

Learning Perceptual Texture Similarity and Relative Attributes from Computational Features

Jianwen Lou*, Lin Qi*, Junyu Dong*, Hui Yu† and Guoqiang Zhong*

*College of Information Science and Engineering

Ocean University of China, Qingdao, China

Email: dongjunyu@ouc.edu.cn

†School of Creative Technologies

University of Portsmouth, London, UK

Abstract—Previous work has shown that perceptual texture similarity and relative attributes cannot be well described by computational features. In this paper, we propose to predict human’s visual perception of texture images by learning a non-linear mapping from computational feature space to perceptual space. Hand-crafted features and deep features, which were successfully applied in texture classification tasks, were extracted and used to train Random Forest and rankSVM models against perceptual data from psychophysical experiments. Three texture datasets were used to test our proposed method and the experiments show that the predictions of such learnt models are in high correlation with human’s results.

I. INTRODUCTION

Texture is an important visual cue which is known to be used in numerous vision tasks, such as material recognition [1], object detection [2] and many other aspects in image understanding. Although human’s visual perception of textures has not been entirely understood, some pioneer work have been done in perceptual similarity and relative attributes [3]. Perceptual texture similarity concerns the perceived likeness of two textures, e.g. ‘texture A and B is more similar than texture C and D’. Relative attributes concern the perceived strength of semantic attributes in a texture image with respect to other texture images, e.g. ‘texture A looks more spotted than texture B’. Accurate predictions of the fore-mentioned texture perceptions can benefit many computer vision tasks. For example, an ideal content-based texture retrieval system is required to compare the query texture with those in the database in a manner which is consistent with human’s perception of texture similarity. A texture annotation system should interpret textures with semantic attributes that are in consistency with human’s perception.

Perceptual data of textures were collected by psychophysical experiments, using methods like ‘free grouping’, ‘perceptual features rating’ and ‘pairwise comparison’ [4]–[7]. The experimental results are analysed to get perceptual similarity matrix and relative attributes. Perceptual texture space(PTS) is constructed with dimensions describing human’s perception [5] [6]. Those perceptual spaces are shown to accommodate a metric that can effectively measure perceived similarity between textures. Meanwhile, each dimension in PTS seems to correspond with some specific perceptual features, such as repetitiveness, directionality and coarseness [5].

Computational features have been successfully used in texture classification, and this motivates researchers to use computational features to predict human’s perceptual similarity and relative attributes. However, it has been found that the mapping from feature space to the perceptual space is not linear [8] [9]. Previous approaches established a two-stage mapping: constructing a low-dimensional embedding from psychophysical data, and then mapping computational features to each dimension in the built perceptual space [10]–[14]. Whereas, in this paper, we learn direct mappings from computational features to psychophysical data. Besides, using hand-crafted features, we also investigated deep features extracted from convolutional neural networks (CNN) which is the state-of-the-art approach in speech recognition, visual object recognition and many other domains.

We utilize Random Forest [15] and rankSVM [16] to build the mapping from computational feature space to perceptual space. We tested our method on three texture datasets and the experiments show that the prediction of the learnt models are in high correlation with human’s results.

II. RELATED WORK

A. Psychophysical Data Acquisition

Psychophysical experiments are designed to identify high-level texture features perceived by humans. Tamura et al. computed similarity between textures based on subjects’ ratings on given visual properties [4]. Because of the between-subject variation and the individual scale variation of rating, the generated similarity is not convincing. Rao and Lohse [5] used ‘Free Grouping’, in which subjects were asked to assign 56 textures into as many groups as they wanted without any instruction. The texture pair being grouped together by more subjects was considered to have higher similarity.

To enrich texture analysis with semantic data, Matthews et al. conducted pairwise comparisons [7]. For a certain attribute, the subjects were given a pair of texture in each trial, and chose the one that exhibits stronger magnitude. Rankings on each semantic attribute (relative attribute) were then derived.

