LBP	-	-	0.2207
	Gabor	-	-	0.4447
	PCANet-48D	-	-	0.5782
	CNN-48D	-	-	0.6266
Random Forest[16]	LBP	0.072	0.012	0.8272
	Gabor	0.065	0.010	0.8657
	PCANet-48D	0.088	0.016	0.8044
	CNN-48D	0.092	0.017	0.8048
Auto-Encoder[55]	LBP	0.108	0.022	0.6113
	Gabor	0.073	0.012	0.8077
	PCANet-48D	0.074	0.013	0.7915
	CNN-48D	0.062	0.010	0.8560
Ours	Layer-wise	**0.043**	**0.004**	**0.9402**

* The correlation coefficients between the distance of computational features and the psychophysical data, and the correlation coefficients between the predicted similarity values and similarity values obtained from psychophysical experiments.

Red indicates that the perceptual similarity is 1, while white indicates that the perceptual similarity is zero. The more red the color, the more similar the two textures are.

It can be seen from the figure that the predicted texture perceptual similarity matrix and the perceptual similarity matrix obtained by psychophysical experiments are very similar to each other. This further proves that the perceptual similarity predicted by the similarity network can fit the psychophysical data well. We compare our method with the similarity regression method proposed in [16] and [55], the results are shown in Table. I. As can be seen from Table. I, our method produces better results with smaller Mean Squared Errors (MSE) and higher correlation coefficients. It should be noted that the methods reported in [16] and [55] require to select the best features but ours does not need this stage.

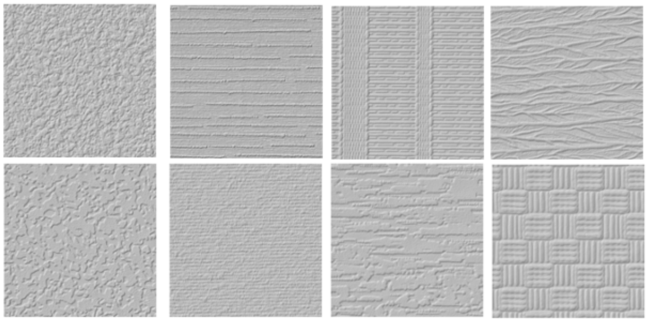

Fig. 5. Paired texture images in Pertex. From left to right, the perceptual similarity values of the textures set are 0.7451, 0.4482, 0.2127 and 0.1280, whereas the predicted similarity values are 0.7108, 0.6160, 0.4494 and 0.1933.

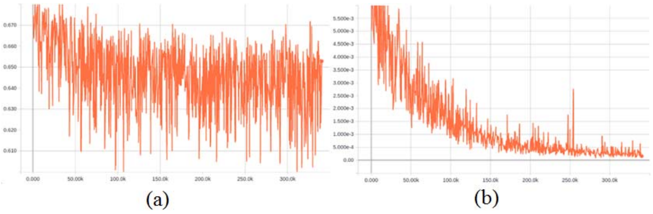

Fig. 6. Training process on Pertex. Optimize the network with quadratic loss. The cross-entropy loss and quadratic loss are calculated. Figure (a) illustrates the curve of the cross-entropy loss, and Figure (b) illustrated the curve of the quadratic loss.

TABLE II

EXPERIMENTAL RESULTS ON PERTEX DATASET

Method	Features	Deviations	MSE	ρ*
Random Forest[16]	LBP	0.121	0.025	0.5430
	Gabor	0.111	0.021	0.6890
	PCANet-48D	0.122	0.026	0.6259
	CNN-48D	0.114	0.024	0.6171
Auto-Encoder[55]	LBP	0.128	0.030	0.3564
	Gabor	0.105	0.019	0.6696
	PCANet-48D	0.138	0.032	0.4687
	CNN-48D	0.108	0.019	0.7037
Ours	Layer-wise	**0.088**	**0.013**	**0.7805**

* The correlation coefficients between the predicted similarity values and similarity values obtained from psychophysical experiments.

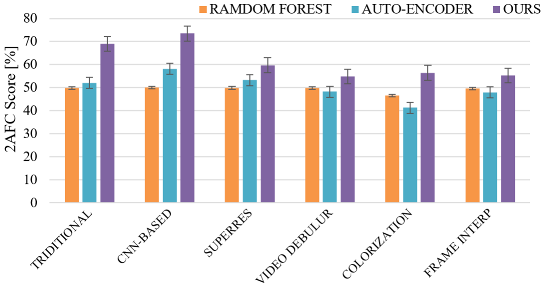

Fig. 7. The 2AFC scores are calculated for the predicted similarity values. The results of different predicting methods are shown in the histogram.

