

## CFS: an effective statistical texture descriptor

Alireza Akoushideh, Mohamadreza Modabernia

Department of Electronic Engineering, Technical and Vocational University (TVU), Iran

---

### Article Info

#### Article history:

Received Nov 9, 2019

Revised Jan 27, 2020

Accepted Feb 12, 2020

---

#### Keywords:

Coarseness

Feature extraction

Fineness

Texture classification

---

### ABSTRACT

Coarseness is an effective description for texture analysis and many approaches capture this property of texture in different methods. In this research, we propose a new texture descriptor (CFS) that works based on coarseness and the fineness similarity score. For tuning of its configuration, we collect coarse and fine textures based on human visual perception at first. After that, we relabel the "coarse" and "fine" categories of the gathered textures during the configuration of the operator in a proposed framework. We concatenated the features of two pyramid representations with the CFS information of the original texture (PCFS). In addition, we combine the PCFS information with our last proposed feature selection approach to improve the efficiency of the CFS. We evaluated the proposed feature extraction method with classification of one of the well-known data set, Outex. Experimental results depict satisfactory performance of the CFS with very low-dimensional feature length.

Copyright © 2020 Institute of Advanced Engineering and Science.  
All rights reserved.

---

### Corresponding Author:

Alireza Akoushideh,  
Department of Electronic Engineering,  
Technical and Vocational University (TVU), Iran.  
Email: akushide@tvu.ac.ir

---

## 1. INTRODUCTION

Texture is not a new subject in image processing, but it plays an important role in low-level analysis and explanation of an image in the machine vision field. Texture analysis is important in a wide range of applications such as image classification and segmentation [1], image retrieval [2], surface inspection [3], remote sensing [4], object recognition [5], environment modelling [6], satellite or aerial imagery [7] and motion detection [8], fingerprint spoof detection [9] and the nude image identification [10]. Regarding to different texture analysis categories [11], statistical methods are the most popular texture analysis categories. Statistical methods consider the distribution of gray values by computation of local features in the image and extract a set of statistics from the distributions of the local features. Depending on the number of pixels that define the local feature statistical methods, statistical methods can be classified into first-order, two-order, three-order statistics and more [12]. Textures are classified into two classes, namely, touch and visual. Touch textures relate to the touchable feel of a surface and range from the smoothest (little difference between high and low points) to the roughest (large difference between high and low points). Visual textures relate to the human image observation [13].

Coarseness and fineness are one of the most popular properties of the texture [14]. Some traditional methods for surveys have been introduced by Haralick [15], Chellappa et al. [16], Tuceryan and Jain [11], Tamura et al. [17], and Ojala et al. [18]. Our contribution in this paper is to focus on the coarseness and fineness scores of texture and using them as a texture descriptor. The proposed descriptor has very low feature dimensionality and will reduce the complexity of classifiers and improve performance. This descriptor uses Haralick's features and some statistical methods like standard deviation and skew. We use Outex [19] data sets to generalize and evaluate our results. The rest of this paper is organized

as follows. Section 2 describes related works on textural features extraction methods. Section 3 describes proposed approach. Section 4 presents the experimental results and Section 5 concludes the paper.

## 2. RELATED WORKS

One of the most widely used statistical methods is co-occurrence features, which has inspired a variety of modifications later on. Haralick et al. [20] have defined the co-occurrence matrix as a second order histogram statistics and it is one of the most popular statistical features for texture analysis. The extraction of the Haralick features is divided into two parts. First, is the calculation of gray level co-occurrence matrix (GLCM) and the second is the extraction of features from the GLCM. The GLCM matrix defines the probability of joining two pixels  $P_{d,\theta}(i,j)$  that have values  $i$  and  $j$  with distance  $d$  and  $\theta$  as an orientation angular. Each element of the GLCM is the number of occurrences of the pair of pixels with value  $i$  and  $j$  which is at distance  $d$  [21-23]. In other words, the GLCM is a square matrix with  $N_g$  dimension, where  $N_g$  is the number of gray levels in the image. More details about the calculation of GLCM are in [15, 20, 24]. Tamura et al. [17], considered visual perception of texture features includes coarseness, contrast, directionality, line-likeness, regularity, and roughness to describe various texture properties. Tamura properties are very meaningful in context to texture feature extraction. Some research on content-based image retrieval (CBIR) depicts that the Tamura's features have better performance than other methods [25, 26]. In addition, some researches compare performance of different coarseness features in different challenges like noise, illumination, and scale. In recent researches, pyramid representation has been used for achieving an effective local binary patterns descriptor [27, 28], improving the scene categorization [29, 30] and image classifying by the object categories they contain [31]. We use this approach to improve the power of our proposed texture descriptor. In this research, we implemented the proposed feature extraction operators on two pyramid representations and concatenated them with the original texture as a new feature vector.

## 3. PROPOSED APPROACH

In this section, we propose a framework to find the coarse and soft textures. After that, a method to find the coarseness score (CNS) and fineness score (FNS) for a texture will be proposed. Finally, we use these features for texture classification in normal, pyramid, and combination with a feature selection approach.

