INVARIANT THAI PRINTED CHARACTERS RECOGNITION

PISIT PHOKHARATKUL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF ENGINEERING IN ELECTRICAL ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUE OF TECHNOLOGY LADKRABANG

2002

ISBN 974-9546-94-6

หัวข้อวิทยานิพนธ์	การรู้จำตัวอักษรพิมพ์ภาษาไทยแบบอินแวเรี้ยนท์
นักศึกษา	นายพิศิษฏ์ โภคารัตน์กุล
รหัสประจำตัว	40060004
ปริญญา	วิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2545
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.บุญธีร์ เครือตราชู
อาจารย์ผู้ควบคุมวิทยานิพนธ์ร่วม	รศ.ดร.ชม กิ้มปาน

บทคัดย่อ

ในวิทยานิพนธ์นี้เป็นการตรวจสอบพิเคราะห์วิธีการรู้จำแบบริงโปรเจ็คชั่น (ring projection method) และตัวอธิบายฟูเรียร์ (Fourier descriptors method) ในการรู้จำตัวอักษรตัวพิมพ์ภาษาไทยแบบอินแวเรี้ยนท์ (invariant Thai printed character) โดยเฉพาะกรณีการเลื่อนตำแหน่ง การหมุนและขนาดที่แตกต่างกันของภาพตัวอักษร ปัญหาความยุ่งยากของการรู้จำแบบอินแวเรี้ยนท์อยู่ที่จำนวนข้อมูลจำนวนมากที่ใช้ในการฝึกของระบบการรู้จำ ซึ่งในวิทยานิพนธ์นี้แบ่งการพัฒนาออกเป็น 2 เฟสดังต่อไปนี้

เฟสแรกเป็นการพัฒนาระบบการรู้จำโดยวิธีริงโปรเจ็คชั่นและรัฟเซ็ต วิธีริงโปรเจ็คชั่นเป็นวิธีที่ไม่ขึ้นต่อการหมุน การเลื่อนตำแหน่ง และขนาดของตัวอักษร โดยอาศัยหลักการนับจำนวนจุดภาพของตัวอักษรทั้งหมดที่กระจายอยู่ในแต่ละรัศมีวงกลม ซึ่งจะเก็บค่าข้อมูลเหล่านี้ไปกำหนดเป็นแอตทริบิว (attribute) ของแต่ละวัตถุ (object) ระบบรัฟเซ็ตจะถูกใช้ในการลดจำนวนข้อมูลของแอตทริบิวและสร้างกฎ เพื่อการตัดสินใจในการแบ่งกลุ่มข้อมูลแบบหยาบและละเอียดต่อไป

เฟสที่สองเป็นการพัฒนาระบบการรู้จำโดยใช้ตัวอธิบายฟูเรียร์และโครงข่ายเซลล์ประสาท โดยตัวอักษรจะถูกแบ่งกลุ่มในตอนแรกโดยใช้จำนวนวงรอบภายในของภาพตัวอักษร จากนั้นจะแบ่งกลุ่มของตัวอักษรต่อโดยใช้สัมประสิทธิ์ของตัวอธิบายฟูเรียร์ของวงรอบภายนอกของภาพตัวอักษร โครงข่ายเซลล์ประสาทแบบหลายชั้นจะถูกใช้เป็นตัวแบ่งแยก เรียนรู้และคำนวณค่าน้ำหนักของสัมประสิทธิ์เหล่านี้เพื่อใช้ในการแบ่งกลุ่มข้อมูล

ในการทดสอบกับตัวอักษรทั้งหมด 2,752 ตัวอักษรประกอบด้วย 4 ฟอนท์ คือ อังสนานิว บราวน์วาเลียนิว คอเดียนิว ยูคอเซีย ยูพีซี แต่ละฟอนท์มี 4 ขนาดคือ 14 ,18 , 22 และ 28 ปรากฏว่าอัตราการรู้จำ 87% สำหรับวิธีริงโปรเจ็คชั่นและรัฟเซ็ต และ 91% สำหรับวิธีตัวอธิบายฟูเรียร์และโครงข่ายเซลล์ประสาทตามลำดับ การแบ่งกลุ่มผิดในเฟสแรกเป็นสาเหตุที่ทำให้อัตราการรู้จำต่ำกว่าวิธีในเฟสที่สอง การแบ่งกลุ่มตัวอักษรในระหว่างกลุ่มที่มีความสลับซับซ้อนมีความสำคัญมาก ซึ่งการหาตัวแบ่งกลุ่มที่ดีจะช่วยให้อัตราการรู้จำสูงขึ้น

Thesis Title	Invariant Thai Printed Characters Recognition
Student	Mr. Pisit Phokharatkul
Student ID.	40060004
Degree	Doctor of Engineering
Programme	Electrical Engineering
Year	2002
Thesis Advisor	Assoc. Prof. Dr. Boontee Kruatrachue
Thesis Co-Advisor	Assoc. Prof. Dr. Chom Kimpan

ABSTRACT

This thesis investigated ring projections and Fourier descriptors in an invariant Thai printed characters recognition system. In particular we emphasise character invariance with respect to translation, rotation, and scale differences in input character images. The problem of invariance is difficult because of the large number of training samples for which the classifier needs to be trained. This thesis development consists of two phases.

The first phase involves a recognizer using a ring projection and rough set method. The ring projection method is invariant to rotations, translations, and scales, which are based on the total number of foreground pixels as distributed along circular rings. Then the object's attributes are set up by these values. The rough set system is used to reduce the object's attribute data and generate the decision-making rules for coarse and fine classification.

The second phase involves a recognizer using Fourier descriptors and neural networks. A character is first classified into group based on the number of inner perimeter of a character. Then the group is further classified by Fourier descriptor coefficients of the outer perimeter of a character. Multi-layer neural network is used as a classifier to learn these coefficients of each group and compute the weighting of networks for classification.

The test character set consists of 2,752 characters (four fonts, Angsana New, Browalia New, Cordia New, Eucrosia UPC, four sizes in each font, 14, 18, 22, and 28 points). The recognition rate 87% accuracy for ring projection and rough set method and 91% for Fourier descriptor and neural network method respectively. Some mis-classified to be groups are caused in the first phase, which causes the total rate of recognition to be lower than the method in the second phase. Confusing classes are hard to distinguish. The classification of characters relies on how good the discriminator is. In doing so, confusing classes are better separated, to increase the recognition rate.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Assoc. Prof. Dr. Boontee Kruatrachue, and Assoc. Prof. Dr. Chom Kimpan for having created a very stimulating and well-equipped Lab environment, through many years of dedicated efforts.

I am grateful to all the people who helped me one way or another during those years at KMITL. Special thanks go to Tirach Suebsanit, Worakran Wongsaichua, Ukrit Marang, and Sombut Foitong for their technical competence, for maintaining our computer network.

I am also thankful to Prof. Dr. Makoto Sato of Tokyo Institute of Technology, Japan, for helpful exchanges of ideas.

I am thankful to Dr. Nara Dhiranuta for her support and help through my studies.

I am thankful to Mr. Seri Pansang, Mr. Adirak Kanchanaharuthai, Mr. Watjanapong Kasemsiri, Mr. YingYod Sriboourong, Ms. Rachada Kongkrachandra, and Mr. Kriengsak Tamee for their friendships.

I would like to express my gratitude to my examiner Assoc. Prof. Dr. Kosin Chamnongthai as well as to the other members of my examining committee, Assoc. Prof. Dr. Ouen Pin-Ngem, Assoc. Prof. Dr. Manus Sangworasilp, Assoc. Prof. Dr. Boonwat Attachoo, Assoc. Prof. Dr. Chom Kimpan, and Assoc. Prof. Dr. Boontee Kruatrachue. Their comments helped to improve the quality of the final version of this thesis.

I would like to thank the Japan International Cooperation Agency (JICA), and Research Center for communication and Information Technology (RECCIT) for their continuing support and guidance during this study.

Special thanks to Mr. Graham Rogers and Dr. Chatchai Neatpisarnvanit of Mahidol University, Thailand for their help with the manuscript.

Finally, I would like to thank my father, mother, and family for motivating me to pursue my studies.

Pisit Phokharatkul

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (CONTINUED)

LIST OF FIGURES (CONTINUED)

CHAPTER 1
INTRODUCTION

1.1 Character recognition

Character recognition is a topic that has been investigated for many years. No solution has been offered that solves the problem both efficiently and completely because the problem is complex in nature. This problem depends on a process that segments the input into individual characters, and the recognition methods.

In the case of Thai character recognition, there are many approaches that have been tried to make computers recognize the characters. The preliminary stage of this field, which concerns the use of the matching method was described by Kimpan, Itoh, and Kawanishi (1983). Subsequently, the topological properties Kimpan (1989) are used to improve recognition accuracy. In 1987, the same author proposed the recognition method of printed Thai characters using the Karhunen Loève expansion by Kimpan, Itoh, and Kawanishi (1987). Hiramvanichakorn, Agui and Nakajima (1982, 1984) used local features of Thai characters, such as convexity and concavity of stroke in their recognition system. Airpaiboon et al. (1989, 1996) used the heads of Thai characters to recognize the handprinted Thai characters. From the example methods as mentioned the recognition rates of these methods were satisfactory, but there is the rotated and scaling character problem.

1.2 Character pattern recognition system

An OCR system (David Freedman M., 1974) such as shown in Figure 1.1, is made up of three major subsystems: an electrooptical converter, a preprocessor, and recognition method. The electrooptical converter scans the characters and converts them into electrical signals for further processing and at the same time, filters the noise out from them.

The preprocessor examines the output of the scanner and performs quantization. In addition, the preprocessor will transform the signal suitable for the operation of recognition method and to make it easier to store them.

The recognition method examines the output of the preprocessor, locates and separates the data derived from each character, and then processes these data to determine the identity of each character.

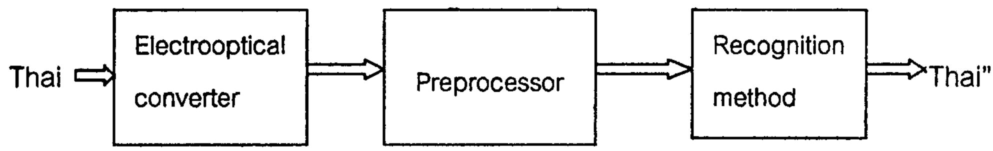


Figure 1.1 Major components of an optical character recognition system are shown between "input" and "output"

1.3 System overview

The system consists two major phases as illustrated by Figure 1.2. The first is a training phase, where we attempt to find the patterns that best represent the shapes of a given class. Training is typically done once at the creation of the system, and kept for the recognition of the unknowns.

The second phase of the system consists of a recognition phase where, once the set of models is established, unknowns are presented to the system. The value of unknowns is computed and compared with the model base. If the value of an unknown falls in rule-based or a given threshold, the pattern will be identified as belonging to a certain class.

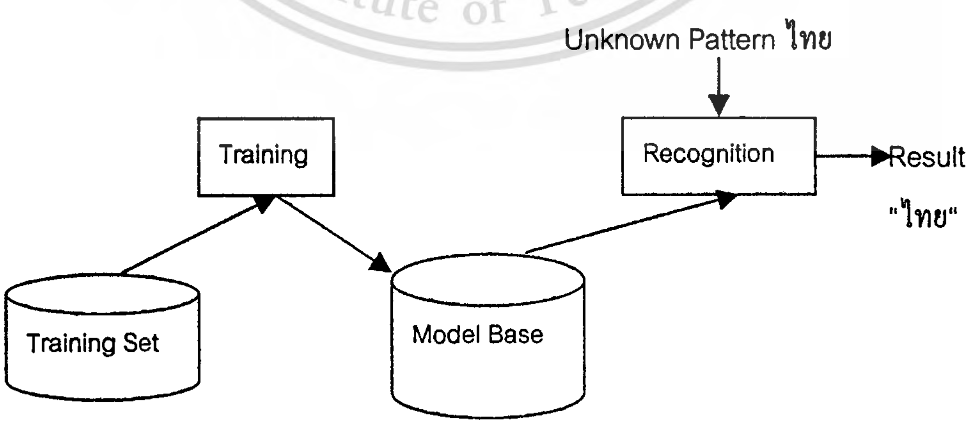


Figure 1.2 Overview of the system architecture

1.4 Objectives of the study

The objectives of the study are:

1. To design a hybrid algorithm of the ring projections and rough sets for invariant printed Thai character recognition system.
2. To design a hybrid algorithm of the Fourier descriptors and neural networks for invariant printed Thai character recognition system.
3. To achieve high recognition accuracy.

1.5 Scope of the study

The scope of the study will be limited to a hybrid implementation of the ring projections and rough sets, and the Fourier descriptors and back-propagation neural networks for invariant printed Thai characters recognition system.

To implement the ring projections and rough sets experiment, simulation programs are developed using Matlab 5.3 and run in Window 98 environment of Compaq personal computer Pentium 60.

In the case of Fourier descriptor and neural networks experiment, simulation programs are developed using turbo C++ version 3.0 and DOS environment of Compaq personal computer Pentium 60.

1.6 Thesis outlines

The first few chapters of this thesis provide some essential background and a summary of related work in pattern recognition:

Chapter 1 Introduction

Chapter 2 Literature Review

Chapter 3 reviews the Thai character and preprocessing

Chapter 4 reviews the field of neural networks

Chapter 5 reviews the field of rough sets. The remainder of the thesis describes our own research, evaluating hybrid system

Chapter 6 presents our research with ring projections and rough sets, and explains the core, reducts, generating rules from reducts, and the classification results

Chapter 7 presents our research with Fourier descriptor and neural networks, and the classification results.

Chapter 8 presents the discussions of this thesis.

Chapter 9 presents the conclusions of this thesis.

CHAPTER 2

LITERATURE REVIEW

This chapter contains a brief overview of literature related to the invariant character recognition. The first section presents the related literature on invariant recognition method, and the second section is a brief survey of the rough sets in character recognition.

2.1 Invariant recognition methods

Pattern recognition and character recognition, in particular, has been attempted with many different systems and algorithms. We will briefly look at the special invariant recognition method.

The aim of invariant recognition is to identify an object independently of its position (translated or rotated) and size (larger or smaller). In invariant pattern-recognition models, preprocessing is defined as the extraction of appropriate invariant features that are then used for recognition by a classification system.

2.1.1 Moment method

Moment invariants have become a classical tool for object recognition since the last 30 years. The moment invariant can be derived from the definition (Nadler and Smith, 1992):

$$M_{pq} = \iint x^p y^q \ f(x, y)\, dx\, dy \qquad (2.1)$$

For instance, the image parameter I(x,y), and p,q are positive integers 0,1,2, ... Integration is over the entire plane. Discrete pixels x,y are their coordinates. We replace the double integrate by a double summation. We can interpret M_{00} as the total blackness ("mass") of the image. M_{20} and M_{02} as the moments of inertia of the pattern about the x- and y-axes, and so on. It can be shown that under certain general conditions an infinite sequence of the coefficients M_{pq}, where p, q =0, 1, 2, ..., ∞, is uniquely determined by $f(x,y)$ and, conversely, $f(x,y)$ is uniquely determined by the infinite sequence M_{pq}.

2.1.2 Fourier descriptors

The pattern recognition problem can be viewed as a signal-processing problem where the closed contour of a pattern is considered as a periodic signal. Such a signal can be expressed as a series of sums of complex numbers as given by the coefficient of the Fourier series (Persoon and Fu, 1977). This method is based on using the coordinate values of the pixels of the character primary part to obtain the Fourier descriptors. The character's closed boundary can be represented by a periodic function in a two-dimension x-y plane tracing once around it. The character contour yields a complex function that denotes the parametric representation of the boundary coordinates. The normalized Fourier descriptors are invariant with respect to translation, rotation, and scale of characters. Hence, larger pitch characters may be recognized without the modification of the model features. However Fourier coefficients contain enough information to properly reconstruct the original shape suggesting that they do keep the bulk of the information. A review of the works shows that Fourier descriptors have been used in the past for recognizing characters as following.

Persoon et al (1977) introduced Fourier descriptors (FDs) as a form of shape discrimination. In this research Fourier descriptor is used for obtaining skeletons of objects, and uses optimal curve matching between two objects for recognizing.

Man and Poon Joe (1992), Chunge and Wong (1997) introduced hybrid Fourier descriptors and neural networks. This algorithm uses Fourier descriptors as unique features in representation of contours. The characteristic of this algorithm is to represent the object by several sets of FDs, which represent different portions of the object. The learning approach in this method uses back-propagation neural networks.

Mahmoud (1994) presented the use of Fourier descriptors for the Arabic recognition system. The recognition method based on estimating the Fourier descriptors and curvature features of the contour of the primary part of Arabic characters. Ten Fourier descriptors, 16 direction and direction length features, and 20 concave and convex features for the quadrants of a character and for the whole character are extracted. In addition, the number and location of dots and the number of holes are estimated by two methods. The first method estimates the number from external contours and the other by dividing the total area of dots by the average dot area found in the training phase. The use of several features is necessary to achieve higher recognition rates of Arabic characters.

Taxt and Bjerde (1994) applied the properties of elliptic Fourier descriptors for the classification of handwritten vector symbols.

Szmurlo (1994) investigated the application of Fourier series in pattern recognition. This technique is used in obtaining the starting point shift and normalizes it by rotating the coordinates of the contours.

Jeong Cha-Sup and Jeong Dong-seok (1999) used Fourier descriptors and contour information for handwritten digit recognition. This technique uses the similarity function to measure the similarity distance between normalized Fourier descriptors of input digit.

Chen and Bui (1999) presented the use of Fourier-wavelet descriptors for pattern recognition. This technique transforms it to polar coordinate using the center of mass of the pattern as origin; then applying 1-D Fourier transform and wavelet transformation with the features in polar form.

Phokharatkul et. al. (2000) invented the hybrid Fourier descriptors between genetic-neural networks and fuzzy neural networks for the Thai printed characters recognition system.

There are several interesting research issues in using Fourier descriptors for pattern recognition. It has many useful properties, one of which is that shifts in the time domain do not affect the spectrum in the frequency domain, i.e. Fourier transformation is translation invariant with respect to the spectrum.

2.1.3 Rings projection

In recognition of digital figures, many methods have been introduced. However, it is a well-known fact that in many real-life pattern recognition situations, such as optical character recognition, patterns are often found to be rotated due to experimentation constraints or errors. This implies that a new pattern recognition method must be developed that is invariant to rotations. Then, a two-dimensional histogram is proposed to transform 2-D patterns into 1-D pattern obtained from ring projection invariant to rotations Tang et. al. (1996, 1998). Ring projection is defined as the total number of foreground pixels as distributed along circular rings. These have a radius dependent on a centroid of the shape, but independent of the shape's position, orientation, and scale.

Tang et. al. (1998) presented a recognition method that utilizes ring-projection-wavelet-fractal signatures. This technique employed a ring projection method to reduce the dimensionality of the origin input pattern, and a wavelet transformation is used to transform the derived pattern into a set of subpatterns, from which the fractal dimensions can readily be computed.

Phokharatkul et. al. (2001, 2002) used the hybrid ring projection and rough sets. A rough set method is used to transform examples in a reduced set of rules. This method uses rough set theory, concepts of core and reduct of knowledge, and decision-making of max-min boundaries.

In this thesis, we proposed the procedures to develop the invariant characters recognition for multi fonts printed Thai character, which will be described in chapters 6 and 7 respectively.

2.2 The application of rough set theory in characters recognition

Rough set theory constitutes a framework for inducing minimal decision rules. These rules in turn can be used to perform a classification task. Rough sets have already been applied to a very wide variety of application domains with satisfactory results. There has been an attempt to make application of this theory into pattern recognition, but there are few implementations.

Nejman (1995) presented the use of rule based and matching method in the handwritten numeral recognition.

Kim and Bang (1999) proposed the tolerant rough set for handwritten numeral recognition. This proposed method uses the similarity measure between two data. It is described by the distance function of all constituent attributes and they are defined to be tolerant when their similarity measure exceeds a similarity threshold value.

Kasemsiri et. al. (2000, 2001) used rough sets and the hybrid of fuzzy-rough sets for printed Thai characters recognition, but they do not consider invariant problem as in the hybrid ring projection and rough sets (Phokharatkul et. al., 2001, 2002).

Furthermore, there has been development in rough sets with other fields such as image processing, data mining, and power system. The detail of these applications of rough set theory for invariant Thai printed character recognition will be described in the next chapters.

CHAPTER 3

THAI CHARACTERS AND PREPROCESSING

3.1 Thai characters

Thai characters consist of 42 consonants, 18 vowels, 4 controlled voice tones, and 3 special symbols as shown in Figure 3.1. As a result, 87 fundamental characters (including Thai numerals and Arabic numerals) are obtained.

The structure of most Thai characters consists of small loops (heads of character) combined with curves and lines. The heads appear in various locations of each character. By counting the number of Thai characters, we can categorize three groups of Thai characters.

- No-head characters: eg. "ก", "ธ", "า".
- One-head characters: eg. "ง", "ค", "ว", "ภ".
- Two-headed characters: eg. "น", "ม", "ฆ", "ห".

Writing a Thai character always starts with a head (if the character has a head or loop) followed by the other component. Most Thai characters consist of only one stroke, except "ฑ", "ฬ", and "+".

A Thai language sentence, such as that in Figure 3.2, is composed of consonants, and vowels on different levels. The vertical level can be divided into four parts as shown in Figure 3.3.

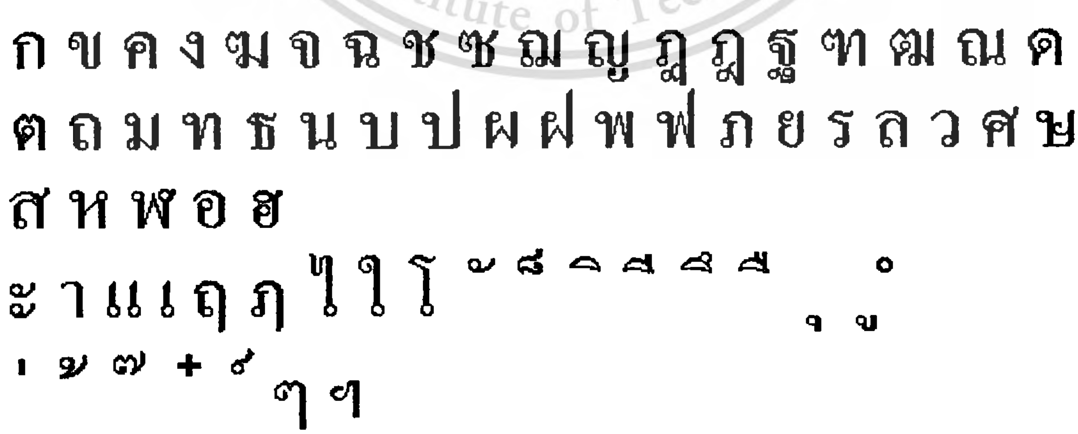


Figure 3.1 Thai character set consists of 42 consonants, 18 vowels, 4 controlled voice tones, and 3 special symbols

In general, a Thai word is noncursive in printing. This mean that there is no connected part between two consecutive characters in a word. This suggests easy segmentation of each character from a word. Unfortunately, the Thai word has a complicated line level structuring, as shown in Figure 3.2. Overlapping of characters will occur when a word consists of consonants, vowels, and tonal symbols, as shown in Figure 3.2. From Figure 3.2, the vowel "◌้" overlaps with both consonants: eg. "น" and "ก", "จ" and "ย", "จ" and "ด",

Tonal symbols are located on the tonal line level. However, when a word has no upper vowel characters, the location of tonal symbols will be shifted down to the position of the upper vowel line level.