B. Results on Pertex Dataset

The Pertex Dataset contains 334 textures with different resolutions ingested in a variety of real surface textures, including: canvas, woven wallpaper, carpets, curtains, soft fabrics, building materials, and product packaging. Fig. 5 shows four pairs of textures in the Pertex Dataset with their fine-grained perceptual similarity values obtained in pairwise comparison experiments. For the comparison purpose, we also list the similarity values predicted by our methods. The perceptual similarity values of Pertex dataset are in a range from 0 to 1. A value close to 1 means that the two texture images are perceived very similar, whereas a value of 0 represents that the two textures are not similar at all. In Pertex dataset, we randomly choose 300 images to form the training dataset, and the other 34 images as the test set. There are totally 45,150 paired textures with perceptual similarity values in the training dataset, and 595 paired textures in the test set. The hyper-parameters of the experiment are the same as those in the PTD, and the texture images are also resized to 224 × 224 for the reasons explained in the above section. The training process is illustrated in Fig. 6. The optimization converges at 340,000 iterations, and the MSE on the test set is 0.013. We also compare our method with the similarity regression method proposed in [16] and [55], and the results are shown in Table. II.

C. Transfer to Natural Images

To demonstrate the generalization ability of our proposed model, we further transfer our trained model generated from PTD to natural images. In [27], Zhang *et al.* proposed a dataset consisting of natural images and their distortion counterparts. In the dataset, two types of validation sets are given. The first type consists of traditional and CNN-based distortion methods, and the second type consists of real algorithms. In the dataset, each sample is a triplet consisting of one reference image and two distorted images. The distorted images are the outputs of certain algorithms, which take the reference images as the inputs and output processed results. In the triplet, one similarity prediction algorithm will pick up one distorted image similar to the reference. The algorithm is evaluated by a metric called two alternative forced choice(2AFC) score, that asks which of two textures is more similar to a reference. Six VAL sets in [27] of 2AFC similarity evaluation experiments in Berkeley-Adobe Perceptual Patch Similarity (BAPPS) dataset are tested with our method and the methods proposed in [16] and [55], and the experimental results are shown in Fig. 7. In the 2AFC similarity judgments experiments [27], two distortions are applied in a reference image patch, humans are supposed to decide which distortion is closer to the original patch. We test our model on this dataset by predicting the fine-grained similarity values between each distortion and the original patch, and the predicted values show which distortion is similar to the original one. It should be noted that, samples in PTD are texture images, which are very different from natural images. That is to say, the target domain is really distinguished from the source domain and the transfer is essentially a difficult task.

The experimental results demonstrate that our method can be transferred to natural images without fine-tuning on the target dataset, which cannot be implemented for the absence of similarity data in the dataset. It is understood that fine-tuning can boost the performance of our method on natural images. Additionally, our model performs much better than that proposed by Lou [16] in the transfer task. All these results demonstrate that our method has a good generalization ability.

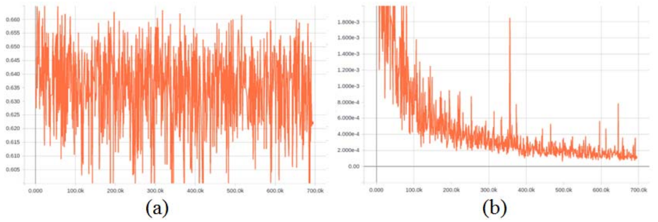

Fig. 8. Optimize the network with cross-entropy loss. The cross-entropy loss and Euclidean loss in the training process are calculated. Figure (a) illustrates the curve of the cross-entropy loss, and Figure (b) illustrates the curve of the quadratic loss during training.

TABLE III

RESULTS ON PTD FOR LEARNING WITHOUT CONTOUR INFORMATION AND OPTIMIZING WITH DIFFERENT LOSSES

Method	Deviations	MSE	ρ^*
Textures+Cross Entropy Loss	0.051	0.005	0.9016
Textures+Quadratic Loss	0.047	0.005	0.9082
Textures+Contours+Cross Entropy Loss	0.044	0.005	0.9361
Textures+Contours+Quadratic Loss	**0.043**	**0.004**	**0.9402**

* The correlation coefficients between the predicted similarity values and similarity values obtained from psychophysical experiments.

V. Ablation Experiments

In this section, we supplement some ablation experiments, including the comparison of different losses, the importance of contour information, the visualization of similarity network weights, the correlation between perceptual similarity and perceptual attributes, and the retrieval-based evaluation experiments. The experimental results are as follows.

A. Quadratic Loss vs. Cross-Entropy Loss

Quadratic loss and cross-entropy loss preform differently in different computer vision tasks [61]–[63]. In order to prove that quadratic loss performs better than cross entropy loss in the similarity prediction task, we carry out a contrastive experiment, in which we alter to optimize the cross-entropy loss. The experiment is conducted on PTD. The training process is illustrated in Fig. 8, where Fig. 8(a) illustrates the varying curve of the cross-entropy loss, and Fig. 8(b) illustrates the quadratic loss curve. The optimization is performed with 690,000 iterations, and the MSE on the test set is 0.005, which is shown in Table. III. Comparing the previous Fig. 3 with Fig. 8, the training process are optimized by quadratic Loss and cross-entropy loss respectively. The training curves are very similar to each other, while the training process optimized by quadratic seems more stable.

