Tibetan Recognition System Based on the Improved Extraction Algorithm of Complexity Index Feature

Zhiyuan Mai, Wei Xiang*
College of Electrical & Information Engineering
Southwest University for Nationalities
Chengdu, Sichuan, China
*Corresponding author.Email:wei.xiang@foxmail.com.

***Abstract*—Tibetan character recognition is a significant module of multi-language information processing system in China. Owing to the special structure of Tibetan characters, the recognition of traditional Tibetan characters encounters the problems of low recognition rates and poor recognition effects. Through an in-depth study on features of Tibetan characters, this paper compared and analyzed the character feature extraction algorithm are used widely, and on this basis, a Tibetan character recognition system based on BP neural network was designed. The results of the experiments indicate that the improved extraction algorithm of complexity index feature to deal with Tibetan characters has the higher recognition rate and recognition speed.**

Keywords-complexity index feature; feature extraction; Tibetan character recognition

I. INTRODUCTION

Tibetan traditional culture which is one of the most wonderful work among cultural treasure of the human world[1], and one out of a multitude outstanding achievements in this splendid Tibetan culture is the Tibetan language, which is the precious heritage in the history of world culture[2]. As other language recognition, the Tibetan which is printed or written on paper can be distinguished and texted automatically by computers , people's mental and physical labor intensity can be greatly reduced through this kind of ideal means of high-speed text input, we can describe visually, Tibetan recognition which is a technique to let the computer 'know' the Tibetan language[3]. It will greatly promote to the Tibetan compatriots in economy, culture and education, of common prosperity, stability and development.

II. THE TRADITIONAL EXTRACTION ALGORITHM OF COMPLEXITY INDEX FEATURE

Complexity index[4] is a kind statistical feature of character, the basic theory is according to the feature of various complexity index to distinguish characters. The feature is not sensitivity to the size and the location of the characters, but it owns quite good ability of classification[5]. Generally, to realize the complexity index right now, is to computing characters of transverse and longitudinal centroid of the square root of the second moment at the first place, secondly is to find out the line length where the characters in x and y direction, the final step is to calculate the ratio to get the complexity index in x and y direction[6]. The simple method of how to computing complexity index can reflect the statistical characteristics of Tibetan characters, in the application it owns great effect on recognition. However, we can observe lots of problems to improve the algorithm through careful analysis, such as: (1) the influence of complexity index coefficient is too simple[7]; (2) the simple statistics of x and y direction can't completely reflect the complexity index of image[8].

III. THE IMPROVED EXTRACTION ALGORITHM OF COMPLEXITY INDEX FEATURE

If the binary image after being processed is $N \times N$ array, with $P(x, y)$ represents the pixel values which is a point on abscissa for x and on the ordinate for y. While the expression of existing pixel values is as follow:

$$P(x, y) = \begin{cases} 1 \\ 0 \end{cases} (x, y = 0,1,2...N-1) \tag{1}$$

With δ_x and δ_y represent the length of the characters of horizontal and vertical line segments, with C_x and C_y represent the complexity index of the characters on x and y directions, with L_x and L_y represent the length of the characters in the x and y directions. While there are defined expressions as follows:

$$\delta_x = 2\sqrt{\frac{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}(x-G_x)^2 P(x,y)}{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}P(x,y)}} \tag{2}$$

$$\delta_y = 2\sqrt{\frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(j-G_y)^2 P(i,j)}{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}P(i,j)}} \tag{3}$$

Then C_x and C_y can be expressed as:

$$C_x = \frac{L_x}{\delta_x} \tag{4}$$

$$C_y = \frac{L_y}{\delta_y} \tag{5}$$

While G_x and G_y is the coordinate value of the image centroid of the character. The formula is as follows:

$$G_x = \frac{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} x \cdot P(x,y)}{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} P(x,y)} \tag{6}$$

$$G_y = \frac{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} y \cdot P(x,y)}{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} P(x,y)} \tag{7}$$

There are sixteen kinds of combination of grid in total, after separating all the images of character into 2×2 grids. While L_x and L_y are the calculated value which is based on the 16 types of grid in the images affect the x direction line and y direction line. Then choose different impact factors generally, and to count the 16 types of complex values L_x and L_y for x and y direction at last:

$$L_x = \sum_{i=0}^{15} L_{xi} \cdot S_i \tag{8}$$

$$L_y = \sum_{i=0}^{15} L_{yi} \cdot S_i \tag{9}$$

Finally, according to the formula (4) and (5) to calculate the complexity index C_x and C_y. Statistical amount of the type I grid in the whole image is S_i.

To solve the mentioned shortcomings, we will make the following improvements of the traditional extraction algorithm of complexity index feature in this paper.

Change the way to describe the statistical value of complexity index. We only use the value of influence before, but now we combine the weight coding and the impact factor instead, that is to say, from the top to the bottom, from the left to the right, each square takes up the value of 2^0 to 2^3, therefore, from the left to the right, from the top to the bottom, the value of mesh from each gridding just expressed by Fig.1.