นับตั้งแต่โครงการจัดตั้งสำนักวิจัยการสื่อสารและ
เทคโนโลยี่สารสนเทศได้ก่อตั้งขึ้นเมื่อปี พ.ศ. 2540

Figure 3.2 Example of Thai language sentence. Translation: Since research center for communications and information technology has been in B.E. 2540

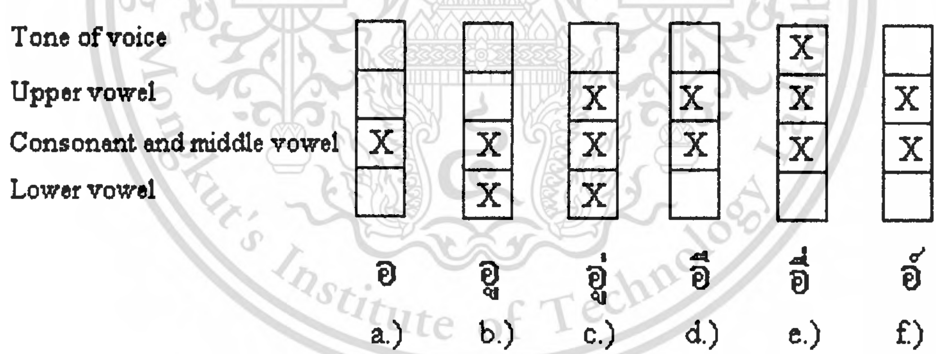


Figure 3.3 Composition of Thai character word or sentence. The Thai language sentences are composed of consonants, vowels and controlled voice tones, on different levels

The difficulty of Thai character recognition is the problem of similar characters. The consonant characters "ท,ฑ", "ฤ,ฦ", "ถ,ภ,ย,", "ค,ด,ต", "ฆ,ณ", and "ฎ,ฏ" are examples of similar characters. The differences are the stroke of head, and the fork point of the characters, which lead to the problem of recognition.

3.2 Contour invariant in digital images

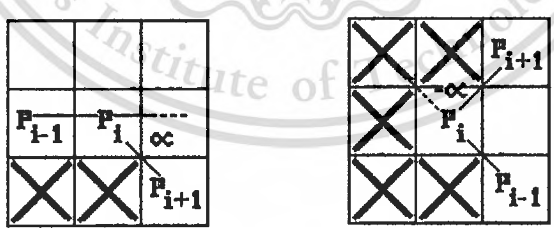
We will describe the property of invariance to digital images. Discussing the contour invariant provides an universal theory for 2D digital images which is completely independent of neighborhood used. The basic definition on discrete geometry is defined as follows: (XIA, 1995)

Definition 3.1: Let P_{i-1}, P_i, and P_{i+1} be three consecutive contour pixels and separate its j -neighbors into object side and background side. Let $\angle P_{i-1} P_i P_{i+1}$ be the interior angle (in objects side) at P_i. Local curvature at P_i LC(P_i) represents the complementary angle of $\angle P_{i-1} P_i P_{i+1}$ as shown in Figure 3.4. Note that X pixels are interior points.

The value of local curvature LC(P_i) ≥ 0 if $\angle P_{i-1} P_i P_{i+1}$ is convex (Figure 3.4 (a)) or LC(P_i) < 0 when $\angle P_{i-1} P_i P_{i+1}$ is concave (Figure 3.4 (b)). As local curvature in Figure 3.5 is an example of different local curvature in 8-neighbors. By the definition of contour, LC may have discrete values { -2, -1, 0, 1, 2, 3, 4}. In the Figure 3.5, X pixels designate object/contour pixels, a circle pixel stands for the current edge points, and the numeric in center represents LC at these points.

Definition 3.2 : Total curvature TC(∂C) is the sum of local curvature LC on a contour ∂C and denoted by $\sum_i LC(P_i \mid P_i \in \partial C)$.

Corollary 3.3 : In digital images, TC(∂C^+) = - TC(∂C^-) = N, where N equals 8 for rectangular tessellation and 6 for hexagonal tessellation.



(a) (b)

Figure 3.4 Local curvature in 8 — neighbors

In this research, digital boundary tracing of characters is made and the traced boundary is coded by the definition of contour invariant. The definition of the contour invariant is used to obtain the local curvature codes of each Thai character. The definition of contour invariant is independent of contour-tracing orientation. Figure 3.6

illustrates the resulting images in which *a* and *b* represent local curvature values—1 and —2 on contour pixels. The local curvature of contour invariant is used for smoothing and to compute the concavity and convexity of boundary in the next section respectively.

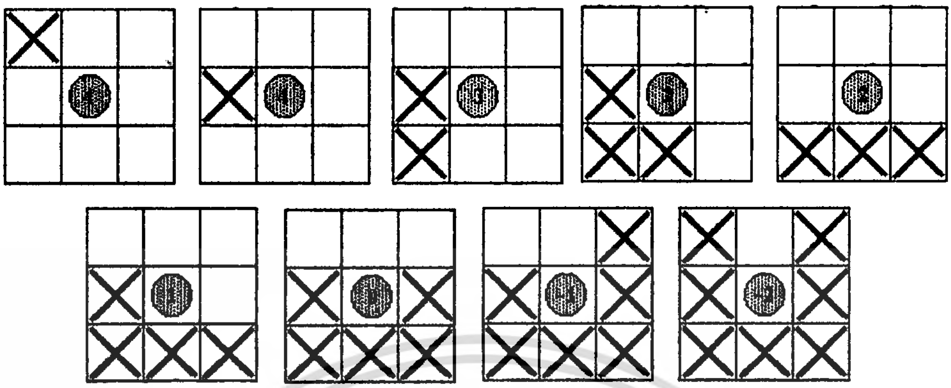


Figure 3.5 Configuration of different local curvatures

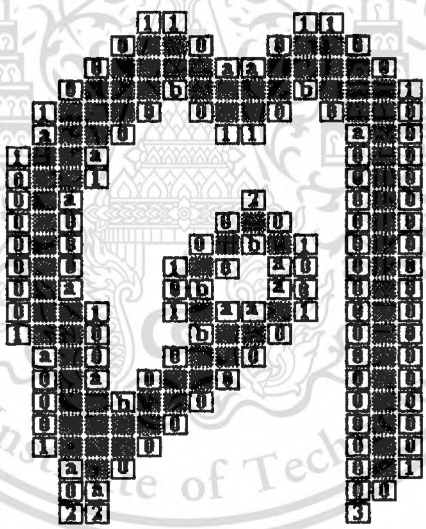


Figure 3.6 Component with distinguished outer and inner contours. The contour codes 0, 1, 2, 3, 4, a, and b are coded by the definition of contour invariant

3.3 Preprocessing

3.3.1 Isolated points removal

Isolated points (Figure 3.7) in a binary image are probably caused by noise, so any 3x3 mask to apply to any image is one that would remove such points. A mask is passed over the image to decide whether pixels should be removed or added. A black pixel is removed when the sum of all eight neighbors is 0. Similarly a white pixel is removed (that is a black pixel is added) when it matches with a mask shown in Figure 3.7 (b). Note that X pixels are black points and 0 pixels are white points.

0	0	0
0	X	0
0	0	0

(a) Isolated black pixel

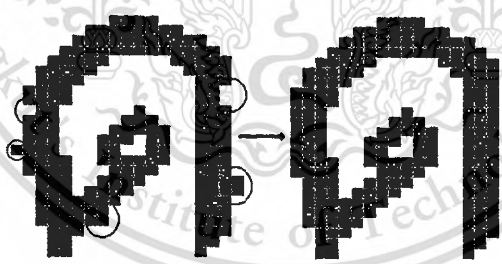
?	X	?
X	0	X
?	X	?

(b) Isolated white pixel

Figure 3.7 A 3x3 mask for the removal of isolated pixel. In the mask, the letter '0' stands for a white or background pixel, 'X' stands for a black or foreground pixel and '?' can take either value black or white

3.3.2 Local smoothing of character image borders

In some cases, we might want to remove or add one or two pixels to the character image borders. The algorithm that smoothes the border of characters performs two operations: removing and filling. Removing is performed when the border is locally convex. It clears some border pixels in order to remove the local convexity. Filling copes with local convexity. In Figure 3.8 (a), we wish to remove and add the consecutive pixels



(a) (b)

Figure 3.8 (a) Original image (b) Smoothed image

that are circled. The smoothing operations on the border of a character image are performed by applying the contour-following (Richard and Peter, 1973) and the contour invariant of character images traces the external boundary points. We use the invariant codes of the image to detect local curvature noise: such as convex (01b10, and 01aa10 codes) and concave (0a2a0, 0a11a0 codes) in the character image borders. Local curvature noise is removed or added when it matches one of the situations described in Figure 3.9. Thus, we define a simple smoothing operation, to remove or add one or two pixels to external boundary contours of character image, using the contour invariant codes.

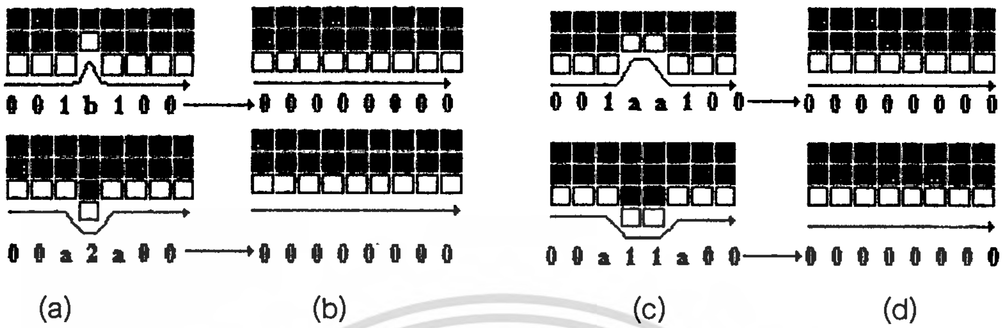


Figure 3.9 Contour: (a) and (c) original, (b) and (d) smoothed

3.3.3 Stroke thickness normalization

Stroke thickness normalization is computed as defined by Schurmann (1992). First, it is assumed that each character is written with a fixed stroke width, or the stroke width is the same throughout the entire image. Second, we assume that each character shape can be modeled as a line with length l and width w. Then the number of black pixels P is equal to $l \times w$ and the circumference C of the shape is equal to $2\ (l + w)$. From the preprocessing stage, the values of P and C are already known for all connected components. Solving these equations, the stroke width, w, of the character model can be readily determined.

3.3.4 Segmentation of Thai characters from the sentence

The segmentation of Thai characters (Airphaiboon et. al., 1996) from the sentence must be done before the decision of the unknown pattern process, because the construction of the Thai word, as shown in Figure 3.10 (a), is different from an English word: it is more complex. But the Thai word is noncursive and has no connected part between two characters in a word. Then, it is easy to produce segments from each character from a word. The procedure of segmentation is applied to extract each character (CS, UV, MV, LV or TS) from a given word. The algorithm is based on histogram analysis and contour tracing. An example of this algorithm is shown in Figure 3.10.

Figure 3.10 (a) shows a complicated Thai word, which consists of three CSs ("ว", "ม", "ร"), one UV ("ิ"), one MV ("ไ"), one LV ("ู"), and two TS ("่", "้"). Using the average value of the horizontal projection profile of this word, the component line level is obtained as shown in Figure 3.10 (b.). In this step, the upper part of the middle vowel "ไ" is deleted because it is located in the upper vowel line (the horizontal projection profile value is less than the average). To restore the deleted parts, a contour tracing algorithm is applied, starting from one pixel (Xs, Ys) and going to a pixel (Xe, Ye) of each deleted part (Figure3.10 (b)).

(a) (b) (c)

Figure 3.10 Single-character segmentation is base on histogram analysis and contour tracing

For each character in the consonant line level, UV, TS, and LV are searched, and their locations are determined in the upper vowel, tonal and lower vowel line levels, respectively (Figure 3.10 (c)).

In an orientation case, we cannot use a segmented character as mentioned above, because some characters may be digital images shifted or rotated from the horizontal or vertical line as shown in Figure 3.11. So, we must scan digital images in a left to right and top to bottom direction. If the pattern of each character is found, then we pull out one character from the other. To obtain a full scan we continue to separate characters until the end of the digital images. With each character from the orientation case, it is not necessary to normalize the rotation and scaling of the character, because the normalized ring projections and Fourier descriptors can solve these problems, as shown in the Chapter 6 and 7 respectively.

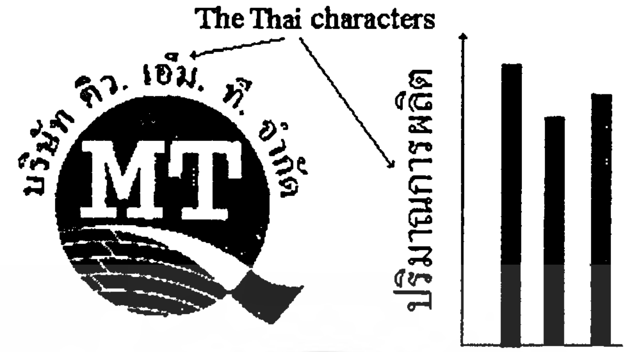


Figure 3.11 An example of the orientation characters

CHAPTER 4

NEURAL NETWORKS

4.1 Fundamentals of neural networks

In this chapter we briefly review the fundamentals of neural networks. There are many different types of neural networks, but they all have four basic attributes:

- A set of processing units,

- A connectivity option,

- A computing procedure, and

- A training procedure.

Let us now discuss each of these attributes.

4.1.1 Processing units

The units of a network are typically divided into three units as follows:

- Input units, which receive data from the information,

- Hidden units, which may internally transform the data representation, and

- Output units, which represent decisions.

4.1.2 Connectivity options

A network can be connected with any kind of connectivity (Nelson and Illingworth, 1990). Connectivity has to do with how the outputs are channeled to become inputs. The output signal from a node may be passed on as input to other processing elements, or even possibly sent back as an input to itself. Each kind of topology is best suited to a particular type of application. For example:

- When no processing element output can be an input for a node on the same layer or a preceding layer, the network is described as a feed-forward network.

- When outputs can be directed back as inputs to previous or some-level nodes, the network is a feedback network (figures 4.1 and 4.2). Feedback networks that have closed loops are recurrent systems.

4.1.3 Computation

The basic operation of an artificial neuron (Tebelskis, 1995) involves summing its weighted input signals and applying an activation function to generate the output. Let X = $(x_1, x_2, \ldots, x_n)^t$ represent inputs applied to the artificial neuron. w_{ji} is the weight from unit i to unit j.

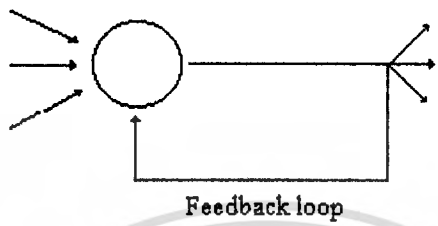


Figure 4.1 A single node with feedback to itself

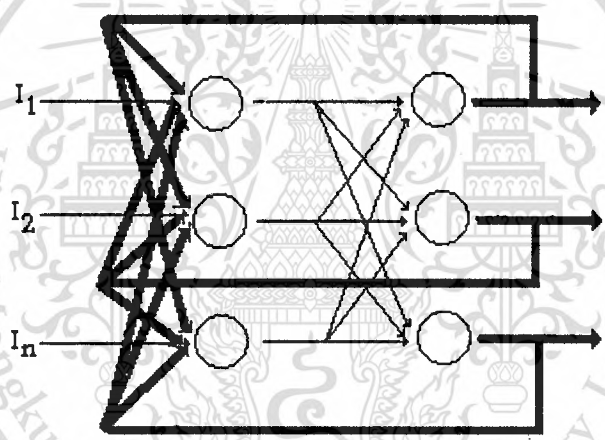


Figure 4.2 Network with feedback to previous nodes

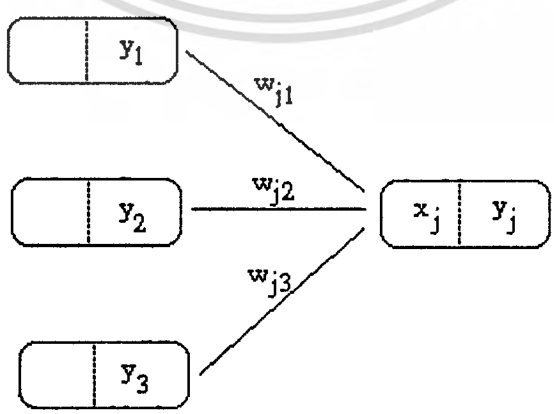


Figure 4.3 Computing unit activation, X = net input, Y= activation

The net input for unit j is given by

$$x_j = \sum_i y_i w_{ji} \tag{4.1}$$

where y_i is the output activation of incoming unit, and

w_{ji} is the weight from unit i to unit j

In general, the net input is offset by a variable bias term, θ, so that for example equation (4.1) is actually:

$$x_j = \sum_i y_i w_{ji} + \theta_j \tag{4.2}$$

The net input, x_j is further processed by an activation function $f(x)$ to produce the neuron output signal O(x). Often, activation values are restricted to the range [0,1]. The most commonly used activation function is the sigmoid function, which is given by

$$f(x) = 1/(1 + \exp(-x)) \tag{4.3}$$

The output of the neuron with the sigmoid activation function is then given by

$$O(x_j) = 1/(1 + \exp(-x_j)) \tag{4.4}$$

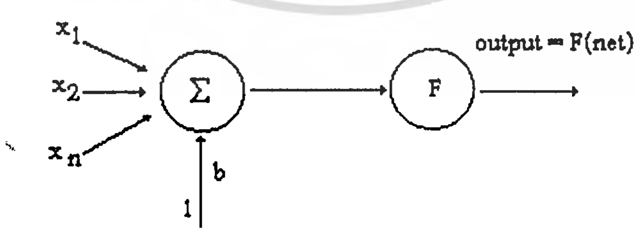


Figure 4.4 Artificial neuron with bias

A schematic diagram of an artificial neuron with multiple input and a bias is shown in Figure 4.4. Let X = (x_1, x_2, ..., x_n) represent the n input applied to the artificial neuron. The output of the neuron is then given by equation (4.4). The neural outputs pass through unidirectional synaptic connections. There are many types of activation functions. Figure 4.5 shows the various activation functions.

The most common function is the sigmoidal function, illustrated in Figure 4.5 c.) Sigmoidal functions have the advantages of nonlinearity, continuousness, and differentiability, enabling a multilayer network to compute any arbitrary real-valued function, while also supporting a practical training algorithm, back-propagation, based on gradient descent.

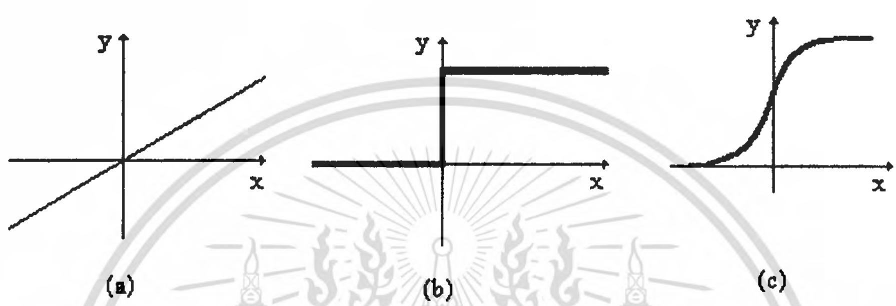


Figure 4.5 The activation functions: (a.) linear, (b.) Threshold, (c.) sigmoidal

4.1.4 Training

Training a network, in the most general sense, means adapting its connections so that the network exhibits the desired computational behavior for all input patterns. The process usually involves modifying the weights, but sometimes it also involves modifying the actual topology of the network, i.e., adding or deleting connections from the network.

Finding a set of weights that will enable a given network to compute a given function is usually a nontrivial procedure. An analytical solution exists only in the simplest case of pattern association, i.e., when the network is linear and the goal is to map a set of orthogonal input vectors to output vector. In this case, the weights are given by

$$w_{ji} = \sum_{p} \frac{y_i^p t_j^p}{\| y^p \|^2} \tag{4.5}$$

where y is the input vector,

t is the target vector, and

p is the pattern index.

In general, networks are nonlinear and multilayer, and their weights can be trained only by an iterative procedure, such as gradient descent on a global performance measure. This requires multiple passes of training on the entire training set (rather like a person learning a new skill). Each pass is called an iteration or an epoch. Moreover, since the accumulated knowledge is distributed over all of the weights, the weights must be modified very gently so as not to destroy all the previous learning. A small constant called the learning rate (ε) is thus used to control the magnitude of weight modifications. Most training procedures, including equation (4.5), are essentially variations of the Hebb rule, which reinforces the connection between two units if their output activation are correlated:

$$\Delta w_{ji} = \varepsilon y_i y_j \qquad (4.6)$$

One important variation of the above rule is a correlation between the first unit's activation y_i and the second unit's error relative to its target t_j:

$$\Delta w_{ji} = \varepsilon(t_j - y_j) \qquad (4.7)$$

To force each actual output y_j toward the correct output target t_j, we adjust the weights so as to minimize the total sum-squared error E between the targets t_j and the actual outputs y_j, over all Q examples. The total sum-squared error is defined via the Euclidean distance to be

$$E = \sum_{j=1}^{Q} | t_j - y_j |^2 \qquad (4.8)$$

There are three main classes of learning procedures:

- Supervised learning, in which a "teacher" provides output targets for each input pattern, and corrects the network's errors explicitly,

- Semi-supervised (or reinforcement) learning, in which a teacher merely indicates whether the network's response to a training pattern is "good" or "bad", and

- Unsupervised learning, in which there is no teacher, and the network must find regularities in the training data by itself.

4.2 Feed-forward neural network and back-propagation

The back-propagation network (Hagan, Dernuth and Beale, 1996) is probably the best known and widely used among the current types of neural network systems available. The back-propagation is a multi-layer feed forward netwŏrk with a different transfer function in the artificial neural network and a more powerful learning rule. The learning rule is known as back-propagation, which is a kind of gradient descent technique with backward error (gradient) propagation, as depicted in Figure 4.6.

