A Fuzzy Texture Descriptor Using Combined Neighborhood Differences

Khairul Muzzammil Saipullah, Nuraishah Sarimin, Nurul Atiqah Ismail

Abstract—Texture information is a main element in object recognition and classification. To perform an accurate classification, the texture feature used in the classification must be highly discriminative. This paper presents a fuzzy texture descriptor using the Combined Neighborhood Differences (CND) called as the Fuzzy Combined Neighborhood Differences (FCND) which is insensitive to noise. Noise insensitive texture is a challenging task in texture analysis because noise itself can be detected as texture and the combination between texture and noise will produce a new texture. There are two types of the local neighborhood differences, the neighborhood surrounding differences and neighborhood centralized differences are combined and fuzzy thresholding in 8-bit binary code. Due to the implementation of fuzzy thresholding, the histogram becomes smoother and less sensitive to small changes in texture image. The texture classification accuracies using Brodatz and Outex texture database show FCND achieve an average 96% of accuracy, whereas those of local binary patterns, local neighborhood differences and Gabor filter are 90%, 92%, and 85% respectively.

Keywords—Texture features, Fuzzy Combined Neighborhood Differences, Local Binary Patterns, Local Neighbor Differences.

I. INTRODUCTION

Texture analysis is a basic issue in image processing and plays an important factor in visual perception and discrimination of image content. In some of the practical applications, it is difficult to ensure that obtained images have the same translation, rotation or scaling. This requires that texture analysis should be invariant to viewpoint, as it is always perceived as the same texture image by a human observer.

The most important things in texture analysis are the texture itself that represented by the coarseness and statistical characteristics of the local variation of brightness between neighboring pixels. Texture can be modeled by basic texture primitives that form texture elements, called textons [1] or texels [2]. An effective texture feature must be able to describe the textons in texture images. This can be done by extracting the texture feature locally because textons are determined by the spatial relations between neighboring pixels.

Over the years, there are a lot of studies regarding texture feature extraction and texture descriptor. Among the most popular texture descriptors are the Gabor wavelet [3] and local binary pattern (LBP) [4]. The Gabor representation has been shown to be optimal in the sense of minimizing the joint two dimensional uncertainties in space and frequency. The Gabor filters can be considered as orientation and scale tunable edge and line (bar) detectors, and the statistics of these micro-features in a given region are often used to characterize the underlying texture information. Zhou et al. [5] proposed a texture descriptor by using the magnitude of the 1D local Fourier transform with a 3x3 local window named Local Fourier Histogram (LFH). K. Muzzammil et al. [6] utilize the phase of the differences between the neighboring pixels and the center pixel to extract texture feature which is invariant to image blurring, scale changes and illumination changes. Besides that, LBP able to gain increasing attention due to its simplicity and excellent performance in various texture and face image analysis tasks.

However, LBP only consider the differences between the neighboring pixels and the center pixel of a local neighborhood. The differences between each of the neighboring pixels produce more discriminative information. K. Muzzammil et al. [7] proposed Combined Neighborhood Differences (CND) based on the differences between each of the neighboring pixels named Local Neighbors Differences (LND) [8]. The differences between each of the neighboring pixels are thresholded into 8-bit binary code. A binomial factor is assigned to the binary code in order to form a unique CND value which represents the texture feature of the local neighborhood. The distribution of this CND value is then computed to form a 256 dimensional histogram.

In this paper, we propose a new method by applying a Fuzzy Logic with CND and namely as Fuzzy Combined Neighborhood Differences (FCND). The FCND is based on the combination of neighborhood surrounding and centralized differences and the Fuzzy logic is used to threshold the part of CND.

The remainder of this paper is organized as follows: in Section 2, related work of the current research is presented; in Section 3, detailed construction of the FCND method is presented; experimental studies and evaluations are described in Section 4 and finally conclusions and future works are given in Section 5.
II. RELATED WORKS

A. Local Binary Pattern

Ojala et al. [4] proposed a robust way for describing pure local binary patterns (LBP) of texture in an image. The original 3x3 neighborhood (Fig. 1a) is thresholded by the value of the center pixel. If the neighboring pixel values are larger than or equal to the center pixel, the values are set to 1, otherwise they are set to 0. The values of the pixels in the thresholded neighborhood (Fig. 1b) are then multiplied by the weights given to the corresponding pixels (Fig. 1c). The results for this example are shown in Fig. 1d. Finally, the values of the eight pixels are summed to obtain the number 169 for this texture unit. The LBP method is invariant to gray scales and the enhanced version of LBP [9] implements circular neighborhoods instead of square neighborhoods.

