COMPARISON OF IMAGE ANALYSIS FOR THAI HANDWRITTEN CHARACTER RECOGNITION

Olarik Surinta, Chatklaw Jareanpon
Department of Computer Science and Management Information System
Faculty of Informatics, Mahasarakham University
Mahasarakham City, Thailand
e-mail: olarik.s@msu.ac.th, Chatklaw.j@msu.ac.th
Telephone: +6643754322 ext 2497
Fax: +6643754359

Abstract: This paper is proposing the method for Thai handwritten character recognition. The methods are Robust C-Prototype and Back-Propagation Neural Network. The objective of experimental is recognition on Thai handwritten character. This is the result of both methods to be appearing accuracy more than 85%.

Keywords: Robust C-Prototype, Back-Propagation Neural Network, Thai Handwritten Character Recognition

1. INTRODUCTION

Pattern recognition scheme has numerously become a suffice tool utilized in character recognition. Generally, computer will be programmed to provide the functionality in order to classify each of character's property separately, defined as input character. These inputs will be determined for matching with the provided character patterns consequently. This paper proposes the offline processing, which the input data, gray scale of 256 levels. The processing is based on Thai characters on which preprocessing have been conducted. There are 44 Thai characters:

Please use the following format when citing this chapter:

Surinta, O., Jareanpon, C., 2006, in IFIP International Federation for Information Processing, Volume 228, Intelligent Information Processing III, eds. Z. Shi, Shimohara K., Feng D., (Boston: Springer), pp. 373–382.

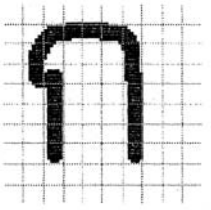
ก ข ฃ ค ต ฆ ง จ ฉ ช ซ ฌ ญ
ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท ธ น
บ ป ผ ฝ พ ฟ ภ ม ย ร ล ว
ศ ษ ส ห ฬ อ ฮ

2. DATA PREPROCESSING

Character-images are images of Thai handwritten characters written down on a piece of paper. The outputs are stored as digital data by scanning. One bitmap file with grey scale pattern (256 levels) specifies one character.

2.1 Edge Detection

Edge detection is an important step of the image processing phase. Detecting edges in any object has two important conventions: the object (image) must be a continuous image and the image must be scaled as black and white tone. This paper uses a Chain code technique to detect the image's edge. Simply put this technique move along the edge of the image and stops at the beginning position in order to get the image's edge. [1]



255	255	250	255	251	255	255	254	255	255
255	0	0	8	4	1	0	1	255	255
251	15	242	254	243	255	255	0	255	255
254	4	14	1	255	254	255	0	255	255
254	255	239	2	255	242	255	0	255	255
246	254	255	0	255	255	253	5	255	255
254	242	254	5	247	254	253	0	255	255
248	255	246	3	254	255	253	1	255	255
255	254	255	2	255	255	255	0	255	255
253	255	252	254	255	255	252	255	255	255

1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	1	1
1	0	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	0	1	1
1	1	1	0	1	1	1	0	1	1
1	1	1	0	1	1	1	0	1	1
1	1	1	0	1	1	1	0	1	1
1	1	1	0	1	1	1	0	1	1
1	1	1	0	1	1	1	0	1	1
1	1	1	1	1	1	1	1	1	1

(a) (b) (c)

Figure 1. A diagram of extracting the object component from the background component an the image.

2.2 Binarization

Binarization converts grey-level image to a black-white level image. Basically, an image is separated into two components. The first component is the object, while the other is background. The object in general is smaller (in size) than the background. To extract the object component from the background component, this scheme checks every point of pixel value with one intermediate value (called the Thresholding value). The thresholding value can be calculated from the following formula: [2]

$$g'(x,y)=\begin{cases} 0 \;\; if \;\; g(x,y)<T \\ 1 \;\; if \;\; g(x,y)\geq T \end{cases} \quad (1)$$

In this paper, data (the image file of the Thai handwritten character) is stored in bitmap pattern. The individual bit carries one of two possible values:

1 refers to background and,
0 refers to object or content

2.3 The extraction of the outer edge from the object component

From the previous step, we have got the object and have got rid of its background in which all bits contain 1. This step is then used to detect the outer rims (laying on the object's edge) of the object and to separate them from the object in order to get the edge. There are several methodologies used on this particular case, e.g. chain code, morphology, etc. This paper uses a chain code convention. [2]

The resulting image (from binarization) is actually a structural character, comprising many points lying on the image. Therefore, detecting the direction of those points has been applied in order to simplify the processing. This implementation is based on a chain code technique to change the points to a numerical representation. Eventually, the direction is classified by 8 categories.

