A MODIFIED LBP METHOD TO EXTRACT FEATURES FROM COLOR IMAGES

BILAL ZAHRAN* JAMIL AL-AZZEH * ZIAD ALQADI* MOHD–ASHRAF AL–ZOGHOUL* SALEH KHAWATREH#

*Department of Computer Engineering, Faculty of Engineering Technology, Al-Balqa Applied University, Jordan
# Faculty of Engineering, Al Ahliyyah Amman University, Jordan

ABSTRACT

Local image texture descriptors are widely used in image analysis. The local binary pattern (LBP) is a texture descriptor that is simple and efficient. LBP has been utilized in many applications in image processing field such as face recognition, pattern recognition and feature extraction. In this paper, a modified LBP method was proposed to extract texture features. The proposed algorithm was implemented on many digital 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.

Keywords: Local Binary Patterns (LBP), Local Features, Cyclic Symmetric Reduced LBP (CSLBP), Mean Square Error (MSE) And Peak Signal-To-Noise Ratio (PSNR).

1. INTRODUCTION

An image texture is a set of metrics calculated in image processing designed to quantify the perceived texture of an image. Image texture gives us information about the spatial arrangement of color or intensities in an image or selected region of an image [1]. Texture analysis is a technique for evaluating the position and intensity of signal features, that is, pixels, and their gray level intensities. The distribution of these pixels can be computed to produce mathematical parameters which characterize the texture type and thus the underlying structure of the objects shown in the image; these values are also known as texture features. Real world image textures are often not uniform due to many different variations such as illumination conditions and arbitrary spatial rotations constantly. Basically the methods of textures analysis are categories into four types [2]: (1) Structural, (2) Statistical, (3) Model based and (4) Transforms.

Structural: These methods represent texture by primitive patterns which are regular in appearance and systematically located on the surface. Statistical: they represent the texture by non-deterministic properties of image pixels and regions which are usually natural and consist on randomly distributed surface elements.

Model: The model based texture analysis methods are combination of fractal and stochastic models.

Transforms: In transform based methods the image is represented according to the coordinate system. For example Fourier, Wavelet and Gabor transforms interpreted the closely related texture characteristics from their coordinate system. A major problem in early texture classification algorithms, which concentrate on the statistical analysis of images, is that they impose lots of constraints on texture in order to give quality results. This shows how crucially good descriptors are needed and in demand [3-5].

The local binary pattern (LBP) texture analysis operator is defined as a gray-scale invariant texture measure, derived from a general definition of texture in a local neighborhood. LBP was originally proposed for texture analysis [5, 6]. LBP summarizes local structures of images efficiently by comparing each pixel with its neighboring pixels. LBP operator transforms an
image into an array or image of integer labels describing small-scale appearance of the image. These labels or their histogram, are then used for further image analysis. The most widely used versions of the operator are designed for monochrome still images but it has been extended also for color (multi-channel) images as well as videos and volumetric data. The most important properties of LBP are its tolerance regarding monotonic illumination changes and its computational simplicity.

The LBP operator with several variations has been made into a really powerful measure of image texture, showing good results in terms of accuracy and computational complexity in many empirical studies. The LBP operator unifies the traditionally divergent statistical and structural models of texture analysis. It has been widely utilized in many applications, for instance, face image analysis [7], image and video retrieval [8, 9], environment modeling [10], visual inspection [11], motion analysis [12], biomedical [13], and so forth. Recently, there are several variations of LBP methodologies to enhance performance in different applications. These variations concentrate on different aspects of the basic LBP operator such as improvement of its discriminative capability, improvement of its robustness, selection of its neighborhood, extension to 3D data, combination with other approaches and so forth.

In this paper, a modified LBP method was proposed to reduce the size of the features extracted from the image and to enhance the speed and accuracy of the classifier.

2. RELATED WORKS

Despite the success of the basic LBP in image processing applications, its underlying working mechanism went under more investigation. Several variants of the basic LBP algorithm have been developed to improve its characteristics.