B. Estimation of texture perceptions

Several studies attempted to utilise computational texture features to estimate perceptual similarity. In [8] [9], Euclidean

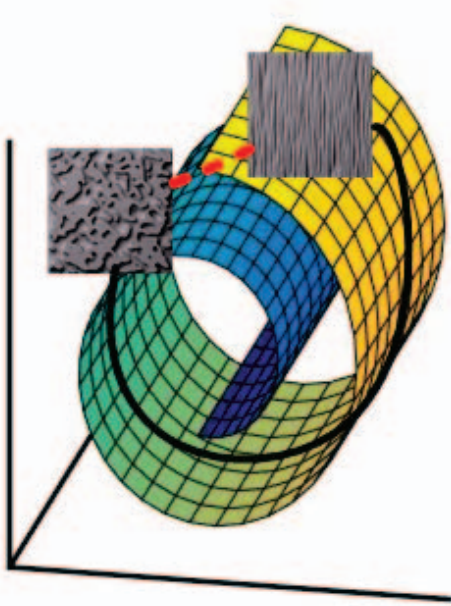


Fig. 1. A classical example of the manifold: the visually perceived textures lie on a “Swiss roll”, while the computational feature space is denoted by the Cartesian coordinates. The computational similarity based on Euclidean distance in the computational feature space (length of the red dashed line) can’t describe the perceptual similarity based on the geodesic manifold distance (length of the black solid curve).

distance or Chi-square(χ^2) distance [17] between computational features was regarded as the prediction of perceptual similarity. However, those distance measures do not correlate well with human’s data.

To enable computational features to describe texture perceptions, many efforts have been made on building mappings from computational feature space to perceptual space. Subspace transformation techniques such as multidimensional scaling(MDS) or isometric mapping algorithm(Isomap) were first applied to construct a perceptual texture space underlying the perceptual similarity matrix [10]–[14] [18] [19]. Then, support vector machines(SVM) or neural networks were used to construct a mapping from computational features to perceptual space [11]–[14]. Such method was reported to improve the perceptual consistency of computational features. However, it involves two-stages of mapping. Especially, in the second stage of mapping, one regression is needed for each dimension in the perceptual space which is supposed to have relatively high complexities. We cannot ensure if there exists loss of information which is essential to the underlying perceptual space during the transformation.

A similar approach [7] has been applied on estimating another kind of texture perception - relative attributes. Some previous work focused on attributes as binary predicates indicating the presence of a certain property in a texture(e.g., a texture is ‘blemished’ or not) [20] [21]. Apparently, relative attribute can capture more general semantic relationships between textures(e.g., texture A is more ‘marble’ than texture B). In [7], rankSVM [16] was used to map from hand-crafted

texture features to relative attribute. It demonstrated that no satisfying correspondence between the textures’ feature space and the perceptual space was built.

C. Computational texture features

Low-level features, such as Gabor features [22] and local binary patterns(LBP) [23], have been successfully applied in texture classification tasks. Unsupervised features learned from training examples by deep neural networks(DNNs) have been shown to be comparable or outperform those hand-crafted features [24] [25]. Deep features are multiple levels of abstraction of the textures image, which is expected to represent more abstract semantics [26].

The most popular deep learning model used in computer vision is convolutional neural network(CNN), which generally composes of three layer types - a convolutional filter bank layer, a nonlinear processing layer and a feature pooling layer. Many variations of CNN have been proposed for different vision tasks, we chose a pre-trained CNN on the ImageNet ILSVRC 2012 [27] and the PCANet [26] in our experiment. The first one consists of five convolutional layers, 60 million parameters and 650,000 neurons. The PCANet comprises only a cascaded PCA, followed by a nonlinear output stage. Both of the two networks brought breakthroughs in various vision tasks, such as image classification and face recognition.

