A Novel Content-Based Image Indexing and Retrieval Framework Using Clockwise Local Difference Binary Pattern (CWLDBP)

M. Ravinder and M. Tirupathamma

Abstract In this chapter, we propose a novel content-based image indexing and retrieval framework based on clockwise local difference binary pattern. Local binary pattern (LBP) is a popular texture content-based image indexing and retrieval framework proposed by Ojala et al. [1]. In our proposed method, we find the pair-wise difference between the pixels in clockwise direction of 3 × 3 neighborhood and the result of difference is used to find the binary pattern. Three novel methods such as CWLDBP1, CWLDBP2, and CWLDBP3 are proposed. The main advantage of the proposed methods such as CWLDBP1 and CWLDBP2 is the use of few binary patterns compared to LBP. That is, our proposed methods CWLDBP1 and CWLDBP2 use only sixteen binary patterns to index an image. To test our proposed method, we use Corel 1k database. Our proposed method has shown reasonable results when compared to 3 × 3 neighborhood-based LBP rotational invariant, LBP uniform, and LBP uniform rotational invariant methods.

Keywords LBP · Local difference · Neighborhood · Pattern · Image · CWLDBP

1 Introduction

Increase in capability of data storage technology has witnessed a huge growth of multimedia content especially image repositories. The result of which is the need for an efficient content-based image indexing and retrieval framework. Here, the content means the features of image such as color, texture, and shape. The advantage of content-based image indexing is the automatic assignment of index to

M. Ravinder (✉)
JNTUK, Kakinada, AP, India
e-mail: ravinder.rsh@gmail.com

M. Tirupathamma
Department of ECE, JNTUHCEJ, Karimnagar, Telangana, India
e-mail: tirupathamma.jntu@gmail.com

S.S. Dash et al. (eds.), *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, Advances in Intelligent Systems and Computing 394,
DOI 10.1007/978-81-322-2656-7_93

an image based on content instead of using a textual keyword. The growth of new techniques based on content of image has shown tremendous results in solving the problem of retrieving relevant image information from huge repositories of images.

Local binary pattern popularly known as LBP is a texture content-based image indexing and retrieval framework proposed by Ojala et al. [1]. LBP is a simple and efficient framework based on extracting texture features from an image using a 3 × 3 neighborhood. LBP operator represents an image using two hundred and fifty six binary patterns.

Other variants of LBP are rotationally invariant LBP, and uniform LBP which is useful in representing the image efficiently with less number of feature vectors is compared to normal LBP.

Due to its effectiveness and simplicity, the LBP operator has been used in number of applications such as face recognition task [2], facial expression recognitions task [3], segmentation of texture [4], and texture-based classification [5–7].

Various extensions to the standard LBP have been proposed in the literature for image and face classification applications [8–13].

An extension of the LBP operator which is used for representing a texture video using volume local binary pattern (VLBP) has been proposed [10, 11].

The rest of this chapter is organized as follows: Sect. 2 discusses the LBP in detail; Sect. 3 describes our proposed novel frameworks; Sect. 4 discusses experiments and results; and Sect. 5 concludes the chapter.

2 Local Binary Pattern

Local binary pattern operator proposed by Ojala et al. [1] is mainly used for texture-based image classification and retrieval. The standard LBP operation is based on a 3 × 3 neighborhood. In this chapter, we mainly concentrate on a 3 × 3 neighborhood for proposing the new frameworks. The standard steps in finding the LBP code for an image are given as follows:

Step 1: After selecting a 3 × 3 neighborhood, threshold the eight neighbor pixels based on sign obtained by finding the difference between the neighbors and the center pixel values.

Step 2: Multiply the thresholding result with the binary weights and sum up all the values to get the binary value code for the center pixel.

