Journal of Physics: Conference Series

PAPER • OPEN ACCESS

A novel hybrid feature extraction method using LTP, TFCM, and GLCM

To cite this article: Fallah H. Najjar *et al* 2021 *J. Phys.: Conf. Ser.* **1892** 012018

View the article online for updates and enhancements.

A novel hybrid feature extraction method using LTP, TFCM, and GLCM

Fallah H. Najjar[1], Hassan M. Al-Jawahry[1], Mustafa Saleh Al-Khaffaf[1], Ahmed T. Al-Hasani[2]

[1] Al-Furat Al-Awsat Technical University, Technical College of Management, Iraq
[2] Al-Furat Al-Awsat Technical University, College of Health and Medical Techniques, Iraq

E-mail: fallahnajjar@atu.edu.iq

Abstract. Image classification and feature extraction have been studied extensively and used efficiently in several applications. This paper suggests a novel method by combining three main methods for texture feature extraction. The proposed method is based on Local Ternary Pattern (LTP), Texture Feature Coding Method (TFCM), and Gray Level Cooccurrence Matrix (GLCM). We have entitled our method as GCLTP which is stand for Gray Coding Local Ternary Pattern. The combination of LTP, TFCM, and GLCM is assigned a unique value used to extract the features of an image. GCLTP is tested using images are taken from the Brodatz database. A set of 22 features were extracted from images. GCLTP is experimentally accomplished a high accuracy in classification by using the most known classifiers.

1. Introduction

The classification of images is used in many areas, for example, the medical classification of the disease. It is also utilized in the classification of texture images. There are many ordinary methods in this area which depend on the extraction of texture information, like Gray Level Cooccurrence Matrix (GLCM) [1, 2], Texture Spectrum [3], Local Binary Pattern (LBP) [4], Local Ternary Pattern (LTP) [5], and Texture Feature Coding Method (TFCM) [6]. The features extraction depends on the local texture properties for each pixel of the images, thus obtaining many statistics. However, the researchers have developed several hybridization methods to reach more superior results for feature extraction and classification. In the literature, the combination of LTP, TFCM, and GLCM is never done before. Therefore, this paper proposes a novel method by combining three principal methods for texture feature extraction.

The rest of the paper is divided into five sections. Firstly, the related works are discussed in section two. Secondly, section three states the explanation of the methodology of the proposed method. Next, the performance results are illustrated in section four. Finally, section five concludes the whole study.

2. Related work

Classification and feature extraction has been studied widely and used efficiently in several applications. The extensively used feature extraction methods are the LBP, GLCM, LTP, TFCM, wavelet, and Gabor filter methods. Each method falls under one of three categories: model-based, statistical, and structural methods. In the next subsections, the main methods LBP, LTP, TFCM, and GLCM are discussed: readers are recommended to see reference [7].

LBP method was proposed by [4]. LBP is a special case of the texture spectrum [3]. The main process of LBP is to binarize the pixel information in an image by comparing the eight surrounding pixels with the focal (center) pixel [8]. LBP is often employed in detection [9, 10], recognition [11, 12], classification [13, 14], as well as object tracking [15].

LTP [5] is another method that is used to encode the information pixel of an image. Also, it compares the surrounding pixels with the focal pixel. The crucial differences between LTP and LBP are the following, LTP is the use of triple threshold, while LBP is not. Secondly, the comparison results of LTP map the values into three types [-1, 0, and 1], while in LBP the values are [0 and 1]. Furthermore, LTP can be divided into LTP Upper, LTP Lower, and LTP Normal matrices.

TFCM method is a coding representation that converts the original images into texture feature images whose pixels are characterized by texture feature numbers. The texture feature number of individual pixels is produced based on its 8 surrounded pixels [6].

GLCM method was proposed by [1, 2] as a statistical procedure to extract texture features from an image. It was designed by calculating transitions between couples of two pixels. The dimension of the produced matrix is equal to the maximum value of the original image. GLCM depends on two parameters: distance and angular.

Support Vector Machine (SVM) is a common supervised machine learning algorithm which is used for classification or regression functions. It was introduced in (1964) by Vapnik. SVM was developed for two-class datasets cases that are separable by a linear classifier. Then it gets expanded to none separable data before it is extended to multi classes dataset. For more details, see [16]. SVM classifier has several kernels, in addition to linear SVM kernel (L-SVM), like Gaussian, polynomial and sigmoid kernels.