III. LEARNING TEXTURE PERCEPTIONS FROM COMPUTATIONAL FEATURES

A. Perceptual texture space

A perceptual texture space should preserve perceived similarity in terms of distances. It is reported that distance measures in the computational feature space don’t correlate well with perceptual similarity [8]. The perceptual texture space is a manifold which can not be described directly by the computational feature. Fig. 1 illustrates a classical example of a manifold, in which textures lie on a two-dimensional “Swiss roll”. Two textures far apart from each other on a underlying manifold may become close in the three-dimensional feature space measured by their Euclidean distance. Subspace transformation techniques such as multidimensional scaling(MDS) and isometric mapping(Isomap) has been applied to learn a low-dimensional embedding from the perceptual similarity matrix [5] [10]–[14] [18] [19]. In this paper, we propose to learn a direct mapping from computational features to texture perceptions with non-linear regression methods.

B. Texture datasets

Primarily, the dataset is required to be available with human data obtained from psychophysical experiments. To offer more reliable results, the dataset should also contain adequate textures with various appearances. Accordingly, we chose the following public datasets:

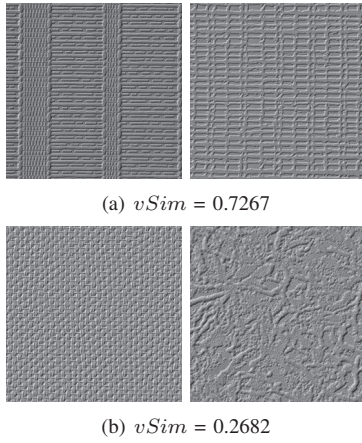


Fig. 2. Two texture pairs of Pertex with their perceptual similarity values - $vSim$.

1) *Pertex* [28]: contains 334 textures, including examples of embossed vinyl, woven wall coverings, carpets, rugs, window blinds, soft fabrics, building materials, product packagings, etc. The perceptual similarity matrix was obtained through a grouping experiment with 14 participants. The similarity value ($vSim$) of each texture pair was calculated by dividing the number of times the textures have been grouped into the same subset with the total number of participants. The resulting similarity matrix is a 334×334 symmetric matrix whose element value varies from 0 to 1. However, the initial similarity matrix contains many zero entries. To better examine our approach, we chose the non-sparse 8D Isomap similarity matrix, which has been validated using human pair-of-pairs judgements [29]. Two texture pairs of the dataset along with their perceptual similarity values are shown in Fig. 2.

2) *Procedural texture dataset* [10]: comprises of 450 textures generated by 23 procedural texture models. By adjusting parameters of each model, numerous textures covering a wide range of appearances were first generated. Then, texture samples looking non-general were filtered out. The authors also conducted a grouping experiment which was similar to the fore-mentioned one. The main improvement was that during the experiment, all participants should offer an extra confidence score for each action. Such a strategy well solved the sparsity problem of the perceptual similarity matrix. Fig. 3 shows two texture pairs of the dataset along with their perceptual similarity values.

3) *dataset proposed in [7]*: consists of 319 texture classes selected from Outex [30]. Adopting three kinds of illuminants (horizon, inca, t184) and four rotation angles (0° , 30° , 60° , 90°), each texture class contains twelve samples. 7,689 pairwise comparisons describing relative attributes were obtained from ten subjects along with the paper authors according to eleven attributes, including blemished, bumpy, lined, marbled, random, repetitive, speckled, spiralled, webbed, woven, and wrinkled. Fig. 4 illustrates two pairs of textures of the dataset with their relative attributes.

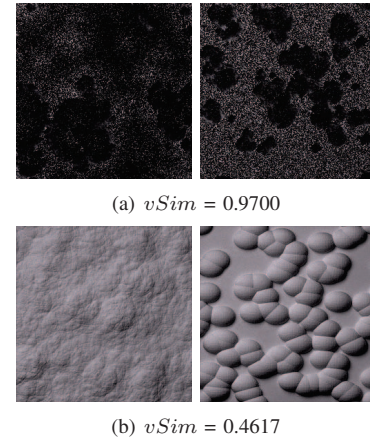


Fig. 3. Two texture pairs of Procedural texture dataset with their perceptual similarity values - $vSim$.