Consider p_1, p_2, … p_7, p_8 are the neighbor pixels and p_c is the center pixel of a 3 × 3 neighborhood. Then, the LBP for the center pixel can be obtained by the following equation:

$$\text{LBP}(\text{Center pixel}) = \sum_{a=1}^{8} 2^{a-1} * f(p_a - p_c) \tag{1}$$

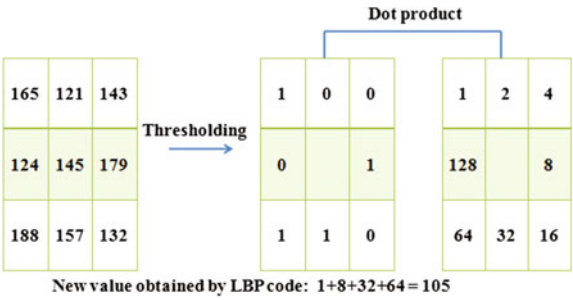

Fig. 1 Calculation of LBP values for center pixel of a 3 × 3 neighborhood

where $f(v)$ is the threshold function

$f(v) = 1$ if $v >= 0$,

$f(v) = 0$, otherwise. And p_c is the center pixel and p_a is the neighbor pixel.

The procedure of finding local binary pattern (LBP) is as shown in Fig. 1.

3 Proposed Method

In this section, we propose three novel clockwise local difference binary pattern algorithms meant for content-based image indexing and retrieval framework. Our framework is based on 3 × 3 neighborhood pixels of an image. Consider the following 3 × 3 neighborhood as shown in Fig. 2.

As shown in Fig. 2, P1, P2, … P9 are the positions of pixels in a 3 × 3 neighborhood. As explained earlier, our proposed method mainly focused on a 3 × 3 neighborhood.

3.1 Algorithm 1: For CWLDBP1

Step 1: In a 3 × 3 neighborhood of an image, find the difference between values of pixels positioned at P1 and P7, followed by P4 and P8, by P9 and P3, and by P6 and P2. That is

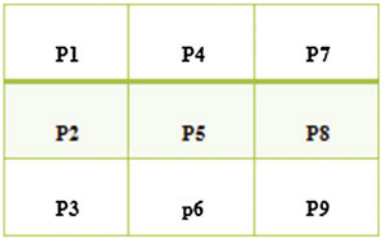

Fig. 2 An example of 3 × 3 neighborhood

V1 = pixel value at (P1) − pixel value at (P7)
V2 = pixel value at (P4) − pixel value at (P8)
V3 = pixel value at (P9) − pixel value at (P3)
V4 = pixel value at (P6) − pixel value at (P2)

Step 2: Threshold the values V1, V2, V3, and V4 using a threshold T. In our case, value of T is 0. That is, find out whether the values V1, V2, V3, and V4 are greater than zero or not. If the value is greater than zero, then the new value will become one otherwise the new value will become zero. For example

If V1 > 0
V1 = 1
Else
V1 = 0
End

Step 3: To the threshold resultant values vector [V1, V2, V3, V4] obtained in Step 2, we perform dot product with the vector [1, 2, 4, 8] and find out the sum of all to get the binary code for the center pixel P5.

The procedure followed in the proposed Algorithm 1 is as shown in Fig. 3 with an example of calculation of new value for center pixel at position P5 (in Fig. 2).