Decision Trees (DT) [17] is an important and common supervised machine learning technique that classify instances by sorting them depending on the values of the attribute. Each attribute in the dataset is represented as a node in DT to be classified. Then, each branch characterizes a value that the node can assume. An initial state of DT represents the root node which is assigned to all data. No more decisions are needed only if the value at the node belongs to the same class. Otherwise, the node has to be split, and the node will be a decision node. Finally, the calculation of the decision parameters is based on the information maximization.

3. Methodology

Since our proposed method focuses on feature extraction based on LTP, TFCM, and GLCM, it will be known as GCLTP for Gray Coding Local Ternary Pattern. The mentioned methods which are applied individually and in the hybrid mode on an image, and a set of 22 features (Inverse difference (INV), Inverse difference normalized (INN), Inverse difference moment normalized, Homogeneity, Contrast, Autocorrelation, Correlation, Cluster Prominence, Cluster Shade, Dissimilarity, Energy, Entropy, Maximum probability, Sum of Squares, Variance, Sum average, Sum variance, Sum entropy, Difference variance, Difference entropy, Information measure of correlation1, and Information measure of correlation2) are extracted. Then, the extracted features – for each method – will be combined into one dataset. GCLTP is described as follows:

3.1. Feature Extraction

In the beginning, each image in the dataset is divided into 4 sub-images. Then, we utilize LTP applied to the first, second, third, and the last sub-image. Consequently, we get three matrices (LTPU, LTPN, and LTPL). The emerged matrices are used to extract the features set. Also, these resulting matrices will be inputted to the second method TFCM. Afterwards, TFCM will export three new matrices (TFCMU, TFCMN, and TFCML), and the feature will be extracted from them.

After that, the last resulting matrices are considered as an input to the third method GLCM through which the features are also extracted. GLCM produces five matrices for each of the previous outputs (GLCMA, GLCM1, GLCM2, GLCM3, and GLCM4). Moreover, the features are extracted from the five matrices.

Finally, the extracted features from (LTPU, TFCMU, and GLCMAU) are combined into one matrix. Likewise, the same procedure will be applied to the rest of GLCM1U, GLCM2U, …, and so on. As a final result, GCLTP will provide fifteen datasets. Each dataset with 66 observations/columns/features and 364 samples/rows, and one label/class with 13 classes. Figure 1 demonstrates the GCLTP structure.