B. Contour Information

Contour cues are proved to be important in human visual system [28], [29] attributed to their long-range interactions encoded by contours [23], [30]. The contour maps of the images are extracted using perceptually motivated image features (PMIF) [24], as shown in Fig. 10. The first line is the original texture image, and the second line is the corresponding contour map. We emphasize that contour information aids similarity predicting task. We design experiments to demonstrate this process. In this experiment, we use the same architecture and configuration as mentioned before, except that the contour information is dropped and the batch size is set as 80. The training process is illustrated in Fig. 9, and the final result is shown in Table. III. The first two lines are the results of the trained model with only original texture images,while the last two lines shows the results with input of both texture images and contour maps. The best result of the experiment without contour information is shown in the second line. The optimization is performed by 780,000 iterations, and the MSE is 0.005, which is no good as the MSE calculated with the network trained with both textures and contour maps. This has a more obvious effect on the correlation analysis. When the training network without contour information, the correlation between the predicted similarity and ground truth is 0.9082, it is much higher when training the network with contour information. This demonstrates that contour information indeed helps the similarity predicting task.

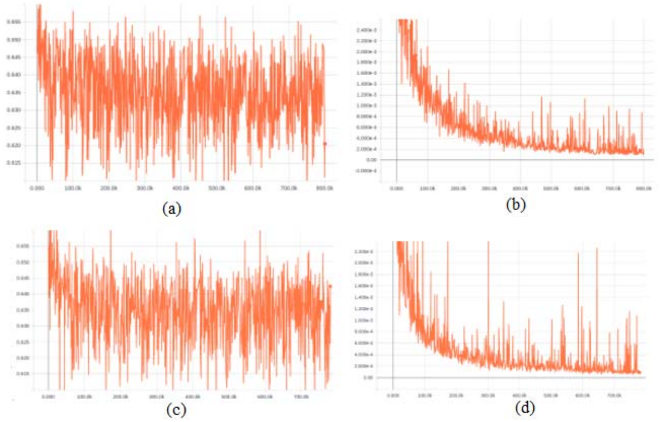

Fig. 9. Learning similarity without contour information. (a) and (b) are the cross entropy loss curve and the quadratic loss curve when optimized by cross entropy loss, (c) and (d) are the cross entropy loss curve and the quadratic loss curve when optimized by quadratic loss.

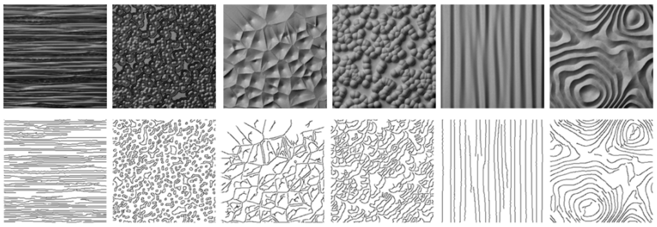

Fig. 10. Original textures and its contour maps. The first line is the original texture image, and the second line is the corresponding contour map.

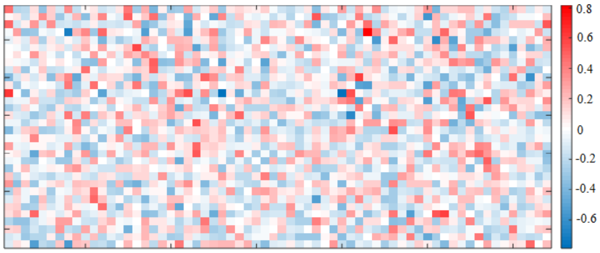

Fig. 11. Similarity network weight visualization results. Each row represents a similarity values calculated in certain layer, and each column represents the weights connected to a neuron in the first fully connected layer.

C. Weight Visualization

As shown in [25], people tend to understand which layer is important in texture tasks, such as texture synthesis, similarity predicting, and so on. It is tedious to design many experiments

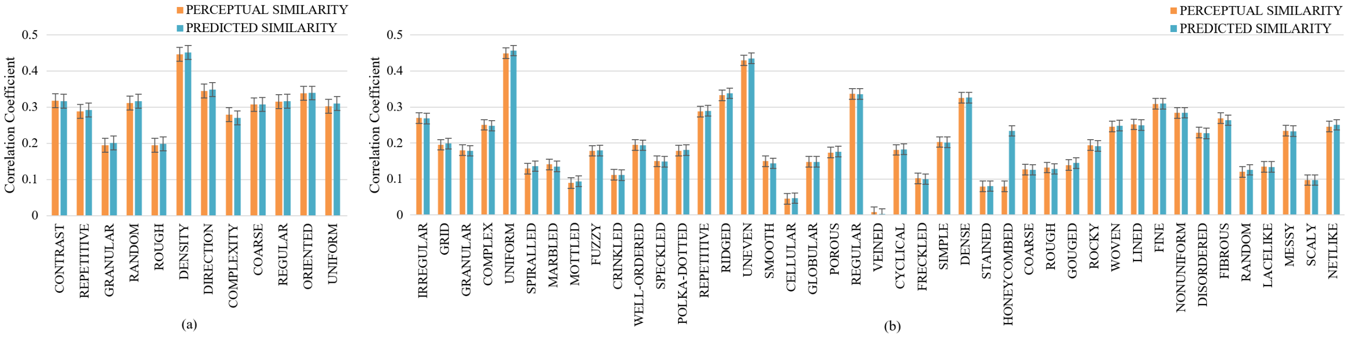

Fig. 12. Correlation analysis. The orange lines represents the correlation between the real perceptual similarity and the perceptual attributes, and blue lines represents the correlation between the predicted perceptual similarity and the perceptual attributes. Figure (a) shows 12 perceptual attributes, and Figure (b) shows 43 semantic attributes.