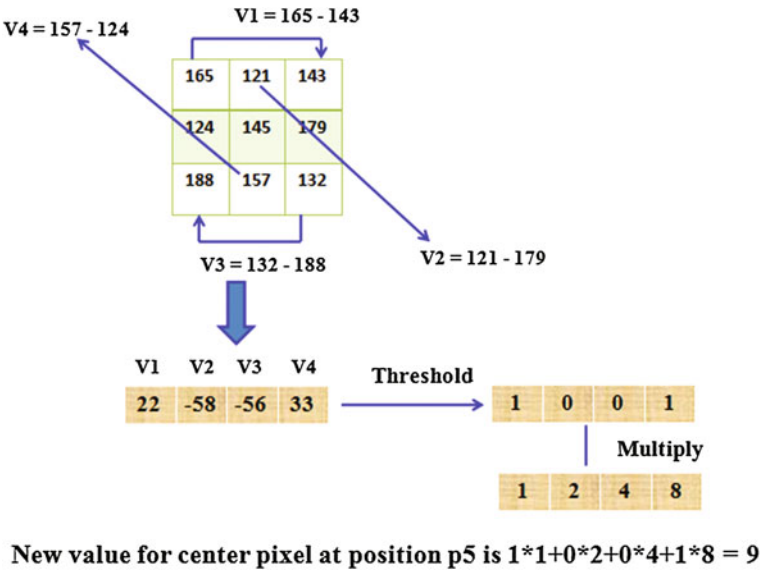

Fig. 3 Example calculation of center pixel value using Algorithm 1

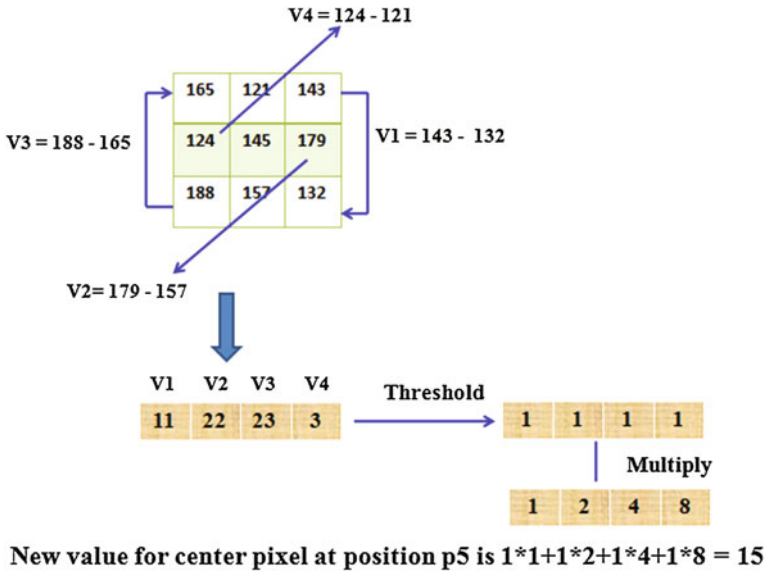

Fig. 4 Example calculation of center pixel value using Algorithm 2

3.2 Algorithm 2: For CWLDBP2

Step 1: In a 3 × 3 neighborhood of an image, find the difference between values of pixels positioned at P7 and P9, followed by P8 and P6, by P3 and P1, and by P2 and P4. That is

V1 = pixel value at (P7) − pixel value at (P9)
V2 = pixel value at (P8) − pixel value at (P6)
V3 = pixel value at (P3) − pixel value at (P1)
V4 = pixel value at (P2) − pixel value at (P4)

Step 2: Threshold the values V1, V2, V3, and V4 using a threshold T. In our case value of T is 0. That is, find out whether the values V1, V2, V3, and V4 are greater than zero or not. If the value is greater than zero, then the new value will become one otherwise the new value will become zero. For example

If V1 > 0
V1 = 1
Else
V1 = 0
End

Step 3: To the threshold resultant values vector [V1, V2, V3, V4] obtained in Step 2, we perform dot product with the vector [1, 2, 4, 8] and find out the sum of all to get the binary code for the center pixel P5.
The procedure followed in the proposed Algorithm 2 is as shown in Fig. 4 with an example of calculation of new value for center pixel at position P5.

3.3 *Algorithm 3: For CWLDBP3*

Step 1: In a 3 × 3 neighborhood of an image, find the difference between values of pixels positioned at P1 and P7, followed by P4 and P8, P9 and P3, P6 and P2, P7 and P9, P8 and P6, P3 and P1, and by P2 and P4. That is

V1 = pixel value at (P1) − pixel value at (P7)
V2 = pixel value at (P4) − pixel value at (P8)
V3 = pixel value at (P9) − pixel value at (P3)
V4 = pixel value at (P6) − pixel value at (P2)
V5 = pixel value at (P7) − pixel value at (P9)
V6 = pixel value at (P8) − pixel value at (P6)
V7 = pixel value at (P3) − pixel value at (P1)
V8 = pixel value at (P2) − pixel value at (P4)

Step 2: Threshold the values V1, V2, V3, V4, V5, V6, V7, and V8 using a threshold T. In our case value of T is 0. That is, find out whether the