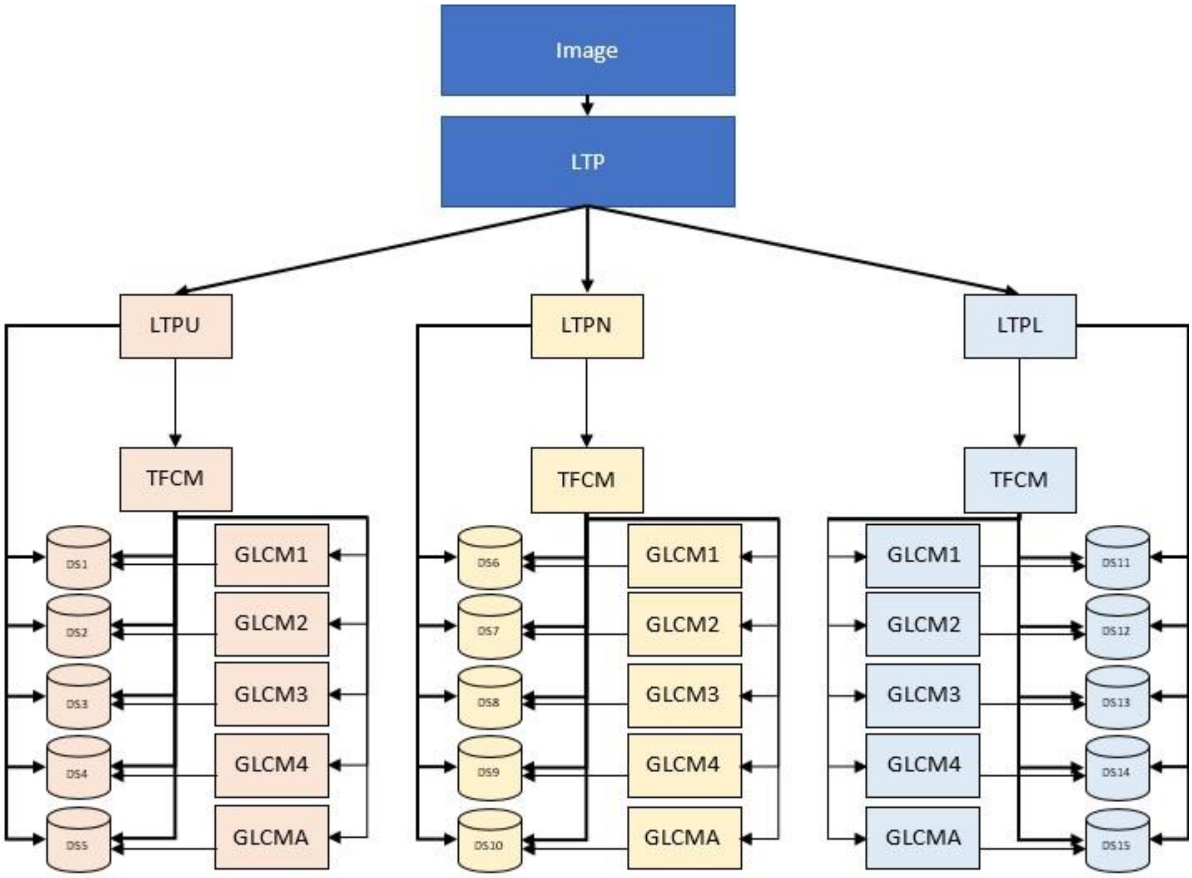

Figure 1. The proposed method structure.

The produced dataset is named as (GCLTPL1, GCLTPL2, GCLTPL3, GCLTPL4, GCLTPLA, GCLTPN1, GCLTPN2, GCLTPN3, GCLTPN4, GCLTPNA, GCLTPU1, GCLTPU2, GCLTPU3, GCLTPU4, and GCLTPUA).

3.2. Classification

In this section, each dataset is divided into a training and testing set. Training set with 80% of the original dataset, while the testing set is 20%. As it was previously mentioned, the support vector machine is used with a linear kernel and decision tree classifiers to evaluate our GCLTP. In the next section, the discussion of the experimental results will be shown—the calculation of the precision, recall, f1-score, and accuracy for each dataset.

Precision or sensitivity is a measurement that counts the number of accurate positive predictions made [18]. It can be calculated using the following formula:

$$P = \frac{Sum\ c\ in\ C\ TruePositives\_c}{Sum\ c\ in\ C\ (TruePositives\_c\ +\ FalsePositives\_c)} \quad \textbf{(1)}$$

Recall is a measurement that counts the number of accurate positive predictions made out of all positive predictions that might have been made [18]. It can be calculated using the following formula:

$$R = \frac{Sum\ c\ in\ C\ TruePositives\_c}{Sum\ c\ in\ C\ (TruePositives\_c\ +\ FalseNegatives\_c)} \quad \textbf{(2)}$$

F1-Score or F-Measure is a combination of precision and recall in one measure formula [18]. It can be calculated by:

$$R = \frac{2 * P * R}{(P + R)} \quad \textbf{(3)}$$

Finally, accuracy [18] can be calculated based on the confusion matrix by:

$$A = \frac{TruePositives + TrueNegative}{(TruePositives + TrueNegative + FalsePositives + FalseNegative)} \quad \textbf{(4)}$$

4. Experimental Results

In this study, we have conducted two classifiers (L-SVM and DT) to verify the efficiency of GCLTP. We have used a part of a well-known dataset, so-called Brodatz texture images. This part consists of thirteen images with seven different rotation angles: 0, 30, 60, 90, 120, 150, and 200 degrees. All images are 512x512 pixels [18]. The GCLTP is implemented using MATLAB R2018a, while classification algorithms are established using Python v3 with Spyder under anaconda. All experiments are run on a machine with Intel Core i7-6500U CPU of 2.50 GHz and 8.0 GB RAM. Figure 2 show the main thirteen images in the dataset.

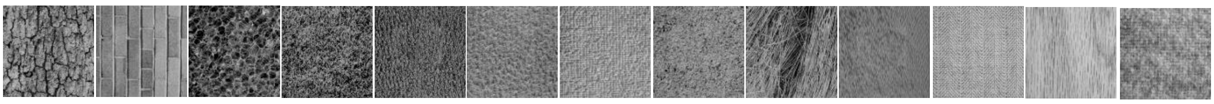

Figure 2. The main thirteen images in the Brodatz texture dataset [19]

In this subsection, we will introduce our method results of testing the datasets' performance that are produced from the GCLTP system. We use SVM with the linear kernel and DT classifiers. To account for the imbalance data label, we have considered the precision, recall and F1 score. Accuracy as the common performance measuring is calculated for each benchmark. Table 1 and Table 2 illustrate the datasets of classification results. Consequently, we further summarize F1-score for GCLTPL, GCLTPN, and GCLTPU datasets in Figure 3, Figure 4, and Figure 5, respectively.