A standard back propagation neural net is shown in Fig. 4.6. The first layer consists of N inputs, $X_i^P, 1 \leq i \leq N$. (A threshold is also required - see later - and it is convenient to incorporate it using a further node, x_0^P, which is always on. Thus the full range of i is $0 \leq i \leq N$). Each of the N input nodes is connected to each of the H nodes in the second or hidden layer. The H outputs of the hidden layer (Z_j^P, $1 \leq j \leq H$) are all connected to each of the M nodes in the output layer. The output of each output node is y_k^P, $1 \leq k \leq M$.

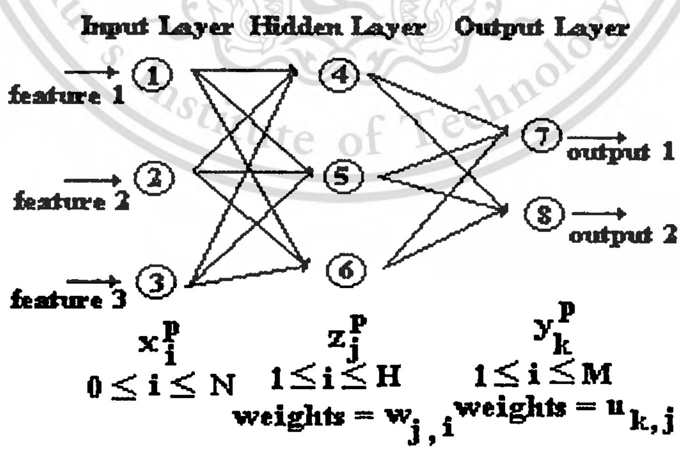


Figure 4.6 A 3:3:2 back propagation artificial neural network

In the case where the output is not modified by a squashing function, the output of the k^{th} output node is given by:

$$y_k^P = \sum_{j=1}^{H} U_{k,j}^P . Z_j^P \tag{4.9}$$

where $U_{k,j}^P$ is the weight given at the k^{th} output node to the input from the j^{th} hidden node, Z_j^P

A non-linear squashing function is used for the output of the hidden layer. Commonly this function is the sigmoid which varies smoothly from 0 to 1. The sigmoid requires a bias or threshold term which can be incorporated into the notation by having a weight corresponding to a zeroth input node that is always on, *ie* the bias for the j^{th} node is given by $w_{j,0}^P\, x_0^P$ with $x_0^P = 1$. The output of a node in the hidden layer is, therefore,

$$Z_j^P = f(\theta) = f\left(\sum_{i=0}^{N} w_{j,i}^P . x_i^P\right) \tag{4.10}$$

where often the activation function is the sigmoid, ie,

$$f(\theta) = 1 / (1 + e^{-\theta}) \tag{4.11}$$

The network is trained by presenting input values and comparing the output of the net with the desired one. A correction to the weights proportional to this error is back propagated. The method has been discovered and rediscovered several times.

Corrections, $\Delta U_{k,j}$ or $\Delta W_{j,i}$ are added to each weight. For the n^{th} time the network is presented with a training set, the $(n+1)^{th}$ values of the weights in the output layer are:

$$U_{k,J}(n+1) = U_{k,J}(n) + \Delta U_{k,J}(n) \tag{4.12}$$

where

$$\Delta U_{k,J}(n) = n . \delta_k^P . Z_j^P \tag{4.12}$$

and the error δ_k^P , is given by:

$$\delta_k^P = t_k^P - y_k^P \tag{4.13}$$

and E_P represents a scalar measure of the output error

$$E_p = (1/2) \sum_k (t_k^P - y_k^P)^2 \tag{4.14}$$

where t_k^P is the true or required result. For the hidden layer the correction is also proportional to the gradient of the sigmoid at that point. It can easily be verified by differentiating Eq. 4.10 that if the value of the sigmoid at point j is Z_j^P the slope at that point is $Z_j^P(1- Z_j^P)$. The new hidden layer weights are thus given by:

$$W_{j,i}(n+1) = W_{j,i}(n) + \Delta W_{j,i}(n) \tag{4.15}$$

where

$$\Delta W_{j,i}(n) = \eta \,.\, \delta_j^P(n) \,.\, x_i \tag{4.16}$$

and

$$\delta_j^P(n) = z_j^P(1-z_j) \sum_{k=1}^{M} \delta_k^P \,.\, u_{k,j}^P(n) \tag{4.17}$$

The constant η is variously called the learning rate or the step length.

Several iterations are performed to reduce error below tolerable value. So, we have the trained weights of network at the end of iteration. The training algorithm is as follows: (Fausett , 1994)

X Input training vector:

$$X = (x_1, \ldots, x_i, \ldots, x_n),$$

T Output target vector:

$$T = (t_1, \ldots, t_k, \ldots, t_m),$$

δ_k Portion of error correction weight adjustment for w_{jk} that is due to an error at output unit Y_k ; also, the information about the error at unit Y_k that is propagated back to the hidden units that feed into unit Y_k.

δ_j Portion of error correction weight adjustment for that is due to the back-propagation of error information from the output layer to the hidden unit

α Learning rate.

X_i Input unit i,

For an input unit, the input signal and output signal are the same, namely, x_i,

v_{oj} Bias on hidden unit j.

Z_j Hidden unit j:

The net input to Z_j is denoted z_in_j:

$$z\_in_j = v_{oj} + \sum_i x_i v_{ij}.$$

The output signal (activation) of Z_i is denoted z_j :

$$z_j = f(z\_in_j).$$

w_{ok} Bias on output unit k.

Y_k Output unit k:

Algorithm 4.1 The training algorithm for the back-propagation neural networks (Fausett , 1994)

The net input to Y_k is denoted y_in_k:

$$y\_in_k = w_{ok} + \sum_j z_j w_{jk}.$$

The output signal (activation) of Y_k is denoted y_k:

$$y_k = f(z\_in_k)$$

Step 0. Initialize weights.

(Set to small random values).

Step 1. While stopping condition is false, do Steps 2-9.

Step 2. For each training pair, do Steps 3-8.

Feedforward:

Step 3. Each input unit (X_i, i = 1, ..., n) receives input signal x_i and broadcasts this signal to all units in the layer above (the hidden units).

Step 4. Each hidden unit (Z_j,j=1,...,p) sums its weighted input signals,

$$z\_in_j = v_{oj} + \sum_{i=1}^{n} x_i v_{ij}.$$

applies its activation function to compute its output signal,

$$z_j = f(z\_in_j).$$

and sends this signal to all units in the layer above (output units).

Step 5. Each output unit (Y_k, k = 1, ..., m) sums its weighted input signals,

Algorithm 4.1 (Continued, Fausett, 1994)

$$y\_in_k = w_{ok} + \sum_{j=1}^{p} z_j w_{jk}.$$

and applies its activation function to compute its output signal,

$$y_k = f(z\_in_k)$$

Backpropagation of error:

Step 6. Each output unit (Y_k, k = 1, ..., m) receives a target pattern corresponding to the input training pattern, computes its error information tern,

$$\delta_k = (t_k - y_k)\ f'(y\_in_k).$$

calculates its weight correction term (used to update w_{ij} later)

$$\Delta w_{jk} = \alpha\ \delta_k\ z_j,$$

calculates its bias correction term (used to update w_{ok} later),

$$\Delta w_{ok} = \alpha\ \delta_k,$$

and send to units in the layer below.

Step 7. Each hidden unit (Z_j j = 1, ..., p) sums its delta inputs (from units in the layer above),

Algorithm 4.1 (Continued, Fausett, 1994)

$$\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk},$$

multiplies by the derivative of its activation function to calculate its error information term,

$$\delta_j = \delta\_in_j \, f'(z\_in_j)$$

calculates its weight correction term (used to update v_{ij} later),

$$\Delta v_{ij} = \alpha \, \delta_j \, x_i.$$

and calculates its bias correction term (used to update v_{oj} later),

$$\Delta v_{oj} = \alpha \, \delta_j.$$

Update weights and biases:

Step 8. Each output unit (Y_k, k = 1, ..., m) updates its bias and weights (j = 0, ..., p):

$$W_{jk}(new) = w_{jk}(ok') + \Delta w_{jk}.$$

Each hidden unit (Z_j, j = 1, ..., p) updates its bias and weights (i = 0, ..., n):

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij}.$$

Step 9. Test stopping condition.

Algorithm 4.1 (Continued, Fausett, 1994)

CHAPTER 5

ROUGH SET THEORY

This chapter introduces the rough set theory. The theory was introduced by Pawlak in the early 1980's, and based on this theory one can propose a formal framework for the automated transformation of data into knowledge. The knowledge product is used mainly for two purposes: as a model of some phenomena for human interpolation in order to gain insight about the phenomena, and to learn how to classify objects. The use of the end product may be different but the task is the same. The main concepts of the theory are presented, and illustrated with an example. The current chapter is not an attempt to present a complete framework for rough set analysis only a brief introduction is included. The focus of this thesis is on a prepositional rule based system in classification or recognition in the next chapter.

5.1 Information systems

An information system contains a data set represented in a table. Each row in the table represents a core, an event, a patient, or simply an object. Every column represents an attribute (a variable, an observation, a property, etc.) that can be measured for each object, the attribute may be also supplied by a human expert or user.

An information system, S, is defined as:

$$S = (U, A, \{ V_a : \in A \}) \tag{5.1}$$

where

U is a non-empty finite set of objects,

A is a finite, nonempty set of mappings,

$a : U \rightarrow V_a$, and

V_a is the value set of a.

Example 5.1: To illustrate the concept of information systems, we present an example as a Table 5.1. There are five animals or objects, and four attributes (Size, Type, Color, and Flying).

Table 5.1 An example information system

	Size	Type	Color	Flying
X1	small	bird	white	yes
X2	medium	bird	black	yes
X3	medium	cat	brown	no
X4	small	bird	white	yes
X5	big	horse	white	no

We interpret U as a set of objects (or situations) and A as a set of attribute mappings each of which assigns to an object a value which a may take under the respective attribute. This information is expressed by one distinguished attribute called decision attribute, the process is known as supervised learning. Information system of this kind is called decision systems. The information system in table 5.1 is also a decision system S = (U, $\{$Size, Type, Color$\}$ $\cup$ $\{$Flying$\}$) with decision attribute Flying.

5.2 Indiscernibility

One of the most important concepts of rough set theory is indiscernibility, which is used to define equivalence classes for the objects. Each subset of attributes B $\subseteq$ A defines an equivalence relation, denote IND(B) and will be called an indiscernibility relation. This indiscernibility relation is defined as:

$$\mathrm{IND}(B) = \{ (x,y) \in U \times U : f(x, a) = f(y, a) \text{ for every } a \in B \} \qquad (5.2)$$

In equation 5.2, the subset of attributes B will define a partitioning of the universe into sets such that each object in a set cannot be distinguished from other objects in the set using only the attribute in B. The sets of the condition as mentioned on above are called equivalence classes. The family of all equivalence classes of the equivalence relation IND(B) is denoted U/IND (B).

Example 5.2: In the information system shown in Table 5.1, we see that objects X_1 and X_4 are indiscernible. The indiscernibility relations from the table 5.1 are as follows.

IND(Size) $= \{ \{x_1, x_4\}, \{x_2, x_3\}, \{x_5\} \}$

IND(Type) $= \{ \{x_1, x_2, x_4\}, \{x_3\}, \{x_5\} \}$

IND(Color) $= \{ \{x_1, x_4, x_5\}, \{x_2\}, \{x_3\} \}$

IND(Flying) $= \{ \{x_1, x_2, x_4\}, \{x_3, x_5\}\}$

IND(Size, Type, Color, Flying) $= \{ \{x_1, x_4\}, \{x_2\}, \{x_3\}, \{x_5\} \}$

5.3 Decision systems

If we add the decision attributes into the information into the information system, this system is called a decision system. The decision system is defined (Bjanger, 1999)

$$S = (U, A \cup \{d\}) \qquad (5.3)$$

where $d \notin A$ is a distinguish attribute called the decision attribute,

The elements of A are called condition attributes, Since a decision system is a special kind of information system, the decision is not necessarily constant on the equivalence classes. That is, although the two objects belonging to the same equivalence class, the values of the decision attribute may be different. In this case, the decision system is inconsistent. The decision system is consistent if there exists any class $E \in A \cup \{d\}$ for which a unique classification can be made.

Example 5.3: The simple information system S = (U, A) shown in Table 5.1 is a result of measuring the three characteristics Size, Color, and Flying of different animals.

Now, let us extend the information system as shown in Table 5.1. The additional object of animal has been stored to the table, which is shown in Table 5.2.

Table 5.2 An example of the extended information system

	Size	Type	Color	Flying
X1	small	bird	white	Yes
X2	medium	bird	black	Yes
X3	medium	cat	brown	No
X4	small	bird	white	Yes
X5	big	horse	white	No
X6	medium	bird	black	No

From this example we can see some of the problems that rough set theory addresses. For instance, objects X_2 and X_6 belong to the same equivalence classes, but they have different values for the decision Flying attribute. This means that the information system in this example is inconsistent.

The objects are classified according to condition attributes C = $\{$Size, Type, Color$\}$, and the classification is represented as the decision attribute set D = $\{$Flying$\}$. By removing the inconsistent attributes and their values. We can search a set of equivalence relations from the decision table, and the objects are represented in terms of object classes, each of which is supported by a number of examples, given in the rightmost column of the table. Thus, there are three object classes such as x_1 and x_4, x_3 and x_5 classes respectively.

5.4 Set approximation

Approximation spaces are the core mathematical concept of rough set data analysis, and their usage reflects the idea the granulation of information can be described by classes of and indiscernibility relation. Given an information system, S = (U, A), and subset of attributes, B $\subseteq$ A, we would like to approximate a set of objects, X, using only the information contained in B.

For X $\subseteq$ U, and B $\subseteq$ A we associate two sets defined as follows (Lken, 1999):

$$B.X = \{ x \mid [x]_B \subseteq X \} \tag{5.4}$$

is the lower approximation or positive region of X, and

$$B^{*}X = \{\, x \mid [x]_B \cap X \neq \varnothing \,\} \tag{5.5}$$

is the upper approximation or possible region of X.

The lower approximation is the set containing all objects for which the equivalence class corresponding to the object, which is a subset of the set we would like to approximate. This set contains all objects, which with certainty belong to the set X.

The upper approximation is the set containing the objects for which the intersection of the object's equivalence class, and this set we would like to approximate is not the empty set. This set contains all objects, which possibly belong to the set X.

A set $BN_B(X) = B^{*}X - B_{*}X$ will be called B-boundary of X.

If $X \subseteq U$ is given by a predicate P and $x \in U$, then

1.) $x \in B_{*}X$ means that x certainly has property P,

2.) $x \in B^{*}X$ means that x possible has property P, and

3.) $x \in U \setminus B^{*}X$ means that x definitely does not have property P.

The area of uncertainly extends over $B^{*}X \setminus B_{*}X$, and the area of certainly is $B_{*}X \cup -B^{*}X$, respectively.

The concepts of lower approximation, upper approximation and B-boundary region are illustrated in Figure 5.1.

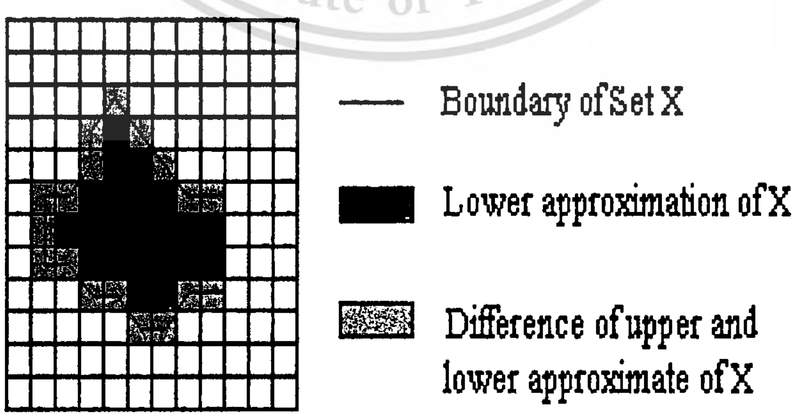


Figure 5.1 Rough set approximation

Example 5.4: Approximating the set of animal sizes in Table 5.1 that can fly is shown in Figure 5.2

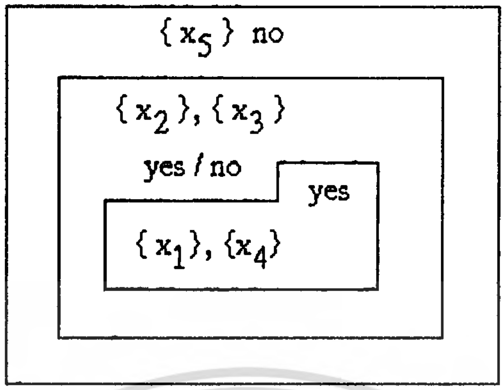


Figure 5.2 Approximation of the set yes

5.5 Core and reduction of attributes

The set of attributes in the information system describes objects in the system. We use the set of attributes to classify the objects into classes that are determined by anywhere from one to several attributes. Sometimes it happens that the set of attributes provides more information about objects then is needed to distinguish them. In such a situation some of the attributes may be reduced without losing the ability to group the objects. In the rough set approach, there are two fundamental concepts used in the reduction of knowledge, a reduct and the core. Intuitively, a reduct is one essential part of an information system, which can discern all objects that are discernible by the original information system. The core comprises the attributes that are indispensable to the discrimination of objects, the attributes that are contained in all the reducts. These concepts are formally defined in Definition 5.5.1 and 5.5.2 (Lken, 1999), but first it is necessary to introduce some auxiliary notions.

Definition 5.5.1: (dispensability) Given S = (U, A), an attribute a is said to be dispensable in $B \subseteq A$ if IND(B) = IND(B - $\{a\}$), otherwise the attribute is indispensable in B. If all attributes $a \in B$ are indispensable in B, then B is called orthogonal or independent.

Definition 5.5.2: (Reducts) Given an information system S = (U, A), let $B \subseteq A$. A reduct of B is a set of attributes $B' \subseteq B$ such that all attributes $A \in B - B'$ are dispensable (B' is orthogonal), and IND(B') = IND(B). The set of reducts of B is denoted RED(B).

$$CORE(B) = \bigcap_{RED \in RED(B)} RED \tag{5.6}$$

From these definition, we can say that attributes $a \in B$ is superfluous in B, if IND (B-$\{a\}$) = IND(B), otherwise it is indispensable.

If all attributes $a \in B$ are indispensable in B. So the set of all indispensable attributes in B will be called the core of B, and will denote CORE(B). The core is a set of attributes such that the removal of a single attribute impacts on the original ability to distinguish the objects. The core of attributes is the interaction of all reductions.

Example 5.5: To discern between the different equivalence classes in the example in Table 5.1, only the attributes Size, Type, and Color are necessary. So $\{$Size, Type, Color$\}$ is an example of a reduct:

$$IND_A(\{Size, Type, Color\}) = IND_A(A) \tag{5.7}$$

Reducts can be computed on the basis of discernibility matrices and discernibility functions.

5.5.1 Discernibility matrices

A discernibility matrix of S is a symmetric n x n matrix with entries (Bjanger, 1999)

$$C_{ij} = a \in A \mid a(x_i) \neq a(x_j) \quad \text{for } i, j = 1, 2, \ldots, n \tag{5.8}$$

The entries for each object are thus the attributes that are needed in order to discern object i from object j.

Example 5.6: The discernibility matrix for our example in table 5.1 is shown in Table 5.3. For readability, the three attributes are abbreviated s, t, c respectively.

Table 5.3 Discernibility matrix

	x_1	x_2	x_3	x_4	x_5
x_1	$\varnothing$	s, c	s, t, c	$\varnothing$	s, t
x_2		$\varnothing$	t, c	s, c	s, t, c
x_3			$\varnothing$	s, t, c	s, t, c
x_4				$\varnothing$	s,t
x_5					$\varnothing$

In a discernibility matrix the diagonal elements are naturally $\varnothing$ and $C_{ij} = C_{ji}$. Therefore, the lower triangular part in Table 5.3 is omitted. In the matrix, $C_{14} = \varnothing$ because the values of decision attributes for object s, t, and c are the same. C_{12} = {s, c}, because only the values of condition attributes s and c are different for the case of examples x_1 and x_4.

5.5.2 Discernibility functions

A discernibility function (Bjanger, 1999) is a Boolean product-of-sum function that expresses how an object or a set of objects can be discerned from a certain subset of the full universe of objects.

From the discernibility matrix, we can build a discernibility function. A discernibility function f_A for an information system S is a Boolean function of m Boolean variables a_1^*, ..., a_m^* (corresponding) to the attributes a_1, ... a_m) defined as below.

$$f_A(a_1^*, \ldots, a_m^*) = \wedge\{\vee C^*_{ij} \mid 1 \leq j \leq i \leq n,\ C_{ij} \neq \varnothing\} \qquad (5.9)$$

where $C^*_{ij} = \{a^* \mid a \in C_{ij}\}$

The discernibility function is a conjunction of all the entries in the discernibility matrix that are not the empty set. The conjunction may, if possible, be simplified. The results of simplification are the possible reducts for the information system.

It is also possible to generate a discernibility function from the discernibility matrix for one of the objects in the information system. This is done by looking at only one row (or column) in the discernibility matrix, and forming a conjunction of all the entries in this row (or column). When we simplify this conjunction, we get possible reducts for the particular object in question.

5.6 Decision rules

For decision systems, S = (U, A $\cup$ {d}), we would like to find an approximation of the decision, d. This can be done by constructing the decision-relative discernibility matrix of S. This matrix tell us how to discern an object from objects belonging to another decision class. The process of computing this matrix is called computing the discernibility matrix modulo the decision attribute.

We can obtain the decision rules from an information system or a reduction of condition attributes. Some condition values may be unnecessary in the decision rules, so we can remove them from the rules or change them into "don't care" values. The process of conditional value removed is called values reduction.

Let S = (U, A, C, D) be an information system with n objects, where U is a finite set of objects. A is a set of attributes further classified into disjointed condition attributes C and decision attributes D, A = C $\cup$ D, V = $\cup$ V_a, V_a is the domain of attribute a $\in$ A. Hence, decision rules are often presented as implications and are often called "If ... then ..." rules. If a, b, c, d are the condition attributes of set A, D is decision attribute. We can express the rules as

Rule I

IF a = 1 THEN D = 1 (for the class I)

Rule II

IF b = S and c = M THEN D = 2 (for the class II)

....

Rule n

IF c = L THEN D = n (for the class n)

.