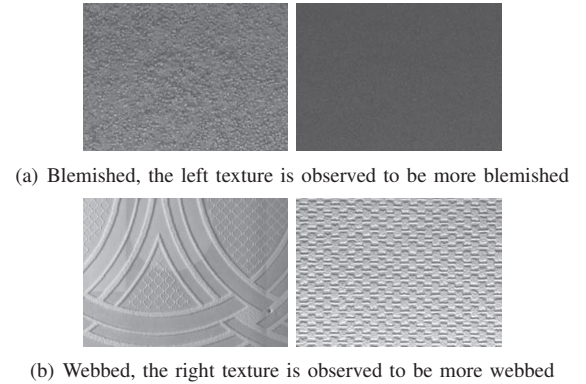


Fig. 4. Two texture pairs of [7] along with their relative attributes.

C. Computational texture features

Two hand-crafted features and two deep features were chosen. A pre-trained convolutional neural network on the ImageNet ILSVRC 2012 (will be abbreviated as CNN in the rest part of this paper) and the PCANet [26] with a simple structure were selected to extract deep features. The followings are some details of the features:

- **Gabor:** Each 48-dimensional feature vector comprises the means and standard deviations of the Gabor wavelet responses for 24 orientation and scale combinations given in [22].
- **Local Binary Patterns (LBP):** Each 36-bin histogram was generated by counting the binary patterns for the 8-pixel neighbour in a circle of radius 2 [23].
- **CNN:** The implementation of the convolutional neural network suggested in [31] was adopted, which output a 4096-dimensional feature vector for each texture image.
- **PCANet:** The two-stage PCANet was applied. The patch size was set to 7×7 and the number of filters for each layer was set to 8. The size of each block for local histogram was set as the quarter of the input texture's size. The network finally output a 32,768-dimensional feature

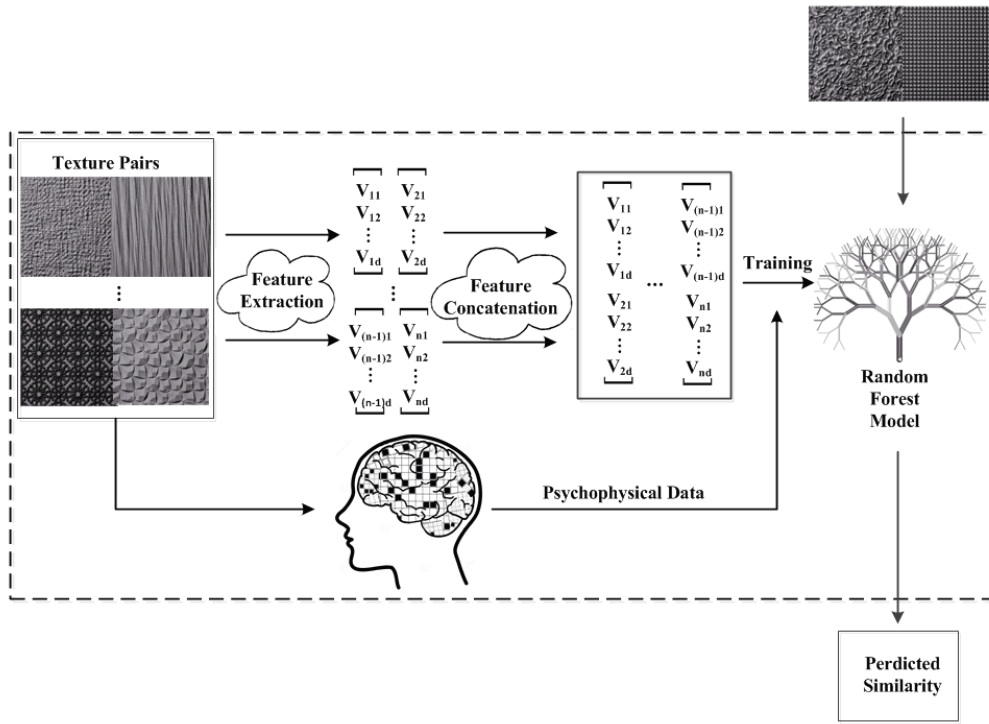


Fig. 5. Illustration of how the proposed method to predict perceptual texture similarity.

vector for each texture image.