to find good recipes of the layers used for texture representation. In our solution, we use a fully connected layer on the calculated cosine similarity values, and let the fully connected layer pick appropriate layers by itself. At last, we visualize the weights learned in the first fully connected layer of the proposed similarity network, and figure out which layer is more important in the similarity prediction task.

The visualization result is shown in Fig. 11. In Fig. 11, each row represents a similarity value calculated in a certain layer. The first 16 rows represent the similarity values calculated from the raw texture images, and the subsequent 16 rows represent the similarity values calculated from the contour maps. Each column represents the weights connected to a neuron in the first fully connected layer. If a similarity value calculated in a certain layer is useless for the perceptual similarity predicting task, the weights connected to this value will tend to be zeros, otherwise, the weights will have a large magnitude, regardless of signs. Therefore, in order to absorb which layer is more important for the similarity prediction task, we take the absolute value of the weights connected to all these similarity values calculated in all the convolutional layers. If the weights connected to a similarity value has a high magnitude, they will be rendered as bright; otherwise, they will be rendered as black. In Fig. 11, we cannot find significant bias of the brightness distribution of the weight in the first fully connected layer. Thus, all the layers from low to high levels contribute equally to the similarity prediction task.

D. Perceptual Similarities V.S. Perceptual Attributes

Texture perception study has recently concentrated on the perceptual attributes [8], [64] and semantic attributes [65]. In the prior work, texture perceptual attributes are treated as abstract concepts which represent humans subjective perception, while texture semantic attributes are related to specific words. Both perceptual attributes and semantic attributes are used to express the similarities and difference between texture images in order to attain a better understanding of the content of images. Liu *et al.* [12] performed psychophysical experiments on the procedural texture dataset, and obtained 12 kinds of perceptual features from the quantized values of each texture image through perceptual scoring experiments. The 12 perceptual attributes are contrast, repetitive, granular, random, rough, feature density, direction, structural complexity, coarse, regular, oriented, and uniform. Dong *et al.* [66] also semantically annotated PTD through psychophysical experiments and obtained 43 semantic attributes. The perceptual attribute data obtained by the perceptual scoring experiment is based on the quantified perceptual data obtained by the human visual perception system on the texture image, and the semantic attribute of the texture image is obtained by free grouping labeling. Since the above two kinds of perceptual data are closely related to human perception and understanding of texture images, we try to find the inherent connection between these attributes and perceptual similarity. To this end, we analyze the correlation between perception features and perceptual similarity for each perceptual attributes and semantic attribute.

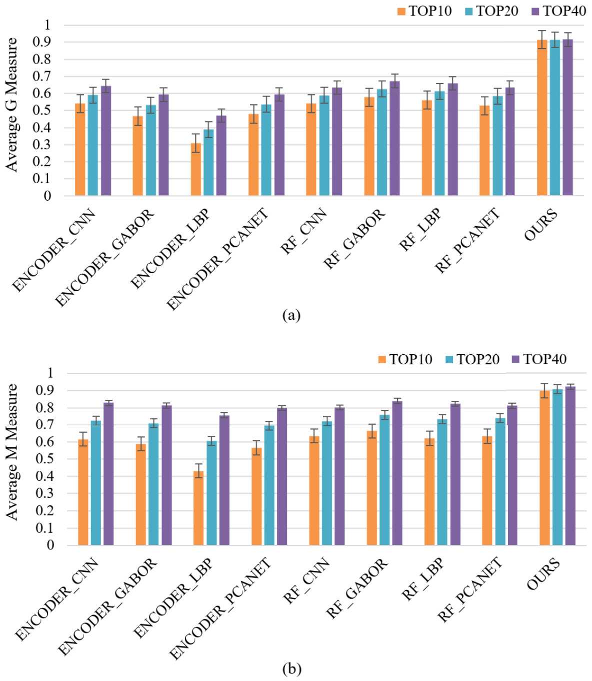

Fig. 13. Bar chart of the average G (a) and average M (b) measures obtained using different similarity prediction methods.

To obtain the correlation between perceptual similarities and perceptual features, firstly, we pre-process the perceptual data, and analyze the perception features according to the 450 texture images. Taking the repetitive as an example, we calculate the distance of the repetitive values between all the texture images. Since the dataset contains 450 texture images, each texture image can be grouped with 450 textures.