![Fig. 1 Basic LBP algorithm](image)

B. Local Neighbors Differences

Fig. 2 shows the detail explanation of LND algorithm is use 3x3 neighborhood. First the images from the neighboring pixels of 3x3 neighborhoods are extracted. Then the differences between each neighbor is computed and threshold to 8-bit binary. A binomial factor is assigned as 2 to transform the 8-bit binary code into a unique LDN number. The LDN is centralized neighborhood differences. Two types of combinations have been tested. The first one is by using the thresholded neighborhood (Fig. 1b) are then multiplied by the weights given to the corresponding pixels (Fig. 1c). The results for this example are shown in Fig. 1d. Finally, the values of the eight pixels are summed to obtain the number 169 for this texture unit. The LBP method is invariant to gray scales and the enhanced version of LBP [9] implements circular neighborhoods instead of square neighborhoods.

![Fig. 1 Basic LBP algorithm](image)

C. Combined Neighbors Differences

The texture descriptor CND is based on the combination of difference values used in LBP and LND, namely the centralized neighborhood differences. Two types of combinations have been tested. The first one is by using the phase between the difference values of LND and LBP as implemented in [10]. The second one is to threshold differences values of LND against the differences values of LBP. The result shows the thresholding of LND and LBP generated more discriminative information compared to that of the phase.

The detail algorithm of CND is shown in Fig. 3. For each 3x3 window of an image, two neighborhood differences which are the centralized \( v(n) \) and surrounding neighborhood \( d(n) \) differences are calculated. \( v(n) \) and \( d(n) \) can be calculated using the following formula:

\[
v(n) = x(n) - x\left(\left\{(n + 9) \mod N\right\}, n = 0,1,\ldots, N - 1\right)
\]

\[
d(n) = c - x(n), n = 0,1,\ldots, N - 1
\]

where \( N \) is the number of neighbors which is eight for 3x3 neighborhood and \( c \) is the center pixel of the window. Then the differences between \( v(n) \) and \( d(n) \) are thresholded against zero in order to convert the difference values into binary codeword, given by

\[
p(n) = \begin{cases} 1, & \text{if } \{v(n) - d(n)\} > 0 \\ 0, & \text{otherwise} \end{cases}, n = 0,2,\ldots,7
\]

The next step is to create a unique value from the binary codeword by assigning binomial factor of 2 for each \( p(n) \) and it can be calculated using the following formula:

\[
CND = \sum_{k=0}^{7} p(k)2^k
\]

Based on (4), this CND is a decimal value between 0 and 255 resulting from the 8-bit binary code. Next, a histogram is constructed with 256 dimensions using the CND codes and the histogram denotes the distribution. Finally, the texture descriptor is obtained from the histogram. The starting point of \( d(n) \) and pixel gap of \( v(n) \) play an important role in generating informative texture feature of CND. It is because, the wrong location of starting point and pixel gap will result in the same information retrieved by LBP or LND. The reason we chose \( x(0) \) for the starting point of \( d(n) \) and 1 as the pixel gap of \( v(n) \) is because they generated highest histogram entropy based on the evaluation on Brodatz texture database [11].

III. PROPOSED METHOD

A. Fuzzy Logic

Fuzzy Logic is a multi-valued logic that allows intermediate values to be defined between conventional evaluations like true/false, yes/no, and high/low. Notions like rather tall or very fast can be formulated mathematically and processed by computers, in order to apply a more human-like way of thinking in the programming of computers [12].

The central notion of fuzzy systems is that truth values (in fuzzy logic) or membership values (in fuzzy sets) are indicated by a value in the range \([0.0, 1.0]\), with 0.0 representing absolute Falseness and 1.0 representing absolute Truth. For example, the possible interferometric coherence values are the set \( X \) of all real numbers between 0 and 1.

