Multiscale Adaptive Threshold Local Ternary Pattern Encoding Algorithm Postprint

Authors: Zhang Yu, Wang Qiang, Li Bolin, to scale heights

Date: 2018-12-13T00:00:00+00:00

Abstract

To address the issues of limited descriptive information and noise sensitivity in Local Binary Pattern (LBP), a Multi-scale Adaptive Local Ternary Pattern (MSALTP) encoding algorithm is proposed. MSALTP first enlarges the original image; secondly, partitions the image uniformly into several regions and calculates the pixel mean; then computes the deviation between each region' s center pixel and the mean; finally extracts ALTP features and achieves image classification through statistical feature histograms. Experimental results demonstrate that the proposed algorithm exhibits substantially improved recognition rates compared to current state-of-the-art noise-resistant algorithms under various noise conditions.

Full Text

Preamble

Coding Algorithm of Multi-Scaled Adaptive Threshold Local Ternary Pattern

Zhang Yu[1], Wang Qiang[1,2], Li Bailin[1], Gao Pan[1]
[1]School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China
[2]School of Mechanical Engineering, Chengdu Institute of Technology, Chengdu 611730, China

Abstract: To address the limitations of local binary pattern (LBP) in describing single information and its sensitivity to noise, this paper proposes a multi-scale adaptive local ternary pattern (MSALTP) encoding algorithm. MSALTP first enlarges the original image; second, divides the image into several regions equally and calculates the mean value of the pixels; then computes the deviation between the center pixel and the mean value of each region; finally extracts ALTP features and uses the resulting statistical feature histograms for image classification. Experiments demonstrate that the proposed algorithm achieves significantly higher recognition rates compared to current state-of-the-art anti-noise algorithms under various noise conditions.

Keywords: local ternary pattern; adaptive threshold; multi-scale

0 Introduction

Texture represents one of the most important features in images, with different visual images containing distinct textural compositions. Texture analysis and classification serve as fundamental problems in image processing, computer vision, and pattern recognition, possessing significant theoretical value and practical importance [1]. Common texture feature extraction methods include gray-level co-occurrence matrices [2], Markov random fields [3], wavelet transforms [4], and fractal theory [5]. However, these methods suffer from high computational complexity and suboptimal classification performance for images with non-uniform illumination variations, background interference, and noise effects.

The local binary pattern (LBP) texture descriptor has been successfully applied to texture classification [6]. Compared with other algorithms, the LBP operator can simply and effectively extract textural features from images, achieving satisfactory image analysis results. Originating in the field of texture analysis, the LBP method was first proposed by the University of Oulu in Finland [7]. Due to its simple principle and low computational complexity, it has been widely applied to image segmentation, object tracking, face recognition, medical image analysis, and other domains [8].

The original LBP algorithm exhibits poor robustness to image rotation, scale variations, and non-uniform illumination changes. To enhance the anti-noise capability and robustness of LBP, researchers have proposed numerous improved algorithms, such as rotation-invariant uniform patterns and noise-resistant local binary pattern (NRLBP) [9,10,11]. Among these, Tan’ s LTP algorithm [12] introduces a quantization threshold for image encoding, effectively improving robustness to illumination changes and noise effects. However, since the original LTP threshold is a user-defined fixed value requiring extensive experimental selection, it cannot adapt to all samples.

Images at different scales describe different features, and employing multi-scale analysis [7] enables more specific feature representation with better robustness. In this work, the original image dimensions are enlarged by 0.5× and 2× using linear interpolation, and textural features are extracted from all three images. This paper proposes a multi-scale adaptive threshold local ternary pattern encoding algorithm that optimizes the fixed threshold in the original LTP algorithm to an adaptive threshold generated based on each image’ s textural characteristics, enabling the encoding algorithm to adapt to various image types. Additionally, it extends feature extraction from a single image to multiple scales of the image, resulting in more specific feature representation. The local features thus gain both robustness and adaptability, better describing image characteristics and improving recognition accuracy and anti-noise capability.

1.1 Local Binary Pattern

LBP encodes local textural structure information by comparing the gray value of any pixel with its surrounding neighborhood pixels. If a neighborhood pixel' s gray value is not greater than the center pixel' s gray value, the position is marked as 1; otherwise, it is set to 0. Following a unified encoding direction, different weights are assigned to different neighborhood points, and the binary sequence is converted into an unsigned decimal number used as the pixel' s LBP feature value.