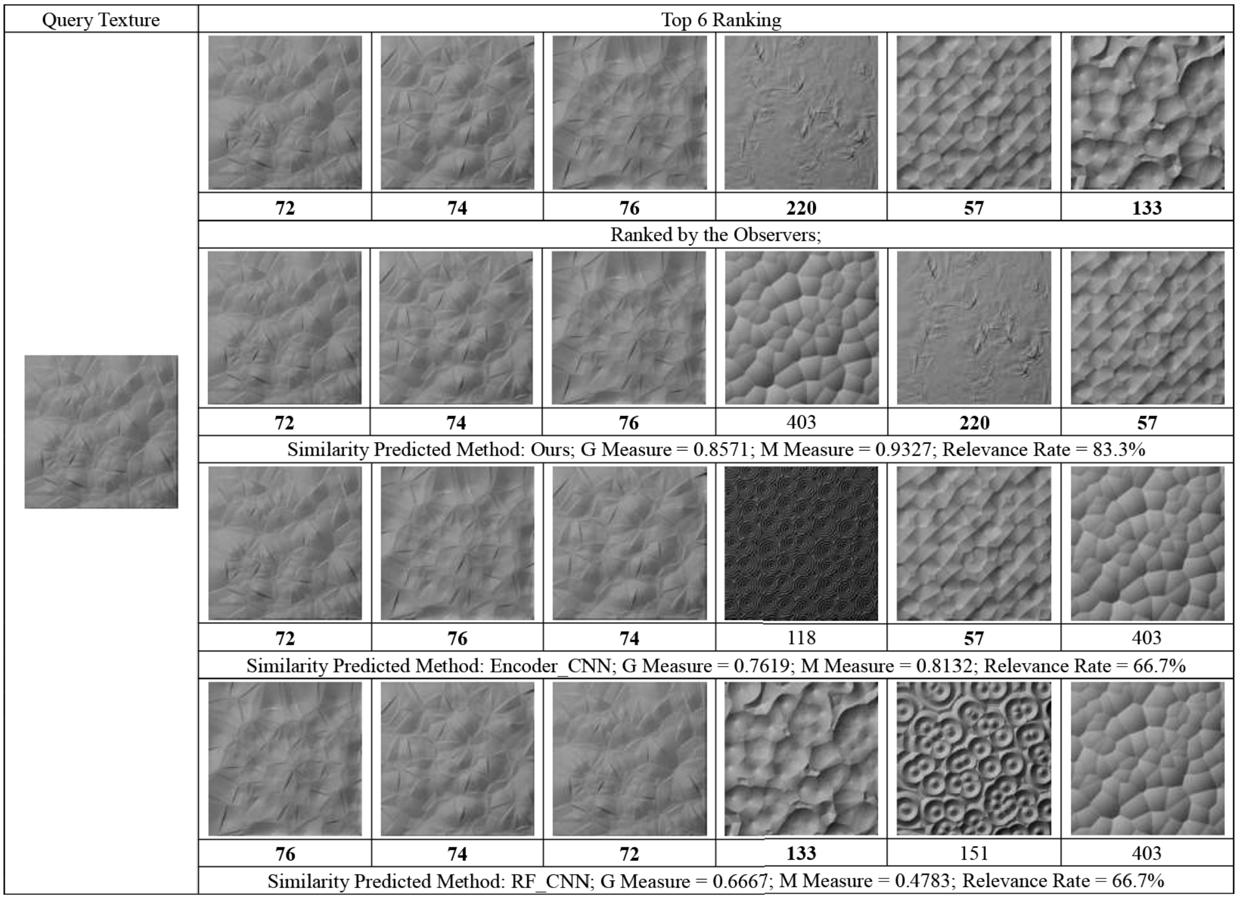
Query Texture
Top 6 Ranking
72
74
76
220
57
133
Ranked by the Observers;
72
74
76
403
220
57
Similarity Predicted Method: Ours; G Measure = 0.8571; M Measure = 0.9327; Relevance Rate = 83.3%
72
76
74
118
57
403
Similarity Predicted Method: Encoder_CNN; G Measure = 0.7619; M Measure = 0.8132; Relevance Rate = 66.7%
76
74
72
133
151
403
Similarity Predicted Method: RF_CNN; G Measure = 0.6667; M Measure = 0.4783; Relevance Rate = 66.7%

Fig. 14. Top 6 ranking of the textures in the retrieval experiments. The first column on the left is the query texture, and the images on the right are the ranking of the top 6 images. The corresponding G Measure and M Measure values are noted below.

If the repeatability between the two images is stronger, then the distance is smaller; conversely, if the distance of the two texture images is larger, the two texture images have large difference in repeatability. We perform correlation analysis between each perceptual attributes and semantic attributes with perceptual similarity as shown in Fig. 12.

Through experimental results, it can be found that there is a certain degree of connection between perceptual attributes, semantic attributes and perceptual similarities. Overall, the correlation between individual perception features and perceptual similarity is not large. Suppose that 12 perceptual attributes and 43 semantic attributes are distributed in a 12-dimensional perceptual space and a 43-dimensional semantic space respectively. The perceptual feature cosine distance between two images in the feature space is compared with the perceptual similarity of the original texture images. The correlation should be significantly higher than the correlation between individual perception features and perceptual similarity. The correlation coefficient between the perceptual distance of the texture image in the perceptual space and the real perceptual similarity is analyzed. The results are shown in Table. IV.