5.7 Rough membership function

In classical set theory, either an element belongs to a set or it does not. The corresponding membership function is the characteristic function for the set, i.e. the function takes values 0 and 1, respectively, In the case of rough sets, the notion of membership is different. The rough membership function quantifies. The degree of relation overlap between the set X and the equivalence $[X_B]$ class to which x belongs. It is defined as follows (Komorowski et.al.):

$$\mu^{S}_{x} : U \rightarrow [0, 1] \text{ and } \mu^{S}_{x}(x) = \frac{[x]_B \cap X}{|[x]_B|} \qquad (5.10)$$

where S = (U, B) is information system and $\emptyset \neq X \subseteq U$.

CHAPTER 6

ROUGH SET METHOD FOR INVARIANT THAI PRINTED CHARACTERS RECOGNITION

We attempt to apply the rough set theory to the recognition of Thai characters. This proposed idea is a new development in invariant Thai character recognition. There had been an attempt to apply this theory into the pattern recognition, but there are only a few implementations such as numeral recognition (Nejman, 1995), handwritten numeral classification (Kim, 2000), printed Thai character (Kasemsiri et. al., 2000, 2001; Mitatha, 2001) and invariant Thai character recognition (Phokharatkul et. al., 2001, 2002). Furthermore, there has been development in rough sets with the other fields such as data mining (Lin and Cercone, 1997), image processing (Lau, 1993), and power system (Pal and Skowron, 1999; Lambert-Torres et. al., 1996). The detail of the application of rough set theory for invariant printed Thai characters will be described in the following sections.

6.1 System overview

We propose a two-stage pattern classification method based on the ring projections. The ring projection values are computed from the set of invariant Thai characters. Figure 6.1 shows the procedure of our proposed two-stage classification method.

In our experiment, we use the characters of four fonts (four sizes in each font) of 86 Thai characters as a training set. All are printed with HP Laser Jet 6 P and scanned with HP Scan Jet 6100C in black and white mode. Figure 6.2 shows the example of training set.

6.2 Feature extraction

We use an invariant projection method to extract features from the Thai Characters. Ring projection (Tang, 1996, 1998) is defined as the total number of foreground pixels as

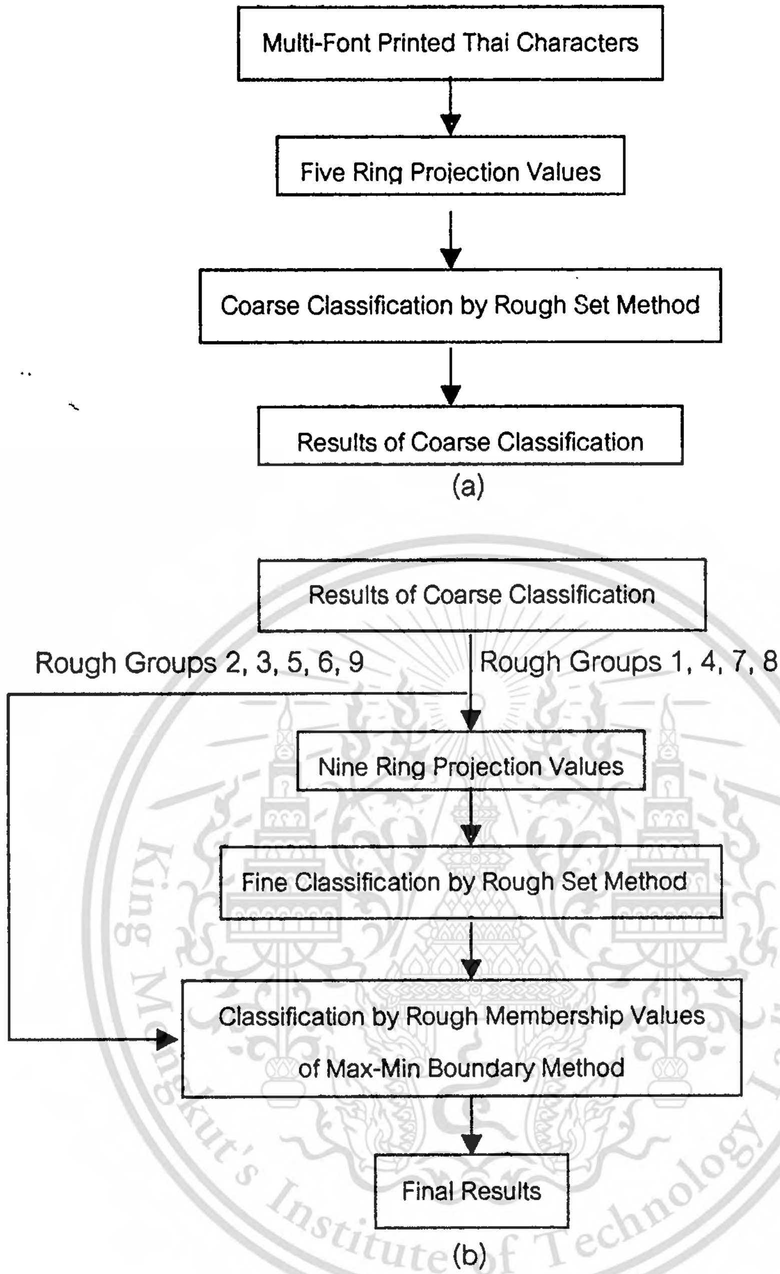


Figure 6.1 a.) The coarse classification stage b.) The fine classification stage.

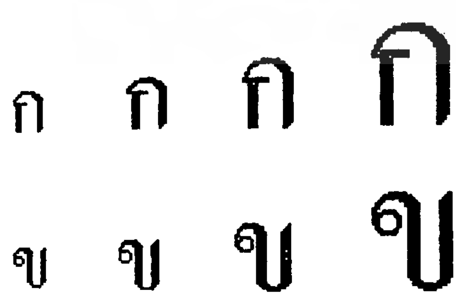


Figure 6.2 Example of training data.

Figure 6.3. We can derive the centroid of the character, as denoted by C (x_o, y_o), and subsequently translate the origin of our reference frame to this centroid. Let

$$R_{max} = \max \left| N(x,y) - C(x_o, y_o) \right| \tag{6.1}$$

where $\left| N(x, y) - C(x_o, y_o) \right|$ represents the Euclidean distance between two points N and C on the plane. We can transform the original Cartesian reference form into a polar form as the following relationship.

$$\left. \begin{aligned} X &= r \cos \theta \\ Y &= r \sin \theta \end{aligned} \right\} \tag{6.2}$$

Then

$$D(x, y) = D(r \cos \theta, r \sin \theta)$$

where $r \in [0, \infty]$, $\theta \in [0, 2\pi]$. For any fixed $r \in [0, R_{max}]$ we can compute the following integral:

$$f(x, y) = \int_0^{2\pi} D(r \cos \theta, r \sin \theta)\, d\theta \tag{6.3}$$

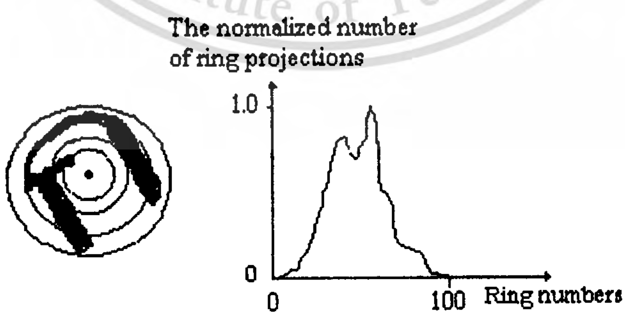


Figure 6.3 An illustration of the ring projection for the Thai character "ท" rotated 30°

The resulting f (x, y) is equal to the total mass as distribution along circular rings, as shown in Figure 6.3. From the concept of ring projection as mention above, the ring projection is calculated by

$$\text{ring projection (l)} = f(0,0) + \sum_{l=1}(x,y),$$

$$\text{ring projection (l)} = \sum_{l=r}(x,y), \quad \text{if } 2 \leq 1 \leq L \tag{6.4}$$

$$\text{ring projection (L)} = \sum_{L=r}(x,y),$$

where r = int ($\sqrt{x^2 + y^2}$) , L = R_{max} is the largest radius, l = 1,2,3,...,L is the ring number, and int ( ) means taking the integer part from a real number. We use these values to set up the object attributes as the sets, and to obtain the decision rules for classification by the rough set as the detail in classification using rough sets.

6.3 Description of the problems

The idea is to transform a set of examples into a set of rules that represent the recognition of a recognizer system. Consider a ring projection database composed by a set of projection values, such as shown in Table 6.1. The classification of Thai Character depends on the ring projection values. Table 6.2 contains the attributes represented by the set a_1, a_2,..., a_5 and the corresponding decision D, where:

- the normalized ring projection values 0.-0.3 are defined by 1;
- the normalized ring projection values 0.31-0.6 are defined by 2; and
- the normalized ring projection values 0.61-1.0 are defined by 3.

The reason behind selection five rings in the coarse classification. The distribution of elements in the coarse groups are not different, when 5, 6, ..., 9 rings are used. For this reason, the five rings are selected..

Table 6.1 Ring projection data of printed Thai characters

Characters	a_1	a_2	a_3	a_4	a_5
ก	0.0388	0.5994	1.0000	0.4982	0.0632
ก	0.0469	0.4982	1.0000	0.3611	0.0586
ก	0.0271	0.6135	1.0000	0.2902	0.0707
...	...	...	...	...	...
ฤ	0.4413	0.8333	1.0000	0.8000	0.0944
ฤ	0.3302	0.6289	0.9562	1.0000	0.1766
ฤ	0.4226	0.7522	1.0000	0.7586	0.0770
...	...	...	...	...	...
ฮ	0.4671	0.9165	1.0000	0.9789	0.0995
ฮ	0.2258	0.6948	1.0000	0.9906	0.0808
ฮ	0.2583	0.7471	1.0000	0.9196	0.0967

Table 6.2 Example of database with range values

No.	Characters	a_1	a_2	a_3	a_4	a_5	Dec.
1	ก	1	2	3	2	1	D9
2	ก	1	2	3	2	1	D9
3	ก	1	3	3	1	1	D3
...	...	...	...	...	...	...	...
17	ฤ	2	3	3	3	1	D1
18	ฤ	2	3	3	3	1	D1
19	ฤ	2	3	3	3	1	D1
...	...	...	...	...	...	...	...
1374	9	1	2	3	3	1	D7
1375	9	1	2	3	3	1	D7
1376	9	1	2	3	3	1	D7

Note: The example number (No.) is for indexing purpose only.

The decision attributes of each character in Table 6.2 is made according to a system, and nine possible outputs can be selected for the examples as follows:

a_2	a_3	a_4	Decision
1	1	1	D1
2	2	2	D1
3	3	3	D1
1	1	2	D2
1	1	3	D2
2	2	3	D2
3	3	2	D3
3	3	1	D3
2	2	1	D3
3	2	2	D4
3	1	1	D4
2	1	1	D4
3	1	3	D5
3	2	3	D5
2	1	2	D5
3	1	2	D5
2	1	3	D5
3	2	1	D6
1	2	2	D7
1	3	3	D7
2	3	3	D7

a_i	a_j	a_k	Decision
1	2	3	D8
1	2	1	D9
1	3	1	D9
2	3	2	D9
2	3	1	D9
1	3	2	D9

6.4. Presentation of the classification algorithms

Let R be a family of equivalence relations. The reduct of R, (RED(R)), and the core of R, (core(R)), are defined as follows:

Reduct - A reduct is a reduction of an information system, which results in no loss of information (classification ability) by removing, attributes (a_i). There may be one or many for a given information system.

Core - A core is the set of attributes (a_i) which are common to all reducts.

The methodology to obtain the reduction of condition can be summarized as follows: (Pal and Skowron, 1999; Lambert-Torres et. Al., 1996; Attoh-Okine, 1997; Sienkiewiez, 1995)

Step 1: Transform continuous values in ranges.

Step 2: Eliminate identical attributes.

Step 3: Eliminate identical examples.

Step 4: Eliminate dispensable attributes.

Step 5: Compute a core of the decision table.

Step 6: Compute a table of the reduct set.

Step 7: merge possible examples and compose the final set of rules.

6.5. Description of the knowledge base reduction

6.5.1. Classification

The set of attributes in an information system describes objects of interest. If it happens that the set of attributes provides more information about objects then we need to distinguish them. In such a situation some of the attributes may be reduced without loosing the ability to classify the objects. The procedure of applying decision table techniques can be described by the following steps:

Step 1:

Set up model or determine the format of decision table, which includes conditional attributes (a_i), the number of attribute levels in each condition (1, 2, 3), and decision attributes (D). In the recognition system we set up the object's attributes from the ring projection vectors as shown in Table 6.2.

Step2 and 3:

The next step of the algorithm is to verify if any attribute can be eliminated by repetition. In Table 6.2, it can be verified that the repetition does not occur. In addition, many examples are identical. The similar examples are also merged, and a resultant set of examples as shown in Table 6.3.

Table 6.3 Resultant set of examples

No.	a_1	a_2	a_3	a_4	a_5	D
1	1	2	3	2	1	D9
2	1	3	3	3	1	D1
3	1	3	3	1	1	D3
4	1	2	3	1	1	D9
5	2	3	3	3	1	D1
6	2	2	3	3	1	D7
7	1	3	3	2	1	D3
8	2	2	3	1	1	D9

Table 6.3 (Continued)

No.	a_1	a_2	a_3	a_4	a_5	D
9	2	2	3	2	1	D9
10	2	1	3	3	1	D7
11	3	3	3	2	1	D3
12	2	3	3	2	1	D3
13	3	3	3	1	1	D3
14	1	1	3	3	1	D7
15	1	3	3	3	2	D1
16	2	3	3	1	1	D3
17	3	3	3	3	1	D1
18	2	1	3	2	1	D9
19	1	3	2	3	1	D5
20	3	3	2	1	1	D6
21	1	3	2	1	1	D6
22	2	3	2	1	1	D6
23	1	2	3	3	2	D7
24	3	2	3	3	1	D7
25	3	2	3	2	1	D9
26	3	2	2	3	1	D2
27	2	2	2	3	1	D2
28	3	2	3	1	1	D9
29	2	3	3	3	2	D1
30	2	3	1	1	1	D4
31	1	1	2	3	2	D8
32	1	1	1	3	1	D2
33	1	1	2	3	1	D8
34	1	2	2	3	1	D2
35	1	2	2	3	2	D2
36	1	1	3	3	2	D7

Table 6.3 (Continued)

No.	a_1	a_2	a_3	a_4	a_5	D
37	3	3	3	3	2	D1
38	2	1	2	3	1	D8
39	2	·1	2	3	2	D8
40	2	2	3	3	2	D7
41	1	2	3	3	1	D7

Step 4:

The next step verifies if the decision table contains only indispensable attributes. This task can be accomplished by eliminating step-by-step each attribute and verifying if the table gives the correct classification. For example, if the attributes a_1 and a_5 are eliminated, the table continues to give a correct classification as shown in Table 6.4. Then, we say that a_1 and a_5 are dispensable attributes for this decision table.

Table 6.4 Set of examples with indispensable attributes

No.	a_2	a_3	a_4	D
1	2	3	2	D9
2	3	3	3	D1
3	3	3	1	D3
4	2	3	1	D9
5	3	3	3	D1
6	2	3	3	D7
7	3	3	2	D3
8	2	3	1	D9
9	2	3	2	D9
10	1	3	3	D7
11	3	3	2	D3
12	3	3	2	D3
13	3	3	1	D3

Table 6.4 (Continued)

No.	a_2	a_3	a_4	D
14	1	3	3	D7
15	3	3	3	D1
16	3	3	1	D3
17	3	3	3	D1
18	1	3	2	D9
19	3	2	3	D5
20	3	2	1	D6
21	3	2	1	D6
22	3	2	1	D6
23	2	3	3	D7
24	2	3	3	D7
25	2	3	2	D9
26	2	2	3	D2
27	2	2	3	D2
28	2	3	1	D9
29	3	3	3	D1
30	3	1	1	D4
31	1	2	3	D8
32	1	1	3	D2
33	1	2	3	D8
34	2	2	3	D2
35	2	2	3	D2
36	1	3	3	D7
37	3	3	3	D1
38	1	2	3	D8
39	1	2	3	D8
40	2	3	3	D7
41	2	3	3	D7

However, when the attribute a_4 is eliminated (Table 6.5), some examples such as the examples 6, 8, and 9 have the same set of attributes but they give different classification. In this case, the attribute a_4 is indispensable for the decision table.

Table 6.5 Verification of the indispensability of attribute a_4

No.	a_1	a_2	a_3	a_5	D
1	1	2	3	1	D9
2	1	3	3	1	D1
3	1	3	3	1	D3
4	1	2	3	1	D9
5	2	3	3	1	D1
6	2	2	3	1	D7
7	1	3	3	1	D3
8	2	2	3	1	D9
9	2	2	3	1	D9
10	2	1	3	1	D7
11	3	3	3	1	D3
12	2	3	3	1	D3
13	3	3	3	1	D3
14	1	1	3	1	D7
15	1	3	3	2	D1
16	2	3	3	1	D3
17	3	3	3	1	D1
18	2	1	3	1	D9
19	1	3	2	1	D5
20	3	3	2	1	D6
21	1	3	2	1	D6
22	2	3	2	1	D6
23	1	2	3	2	D7
24	3	2	3	1	D7

Table 6.5 (Continued)

No.	a_1	a_2	a_3	a_5	D
25	3	2	3	1	D9
26	3	2	2	1	D2
27	2	2	2	1	D2
28	3	2	3	1	D9
29	2	3	3	2	D1
30	2	3	1	1	D4
31	1	1	2	2	D8
32	1	1	1	1	D2
33	1	1	2	1	D8
34	1	2	2	1	D2
35	1	2	2	2	D2
36	1	1	3	2	D7
37	3	3	3	2	D1
38	2	1	2	1	D8
39	2	1	2	2	D8
40	2	2	3	2	D7
41	1	2	3	1	D7

Table 6.4 can be reduced again, because the some examples are identical. The Table 6.6 shows the new set of examples.

Step 5 and 6:

From Table 6.6, the core set of examples is computed. We eliminate each attribute step-by-step and verify if the decision table continues to give the correct answer (i.e., it continues to be consistent).

Table 6.6 New sets of examples with Indispensable attributes

No.	Attributes			D
	a_2	a_3	a_4	(decision)
1	2	3	2	D9
2	3	3	3	D1
3	3	3	1	D3
4	2	3	1	D9
5	3	3	2	D3
6	1	3	3	D7
7	1	3	2	D9
8	3	2	3	D5
9	3	2	1	D6
10	2	2	3	D2
11	3	1	1	D4
12	1	2	3	D8
13	1	1	3	D2
14	2	3	3	D7

The following Tables 6.7-6.10 can represent the core computation:

- The first step, when the attribute a_2 in Table 6.7 is eliminated, the decision table becomes inconsistent because the examples 1, 5, and 7 have the same attributes and different decision. The other examples such as 2, 6, and 14; 3, and 4; 8, 10, and 12 have the same situation. In this case, the examples 1-8, 10, 12, and 14 are ambiguous upon removal of attribute a_2. Therefore, the attribute a_2 is a part of core. The attribute a_2 is called indispensable for the decision table.

- The next step, the attribute a_3 is eliminated is shown in Table 6.8. After considering each example, the examples 2-3, 6, 8-12, and 14 are ambiguous upon removal of attribute a_3. Therefore, attribute a_3 is a part of core.

- The last step, the attribute a_4 is eliminated as shown in Table 6.9. After considering each example, the examples 1-9, and 14 are ambiguous upon removed of attribute a_4. Therefore, the attribute a_4 is a part of core.

Table 6.7 To compute core (a_2 is eliminated)

No.	Attributes			D
	a_2	a_3	a_4	(decision)
1	2	3	2	D9
2	3	3	3	D1
3	3	3	1	D3
4	2	3	1	D9
5	3	3	2	D3
6	1	3	3	D7
7	1	3	2	D9
8	3	2	3	D5
9	3	2	1	D6
10	2	2	3	D2
11	3	1	1	D4
12	1	2	3	D8
13	1	1	3	D2
14	2	3	3	D7

Table 6.8 To compute core (a_4 is eliminated)

No.	Attributes			D
	a_2	A_3	a_4	(decision)
1	2	3	2	D9
2	3	3	3	D1
3	3	3	1	D3
4	2	3	1	D9
5	3	3	2	D3
6	1	3	3	D7
7	1	3	2	D9
8	3	2	3	D5
9	3	2	1	D6

Table 6.8 (Continued)

No.	Attributes			D
	a_2	A_3	a_4	(decision)
10	2	2	3	D2
11	3	1	1	D4
12	1	2	3	D8
13	1	1	3	D2
14	2	3	3	D7

Table 6.9 To compute core (a_3 is eliminated)

No.	Attributes			D
	a_2	a_3	a_4	(decision)
1	2	3	2	D9
2	3	3	3	D1
3	3	3	1	D3
4	2	3	1	D9
5	3	3	2	D3
6	1	3	3	D7
7	1	3	2	D9
8	3	2	3	D5
9	3	2	1	D6
10	2	2	3	D2
11	3	1	1	D4
12	1	2	3	D8
13	1	1	3	D2
14	2	3	3	D7

From the computation of core, the core set of decision tables and the reduct of each example are summarized in Table 6.10 and 6.11 respectively.

Table 6.10 Core of the set of examples

No.	Attributes			D
	a_2	a_3	a_4	(decision)
1	2	-	2	D9
2	3	3	3	D1
3	3	3	1	D3
4	2	-	1	D9
5	3	-	2	D3
6	1	3	3	D7
7	1	-	2	D9
8	3	2	3	D5
9	-	2	1	D6
10	2	2	-	D2
11	-	1	-	D4
12	1	2	-	D8
13	-	1	-	D2
14	2	3	3	D7

Table 6.11 Reduced Rules

No.	Attributes			D
	a_2	a_3	a_4	(decision)
1	2	x	2	D9
2	3	3	3	D1
3	3	3	1	D3
4	2	x	1	D9
5	3	x	2	D3
6	1	3	3	D7
7	1	x	2	D9
8	3	2	3	D5
9	x	2	1	D6

Table 6.11 (Continued)

No.	Attributes			D (decision)
	a_2	a_3	a_4	
10	2	2	x	D2
11	3	1	x	D4
11'	x	1	1	D4
12	1	2	x	D8
13	1	1	x	D2
13'	x	1	3	D2
14	2	3	3	D7

Step 7:

According the Table 6.11, the knowledge existent in the Table 6.2 is expressed by the following set of rules:

Rule 1: If a_2 is 2 and a_4 is 2 then S is D9.

Rule 2: If a_2 is 3 and a_3 is 3 and a_4 is 3 then S is D1.

....

Rule 14: If a_2 is 2 and a_3 is 3 and a_4 is 3 then S is D7.

The set of rules as mentioned above can be put into natural language:

Rule 1: If (the normalized ring projection value a_2 is between 0.31-0.6) and (the normalized ring projection value a_4 is between 0.31-0.6) then the classification of character is group 9.