### 3.1. Method of collecting coarse and fine textures

First, we collected 163 coarse and 81 fine textures from the internet that has been labelled coarse and fine. The textures are in different illumination and scale (form  $128 \times 128$  to  $3072 \times 2300$ ) based on human visual perception. After that, we split the textures to a non-overlapped area with resolution  $128 \times 128$ . As we know, fine textures have a less standard deviation ( $\sigma$ ) than coarse textures. For finding the pure coarse and fine texture, we pre-label the gathered textures based on their  $\sigma$ . After splitting and pre-labelling of the textures, the number of coarse and fine texture was 1892 and 1974, respectively. We organized Texta1, Texta2, Texta3, and Texta4 data sets with 500, 500, 409, 2457 none-overlapped textures with about 50% coarse, and 50% fine textures. Some samples of coarse and fine textures have been shown in Figure 1.

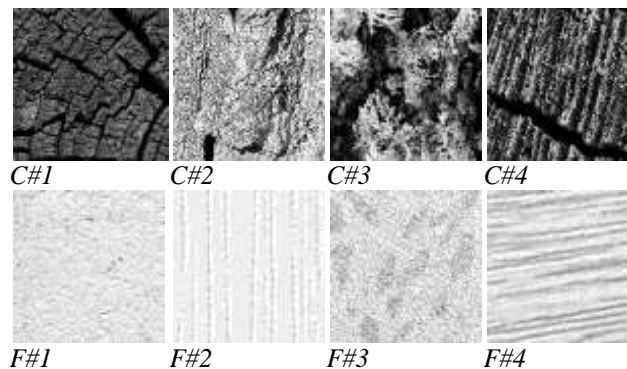


Figure 1. Some samples of coarse and fine textures, coarse and fine textures' labels started by C# and F#, respectively

**3.2. Voting fusing method [32]**

For unsupervised detection of coarse and fine texture, we implement the majority voting fusion that was used for the verification scenario [32]. For this purpose, we used four statistical features (Sigma), shown by (1), and thirteen Haralick’s features (HF) with three similarity measures, Czekannowski (Cze) [33], Chi-2 [34], and Euclidian distances (ED). Table 1 introduces the feature extraction method and similarity measures of the considered classifiers. As the ROC diagrams on Figure 2 shows, the classifiers have different verification performances under different similarity measures. For example, the Cze has the best performance with HF while, this measure with Sigma has the worst efficiency. Hence, we consider all three measures for verification.

$$\text{Sigma} = \langle \text{mean}(I), \text{std}(I), \text{skewness}(I), \text{turkitue}(I) \rangle \tag{1}$$

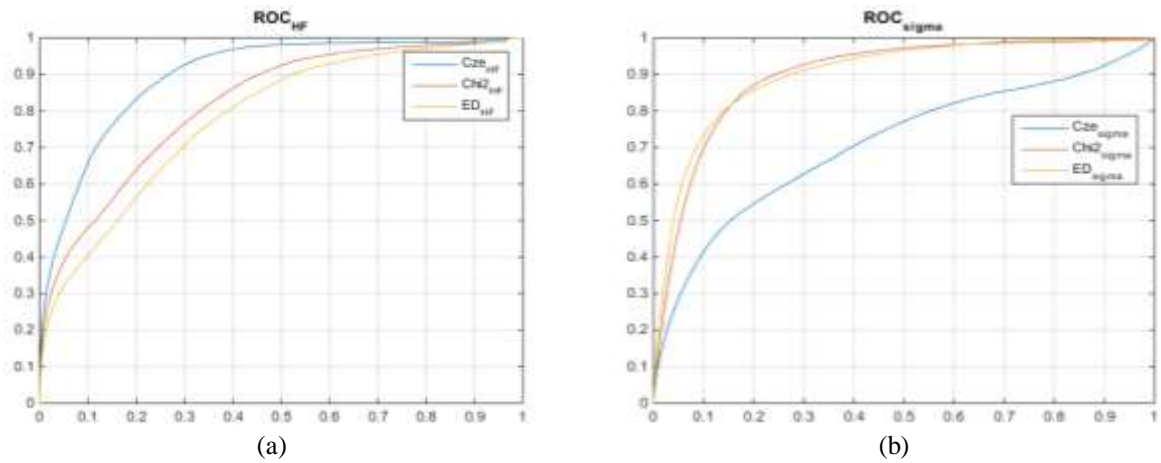


Figure 2. ROC diagrams calculated on texta1 (as a probe data) and texta2 (as a gallery data) under HF and sigma feature extraction methods.

As we can see, the diagrams depict different performances of the measures. (a) ROC diagram under Haralick features and three measures. (b) ROC diagram under Statistical features and three measures.

