TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 3, June 2020, pp. 1224~1228
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i3.14256

A modified symmetric local binary pattern for image features extraction

Majed O. Dwairi
Faculty of Engineering Technology, Communication Engineering Department, Al-balqa Applied University, Jordan

Article Info

Article history:

Received Oct 20, 2019
Revised Feb 12, 2020
Accepted Feb 23, 2020

Keywords:

CSLBP
Extraction time
Image features
Image rotation
LBP
MSLBP

ABSTRACT

The process of identifying images and patterns is one of the most important processes of digital image processing, which is used in many applications such as fingerprint recognition, face recognition and pattern recognition. Due to the large size of the image, the process of identifying the image requires a great time, which in turn leads us to extract some characteristics of the magnitude of the volume, which can be used as an identifier to retrieve the image or recognize it and thus we have devoted a lot of time to identify the image. In this research paper, a modified symmetric local binary pattern (MSLBP) method was proposed to extract texture features. The proposed algorithm was implemented on many digital fingerprint's images and the local structure features of these images were obtained. Several image recognition experiments are conducted on these features and compared with other algorithms. The results of the proposed algorithm showed that the digital image was represented in a very small size and furthermore the speed and accuracy of image recognition based on the proposed method was increased significantly. Unlike the methods based on LBP, the proposed method gives the same features of the image even if the image was rotated with any angle.

Corresponding Author:

Majed O. Dwairi,
Faculty of Engineering Technology, Communication Engineering Department,
Al-balqa Applied University,
Amman, Jordan.
Email: majeddw@bau.edu.jo

1. INTRODUCTION

Gray digital images are large in size and are usually several thousand pixels in size, moreover it takes a great deal of time to process the images to identify them [1], Table 1 shows some different in size images, and the required time to identify each image. From this table we can see that the average time to process each pixel for matching equal 10.945 micro seconds, which is considered a high time. Figure 1 shows that there is a linear relationship between the image size and the required recognition or identifying time.

To avoid this problem, and to minimize the recognition time we have to seek an efficient method capable to represent the image by a set of values called image features, which can be used as an identifier to retrieve or recognize the image. An image features is a set of metrics calculated in image processing and they are created to quantify the perceived texture of an image. Image features give us information about the spatial arrangement of pixels values or intensities in an image or selected region of an image, such as encryption and decryption [2-4]. The extracted features must form a key which can be used as an image identifier, and here these features must be:

- Unique for each image and data [5, 6].
- Small size comparing with the image size [7].
- Capable to reduce the image retrieving time [8].
- Simple to be created.
- Dependable on the image texture [5, 9, 10].
- Unchangeable if the image was rotated.

Table 1. Image size and identifying time

Image	Number of rows	Number of columns	Size (pixel)	Matching time (Seconds)
1	368	267	98256	0.021000
2	265	570	151050	0.031000
3	283	534	151122	0.034000
4	225	675	151875	0.036000
5	600	385	231000	0.038000
6	500	1065	532500	0.051000
7	1079	1950	2104050	0.205000
8	1300	3027	3935100	0.388000
Average			919370	0.1005
Time for each pixel		10.945 microseconds		