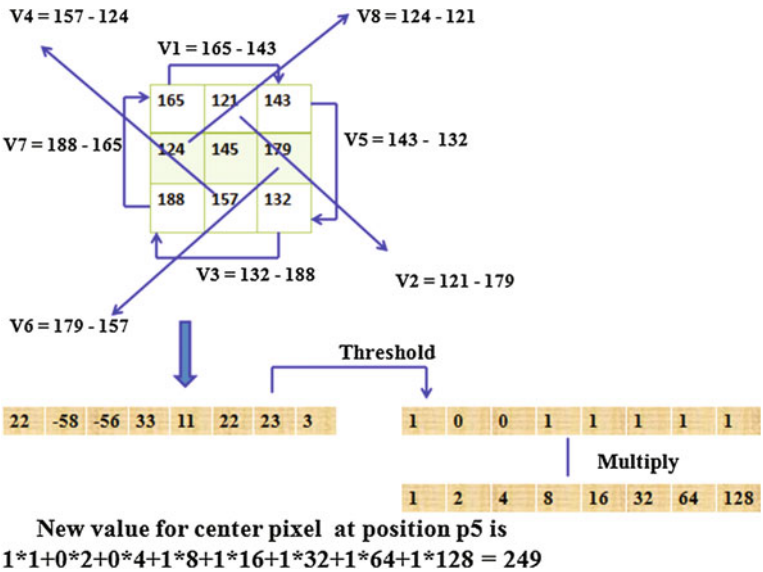

Fig. 5 Example calculation of center pixel value using Algorithm 3

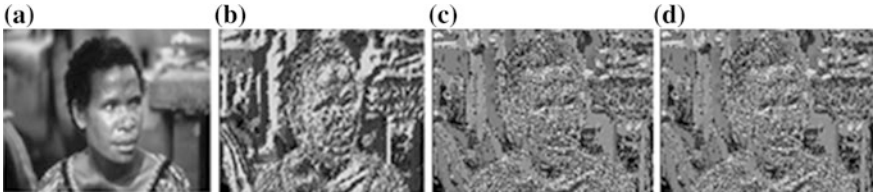

Fig. 6 **a** Gray scale image; **b** result of CWLDBP1; **c** result of CWLDBP2; **d** result of CWLDBP3

values V1, V2 ….V8 are greater than zero or not. If the value is greater than zero, then the new value will become one otherwise the new value will become zero. For example

- If V1 > 0
- V1 = 1
- Else
- V1 = 0
- End

Step 3: To the threshold resultant values vector [V1, V2, V3, V4, V5, V6, V7, V8] obtained in Step 2, we perform dot product with the vector [1, 2, 4, 8, 16, 32, 64, 128] and find out the sum of all to get the binary code for the center pixel at position P5.

The procedure followed in the proposed Algorithm 3 is as shown in Fig. 5 with an example of calculation of new value for center pixel at position P5.

An example gray scale image from Corel-1k dataset and the resultant images after applying the proposed algorithms are shown in Fig. 6.

4 Experiments and Results

In this section, we discuss experiments and results of our proposed algorithms. To evaluate the performance of our proposed algorithms, we have used the Corel 1k dataset which consists thousands of images consisting of ten different categories of images (Hundred images of each category). We compare our proposed algorithms with already-existing methods such as 3 × 3 neighborhood-based LBP Rotational Invariant (**LBP-ri**), LBP Uniform (**LBP-u2**), and LBP Uniform Rotational Invariant (**LBP-riu2**).

In our experiments for finding best matches for a given query image, we have used the *k-nearest neighbor search* algorithm.

The experimental results (Precision % based on top 10 resultant images retrieved for a query image) are shown in Table 1.

The experimental results (Recall % based on top 100 resultant images retrieved for a query image) are shown in Table 2.

Table 1 Results in terms of precision (n = 10) (%) for various categories (S. no) of images in Corel-1k dataset

S. no	LBP-ri	LBP-u2	LBP-riu2	CWLDBP1	CWLDBP2	CWLDBP1 + CWLDBP2	CWLDBP3
1	53.7	56.3	52.3	54.2	46.2	54.0	60.4
2	36.6	46.3	36.6	49.9	44.2	50.9	49.2
3	36.1	38.4	35	43.2	45.6	46.9	46.5
4	87.9	85.7	84.9	86.5	73.7	87.9	92.8
5	96.2	96.9	96.7	96.4	95.3	96.0	95.1
6	36.9	39.5	36.9	39.3	39.9	41.7	43.7
7	81	87	82.2	77.5	73.8	79.9	81.2
8	64.2	64.2	63.8	68.6	74.9	71.8	78.4
9	36.3	34.2	34.2	38.4	30.1	34.2	37.5
10	51.7	49.7	49.5	40.6	35.3	41.2	47.1
Avg	58.06	59.82	57.21	59.46	55.90	60.45	63.19