Table 1. Classifiers specifications

Classifier name	Feature extraction method	Similarity measure
C#1	Sigma	Cze
C#2	Sigma	Chi2
C#3	Sigma	ED
C#4	Haralick	Cze
C#5	Haralick	Chi2
C#6	Haralick	ED

For determining the similarity thresholds for all six classifiers, we considered Texta1 and galley2 data sets for a fixed false acceptance rate (FAR). For each pair of images in the data set, the matching score is determined. Coarse data considered as an imposter and fine data were labeled as genuine. If we plot the fraction of imposter scores larger than the threshold (that is the FAR) as a function of the threshold, we can determine the required similarity threshold for a certain required FAR. To determine the fused decision for the comparison of a probe to a reference image, the scores  $S_i$  for the classifier are compared to the similarity threshold of the classifiers ( $Thri$ ) and the decisions are accumulated by (2).

$$V = \sum_i^{All\ classifiers} \begin{cases} 1 & \text{if } S_i > Thri \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The number of votes is then regarded as a new score and is compared to decision threshold  $VT$ , shown by (3), to reach a decision:

$$D = \begin{cases} 1 & \text{if } V_i > V_T \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

The decision threshold ( $V_T$ ) must be determined using a second data set (Texta3) and again is tuned for a specific FAR. We call the FAR that is used to obtain the first set of similarity thresholds ( $Thri$ ) the *projected FAR* (pFAR). The optimal setting for pFAR can be different from the FAR required for the fused classifier. Optimum pFAR is obtained by minimum EER. Figure 3 Depicts a block diagram of finding the  $Thri$ .

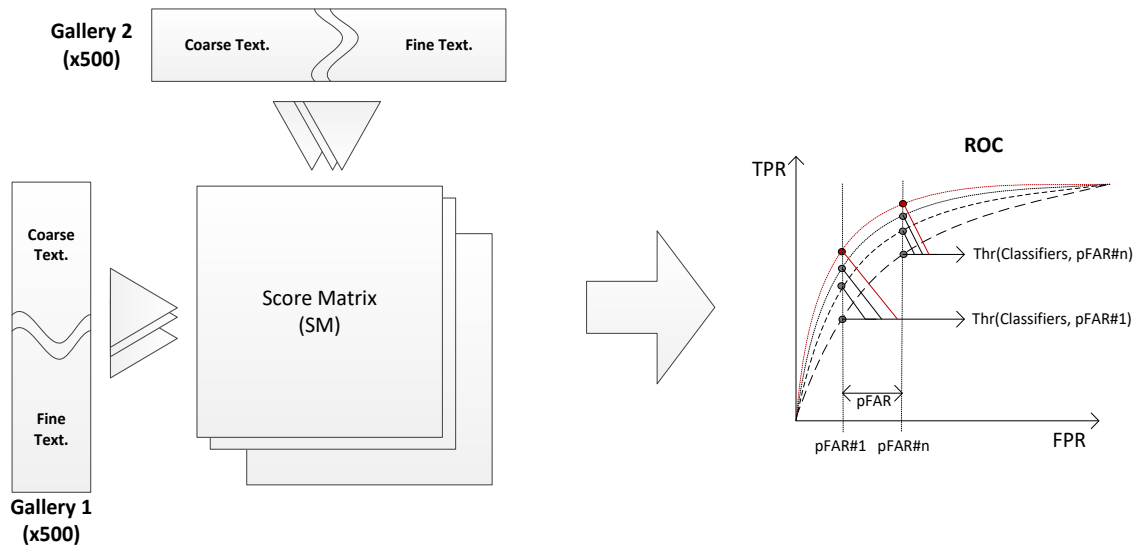


Figure 3. Diagram of finding  $Thri$  thresholds on pFAR

### 3.3. Coarseness and fineness similarity score

In this subsection, we propose a method to find the coarseness and fineness similarity score of a texture. We have four data sets (Texta1 and Texta2, Texta3 and, Texta4) with textures that have been labelled coarse and fine. We consider a similarity score of textures to coarse (CNS) and fine textures (FNS) of each data set under verification scenario [32]. The CFS feature will have dimension eight shown by (4).

$$CFS = \langle CNS_1, CNS_2, CNS_3, CNS_4, FNS_1, FNS_2, FNS_3, FNS_4 \rangle \tag{4}$$

Where,  $CNS_k$  and  $FNS_k$  means CNS and FNS features were obtained from  $k^{th}$  data set.

### 3.4. Textures relabeling using CFS

As we know, the similarity thresholds ( $Thri$ ) and  $V_T$  have been obtained based on the textures in the gallery and probe data sets that labelled coarse and fine by human visual perception. Therefore, machine-based label of the textures may be different with their current label [17]. Two-dimensional plotting of the CFS features, shown in Figure 4(a), proves this idea. As we can see, the FNS value of some coarse textures is higher than their CNS. In addition, the CNS value of some fine textures are higher than their FNS. Note that to the 2-D plot of the CFS, we calculate the average of  $CNS_k$  and  $FNS_k$  values, shown by (5), for all  $k$  data sets ( $k=4$ ).

$$2D - CFS = \langle \overline{CNS}, \overline{FNS} \rangle = \langle \frac{CNS_1 + CNS_2 + CNS_3 + CNS_4}{4}, \frac{FNS_1 + FNS_2 + FNS_3 + FNS_4}{4} \rangle \tag{5}$$