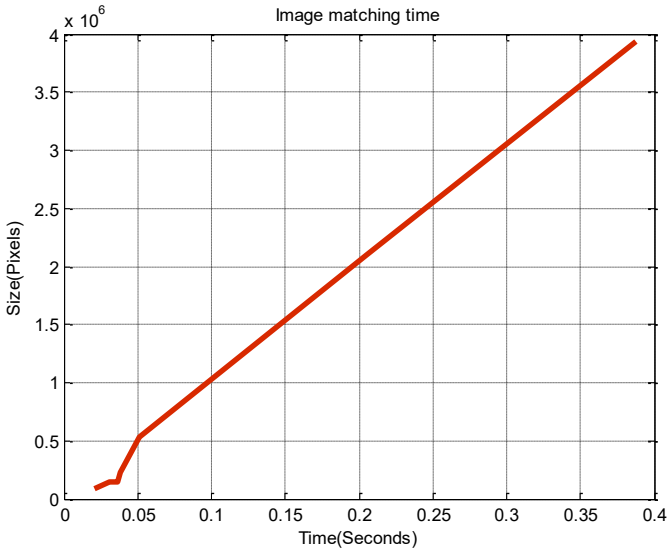

Figure 1. Relationship between image size and identifying time

2. RELATED WORKS

Several algorithms were conducted to extract image features, most of them are based on calculating local binary pattern (LBP) for each pixel, then the repetition of each LBP value is to be find, these repetitions will form the image features [8, 9]. In [6], the dominate LBP operator was proposed, this operator is to be calculated for each pixel, and the repetitions of the operator values form the image features, the proposed method here is very simple and fully based on LBP method [9]. In [10], a window method for an enhanced image [11-17] features extraction was proposed, this method is very simple and efficient but if the image was rotated the features will change, which will cost extra work and time to deal with process of identifying the image. In [18-23] deferent variants of algorithm were proposed, all of them are based on LBP and central symmetric LBP (CSLBP) operators, these methods create a unique feature for each image, but they are very sensitive to the image rotation. Figure 2 (a) shows how to calculate LBP operator for each pixel, while Figure 2 (b) shows how to calculate CSLBP operator for each pixel. CSLBP methods creates for each image a unique features array of 16 values as shown in Table 2, but these features are very sensitive to the image position and if the image was rotated at least for 1 degree the features array will be changed accordingly as shown in Table 3, and this is the major disadvantage of this method.

Calculating LBP operator for each pixel					
110	80	70	1	0	0
105	100	120	1	-	1
30	70	65	0	0	0
110>=80 so 1 80<100 so 0			Binary=11000100 Decimal=196 LBP=196		

(a)

Calculating CSLBP operator for each pixel					
110	80		1		
105	100		1	-	
30	70		0	0	
110>=80 so 1 80<100 so 0			Binary=1100 Decimal=12 LBP=12		

(b)

Figure 2. (a) Calculating LBP operator, (b) Calculating CSLBP operator

Table 2. Images features using CSLBP method

Image features				
Image 1	Image 2	Image 3	Image 4	Image 5
20622	208737	8579	13704	11358
14974	134462	7909	11075	6918
7113	68040	2245	4273	3513
13009	100008	7475	8238	7227
8030	108792	2225	4090	4851
4305	38518	1354	2879	2562
4289	43447	1460	3314	2583
13024	158969	6432	9804	11850
13964	108208	5696	10205	12066
4290	52415	1424	3204	2625
4779	38166	1360	2609	2274
6085	100799	2487	3688	4116
12934	124558	7843	8525	7905
5526	83046	2429	3669	3396
14550	110970	8001	10245	6447
81540	2447315	30071	49862	59801

Table 3. Image features before and after rotation

Image features	
Image 1	Rotated image1
20622	21331
14974	16060
7113	7346
13009	13418
8030	6186
4305	3922
4289	4457
13024	12646
13964	13282
4290	5081
4779	4526
6085	5506
12934	13146
5526	5815
14550	14611
81540	81701

3. THE PROPOSED MSLBP METHOD

The proposed modified symmetric LBP method calculates for each pixel using the pixel neighbors with depth equal 1 and depth equal 2 as shown in Figure 3. Here if rotate the image the pixel neighbors did not change, so the features remain the same after any rotation of the image. MSLBP method can be implemented applying the following steps (for each pixel):

a. Initialize the 4 elements features array to zeros.
b. Find the average of the neighbors with depth = 1(av0).
c. Find the average of the neighbors with depth = 2(av1).
d. If av0 greater or equal pixel value make a0 = 1, else make a0 = 0.
e. If av1 greater or equal pixel value make a1 = 1, else make a1 = 0.
f. Find the index of the features array (I = a0 + 2*a1).
g. Add 1 to the features array with index = I.

Figure 4 shows an example of how to find the features array index for one pixel.

P(I-2, J-2)	P(I-2, J-1)	P(I-2, J)	P(I-2, J+1)	P(I-2, J+2)
P(I-1, J-2)	P(I-1, J-1)	P(I-1, J)	P(I-1, J+1)	P(I-1, J+2)
P(I, J-2)	P(I, J-1)	P(I, J)	P(I, J+1)	P(I, J+2)
P(I+1, J-2)	P(I+1, J-1)	P(I+1, J)	P(I+1, J+1)	P(I+1, J+2)
P(I+2, J-2)	P(I+2, J-1)	P(I+2, J)	P(I+2, J+1)	P(I+2, J+2)

If average of Green neighbors >=P(I,J) a0=1
Else a0=0;
If average of red neighbors >=P(I,J) a1=1
Else a1=0;
Feature index = a0+2*a1

Figure 3. Calculating MSLBP operator

255	205	130	170	100
120	150	160	170	170
130	100	175	180	160
200	200	219	211	120
140	150	160	255	255

Average of Green neighbors =1390/8=173.75
So a0=0.
Average of red neighbors =2820/16=176.25
So a1=1.
Binary = 1 0
Feature index=a0+2*a1=2.