Table 2 Results in terms of recall (n = 100) (%) for various categories (S. no) of images in Corel-1k dataset

S. no	LBP-ri	LBP-u2	LBP-riu2	CWLDBP1	CWLDBP2	CWLDBP1 + CWLDBP2	CWLDBP3
1	33.00	33.18	32.73	35.72	27.18	34.85	35.05
2	22.19	24.05	21.95	28.33	22.80	26.87	25.12
3	19.18	19.19	18.81	19.92	23.44	21.90	22.04
4	50.38	46.88	47.13	53.07	47.19	55.82	57.41
5	68.95	64.04	65.88	84.42	84.24	84.89	81.16
6	19.94	20.36	19.47	21.54	20.29	22.20	22.99
7	40.5	44.04	42.4	38.49	34.83	39.20	39.10
8	33.51	33.17	32.07	42.32	45.50	45.27	47.36
9	24.55	20.44	23.90	21.99	15.31	19.56	20.36
10	26.51	24.31	25.87	21.46	19.23	22.92	24.43
Avg	33.87	32.96	32.03	36.72	34.00	37.34	37.50

In the above tables, various categories of images in Corel-1k are 1-Africans, 2-Beaches, 3-Buildings, 4-Buses, 5-Dinosaurs, 6-Elephants, 7-Flowers, 8-Horses, 9-Mountains, 10-Food, and AVG represents average

5 Conclusion

In this chapter, we have proposed three new algorithms based on a 3 × 3 neighborhood in an image, meant for content-based image indexing and retrieval framework. We have evaluated our proposed algorithms using Corel-1k dataset and our proposed algorithms have shown reasonable results in terms of precision and recall. In our future work, we will try to improve the performance of above-proposed algorithms with the help of Gabor filters.

References

1. Ojala T, Pietikainen M, Harwood D. A comparative study of texture measures with classification based on feature distributions. Pattern Recogn. 1996;29(1):51–9.
2. Ahonen T, Hadid A, Pietikäinen M. Face description with local binary patterns: application to face recognition. IEEE Trans Pattern Anal Mach Intell (PAMI). 2006;28:2037–41. doi:10.1109/TPAMI.2006.244.
3. Shan C, Gong S, McOwan PW (2005) Robust facial expression recognition using local binary patterns. In: Proceedings of the International Conference on Image Processing (ICIP), 2005, p. 370–3.
4. Ojala T, Pietikainen M. Unsupervised texture segmentation using feature distributions. Pattern Recogn. 1999;32(3):477–86. doi:10.1016/S0031-3203(98)00038-7.
5. Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell (PAMI). 2002;24(7):971–87. doi:10.1109/TPAMI.2002.1017623.
6. Topi M, Timo O, Matti P, Maricor S. Robust texture classification by subsets of local binary patterns. In: Proceedings of the International Conference on Pattern Recognition (ICPR), 2000, p. 935–8.
7. Topi M, Matti P, Timo O. Texture classification by multipredicate local binary pattern operators. In: Proceedings of the International Conference on Pattern Recognition (ICPR), 2000, p. 939–942.
8. Liao S, Law M, Chung A. Dominant local binary patterns for texture classification. IEEE Trans Image Process. 2009;18(5):1107–18.
9. Zhao G, Pietikainen M. Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Trans PAMI 2007;29(6):915–928.
10. Zhang S, Yao H, Liu S. Dynamic background modeling and subtraction using spatio-temporal local binary patterns. In: ICIP, 2008, p.1556–9.
11. Guo Z, Zhang L, Zhang D. Rotation invariant texture classification using LBP variance (LBPV) with global matching. Pattern Recogn. 2010;43(3):706–19.
12. Xie S, Shan S, Chen X, Gao W. V-LGBP: volume based local Gabor binary patterns for face representation and recognition. In: Proceedings of the ICPR, 2008.
13. Maenpaa T, Pietikainen M. Multi-scale binary patterns for texture analysis. In: Proceedings of the SCIA, 2003.