The original LBP operator is defined in a 3$×$3 window, using the window' s center pixel as the threshold. Its calculation process is illustrated in [Figure 1: see original paper]. In Figure 1, starting from the neighborhood pixel to the right of the center pixel and rotating counterclockwise yields a binary sequence, from which the corresponding decimal number is obtained as the pixel' s local binary pattern.

The LBP feature calculation formula [1] is as follows:

$$LBP_{p,r} = \sum_{i=0}^{p-1} s(g_i - g_c) \times 2^i, \quad s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

where r represents the neighborhood radius, p represents the number of uniformly distributed pixels on the circular neighborhood with radius r, g_c denotes the gray value of the local neighborhood center pixel, and g_i denotes the gray values of neighborhood pixels.

1.2 Rotation Invariant Uniform Pattern

The square neighborhood system shown in Figure 1 cannot satisfy requirements for different sizes and frequencies. To more accurately reflect texture features, a circular neighborhood structure replaces the square structure. [Figure 2: see original paper] shows circular local binary patterns with different numbers of neighbors and radii [11].

Ojala et al. [5] proposed a rotation-invariant LBP operator that obtains a series of LBP codes by continuously transforming the starting position of the circular neighborhood' s encoding sequence and uses the minimum value as the neighborhood' s LBP value. Its mathematical description is given by Equation (3):

$$LBP_{p,r}^{ri} = \min\{ROR(LBP_{p,r}, i) \mid i = 0, 1, \dots, p-1\}$$

where $ROR(x, i)$ represents cyclically right-shifting the binary number corresponding to x by i bits. The introduction of rotation-invariant LBP significantly improved recognition performance for rotated texture images.

To further enhance LBP feature performance and reduce feature dimensionality, Ojala et al. [6] divided rotation-invariant patterns into rotation-invariant uniform patterns (riu2) and rotation-invariant non-uniform patterns based on the number of 0/1 (or 1/0) transitions in the binary encoding sequence. The rotation-invariant uniform pattern of LBP features ($LBP_{p,r}^{riu2}$) is given by Equations (4) and (5):

$$LBP_{p,r}^{riu2} = \begin{cases} \sum_{i=0}^{p-1} s(g_i - g_c), & \text{if } U(LBP_{p,r}) \leq 2 \\ p+1, & \text{otherwise} \end{cases}$$

$$U(LBP_{p,r}) = |s(g_{p-1} - g_c) - s(g_0 - g_c)| + \sum_{i=1}^{p-1} |s(g_i - g_c) - s(g_{i-1} - g_c)|$$

where U represents the number of 0/1 (or 1/0) transitions in the binary sequence. In this paper, the feature dimension is 10 dimensions with neighborhood points $p = 8$ and radius $r = 2$. This approach greatly reduces the number of binary patterns, effectively decreasing feature dimensionality without losing any effective information, and thus is more widely used.

1.3 Local Ternary Pattern

LTP extends LBP by defining an interval of length $2t$ [13]. If the difference between a neighborhood pixel' s gray value and the center pixel' s gray value falls on the right side of this interval, the input value is encoded as 1; if on the left side, it is encoded as -1; otherwise, it is 0. The calculation formula is as follows:

$$LTP_{p,r,t} = \sum_{i=0}^{p-1} s(g_i, g_c, t) \times 3^i, \quad s(g_i, g_c, t) = \begin{cases} 1, & g_i - g_c \geq t \\ 0, & |g_i - g_c| < t \\ -1, & g_i - g_c \leq -t \end{cases}$$

where r represents the neighborhood radius, p represents the number of uniformly distributed pixels on the circular neighborhood with radius r, g_c denotes the center pixel' s gray value, g_i denotes neighborhood pixels' gray values, and t is the threshold for the local ternary pattern.

To simplify experimental calculations, the LTP encoding process is decomposed into positive and negative computation parts [12]. The upper pattern is defined as the encoding obtained by marking all values except 1 as 0 in the original ternary encoding. The lower pattern is defined as the encoding obtained by marking all values except -1 as 0 and replacing -1 with 1. The upper and lower pattern encodings of LTP are shown in [Figure 3: see original paper], where the threshold t is set to 5.