D. Predicting perceptual texture similarity

The proposed approach to predict perceptual texture similarity is illustrated in Fig. 5. For a texture pair, the computational feature of each texture is extracted first. The computational features of the texture pair are further concatenated into one feature vector. Random Forest is applied to learn the hidden mapping between the concatenated feature vector and the perceptual similarities obtained from psychophysical experiments. At last, with the trained model, we use computational texture features to predict texture similarity.

1) *Experiments*: For each texture dataset, the set of concatenated feature vectors was used to train and test the Random Forest model. The perceptual similarity matrix obtained through psychophysical experiments was regarded as the ground truth. We partitioned the dataset into two sets of 80% for training and 20% for testing. Such a procedure was repeated five times for each combination of feature and dataset, and the model that output the best results was retained.

2) *Analysis*: Deviation and Spearman’s rank correlation coefficient were used to assess our approach. In this paper, deviation is defined as the average of the absolute difference between each predicted similarity and the psychophysical data. The correlation coefficient, $-1 \leq \rho \leq 1$, measures the correspondence between the predicted value and the psychophysical data, where $\rho = 1$ indicates a perfect monotonic relationship and $\rho = -1$ indicates a perfect inverse monotonic relationship.

We first calculated distance between computational features of each texture pair and compared them with the psychophysical data. Euclidean distance was applied for Gabor and CNN, while Chi-square(χ^2) static [17] was applied for LBP and PCANet. The Spearman’s rank correlation coefficient between the distances and the perceptual similarity was calculated and shown in Table I. As the results show, the largest correlation coefficient occurs as 0.4766 for CNN on procedural texture dataset. Thus, for both hand-crafted features and deep features, the computational similarity derived from distance measures has a weak correlation with the psychophysical data. It further demonstrated that the perceptual texture space couldn’t be described by the computational texture space.

Because of the high dimensionality of the original deep features, we applied Principal Component Analysis(PCA) algorithm to reduce the original features into a lower-dimensional space. The residual variance was used to determine the appropriate dimensionality of the feature. Fig. 6 plots the relationship between the residual variance and the dimensionality of deep features extracted from each dataset. It shows that the residual variance decreases gradually with added dimension and ceases to decrease significantly when the dimensionality increases to a certain value. The inherent dimensionality of features can be estimated from the curve’s “elbow”. For all curves, the approximate “elbow” appears when the dimension comes to 25. To find a relatively better low-dimensional space, the dimensionality was allowed to vary from 15 to 35 with the step of 5, the optimal value of which was then selected through

TABLE I
SPEARMAN’S RANK CORRELATION COEFFICIENTS BETWEEN THE DISTANCE OF COMPUTATIONAL FEATURES AND THE PSYCHOPHYSICAL DATA.

	Gabor	LBP	PCANet	CNN
Pertex	0.1160	0.0872	0.1265	0.4520
Procedural texture dataset	0.3370	0.2400	0.2982	0.4766

TABLE II
SPEARMAN’S RANK CORRELATION COEFFICIENTS BETWEEN PREDICTED SIMILARITIES AND THE PSYCHOPHYSICAL DATA FOR EACH REGRESSION MODEL.

	Gabor	LBP	PCANet	CNN
Pertex	0.9527	0.9046	0.9298	0.9563
Procedural texture dataset	0.9850	0.9808	0.9776	0.9792

TABLE III
DEVIATIONS BETWEEN PREDICTED SIMILARITIES AND THE PSYCHOPHYSICAL DATA FOR EACH REGRESSION MODEL.

	Gabor	LBP	PCANet	CNN
Pertex	0.0384	0.0531	0.0475	0.0360
Procedural texture dataset	0.0161	0.0192	0.0221	0.0207

the testing procedure.