Table 1. L-SVM Classification Results

DATASET	RECALL	PRECISION	F-SCORE	ACCURACY
GCLTPL1	0.822	0.829	0.810	0.822
GCLTPL2	0.836	0.866	0.829	0.836
GCLTPL3	0.932	0.940	0.930	0.932
GCLTPL4	0.808	0.867	0.805	0.808
GCLTPLA	0.877	0.885	0.868	0.877
GCLTPN1	0.877	0.893	0.875	0.877
GCLTPN2	0.918	0.934	0.918	0.918
GCLTPN3	0.918	0.945	0.919	0.918
GCLTPN4	0.918	0.914	0.914	0.918
GCLTPNA	0.890	0.914	0.893	0.890
GCLTPU1	0.808	0.810	0.801	0.808
GCLTPU2	0.863	0.877	0.864	0.863
GCLTPU3	0.918	0.921	0.916	0.918
GCLTPU4	0.863	0.923	0.867	0.863
GCLTPUA	0.863	0.882	0.858	0.863

Table 2. DT Classification Results

DATASET	RECALL	PRECISION	F-SCORE	ACCURACY
GCLTPL1	0.836	0.850	0.836	0.836
GCLTPL2	0.836	0.835	0.827	0.836
GCLTPL3	0.877	0.882	0.875	0.877
GCLTPL4	0.904	0.919	0.903	0.904
GCLTPLA	0.904	0.915	0.902	0.904
GCLTPN1	0.904	0.928	0.901	0.904
GCLTPN2	0.904	0.924	0.906	0.904
GCLTPN3	0.877	0.881	0.877	0.877
GCLTPN4	0.836	0.888	0.843	0.836
GCLTPNA	0.863	0.880	0.863	0.863
GCLTPU1	0.863	0.863	0.857	0.863
GCLTPU2	0.877	0.896	0.877	0.877
GCLTPU3	0.918	0.941	0.918	0.918
GCLTPU4	0.836	0.907	0.847	0.836
GCLTPUA	0.822	0.876	0.823	0.822

Figure 3 and Figure 5 conclude that F1-score accuracy is remarkably high for all GCLTPL and GCLTPU benchmarks with no less of 0.801, while the GCLTPN (Figure 4) has the highest F1-score accuracies in both SVM-L and DT classifiers.