In [14], the dominant local binary patterns (DLBP) are proposed as a texture classification approach. The DLBP approach is able to represent the dominant patterns in the texture images. Furthermore, it retains the rotation invariant and histogram equalization invariant properties of the conventional LBP approach. It is simple and computationally efficient. The results drawn from experiments show that it has good texture classification. The global features extracted by using the circularly symmetric Gabor filter responses encapsulate the spatial relationships between distant pixels. The features extracted are rotation invariant and less sensitive to histogram equalization. They, therefore, complement with the DLBP local features. Guo et al. developed an adaptive LBP (ALBP) [15] by incorporating the directional statistical information for rotation invariant texture classification. In [2] researchers concluded that no single descriptor variants has been developed which covers all the limitations such as illumination changes, rotation, scaling and bur.

In [16] Guo et al. proposed a hybrid LBP scheme to better exploit the local and global information in texture images. The principal orientations of the texture image were first estimated and then the LBP histograms can be aligned. These histograms were in turn used to measure the dissimilarity between images. LBP variance (LBPV) was proposed as a new descriptor to improve the performance of LBP by exploiting the local contrast information. Finally, a feature size reduction method was proposed to speed up the matching scheme. The experimental results on two large databases showed that the proposed global rotation invariant matching scheme with LBPV feature leads to much higher classification accuracy than traditional rotation invariant LBP.

The aim of [17] was to conduct a performance evaluation where several texture descriptors such as LBP, Coordinated Clusters Representation (CCR) and Improved Local Binary Patterns (ILBP) are applied for granite texture classification. Among the three texture models, the ILBP provided significantly higher results, whereas the performances obtained with the LBP and the CCR are comparable. The analysis of robustness against rotation generated different outcomes, depending on the method used to rotate the images. In [18], a completed modeling of the LBP operator is proposed and an associated completed LBP (CLBP) scheme is developed for texture classification. The researchers analyzed LBP from a viewpoint of local difference sign-magnitude transform (LDSMT). Three operators, CLBP_C, CLBP_S and CLBP_M, were defined to extract the image local gray level, the sign and magnitude features of local difference, respectively. The researchers demonstrated that the sign component is more important than the magnitude component in preserving the local difference information.

Video texture synthesis is the process of providing a continuous and infinitely varying stream of frames. Guo et al.[19] proposed a variant of LBP-TOP called Multi-frame LBP-
TOP to find the most appropriate matching pairs of frames for video texture synthesis. Video textures characterize objects and visual patterns in a dynamic manner. The analysis of dynamic textures can provide useful hints to many applications such as video compression and retrieval. In [20] a variant of the Local Binary Patterns method based on fuzzy model was presented. The presented generalised Fuzzy Binary Patterns model was applied to the classic Local Binary Patterns method as well as to the Local Binary Patterns with Contrast measure method. Supervised classification experiments were conducted on several of natural and medical texture images, degraded by different types and intensities of additive noise. The method based on fuzzy outperform the methods based on the classic Binary Patterns model for all types of images and noise, indicating the efficiency of fuzzy modelling in coping with the uncertainty introduced to texture due to noise.

Zhao et al. [21] proposed an approach to compute rotation invariant features from histograms of local, noninvariant patterns. They applied this approach to both static and dynamic LBP descriptors. For static textures, they presented Local Binary Pattern Histogram Fourier features (LBPHF). Song and Li [22] combined wavelet transform and LBP. They build up the image description using a hierarchical framework based on low dimensional WaveLBP features, which not only extracts multiscale oriented features and local image patterns, but also captures multilevel (pixel, patch, image) features. Qi et al. [23] introduced a pairwise rotation invariant co-occurrence LBP (CoLBP), which incorporates two types of context: spatial co-occurrence and orientation co-occurrence. The method aims to preserve the relative angle between the orientations of individual features. The proposed method aims to preserve relative angle information between co-occurrence pairs. The experimental results show that the proposed feature effectively captures local curvature information and demonstrates a great rotation invariant property.