Table II and III list the performance measurements of regression models trained for different computational texture features and datasets. As the Table II shows, the correlation coefficient varies from 0.9298 to 0.9563 on Pertex, while it varies from 0.9776 to 0.9850 on Procedural texture dataset. This demonstrates that on both datasets, all selected computational features have predicted texture similarities which have quite strong correlation with the psychophysical data. Meanwhile, the Table III shows that on both datasets, the predicted texture similarities also have very small deviations from the psychophysical data with the average deviation less than 0.032. It further indicates that the predicted texture similarity is well consistent with the human perceived one. For each dataset, we also chose and visualised the trained regression model with the best performance. For those models, we plotted the curve of the relationship between the predicted texture similarity and the psychophysical data, which is shown in Fig. 7. It can be seen from the figure that the predicted texture similarity is close to the psychophysical data. Consequently, the proposed approach is shown to predict texture similarities complying with human’s perception.

Additionally, the Gabor achieves the best performance on Procedural texture dataset with the highest correlation coefficient - 0.9850 and the lowest deviation - 0.0161. The reduced CNN performed best on Pertex with the highest correlation coefficient - 0.9563 and the lowest deviation - 0.0360. It is interesting that the reduced deep features are still on par with or even outperform hand-crafted features.

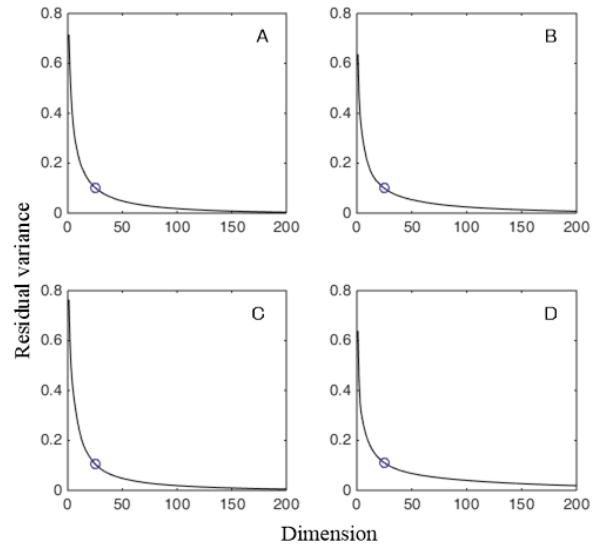


Fig. 6. Residual variances of deep features extracted from Pertex and Procedural texture dataset when conducting dimensionality reduction. The intrinsic dimensionality of the deep feature can be estimated by looking for the “elbow” at which the curve ceases to decrease significantly with added dimensions. Blue circles mark the “elbow” of the curves. (A) Pertex - CNN. (B) Pertex - PCANet. (C) Procedural texture dataset - CNN. (D) Procedural texture dataset - PCANet

E. Predicting relative attributes

Relative attributes can offer us rich semantic information of textures. Although several works have been done to bridge the gap between computational texture features and the semantic texture space, no satisfying results have been obtained. We decided to apply the same learning algorithm in [7] to discover correspondences existed between textures’ deep feature space and their relative attributes.

1) *Learning relative attributes*: Attributes are properties observable in images that have human-designed names (e.g., ‘striped’, ‘bumpy’). The relative attribute indicates the strength of an attribute in an image with respect to other images [3]. Such properties can also be used to describe textures, which is useful in tasks, such as texture annotation.

With series of pairwise comparison conducted by humans, we can get ordering of textures based on the strength of a specific attribute presented in them. Then, a linear ranking function that enforces such a desired ordering can be learnt with the rankSVM method, which is used to map computational texture features to their relative attributes [3] [16].

2) *Experiments*: In [7], the authors compared a number of existing texture descriptors according to how well they could predict the relative attributes. In their experiments, five hand-crafted features were chosen, including co-occurrence matrices, Gabor, Liu noise-resistant features etc. Then they evaluated these texture descriptors in terms of how well they reflected the semantic information hidden in the relative attributes. For each combination of feature and attribute, they