2.1 Multi-Scale Analysis

Multi-scale image technology, also known as multi-resolution technology, involves expressing images at multiple scales and processing them separately at each scale. Features exhibited by images at different scales vary. The rationale is that characteristics not easily visible or obtainable at one scale may be readily discovered or acquired at another scale, making multi-scale technology more commonly used for feature extraction.

When extracting image features, using fixed-scale feature detection yields results biased toward that scale while missing many features at other scales. To ensure that identical features in images are detected across different scales, detection and matching must be performed at multiple scales. Different scales reveal different image characteristics.

Before extracting ALTP features, the original image I is scaled by $0.5\times$ and $2\times$ using bilinear interpolation to obtain images $I_{1/2}$ and I_2. If noise is abundant, the scaled images will correspondingly contain more noise; if noise is scarce, the noise will be reduced accordingly. After scaling, a resampling is performed, and auxiliary calculations on the recomputed data ensure final result accuracy.

Feature extraction is performed at three scales for each image to enhance feature description capability. Bilinear interpolation involves large computational cost but produces high-quality scaled images without pixel value discontinuities. Gradient magnitude matrices are computed for all three images, and these gradient magnitudes are used to weight the histogram statistics. ALTP features are extracted from images at different scales, and the results are combined for histogram statistics used in classification. Consequently, the MSALTP method can extract more image features and achieve better robustness and accuracy.

2.2 Selecting Adaptive Threshold

Since the threshold t in traditional LTP operators is fixed and unsuitable for all images, extensive experiments are required to select the threshold for each experiment, increasing experimental difficulty and lacking adaptability. The adaptive threshold method for LTP proposed in this paper selects threshold

t adaptively based on the image' s own characteristics, demonstrating better robustness to illumination changes and noise while enhancing the classification performance of local texture features.

The method first divides the image into several regions to obtain the pixel mean value μ_b for each region. It then calculates the deviation σ_b between each region' s center pixel and the regional mean. Local deviation reflects the dispersion degree of each pixel point relative to the center pixel dataset and reduces image noise interference [15]. Therefore, σ_b is used as the adaptive threshold t, with specific calculations as follows:

First, set the image width as W and height as H. Divide the image into non-overlapping regions of size $W_b \times W_b$ blocks, compute local variance, and generate a local variance histogram. The mean and deviation formulas for each region are given by Equations (8)-(10):

$$\mu_b(x,y) = \frac{1}{N} \sum_{i=-\frac{W_b}{2}}^{\frac{W_b}{2}} \sum_{j=-\frac{W_b}{2}}^{\frac{W_b}{2}} I(x+i, y+j)$$

$$\sigma_b(x,y) = \sqrt{\frac{1}{N} \sum_{i=-\frac{W_b}{2}}^{\frac{W_b}{2}} \sum_{j=-\frac{W_b}{2}}^{\frac{W_b}{2}} \left[I(x+i, y+j) - \mu_b(x,y)\right]^2}$$

where $N = W_b \times W_b$ represents the number of pixels in the divided region, $I(x,y)$ denotes the pixel value at point (x,y), $\mu_b(x,y)$ represents the pixel mean of the region where point (x,y) is located, and $\sigma_b(x,y)$ represents the pixel deviation of the region where point (x,y) is located.

The specific process is as follows:

a) Scale the original image I by 0.5× and 2× using bilinear interpolation to obtain images $I_{1/2}$ and I_2.

b) Divide each image into several regions equally, compute the pixel mean μ_b for each region using Equations (8)-(10), then extract the deviation σ_b between each region' s center pixel and regional mean.

c) Use the extracted variance σ_b as the adaptive threshold for local ternary pattern encoding, compute each image' s code values using Equation (11), extract ALTP features from the images, and generate feature histograms.

d) Iterate through all images in the database, feed the resulting feature histograms into a support vector machine for template training and texture classification recognition.

In Equation (11), if the difference between a neighborhood pixel' s gray value and the regional pixel mean is not less than σ_b, the input value is encoded as 1;

if not greater than $-\sigma_b$, the input is encoded as -1; if the difference lies between $-\sigma_b$ and σ_b, the input is encoded as 0.

