Neurocomputing 207 (2016) 354–364

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Local Quantization Code histogram for texture classification

Yang Zhao [a,b], Rong-Gang Wang [b,*], Wen-Min Wang [b], Wen Gao [b]

[a] School of Computer and Information, Hefei University of Technology, Hefei 230009, China
[b] School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen 518055, China

ARTICLE INFO

Article history:
Received 16 September 2014
Received in revised form
6 October 2015
Accepted 4 May 2016
Communicated by Qi Li
Available online 16 May 2016

Keywords:
Local binary pattern
Texture classification
Local quantization

ABSTRACT

In this paper, an efficient local operator, namely the Local Quantization Code (LQC), is proposed for texture classification. The conventional local binary pattern can be regarded as a special local quantization method with two levels, 0 and 1. Some variants of the LBP demonstrate that increasing the local quantization level can enhance the local discriminative capability. Hence, we present a simple and unified framework to validate the performance of different local quantization levels. In the proposed LQC, pixels located in different quantization levels are separately counted and the average local gray value difference is adopted to set a series of quantization thresholds. Extensive experiments are carried out on several challenging texture databases. The experimental results demonstrate the LQC with appropriate local quantization level can effectively characterize the local gray-level distribution.

1. Introduction

Texture classification is a basic issue in image processing and computer vision, and playing a significant role in many applications, such as remote sensing, biomedical image analysis, image recognition and retrieval. In these practical applications, it is very difficult to ensure that captured images have the same viewpoint. Hence, texture classification methods should be ideally invariant to translation, rotation and scaling.

More and more attention has been paid on invariant texture classification. So far, many approaches have been proposed to achieve rotation invariance for texture classification that can be broadly divided into two categories, i.e., statistical methods and model-based methods, respectively. In statistical methods, texture is generally described by the statistics of selected features, e.g., invariant histogram, texture elements, and microstructures. Davis et al. [1] exploited polarograms and generalized co-occurrence matrices to obtain rotation invariant statistical features. Duvernoy et al. [2] introduced Fourier descriptors to extract the rotation invariant texture feature on the spectrum domain. Goyal et al. [3] proposed a method by using texel property histogram. Eichmann et al. [4] presented texture descriptors based on line structures extracted by Hough transform. In [33], Hanbay et al. presented four effective rotation invariant features based on histograms of oriented gradients (HOG) and co-occurrence HOG (CoHOG). In model-based methods, texture is usually presented as a probability model or as a linear combination of a set of basis functions. Kashyap et al. [5] developed a circular simultaneous autoregressive (CSAR) model for rotation invariant texture classification. Cohen et al. [6] characterized texture as Gaussian Markov random fields and used the maximum likelihood to estimate rotation angles. Chen and Kundu [7] addressed rotation invariant by using multichannel sub-bands decomposition and hidden Markov model (HMM). Porter et al. [8] exploited the wavelet transform for rotation invariant texture classification by means of the Daubechies four-tap wavelet filter coefficients. Recently, Xu et al. [30–32] proposed a scale invariant texture feature by means of the multi-fractal spectrum.

McLean [15] proposed to use vector quantization for texture classification. But the quantization step is processed on large image area and thus loses the details of the local neighborhood grayscale distribution. However, the local distribution has been proven to be the important discriminative information of texture. For example, Haralick proposed that central to virtually all aspects of texture classification is the identification of a “texture cell” that defines a local region containing the essence of the repeated structure [16]; Effective texton-based methods [17,27–29] also proved that texture classification can be tackled effectively by employing only local neighborhood distributions .

Local texture descriptor is another example to prove the importance of the local neighborhood distribution. In [9], Ojala et al. proposed an efficient local operator, namely Local Binary Pattern (LBP). The LBP extracts the local pattern and it is proven to be invariant to monotonic grayscale transformation. Nowadays, the LBP is one of the most popular local texture descriptors, since it is simple and effective [26]. Many LBP-like local texture

* Corresponding author.
E-mail address: rgwang@pkusz.edu.cn (R.-G. Wang).

http://dx.doi.org/10.1016/j.neucom.2016.05.016
0925-2312/

operators have been proposed after Ojala's work, e.g., Heikkila et al. [10] proposed center-symmetric LBP (CS-LBP) by comparing center-symmetric pairs of pixels instead of comparing neighbors with central pixels. Liao et al. [11] presented Dominant LBP (DLBP), in which dominant patterns were experimentally chosen from all rotation invariant patterns. Tan and Triggs [12] proposed the method of Local Ternary Pattern (LTP), which extends original LBP to 3-valued codes. Recently, Guo et al. [13] developed the completed LBP (CLBP) by combining the conventional LBP with the measures of local intensity difference and central gray level. Khellah [14] presented a new method for texture classification, which combines Dominant Neighborhood Structure (DNS) and traditional LBP, Zhao et al. [18] proposed to use local binary count (LBC) to extract the local neighborhood distribution, Zhang et al. [19] presented a new local energy pattern for texture classification, Li et al. [20] proposed a scale invariant LBP by means of scale-adaptive texton. Most Recently, Guo et al. [36] present a scale selective CLBP. In addition, there is other way to look at LBP, e.g., LBP is regarded as a special filter-based method [24,25].