Once the edge of the image has discovered, the process needs to find the character line. The coordinate (x_k, y_k) is then represented by a complex number as the following formula:

$$u_k = x_k, iy_k \quad (2)$$

2.4 Fourier Descriptors

Fourier Features are used to describe an edge of an object. They work by identifying coordinates (x_k, y_k); $k = 0,1,\ldots, N-1$ where N is any other area in the image. All points (x_k, y_k) are represented as complex numbers, shown as: [3, 4, 5]

$$u_k = x_k + iy_k \quad (3)$$

Where

$$i = \sqrt{-1}$$

Therefore, the DFT (Discrete Fourier Transform) (f_l) can be derived from:

$$f_l = \sum_{k=0}^{N-1} u_k \exp\left(-j\frac{2\pi}{N}lk\right) \quad (4)$$

Where

$l = 0,1,...,N-1$

From the above formula, a coefficient vector is automatically calculated. This vector fits as one dimension with the size of (1×10) or $(1\times n)$

3. FUZZY C-MEAN (FCM) ALGORITHM

The FCM algorithm minimizes the following objective function [6]

$$J_F(B,U;X) = \sum_{i=1}^{C}\sum_{j=1}^{N}(u_{ij})^m d_{ij}^2 \quad (5)$$

Where

$u_{ij} \in [0,1]$ and $\sum_{i=1}^{C} u_{ij} = 1 \quad \forall \quad j$

X = the set of feature vectors $\bar{x}_j$

$j = 1,2,...,N : N$ represents the image pixel

C = the group of images

$m \in [1,\alpha)$ = Fuzzifier

d_{ij}^2 = the distance from Feature Vector to the group of patterns

u_{ij} = member indicator of x_j in β_i

$B = (\beta_1,...,\beta_C)$ = C-tuple indicating C - Cluster

$U = \lfloor u_{ij} \rfloor$ = matrix, $C \times N$ in size, representing the condition

3.1 Object-Function Minimization in Robust C-Prototypes (RCP)

RCP can be determined in grouping phase in order to estimate C-Prototypes spontaneously, utilizing loss function (ρ) and square distance to reduce some noise. The definition can be expressed as: [6, 7]

$$J_F(B,U;X)=\sum_{i=1}^{C}\sum_{j=1}^{N}(u_{ij})^m \rho_i d_{ij}^2 \tag{6}$$

Where

$$u_{ij}\in[0,1] \text{ and } \sum_{i=1}^{C}u_{ij}=1 \quad \forall j$$

The highest efficiency can be calculated as follow:

$$u_{ij}=\frac{1}{\sum_{k=1}^{C}\left[\frac{\rho_i(d_{ij}^2)}{\rho_i(d_{ik}^2)}\right]^{\left(\frac{1}{m-1}\right)}} \tag{7}$$

The best portion of weight function is that function with significant impact and efficiency in the RCP-Algorithm. This weight function used for general estimation, has been designed as symmetric distribution, whose center is the original point that differs from RCP-Algorithm used in this experiment. Therefore, the weight function must be rebuilt:

$w:\Re^+\rightarrow[0,1]$ Subject to

$\Im_1: w(d^2)$; Monotonically function

$\Im_1: w(d^2)=0$ for $d^2>T+\alpha S$

T and S are robust estimates of the average of the square of distance and standard deviation, while α is any constant.

$$p_1: w(0)=1, w(T)=0.5, w'(0)=0$$

T and S are truly important for constructing the weight function. Therefore, the process needs the efficiency estimation. Obviously, Med-Median and Median of Absolute Deviation have been identified to be used in estimation, which is:

$$T_i=\underset{x_i\in X_j}{Med}(d_{ij}^2) \text{ and} \tag{8}$$

$$S_i=1.418\times\underset{x_i\in X_j}{MAD}(d_{ij}^2) \tag{9}$$

Where

$$X_i=\left\{x_j \mid d_{ij}^2\le d_{kj}^2 \quad \forall k\neq i\right\}$$

The weight function $w:\Re^+\rightarrow[0,1]$ is defined as $\Im_1-\Im_3$, which is :

$$w_i(d^2)=\begin{cases} 1-\dfrac{d^4}{2T_i^2} & if\ d^2\in[0,T] \\ \dfrac{\left[d^2-(T_i\alpha S_i)\right]^2}{2\alpha^2 S_i^2} & if\ d^2\in[T_i,T_i+\alpha S_i] \\ 0 & if\ d^2>T_i+\alpha S_i \end{cases} \quad (10)$$