Rule 2: If (the normalized ring projection value a_2 is between 0.61-1.0) and (the normalized ring projection value a_3 is between 0.61-1.0) and (the normalized ring projection value a_4 is between 0.61-1.0) then the classification of the character is group 1.

....

Rule 14: If (the normalized ring projection value a_2 is between 0.31-0.6) and (the normalized ring projection value a_3 is between 0.61-1.0) and (the normalized ring projection value a_4 is between 0.61-1.0) then the classification of the character is group 7.

This algorithm has been implemented in Matlab 5.3 under Windows 98. The algorithm has also been applied to the character date set. The result is that all testing data are correctly classified as shown in Table 6.12.

Table 6.12 Classification results

Groups	number	Alphabets
1	438	ก ข ฆ ง จ ฉ ช ซ ฎ ฏ ฐ ฑ ฒ ถ ท ธ น บ ผ พ ภ ม ย ร ล ว ษ ส ห ฬ ฮ อั า อำ อิ อู เ ใ ไ ๅ อ่ อี๊ อี๋ อ์ ๓ ๗ 1 5 8 9
2	23	โ ๐ ๑ ๓ 1 3 5 8
3	459	ก ศ จ ฐ ฑ ท ป ผ ฟ ศ ษ ส ห ฬ ฮ อั า อิ อี อึ อื อุ อู เ ใ ไ อ็ ฤ ๅ อ้ อี๊ อี๋ ๆ ฯ อ์ ๒ ๔ ๕ ๖ ๗ ๘ ๙ 1 4 6 8 9
4	2	ๆ
5	3	ธ
6	30	ฝ ฤ ๆ อ์ ๖ 7
7	308	ก ข ค ฆ ฌ ญ ฑ ฒ ณ ด ต ถ บ ผ ภ ม ย ร ล ว ษ อ อำ อู โ ใ ไ ๐ ๑ ๓ ๗ 0 2 3 6 8 9
8	35	๐ ๑ 2 3
9	81	ก ค ซ ฑ ด ต ถ บ ผ ภ ศ ษ ส โ ๅ ๒ 7

6.5.2 Fine classification

In the fine classification, it is important to classify characters in the coarse classes 1, 3, 7 and 9 respectively. The same method as in the Section 6.2 and 6.5 is implemented, but, in this case, nine ring projection values are used. By the same method in Section 6.5, we find the indispensable attributes and compute the core and reduct sets of the examples in the Table 6.13. A set of reducts is used to generate decision rules. Using the decision rules to classify the printed characters in each example of coarse classes, we have the small group of characters to compute the max-min boundary of these characters in a final stage. The max-min boundary of each character (Figure. 6.4) is used

Table 6.13 Databases of Coarse Group 1

No.	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	D
1	1	1	2	3	3	3	2	1	1	D9
2	1	1	1	3	3	3	2	1	1	D9
3	1	1	1	3	3	3	1	1	1	D9
4	1	1	1	2	3	3	2	1	1	D7
5	1	1	1	2	3	3	1	1	1	D9
6	1	2	1	2	3	3	1	1	1	D3
7	1	2	2	2	3	3	1	1	1	D3
8	1	2	2	2	3	3	2	1	1	D1
9	1	2	1	2	3	3	2	1	1	D1
10	1	2	2	3	3	3	3	2	1	D7
11	2	2	2	2	3	3	3	1	1	D2
12	2	2	2	2	3	3	2	1	1	D1
13	1	2	2	2	3	3	3	1	1	D2
14	1	2	2	3	3	3	2	1	1	D9
15	1	1	1	2	3	3	3	1	1	D8
16	1	1	1	2	3	3	3	2	1	D8
17	1	1	1	3	3	3	3	1	1	D7
18	1	1	2	2	3	3	2	1	1	D7
19	1	1	2	3	3	3	1	1	1	D9
20	1	1	2	2	3	2	1	1	1	D9
21	1	2	1	2	3	2	1	1	1	D3
22	1	2	2	3	3	3	3	1	1	D7
23	1	2	2	2	3	3	3	2	1	D2
24	1	2	2	3	3	3	2	2	1	D9
25	2	2	2	2	3	3	2	2	1	D1
26	2	3	2	2	3	3	1	1	1	D6
27	2	2	2	2	3	3	1	1	1	D3
28	1	2	2	3	3	3	1	1	1	D9

to compute the rough membership function for decision rules and the numerals "5" and "1" are the set of all decision rules that fits the new information vector and rm [D], where D is the decision class (Figure. 6.5). From the example as mentioned above, the chosen answer depends on the max-min boundary classifier.

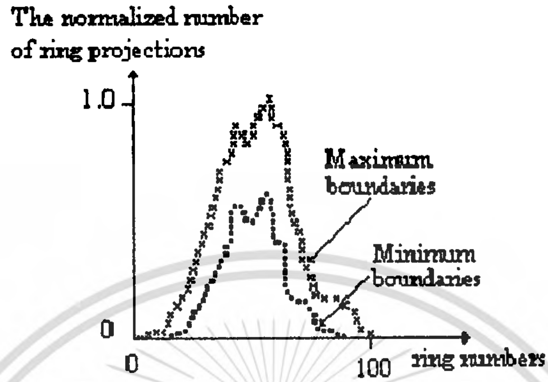


Figure 6.4 The examples of max-min boundaries.

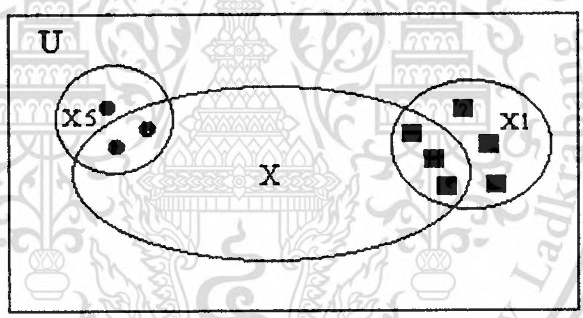


Figure 6.5. The rough membership values for decision rules.

6.6 Experiment study results

In the experiments, the learning sets consist of 1,376 training characters (four fonts Angsana New, Browallia New, Cordia New, and Eucrosia UPC, four sizes in each font, 14, 18, 22, and 28 points, as shown in Appendix A). The decision rules for coarse and fine classification can be obtained using the unknown sets as shown in Appendix B. In the max-min boundary classification, we compute rough membership function from a set of attributes that have one hundred max-min condition attributes based on the total pixels of a ring projection of each character. The unknowns used in this experiment are 2,752 Thai printed characters tested with 4 fonts that have 4 different sizes. The results obtained are shown in Table 6.14. From the results, invariant character recognition is obtained is accuracy above 87 % from images with sizes between 14X14 and 28X28 pixels.

Table 6.14 The recognition results for each character (four fonts, four sizes in each font) of the ring projection and rough set method.

Characters	Recognition rate (%)	Mis-classified to	Mis-recognized to
ก	81.25	-	ถ
ข	50.00	-	ช ล จ ษ (8)
ค	62.50	-	ด ฮ ๕ (6)
ฆ	81.25	-	ณ
ง	75.00	-	ใ ฎ
จ	75.00	-	ฉ ๕
ฉ	81.25	-	ฎ
ช	81.25	-	ซ
ซ	81.25	-	ช
ฌ	68.75	-	ญ ณ ถ
ญ	81.25	-	ฌ
ฎ	75.00	-	ข ฏ
ฏ	75.00	-	ข ฎ
ฐ	75.00	-	ว ฐ
ฑ	81.25	-	ด
ฒ	81.25	-	ร
ณ	81.25	-	ฆ
ด	62.50	ร (4)	ค ต (1)
ต	68.75	-	ค ๗
ท	75.00	๕	อี
ถ	81.25	-	ก
ธ	75.00	ถ	-
น	68.75	-	ซ
บ	68.75	-	ภ 0
ป	50.00	ฦ	-
ผ	68.75	-	ย ฝ
ฝ	81.25	-	ผ

Table 6.14 (Continued)

Characters	Recognition rate (%)	Mis-classified to	Mis-recognized to
ฟ	81.25	-	พ
ภ	68.75	บ (4)	ก (1)
ม	50.00	ท (7)	ฒ (1)
ย	75.00	-	ณ 5
ร	81.25	-	ไ
ล	68.75	-	ซ 5
ว	68.75	-	ฎ ไ
ศ	43.75	-	ค (8) 5 (1)
ษ	50.00	-	ท ฑ ม ก ฐ
ส	75.00	-	ศ
ห	75.00	-	ซ
ฬ	75.00	-	ซ อ็
อ	81.25	-	3
ฮ	81.25	-	ส
อั	75.00	-	เ
า	68.75	-	ๅ
อำ	62.50	๐	-
อิ	75.00	-	อั
อี	75.00	-	อี
อึ	75.00	-	อี
อื	68.75	-	อึ อ้
อุ	75.00	-	เ
อู	75.00	-	ข อ้
เ	81.25	-	อ่
โ	68.75	ร	3
ใ	81.25	-	ไ
ไ	75.00	-	ใ
อ็	68.75	ๅ	๗ ๘
ฤ	75.00	-	ถ
ฦ	75.00	-	ภ
ๅ	68.75	-	า
อ่	68.75	-	1

Table 6.14 (Continued)

Characters	Recognition rate (%)	Mis-classified to	Mis-recognized to
อ้	75.00	-	4
อ๊	75.00	-	๘
อ๋	68.75	-	๐
ๆ	75.00	-	ฯ
ฯ	68.75	-	ๆ
อ์	81.25	-	๙
0	81.25	-	๐
1	87.50	-	อ่
2	87.50	-	๑
3	87.50	-	โ
4	81.25	ง	-
5	81.25	-	โ
6	81.25	-	อำ
7	81.25	-	๒
8	75.00	-	3 ๓
9	75.00	-	ฤ
๐	81.25	-	๐
๑	81.25	๓	-
๒	81.25	-	7
๓	81.25	๑	-
๔	81.25	-	๔ ๖
๕	75.00	-	๔
๖	75.00	-	๗ ๔
๗	81.25	-	๖
๘	81.25	ฯ (1)	ฤ ฦ (2)
๙	75.00	-	ฯ ๔

CHAPTER 7

FOURIER DESCRIPTORS FOR INVARIANT PRINTED THAI CHARACTERS RECOGNITION

This chapter presents a method of recognition of Thai characters using Fourier descriptor for solving the problem of rotated, translation, and scaling character recognition. The Fourier descriptor can also be used detection of boundary features from a character image. The boundary representation is essential in shape description, and recognition. The high quality boundary shape representation obtained using the low frequency coefficients is the ones of greatest interest. A neural network classifies the boundary information of Fourier descriptors. A typical character recognition system consists of three stages as follows.

The preprocessing stage involves noise removal, smoothing borders of character, stroke thickness normalization, skew correction, character image segmentation, and contour-following respectively.

In the second stage, the Thai character was classified into three rough classes using the number of inner contours. Then, the Fourier coefficients of the outer contour are computed using Fourier descriptors and trained by the neural networks.

In the testing stage, testing data is fed through the preprocessing stage and contours of the character are extracted. Then, the characters are classified using the inner contour and compute the Fourier coefficient of the outer contour. Finally, the twelve Fourier coefficients activate the trained neural network.

7.1 Features extraction

7.1.1 Inner contours (loop contours) extraction

We applied a contour-following algorithm (Richard and Peter, 1973), and obtained the detection of a Thai character's head. Figure 7.1 (a) shows a normal pattern for inner contour (loop contours) extraction strategies. Using a contour-following algorithm, an external edge of the character appears as shown in Figure 7.1 (b). Later, the contour-following algorithm detects an internal edge. The internal edge appears, which we refer to as the " head " of the character, as shown in Figure 7.1 (d).

7.1.2 Outer contour detection based on Fourier descriptor

The Fourier descriptor has been used as the popular curve descriptor in many applications. It is the method of describing a closed curve by a set of Fourier coefficients. Because a character contour is also a closed curve, the character contour can be represented by Fourier coefficients.

7.2 Classification

The recognition system was separated into two stages. First, in the rough classification stage, the loop contours of a character from the previous section were classified into 3 classes using the number of loop contours such as class 1, class 2, and class 3. These are the groups of characters as shown in Table 7.1, which consist of no loop, one loop, and two loops respectively. Second, in the fine classification stage, a neural network classifies the outer contour information of Fourier descriptors.

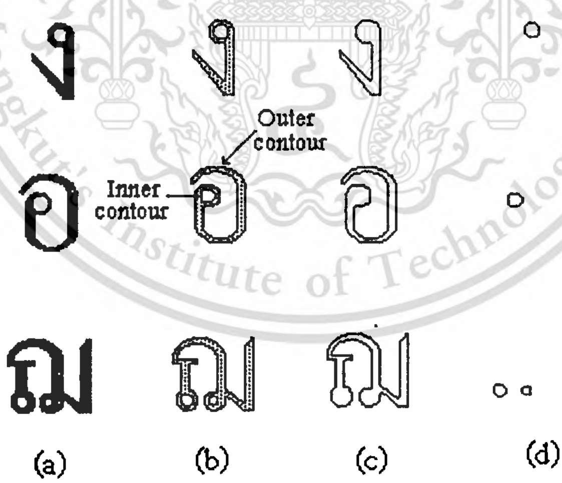


Figure 7.1 Examples of feature extraction

(a) Original image of printed Thai characters "ง", "อ", and "ฌ

(b) Outer and inner contour of printed Thai characters "ง", "อ", and "ฌ".

(c) Outer contour of printed Thai characters "ง", "อ", and "ฌ".

(d) Inner contour of printed Thai characters "ง", "อ", and "ฌ"

Table 7.1 Three classes in Thai alphabet

Level / Class	Consonant	Vowel: Upper	Vowel: Middle	Vowel: Lower	Tonal
1	ก ธ		า		+ ่
2	ฃ ค ง จ ซ ช ฑ ด บ ป ผ ฝ ห ฟ ภ ย ต ถ ท ร ว ศ ส อ ล ฤ ฦ ๆ ฯ ฌ ธ	ั ิ ี ํ ็	เ แ ไ โ	ุ ู	้ ๊ ๋ ์
3	ฆ ฉ ณ ฒ ฎ ฏ ญ น ม ษ ห พ ฮ	ื	ะ ๆ		

7.3 Fourier Descriptors

Fourier descriptors in this system as shown in Figure 7.2. The character boundary was presented by co-ordinates $P_m(X_m,Y_m)$ and computed

$$a[k] = \frac{1}{L} \sum_{m=1}^{L} x[m]\, e^{-jk(2\pi/L)m} \tag{7.1}$$

$$b[k] = \frac{1}{L} \sum_{m=1}^{L} y[m]\, e^{-jk(2\pi/L)m} \tag{7.2}$$

where a[k], b[k] are the coefficients of Fourier descriptors

L is the curve length, m is 1 to L, and

k is 0 to L-1.

Then, to cut the DC component of the Fourier descriptor a[0] and b[0], the other data can be used to independently perform the position of the character boundary. Computed by

$$r(k) = \sqrt{|a[k]|^2 + |b[k]|^2} \tag{7.3}$$

where k varies from 1 to L-1. $|a[k]|$ and $|b[k]|$ are absolute values, and complex numbers. The r[k] are independent of both the translation, and rotation of the character.

In the case of the application for independent scaling work, the Fourier descriptors S[k] are derived as

$$S[k] = r[k]/r[1] \tag{7.4}$$

The first 10 - 15 descriptors s[k] are sufficient for character description.

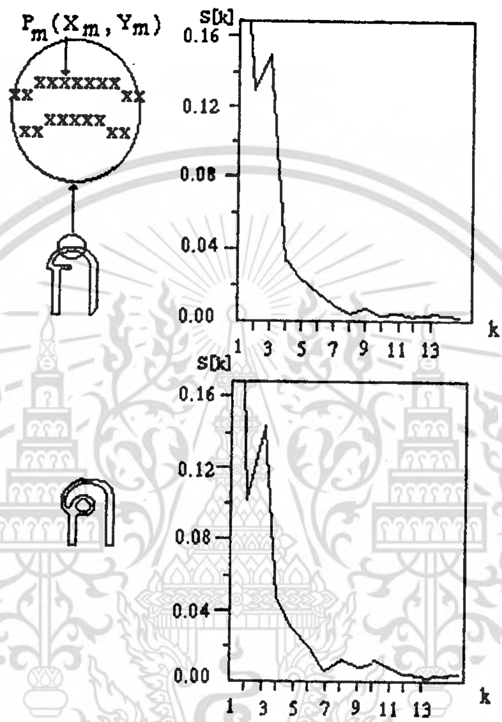


Figure 7.2 Fourier descriptors of 'ฑ' and 'ฅ'

Fourier descriptors s[k] are symmetrical around position (L-1)/2. The s[l] of character is the same for every characters. For an application to be recognized we neglected s[l], and used the first half of the Fourier descriptors s[2] to s[(L-1)/2]. Because of this consideration, Fourier descriptors of character graphs, s[k] are reduced to near zero when k increases to more than 15 as shown in Figure 7.2. The lower order coefficients are the high quality boundary shape representation. We can select the first order coefficients 10 - 15 descriptors for sufficient character recognition. However, since some Thai characters are very close in shape, the use of 10 Fourier descriptors gives better results.

7.4 Back-propagation neural classifier

The back-propagation neural classifier consists of three layers of processing units, that performed a transformation on their summed inputs and produced continuous outputs between 0 and 1. Back-propagation is an iterative algorithm designed to minimize the mean square error between the actual input and desired output. In Thai character recognition, the multi-layer neural networks are fed by 12 coefficients of Fourier descriptors to the input nodes. The hidden layer of the network has a maximum of 25 neurons and the output layer has 8 neurons, corresponding to the number of recognized characters. In the experiment, the neural networks are operated in two

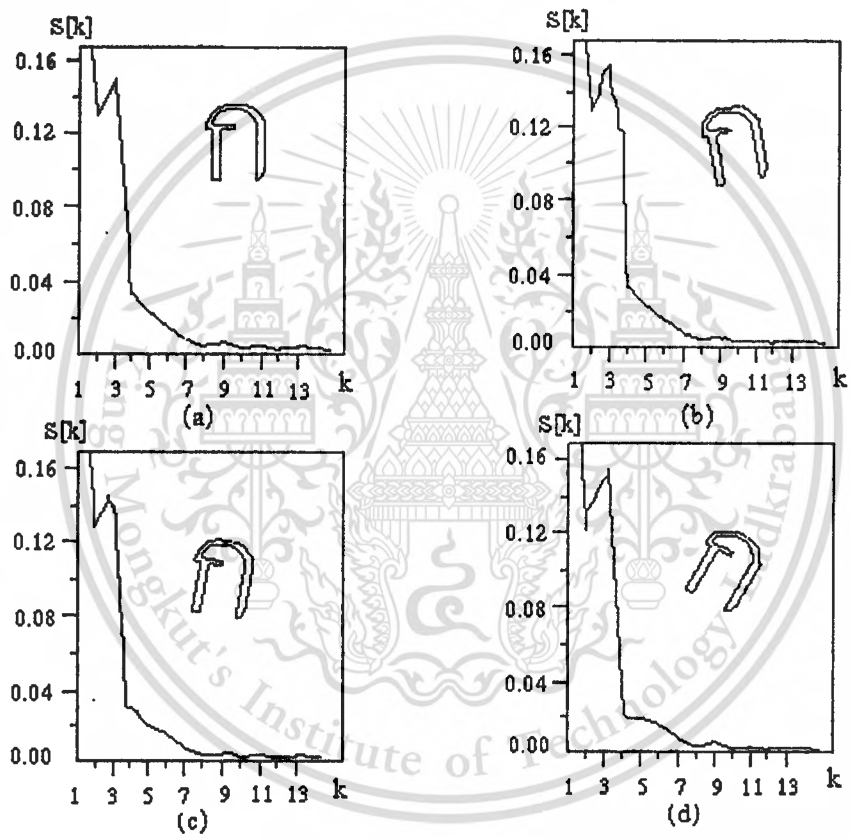


Figure 7.3 The results of Fourier descriptors from the character "ก"

(a) The normal character

(b) The character "ก" rotated to -10°

(c) The character "ก" rotated to 10°

(d) The character "ก" rotated to 30°

modes: training and testing. In the training mode, a set of training data is used to adjust the weights of the network interconnections. The training neural networks are determined by these weights. In testing mode, the trained network is activated by the testing data.

7.5 Experimental results

In our experiments we use the unknown printed Thai characters of 4 fonts and 4 sizes of 86 characters as shown in Appendix B. All are printed with COMPAQ Ink Jet IJ 650 and scanned with ACER SZW 3000U in black and white mode. We detect the outer contour of the character image, and the inner contours according to the section 7.1. After that the characters are divided into three groups by rough classification using the number of inner contours (the head of the character) of the characters. Then the outer contours of the character image are transformed using Fourier descriptors. We found that the graphs of the Fourier coefficients do not depend on the rotation of character "ก"as rotated in different angles -10°,0°,10° and 30°, as shown in Figure 7.3.

Graphs of the different sized characters are shown in Figure 7.4. In this figure, the graphs of the Fourier descriptors of the contour of character "วิ" is almost the same, because the contour of the character image is still the same size and is not a changeable ratio. The contour information of the character transformation using Fourier descriptors are brought into the back-propagation neural classifier. We used learning by this network for finding the weights of the network for recognition system.

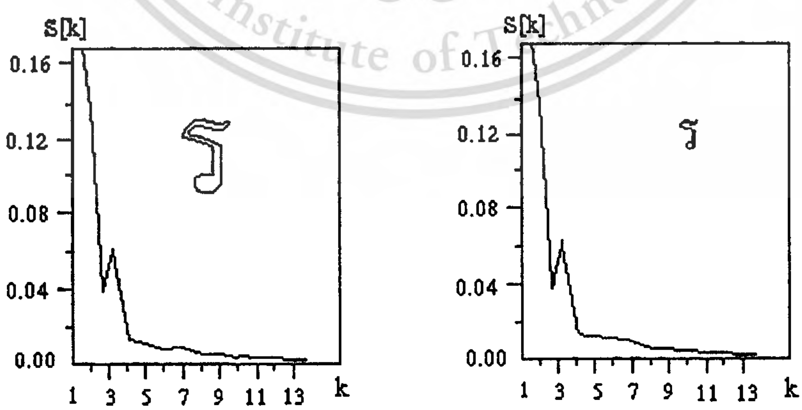


Figure 7.4 The result of the Fourier descriptors from Thai character "วิ" with different sizes

In our experiments the learning sets consist of 1,376 training characters (four fonts, four sizes in each font, Appendix A). The largest characters have 28X28 pixels. The smallest characters have 14X14 pixels. The unknowns used in this experiment are 2,752 Thai characters tested with 4 fonts that have 4 different sizes. The results obtained are shown in Table 7.2. From the results, invariant character recognition is obtained with above 90 % accuracy from images with size between 14X14 and 28X28 pixels. The performance of the model decreases slightly for smaller characters.