The upper and lower pattern images extracted using ALTP features are shown in [Figure 4: see original paper], while features extracted using the LBP operator are shown in [Figure 5: see original paper]. Figures 4 and 5 demonstrate that the LTP algorithm can describe bright points, dark points, and edge information in detail. As illustrated, the LTP algorithm extracts clearer textures than LBP, providing better recognition capability; consequently, features extracted by MSALTP perform better.

Since noise intensity varies across different regions of an image, ALTP selects different thresholds based on each region' s noise intensity, achieving adaptive threshold local ternary pattern encoding. This algorithm not only effectively solves the noise sensitivity problem but also eliminates the need for extensive experiments to determine thresholds required by the original LTP algorithm. The ALTP method replaces the unified fixed threshold t with an adaptive threshold σ_b, generating different thresholds based on noise levels in each region of every image, thereby demonstrating better anti-noise capability than the LBP operator. Additionally, feature extraction from images does not require extensive experimental validation, making the ALTP operator more convenient to use.

3.1 Experimental Parameter Settings and Data Selection

This study employs the KTH-TIPS, Outex, and UIUC standard databases, along with a railway fastener database obtained from actual photography for an automatic railway facility inspection project. Experiments on the railway fastener database verify the reliability of the MSALTP operator in practical operations. The number of images, categories, and training samples for each texture library used in the experiments are shown in .

In the experiments, LBP, LTP with fixed threshold $t = 5$, improved adaptive threshold LTP, and NRLBP with interval value 2 are used for image processing and comparison. For accurate experimental data, all four modes employ rotation-invariant uniform patterns with radius $r = 2$ and neighborhood count $p = 8$. Gaussian noise with $\sigma = 0.05, 0.10, 0.15$ and salt-and-pepper noise with $p = 0.2, 0.4, 0.6$ are added to images from the KTH-TIPS, Outex, and UIUC standard databases. Original images and images with different salt-and-pepper noise values from each database are shown in Figures 5–8. The railway fastener database is obtained from actual photography and thus requires no additional noise; classification and recognition are performed based on actual image conditions. To avoid random deviation, each experiment is conducted 10 times on every texture library, and the average value is calculated as the final result. Experiments are based on Matlab2014 and Vlfeat0.9.21 open-source function libraries, using χ^2 kernel support vector machines [16] to evaluate classification performance of each algorithm.

3.2 Experimental Results Analysis and Discussion

The KTH-TIPS database contains complex illumination and rotation variations, with different lighting conditions, shooting angles, and scales for each image category, primarily used to test robustness to complex environmental changes. shows recognition rates using LBP, LTP, NRLBP, and MSALTP algorithms on the KTH-TIPS database without noise and with Gaussian noise ($\sigma = 0.05, 0.10, 0.15$) and salt-and-pepper noise ($p = 0.2, 0.4, 0.6$).

As shown in , the MSALTP algorithm demonstrates significantly higher recognition rates than the original LTP algorithm under different noise conditions, indicating that the proposed method has better robustness to illumination changes.

[Figure 11: see original paper] compares recognition rates of LTP, NRLBP, and ALTP features extracted from KTH-TIPS database images with $p = 0.2$ salt-and-pepper noise while varying the threshold t for LTP and NRLBP operators.

As shown in [Figure 11: see original paper], different threshold selections for LTP and NRLBP operators yield different recognition rates. Therefore, the optimal threshold t for recognition cannot be determined before extensive experiments, and if images change, the threshold t must be reselected. The ALTP operator automatically determines threshold t based on image textural features, offering convenient usage, and as shown in [Figure 11: see original paper], achieves higher recognition rates than LTP and NRLBP operators with different thresholds.

Both TC12 and UIUC databases exhibit obvious rotation variations, primarily used to test robustness to rotation changes. The TC12 database includes three illumination conditions. and present recognition rates using LBP, LTP, NRLBP, and MSALTP algorithms on TC12 and UIUC databases without noise and with Gaussian noise ($\sigma = 0.05, 0.10, 0.15$) and salt-and-pepper noise ($p = 0.2, 0.4, 0.6$).

As shown in and , the MSALTP algorithm achieves significantly higher recognition rates than the original LTP algorithm under different noise conditions, demonstrating that the improved MSALTP algorithm has better robustness to rotation variations.

The railway fastener database is constructed for an actual railway fastener automatic safety inspection project, performing recognition based on real images. shows recognition rates using LBP, LTP, NRLBP, and MSALTP algorithms on the railway fastener database without any added noise.