TABLE IV
THE CORRELATION COEFFICIENT BETWEEN THE PERCEPTUAL DISTANCE IN THE PERCEPTUAL SPACE AND THE PERCEPTUAL SIMILARITY

	Similarity Value	Predicted Similarity	Perceptual Attributes	Semantic Attributes
Similarity Value	1	0.9929	0.4977	0.7739
Predicted Similarity	0.9929	1	0.4711	0.7741
Perceptual Attributes	0.4977	0.4711	1	0.6386
Semantic Attributes	0.7739	0.7741	0.6383	1

When the perception feature is 0 in the perceptual space, the two texture images are consistent, and the corresponding perceived similarity is 1; when the distance between perception features in the perceptual space becomes larger, the similarity of the two texture images decreases, and the corresponding perceived similarity value becomes smaller. It is found in the experiments that the perception feature distance in the perceptual space has a stronger correlation with the real texture perceptual similarity than the individual perceptual properties. It is witnessed that humans consider a variety of perceptual factors when they perceive the similarity of texture images.

E. Retrieval-Based Evaluation Experiments

In order to augment the predicting results obtain in Section.IV, another evaluation experiment named texture retrieval is reported. The retrieval method we used is proposed in [13], which compares the top N rankings sorted by humans with the top N rankings sorted using predicted similarity values. It allows a sequence sorted from large to small according to the values of the similarity, which visually shows the similarity between one query texture and other texture images. The measure method we used is M measure and G measure, and the data we used for ranking is the test set including 50 images. G and M measures can compare both two identical rankings and two nonidentical rankings. Fig. 13 presents the average M measures and average G measures using different similarity values predicted with different methods.

We test the similarities obtained by different similarity prediction methods for texture retrieval as shown in Fig. 14. Experimental results show that the proposed fine-grained perceptual similarity prediction method can achieve the best retrieval results. More retrieval experimental results can be found in the supplementary material.

VI. Conclusion

In this paper, we propose to learn the fine-grained perceptual similarity values. As perceptual similarity is very subjective, and the fair data is very hard to obtain, there are few datasets containing images and complete similarity data. As a result, very few studies intend to learn the fine-grained perceptual similarity values, in spite of its importance. As rank data is relatively easy to obtain, many efforts on similarity learning have focused on this area. In order to solve this problem, we proposed to use deep convolutional networks, joint with contour information for fine-grained perceptual similarity value predicting. The paired original textures and the contour maps of the textures are fed to the network to calculate the cosine similarities in the feature spaces. The calculated cosine similarity values were concatenated together as a feature vector, and then fed into a full connected network to predict the perceptual similarity value of the paired textures.

We conducted several ablation experiments, and demonstrate that the contour information indeed helps the similarity predicting task. The visualization results of the fully connected layer in the similarity network indicate that the information coming from different layers in the convolutional network contribute equally to the similarity predicting task. It is consistent with our assumption that human beings always give similarity result by observing the texture images from local regions to global patterns. As predicting perceptual similarity values is a very fundamental problem, it deserves more efforts in this area. In future work, we will also consider integrating more texture encoding methods with the hope to add better features to the existing architecture, and we also would like to see texture similarity learning can be to other areas, such as material similarity prediction.

References

[1] L. Sharan, C. Liu, R. Rosenholtz, and E. H. Adelson, "Recognizing materials using perceptually inspired features," *Int. J. Comput. Vis.*, vol. 103, no. 3, pp. 348–371, Jul. 2013.

[2] G. Schwartz and K. Nishino, "Automatically discovering local visual material attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3565–3573.

[3] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[4] R. Liu *et al.*, "Multiscale road centerlines extraction from high-resolution aerial imagery," *Neurocomputing*, vol. 329, pp. 384–396, Feb. 2019.

[5] J. Zhu, H. Zeng, S. Liao, Z. Lei, C. Cai, and L. Zheng, "Deep hybrid similarity learning for person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 11, pp. 3183–3193, Nov. 2018.

[6] B. Julesz, "Textons, the elements of texture perception, and their interactions," *Nature*, vol. 290, no. 5802, pp. 91–97, 1981.

[7] B. Julesz, "Experiments in the visual perception of texture," *Sci. Amer.*, vol. 232, no. 4, pp. 34–43, Apr. 1975.

[8] B. Julesz, "Visual pattern discrimination," *IRE Trans. Inf. Theory*, vol. 8, no. 2, pp. 84–92, Feb. 1962.

[9] H. Tamura, S. Mori, and T. Yamawaki, "Textural features corresponding to visual perception," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-8, no. 6, pp. 460–473, Jun. 1978.

[10] H. Christopher and H. Stephen, "Similarity and features of natural textures," *J. Exp. Psychol., Human Perception Perform.*, vol. 25, no. 2, pp. 299–320, Apr. 1999.

[11] A. R. Rao and G. L. Lohse, "Towards a texture naming system: Identifying relevant dimensions of texture," *Vis. Res.*, vol. 36, no. 11, pp. 1649–1669, Jun. 1996.

