Unifying the Visual Perception of Humans and Machines on Fine-Grained Texture Similarity

Weibo Wang
wwb@stu.ouc.edu.cn
Xinghui Dong†
xinghui.dong@ouc.edu.cn

School of Computer Science and Technology
Ocean University of China
Qingdao, China
† Corresponding author

Abstract

Texture similarity plays an important role in texture analysis and material recognition. However, the prediction of perceptually-consistent fine-grained texture similarity is a challenging task. It has been found that the discrepancy between the texture representation methods and the similarity metrics utilised by humans and algorithms should account for the dilemma. To address this problem, we propose a novel Perceptually Motivated Texture Similarity Prediction Network (PMTSPN), which comprises a siamese Conformer with multi-scale bilinear pooling (SC-MSBP) and a metric learning network (MLN). The SC-MSBP learns a texture representation capturing the Higher Order Statics (HOS) in different spatial scales, while the MLN learns a similarity metric from the features which encode the short-range, long-range and lateral interactions. The PMTSPN can be trained using a set of human perceptual similarity data. Our results show that the PMTSPN produces the more consistent similarity prediction with human perception, compared with its counterparts. We attribute the promising performance to both the powerful texture representation and the effective similarity metric learnt by the PMTSPN. Code is available at: https://github.com/INDTLab/PMTSPN.

1 Introduction

As one of the inherent properties of the material surface, texture enables humans to accurately perceive and identify objects. Texture similarity aims at judging whether or not two textures are similar, or the extent that they are similar. However, this task encounters two challenges: texture representation and similarity metric. As a result, the prediction of perceptually-consistent fine-grained texture similarity using algorithms is struggling. In Figure 1, the four textures shown at the right side share the same **SSIM** [42] value compared with the query texture. In other words, it was decided using the **SSIM** [42] that they show the same level of similarity to the query. Nevertheless, it is not the case for human perception.

Clarke *et al*. [8] used four algorithms to predict the fine-grained perceptual texture similarity data collected using 30 human subjects. The experiment demonstrated that only a weak correlation (Spearman's correlation coefficient $\rho = 0.21$) between the algorithmic decisions

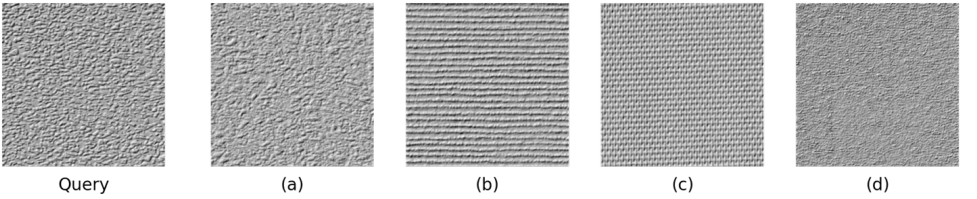

Figure 1: A query texture and four retrieval textures contained in the *Pertex* data set [8]. Each retrieval texture has the same SSIM [42] value (0.0264) compared with the query texture. However, the corresponding similarity values contained in the Isomap perceptual similarity matrix [9] manifest great variations, which are 0.85286, 0.37449, 0.36677 and 0.544 in turn.

and human perception was obtained. In the recent study, Dong *et al*. [13] showed that the similarity data obtained using 51 texture descriptors were considerably inconsistent with that perceived by humans. They attributed this inconsistency to the finding that these algorithms do not exploit long-range interactions (dependencies) as well as humans.

Traditionally, texture descriptors were normally designed based on the response of filters [1, 29], the statistics of gray level values [18], the structural characteristics [27, 42] and a texture model [30]. With the development of deep learning, in particular, the Convolutional Neural Network (CNN) techniques have been widely applied to computer vision tasks and produced the better results than traditional methods. These techniques can directly learn features from a large set of training samples. It has been demonstrated that the features extracted from the convolutional layers of a pre-trained CNN model can be used for texture representation [7]. Motivated by this study, Gao *et al*. [16] proposed a texture similarity prediction framework on top of the pre-trained VGG-VD-19 model [35]. In this framework, the cosine similarity was computed in the feature space in order to measure the similarity between a pair of textures. However, there is no evidence whether or not the cosine similarity is consistent with human perception.