To find the machine-based coarse and fine textures in the data sets, we relabel the textures based on their CNS and FNS values. After relabeling, we recalculate the  $Thri$  and  $V_T$  parameters and the CFS of the textures in the data sets. We repeat this process until there is not any coarse (fine) texture that its CNS (FNS) value is less than its FNS (CNS). Figure 4(b) depicts 2-dimensional diagram of the CFS for Texta4 data set after 10 times relabeling. Iterative algorithm to find true coarse and fine textures is in the following:

- a)  $k=1$ .
- b) Consider the  $k^{\text{th}}$  version of the data sets (Texta1(k) , Texta2(k), Texta3(k), and Texta4(k)).
- c) Calculate Similarity thresholds of classifiers under different pFAR on Texta1(k) and Texta2(k) as a gallery and probe, respectively.
- d) Calculate decision thresholds on Texta3(k).
- e) Calculate optimum pFAR(k), pFAR(k)<sub>opt</sub>, based on minimum EER at  $k^{\text{th}}$  repetition.
- f) Obtain system configuration;  $V_T$  and individual similarity thresholds based on pFAR(k)<sub>opt</sub>.
- g) Analysis coarseness and softness of  $k^{\text{th}}$  version of all data sets. And make new new version of the data sets ( $k=k+1$ ).
- h) If there is no change in the textures of data sets, go to step 10.
- i) Go to step 2
- j) Find the final  $V_T$  and individual similarity thresholds ( $Thri$ )

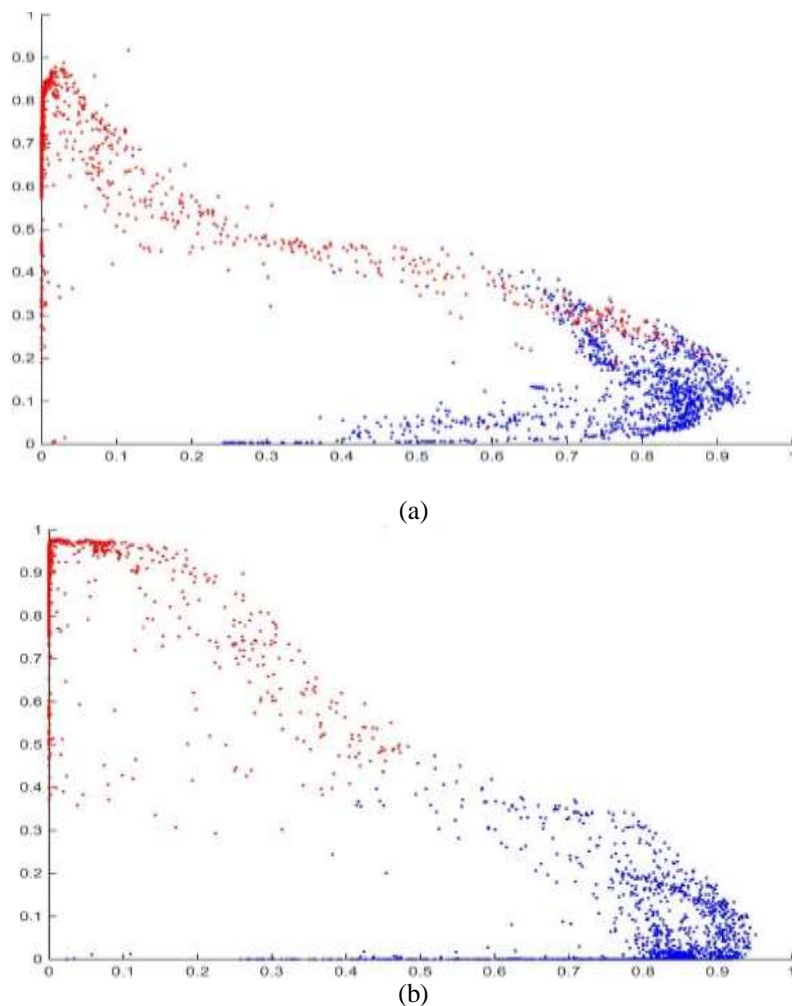


Figure 4. Two-dimensional plot of the CFS on Texta4. The vertical and horizontal axes are  $\overline{CNS}$  and  $\overline{FNS}$  values of 2D- CFS, respectively. (a) before refining. (b) After 10 times refining

Figure 5(a) depicts similarity threshold values of six classifiers ( $Thri$ ) under iteration 6 on pFAR (1% to 20%). Optimum pFAR is obtained on the Texta3 in the minimum value of EER. Figure 5(b) depicts diagram of EER on pFAR. Under minimum EER, the optimum pFAR is obtained 0.034 (or 3.4%). The similarity threshold values for six classifiers under optimum pFAR have been shown in the Table 2. Note that to change distance criterion to similarity, we multiply the distance values, calculated by distance measures, by -1. Figure 6 depicts verification parameters such as false acceptance rate (FAR), false rejection rate (FRR), true positive rate (TPR), and EER under tuned  $Thri$  on Texta4.