3. LOCAL BINARY PATTERNS

3.1 Histogram

Gray images [1] and color images can be represented by histogram, which is any array, whose elements values specify the repetition of each color value [4, 5]. Fig.1 shows the distribution of each color for the color image peppers.png [9]:

Histogram can be used to identify the image, by using the array of color repetition as an identifier to recognize the image, here we will decrease the number of comparisons comparing to the number of comparisons when we match the images pixel by pixel, Table 1 summarize the results of comparisons for some images. Here for the gray images we reduce the matching time using histogram 1.6049 times and for color images 5.6715 times.

3.2 LBP

The original LBP operator labels the pixels of an image with decimal numbers, called Local Binary Patterns or LBP codes, which encode the local structure around each pixel. The operator assigns a label to every pixel of an image by thresholding the 3x3-neighborhood of each pixel (Fig. 2) with the center pixel value and considering the result as a binary number.

<table>
<thead>
<tr>
<th>I-1, j-1</th>
<th>A(I-1, j)</th>
<th>A(I-1, j+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3)</td>
<td>(4)</td>
<td>(4)</td>
</tr>
<tr>
<td>A(I, j-1)</td>
<td>A(I, j)</td>
<td>A(I, j+1)</td>
</tr>
<tr>
<td>(3)</td>
<td>(3)</td>
<td>(1)</td>
</tr>
<tr>
<td>A(I+1, j-1)</td>
<td>A(I+1, j)</td>
<td>A(I+1, j+1)</td>
</tr>
<tr>
<td>(2)</td>
<td>(2)</td>
<td>(1)</td>
</tr>
</tbody>
</table>

Figure 2: Pixel neighbors

Each pixel is compared with its eight neighbors in a 3x3 neighborhood by subtracting the center
pixel value; The resulting strictly negative values are encoded with 0 and the others with 1; A binary number is obtained by concatenating all these binary codes in a clockwise direction starting from the top-left one and its corresponding decimal value is used for labeling (see Fig.3) The derived binary numbers are referred to as Local Binary Patterns or LBP codes. This method does not reduce the number of values and the range still between 0 and 255.

Figure 3: Calculating the Basic LBP

For an illustration of the basic LBP operator. An example of an LBP image and histogram are shown in Fig. 4.

3.2.1 Cyclic Symmetric reduced LBP (CSLBP)

Some modification has been done [13, 14] by generating only 16 values by using cyclic symmetric reduced LBP (CSLBP) and the values in the feature array can be reduced to 16, each of these value, for each pixel can be calculated as shown in Fig. 5. Using CSLBP method we can reduce the histogram values to 48 values (16 values for each color), and if convert the color image matrix from 3 dimensional matrix to two dimensional matrix we can reduce the values to 16. Table 2 show the features array for the image peppers.png generated as a results of implementing CSLBP.

<table>
<thead>
<tr>
<th>Value</th>
<th>Features using CSLBP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>77999</td>
</tr>
<tr>
<td>1</td>
<td>51903</td>
</tr>
<tr>
<td>2</td>
<td>11870</td>
</tr>
<tr>
<td>3</td>
<td>82931</td>
</tr>
<tr>
<td>4</td>
<td>19215</td>
</tr>
<tr>
<td>5</td>
<td>15272</td>
</tr>
<tr>
<td>6</td>
<td>7938</td>
</tr>
<tr>
<td>7</td>
<td>61518</td>
</tr>
<tr>
<td>8</td>
<td>101840</td>
</tr>
<tr>
<td>9</td>
<td>6810</td>
</tr>
<tr>
<td>10</td>
<td>15032</td>
</tr>
<tr>
<td>11</td>
<td>11271</td>
</tr>
<tr>
<td>12</td>
<td>66610</td>
</tr>
<tr>
<td>13</td>
<td>5208</td>
</tr>
<tr>
<td>14</td>
<td>27902</td>
</tr>
<tr>
<td>15</td>
<td>23181</td>
</tr>
</tbody>
</table>

The feature array is unique and there is only one array for each image, and it is sensitive for any changes in the image. If we change the pixel image (100, 200, 3) from 154 to 112 this will be reflected on the feature array with some changes as shown in Table 3.