The Loss function derived from the weight function can be calculated regarding to (10)

$$\rho_i(d^2)=\begin{cases} d^2-\dfrac{d^6}{6T_i^2} & if\ d^2\in[0,T] \\ \dfrac{\left[d^2-(T_i+\alpha S_i)\right]^3}{6\alpha^2 S_i^2}+\dfrac{5T_i+\alpha S_i}{6} & if\ d^2\in[T_i,T_i+\alpha S_i] \\ \dfrac{5T_i+\alpha S_i}{6}+K_i & if\ d^2>T_i+\alpha S_i \end{cases} \quad (11)$$

K_i is a constant;

$$K_i=\max_{1\le j\le c}\left\{\frac{5T_j+\alpha S_j}{6}\right\}-\frac{5T_i+\alpha S_i}{6}$$

for $i=1,...,C$

In (11), K_i must be added in order to impede any noise. This will force the total values of the loss function to be at least the average value of normal data, which every point of noise has the same member value.

3.2 Calculating the distance between groups of images and the relation among them

The required conditions to adjust the pattern characters for the Mahalanobis Distance is

$$d_{ij}^2=\left(\bar{x}_j-\bar{c}_i\right)^T M\left(\bar{x}_j-\bar{c}_i\right) \quad (12)$$

$\bar{x}_j$; The feature vector of group of data

$\bar{c}_i$; Center vector of each cluster

M_i ; Symmetric vector, which is a positive definite matrix derived from

$$\bar{c}_i = \sum_{j=1}^{N} \left(u_{ij}\right)^m w_{ij} \bar{x}_j \Big/ \sum_{j=1}^{N} \left(u_{ij}\right)^m w_{ij} \tag{13}$$

and

$$M_i = \left|R_i\right|^{1/n} R_i^{-1} \tag{14}$$

Where

$$C_i = \sum_{j=1}^{N} \left(u_{ij}\right)^m w_{ij} \left(\bar{x}_j - \bar{c}_i\right)\left(\bar{x}_j - \bar{c}_i\right)^T \Big/ \sum_{j=1}^{N} \left(u_{ij}\right)^m w_{ij}$$

C_i is the "Robust Fuzzy Covariant Matrix"

4. ARTIFICIAL NEURAL NETWORKS

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Figure 2 shows the diagram of a neural network. [8]

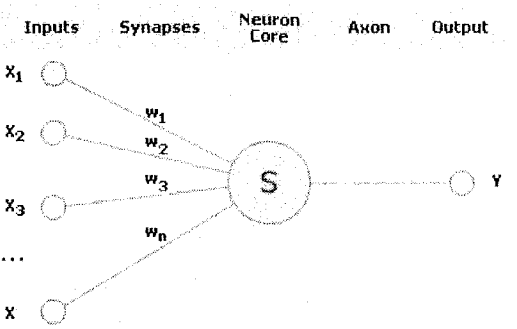


Figure 2. A diagram of a neural network [8]

Figure 2 shows that an artificial neuron consists of synapses connecting the neuron inputs with the nucleus, a neuron nucleus processing input signals and an axon connecting the neuron with those of the next layer. Every synapse has its own weight, which determines how the corresponding neuron input influences its condition. The neuron condition is calculated by the following formula:

$$S = \sum_{i=1}^{n} x_i w_i \tag{15}$$

Where

n = number of neuron inputs

x_i = neuron input value

w_i = synapse weight

4.1 Back-Propagation Neural Network

Back-Propagation neural networks are tools for searching regularities, forecasting, and qualitative analysis. Back propagation neural network use a learning algorithm use in which an error moves from the output layer to the input one.

Back-Propagation neural networks consists of several neuron layers, each neuron of layer i being connected to each neuron of layer $i+1$.