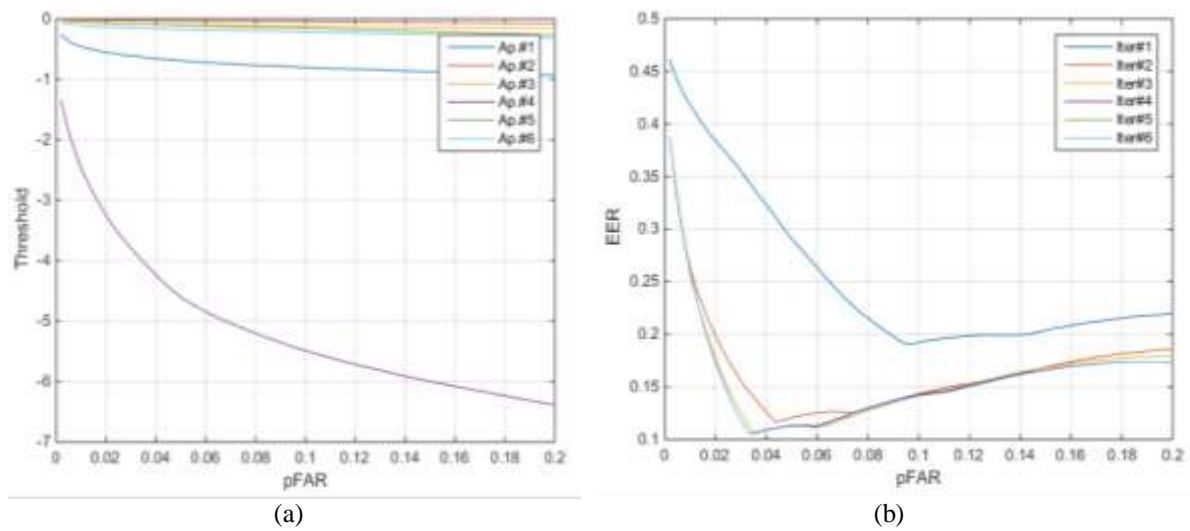


Figure 5. Diagram of the similarity thresholds of six classifiers (shown by labels Ap#1 to Ap#6) and EER on pFAR=0.1% to 20%. The values obtained on the Texta3. (a) Similarity thresholds values (Thri) on pFAR. (b) EER on pFAR

Verification parameters, false acceptance rate (FAR), false rejection rate (FRR), true positive rate (TPR), and EER under tuned Thri on Texta4 as shown in Figure 6.

Table 2. The similarity threshold values for six classifiers under optimum pFAR ( $pFAR_{opt} = 0.034$ )

	C#1	C#2	C#3	C#4	C#5	C#6
<i>Thr</i>	-0.63	-0.026	-0.085	-3.977	-0.086	-0.152

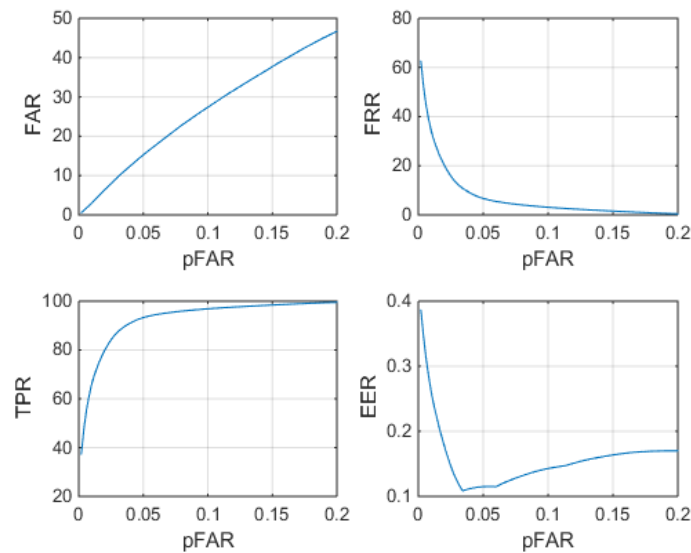


Figure 6. Verification parameters, false acceptance rate (FAR), false rejection rate (FRR), true positive rate (TPR), and EER under tuned Thri on Texta4

For calculation of CFS in pyramid representation, we repeat the calculation of recognition parameters such as Thri and VT for textures with resolution 64×64 and 32×32. In pyramid approach, we concatenated the CFS features of two representations with the original texture. Table 3 manifests

the similarity threshold values of classifiers under optimum pFAR. It can be seen that the threshold values have not huge change in different representations. However, the optimum FAR increased up to 3.8%.

Table 3. Similarity threshold values of classifiers under optimum pFAR on 64×64 and 32×32 textures

	@FARopt	C#1	C#2	C#3	C#4	C#5	C#6
64×64	0.038	-0.67	-0.025	-0.078	-4.03	-0.10	-0.17
32×32	0.038	-0.68	-0.022	-0.073	-3.66	-0.10	-0.18

#### 4. EXPERIMENTAL RESULTS

##### 4.1. Data set

The Outex data set is one of the well-known benchmark for texture classification algorithms. Our experiments on the Outex are tested on TI84 test suite. It contains 24 categories of textures with nine different rotation angles (0°, 5°, 10°, 15°, 30°, 45°, 60°, 75°, and 90°). There are 20 non-overlapping 128×128 texture samples for each category. In this research, we consider this data set for implementation of our proposed feature extraction operator.