The LBP can be regarded as a special local quantization method with two quantization levels, 0 and 1. How to select the quantization level is a basic issue of the traditional quantization methods. The specified gray value of each individual pixel is sensitive to noise and illumination, thus lower quantization level is more robust to the illumination changes. But reducing the quantization level also loses detailed gray value information of pixel at the same time. The LBP is insensitive to monotonic illumination changes by quantizing the local gray level into only two levels. Meanwhile, the two values (0 and 1) extract scarcely any gray value information of the pixel. Although the LBP discards almost all the gray value information of individual pixel, the quantized neighbor pixels are combined together to describe the local pattern. Therefore, the LBP can effectively characterize the local distribution. Is the local quantization level needed to be increased? Does an optimal local quantization level exist? How to describe the local distribution when the quantization level is increased? In this paper, we shall try to address these questions by proposing a new local operator named Local Quantization Code (LQC). Experimental results illustrate that the LQC with appropriate local quantization level can effectively characterize the local neighborhood distribution.

The rest of this paper is organized as follows: Section 2 briefly reviews the basic principle of the relative variants of the LBP. Section 3 presents the LQC in detail. Experimental results are presented in Section 4, and Section 5 concludes the paper.

2. Related works and analyses

In this section we provided a brief review of the LBP and related variants of the LBP, i.e., the LTP, the CLBP and the LBC.

As shown in Fig. 1, the algorithm of LBP contains three main steps. First, the values of neighbor pixels are turned into binary values (0 or 1) by comparing them with the central pixel. Second, the binary numbers are encoded to characterize a local structure pattern, and then the code is transformed into decimal number. Finally, after the LBP code of each pixel is defined, a histogram will be built to represent the texture image.

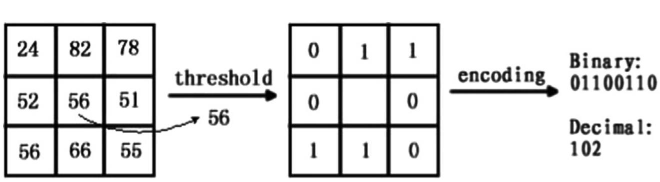

Fig. 1. Illustration of the LBP process. (P=8, R=1).

Usually, the LBP encoding strategy can be described as follows:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p, \quad s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \tag{1}$$

where g_c represents the gray value of the center pixel and g_p (p=0, …,$P-1$) denotes the gray value of the neighbor pixel on a circle of radius R, and P is the total number of the neighbors. As aforementioned, although the LBP is robust to monotonous illumination changes, the binary quantization process also loses the detailed gray value information of pixels. Hence, it seems that increasing the local quantization level can enhance the discriminative capability of the LBP. However, it is very difficult to compute the LBP codes directly when the local quantization level is increased. It is easy to be found that the length of LBP feature become L^P if the quantization level increases to L, e.g., if local quantization level is 4 and 16 neighbors are calculated, the length of LBP-like feature will be 4^{16}.

Although it is hard to increase the local quantization level directly in the LBP-like way, many works have been proposed to extract the gray value information that omitted in the binary quantization step of the LBP.

Tan and Triggs [12] proposed local ternary pattern (LTP) to quantize the local neighbors into three levels. As illustrated in Fig. 2, 2-valued (0, 1) LBP code is extended to 3-valued (-1, 0, 1) ternary code by means of a threshold t. The upper pattern and lower pattern are then encoded in LBP-like way, respectively. LTP codes can extract more gray value difference information, but no longer strictly invariant to monotonic gray scale transformation since threshold t is specified by user. It also should be noticed that the threshold t is set as 5 on many texture databases according to the experimental performance. Since the gray value of the pixels can be 0–255, the local quantization threshold seems quite small, and this will also be discussed later in Section 3.2.

Guo et al. [13] proposed a completed framework of LBP (CLBP) by combining the sign (0 or 1) feature with the magnitude (the gray value difference) feature. Although the CLBP did not directly increase the local quantization level, the magnitudes feature provided complementary gray value difference information that lost during the binary quantization process. Moreover, Guo et al. observed that the center pixel also had discriminative information. The CLBP extended original LBP to a completed framework and achieved impressive classification results.

These variants of the LBP demonstrate that increasing the local quantization levels can enhance the discriminative capability. Then the key question becomes how to increase the local quantization level in an efficient and unified framework.

In [18], Zhao et al. proposed the local binary count (LBC) by means of a local counting method to encode the rotation invariant local distribution after local neighbors are quantized into two levels. In the LBC, the number of value 1's in the binary neighbor sets is simply counted. As illustrated in Fig. 3, the number of value 1's is 4 in the binary neighbor set, thus the LBC code of the central pixel is 4. The LBC reveals another cursory encoding method to characterize the local neighborhood distribution, and the LBC-like encoding is easy to expand.

3. Local Quantization Code histogram method

3.1. Calculation of the Local Quantization Code

In the conventional LBP and its variants, each pixel in the local neighbor set is turned into binary form by comparing it with the central pixel. To increase the quantization level, a series of quantization thresholds (σ_1, σ_2, σ_3, σ_4 …) need to be used. After these