[12] J. Liu, J. Dong, X. Cai, L. Qi, and M. Chantler, "Visual perception of procedural textures: Identifying perceptual dimensions and predicting generation models," *PLoS ONE*, vol. 10, no. 6, Jun. 2015, Art. no. e0130335.

[13] X. Dong, "Perceptual texture similarity estimation," Ph.D. dissertation, Heriot-Watt Univ., Edinburgh, U.K., 2014.

[14] V. Andrearczyk and P. F. Whelan, "Convolutional neural network on three orthogonal planes for dynamic texture classification," *Pattern Recognit.*, vol. 76, pp. 36–49, Apr. 2017.

[15] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[16] J. Lou, L. Qi, J. Dong, H. Yu, and G. Zhong, "Learning perceptual texture similarity and relative attributes from computational features," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2016, pp. 2540–2546.

[17] Y. Gao, L. Wang, and K. L. Chan, "Learning texture similarity with perceptual pairwise distance," in *Proc. 4th Int. Workshop Texture Anal. Synth.*, 2005, pp. 83–88.

[18] H. Long and W. K. Leow, "A hybrid model for invariant and perceptual texture mapping," in *Proc. Object Recognit. Supported User Interact. Service Robots*, Aug. 2002, pp. 135–138.

[19] M. Wang, X. S. Hua, J. Tang, and R. Hong, "Beyond distance measurement: Constructing neighborhood similarity for video annotation," *IEEE Trans. Multimedia*, vol. 11, no. 3, pp. 465–476, Apr. 2009.

[20] S. Zhou, J. Wang, R. Shi, Q. Hou, Y. Gong, and N. Zheng, "Large margin learning in set-to-set similarity comparison for person reidentification," *IEEE Trans. Multimedia*, vol. 20, no. 3, pp. 593–604, Mar. 2018.

[21] S. Berretti, A. D. Bimbo, and P. Pala, "Retrieval by shape similarity with perceptual distance and effective indexing," *IEEE Trans. Multimedia*, vol. 2, no. 4, pp. 225–239, Dec. 2000.

[22] M. Koskela, A. F. Smeaton, and J. Laaksonen, "Measuring concept similarities in multimedia ontologies: Analysis and evaluations," *IEEE Trans. Multimedia*, vol. 9, no. 5, pp. 912–922, Aug. 2007.

[23] X. Dong and M. J. Chantler, "The importance of long-range interactions to texture similarity," in *Proc. Int. Conf. Comput. Anal. Images Patterns*, 2013, pp. 425–432.

[24] X. Dong and M. J. Chantler, "Perceptually motivated image features using contours," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5050–5062, Nov. 2016.

[25] L. A. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 262–270.

[26] J. A. Movshon and E. P. Simoncelli, "Representation of naturalistic image structure in the primate visual cortex," *Cold Spring Harbor Symposia Quant. Biol.*, vol. 79, pp. 115–122, Jan. 2014.

[27] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[28] D. J. Field, A. Hayes, and R. F. Hess, "Contour integration by the human visual system: Evidence for a local 'association field'," *Vis. Res.*, vol. 33, no. 2, pp. 173–193, Jan. 1993.

[29] J. De Winter and J. Wagemans, "The awakening of attneave's sleeping cat: Identification of everyday objects on the basis of straight-line versions of outlines," *Perception*, vol. 37, no. 2, pp. 245–270, Jan. 2008.

[30] X. Dong and J. Dong, "The visual word booster: A spatial layout of words descriptor exploiting contour cues," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3904–3917, Aug. 2018.

[31] G. A. Gescheider, *Psychophysics: The Fundamentals*. Hillsdale, NJ, USA: L. Erlbaum Associates, 1997, pp. 1035–1042.

[32] B. P. I. Hersey, *Textures: A Photographic Album for Artists and Designers*, vol. 1. Los Angeles, CA, USA: Leonardo, 1968, no. 1.

[33] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, and S. Huovinen, "Outex—New framework for empirical evaluation of texture analysis algorithms," in *Proc. Object Recognit. Supported User Interact. Service Robots*, Aug. 2002, pp. 701–706.

[34] K. J. Dana, S. K. Nayar, B. van Ginneken, and J. J. Koenderink, "Reflectance and texture of real-world surfaces authors," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1997, pp. 151–157.

[35] M. Amadasun and R. King, "Textural features corresponding to textural properties," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 5, pp. 1264–1274, Sep./Oct. 1989.

[36] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 837–842, Aug. 1996.

[37] E. P. Simoncelli and J. Portilla, "Texture characterization via joint statistics of wavelet coefficient magnitudes," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 1998, pp. 62–66.

[38] N. Abbadeni, "Computational perceptual features for texture representation and retrieval," *IEEE Trans. Image Process.*, vol. 20, no. 1, pp. 236–246, Jan. 2011.

[39] W. Ma and B. S. Manjunath, "Texture features and learning similarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 1996, pp. 425–430.