Table 7.2 The recognition results for each character of the Fourier descriptors and neural networks

Characters	Recognition rate (%)	Mis-recognized to
ก	93.75	ถ
ข	81.25	บ
ค	50.00	ด ต
ฆ	93.75	ม
ง	93.75	จ
จ	93.75	ง
ฉ	93.75	น อ
ช	50.00	ซ
ซ	50.00	ช
ฌ	87.50	ญ ณ
ญ	87.50	ฌ ณ
ฎ	87.50	ฏ
ฏ	87.50	ฎ
ฐ	87.50	จ ร
ฑ	93.75	ท ห
ฒ	81.25	ณ ญ
ณ	81.25	ณ ญ
ด	50.00	ค ต
ต	50.00	ค ด
ท	87.25	ฑ ห
ถ	93.7	ก
ธ	93.75	ร

Table 7.2 (Continued)

Characters	Recognition rate (%)	Mis-recognized to
น	93.75	ม บ
บ	50.00	ป ฦ
ป	50.00	บ ฦ
ผ	75.00	พ ฝ
ฝ	75.00	ผ ฟ
ฟ	75.00	พ ฝ
ภ	93.75	ก
ม ·	87.25	น บ
ย	87.50	บ
ร	75.00	โ
ล	93.75	ส
ว	93.75	า
ศ	50.00	ค ด
ษ	93.75	ผ
ส	93.75	ล
ห	93.75	ท
ฬ	93.75	ฟ
อ	93.75	ฮ
ฮ	93.75	อ
อั	93.75	อ้
า	50.00	ๅ
อํ	62.50	๐
อิ	93.75	อี
อี	93.75	อิ
อึ	87.50	อี
อื	75.00	อึ
ฤ	62.50	เ
ฎ	81.25	ข บ
เ	93.75	ฤ
โ	75.00	ร
ใ	93.75	ไ
ไ	93.75	ใ
อ๊	75.00	๔ ๕ ๘

Table 7.2 (Continued)

Characters	Recognition rate (%)	Mis-recognized to
ฤ	81.75	ถ
ฦ	81.75	ภ
า	50.00	า
อ่	75.00	1
อ้	81.25	อั
อ๊	93.75	๙
อ๋	100.00	-
ๆ	81.25	ฯ
ฯ	93.75	ๆ
อ์	87.50	๙
0	93.75	๐
1	100.00	อ่
2	93.75	7
3	100.00	-
4	100.00	-
5	93.75	ร
6	43.75	9
7	93.75	2
8	93.75	0
9	43.75	6
๐	43.75	๐
๑	93.75	๐ 0
๒	93.75	๓
๓	93.75	๒
๔	93.75	๕
๕	87.50	๔ ๘
๖	87.50	๗
๗	87.50	๖ อ๊
๘	87.50	๔ อ๊
๙	87.50	อ์

CHAPTER 8

DISCUSSION

8.1 A comparison invariant recognition with other recognition

An invariant is a property of an object or class of object that does not change with viewpoint or object pose, which can therefore be used to help distinguish it from other objects. The different properties of this method with other methods are:

- it is independent of the translation parameter of the objects;
- it is independent of the orientation parameter of the objects; and
- it is unchanged for different objects of the same type (scale).

In pattern recognition systems the features are used to decide the class to which the observation vector belongs. So, feature extraction is important in the recognition process. The purpose of feature extraction is to reduce data by measuring certain "features" or "properties" that distinguish input patterns. In invariant recognition, we consider invariance with respect to translation, rotation, and scale differences in input images. The problem of invariance is difficult because of the large number of training samples for which the classifier needs to be trained. To alleviate the difficulty, we use rough set theory to reduce the data redundant in chapter 6, and use just the fifteen Fourier coefficients and contour loops for recognition in chapter 7. There are several invariant methods such as moment method, Fourier descriptors, ring projection, etc. In this thesis, we use ring projection and Fourier descriptors for feature extraction.

8.2 Recognition system analysis based on rough classification

The classification procedure consists of three phases: feature extraction using ring projection method, coarse classification, and fine classification. The method uses attribute data from ring projection and redefines the value of each attribute according to a certain metric. In the next step, we create the decision attribute of each character using the condition in section 6.3. Using the rough set theory that has two major concepts, reduct and core is defined. The main idea can be obtained by the following procedures:

- calculate the core of the problem,
- eliminate or substitute a variable by another one, and
- redefine the problem using new basic categories.

These concepts are important in the knowledge base reduction. This means that rough modeling can be used to find models from large database, and find the most important attributes in a data set where the attribute number is large.

The advantage of the ring projection and rough set method can reduce the redundant data, and be used to solve the problem of invariant characters recognition. In practice, there will be some differences among the training characters and the unknown characters which from the limited resolution and error of the edge characters. These differences lead to mis-classification ring-projection problem.

In order to make shape characters 100% identical, for a given object, we must have ideal conditions, where the resolution is infinite and the scanner is perfect. Obviously these conditions cannot be met in the real world, although it is possible to get close to them.

8.3 Recognition system analysis based on Fourier descriptors

Fourier descriptor has been a tool for pattern recognition. It requires closed curves. This, of course, is satisfied by the boundaries of scanned graphic symbols. Essentially, Fourier descriptors can describe any pattern that can be approximated by an infinitesimally thin boundary curve. We are going to derive invariant features from the pattern image $f(x,y)$ which may consist of boundary oriental characters. The translation invariance can be achieved by translating the origin of the coordinate system to the center of mass of the pattern, denoted by (x_0, y_0). The starting point of the curve is arbitrarily chosen, and the curve is traced clockwise over the entire boundary until it returns to the starting point.

The scale invariance can be obtained by the Fourier transforming coefficients of the boundary of pattern image into a polar coordinate system, and normalizing them. So, the features s(k) in equation 7.4 obtained in this way are also invariant to translation, rotation, and scaling respectively. The Fourier descriptor values are reduced to near zero when k increases to more than 15. Thus, we can select the first order coefficients

10-15 descriptors for sufficient recognition. From the properties of Fourier descriptors, we can use the fifteen features that have enough data to recognize characters. The disadvantages of this method are that it the inner loops or heads of character cannot be detected, or if there is a solid head or blurring, then particular method of analysis cannot be used.

8.4 Why Rough Sets?

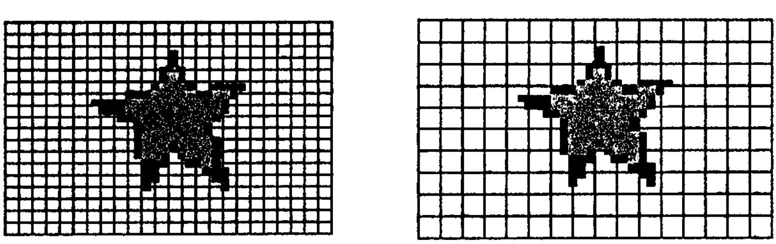
Rough sets have already been applied to a wide variety of application domains with satisfactory results. Rough set theory is a formal mathematical tool that can be applied to reducing the dimensionality of data sets used for training classifiers. Rough set dimensionality reduction removes redundant conditional attributes from nominal data sets, while making sure that no information is lost. The approach is fast and efficient, making use of standard set theory operations. From these reasons, rough sets are suitable for use in the classification of recognition system.

8.5 The difference between fuzzy sets and rough sets

The difference between fuzzy sets and rough sets as follows:

Fuzzy sets: how gray is the pixel.

Rough sets: How big is the pixel.



CHAPTER 9

CONCLUSION OF INVARIANT THAI PRINTED CHARACTERS RECOGNITION

In this thesis, an attempt has been made to use a ring projection and rough set method, Fourier descriptor and neural networks for recognizing multi-font Thai printed characters. Strengths and weaknesses of the method to recognize 2-D object invariant to translation, rotation, and size is reported as follows:

9.1 Conclusion of ring projection and rough set method for invariant Thai printed characters recognition

A ring projection method is used to extract features from the invariant character images. The ring projection value is obtained using the total number of foreground pixels as distributed along circular rings, but independent from the shape's position, orientation, and scaling of character. Thus, they have invariant properties. Furthermore, this method does not concern the thinning method, and edge detection.

Rough sets have been introduced as a tool to deal with inexact, uncertain or vague knowledge in artificial intelligence applications. The idea of rough sets works with lower and upper approximation. The main implementation of the rough set in this work is to reduce a database of characters using the core and reduct of equivalence relations. Two major concepts in rough set theory reduct and core, will be defined from these equivalence relations. These concepts are important in the knowledge base reduction.

The core of R, COR(R), is the set of relations, which appear in all reducts of R, i.e. the set of all indispensable relation to characterize the relation R.

The reduct of R, RED(R), is defined as a reduced set of relations which conserve the same inductive classification of set R, where R is a family of equivalence relations.

Conclusion of the computation of core as follows:

- Eliminate one attribute at a time and see if the training set is ambiguous: only that attribute can discriminate between the ambiguous signals.

- Accumulate the attribute resulting in ambiguous data: that is the core.
- These attributes from core must be in every reduct.

Conclusion of the computed minimal reducts process as follows:

- To the core add one attribute at a time and compute the number of ambiguities.
- Select the attribute(s) with the fewest ambiguities: there may be several-save these, as we will use them to compute the reduct.
- Add that attributes to the core and repeat a review step until there are no ambiguities: This is a reduct.
- Calculate reducts for all attributes with equivalent number of ambiguities- yields multiple reducts.

The procedures used a ring projection method to extract features from the examples and to classify them by the rough set method. The classification process is divided into 2 levels: coarse classification and fine classification, respectively. In the fine classification, the same method is used as in the coarse classification. After classification, if the examples have many members of characters, the rough membership value of max-min boundary ring projection use to classify the character from the other.

It has been established that the ring projection and rough set technique proposed in this thesis succeeds in solving invariant features and reducing the dimensionality of data sets used in classification tasks and that the reduced data sets are still rich in information. An exhaustive comparative study of the applicability of this technique to the neural networks should be performed. The results show that the neural classifier cannot be used directly. We must group the ring projection values for learning. This means that the coarse classification is important in the computation of neural network's weight. Because of the ring projection values, some characters overlap.

9.2 Conclusion of Fourier descriptors and neural networks for invariant Thai printed characters recognition

This approach described a method to invariant character recognition that utilizes the number of heads for the coarse classification and a Fourier descriptor with back-propagation neural classifier for the fine classification of learning pattern input data. Fourier descriptor is translation invariant with respect to the spectrum. If we cut the DC component of Fourier descriptor, the other data set can be used to make independent the position of the character boundary. The most important factor of the Fourier descriptor method is the boundary of characters. The defects of inner and outer contours are the problem of this method. The back-propagation neural classifier is used to learn the training data in the training mode. A set of training data is used to adjust the weights of the network interconnections. The training neural networks are determined by their weights. After the training procedure, the trained network is used to classify the testing data.

The weak points of Fourier descriptors and neural networks are as follows:

- Fourier descriptors cannot classify similar characters. It is necessary to use the special techniques for recognizing similar characters.

- If the head of the characters has defects such as a broken head, this particular method of analysis cannot be used.

- The recognition rate is reduced if the character image borders and stroke thickness of character has a defect or error. Then, it is necessary to use the local smoothing of character image borders and stroke thickness normalization.

9.3 The comparison of the two invariant character recognition methods

The character classifiers were trained on a set of 1,376 characters extracted from the training images. We tested on a different set of 2,752 characters from our testing databases. The testing databases are shown in Appendix B. The considered images have differences in scale, translation, and rotation. They are first normalized with respect to the centroid of character images. Then, they are normalized with respect to scale and rotation using ring projections or Fourier descriptors based techniques.

In the comparison of the recognition rate between the two invariant object recognition methods, the recognition rate of rough classification is slightly lower than the Fourier descriptor as shown in Table 9.1

The principal weakness of the current approach is its difficulty in the similar characters. This problem is solved using the special techniques. Furthermore, the defect or error of character image borders is a problem to reduce the recognition rate. The recognition results are dependent on many steps, most of which we have not attempted to improve, but their influences are measurable in overall system performance.

Table 9.1 The comparison of the recognition results

Recognition methods	Recognition rate (%)
Ring projection and rough sets	87
Fourier descriptors and neural networks	91

9.4 Future work

Future work will concentrate on the following items:

- The ring projection method in this thesis might be used to recognize handwritten Thai characters.
- The combination between Fourier descriptors and rough set method will be investigated to recognize printed or handwritten Thai characters.
- The frequency information obtained from the Fourier descriptor is global, and therefore, local variation of the shape can affect all Fourier coefficients The Fourier descriptor does not have a multi-resolution representation. From the above reason, Fourier descriptor can be developed by adding the wavelet transformation. This approach includes the advantages of Fourier descriptor and wavelet transformation. Then Fourier-wavelet descriptors have the invariant properties and a multi-resolution representation.

BIBLIOGRAPHY

Airphaiboon, S., Sangworasil M., Attachoo B. and Kimpan C. 1989. “Recognition of Handwritten Thai Characters Considering the Head of Character.” IEEE International Conference on Image Processing ICIP’89, Singapore, pp.1296-1304.

Airphaiboon, S. and Kondo, S. 1996. “Recognition of Handprinted Thai Characters using Loop Structures.” IEICE Trans. INF.&SYST. E79-D(9), pp.1296-1304.

Attoh-Okine N.O., 1997. "Rough et Application to Data Mining Principles in Pavement Management Database." Journal of Computing in Civil Engineering, Vol. 11, No. 4, pp. 231-237.

Bjanger M.S. 1999. "Vibration Analysis in Rotating Machinery using Rough set Theory and ROSETTA." Report 78070 Databehandling Projektarbeid, Department of Computer and Information Science, Norwegian University of Science and Technology.

Chen G., and Bui T.D. 1999. "Invariant Fourier-Wavelet Descriptor for Pattern Recognition.", Pattern Recognition, Vol. 32, pp.1083-1097.

Chung Y.Y., and Wong M.T. 1997. "Handwritten Character recognition by Fourier Descriptors and Neural Network.", IEEE TENCON Speech and Image Technologies for Computing and Telecommunications, pp. 391-394.

David Freedman M. 1974. “Optical Characters Recognition.” IEEE Spectrum.

Duntsh I. and Gunther Gediga G. 1998. "Uncertainly measures of rough set prediction.", Artificial intelligence 106, pp.109-137.

Fausett L.1994. **Fundamentals of Neural Networks Architectures, Algorithms, and Applications.**, Prentice Hall International Editions.

Hagan M.T., Dernuth H.B. and Beale M. 1996. **Neural Network Design.**, PWS Publishing Company.

Hirananichakorn P., Agui, T. and Nakajima M. 1982. “A Recognition Method of Thai Characters.” IECE trans. E65(12), pp.737-744.

Hirananichakorn P., Agui, T. and Nakajima M. 1984. “Recognition of Thai Characters by using Local Features.” IECE trans. E67(8), pp.425-432.

Jeong Chaj-Sup and Jeoug Doug-Seok, 1999. "Handwritten digit Recognition Using Fourier Descriptors and contour Information.", The 1999 IEEE Tencon, Vol. II, Sep. pp.1283-1286.

Kasemsiri W. and Kimpan C. 2000. "Printed Thai Characters Recognition using 2 Class Levels Classification and Rough Sets.", Proceedings 2000 Asia-Pacific Symposium on Broadcasting and Communications APSBC 2000, pp. 412-416.

Kasemsiri W., and Kimpan C. 2001. "Printed Thai Character Recognition Using Fuzzy-Rough Sets.", TENCON Proceedings of IEEE Region 10 International Conference, Vol. 1, pp. 326-320.

Kim D., and Barng Suan-Yamg 2000. "A Handwritten Numeral Character Classification Using Tolerant Rough set.", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.22 , No.9. pp. 923-937.

Kimpan C., Itoh A. and Kawanishi K. 1983. "Recognition of Printed Thai Characters using a Matching Method." Proc. IEE. 130(Pt.E. 6), pp.183-188.

Kimpan C., Itoh A. and Kawanishi K. 1987. "Fine Classification of Printed Thai Character Recognition using the Karhunen-Loève Expansion." Proc. IEE. 134 (Pt.E.5), pp. 257-264.

Kimpan C. 1989. "Printed Thai Character Recognition using Topological Properties Method." INT.J. Electronics. 60(3), pp. 303-329.

Komorowski J., Pawlak Z., Polkowski L. and Skowron A. Rough Sets: A tutorial.

Lambert-Torres G., Silva A.P.A. da, Quintana V.H. and Silva L.E.B. da 1996. "Classification of power system operation point using rough set techniques." IEEE International Conference on Systems, Man, and Cybernetics, Vol. 3, p.p. 1898 – 1903.

Lau S.S.Y. 1993. "Image Segmentation Based on the Indiscernibility Relation." Proceedings of the International Workshop on Rough Sets and Knowledge Discovery (RSKD'93). Alberta, Canada, pp.395-402.

Lin T.Y., and Cercone N. 1997. **Rough Sets and Data Mining Analysis of Imprecise Data.** Kluwer Academic Publishers.

Lken T. 1999. "Rough Modeling Extracting Compact Models from Large Databases." Report Knowledge Systems Group, Norwegian University of Science and Technology.

Mahmoud S.A. 1994. "Arabic Character Recognition Using Fourier Descriptors and Character Contour Encoding.", Pattern Recognition, Vol. 27, No.6, pp.815-824.

Man, Gary M.T. and Poon, Joe C.H. 1992. "An Enhanced Approach to Character Recognition by Fourier Descriptor." IEEE Proceedings ICCS/ISITA'92. Singapore, pp.558-562.

Marang U., Phokharatkul P., and Kimpan C. 2000. "Recognition of Printed Thai Characters using Boundary Normalization and Fuzzy Neural Networks.", The Fourth Symposium on Natural Language Processing SNLP 2000, pp.154-164.

Mitatha S., Dejharn K., Chevasuvit F., Chankuang B. and Kasemsiri W. 2001. "Experimental Results of Using Rough Sets for Printed Thai Characters Recognition." TENCON Proceedings of IEEE Region 10 International Conference, Vol. 1, pp. 331-334.

Nadler M. and Smith E.P. 1992. **Pattern Recognition Engineering.** New York: John Wiley & Sons Inc.

Nejman D. 1995. "Rough Sets in Handwritten Numerals Recognition.", ICA WUT Reports on Rough set.

Nelson M.M. and Illingworth W.T. 1990. A Practical Guide to Neural Nets, Addison-Wesley Publishing, pp. 50-52.

Pal S.K. and Skowron A. 1999. Rough Fuzzy Hybridization: A New Trend in Decision-Making. Springer. pp. 263-274.

Persoon E. and Fu K.S. 1977. "Shape Discrimination using Fourier Descriptors." IEEE Transactions on systems, Man, and Cybernetics. SMC-7(3), pp.170-179.

Phokharatkul P., Marang U. and Kimpan C. 2000. "Recognition of Printed Thai Characters Elliptic Fourier Descriptors and Genetic Neural Networks.", Proceedings 2000 Asia-Pacific Symposium on Broadcasting and Communications APSBC 2000, pp.406-411.

Phokharatkul P. and Kimpan C. 2001. "A Rough Ring Projection Method for Invariant Thai Character Recognition.", World Multiconference on Systemics, Cybernetics and Informatics SCI/ISAS 2001, Orlando, USA, pp. 66-70.

Phokharatkul P., Suebsanit T., Pantaragphong P., Pinngern O. and Kimpan C. 2002. "Printed Thai Characters Recognition Using Rough Classification.", Proceedings of the International Conference on Artificial Intelligence in Engineering & Technology ICAIET 2002, pp. 284-290.

Phokharatkul P. and Kimpan C. 2002. "Handwritten Thai Character Recognition Using Fourier Descriptors and Genetic Neural Networks.", Computational Intelligence An International Journal (accepted for publication).

Richard, D.D.; and Peter E.H.; 1973. Contour Following, Pattern Classification and Scene Analysis., A Miley-Interscience Publication, pp. 290-293.

Schurmann, J., 1992. "Document analysis-from pixels to contents.", Proceeding of the IEEE 80(7):, pp. 1101-1119.

Sienkiewicz J., 1995. "Rough Sets for Boolean Functions Minimization." Electronic Bulletin of the Rough Set Community EBRSC Vol. 3, No. 2.

Szmurlo M. 1997. "Boundary Normalization for Recognition of Non-Touching Non-degraded Characters." Proceedings of the Fourth International Conference on Document Analysis and Recognition, Vol. 2, pp.463-466.

Tang, Y.Y., Bing F. Li, Hong Ma and Jimming Liu, 1996. "A Novel Approach to Optical Character Recognition Based on Ring-Projection-Wavelet-Fractal Signatures." IEEE Processing of ICPR'96, pp. 325 – 329.

Tang, Y.Y., Bing F. Li, Hong Ma and Jimming Liu, 1998. "Ring-Projection-Wavelet-Fractal Signatures: A novel Approach to Feature Extraction." IEEE Transactions on Circuit and Systems-II: Analog and Digital Signal Processing. Vol. 45, No. 8, pp.1130 – 1134.

Taxt T., and Bjerde K.W. 1994. "Classification of Handwritten Vector Symbols using Elliptic Fourier Descriptors.", Proceedings of the 12th IAPR International Conference on Computer Vision & Image Processing, Vol. 2, pp.123-128.

Tebelskis J. 1995. "Speech Recognition Using Neural Networks." Doctoral Thesis, Carnegie Mellon University.

XIA, F. 1995. "On a New Basic Concept and Topological Invariant.", Image Analysis and Processing 8th International Conference, ICIAP '95 , pp.341-346.

APPENDIX A

TRAINED SAMPLES

The samples selected by the training algorithm. This data is printed with HP Laser Jet and scanned with HP Scan Jet 6100 C in black and white mode. All the samples used for training are shown in Figure A.1-A.4.