Table 3: Features array before and after adjustment

<table>
<thead>
<tr>
<th>Value</th>
<th>Features before adjustment</th>
<th>Features after adjustment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>77999</td>
<td>77999</td>
</tr>
<tr>
<td>1</td>
<td>51903</td>
<td>51903</td>
</tr>
<tr>
<td>2</td>
<td>11870</td>
<td>11870</td>
</tr>
<tr>
<td>3</td>
<td>82931</td>
<td>82932</td>
</tr>
<tr>
<td>4</td>
<td>19215</td>
<td>19215</td>
</tr>
<tr>
<td>5</td>
<td>15272</td>
<td>15272</td>
</tr>
<tr>
<td>6</td>
<td>7938</td>
<td>7938</td>
</tr>
<tr>
<td>7</td>
<td>61518</td>
<td>61517</td>
</tr>
<tr>
<td>8</td>
<td>101840</td>
<td>101840</td>
</tr>
<tr>
<td>9</td>
<td>6810</td>
<td>6810</td>
</tr>
<tr>
<td>10</td>
<td>15032</td>
<td>15032</td>
</tr>
<tr>
<td>11</td>
<td>11271</td>
<td>11271</td>
</tr>
<tr>
<td>12</td>
<td>66610</td>
<td>66610</td>
</tr>
<tr>
<td>13</td>
<td>5208</td>
<td>5208</td>
</tr>
<tr>
<td>14</td>
<td>27902</td>
<td>27902</td>
</tr>
<tr>
<td>15</td>
<td>23181</td>
<td>23181</td>
</tr>
</tbody>
</table>

The researchers in [2] explained the advantages and disadvantages of LBP variants. These advantages and disadvantages are summarized in Fig. 6.
4. MODIFIED LOCAL BYTE PATTERN (MLBP)

The proposed method implementation steps:

1- Get the original image (if the image is a color one, change the 3 dimensional matrices representing the color image to two dimensional matrices).

2- Retrieve the number of rows and columns of the matrix.

3- Initialize features array (FA) to zeros.

4- For each pixel do the following to compute MLBP:

A. Find a threshold value (T) using the neighbors as shown in Figure (2)

\[ T = \frac{(I(i,j+1)+I(i-1,j)+I(i,j-1)+I(i+1,j)+I(i+1,j+1)+I(i+1,j-1)+I(i-1,j+1)+I(i-1,j-1)-8*I(i,j))}{9}; \]

B. Calculate a and b as follows:

\[ a = ((I(i,j+1)+I(i+1,j)-I(i,j-1)-I(i-1,j)) > T) \times 2^0; \]

\[ b = ((I(i+1,j+1)+I(i+1,j-1)-I(i-1,j+1)-I(i-1,j-1)) > T) \times 2^1; \]

C. Compute c = a + b.

D. Add 1 to the feature array index c:

\[ FA(c+1) = FA(c+1) + 1. \]

5- Save FA.

We've conducted several experiments on color images using our proposed method MLBP. Table 4 shows some samples of the results. Table 5 shows the time needed to extract the features and the time needed to use the features to identify the image. The same samples were treated using CSLBP and Table 6 shows the results of implementation. From Tables 5 and 6 we can find the advantage of the proposed MLBP method over CSLBP by calculating the efficiency of the proposed method using the speed up parameter, the results of calculation the speed up are shown in Table 7. So the proposed MLBP method is in average faster than CSLBP method with approximately 2.6 times. The proposed MLBP method is sensitive to any changes in the original color image and the changes can be reflected to the calculated mean square error (MSE) (define in equation 1) and peak signal-to-noise ratio (PSNR) (define in equation 2) parameters [24], Table (8) shows the results of implementing a change on an image.

\[ MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I(i,j) - K(i,j))^2 \]  

Where I: noise-free \( m \times n \) monochrome image, K: approximation noisy image

\[ PSNR = 10 \log_{10} \left( \frac{MAX}{10} \right) - \frac{10 \log_{10}(MSE)}{2} \]

Where \( MAX \): the maximum possible pixel value of the image.