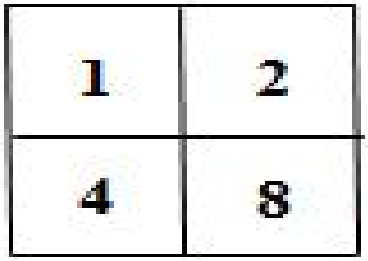

Figure 1 . The weight distribution of meshes

Then the central value represents the gridding from the first to the sixteenth types is zero to fifteen. We can increase the contrast where pixel of characters goes in x direction and y direction of the distribution of gridding, because the pixel of characters takes up different places in the gridding. At the same time, because of the shapes are different from other characters, the shapes all towards the x direction from the top to the bottom, an increasing amount of characters turns black into white, the other growing quality of characters turns white into black. At this moment, we could not affect the feature of Tibetan characters exactly by using simple complexity index[9], merely, we could calculate different statistical value with the method of measure the trend of binary coding, because of the different location of the pixel in x or y direction. Not only improves the ability to sort, but also reflects differences between the two types well.

To an image of character, the statistical feature is reflected well by the distribution feature of the pixel of characters. Different characters have different shapes and trend, based on this to increase the quality of complexity index, according to the color of the pixel to add the complexity index of white pixels C_w and black pixels C_b, in line with the trend of characters distribution to add the complexity index C_l and C_r which is lean to the left and right.

The impact factors of feature L_x, L_y, L_w, L_b, L_l, L_r are represented by sixteen types of gridding $L_{y0} \sim L_{y15}$, $L_{w0} \sim L_{w15}$, $L_{b0} \sim L_{b15}$, $L_{l0} \sim L_{l15}$ $L_{r0} \sim L_{r15}$, like the Table to choose the value:

Through the formula (8) and (9) to calculate L_x and L_y, also we can calculate L_w, L_b, L_l and L_r in a similar way. By making use of formula (4), (5) and the value we just calculated to get complexity index C_x, C_y; C_w, C_b; C_l, C_r. After counting we could get the feature of six-dimensional complexity index. To separate the characters in similar shape easily, the characters feature is reflected by this complexity index in a better way. Therefore, we could get six-dimensional component after counting the feature of complexity index.

TABLE I THE IMPROVED IMPACT FACTORS OF COMPLEXITY INDEX

$L_{x0}=0$	$L_{y0}=0$	$L_{w0}=1$	$L_{b0}=1$	$L_{l0}=0$	$L_{r0}=0$
$L_{x1}=0.5$	$L_{y1}=0.5$	$L_{w1}=0.75$	$L_{b1}=0.25$	$L_{l1}=0.5$	$L_{r1}=0.5$
$L_{x2}=0.5$	$L_{y2}=0.5$	$L_{w2}=0.75$	$L_{b2}=0.25$	$L_{l2}=0.5$	$L_{r2}=0.5$
$L_{x3}=1$	$L_{y3}=0$	$L_{w3}=0.5$	$L_{b3}=0.5$	$L_{l3}=0$	$L_{r3}=0$
$L_{x4}=0.5$	$L_{y4}=0.5$	$L_{w4}=0.75$	$L_{b4}=0.25$	$L_{l4}=0.5$	$L_{r4}=0.5$
$L_{x5}=0$	$L_{y5}=1$	$L_{w5}=0.5$	$L_{b5}=0.5$	$L_{l5}=0$	$L_{r5}=0$
$L_{x6}=0.5$	$L_{y6}=0.5$	$L_{w6}=0.5$	$L_{b6}=0.5$	$L_{l6}=1$	$L_{r6}=0$
$L_{x7}=0$	$L_{y7}=0$	$L_{w7}=0.25$	$L_{b7}=0.75$	$L_{l7}=1$	$L_{r7}=0$
$L_{x8}=0$	$L_{y8}=0$	$L_{w8}=0.75$	$L_{b8}=0.5$	$L_{l8}=0.5$	$L_{r8}=0.5$
$L_{x9}=0$	$L_{y9}=0$	$L_{w9}=0.5$	$L_{b9}=0.5$	$L_{l9}=0$	$L_{r9}=1$
$L_{x10}=0$	$L_{y10}=1$	$L_{w10}=0.5$	$L_{b10}=0.5$	$L_{l10}=0$	$L_{r10}=0$
$L_{x11}=0$	$L_{y11}=0$	$L_{w11}=0.25$	$L_{b11}=0.25$	$L_{l11}=0$	$L_{r11}=1$
$L_{x12}=1$	$L_{y12}=0$	$L_{w12}=0.5$	$L_{b12}=0.5$	$L_{l12}=0$	$L_{r12}=0$
$L_{x13}=0$	$L_{y13}=0$	$L_{w13}=0.25$	$L_{b13}=0.75$	$L_{l13}=0$	$L_{r13}=1$
$L_{x14}=0$	$L_{y14}=0$	$L_{w14}=0.25$	$L_{b14}=0.75$	$L_{l14}=1$	$L_{r14}=0$
$L_{x15}=0$	$L_{y15}=0$	$L_{w15}=0$	$L_{b15}=$	$L_{l15}=0$	$L_{r15}=0$

IV. THE STEPS OF SXPERIMENT AND RESULTS OF ANALYZING

We choose seventy-five different Tibetan characters in serif font to text, each character has three font sizes(number14 ,number20, number26), and corresponds to a model, that is to say seventy-five Tibetan characters in serif font have seventy-five models. In this paper under the environment of Windows 7, we program with MATLAB and do simulation experiments on it.