##### 4.2. Distance measure

In the classification phase, non-supervised techniques are used. Since these types of classifiers can handle a large number of categories, they avoid parameter over-fitting, and need no learning and training. In this experiment, we classify the samples using the Chi-2 ( $\chi^2$ ) distance, the Euclidean distance (*ED*), and the Czekannowski distance (*Cze*), which are calculated as follows:

$$\chi^2(S, M) = \sum_{b=1}^B \frac{(S_b - M_b)^2}{S_b + M_b} \tag{6}$$

$$ED(S, M) = \sqrt{\sum_{b=1}^B (S_b - M_b)^2} \tag{7}$$

$$Cze(S, M) = \sum_{b=1}^B \frac{\|S_b - M_b\|}{S_b + M_b} \tag{8}$$

Where *B* denotes the number of histogram bins. In addition, *S* and *M* are histograms of sample and model, respectively.

##### 4.3. Texture classification with CFS descriptor

Figure 7 Depicts 2D-CFS values of Outex images in a normal and pyramid approaches. It can be seen that the proposed feature extraction can separate the textures of 24 categories in a good order. As we can see, the textures property of the this data set is close to coarse. While, after downsampling using Gaussian filter, the textures property goes to fine textures. Table 4 depicts the classification results of the Outex with normal and pyramid representation (PCFS). Although pyramid representation makes better results (upto 37%), but it has lower performance in comparison with state-of-the-arts texture descriptors, shown in Table 5. Thus, we combine features of PCFS with one of our latest feature selection approach, ML+MR [28], to obtain acceptable accuracy. Of course, the proposed approach has considerable low feature dimension against other operators. Table 5 depicts comparing of the best classification scores and feature length achieved using the proposed approach with the various state-of-the-art methods on the Outex-TI84 data set. It can be seen that with consideration of both accuracy and feature length, the proposed texture descriptor has a better performance than other approaches.

Table 4. Outex-tl84 classification under CFS operator

Operator	Cze	Chi2	ED	Feature length
CFS	66.67	64.92	64.71	8
PCFS <sup>a</sup>	87.67	87.63	88.58	24
CFS+MLMR <sup>b</sup>	98.79	98.38	98.33	136 (24+112)

a) a CFS under two Gaussian pyramid representations.

b) b MLMR is the ML+MR feature selection approach proposed by Akoushideh et al. [28]

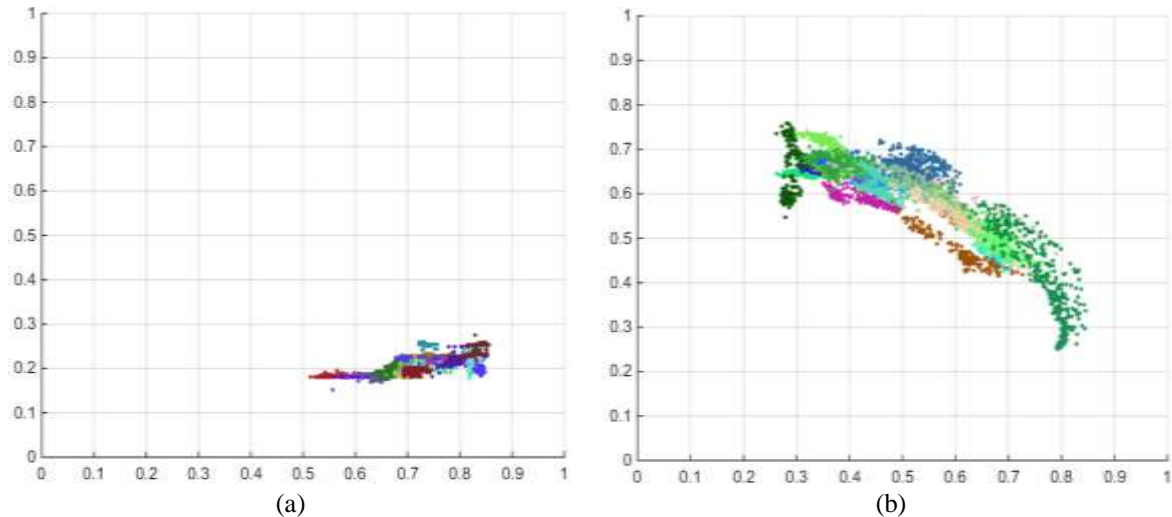


Figure 7. Diagram of two-dimensional CFS (2D-CFS) for the textures of Outex-TI84. The vertical and horizontal axes are  $\overline{CNS}$  and  $\overline{FNS}$ , respectively. Each category was depicted in different colour.  
 (a) 2D-CFS values on normal textures. (b) 2D-CFS values in pyramid representation (level one), down-sampled by Gaussian filter

Table 5. The comparing of the best classification scores and feature length achieved by the CFS with various state-of-the-art feature extraction operator for texture classification on the Outex-TI84 data set