To derive the perceptually-consistent fine-grained texture similarity, we introduce a novel Perceptually Motivated Texture Similarity Prediction Network (PMTSPN). This network comprises two subnetworks: a siamese Conformer [33] with multi-scale bilinear pooling (SC-MSBP) and a metric learning network (MLN). The SC-MSBP takes a pair of textures as the input. The output encodes the Higher Order Statics (HOS) in different spatial extents. The features extracted from a pair of textures at a layer are concatenated. Then, these features are sent to the MLN which consists of a dual Swin Transformer [26] module and a decision module. The Transformer module is used to learn the lateral interactions [46] between different feature channels and compress the features. All the features are further aggregated to measure the similarity of the two textures, which encodes the short-range, long-range and lateral interactions. Finally, they are fed into the decision module and a similarity score is predicted. The PMTSPN can be trained using a set of textures with the human perceptual similarity data, *e.g*. *Pertex* [8, 9] and *PTD* [25].

The contributions of this study can be summarised as twofold. First, we introduce a multi-scale bilinear pooling based siamese Conformer, which is able to learn the HOS in multiple spatial scales. Second, we design a simple metric learning network on top of a dual Swin Transformer [26] module and a decision module, which can predict the fine-grained texture similarity by exploiting the short-range, long-range and lateral interactions. To our

knowledge, none of existing studies have utilised such tactics for the texture similarity task.

The rest of this paper is organised as follow. We review the literature related to this study in Section 2. The PMTSPN is described in Section 3. In Sections 4 and 5, the experimental setup and results are reported respectively. Finally, we draw our conclusion in Section 6.

2 Related Work

2.1 CNNs and Transformers

With the development of the hardware and the occurrence of large data sets, deep learning has been making great progress. For example, many studies were conducted in the field of computer vision in the past decade, including AlexNet [23], VGG-VD [36] and ResNet [19]. In 2017, Vaswani *et al*. [40] established a Transformer architecture and applied it to Natural Language Processing (NLP). More studies were then introduced based on the Transformer architecture, such as BERT [10] and GPT-3 [4]. Dosovitskiy *et al*. [12] further applied it to computer vision. Since then, both CNNs and Transformers have become the dominant methods for vision tasks. In contrast to CNNs which exploit local features, Transforms encode the global representation by computing the self-attention data. Therefore, we were motivated to utilise the features extracted using both the techniques for texture representation.

2.2 Metric Learning

Metric learning [21] aims at learning representations in the embedding space. The primary purpose of it is to learn a new metric which can reduce the distance between similar samples and increase the distance between dissimilar samples [15]. Traditionally, the Euclidean [43] and Mahalanobis [44] distances were used to measure the similarity between two samples. However, these distances can only capture a few nonlinear information. For instance, the Euclidean distance cannot encode the non-isotropic distance and the class structure. Instead, deep metric learning techniques have been used to learn the nonlinear information. In general, these techniques were designed based on two structures, *i.e.* the Siamese network [3] and the Triplet network [20], respectively. Loss functions are important to differentiating and distinguishing between positive and negative samples. The contrastive loss [17], triplet loss [20] and mixed loss [6] normally focus on modelling the relationship between samples and the distance between positive and negative samples, while the clustering loss [31] and Magent loss [34] pay attention to preventing the overlapping between different categories. Besides, the structured loss [37] learns to distinguish between positive and negative samples using the interactions of multiple samples. In this study, we designed a metric learning subnetwork on top of the siamese structure with the Mean Squared Error (MSE) loss.

2.3 Texture Similarity

Texture similarity is key to human perception and machine vision for recognition of objects. However, it is challenging to derive the perceptually-consistent fine-grained texture similarity using computational approaches [13]. In [8], Clarke *et al*. derived a 334×334 fine-grained perceptual similarity matrix by conducting a free-grouping experiment. Clarke *et al*. [9] further used the Isomap dimensionality reduction [39] to obtain a more compact Isomap similarity matrix. A similar study was conducted by Liu *et al*. [25] on procedural textures.