ก ข ค 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 อ้ า อํ อิ อี อึ อื อุ อู
เ โ ใ ไ อ็ ฤ ฦ ฦ ๅ อ่ อ้ อ๊ อ๋ ๆ ฯ อ์ ๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

ก ข ค 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 อ้ า อํ อิ อี อึ อื อุ อู เ โ ใ ไ อ็ ฤ ฦ ฦ ๅ อ่ อ้ อ๊ อ๋ ๆ ฯ อ์ ๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0
1 2 3 4 5 6 7 8 9

ก ข ค 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 อ้ า อํ อิ อี อึ อื อุ อู เ โ ใ ไ อ็ ฤ ฦ ฦ ๅ อ่ อ้ อ๊ อ๋ ๆ ฯ
อ์ ๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

ก ข ค 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 อ้ า อํ อิ อี อึ อื อุ อู เ
โ ใ ไ อ็ ฤ ฦ ฦ ๅ อ่ อ้ อ๊ อ๋ ๆ ฯ อ์ ๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘
๙ 0 1 2 3 4 5 6 7 8 9

Figure A.1 Thai printed characters (font Angsana New, size 14, 18, 22 and 28 points)

ก ข ค 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 อ้ า อํ
อิ อี อึ อื อุ อู เ โ ใ ไ อ็ ฤ ฦ ฦ ๅ อ่ อ้ อ๊ อ๋ ๆ ฯ อ์ ๐ ๑ ๒ ๒๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

ก ข ค 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 อ้ า อํ อิ อี อึ อื อุ อู เ โ ใ ไ อ็ ฤ ฦ ฦ ๅ อ่ อ้ อ๊ อ๋ ๆ ฯ อ์ ๐ ๑ ๒
๒๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

ก ข ค 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 อ้ า อํ อิ อี อึ อื อุ อู เ โ ใ ไ อ็ ฤ ฦ
ฦ ๅ อ่ อ้ อ๊ อ๋ ๆ ฯ อ์ ๐ ๑ ๒ ๒๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8
9

ก ข ค 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 อ้ า อํ
อิ อี อึ อื อุ อู เ โ ใ ไ อ็ ฤ ฦ ฦ ๅ อ่ อ้ อ๊ อ๋ ๆ ฯ อ์ ๐ ๑
๒ ๒๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

Figure A.2 Thai printed characters (font Browalia New, size 14, 18, 22 and 28 points)

ก ข ค 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 อั า อํ อิ อี อึ อื
อุ อู เ โ ใ ไ อ็ ฤ ฦ ฦ ๅ อ่ อ้ อ๊ อ๋ ๆ ฯ อ์ ๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

ก ข ค 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 อั า อํ อิ อี อึ อื อุ อู เ โ ใ ไ อ็ ฤ ฦ ฦ ๅ อ่ อ้ อ๊ อ๋ ๆ ฯ อ์ ๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗
๘ ๙ 0 1 2 3 4 5 6 7 8 9

ก ข ค 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 อั า อํ อิ อี อึ อื อุ อู เ โ ใ ไ อ็ ฤ ฦ ฦ ๅ อ่ อ้ อ๊
อ๋ ๆ ฯ อ์ ๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

ก ข ค 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 อั า อํ อิ อี อึ อื
อุ อู เ โ ใ ไ อ็ ฤ ฦ ฦ ๅ อ่ อ้ อ๊ อ๋ ๆ ฯ อ์ ๐ ๑ ๒ ๓ ๔ ๕
๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

Figure A.3 Thai printed characters (font Cordia New, size 14, 18, 22 and 28 points)

กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยรลวศษสหฬอฮอัาอํอิ
อีอึอือุอูเโใไอ็ฤฦ ฦๅอ่อ้อ๊อ๋ ๆฯอ์๐๑๒ ๒๓๔๕๖๗๘๙0 1 2 3 4 5 6 7 8 9

กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยรลว
ศษสหฬอฮอัาอํอิอีอึอือุอูเโใไอ็ฤฦ ฦๅอ่อ้อ๊อ๋ ๆฯอ์๐๑๒
๒๓๔๕๖๗๘๙0 1 2 3 4 5 6 7 8 9

กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟ
ภมยรลวศษสหฬอฮอัาอํอิอีอึอือุอูเโใไอ็ฤฦ ฦๅ
อ่อ้อ๊อ๋ ๆฯอ์๐๑๒ ๒๓๔๕๖๗๘๙0 1 2 3 4 5 6 7 8 9

กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธน
บปผฝพฟภมยรลวศษสหฬอฮอัาอํอิ
อีอึอือุอูเโใไอ็ฤฦ ฦๅอ่อ้อ๊อ๋ ๆฯอ์๐๑
๒ ๒๓๔๕๖๗๘๙0 1 2 3 4 5 6 7 8 9

Figure A.4 Thai printed characters (font Eucrosia UPC, size 14, 18, 22 and 28 points)

APPENDIX B

TESTED SAMPLES

This appendix shows some samples of invariant Thai printed characters from our testing databases. This data is printed with COMPAQ Ink Jet IJ 650 Series and scanned with ACER SZW 3000U in black and white mode. The characters data used for testing by the recognizer are shown in Figure B.1-B.9.

กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยรลวศษสหฬอฮ
อัาอํอิอีอึอือุอูเโใไอ็ฤฦๅอ่อ้อ๊อ๋ๆฯอ์๐๑๒๓๔๕๖๗๘๙0123456789
กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยรลวศษสหฬอฮ
อัาอํอิอีอึอือุอูเโใไอ็ฤฦๅอ่อ้อ๊อ๋ๆฯอ์๐๑๒๓๔๕๖๗๘๙0123456789
กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยรลวศษสหฬอฮ
อัาอํอิอีอึอือุอูเโใไอ็ฤฦๅอ่อ้อ๊อ๋ๆฯอ์๐๑๒๓๔๕๖๗๘๙0123456789
กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยรลวศษสหฬอฮ
อัาอํอิอีอึอือุอูเโใไอ็ฤฦๅอ่อ้อ๊อ๋ๆฯอ์๐๑๒๓๔๕๖๗๘๙0123456789

Figure B.1 Thai printed characters (four fonts Angsana New, Browallia New, Cordia New, and Eucrosia UPC, size 14 points)

กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยรลวศษสหฬอฮ
อัาอํอิอีอึอือุอูเโใไอ็ฤฦๅอ่อ้อ๊อ๋ๆฯอ์๐๑๒๓๔๕๖๗๘๙0123456789
กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยรลวศษสหฬอฮ
อัาอํอิอีอึอือุอูเโใไอ็ฤฦๅอ่อ้อ๊อ๋ๆฯอ์๐๑๒๓๔๕๖๗๘๙0123456789
กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยรลวศษสหฬอฮ
อัาอํอิอีอึอือุอูเโใไอ็ฤฦๅอ่อ้อ๊อ๋ๆฯอ์๐๑๒๓๔๕๖๗๘๙0123456789
กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยรลวศษสหฬอฮ
อัาอํอิอีอึอือุอูเโใไอ็ฤฦๅอ่อ้อ๊อ๋ๆฯอ์๐๑๒๓๔๕๖๗๘๙0123456789

Figure B.2 Thai printed characters (four fonts Angsana New, Browallia New, Cordia New, and Eucrosia UPC, size 18 points)

กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยรลวศษ
สหฬอฮอ้าอำอิอีอึอือุอูเโใไอ็ฤฦาอ่อ้อ๊อ๋ๆฯอ์๐๑๒๓๔๕๖๗๘
๙0123456789
กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมย
รลวศษสหฬอฮอ้าอำอิอีอึอือุอูเโใไอ็ฤฦาอ่อ้อ๊อ๋ๆฯอ์๐
๑๒๓๔๕๖๗๘๙0123456789
กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยรลว
ศษสหฬอฮอ้าอำอิอีอึอือุอูเโใไอ็ฤฦาอ่อ้อ๊อ๋ๆฯอ์๐๑๒๓๔๕
๖๗๘๙0123456789
กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยร
ลวศษสหฬอฮอ้าอำอิอีอึอือุอูเโใไอ็ฤฦาอ่อ้อ๊อ๋ๆฯอ์๐๑
๒๓๔๕๖๗๘๙0123456789

Figure B.3 Thai printed characters (four fonts Angsana New, Browallia New, Cordia New, and Eucrosia UPC, size 22 points)

กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภม
ยรลวศษสหฬอฮอ้าอำอิอีอึอือุอูเโใไอ็ฤฦาอ่อ้อ๊อ๋ๆฯ
อ์๐๑๒๓๔๕๖๗๘๙0123456789
กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝ
พฟภมยรลวศษสหฬอฮอ้าอำอิอีอึอือุอูเโใไอ็ฤ
ฦาอ่อ้อ๊อ๋ๆฯอ์๐๑๒๓๔๕๖๗๘๙0123456789
กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภ
มยรลวศษสหฬอฮอ้าอำอิอีอึอือุอูเโใไอ็ฤฦาอ่อ้อ๊อ๋
ๆฯอ์๐๑๒๓๔๕๖๗๘๙0123456789
กขคฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพ
ฟภมยรลวศษสหฬอฮอ้าอำอิอีอึอือุอูเโใไอ็ฤฦ
าอ่อ้อ๊อ๋ๆฯอ์๐๑๒๓๔๕๖๗๘๙0123456789

Figure B.4 Thai printed characters (four fonts Angsana New, Browallia New, Cordia New, and Eucrosia UPC, size 28 points)

Figure B.5 Thai printed characters (four fonts Angsana New, Browallia New, Cordia New, and Eucrosia UPC, size 14 points)

Figure B.6 Thai printed characters (four fonts Angsana New, Browallia New, Cordia New, and Eucrosia UPC, size 18 points)

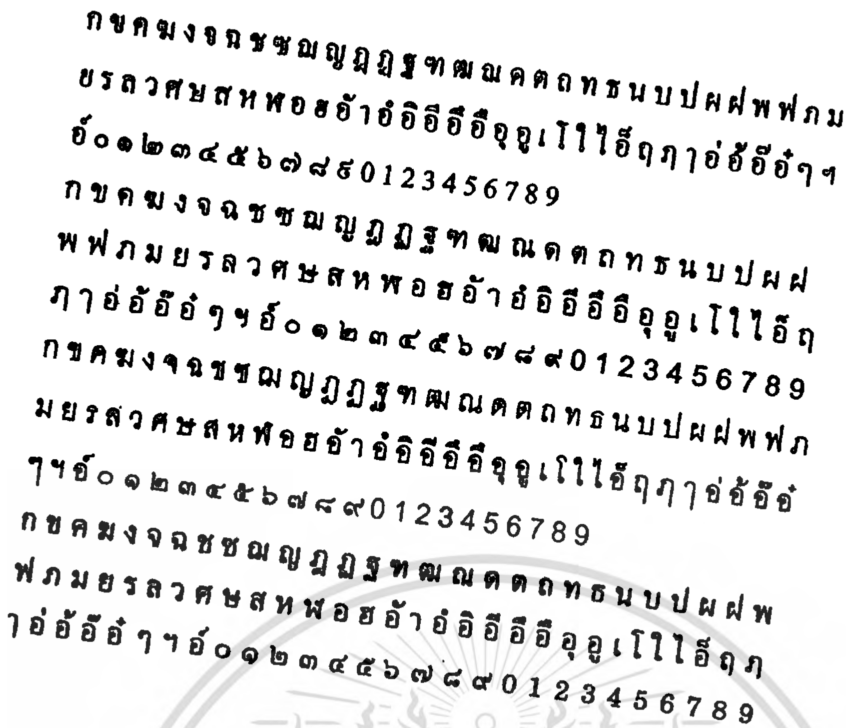


Figure B.7 Thai printed characters (four fonts Angsana New, Browallia New, Cordia New, and Eucrosia UPC, size 22 points)

Figure B.8 Thai printed characters (four fonts Angsana New, Browallia New, Cordia New, and Eucrosia UPC, size 28 points)

ไปโรงเรียน

ขนมไทย

โปรแกรม

Figure B.9 Some example of words taken from our Thai testing database

APPENDIX C

SOME COMPUTER PROGRAMMING

C.1 Ring-projection program

```
function [feature1] = ring(ox,oy,M)

%ring projection transforms 2-D patterns to 1-D pattern
%angle in radian
%radian = (2*pi/360)*degree
%M = length of maximun radius

k=1;
feature1(1,:)=0

%M is real and n is integer
n=floor(M)+1;

%(2*pi*M) is real and m is integer
m=floor(2*pi*M)+1;

for i = 0:n-1
  sumval = 0;
  for j = 0:m-1
    r = i*(M/(n-1));
    angle = j*((2*pi)/m);              %angle(radian)
    x = round(ox+r*cos(angle));
    y = round(oy+r*sin(angle));
     if   A(x,y) == 0;
      sumval=sumval+1;
     end
   end
```

```
  feature1(1,k)=sumval;
  k=k+1
end
```

C.2 Finding attributes program

```
function [newatt] = find_att(PosAtt,nn_core,oldatt,s_p,be_c)
newatt=0;
for p=(s_p+1):nn_core
  oldatt(be_c+1)=PosAtt(p);
  if (newatt==0) newatt=oldatt;
  else
    newatt=[newatt;oldatt];
  end;
end;
```

C.3 Reducing attributes program

```
function [indispensable]=reduce_attribute(reduce_all)
n=1;
nmin=0;

%for(i=1:5)
 while(n~=nmin)
    if(nmin~=0)          reduce_all=reduce_att; end;
    [n i_attribute]=size(reduce_all);
    still_n = zeros(1,i_attribute-1);

    for att=1:i_attribute-1
      if (reduce_all(1,att)~=0)
        %att=4;                                   %attribute which reduced
        canred=1;
        same_char=reduce_all;
        same_char(:,att)=zeros(n,1);
        for Chk=1:n
          for Chk2=1:n
            if(Chk~=Chk2)
              if (sum(abs(same_char(Chk,1:i_attribute-1)-same_char(Chk2,1:i_attribute- 1)))==0)
                if (same_char(Chk,i_attribute)~=same_char(Chk2,i_attribute))
                  canred=0;
                  break;
                end;
              end;
            end;
          end;
        end;
```

```
    if (canred==1)
      [reduce_att]=reduce_same(same_char);
      [red_r red_c]=size(reduce_att);
      still_n(1,att)=red_r;
    else
      still_n(1,att)=n;
    end;
  elseif (reduce_all(1,att)==0)
    still_n(1,att)=n;
  end;
end;
nmin=min(still_n);
%n_nnim=length(nmin);

if(n>nmin)
  for k=1:i_attribute-1
    if(still_n(1,k)==nmin)
      same_char=reduce_all;
      same_char(:,k)=zeros(n,1);
      [reduce_att]=reduce_same(same_char);
      break;
    end;
  end;
elseif (n==nmin)
  same_char=reduce_all;
  for k=1:i_attribute-1
    Chksame=same_char(:,k);
    LChksame=length(Chksame(Chksame==Chksame(1,1)));
    if (length(Chksame)==LChksame)
      same_char(:,k)=zeros(n,1);
      [reduce_att]=reduce_same(same_char);
```

```
        end;
      end;

    end;

  end;
  indispensable=0;
  for  att=1:i_attribute
    if (reduce_att(1,att)~=0)
      if (indispensable==0) indispensable=[att;reduce_att(:,att)];
      else indispensable=[indispensable [att;reduce_att(:,att)]];         end;
    end;
  end;

%%%%%%%%%%%%check eliminated%%%%%%%%%%%%
%     check=reduce_att;
%     [n_still i_attribute]=size(reduce_att);
%     m=1;
%     for i=1:n_still
%       if (sum(check(i,:))~=i_attribute)
%         check_same=check(i,1:(i_attribute-1));
%         collect_same(m,1)=check(i,i_attribute);
%         check(i,:)=ones(1,i_attribute);
%         n=1;
%         for j=1:n_still
%           if (sum(check(j,:))~=i_attribute)
%             inspect=check(j,1:(i_attribute-1));
%             if (sum(abs(check_same-inspect))==0)
%               n=n+1;
%               collect_same(m,n)=check(j,i_attribute);
%               check(j,:)=ones(1,i_attribute);
```

```
%            end;
%          end;
%        end;
%        m=m+1;
%      end;
%    end;
%%%%%%%%%%%%%check eliminated%%%%%%%%%%%%%
```

C.4 Computing core program

```
clear;
address2 = 'D:\Kmitl\Thesis\rough_class\feature_file4\';
result9=strcat(address2,'redatt_level2');
load(result9,'r_dec2_1','r_dec2_4','r_dec2_7','r_dec2_8');
%result4=strcat(address2,'idec_level2');
%load(result4,'i_dec2_1','i_dec2_4','i_dec2_7','i_dec2_8');
for i=1:4
  if(i==1)      indispensable=r_dec2_1;
  elseif(i==2)  indispensable=r_dec2_4;
  elseif(i==3)  indispensable=r_dec2_7;
  elseif(i==4)  indispensable=r_dec2_8;
  end;
%indispensable=[11 12 13 14; 2 15 30 102; 1 25 20 103;2 15 10 100;2 5 10 101;2 15 20
100;2 25 30 102;1 25 30 103;2 5 20 101;1 5 30 103];
[core]=findcore(indispensable);
[reduct] = findreduct(core,indispensable);
if(i==1)
  core2_1=core;
  reduct2_1=reduct;
elseif(i==2)
  core2_4=core;
  reduct2_4=reduct;
elseif(i==3)
  core2_7=core;
  reduct2_7=reduct;
elseif(i==4)
  core2_8=core;
  reduct2_8=reduct;
end;
```

```
end;
result9=strcat(address2,'core_level2');
save(result9,'core2_1','core2_4','core2_7','core2_8');
result10=strcat(address2,'reduct_level2');
save(result10,'reduct2_1','reduct2_4','reduct2_7','reduct2_8');
```

C.5 Elliptic Fourier descriptors program

```
ip=imread('t1.bmp');
[xx,yy] = size(ip);
cs=contour(ip,1);
cx=cs(1,2:length(cs)-1);
cy=cs(2,2:length(cs)-1);

% ==================== Elliptic Fourier Analysis ===================
x=cx;  y=cy;
x1 = x(1); y1 = y(1);
disp(x1); disp(y1);
xmid=0.0;
 ymid=0.0;

N=length(x);
for i=1:N-1
     dx(i) = x(i+1)-x(i);
     dy(i) = y(i+1)-y(i);
     xmid=xmid+x(i);
     ymid=ymid+y(i);
 end

 dx(N) = x(1)-x(N);
 dy(N) = y(1)-y(N);

 %   compute midpoint (centroid) of outline
 xmid=(xmid+x(N))/N;
 ymid=(ymid+y(N))/N;
```

```
% ====================================================================
% Compute elliptic fourier coefficients from delta x,y
% Reference: Kuhl, F P and C R Giardina 1982. Elliptic fourier features of a closed
contour
 tsum = 0.;
 xsum = 0.;
 ysum = 0.;
 a0=0.;
 c0=0.;
 xi = 0.;
 yi = 0.;
 l=0.;kk=0.;
 for i=1:N
     t(i) = sqrt(dx(i)*dx(i)+dy(i)*dy(i));
     l(1) = kk + sqrt(dx(i)*dx(i)+dy(i)*dy(i));
      if i>1
         l(i) = l(i-1)+ sqrt(dx(i)*dx(i)+dy(i)*dy(i));
         if i==N
             lengthperimeter=l(i);
         end
      end
      rdx(i) = dx(i)/t(i);
      rdy(i) = dy(i)/t(i);
      tnew = tsum+t(i);
       if i>1
           xi = xsum-rdx(i)*tsum;
            yi = ysum-rdy(i)*tsum;
      end
     t1 = t(i);
     t2 = tnew*tnew-tsum*tsum;
     a0 = a0+.5*rdx(i)*t2+xi*t1;
```

```
    c0 = c0+.5*rdy(i)*t2+yi*t1;
    tsum = tnew;
    xsum = xsum+dx(i);
    ysum = ysum+dy(i);
end

if tsum>0.0
    a0 = x1+a0/tsum;
    c0 = y1+c0/tsum;
end
tlen = tsum;

%-----------------------------------------------
%    loop on harmonics;
%    k = harmonics t;
k=1;
%-----------------------------------------------

for h=1:k
     fh = h;
     fact1 = fh*6.283185/tsum;
     angprv = 0.;
     asum   = 0.;
     bsum   = 0.;
      csum   = 0.;
     dsum   = 0.;
for i=1:N
     ang = angprv+t(i)*fact1;
     wtac = cos(ang)-cos(angprv);
     wtbd = sin(ang)-sin(angprv);
     angprv = ang;
```

```
    asum = asum+rdx(i)*wtac;
    bsum = bsum+rdx(i)*wtbd;
    csum = csum+rdy(i)*wtac;
    dsum = dsum+rdy(i)*wtbd;
    continue;
 end
 fact2 = fact1*fh*3.141596;
 aa(h) = asum/fact2;
 bb(h) = bsum/fact2;
 cc(h) = csum/fact2;
 dd(h) = dsum/fact2;
 continue;
end

%=================================================================
%   Normalizes k elliptic fourier coefficients in aa,bb,cc & dd.
%   The normalization quantities used are returned in size, rotate
%   and start.  To normalize for object location, ignore xmid,ymid
%   returned by XY2DEL.  To normalize
%   for trace orientation, compute coefficients only for counter-
%   clockwise traces
%   Reference:  Kuhl, F P and C R Giardina 1982. Elliptic fourier
%   features of a closed contour. Computer graphics and image

a=aa(1);
b=bb(1);
c=cc(1);
d=dd(1);
denom=a*a+c*c-b*b-d*d;
 theta=0.5*atan2(2.*(a*b+c*d),denom);
r=cos(theta);
```

```
s=sin(theta);
astar=r*a+s*b;
cstar=s*d+r*c;
psi=atan2(cstar,astar);
estar=sqrt(astar*astar+cstar*cstar);
if(estar == 0.0)
    estar=1.0;
end
r=cos(psi);
s=sin(psi);
a0n = (r*a0+s*c0)/estar;
c0n = (-s*a0+r*c0)/estar;
for n=1:k
    thetan=theta*(n);
    rn=cos(thetan);
    sn=sin(thetan);
    one=aa(n)*r+cc(n)*s;
    two=bb(n)*r+dd(n)*s;
    thr=cc(n)*r-aa(n)*s;
    four=dd(n)*r-bb(n)*s;
    aa(n)=(rn*one+sn*two)/estar;
    bb(n)=(rn*two-sn*one)/estar;
    cc(n)=(rn*thr+sn*four)/estar;
    dd(n)=(rn*four-sn*thr)/estar;
    size=estar;
    rotate=psi;
    a0 = a0n;
    c0 = c0n;
end

twopi = 6.283185;
```

```
n=N;
for i=1:n+1
  xi = a0;
  yi = c0;
  delang = twopi*(i-1.)/n;
  ang = delang;
  for h=1:k
     xi  = xi+aa(h)*cos(ang)+bb(h)*sin(ang);
     yi  = yi+cc(h)*cos(ang)+dd(h)*sin(ang);
     ang = ang+delang;
     continue; */}
  end
  Xrs(i) = xi;
  Yrs(i) = yi;
  continue;
end

%*****************************************************
% X only
Rx=Xrs;
% Size Normalization to scaling is performed in and the
% three step:
% Compute Xm and XM , the minimum and the maximum of Xrs
Rxmin=min(Rx);
Rxmax=max(Rx);
% Rescale Xrs(I) in Xscal
for i=1:n
  Rxscal(i)=(Rx(i)-Rxmin)/(Rxmax-Rxmin);
end
```

```
% Resample Xscal into Xt using interpolate
Rxsamp=interpft(Rxscal,300);
Rx=Rxsamp';
% Rsamp
%***************************************************
%***************************************************
%  Y only
Ry=Yrs;
% Size Normalization to scaling is performed in and the
% three step:
% Compute Xm and XM , the minimum and the maximum of Xrs
Rymin=min(Ry);
Rymax=max(Ry);
% Rescale Xrs(I) in Xscal
for i=1:n
  Ryscal(i)=(Ry(i)-Rymin)/(Rymax-Rymin);
end

% Resample Xscal into Xt using interpolate
Rysamp=interpft(Ryscal,300);
Ry=Rysamp';
 % Rsamp
%***************************************************
% catesian to polar
Rs=sqrt(Xrs.^2+Yrs.^2);
% Size Normalization to scaling is performed in and the
% three step:
% Compute Xm and XM , the minimum and the maximum of Xrs
Rsmin=min(Rs);
Rsmax=max(Rs);
% Rescale Xrs(I) in Xscal
```

```
for i=1:n
  Rscal(i)=(Rs(i)-Rsmin)/(Rsmax-Rsmin);
end

% Resample Xscal into Xt using interpolate
        Rsamp=interpft(Rscal,300);
        Rxy=Rsamp';
  % Rsamp
  figure,plot(Xrs,Yrs);
%****************************************************
% Y only
% Size Normalization to scaling is performed in and the
% three step:
% Compute Xm and XM , the minimum and the maximum of Xrs
  lymin=min(I);
  lymax=max(I);
% Rescale Xrs(I) in Xscal
        for i=1:N-1
    lyscal(i)=(I(i)-lymin)/(lymax-lymin);
  end

% Resample Xscal into Xt using interpolate
        lysamp=interpft(lyscal,300);
        ly=lysamp';
  % Rsamp
%****************************************************
% Normalized Rotation
%           disp(I);
  delta=theta*lengthperimeter/6.2832;
  mx=max(x);
  if abs(mx-x(1))<=3
```

```
if psi<=2.64
  lp1=lengthperimeter-delta;
  lp2=lengthperimeter+delta;
  disp(delta); disp(psi);
else
  lp1=abs(delta);
  lp2=abs(delta);
  disp(delta); disp(psi);
end
 If psi<0
  if theta<0
   for i=1:N
    if ((l(i)<=lp2)&(lp2<l(i+1)))
     if i<N/2
       jk=i+N/2;
     else
       jk=i-N/2
     end
     starting=l(i); xre=x(i); yre=y(i);
     disp(starting);
     disp(xre); disp(yre);
     break;
    end
   end
  end
  if theta>=0
   for i=1:N
    if ((l(i)<=lp1)&(lp1<l(i+1)))
     if i<N/2
       jk=i+N/2;
     else
```

```
        jk=i-N/2
      end
      starting=l(i); xre=x(i); yre=y(i);
      disp(starting);
      disp(xre); disp(yre);
      break;
    end
  end
 end
end
if psi>=0
 if theta<0
  for i=1:N
   if ((l(i)<=lp2)&(lp2<l(i+1)))
    if i<N/2
      jk=i+N/2;
    else
      jk=i-N/2
    end
    starting=l(i); xre=x(i); yre=y(i);
    disp(starting);
    disp(xre); disp(yre);
    break;
   end
  end
 end
 if theta>=0
  for i=1:N
   if ((l(i)<=lp1)&(lp1<l(i+1)))
    if i<N/2
      jk=i+N/2;
```

```
        else
          jk=i-N/2
        end
        starting=l(i); xre=x(i); yre=y(i);
        disp(starting);
        disp(xre); disp(yre);
        break;
      end
    end
  end
 end
end
if abs(mx-x(1))>3
 lp1=lengthperimeter-delta;
 lp2=lengthperimeter+delta;
 disp(delta); disp(psi);
 if psi<0
  if theta<0
   for i=1:N
    if ((l(i)<=lp2)&(lp2<l(i+1)))
      jk=i;
      starting=l(i); xre=x(i); yre=y(i);
      disp(starting);
      disp(xre); disp(yre);
      break;
    end
   end
  end
  if theta>=0
   for i=1:N
    if ((l(i)<=lp1)&(lp1<l(i+1)))
```

```
      jk=i;
      starting=l(i); xre=x(i); yre=y(i);
      disp(starting);
      disp(xre); disp(yre);
      break;
     end
    end
   end
end
if psi>=0
 if theta<0
  for i=1:N
   if ((l(i)<=lp2)&(lp2<l(i+1)))
      jk=i;
      starting=l(i); xre=x(i); yre=y(i);
      disp(starting);
      disp(xre); disp(yre);
      break;
     end
    end
   end
 if theta>=0
  for i=1:N
   if ((l(i)<=lp1)&(lp1<l(i+1)))
      jk=i;
      starting=l(i); xre=x(i); yre=y(i);
      disp(starting);
      disp(xre); disp(yre);
      break;
     end
    end
```

```
    end
   end
 end
 jkk=N-jk;
 for i=1:N
   if i>=jk
     xx(i-jk+1)=x(i);
     yy(i-jk+1)=y(i);
   else
     xx(jkk+1+i)=x(i);
     yy(jkk+1+i)=y(i);
   end
 end
 kk=0;
 for i=1:N
   ll(1) = kk + sqrt((xx(2)-xx(1))*(xx(2)-xx(1))+(yy(2)-yy(1))*(yy(2)-yy(1)));
   if (i>1)&(i<N)
     ll(i) = ll(i-1)+ sqrt((xx(i+1)-xx(i))*(xx(i+1)-xx(i))+(yy(i+1)-yy(i))*(yy(i+1)-yy(i)));
   end
 end
 ll(N) = ll(N-1)+sqrt((xx(1)-xx(N))*(xx(1)-xx(N))+(yy(1)-yy(N))*(yy(1)-yy(N)));
 for i=1:N
   xri(i)=xx(i)*cos(psi)+yy(i)*sin(psi);
   yri(i)=-xx(i)*sin(psi)+yy(i)*cos(psi);
 end
 disp(l(N));
 figure,plot(l,x);
 figure,plot(ll,xx);
%   figure,plot(xri,yri);
```