B. Fuzzy Combined Neighborhood Difference

In order to enhance the CND approach so as to cope with the uncertainty introduced by the speckle noise, we have considered fuzzy logic, as a means to cope with inexactness and improve the discrimination power of the CND approach in noise degraded images. Fuzzy logic resembles human decision making, with ability to finding precise solutions in approximate datasets. The fuzzification of the CND approach includes the transformation of the input variables to respective fuzzy variables, according to a set of fuzzy rules. To that direction, we introduce fuzzy rules to describe the relation between the intensity values of the central pixel with the neighborhood surrounding of a 3x3 neighborhood. As a membership function \( p_1(n) \) we consider the decreasing function (Fig. 5) defined as follows:
\[ p_1(n) = \begin{cases} 
0, & z(n) < T \\
\frac{1 + z(n)}{2}, & -T \leq z(n) \leq T \, , \, z(n) = v(n) - d(n) \\
1, & z(n) > T 
\end{cases} \quad (5) \]

On the other hand, membership function \( p_0(n) \) defines the degree to which \( p_i \) has a greater value than \( p_{\text{center}} \), and thus the degree to which \( d_i \) is 1. The membership function considered is (Fig. 4):

\[ p_0(n) = 1 - p_1(n) \quad (6) \]

For both \( p_0(n) \) and \( p_1(n) \), \( T \in [0, 255] \) represents a parameter that controls the degree of fuzziness. For each local neighborhood, \( n=0,1,...,7 \), we can create more than one CND codes. Everything \( z(n) \) is fuzzy, it becomes 0 and 1 with different membership value \( p_0 \) and \( p_1 \).

\[ FCND(x, y, i) = \sum_{n=0}^{N-1} b_n(i) p_1(n) + (1 - b_n(i)) p_0(n) \quad (7) \]

\[ H_{FCND}(i) = \sum_{x, y} FCND(x, y, i), i = 0, 1, ..., 2^F - 1 \]

Fig. 5 A simple example of FCND computation schema on a 3x3 neighborhood for \( T=0.11 \). (a) 3x3 neighborhood. (b) Differences between surrounding pixels and neighborhood pixel with the center. (c) The value of each pixel. (d) FCND codes from the subtraction of \( v(n) - d(n) \).

Fig. 5 is the example of Fuzzy in Combined Neighborhood Differences where local neighborhood is \( T=0.11 \). Fig. 5(a) is a pixels 3x3 neighborhood compute from the whole image. \( V(n) \) is the difference value of two image pixels between neighbors. For Fig. 5(b) is the difference value between a center pixel value with the neighbor image. Fig. 5(c) is the value of each pixel where it will be the reference value to find \( v(n) \) and \( d(n) \). The result of \( v(n) \) is 0.11, 0.56, -0.72, 0.20, 0.0, -0.6, 0.1, -0.3 and \( d(n) \) is -0.50, -0.39, 0.17, 0.1, 0.3, 0.3, -0.3, -0.2. The result from subtract of \( v(n) \) and \( d(n) \) is 0.61, 0.95, -0.24, 0.1, -0.3, -0.9, 0.4, -0.1 where the fuzzy range is -1.1<\text{fuzzy}<1.1. The value is near to 1 we declare as 1 because the probability the value of 1 is higher and otherwise to the value near to 0, we declare as 0. There are two possible values 0.1 and -0.1 to become 1 or 0, we estimate the value of F1 is 0.555 or 0.445 and for F2 the possibility is 0.445 and 0.555. The summation of the possibility is equal to 1.

Table I is the probability of F1 and F2 to become 0 or 1. From that table the highest probability is the 11010010 in binary and 210 in decimal value. The starting point of \( d(n) \) and pixel gap of \( v(n) \) play an important role in generating informative texture feature of CND. It is because, the wrong location of starting point and pixel gap will result in the same information retrieved by LBP or CND. The reason we chose \( x(0) \) for the starting point of \( d(n) \) and 1 as the pixel gap of \( v(n) \) is because they generated highest histogram entropy based on the evaluation on Lenna image. The entropy of a histogram can be calculated using the following formula:

\[ Entropy = \sum_{i=0}^{2^5} p_i \log(p_i) \quad (8) \]

where \( D \) is the histogram dimension and \( p_i \) is the probability for each bin of the histogram.