Figure 4. Calculating features array index

4. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The proposed MSLBP method was implemented using various fingerprint images with various sizes, and for each image the image features array was a unique for each image. Each image was rotated for various degrees, and the resulting features remain the same without any change. Figure 5 shows an original fingerprint image, and the rotated for 90 degrees image, while Table 4 shows the features array for the image before and after rotation.

The original fingerprint image was taken, the features array was calculated, and the same thing was done for different variants of rotated fingerprint image, the results of implementation is shown in Table 5. From Table 5 we can see that the features array of the image does not change due to image rotation, this gives the proposed algorithm a big advantage over the other used method, and regardless to the image position the features remain the same. The features array extraction (calculation) times were calculated for images with different sizes using both CSLBP and MSLBP methods, the results of calculations are shown in Table 6. From the results shown in Table 6 we can see that the extraction time to create an image features array using MSLBP method is small and it is acceptable, but CSLBP method is more efficient for this case. This disadvantage can be ignored taking the following facts into consideration:

a. The features array using MSLBP method has only 4 elements, while the features array using CSLBP method has 16 elements.
b. The database which can be used to store the features arrays (keys) using MSLPB method requires a smaller memory size.
c. If we use artificial neural network (ANN) [24] as a tool to identify the image using MSLBP, to this ANN architecture will be simpler.
d. Taking 3 into consideration ANN training time will be smaller.
e. Taking 3 into consideration image retrieving time using ANN will be also smaller, and this will compensate the bigger extraction time.

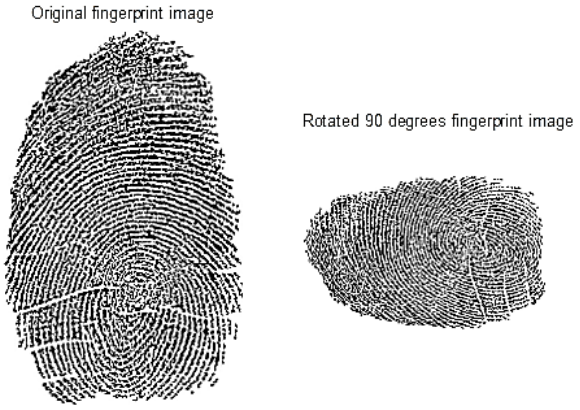

Figure 5. Original and rotated fingerprint images

Table 4. Features before and after rotation

Image	Image features			
Original	168067	6098	3169	48766
Rotated	168067	6098	3169	48766

Table 5. Features for the image with various positions

Image	Features			
Original	168067	6098	3169	48766
Rotated 1 degrees	168067	6098	3169	48766
Rotated 5degrees	168067	6098	3169	48766
Rotated 7 degrees	168067	6098	3169	48766
Rotated 10 degrees	168067	6098	3169	48766
Rotated 20 degrees	168067	6098	3169	48766
Rotated 48 degrees	168067	6098	3169	48766
Rotated 73 degrees	168067	6098	3169	48766
Rotated 90 degrees	168067	6098	3169	48766

Table 6. Features extraction time

Image	Size (pixels)	CSLBP features extraction time (second)	MSLBP features extraction time(second)
1	600x 385	0.034000	0.174000
2	1300x 3027	0.248000	1.001000
3	368x 267	0.007000	0.080000
4	265x 570	0.009000	0.040000
5	283x 534	0.010000	0.043000

5. CONCLUSION

A simple and highly efficient MSLBP method for image features extraction was, proposed tested and implemented. The proposed method can suit any application in the field of image recognition or image retrieval. The proposed method provides some achievements such as: small feature array size, small features extraction time, and the features array values are not sensitive to the image position, rotating the image does not lead to any changes in the image features array.

REFERENCES

[1] B. Zahran, et al., "A Modified Lbp Method to Extract Features from Color Images," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 10, pp. 3014-3024, May 2018.