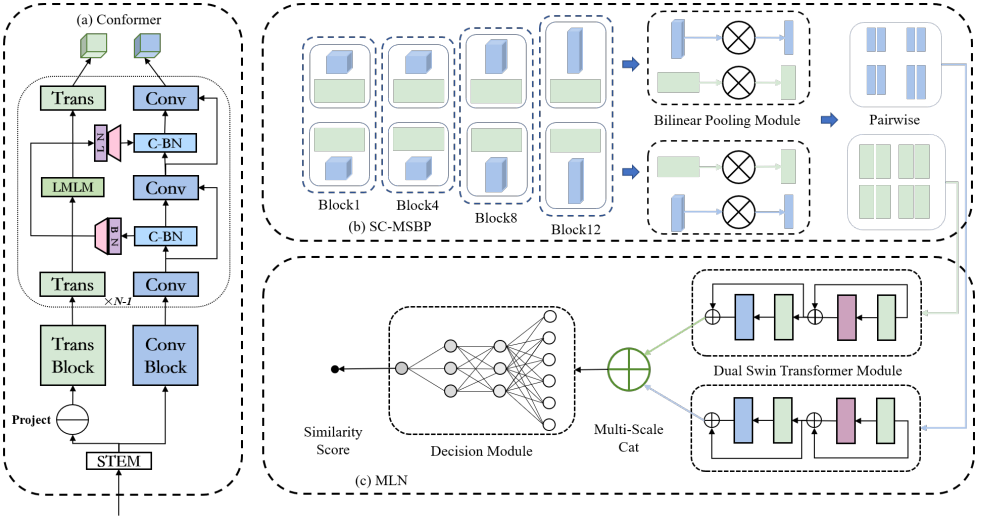

Figure 2: The architecture of the proposed PMTSPN, in which (a) shows the structure of the Conformer [33]; (b) is the SC-MSBP subnetwork, containing a siamese Conformer with multi-scale bilinear pooling [24]; and (c) is the MLN subnetwork, including a dual Swin Transformer [26] module and a decision module.

Using both sets of similarity data, Gao *et al*. [16] trained a texture similarity network for predicting the perceptual texture similarity. Ding *et al*. [12] investigated the characteristics of human perception and proposed a model, referred to as DISTS, that jointly optimised both the texture similarity and the structure similarity. Ding *et al*. [11] also compared other methods with the DISTS approach in low-level vision tasks, such as image denoising, blind image deblurring, single image, super-resolution and lossy image compression. Nevertheless, the above studies were only focused on texture representation but did not consider the similarity metric. In contrast, we exploited both the techniques for texture similarity prediction.

3 Perceptually Motivated Texture Similarity Prediction Network (PMTSPN)

In this section, we will introduce the proposed PMTSPN in detail. Specifically, it comprises two subnetworks: a siamese Conformer with multi-scale bilinear pooling (SC-MSBP) and a metric learning network (MLN). The architecture of the PMTSPN is shown in Figure 2.

3.1 Siamese Conformer with Multi-scale Bilinear Pooling (SC-MSBP)

The SC-MSBP is designed for learning texture representation. It is built on top of a pre-trained Conformer [33] and takes a pair of textures as the input. The Conformer is able to exploit both the local features and global representation of images. Regarding each stream of the SC-MSBP, the local and global features extracted at a layer of the backbone are separately processed using bilinear pooling. The result is a descriptor which encodes the HOS. Each type of features computed from a texture pair at a layer are then concatenated. All the

concatenated features are grouped in terms of the CNN and Transformer branches.

3.1.1 Conformer

As shown in Figure 2 (a), the Conformer [33] has a recurrent and parallel structure, designed to extract and integrate both local and global features using two branches: CNNs and Transformers respectively. It contains 12 blocks and each block undergoes an exchange of the local and global features except the first block. During each exchange, the shape of both types of features is transformed using the 1×1 convolution to achieve the same dimensionality, while the difference caused by *LayerNorm*/*BatchNorm* is neutralised using *BatchNorm*/*LayerNorm* in the Feature Coupling Unit (FCU). Motivated by the illustration that the Human Visual System (HVS) processes images using multiple resolutions [22], the coupling features extracted at four blocks: $Block_1$, $Block_4$, $Block_8$ and $Block_{12}$ are sent to the bilinear pooling module individually, which computes the pair-wise statistics over all feature channels and pools them across each feature map. The reason for using this module is that it computes the HOS in a large spatial extent which are known to be used by the HVS [38].