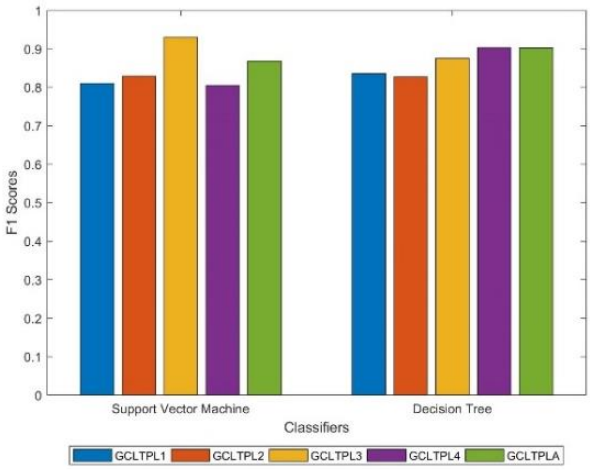

Figure 3. GCLTPL F1-Score Accuracies

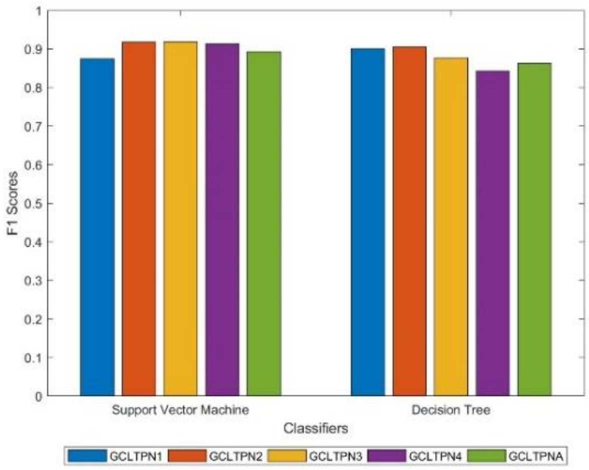

Figure 4. GCLTPN F1-Score Accuracies

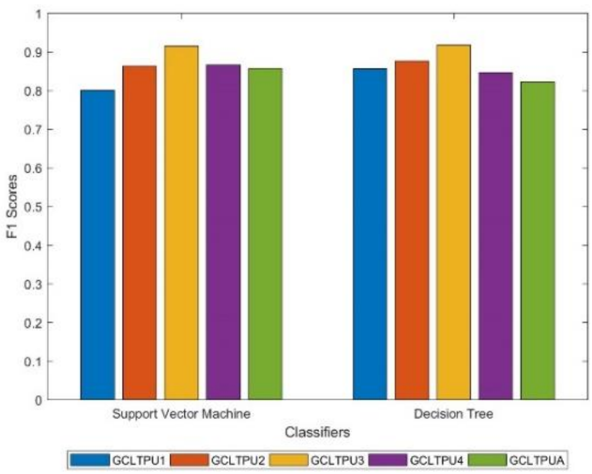

Figure 5. GCLTPU F1-Score Accuracies

5. Conclusions

In this paper, we have proposed a novel texture feature extraction method. It has aimed to extract feature from the image. We combined three main feature extraction methods known as (LTP, TFCM, and GLCM). The proposed method is named as GCLTP. To evaluate the performance of GCLTP, first, we carried out experiments on Brodatz dataset. Second, two classifiers are used to evaluate the performance of our method. The results have shown that GCLTP performance is at least as accurate as other well-established approaches in the literature. The main interesting future research is applying the GCLTP to colored image and enhancing the feature extraction's time.

References

[1] R. M. J. P. o. t. I. Haralick, 1979, *Statistical and structural approaches to texture*, vol. **67**, no. 5, pp. 786-804.
[2] R. M. Haralick, K. Shanmugam, I. H. J. I. T. o. s. Dinstein, man, and cybernetics, 1973, *Textural features for image classification*, no. 6, pp. 610-621.
[3] D.-C. He and L. J. P. r. Wang, 1991, *Texture features based on texture spectrum*, vol. **24**, no. 5, pp. 391-399.
[4] T. Ojala, M. Pietikainen, T. J. I. T. o. p. a. Maenpaa, and m. intelligence, 2002, *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*, vol. **24**, no. 7, pp. 971-987.
[5] X. Tan and B. J. I. t. o. i. p. Triggs, 2010, *Enhanced local texture feature sets for face recognition under difficult lighting conditions*, vol. **19**, no. 6, pp. 1635-1650.
[6] M.-H. Horng, X.-J. Huang, and J.-H. J. J. o. O. E. Zhuang, 2003, *Texture Feature Coding Method for Texture Analysis and It's Application*, vol. **42**, no. 1, pp. 228-238.
[7] M. Nixon and A. Aguado, 2019, *Feature extraction and image processing for computer vision*. Academic press.
[8] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen, 2011, *Computer vision using local binary patterns*. Springer Science & Business Media.
[9] T. Mahmood, A. Irtaza, Z. Mehmood, and M. T. J. F. s. i. Mahmood, 2017, *Copy–move forgery detection through stationary wavelets and local binary pattern variance for forensic analysis in digital images*, vol. **279**, pp. 8-21.
[10] A. Alahmadi *et al.* , 2017, *Passive detection of image forgery using DCT and local binary pattern*, vol. **11**, no. 1, pp. 81-88.
[11] C.-S. Yang and Y.-H. J. P. R. L. Yang, 2017, *Improved local binary pattern for real scene optical character recognition*, vol. **100**, pp. 14-21.
[12] A. Bolotnikova, H. Demirel, G. J. A. I. C. Anbarjafari, and S. Processing, 2017, *Real-time ensemble based face recognition system for NAO humanoids using local binary pattern*, vol. **92**, no. 3, pp. 467-475.
[13] Z. Pan, Z. Li, H. Fan, and X. J. E. S. w. A. Wu, 2017, *feature based local binary pattern for rotation invariant texture classification*, vol. **88**, pp. 238-248.
[14] M. H. Shakoor, R. J. M. T. Boostani, and Applications, 2018, *A novel advanced local binary pattern for image-based coral reef classification*, vol. **77**, no. 2, pp. 2561-2591.
[15] J. Kim, S. Yu, D. Kim, K.-A. Toh, and S. J. P. R. Lee, 2017, *An adaptive local binary pattern for 3D hand tracking*, vol. **61**, pp. 139-152.
[16] S. J. S. E. Jeyanthi and R. Center, 2007, *Efficient classification algorithms using sVMs for large datasets*.
[17] J. R. J. I. j. o. m.-m. s. Quinlan, 1987, *Simplifying decision trees*, vol. **27**, no. 3, pp. 221-234.
[18] M. Hossin, M. J. I. J. o. D. M. Sulaiman, and K. M. Process, 2015, *A review on evaluation metrics for data classification evaluations*, vol. **5**, no. 2, p. 1.
[19] P. Brodatz, 1966, *Textures: a photographic album for artists and designers*. Dover Pubns.