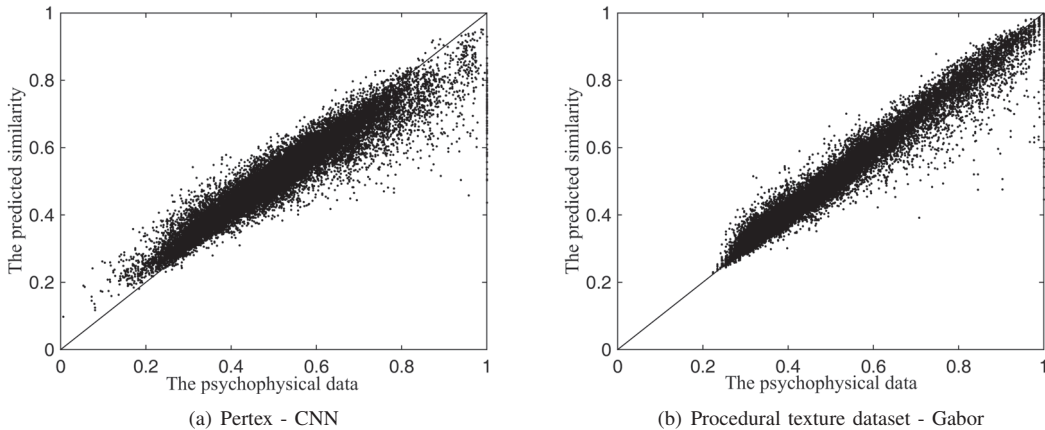


Fig. 7. The relationship between predicted similarities and the psychophysical data of the trained model with the best performance on each dataset. The black dot represents a texture pair. The solid line represents the ideal relationship, which is $y = x$.

TABLE IV
MISCLASSIFICATION RATES AND SPEARMAN’S RANK CORRELATION COEFFICIENTS.

Attribute	Misclassification rates			Spearman’s Rank correlation coefficients		
	Best result [7]	PCANet	CNN	Best result [7]	PCANet	CNN
Blemished	0.27	0.05	0.06	0.47	0.92	0.81
Bumpy	0.24	0.05	0.06	0.46	0.93	0.87
Lined	0.22	0.05	0.04	0.53	0.90	0.78
Marbled	0.17	0.02	0.03	0.48	0.93	0.86
Random	0.26	0.08	0.10	0.70	0.98	0.94
Repetitive	0.28	0.07	0.06	0.42	0.95	0.86
Speckled	0.21	0.03	0.04	0.44	0.91	0.85
Spiralled	0.18	0.04	0.02	0.26	0.76	0.54
Webbed	0.21	0.05	0.03	0.34	0.80	0.61
Woven	0.20	0.05	0.04	0.36	0.87	0.75
Wrinkled	0.28	0.07	0.07	0.36	0.87	0.75

trained and tested a ranking function on the third dataset mentioned previously.

We conducted the same experiments in [7] by using two deep features introduced in the previous section. Without any process on those deep features, 11 ranking functions were learnt for each of them. These ranking functions were further tested in the hold-out set and generated rankings of the correspondent attribute. Misclassification rates and Spearman’s rank correlation coefficients for the predicted ranking and the ideal one (derived by psychophysical data) were also calculated for analysis.

3) *Analysis*: The results of our experiments are shown in Table IV. Compared to the best results in [7], the rankSVM learned a rather good mapping from textures’ deep features to their relative attributes with very small misclassification rates and high correlation coefficients. With the learned non-linear mapping, we can predict textures’ relative attributes very consistent with human perception.

The results also showed that the ability of computational features’ description varies from different attributes. For example, with a high correlation coefficient - 0.98, the PCANet

is shown to describe ‘Random’ perceived by human accurately. However, it is shown to describe ‘Spiralled’ not so well, with a relatively lower correlation coefficient - 0.76. Meanwhile, in this context, deep features were shown to correlate better with texture’s semantic space than hand-crafted features.

IV. CONCLUSION

Predicting texture perceptions using computational features can benefit image retrieval and understanding. In this paper, we propose a method learning a non-linear mapping from computational features to perceptual texture similarity and relative attributes. We tested both hand-crafted features and deep features, which were used to train regression models against perceptual data from psychophysical experiments. We tested the trained model on three datasets and the results show the predicted perceptions are in high correlation with human’s results.