3.1.2 Multi-scale Bilinear Pooling

Bilinear pooling [24] computes a set of pair-wise statistics between feature channels and captures the channel-to-channel relationship. The computation of it can be expressed as:

$$\mathbf{BP} = \mathbf{sign}(\cdot) \odot |\frac{1}{m \cdot n} \cdot \sum_{l \in \mathcal{L}} F_l^T \times F_l|^{\alpha}. \tag{1}$$

Here, F_l represents the vector of the feature map on the channel $l \in \mathcal{L}$, α is the power law coefficient, $\mathbf{sign}(\cdot)$ denotes the sign function of the sum and $\odot$ is the element-wise multiplication operator. Figure 3 presents the changing process of the shape of feature maps during the computation of bilinear pooling. In this work, we use bilinear pooling to compute the pair-wise relationship between the homologous feature channels. Since the pooling operation is performed across the entire feature map, the resultant features are suitable for comparing the similarity of two textures which have different sizes.

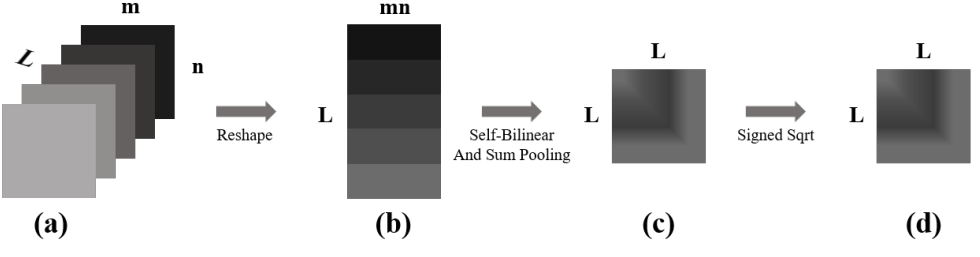

Figure 3: The changing process of the shape of feature maps during bilinear pooling [24].

3.2 Metric Learning Network (MLN)

Both the CNN and Transformer features extracted using the SC-MSBP are sent to the MLN, which contains a dual Swin Transformer [26] module and a decision module. The Swin

Transformer uses a hierarchical and localised ideological structure and is often used as a generic backbone network. The Transformer module contains three Swin Transformer blocks and can be used for two purposes. First, they are used to learn the lateral interactions [46] from the CNN or Transformer features extracted at a single layer across different feature channels, which are also important to the HVS. Second, they are utilised for compressing the features received from the SC-MSBP because these features have a very high dimensionality. All the compressed features in the two streams are further aggregated across multiple layers to measure the similarity of the two textures. This representation encodes the short-range, long-range and lateral interactions. It is fed into the decision module, which consists of three fully-connected layers ($N \rightarrow 1024, 1024 \rightarrow 1024, 1024 \rightarrow 1$), two *ReLU* activation functions and a *Sigmoid* activation function. The output is the fine-grained similarity score between the two textures.

4 Experimental Setup

In this section, we will describe the setup used in our experiment, including the data sets, performance measures and implementation details.

4.1 Data Sets

The *Pertex* [8] and *PTD* [25] data sets that we used contain two different types of textures, *e.g.* natural and procedural textures, respectively. The *Pertex* data set [8] contains 334 textures, captured from the decorative materials, such as wallpaper, canvas, carpet and drapery. A free-grouping experiment was conducted using this data set and a 334×334 similarity matrix was derived. The Isomap dimensionality reduction [39] method was applied to it. The result was a more compact Isomap similarity matrix. The human perceptual similarity scores contained in this matrix were linearly normalised to the range of [0, 1]. In contrast, the *PTD* data set [25] comprises 450 textures, generated using 23 procedural texture models. The similar experiment was performed and an Isomap similarity matrix was achieved.

Following the setup used by Gao *et al.* [16], we randomly divided the *Pertex* textures into two subsets, which contained 300 training images and 34 test images, respectively. To keep consistent with the work presented in [16], we fixed the random seed. As a result, we derived a total of 45,150 pairs of textures for training and a set of 595 pairs of textures for testing. Likewise, the 450 *PTD* textures were randomly split into two groups, including 400 training images and 50 test images, respectively. In this case, a set of 80,200 pairs of textures were used for training and 1,275 pairs of textures were utilised for testing.