Operators	Feature length	Accuracy	Similarity measure
<i>LBP</i> and <i>LBP/VAR</i> [35]	54	90.20%	<i>Chi-2</i>
<i>NI/RD/CI-LBP</i> [36]	1100	98.70%	<i>Chi-2</i>
<i>VZ-MR8</i> [37, 38]	2440	92.50%	<i>Chi-2</i>
<i>VZ-Joint</i> [39]	610	92.00%	<i>Chi-2</i>
<i>CLBP</i> [40, 41]	2200	95.40%	<i>Chi-2</i>
<i>MPLBP<sup>ru</sup></i> [27]	30	88.96%	<i>Chi-2</i>
<i>BRINT2_CS_CM</i> [42]	1296	99.35%	<i>NNC+Chi-2</i>
<i>LBP<sup>NT</sup></i> [42]	444	99.24%	<i>NNC+Chi-2</i>
<i>NRLBP</i> [42]	90	93.44%	<i>NNC+Chi-2</i>
ML+MR [28]	140	94.71%	<i>Chi-2</i>
ML+MR+MB [28]	224	97.88%	Cze
CFS	8	66.67%	Cze
PCFS	24	88.58%	ED
CFS+MLMR	136	98.79%	Cze

## 5. CONCLUSION

The experimental results show using proposed texture descriptor (CFS), we can receive to acceptable classification accuracy, with very low feature dimensionality vector. Further, with pyramid approach (PCFS) we can improve the performance of this operator up to 37% on the Outex. In addition, the combination of the PCFS with ML+MR makes considerable classification accuracy with very low feature length. For future work, we can consider new coarse and fine data sets to assess the performance of the CFS descriptor on different data. In addition, we can fuse features of the CFS with other state-of-the-art descriptors to improve classification accuracy. Furthermore, we can use the coarseness feature extractors, which have been proposed in the literatures, instead of (or with) considered features to improve the texture classification accuracy.

## REFERENCE

- [1] A. Paiement, M. Mirmehdi, X. Xianghua, and M. C. K. Hamilton, "Integrated Segmentation and Interpolation of Sparse Data," *Image Processing, IEEE Transactions on*, vol. 23, pp. 110-125, 2014.



- [2] H. Farsi and S. Mohamadzadeh, "Colour and texture feature-based image retrieval by using hadamard matrix in discrete wavelet transform," *Image Processing, IET*, vol. 7, pp. 212-218, 2013.
- [3] M. A. Ferrer, J. F. Vargas, A. Morales, and A. Ordonez, "Robustness of Offline Signature Verification Based on Gray Level Features," *Information Forensics and Security, IEEE Transactions on*, vol. 7, pp. 966-977, 2012.
- [4] P. Filiberto, G. Gema, G.-S. Pedro, M. Mirmehdi, and X. Xianghua, "Multi-spectral Texture Characterisation for Remote Sensing Image Segmentation," in *Proceedings of the 4th Iberian Pattern Recognition and Image Analysis Conference*, 2009.
- [5] A. Satpathy, J. Xudong, and E. How-Lung, "LBP-Based Edge-Texture Features for Object Recognition," *Image Processing, IEEE Transactions on*, vol. 23, pp. 1953-1964, 2014.
- [6] B. R. Rowshan, "Texture Classification in Underwater Images," MSc Erasmus Mundus in *Vision and RoBOTics (ViBOT)*, Department of Computer Architecture and Technology, 2011.
- [7] C. M. R. Caridade, A. R. S. Marc'al, and T. Mendonc, "The use of texture for image classification of black & white air-photographs," presented at the *New Developments and Challenges in Remote Sensing*, Z. Bochenek (ed.), Rotterdam, 2007.
- [8] Z. Baochang, G. Yongsheng, Z. Sanqiang, and Z. Bineng, "Kernel Similarity Modeling of Texture Pattern Flow for Motion Detection in Complex Background," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, pp. 29-38, 2011.
- [9] L. F. A. Pereira, H. N. B. Pinheiro, G. D. C. Cavalcanti, and R. Tsang Ing, "Spatial surface coarseness analysis: technique for fingerprint spoof detection," *Electronics Letters*, vol. 49, pp. 260-261, 2013.
- [10] K. Yung-Ming, L. Jiann-Shu, and C. Pau-Choo, "The nude image identification with adaptive skin chromatic distribution matching scheme," in *Computer Engineering and Technology (ICET), 2010 2nd International Conference on*, pp. V7-117-V7-120, 2010.
- [11] M. Tuceryan and A. K. Jain, "Texture analysis: The Handbook of Pattern and Computer Vision," 1998.
- [12] T. Ojala and M. Pietikainen, "Texture classification, 2001. Available: [http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/OJALA1/texclas.htm](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OJALA1/texclas.htm)
- [13] M. Roumi, "Implementing Texture Feature Extraction Algorithms on FPGA," MSc, *Electrical Engineering*, Delft University of Technology, 2009.
- [14] J. Chamorro-Martinez and P. Martinez-Jimenez, "A comparative study of texture coarseness measures," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pp. 1337-1340, 2009.
- [15] R. M. Haralick, "Statistical and Structural Approaches to Texture," *Proceedings of the IEEE*, vol. 67, pp. 786-804, 1979.
- [16] R. Chellappa, R. L. Kashyap, and B. S. Manjunath, "Model-based Texture Segmentation and Classification," In: Chen, C.H., Pau, L.F., Wang, P.S.P. (eds.) *The Handbook of Pattern Recognition and Computer. Singapore: World Scientific*, 1998.
- [17] H. Tamura, S. Mori, and T. Yamawaki, "Textural Features Corresponding to Visual Perception," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 8, pp. 460-473, 1978.
- [18] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, pp. 51-59, 1996.
- [19] T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, and S. Huovinen, 2002. Available: <http://www.outex.oulu.fi>
- [20] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural Features for Image Classification," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 3, pp. 610-621, 1973.
- [21] M. V. Boland, "Quantitative Description and Automated Classification of Cellular Protein Localization Patterns in Fluorescence Microscope Images of Mammalian Cells," *Pittsburgh*, 1999.
- [22] L.-E. Santos and C. Rene, "Decision Tree Based FPGA-Architecture for Texture Sea State Classification," in *Reconfigurable Computing and FPGA's, 2006. ReConFig 2006. IEEE International Conference on*, 2006, pp. 1-7.
- [23] J. M. H. du Buf, M. Kardan, and M. Spann, "Texture Feature Performance for Image Segmentation," *Pattern Recognition*, vol. 23, pp. 291-309, 1990.
- [24] A. R. Akoushideh, A. Shahbahrani, and B. Mazloom-Nezhad M., "High performance implementation of texture features extraction algorithms using FPGA architecture," *Real-Time Image Processing*, 2012.
- [25] Z. Haiying, X. Zhengguang, and H. Peng, "Performance Evaluation for Three Classes of Textural Coarseness," in *Image and Signal Processing, 2009. CISP '09. 2nd International Congress on*, pp. 1-4, 2009.
- [26] P. Howarth and S. Ruger, "Evaluation of Texture Features for Content-Based Image Retrieval," in *Image and Video Retrieval*, vol. 3115, P. Enser, Y. Kompatsiaris, N. O'Connor, A. Smeaton, and A. M. Smeulders, Eds., ed: Springer Berlin Heidelberg, pp. 326-334, 2004.
- [27] X. Qian, X.-S. Hua, P. Chen, and L. Ke, "PLBP: An effective local binary patterns texture descriptor with pyramid representation," *Pattern Recognition*, vol. 44, pp. 2502-2515, 2011.
- [28] A. Akoushideh and B. M.-N. Maybodi. (2015, Efficient levels of spatial pyramid representation for local binary patterns. Institution of Engineering and Technology, -. Available: <http://digital-library.theiet.org/content/journals/10.1049/iet-cvi.2015.0028>
- [29] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006, pp. 2169-2178.
- [30] X. Qian, G. Liu, D. Guo, Z. Li, Z. Wang, and H. Wang, "HWVP: hierarchical wavelet packet descriptors and their applications in scene categorization and semantic concept retrieval," *Multimedia Tools and Applications*, vol. 69, pp. 897-920, 2012.