[40] J. Portilla and E. P. Simoncelli, "Texture modeling and synthesis using joint statistics of complex wavelet coefficients," in *Proc. IEEE Workshop Stat. Comput. Theories Vis.*, Jun. 1999, pp. 49–71.

[41] F. Halley, "Perceptually relevant browsing environments for large texture databases," Ph.D. dissertation, Heriot-Watt Univ., Edinburgh, U.K., 2012.

[42] A. D. F. Clarke, F. Halley, A. J. Newell, L. D. Griffin, and M. J. Chantler, "Perceptual similarity: A texture challenge," in *Proc. Brit. Mach. Vis. Conf.*, Jan. 2011, pp. 1–10.

[43] J. B. Tenenbaum, V. D. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

[44] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[45] C. H. Chen, *Handbook of Pattern Recognition and Computer Vision*, vol. 50. Singapore: World Scientific, 2005, p. 996, no. 2.

[46] M. Mirmehdi, X. Xie, and J. Suri, *Handbook of Texture Analysis*. London, U.K.: Imperial College Press, 2008, p. 424.

[47] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[48] L. Sifre and S. Mallat, "Rotation, scaling and deformation invariant scattering for texture discrimination," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1233–1240.

[49] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[50] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[51] H. Zhang, J. Xue, and K. Dana, "Deep ten: Texture encoding network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2896–2905.

[52] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4353–4361.

[53] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "MatchNet: Unifying feature and metric learning for patch-based matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3279–3286.

[54] K. He, Y. Lu, and S. Sclaroff, "Local descriptors optimized for average precision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 596–605.

[55] Y. Gao, Y. Gan, J. Dong, L. Qi, and H. Zhou, "Perceptual texture similarity learning using deep neural networks," in *Proc. 13th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery*, Jul. 2017, pp. 856–860.

[56] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[57] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2818–2826.

[58] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 2672–2680.

[59] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[60] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015.

[61] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," Feb. 2017, *arXiv:1702.05659*. [Online]. Available: https://arxiv.org/abs/1702.05659

[62] L. Rosasco, E. D. Vito, A. Caponnetto, M. Piana, and A. Verri, "Are loss functions all the same?" *Neural Comput.*, vol. 16, no. 5, pp. 1063–1076, May 2004.

[63] H. Masnadi-Shirazi and N. Vasconcelos, "On the design of loss functions for classification: Theory, robustness to outliers, and SavageBoost," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1049–1056.

[64] A. R. Rao and G. L. Lohse, "Identifying high level features of texture perception," *CVGIP, Graph. Model Image Process.*, vol. 55, no. 3, pp. 218–233, May 1993.

[65] D. Parikh and K. Grauman, "Relative attributes," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 503–510.

[66] J. Dong, L. Wang, J. Liu, Y. Gao, L. Qi, and X. Sun, "A procedural texture generation framework based on semantic descriptions," *Knowl.-Based Syst.*, vol. 163, pp. 898–906, Jan. 2019.

Ying Gao received the B.Sc. from the Chengdu University of Technology in 2015. She is currently pursuing the Ph.D. degree with the Ocean University of China. Her research interests include computer vision and machine learning.

Yanhai Gan received the master's degree from the Ocean University of China in 2017, where he is currently pursuing the Ph.D. degree. He was an Image Algorithm Engineer with Hisense TransTech Company, Ltd., China, in 2018. His research interests include machine learning and image processing.

Lin Qi received the B.Sc. and M.Sc. degrees from the Ocean University of China in 2005 and 2008, respectively, and the Ph.D. degree in computer science from Heriot-Watt University in 2012. He is currently an Associate Professor with the Department of Computer Science and Technology, Ocean University of China. His research interests include computer vision and visual perception.

Huiyu Zhou received the B.E. degree in radio technology from the Huazhong University of Science and Technology, China, the M.Sc. degree in biomedical engineering from the University of Dundee, U.K., and the Ph.D. degree in computer vision from Heriot-Watt University, Edinburgh, U.K. He has published widely in the field. He currently serves as an editorial board member and a guest editor of several refereed journals. He was a recipient of the CVIU 2012 Most Cited Paper Award and the ICPRAM 2016 Best Paper Award in the Area of Applications and was shortlisted for the ICPRAM 2017 Best Student Paper Award and the MBEC 2006 Nightingale Prize. He also serves as the Editor-in-Chief for *Recent Advances in Electrical and Electronic Engineering* and an Associate Editor for the IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS.

Xinghui Dong received the Ph.D. degree from Heriot-Watt University, U.K., in 2014. He is currently a Research Associate with the Centre for Imaging Sciences, The University of Manchester, U.K. His research interests include automatic defect detection, image representation, texture analysis, and visual perception.

Junyu Dong received the B.Sc. and M.Sc. degrees from the Department of Applied Mathematics, Ocean University of China, in 1993 and 1999, respectively, and the Ph.D. degree in image processing from the Department of Computer Science, Heriot-Watt University, U.K., in November 2003. He joined the Ocean University of China in 2004, where he is currently a Professor and the Head of the Department of Computer Science and Technology. His research interests include machine learning, big data, computer vision, and underwater image processing.