In this experiment, first of all, use iterative thresholding and Otsu thresholding in the global threshold binarization to deal with Tibetan characters by binarization processing, to change the gray image of Tibetan characters into binarization image, reduce capacity of data storage and complexity of the subsequent processing.

Secondly, the image of Tibetan characters is operated smoothly with median filtering, not only eliminate noise but also keep the details of image and get rid of isolated noise, interruption and smooth stroke edge.

Separate the image of binary characters into 2×2 gridding, to get the complexity index in X and Y direction according to the method of measure the trend of binary coding, to get the complexity index of white pixel W and black pixel B according to the color of pixel. The last step is to get the complexity index of L and R which lean to the left and right, and to get classification to Tibetan characters by the BP neural network. To compare the results with the Tibetan characters feature extracted through the traditional algorithm of complexity index feature. Table is the construction and parameter about the experiment of BP neural network by using traditional algorithm. Table is construction and parameter about the experiment of BP neural network by using traditional algorithm has been improved.

TABLE II NEUTRON NETWORK ARCHITECTURE AND PARAMETERS

The node point number of input layer	The node point number of hidden layer	The node point number of output layer	Parameter learning	Target error	Iterations
28	10	10	0.01	0.001	5000

TABLE III NEUTRON NETWORK ARCHITECTURE AND PARAMETERS

The node point number of input layer	The node point number of hidden layer	The node point number of output layer	Parameter learning	Target error	Iterations
32	10	10	0.01	0.001	5000

TABLE IV THE DIFFERENT KINDS OF RECOGNITION RATE

The extraction algorithm of character feature	The traditional algorithm	The improved algorithm
Recognition rate	78.66%	82.53%

As Table , recognition rate of the extraction algorithm of complexity index feature has been improved is better than traditional one.

V. CONCLUSION

There are some problems of the traditional extraction algorithm of complexity index feature, such as impact factors of complexity index are too simple and it could not reflect the complexity index of images completely by counting value in x and y direction. In this paper, we put forward the improved extraction algorithm of complexity index feature is based on analyzing the feature of Tibetan language thoroughly, the way to describe statistical value of complexity index has been changed by this algorithm, we only use the value of influence before, but now we combine the weight coding and the impact factor instead, furthermore, the statistical feature of two-dimensional complexity index in traditional algorithm is instead of six-dimensional. A large amount of experimental data indicate that to compare with traditional extraction algorithm of complexity index feature the improved one to deal with Tibetan characters has the higher recognition rate and recognition speed.

ACKNOWLEDGMENT

This work was financially supported by Innovative Research Team of the department of Sichuan Province.(15TD0050) and the Fundamental Research Funds for Central University, Southwest University for Nationalities (13NZYQN10).

REFERENCES

[1] Chenxing Z, Bing Y. The Vast Vistas for Development of Tibetan Information Processing Technology[J]. Journal of Qinghai Normal University (Natural Science), 1999, (01): 9-15.

[2] Gang W, Xijiacuo D, Heming H. Printed Tibetan Character Recognition Technology[J]. Journal of Qinghai Normal University (Natural Science), 2006, (01): 32-37.

[3] Yuzhong C, Shiwen Y. The Tibetan Information Processing Technology Research Status and Prospect[J]. China Tibetology, 2003, (04): 97-107.

[4] Jiaohua Q, Xuyu X. Implementation and Amelioration about Chinese Feature Extraction with Complexity Index[J]. Computer Engineering and Design, 2006, (02): 265-267.

[5] Fujian F, Qian Z, Xin L, et al. The Extraction Method of Character Feature [J]. Software Guide, 2012, (01): 18-19.

[6] Xue G, Lianwen J, Junxun Y. A Stroke-Density Based Elastic Meshing Feature Extraction Method[J]. Pattern Recognition and Artificial Intelligence, 2002, (03): 351-354.

[7] Weilan W, Xiaoqing D, Kunyu Q. Study on Similitude Characters in Tibetan Character Recognition[J]. Journal of Chinese Information Processing, 2002, (04): 60-65.

[8] Hua W, Xiaoqing D. An Algorithm for Multi-Font Printed Tibetan Character Recognition[J]. Computer Engineering, 2004, (13): 18-20.

[9] Yulei W, Yongzhong L, Rushan W. Application of Rough Grid in Feature Extraction of Printed Tibetan Character[J]. Science Technology and Engineering, 2009, (18): 5546-5548.