<table>
<thead>
<tr>
<th>Features</th>
<th># of changed pixels</th>
<th>MSE</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>342649</td>
<td>0</td>
<td>0</td>
<td>Infinite</td>
</tr>
<tr>
<td>167651</td>
<td>166693</td>
<td>247515</td>
<td></td>
</tr>
<tr>
<td>342648</td>
<td>1</td>
<td>5.6955e-004</td>
<td>185.5319</td>
</tr>
<tr>
<td>167651</td>
<td>166694</td>
<td>247515</td>
<td></td>
</tr>
<tr>
<td>342649</td>
<td>5</td>
<td>0.0025</td>
<td>170.6542</td>
</tr>
<tr>
<td>167650</td>
<td>166692</td>
<td>247517</td>
<td></td>
</tr>
<tr>
<td>342649</td>
<td>10</td>
<td>0.0059</td>
<td>162.2246</td>
</tr>
<tr>
<td>167653</td>
<td>166693</td>
<td>247513</td>
<td></td>
</tr>
<tr>
<td>342650</td>
<td>15</td>
<td>0.0085</td>
<td>158.5247</td>
</tr>
<tr>
<td>167653</td>
<td>166691</td>
<td>247514</td>
<td></td>
</tr>
</tbody>
</table>
5. CONCLUSIONS

A modified method for the basic LBP algorithm was proposed. The aim of the proposed method (MLBP) is to extract the features of the images with better characteristics. MLBP was applied to several images with different types and sizes and compared to other traditional methods. After analyzing the results we can conclude the following:

- MLBP method is suitable for any image regardless of its size.
- The image features were minimized to 4 values.
- Image features are unique and not repeated.
- Features can be used to identify and distinguish the image.

The comparison results with other methods demonstrated the efficiency of the proposed method. In future, we should combine our method with others to enhance the overall characteristics.

REFERENCES


Table 1: Efficiency of using histogram

<table>
<thead>
<tr>
<th>Image type</th>
<th>Size</th>
<th>Time to do the matching pixel by pixel (sec.)</th>
<th>Time to do the matching using histogram (sec.)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gray</td>
<td>291*240</td>
<td>0.13000</td>
<td>0.081000</td>
<td>0.13000/0.081000 = 1.6049</td>
</tr>
<tr>
<td>Color</td>
<td>384<em>512</em>3</td>
<td>1.174000</td>
<td>0.207000</td>
<td>1.174000/0.207000 = 5.6715</td>
</tr>
</tbody>
</table>

Figure 4: Example of a basic LBP: a: input image, b: the corresponding LBP image and c: histogram

Figure 5: Calculating CSLBP
Figure 6: Advantages and Disadvantages of LBP and its variants [2]
Table 4: Color image features using MLBP

<table>
<thead>
<tr>
<th>Image #</th>
<th>Size</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>512 384 3</td>
<td>102629 96197 101054 286620</td>
</tr>
<tr>
<td>2</td>
<td>320 256 3</td>
<td>51010 50592 52452 89534</td>
</tr>
<tr>
<td>3</td>
<td>274 184 3</td>
<td>36237 17025 19725 76613</td>
</tr>
<tr>
<td>4</td>
<td>975 750 3</td>
<td>218781 163713 181310 1623500</td>
</tr>
<tr>
<td>5</td>
<td>450 320 3</td>
<td>55574 62402 62994 248214</td>
</tr>
<tr>
<td>6</td>
<td>600 516 3</td>
<td>247515 166693 167651 342649</td>
</tr>
<tr>
<td>7</td>
<td>251 201 3</td>
<td>30954 28329 28550 61816</td>
</tr>
<tr>
<td>8</td>
<td>236 214 3</td>
<td>21673 31872 31712 64503</td>
</tr>
</tbody>
</table>