Table II is the result from probability

<table>
<thead>
<tr>
<th>Method</th>
<th>CND</th>
<th>LBP</th>
<th>FCND</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entropy</td>
<td>5.35</td>
<td>4.76</td>
<td>5.44</td>
</tr>
</tbody>
</table>

This experiment is to compare the histogram entropy of CND, LBP and FCND using (8) on the Lenna images as shown in Fig. 6. The entropy of those histograms is calculated and shown in Table 2. As you can see FCND produces the highest entropy compared to those of LBP and CND. The high entropy of CND is caused by the small number of zero bin in its histogram.
IV. EXPERIMENTAL STUDIES

A. Experiment Setup

The classification accuracy of the CND is measured in terms of various conditions, such as normal and gray value shifted cases and the amount of information containing in CND descriptor is compared. Then we implement CND in the CAD system for classification of Emphysema region. All the experiments execute on a computer with an Intel CoreTM 2 Duo 2.33 GHz and 2GB of main memory. The program is coded with VC++ 6.0 and Window XP operating system. Two other local-based texture descriptors, LBP, LND, CND and the famous Gabor filter have been compared with the proposed FCND method.

Two texture datasets are implemented in the experiment that is Brodatz and Outex_TC_0001. There are 32 classes of Brodatz texture [11] and 24 classes of Outex_TC_0001 are implemented in this experiment as shown in Fig. 7 and Fig. 8 [4][13]. The texture images were corrected by mean and standard deviation in order to minimize discrimination by overall gray-level variation, which is unrelated to local image texture. The correction was applied to the whole 256 x 256 images instead of correcting every sample window separately. The mean gray value and the standard deviation of each corrected image were set to 256 and 40, respectively.

B. Experiment Result

Before conducting the classification, we need to define the threshold value of the FCND. To find a suitable FCND value, we conduct a simple classification test using the Brodatz database. The classification of Brodatz database is conducted using a different fuzziness value of FCND. The result is shown in Fig. 9. As you can see, the highest accuracy is achieved when the fuzziness value is equal to T=0.0625.

The average classification accuracy of each method on Brodatz and Outex_TC_0001 database are calculated. From the results in Fig.10 we can see that FCND performs the best with average accuracy of 95.1% from both texture databases. This is because the texture information described by FCND is more discriminative compared to those of LBP, LND, CND and Gabor filter. For Outex database, the textures are properly captured and the textures are constructed in detail. As a result the classification of Outex is higher than those of Brodatz.

The next experiment is to test the texture classification with noisy textures. In this experiment the testing group is added with different level of Gaussian noise. This is done by changing the value of $\sigma$ in the Gaussian probability density function as follows:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$  \hspace{1cm} (11)

where the value of $\mu$ is set to 0. Different level of Gaussian is applied and the peak signal to noise ratio (PSNR) is calculated between the original and the noisy images.

The noisy texture classification experiment is only conducted on Brodatz textures due to the performances between two textures.

From the result the in the Fig. 13, it can be observed that the accuracies of FCND and Gabor are consistent in difference PSNR. However, the discriminating power of the Gabor is low and this causes a low accuracy compared to the FCND. CND, LND, and LBP are all sensitive to the Gaussian noise since their accuracies started to decline when the PSNR becomes lower than 50 dB. From the curves, it can be summarized that FCND is not just produces high accuracy, but also robust to Gaussian noise.
V. CONCLUSION AND FUTURE WORK

In this paper, a robust texture descriptor is proposed using the fuzzy logic and the combined neighborhood different that is called as FCND. The result shows that FCND achieves high accuracy in texture classification even with noisy images. For future work we would like to increase the robustness of FCND with respect to rotations and scaling.

ACKNOWLEDGMENT

This paper is supported by Universiti Teknikal Malaysia Melaka under PJP/2011/FKEKK(29B)/S00928.

REFERENCES