- [31] A. Bosch, A. Zisserman, and X. Muñoz, "Representing shape with a spatial pyramid kernel," presented at the *CIVR, Amsterdam*, 2007.
- [32] L. Spreuwers, "Fast and Accurate 3D Face Recognition," *International Journal of Computer Vision*, vol. 93, pp. 389-414, 2011.
- [33] S.-H. Cha, "Taxonomy of Nominal Type Histogram Distance Measures," presented at the *American Conference on Applied Mathematics (MATH '08)*, 2008.
- [34] B. Caputo, E. Hayman, M. Fritz, and J.-O. Eklundh, "Classifying materials in the real world," *Image Vis. Comput.*, vol. 28 pp. 150-163, 2010.
- [35] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, pp. 971-987, 2002.
- [36] L. Li, L. Zhao, Y. Long, G. Kuang, and P. Fieguth, "Extended local binary patterns for texture classification," *Image and Vision Computing*, vol. 30, pp. 86-99, 2012.
- [37] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *Int. J. Comput. Vis.*, vol. 62 pp. 61-81, 2005.
- [38] D. P. Dogra, K. Tripathy, A. K. Majumdar, and S. Sural, "A comparative study on texture features used for segmentation of images rich in texture," in *Signal and Image Processing Applications (ICSIPA), 2009 IEEE International Conference on*, pp. 336-339, 2009.
- [39] M. Varma and A. Zisserman, "A statistical approach to material classification using image patches," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31 pp. 2032-2047, 2009.
- [40] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Trans. Image Process*, vol. 9 pp. 1657-1663, 2010.
- [41] R.-j. Liu, M.-s. Fu, and J. Fang, "A new texture image segmentation algorithm," in *Computer Science and Education (ICCSE), 2010 5th International Conference on*, 2010, pp. 574-577.
- [42] L. Li, L. Yunli, P. W. Fieguth, L. Songyang, and Z. Guoying, "BRINT: Binary Rotation Invariant and Noise Tolerant Texture Classification," *Image Processing, IEEE Transactions on*, vol. 23, pp. 3071-3084, 2014.