[2] M. A. Zughoul, et al., "Efficient Methods used to Extract Color Image Features," *International Journal of Computer Science and Mobile Computing,* vol. 6, no. 12, pp. 7-14, Dec 2017.

[3] A. Y. Hendi, et al., "A Novel Simple and Highly Secure Method for Data Encryption-Decryption," *International Journal of Communication Networks and Information Security*, vol. 11, no. 1, pp. 232-237, Jan 2019.

[4] M. O. Al-Dwairi, et al., "An Efficient and Highly Secure Technique to Encrypt and Decrypt Color Images," *Engineering, Technology & Applied Science Research*, vol. 9, no. 3, pp. 4165-4168, Jun 2019.

[5] M. Heikkila and M. Pietikainen, "A texture Based Method for Modeling the Background and Detecting Moving Objects," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657-662, Apr 2006.

[6] M. Heikkila, et al., "Description of interest regions with local binary patterns," *Pattern Recognition*, vol. 42, no. 3, pp. 425-436, Mar 2009.

[7] Y. Lee, et al., "Hierarchical on-line boosting-based background subtraction," *2011 17th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, Ulsan, pp. 1-5, 2011.

[8] S. Liao, et al., "Dominant local binary patterns for texture Classication," *IEEE Transactions on Image Processing*, vol. 18, no. 5, pp. 1107-1118, May 2009.

[9] S. Liao, et al., "Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, pp. 1301-1306, 2010.

[10] Z. A. Alqadi and H. M. Elsayyed, "Window Averaging Method to Create a Feature Victor for RGB Color Image," *International Journal of Computer Science and Mobile Computing*, vol. 6, no.2, pp. 60-66, Feb 2017.

[11] J. Nader, et al., "Analysis of Color Image Filtering Methods," *International Journal of Computer Applications*, vol. 174, no. 8, pp. 12-17, Sep 2017.

[12] Z. A. Alqadi, et al., "True Color Image Enhancement Using Morphological Operations," *International Review on Computers & Software*, vol. 4, no. 5, pp. 557-562, Sep 2009.

[13] Z. A. Alqadi, et al., "ZJICD algorithm for JPEG image compression/decompression," *Digital Processing, Elixir Digital Processing*, vol. 94, pp. 40368-40374, 2016.

[14] M. O. Al-Dwairi, et al., "Optimized True-Color Image Processing," *World Applied Sciences Journal*, vol. 8, no. 10, pp. 1175-1182, 2010.

[15] Z. A. Alqadi, et al., "True Color Image Enhancement Using Morphological Operations," *International Review on Computers & Software*, vol. 4, no. 5, pp. 557-562, Sep 2009.

[16] H. Al-Otum and M. Al-Dwairi, "Image Compression Based on Color Morphological Pyramidal Decomposition," *9th International Symposium on Signal Processing and Its Applications*, Sharjah, pp. 1-4, 2007.

[17] R. S. A. Zneit, et al., "A Methodology to Create a Fingerprint for RGB Color Image," *International Journal of Computer Science and Mobile Computing*, vol. 6, no. 1, pp. 205-212, Jan 2017.

[18] Sajida P., et al., "Review on Local Binary Pattern (LBP) Texture Descritor and Its Variants," *International Journal of Advanced Research*, vol. 5, no. 5, pp. 708-717, May 2017.

[19] M. O. Al-Dwairi, et al., "A new method for voice signal features creation," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 5, pp. 4092-4098, Oct 2019.

[20] A. Fernandez, et al., "Evaluation of robustness against rotation of LBP, CCR and ILBP features in granite texture classification," *Machine Vision and Applications*, vol. 22, no. 6, pp. 913-926, Nov 2011.

[21] Z. Guo, et al., "A Completed Modeling of Local Binary Pattern Operator for Texture Classification," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657-1663, Jun 2010.

[22] A. Y. Hindi, et al., "A Novel Technique for Data Steganography," *Engineering, Technology & Applied Science Research*, vol. 9, no. 6, pp. 4942-4945, 2019.

[23] T. Song and H. Li, "WaveLBP based features for image classification," *Pattern Recognition Letters*, vol. 34, no. 12, pp. 1323-1328, Sep 2013.

[24] K. M. Matrouk, et al., "Investigation and Analysis of ANN Parameters," *European Journal of Scientific Research*, vol. 121, no. 2, pp. 217-225, Jan 2014.