The task of training neural network comes down to finding a functional dependence $y = f(x)$, where x is an input vector and y is an output one. In the general case this task with a limited set of input data has an infinite set of solutions. To limit the search space during the training, the task is allotted to minimize the efficiency function of the neural network error, which is found with the least squares estimator.

$$E(w) = \frac{1}{2}\sum_{j=1}^{p}(y_j - d_j)^2 \tag{16}$$

Where

y_j = network output value

d_j = target value of output

p = number of neurons in the output layer

The neural network training is conducted by the gradient descent method, in each iteration the weight change is made according to the following formula:

$$\Delta w_{ij} = -\eta . \frac{\partial E}{\partial w_{ij}} \tag{17}$$

Where

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{dS_j} \cdot \frac{\partial S_j}{\partial w_{ij}}$$

Where

y_j = neuron output value,

S_j = weighted total of output signals determined by the formula (15).

With the multiplier

$$\frac{\partial S_j}{\partial w_{ij}} \equiv x_i \tag{18}$$

Where

x_i = the neuron input value

In the training phase, the correct class for each record is known (this is termed supervised training), and the output nodes can therefore be assigned "correct" values -- "1" for the node corresponding to the correct class, and "0" for the others. (In practice it has been found better to use values of 0.9 and 0.1, respectively.) It is thus possible to compare the network's calculated values for the output nodes to these "correct" values, and calculate an error term for each node (the "Delta" rule). These error terms are then used to adjust the weights in the hidden layers so that, hopefully, the next time around the output values will be closer to the "correct" values. [9]

5. EXPERIMENTAL RESULT

The data in this experiment have 2 sets. The first set is "the learning set" containing 4,400 characters. The second set is the "test set" containing 440 characters. All data is Thai handwritten and generated by 100 persons.

The experiment is divided into two parts. The first part is "Robust C-Prototype". And the second part is "Back-Propagation neural network". The features of these structures are shown in table 1. The comparisons of both methods are shown in table 2.

Table 1. Back-Propagation Neural Network Structure

Training Algorithm	Back-Propagation
Performance Function	Mean-Square Error
Performance Goal	0.002
Minimum Gradient	le-9

Table 2. Comparison between Robust C-Prototype and Back-propagation neural network

Topic	Robust C-Prototype	Back-Propagation neural network
Time of learning	1.5 Hour	2.45 Hour
Accuracy test on "Test set"	91.5%	88%

6. CONCLUSION

This paper looks at Thai handwritten character recognition. We compared Robust C-Prototype and Back-Propagation neural networks. The Experimental results of both methods have accuracy more than 85%. This paper is concerned with the recognition of only a single character. Future work is to recognize entire handwritten words or signatures.

ACKNOWLEDGEMENTS

We are grateful for the support of the Faculty of Informatics at Mahasarakham University.

REFERENCES

1. Castleman, K. R. Digital Image Processing. Prentice Hall. NJ, n.p. , 1995.
2. Sergios Theodoridis, Konstantinos Koutroumbas. Pattern Recognition. Academic Press 24-28 Oval Road London: Department of Informatics. University of Athens; 1998.
3. Yerin Yoo. Tutorial on Fourier Theory. n.p. , 2001.
4. Yi Lu, Steven Schlosser, Michael Janeczko. Fourier Descriptors and Handwritten Digit Recognition. Machine Vision and Application. 1993.
5. B. Pinkowski. Fourier Descriptors for Characterizing Object Contour. Western Michigan University.
6. Hichem Frigui and Raghu Krishapuram. A Robust Algorithm for Automatic Extraction of an Unknown Number of Clusters from Noisy Data. Pattern Recognition Letters(17). 1223-1232, n.p. , 1996.
7. Olarik Surinta and Supot Nitsuwat. Handwritten Thai Character Recognition Using Fourier Descriptors and Robust C-Prototype. Proceeding of The National Conference on Computing and Information Technology (NCCIT2005). Bangkok, Thailand. 24-25 May 2005.
8. Alexey Starikov. Neural Networks. <http://www.basegroup.ru/neural/math.en.htm> 5 May 2006.
9. Chatklaw Jareanpon and Olarik Surinta. Handwritten Recognition with Neural Network. Proceeding of International Workshop on Advanced Image Technology (IWAIT2006). Okinawa, Japan. 9-10 January 2006.