4.2 Performance Measures

Given a texture data set with the perceptual similarity data, let $\mathbf{x}$ and $\mathbf{y}$ denote a pair of textures and $\mathbf{s}$ stands for the perceptual similarity between them. For the *Pertex* [8] and *PTD* [25] data sets, $\mathbf{s}$ is a decimal number between 0 and 1, where 0 represents that the two textures are dissimilar at all while 1 means that they are completely similar or even are the same. We use $\bar{\mathbf{s}}$ to represent the similarity predicted using an algorithm. For measuring the difference between a set of $\mathbf{s}$ values and a set of $\bar{\mathbf{s}}$ values, two types of metrics were used. First, the Mean Square Error (MSE) [45] was calculated, to measure the average of the differences between each pair of $\mathbf{s}$ and $\bar{\mathbf{s}}$. However, it does not consider the discrepancy between the rankings

of the two sets. Thus, a second type of measures, including Spearman's Rank Correlation Coefficient (SRCC) and Kendall's Rank Correlation Coefficient (KRCC), was used. Also, we utilised Pearson's Correlation Coefficient (PCC) used by Gao *et al*. [16] for comparison.

4.3 Implementation Details

We used the pre-trained Conformer [33] model as the backbone of the SC-MSBP, whose gradients were updated during the training stage. The Stochastic Gradient Descent (SGD) optimiser was used. The learning rate and momentum were set to $2e^{-3}$ and 0.9 respectively. We set the size of mini-batches to 16. The network was trained for 30 epochs. We performed the experiment on a single Nvidia Geforce RTX 3090 graphics card. Typically, the training process took around 10 hours. Specifically, we first normalised each image using the mean and standard deviation of the colours used by the pre-trained model. Then, each image was resized to the resolution of 224×224 pixels. Two images $\mathbf{x}$ and $\mathbf{y}$ along with the associated similarity score $\mathbf{s}$ were fed into the siamese network for training. The goal is to minimise the difference between the values of $\bar{\mathbf{s}}$ and $\mathbf{s}$ using an optimiser. In the test stage, each pair of test images were sent to the model that we trained and the output was the similarity score between them. All scores were comprised of a similarity matrix in terms of the test subset.

5 Experimental Results

We conducted the experiment using the setup described in Section 4. The results will be reported in this section.

5.1 Results Obtained Using Pertex

We first applied the proposed method to the *Pertex* [8] data set. The results were compared with those derived using the combinations of four texture descriptors and Random Forests [2] or Autoencoder [41], PTSNet [16] and DISTS [12]. The results are displayed in Table 1. It can be seen that our method achieved the better performance than the other approaches. We also plotted the three similarity matrices obtained using the PTSNet [16], both free-grouping [8] and Isomap dimensionality reduction [39] and our method respectively in Figure 4. Each matrix $\mathbf{M}$ is a 34×34 symmetric matrix. $\mathbf{M}(\mathbf{i},\mathbf{j})$ denotes the similarity between texture $\mathbf{i}$ and $\mathbf{j}$. The SRCC value computed between the similarity matrix that we obtained and the ground-truth was 0.8783, which was much higher than the value of 0.7188 calculated between the similarity matrix derived using PTSNet [16] and the ground-truth.

5.2 Results Obtained Using PTD

We then tested the proposed method using the *PTD* [25] data set. The same baselines were used as those used in Section 5.1. Table 1 shows the results obtained. Again, our method outperformed the baselines with large margins. In particular, the PCC or SRCC values reached to around 0.98, which suggusted a very high correlation between the similarity matrices obtained using the proposed method and human perception.

Method		Pertex [8] MSE	Pertex [8] PCC	Pertex [8] SRCC	Pertex [8] KLCC	PTD [25] MSE	PTD [25] PCC	PTD [25] SRCC	PTD [25] KLCC
Random Forests [8]	LBP [33]	0.025 [16]	0.5430 [16]	-	-	0.012 [16]	0.8272 [16]	-	-
	Gabor [28]	0.021 [16]	0.6890 [16]	-	-	0.010 [16]	0.8657 [16]	-	-
	PCANet-48D [8]	0.026 [16]	0.6259 [16]	-	-	0.016 [16]	0.8044 [16]	-	-
	CNN-48D [23]	0.024 [16]	0.6171 [16]	-	-	0.017 [16]	0.8048 [16]	-	-
Auto Encoder [43]	LBP [33]	0.030 [16]	0.3564 [16]	-	-	0.022 [16]	0.6113 [16]	-	-
	Gabor [28]	0.019 [16]	0.6696 [16]	-	-	0.012 [16]	0.8077 [16]	-	-
	PCANet-48D [8]	0.032 [16]	0.4687 [16]	-	-	0.013 [16]	0.7915 [16]	-	-
	CNN-48D [23]	0.019 [16]	0.7037 [16]	-	-	0.010 [16]	0.8560 [16]	-	-
PDLF-PTSNet [16]		0.0130 [16]	0.7805 [16]	-	-	0.0040	0.9402	-	-
PDLF-PTSNet* [16]		0.0151	0.7949	0.7188	0.5412	0.0032	0.9546	0.9388	0.7980
DISTS** [24]		0.0100	0.8473	0.7949	0.6048	0.0049	0.9292	0.9005	0.7327
Ours		**0.0056**	**0.9171**	**0.8783**	**0.7076**	**0.0017**	**0.9763**	**0.9628**	**0.9376**