To further assess our method, we plan to test it on more datasets available with psychophysical data. We will also leave as future work to learn the manifold with a perceptually

meaningful structure of textures. Testing more deep features in predicting perceptual texture property is another future work.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China(NSFC)(No.61271405, 61401413, 61501417, 61403353), the Ph.D. Program Foundation of Ministry of Education of China(No. 20120132110018), China Post-doctoral Science Foundation(No.2014M551963), the Fundamental Research Funds for the Central Universities of China(No.201413019), the Open Project Program of the National Laboratory of Pattern Recognition(NLPR).

REFERENCES

- [1] L. Sharan, C. Liu, R. Rosenholtz, and E. H. Adelson, "Recognizing materials using perceptually inspired features," *International journal of computer vision*, vol. 103, no. 3, pp. 348–371, 2013.
- [2] M. Heikkilä and M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 4, pp. 657–662, 2006.
- [3] D. Parikh and K. Grauman, "Relative attributes," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 503–510.
- [4] H. Tamura, S. Mori, and T. Yamawaki, "Textural features corresponding to visual perception," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 8, no. 6, pp. 460–473, 1978.
- [5] R. Rao, G. L. Lohse *et al.*, "Towards a texture naming system: identifying relevant dimensions of texture," in *Visualization, 1993. Visualization'93, Proceedings., IEEE Conference on*. IEEE, 1993, pp. 220–227.
- [6] C. Heaps and S. Handel, "Similarity and features of natural textures." *Journal of Experimental Psychology: Human Perception and Performance*, vol. 25, no. 2, p. 299, 1999.
- [7] T. Matthews, M. S. Nixon, and M. Niranjana, "Enriching texture analysis with semantic data," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 1248–1255.
- [8] X. Dong, "Perceptual texture similarity estimation," Ph.D. dissertation, Heriot-Watt University, 2014.
- [9] A. D. Clarke, F. Halley, A. J. Newell, L. D. Griffin, and M. J. Chantler, "Perceptual similarity: A texture challenge," in *BMVC*, 2011, pp. 1–10.
- [10] 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, p. e0130335, 2015.
- [11] Y. Gao, L. Wang, K. L. Chan, and W.-Y. Yau, "Learning texture similarity with perceptual pairwise distance," 2005.
- [12] H. Long, W. K. Leow, F. K. Chua, and F. Kee, "Perceptual texture space for content-based image retrieval," in *Proc. Int. Conf. on Multimedia Modeling*. World Scientific, 2000, pp. 167–180.
- [13] H. Long and W. K. Leow, "Perceptual texture space improves perceptual consistency of computational features," in *International Joint Conference on Artificial Intelligence*, vol. 17, no. 1, 2001, pp. 1391–1396.
- [14] —, "A hybrid model for invariant and perceptual texture mapping," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 1. IEEE, 2002, pp. 135–138.
- [15] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [16] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 133–142.
- [17] W. H. Press, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.
- [18] R. Shenard, "The analysis of proximities: multidimensional scaling with an unknown distance function, ii," *Psychometrika*, vol. 27, pp. 219–246, 1962.
- [19] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [20] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 3606–3613.
- [21] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3828–3836.
- [22] B. S. Manjunath and W.-Y. Ma, "Texture features for browsing and retrieval of image data," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 8, pp. 837–842, 1996.
- [23] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.
- [24] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [25] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [26] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "Pcnet: A simple deep learning baseline for image classification?" *arXiv preprint arXiv:1404.3606*, 2014.
- [27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [28] F. Halley, "Perceptually relevant browsing environments for large texture databases," Ph.D. dissertation, Heriot-Watt University, 2012.
- [29] A. D. Clarke, X. Dong, and M. J. Chantler, "Does free-sorting provide a good estimate of visual similarity," *Predicting Perceptions*, pp. 17–20, 2012.
- [30] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, and S. Huovinen, "Outex-new framework for empirical evaluation of texture analysis algorithms," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 1. IEEE, 2002, pp. 701–706.
- [31] Y. Jia, "Caffe: An open source convolutional architecture for fast feature embedding," 2013.