Table 5: Calculated times using MLBP

<table>
<thead>
<tr>
<th>Image #</th>
<th>Size in bytes</th>
<th>Feature extraction time(sec.)</th>
<th>Image identifying time(sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>589824</td>
<td>1.269000</td>
<td>0.110000</td>
</tr>
<tr>
<td>2</td>
<td>245760</td>
<td>0.723000</td>
<td>0.093000</td>
</tr>
<tr>
<td>3</td>
<td>151248</td>
<td>0.556000</td>
<td>0.091000</td>
</tr>
<tr>
<td>4</td>
<td>2193750</td>
<td>4.060000</td>
<td>0.149000</td>
</tr>
<tr>
<td>5</td>
<td>432000</td>
<td>1.082000</td>
<td>0.105000</td>
</tr>
<tr>
<td>6</td>
<td>928800</td>
<td>1.683000</td>
<td>0.127000</td>
</tr>
<tr>
<td>7</td>
<td>151353</td>
<td>0.558000</td>
<td>0.091000</td>
</tr>
<tr>
<td>8</td>
<td>151512</td>
<td>0.524000</td>
<td>0.091000</td>
</tr>
</tbody>
</table>

Average for the sample: 605530 1.3069 0.1071

Table 6: Calculated times using CSLBP

<table>
<thead>
<tr>
<th>Image #</th>
<th>Size in bytes</th>
<th>Feature extraction time(sec.)</th>
<th>Image identifying time(sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>589824</td>
<td>0.527000</td>
<td>0.285000</td>
</tr>
<tr>
<td>2</td>
<td>245760</td>
<td>0.396000</td>
<td>0.269000</td>
</tr>
<tr>
<td>3</td>
<td>151248</td>
<td>0.364000</td>
<td>0.262000</td>
</tr>
<tr>
<td>4</td>
<td>2193750</td>
<td>0.987000</td>
<td>0.328000</td>
</tr>
<tr>
<td>5</td>
<td>432000</td>
<td>0.463000</td>
<td>0.277000</td>
</tr>
<tr>
<td>6</td>
<td>928800</td>
<td>0.617000</td>
<td>0.301000</td>
</tr>
<tr>
<td>7</td>
<td>151353</td>
<td>0.367000</td>
<td>0.267000</td>
</tr>
<tr>
<td>8</td>
<td>151512</td>
<td>0.373000</td>
<td>0.269000</td>
</tr>
</tbody>
</table>

Average for the sample: 605530 0.5118 0.2823
Table 7: Comparison of Speed up of CSLBP and MLBP

<table>
<thead>
<tr>
<th>Image #</th>
<th>Image identifying time(sec.) using CSLBP</th>
<th>Image identifying time(sec.) using MLBP</th>
<th>Speed up</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.285000</td>
<td>0.285000</td>
<td>2.5909</td>
</tr>
<tr>
<td>2</td>
<td>0.269000</td>
<td>0.269000</td>
<td>2.8925</td>
</tr>
<tr>
<td>3</td>
<td>0.262000</td>
<td>0.262000</td>
<td>2.8791</td>
</tr>
<tr>
<td>4</td>
<td>0.328000</td>
<td>0.328000</td>
<td>2.2013</td>
</tr>
<tr>
<td>5</td>
<td>0.277000</td>
<td>0.277000</td>
<td>2.6381</td>
</tr>
<tr>
<td>6</td>
<td>0.301000</td>
<td>0.301000</td>
<td>2.3701</td>
</tr>
<tr>
<td>7</td>
<td>0.267000</td>
<td>0.267000</td>
<td>2.9341</td>
</tr>
<tr>
<td>8</td>
<td>0.269000</td>
<td>0.269000</td>
<td>2.9560</td>
</tr>
<tr>
<td>Average for the sample</td>
<td>0.2823</td>
<td>0.2823</td>
<td>2.6359</td>
</tr>
</tbody>
</table>