* This model was reproduced following the work of Gao *et al*. [16]. The results shown here were selected in terms of the highest PCC value. ** Due to the unpublished training source code and the limitation on the number of data sets, we were unable to reproduce the second loss of DISTS.

Table 1: Comparison of different algorithms tested on the *Pertex* [8] and *PTD* [25] data sets.

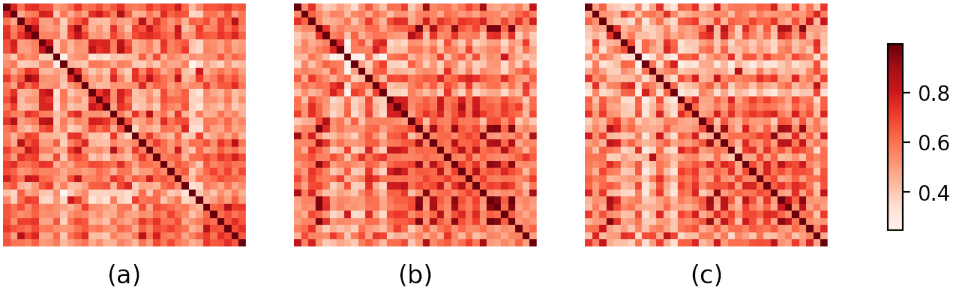

Figure 4: Visualisation of three similarity matrices: (a) the one predicted using the method that Gao *et al*. [16] proposed, (b) the ground-truth derived using a free-grouping experiment [8] and Isomap dimensionality reduction [39] and (c) the one predicted using our method. Compared with the matrix shown in (b), the SRCC values calculated using (a) and (c) are 0.7188 and 0.8783, respectively.

5.3 Effect of the Backbone and Network Modules

Regarding the backbone of the SC-MSBP, we tested three pre-trained models using *Pertex* [8] for comparison. The results are shown in Table 2. As can be seen, the Conformer [33] outperformed its two counterparts. Although the CNN branch of the Conformer has the same structure as that of the ResNet-101 [19], the former produced the better results than the latter. This gain should be due to the complementary relationship between the local and global features extracted in the CNN and Transformer branches of the Conformer.

We also examined the effect of the modules in the SC-MSBP and MLN, including bilinear pooling and Swin Transformer blocks, by replacing them using bilinear interpolation and three convolutional layers, respectively. The results reported in Table 3 show the advantages of using both the modules.

Backbone	MSE	PCC	SRCC	KLCC
VGG-VD-16 [35]	0.0075	0.8834	0.8535	0.6736
ResNet-101 [19]	0.0068	0.9045	0.8540	0.6760
Conformer [33]	**0.0056**	**0.9171**	**0.8783**	**0.7076**

Table 2: Comparison of different backbones on the *Pertex* [8] data set.

Module	MSE	PCC	SRCC	KRCC
-BP [24]	0.0095	0.8785	0.8366	0.6518
-Swin [26]	0.0072	0.8919	0.8627	0.6856
+BP +Swin	**0.0056**	**0.9171**	**0.8783**	**0.7076**

Table 3: Effect of the modules in the SC-MSBP and MLN tested on the *Pertex* [8] data set.

5.4 Texture Retrieval Experiment

We further performed a texture retrieval experiment using the *Pertex* data set [8]. For simplicity, only the best model that we trained was used. As we mentioned in Section 4.1, 34 test textures were randomly selected from the *Pertex* data set. Each test texture was used as a Query texture while the top 10 most similar textures to this texture were retrieved in the descending order with regard to the similarity. Figures 5 (a) and (b) present two sets of ranked textures retrieved using our best model and human observers, respectively, in terms of the Query texture. It is suggested that the great consistency between the visual perception of humans and our method on fine-grained texture similarity has been achieved.