As shown in , the proposed algorithm maintains higher recognition rates in practical project operations.

Based on experimental data from -and [Figure 11: see original paper], the improved LTP algorithm achieves higher recognition rates than other LBP algorithms on KTH-TIPS, Outex, and UIUC texture libraries, indicating strong feature discrimination capability. On Outex, UIUC, and KTH-TIPS databases with diverse illumination and rotation variations, the improved MSALTP algorithm shows significantly higher recognition rates than other algorithms, demonstrating superior robustness to illumination and rotation changes.

Analysis of all experimental results confirms that the ALTP operator provides better recognition capability under different noise conditions across various databases compared to existing algorithms, validating the effectiveness of the encoding algorithm.

---

4 Conclusion

The LBP algorithm suffers from single information description and noise sensitivity issues. The original LTP algorithm strengthens the connection between center and neighborhood pixels, further reducing noise impact on texture features and improving algorithm robustness. However, threshold t selection in the original LTP requires extensive experiments, with different texture libraries requiring different thresholds, increasing experimental difficulty. Building upon LTP, this paper proposes a multi-scale adaptive threshold local ternary pattern encoding algorithm that determines thresholds based on image textural characteristics. This method inherits the advantages of the original LTP algorithm while improving its adaptability, with experiments demonstrating its effectiveness.

References

[1] Liu Li, Xie Yuxiang, Wei Yingmei, et al. Survey of local binary pattern method [J]. Journal of Image and Graphics, 2014, 19(12): 1696-1720.

[2] Vargas J F, Ferrer M A, Travieso C M, et al. Off-Line signature verification based on grey level information using texture features [J]. Pattern Recognition, 2011, 44(2): 375-385.

[3] Huang A, Abugharbieh R, Tam R. A novel rotationally invariant region-based hidden Markov model for efficient 3-D image segmentation [J]. IEEE Trans on Image Processing, 2010, 19(10): 2512-2522.

[4] Choy S K, Tong C S. Statistical wavelet subband characterization based on generalized gamma density and its application in texture retrieval [J]. IEEE Trans on Image Processing, 2010, 19(2): 281-289.

[5] Galloway M. Texture analysis using gray level run lengths [J]. Computer Graphics and Image Processing, 1975, 4(2): 172-179.

[6] Ojala T, Pietikäinen M, Howard D. A comparative study of texture measures with classification based on feature distributions [J]. Pattern Recognition, 1996,

29(1): 51-59.

[7] Huang Chen, Fei Jiyou, Liu Xiaodong. Texture feature classification method based on Gaussian local binary pattern [J]. Application of Electronic Technique, 2018, 44(1): 121-124.

[8] Ojala T, Pietikäinen M. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(7): 971-987.

[9] Song Kechen, Yan Yunhui, Chen Wenhui, et al. Research and perspective on local binary pattern [J]. Acta Automatica Sinica, 2013, 39(6): 730-744.

[10] Ji Zhong, Nie Linhong. Texture image classification based on anti-noise local binary pattern [J]. Journal of Computer Research and Development, 2016, 53(5): 1128-1135.

[11] Ren Jianfen, Jiang Xudong, Yuan Junsong. Noise resistant local binary pattern with an embedded error-correction mechanism [J]. IEEE Trans on Image Processing, 2013, 22(10): 4049-4060.

[12] Tan Xiaoyang, Bill T. Enhanced local texture feature sets for face recognition under difficult lighting conditions [J]. IEEE Trans on Image Processing, 2010, 19(6): 1635-1650.

[13] Zheng Qiumei, Xie Huanli, Wang Fenghua, et al. A deep learning face recognition algorithm based on local three-valued mode [J]. Computer and Modernization, 2018(2): 112-117.

[14] Zhao Lingzhi, Li Weisheng. An image matching method based on LTP feature [J]. Application Research of Computers, 2009, 26(10): 3983-3985.

[15] Chen Huijuan, Dai Shengkui. Improved image noise estimation based on block method [J]. Journal of Computer Applications, 2014, 34(7): 2014-2017.

[16] Akata Z, Perronnin F, Harchaoui Z, et al. Good practice in large-scale learning for image classification [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2014, 36(3): 507-520.

Note: Figure translations are in progress. See original paper for figures.