5.5 Performance Analysis

Besides, we computed the computational complexity and the number of parameters for each model using THOP. The two values computed for the two models proposed in [17] and [16] and ours are (30.720GMac, 14.715M), (78.093GMac, 20.178M) and (26.430GMac, 79.018M) in turn. Given the three models trained using the methods proposed in this paper, [16] and [17] were used, it took approximately 0.09, 0.0086 and 0.0071 seconds, respectively, to run inference on a pair of 224×224 images. The above results are reported in Table 4.

Model	#Param	Comput. Complex. (GMac)	Inference Cost (S)
PDLF-PTSNet [16]	20.178M	78.093	0.0071
DISTS [17]	14.715M	30.720	0.0086
Ours	79.018M	26.430	0.0900

Table 4: Comparison of three state-of-the-art models with regard to the number of parameters, computational complexity and inference cost.

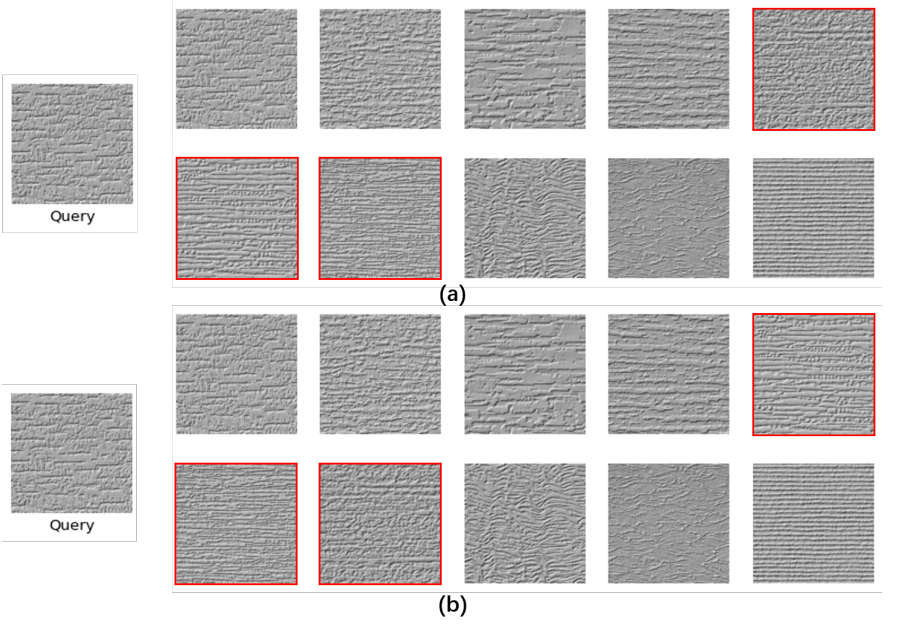

Figure 5: Two sets of ranked textures retrieved using (a) our best model and (b) human observers, respectively, in terms of the Query texture. Here, red boxes indicate the difference between the two ranked lists.

6 Conclusion

In this paper, we addressed the problem with the inconsistency between the predictions of fine-grained texture similarity by humans and algorithms. To this end, we introduced a new Perceptually Motivated Texture Similarity Prediction Network (PMTSPN). This network contains a siamese Conformer with multi-scale bilinear pooling (SC-MSBP) and a metric learning network (MLN). The two subnetworks were used to learn a texture representation which captured the Higher Order Statics (HOS) in different spatial extents, and a similarity metric from the features which encoded the short-range, long-range and lateral interactions, respectively. To our knowledge, they have not been explored for texture similarity tasks. Experimental results showed that the PMTSPN outperformed its counterparts with large margins. This promising performance should be attributed to the capability that our method learns both the powerful texture representation and the effective similarity metric.

Acknowledgements

This study was in part supported by the National Natural Science Foundation of China (NSFC) (No. 42176196) and was in part supported by the Young Taishan Scholars Program (No. tsqn201909060).

References

[1] A.C. Bovik, M. Clark, and W.S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1): 55–73, 1990. doi: 10.1109/34.41384.

[2] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, oct 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

[3] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] Tsung-Han Chan, Kui Jia, Shenghua Gao, Jiwen Lu, Zinan Zeng, and Yi Ma. Pcanet: A simple deep learning baseline for image classification? *IEEE transactions on image processing*, 24(12):5017–5032, 2015.

[6] Long Chen and Yuhang He. Dress fashionably: Learn fashion collocation with deep mixed-category metric learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[7] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, and Andrea Vedaldi. Deep filter banks for texture recognition, description, and segmentation. *International Journal of Computer Vision*, 118(1):65, 2016.

[8] Alasdair DF Clarke, Fraser Halley, Andrew J Newell, Lewis D Griffin, and Mike J Chantler. Perceptual similarity: A texture challenge. In *BMVC*, pages 1–10. Citeseer, 2011.

[9] Alasdair DF Clarke, Xinghui Dong, and Mike J Chantler. Does free-sorting provide a good estimate of visual similarity. *Predicting Perceptions*, pages 17–20, 2012.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Comparison of full-reference image quality models for optimization of image processing systems. *International Journal of Computer Vision*, 129(4):1258–1281, 2021.

[12] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2567–2581, 2022. doi: 10.1109/TPAMI.2020.3045810.

[13] Xinghui Dong, Junyu Dong, and Mike J. Chantler. Perceptual texture similarity estimation: An evaluation of computational features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7):2429–2448, 2021. doi: 10.1109/TPAMI.2020.2964533.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[15] Yueqi Duan, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Deep localized metric learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2644–2656, 2017.

[16] Ying Gao, Yanhai Gan, Lin Qi, Huiyu Zhou, Xinghui Dong, and Junyu Dong. A perception-inspired deep learning framework for predicting perceptual texture similarity. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(10):3714–3726, 2020. doi: 10.1109/TCSVT.2019.2944569.

[17] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2:1735–1742, 2006.

[18] Robert M. Haralick, K. Shanmugam, and Its'Hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, 1973. doi: 10.1109/TSMC.1973.4309314.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[20] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pages 84–92. Springer, 2015.

[21] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11 (9):1066, 2019.

[22] Jan J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[24] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnns for fine-grained visual recognition. In *Transactions of Pattern Analysis and Machine Intelligence (PAMI)*, 2017.

[25] Jun Liu, Junyu Dong, Xiaoxu Cai, Lin Qi, and Mike J. Chantler. Visual perception of procedural textures: Identifying perceptual dimensions and predicting generation models. *PLoS ONE*, 10, 2015.

[26] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021. doi: 10.1109/ICCV48922.2021.00986.

[27] D. G. Low. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.

[28] Bangalore S Manjunath and Wei-Ying Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on pattern analysis and machine intelligence*, 18(8): 837–842, 1996.

[29] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996. doi: 10.1109/34.531803.

[30] Jianchang Mao and Anil K Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern recognition*, 25(2):173–188, 1992.

[31] Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. Deep metric learning via facility location. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5382–5390, 2017.

[32] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

[33] Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 367–376, 2021.

[34] Oren Rippel, Manohar Paluri, Piotr Dollar, and Lubomir Bourdev. Metric learning with adaptive density discrimination. *arXiv preprint arXiv:1511.05939*, 2015.

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale visual recognition, 2015. Proc. International Conference on Learning Representations.

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[37] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. *CoRR*, abs/1511.06452, 2015. URL http://arxiv.org/abs/1511.06452.

[38] Lothar Spillmann and John S. Werner. Long-range interactions in visual perception. *Trends in Neurosciences*, 19(10):428–434, 1996. ISSN 0166-2236. doi: https://doi.org/10.1016/0166-2236(96)10038-2. URL https://www.sciencedirect.com/science/article/pii/0166223696100382.

[39] Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[41] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(110):3371–3408, 2010. URL http://jmlr.org/papers/v11/vincent10a.html.

[42] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[43] Wikipedia contributors. Euclidean distance — Wikipedia, the free encyclopedia, 2022. URL https://en.wikipedia.org/w/index.php?title=Euclidean_distance&oldid=1093854727. [Online; accessed 24-July-2022].

[44] Wikipedia contributors. Mahalanobis distance — Wikipedia, the free encyclopedia, 2022. URL https://en.wikipedia.org/w/index.php?title=Mahalanobis_distance&oldid=1094197569. [Online; accessed 24-July-2022].

[45] Wikipedia contributors. Mean squared error — Wikipedia, the free encyclopedia, 2022. URL https://en.wikipedia.org/w/index.php?title=Mean_squared_error&oldid=1097028555. [Online; accessed 18-July-2022].

[46] Martin Wilson. *Retinal Lateral Interactions*, pages 3513–3517. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-540-29678-2. doi: 10.1007/978-3-540-29678-2_5108. URL https://doi.org/10.1007/978-3-540-29678-2_5108.