C.6 Fourier descriptors program

```
void test::fourier()
{
    complex suma(0,0),sumb(0,0),dum(0,0);
    int m,k;
     #define pi 3.14159
   for(k=0;k<=31;k++)
      {
       for(m=1;m<=l;m++)
         {
            complex a(0,-(k*(2*pi/l)*m));
            aa[m]=(x[m-1])*(exp(a));
             suma=suma+aa[m];
             aa[m]=(y[m-1])*(exp(a));
             sumb=sumb+aa[m];
         }
          ak[k]=suma/l;
          bk[k]=sumb/l;
          suma=dum;
          sumb=dum;
          absa=abs(ak[k]);
          absa=absa*absa;
         absb=abs(bk[k]);
         absb=absb*absb;
         rr[k]=sqrt(absa+absb);
         s[k]=rr[k]/rr[0];
         //cout << "s" << (k+1) <<"="<< s[k]<<". ";
      }
      //cin >> filename;
      //getch();
      //clrscr();
}
```

C.7 Contour-following program

```
void test::contour()
{
  //find bottom-left conner
  int blx,bly,i=width,j=height;
  char bl='n';
  do
  {
     if(data[j][i]==2)
     {
        blx=i;
        bl='y';
      }
      i=i-1;
      if(i==0)
      {
         j=j-1;
         i=width;
       }
  }while(bl!='y');
  bly=(height-j);
  int z=0;
  i=0;
  j=0;

  //find first point
  do
  {
     if(data[j][i]==2)
```

```
    {
        x[0]=i;
        z=1;
    }
    i=i+1;
    if(i==width)
    {
        j=j+1;
        i=0;
    }
}while(z!=1);

y[0]=(height-j);
i=i-1;
//find second point

if(data[j][i-1]==2)
{
  x[z]=i-1;
  y[z]=height-j;
}
else if(data[j+1][i-1]==2)
{
  x[z]=i-1;
  y[z]=height-(j+1);
}
else if(data[j+1][i]==2)
{
    x[z]=i;
  y[z]=height-(j+1);
}
```

```
else if(data[j+1][i+1]==2)
{
   x[z]=i;
  y[z]=height-(j+1);
}
else if(data[j][i+1]==2)
{
  x[z]=i;
  y[z]=height-j;
}
//cout << x[0]<<","<<y[0]<<" "<<x[1]<<","<<y[1];
//getch();
//begin contour at now z=1
int x0,x1,y0,y1;
char blpass='n';
l=2;
do
{
   char check='d';
   x1=x[z-1];//32
   x0=x[z];  //31
    y1=y[z-1];//16
    y0=y[z];//17
   do
   {
       if(y0<y1&&check!='q')//moveleft
       {
           while(x0<=x1&&check!='q')
           {
              x1=x1-1;
             if(data[height-y1][x1]==2)check='q';
```

```
        }
    }
    if(x0>x1&&check!='q')//movedown
    {
        while(y0<=y1&&check!='q')
         {
            y1=y1-1;
             if(data[height-y1][x1]==2)check='q';
         }
    }
    if(y0>y1&&check!='q')//moveright
    {
     while(x0>=x1&&check!='q')
      {
       x1=x1+1;
        if(data[height-y1][x1]==2)check='q';
      }
 }
 if(x0<x1&&check!='q')//upward
 {
     while(y0>=y1&&check!='q')
     {
       y1=y1+1;
        if(data[height-y1][x1]==2)check='q';
     }
   }
}while(check!='q');
z=z+1;
x[z]=x1;
y[z]=y1;
 l=l+1;
```

```
    //cout<<x1<<","<<y1<<" ";
    //getch();
     if(x[z]==blx&&y[z]==bly)blpass='y';
}while((x[z]!=x[0]||y[z]!=y[0])||blpass!='y');

 for(int i=0;i<=l-1;i++)y[i]=height-y[i];
    for(int i=0;i<=l-1;i++)data[y[i]][x[i]]=0;
       int xt,xb,yt,yb;
       xt=x[0];
       xb=x[0];
       yt=y[0];
       yb=y[0];
        for(int i=1;i<l;i++)
        {
            if(xt>x[i])xt=x[i];
            if(xb<x[i])xb=x[i];
            if(yt>y[i])yt=y[i];
            if(yb<y[i])yb=y[i];
        }
        for(int j=yt;j<yb;j++)
           for(int i=xt;i<xb;i++)
              data[j][i]=0;
}
```

C.8 Back-propagation neural networks program

```
#include<iostream.h>
#include<stdlib.h>
#include<math.h>
#include<conio.h>
#include<stdio.h>
#include<alloc.h>

double in[90][31];
double act_h1[90][65];
double out_h1[90][65];
double act_O[90][4];
double out_O[90][4];
double Tout[90][4];
double CR[90][4];
double cr1[90][65];
double w1[65][31];
double W[4][65];
double Dw1[65][31];
double DW[4][65];
double E,Ep[90],Etot=1.0,Serror=1.0;
double sig(double);

void main()
{
  randomize();
  for(int i=0;i<90;i++)
  for(int j=0;j<4;j++)Tout[i][j]=0;
  for(int i=42;i<62;i++)Tout[i][3]=1;
  for(int i=62;i<82;i++)Tout[i][2]=1;
```

```
long int i;
int rule,input,hidden,output;
for(hidden=0;hidden<65;hidden++)
{
  for(input=0;input<31;input++)
  {
    w1[hidden][input] = 9-random(301)/10.0;
  }
}
for(output=0;output<4;output++)
{
  for(hidden=0;hidden<65;hidden++)
  {
    W[output][hidden]= 1-random(201)/100;
  }
}
/* End the random weight process (w,W) */
/* Receive value of beta and iterations */
double beta,neta;
long int iters;
cout<<"What's correction factor? ="; cin>>beta;
cout<<"How many iterations? ="; cin>>iters;
cout<<"How many learn rate? ="; cin>>neta;
FILE  *read;
int count=0;
double S[32];
read=fopen("Charact.dat","rb");
while(!feof(read))
{
  fread(&S,256,1,read);
```

```
 for(int i=1;i<32;i++)in[count][i-1]=S[i];
 count++;
}
fclose(read);
cout << "There is "<<(count-1) <<" Characters."<<endl;
cout <<"How many Characters do you want to learn? ";cin>>count;
count++;
float maxerror =0.012;
/*END SET INPUT AND TARGET OUTPUT*/
for(i=0;(i<Iters)&&(Etot>maxerror);i++)
{
 if(kbhit())
 {
  clrscr();
  cout << "Add 2,000 iters.."<<endl<<"Please any keys..";
  getch();
        iters += 2000;
  getch();
 }
 cout<<"Loop == "<<i<<endl;
  /* Compute output of each hidden node */
 for(rule=0;rule<(count-1);rule++)
 {
   for(hidden=0;hidden<65;hidden++)
   {
     act_h1[rule][hidden]=0;
     for(input=0;input<31;input++)
     {
       act_h1[rule][hidden] += w1[hidden][input]*in[rule][input];
     }
     out_h1[rule][hidden] = sig(act_h1[rule][hidden]);
```

```
  }
}
  /* Compute output of each output node */
for(rule=0;rule<(count-1);rule++)
{
 for(output=0;output<4;output++)
 {
  act_O[rule][output]=0;
  for(hidden=0;hidden<65;hidden++)
  {
    act_O[rule][output] += W[output][hidden]*out_h1[rule][hidden];
  }
  out_O[rule][output] = sig(act_O[rule][output]);
 }
}
  /* Compute Error E */
double E=0.0;
double Sum_of_error[90];
for(rule=0;rule<(count-1);rule++)
{
  Sum_of_error[rule]=0;
  for(output=0;output<4;output++)
  {
    Sum_of_error[rule] += ((Tout[rule][output]-out_O[rule][output])*
                    (Tout[rule][output]-out_O[rule][output]));
  }
  Ep[rule] = Sum_of_error[rule]/2.0;
  E += Ep[rule];
}
Etot = E/((double)count-1.0);
cout << Etot<<endl;
```

```
if(Etot>maxerror)
{
  for(rule=0;rule<(count-1);rule++)
  {
    for(output=0;output<4;output++)
    {
      CR[rule][output] = beta*(Tout[rule][output]-out_O[rule][output])*
                   (1-out_O[rule][output])*out_O[rule][output];
      for(hidden=0;hidden<65;hidden++)
      {
        DW[output][hidden] = neta*CR[rule][output]*out_h1[rule][hidden];
        W[output][hidden] = W[output][hidden]+DW[output][hidden];
      }
    }
  }
/* Compute new weights for input to hidden1 connection */
  double add1[90];
  for(rule=0;rule<(count-1);rule++)
  {
    add1[rule]=0;
    for(output=0;output<4;output++)
    {
      for(hidden=0;hidden<65;hidden++)
      {
        add1[rule] += CR[rule][output]* W[output][hidden];
        cr1[rule][hidden] = add1[rule]*beta*(1-out_h1[rule][hidden])
                             *out_h1[rule][hidden];
        for(input=0;input<31;input++)
        {
          Dw1[hidden][input] = neta*cr1[rule][hidden]*in[rule][input];
          w1[hidden][input] = w1[hidden][input]+Dw1[hidden][input];
```

```
          }
        }
      }
    }
  }
}/* End loop */
cout<<"No. of loop=="<<i<<endl;
cout << Etot<<endl;
getch();
/* Show all using weights *//*
for(int I=0;I<65;I++)
{
  for(int J=0;J<32;J++)
  {
    cout<<"weight1"<<I<<","<<J<<"=="<<w1[I][J]<<endl;
  }
  getch();
}
//cout<<"My name is Warawat"<<endl;
//getch();
for(int I=0;I<4;I++)
{
  for(int J=0;J<65;J++)
  {
    cout<<"WEIGHT"<<I<<","<<J<<"=="<<W[I][J]<<endl;
    getch();
  }
}*/
for(rule=0;rule<(count-1);rule++)
{
  cout<<"Rule"<<(rule+1)<<endl;
```

```
    for(output=0;output<4;output++)
    {
      cout<<"It's Tout"<<(output+1)<<"=="<<Tout[rule][output]<<";;";
      cout<<"It's output"<<(output+1)<<"=="<<out_O[rule][output]<<endl;
    }
    getch();
  }
  char check='n';
  cout << "Do you want to save file?";cin >> check;
  if(check=='y')
  {
    FILE *save_file;
    save_file=fopen("Weight.Dat","ab");
    fwrite(&W,sizeof(W),1,save_file);
    fwrite(&w1,sizeof(w1),1,save_file);
    fclose(save_file);
  }
} /* End Main */
double sig(double x)
{
  double y;
  y = 1.0/(1.0+exp(-x));
  return(y);
} /* End Function Sig */
```

APPENDIX D

LISTS OF NATIONAL PAPERS

1 Phokharatkul P. and Kimpan C. "Printed Thai Characters Recognition Using Quasi-topological Codes"., Proceedings of the 19th Electrical Conference, Nov. 7-8, 1996. pp.DS-17 – DS-22.

2 Phokharatkul P. and Kimpan C. "Printed Thai Characters Recognition Using Hough Transform Method"., Ladkrabang Engineering Journal, Vol. 13 No. 2: April, 1997. pp.20-27.

3 Phokharatkul P., Kumnuenchai S. and Kimpan C. "Printed Thai Characters Recognition Using Fuzzy Similarity Measurement Method"., Proceedings of the 20th Electrical Conference, Nov. 13-14, 1997. pp.349– 354.

APPENDIX E

LISTS OF INTERNATIONAL PAPERS

1 Phokharatkul P. and Kimpan C. "Recognition of Handprinted Thai Characters Using the Cavity Features of Character Based on Neural Network". Proceedings The 1998 IEEE Asia-Pacific Conference on Circuits and Systems, Nov. 24-27,1998. pp.149-152.

2 Marang U., Phokharatkul P., and Kimpan C. "Recognition of Printed Thai Characters using Boundary Normalization and Fuzzy Neural Networks.", The Fourth Symposium on Natural Language Processing SNLP 2000, May 10-12, 2000. pp.154-164.

3 Phokharatkul P., Marang U. and Kimpan C. "Recognition of Printed Thai Characters Elliptic Fourier Descriptors and Genetic Neural Networks.", Proceedings 2000 Asia-Pacific Symposium on Broadcasting and Communications APSBC 2000, Dec. 21-23, 2000. pp.406-411.

4 Phokharatkul P., Kimpan C. and Sato M. "A Rough Ring Projection Method for Invariant Thai Character Recognition.", World Multiconference on Systemics, Cybernetics and Informatics SCI/ISAS 2001, Orlando, USA: July 22-25, 2001. pp. 66-70.

5 Phokharatkul P. and Kimpan C. "Handwritten Thai Character Recognition Using Fourier Descriptors and Genetic Neural Networks.", Computational Intelligence An International Journal (accepted for publication).

Recognition of Handprinted Thai Characters Using the Cavity Features of Character Based on Neural Network

Pisit Phokharatkul and Chom Kimpan
Faculty of Engineering, Research Center for Communications and Information Technology
King Mongkut's Institute of Technology Chaokhun Taharn Ladkrabang, Bangkok 10520, Thailand.
Tel. 662-3269969 ext.112, Fax. ext. 127, Email s0060004@kmitl.ac.th, kkchom@kmitl.ac.th.

Abstract

This paper describes a method of cavity features and neural network for recognizing handprinted Thai characters. The recognition process is implemented using mathematical morphology to detect the cavity features of patterns, and learning to classify by neural network. The stage of recognition divided into three stages. First, the handprinted Thai characters are segmented from the sentence into three different level groups. Then, the cavity features of each handprinted Thai character are detected, and counted the numbers by the Euler number method. Finally, uses the majority area of the cavity features for computed the feature codes of the characters in each class. These codes are trained by neural network for learning in the classification characters.

1. Introduction

The developmental of the method of recognition of the Thai characters at the present time by using different methods can be computerize into different information such as transform pictures format to text mode, and go into the form of text file, and be able to use with the word processor or automatic apply to the different duties. There are several experimental researches written by the researchers. [1-4]

This research will present the method of recognition of handprinted Thai characters by searching for the cavity features of each character based on neural networks for solving the problem in the recognition by considering the head of the handwritten Thai characters. [4] The method of finding the cavity features can solve the problem of the head of the Thai characters in which will be illustrate in the following section.

2. The cavity features of handprinted Thai characters

The nation of cavities involves the spaces that surround the actual character stroke. Figure 1 illustrated the cavity features. A *cavity* [5] is a region of points bounded by the stroke on at least three sides. The cavities are named by the direction in which they open, that is, the side on which they are not bounded. A west cavity is a region open to the west but not open to the north, south, or east. A hole is a region completely enclosed by the stroke. A center cavity is a region that is surrounded on all four sides, but is not a hole. There are six cavity feature types: east, west, north, south, center, and hole. The cavity features are computed using binary mathematical morphology. The language of binary mathematical morphology is that of set theory. Sets in mathematical morphology represent the shapes that are manifested on binary or gray scale images. The set of all the black pixels in a black and white image (a binary image) constitutes a complete description of the binary image. The morphological operation is used in this research is dilation.[6]

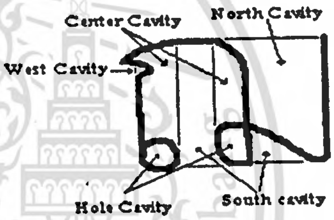


Fig. 1 Cavity features of the handprinted Thai character "ฌ".

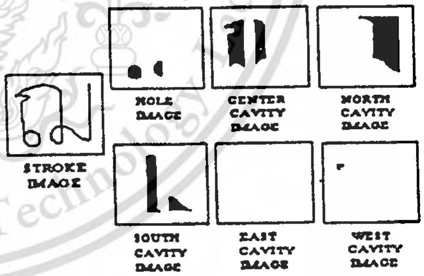


Fig. 2. Feature images used for cavity based zone features.

Let *N*, *S*, *E*, and W denote structuring elements that are rays in the directions north, south, east, and west. These structuring elements are depicted in Fig. 3. The feature images are denoted *I*, *HF*, *CF*, *NF*, *SF*, *EF*, and *WF* in the order given in Fig. 2. In particular, *I* denotes the image of the character itself. Furthermore. let *B* denote the background, or complement, of the image *I*. The feature images are computed according to the following morphological algorithm:

$$NF = I \oplus N \cap \overline{I \oplus S} \cap I \oplus E \cap I \ominus W \cap B$$

$$SF = \overline{I \oplus N} \cap I \oplus S \cap I \oplus E \cap I \ominus W \cap B$$

0-7803-5146-0/98/$10.00 ©1998 IEEE.

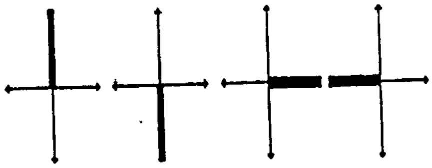


Fig. 3. N, S, E, and W structuring elements used to compute the cavity feature images.

$$EF = I\oplus N \cap I\oplus S \cap I\oplus E \cap \overline{I\oplus W} \cap B$$
$$WF = I\oplus N \cap I\oplus S \cap \overline{I\oplus E} \cap I\oplus W \cap B$$
$$CF = \underline{I\oplus N \cap I\oplus S \cap I\oplus E \cap I\oplus W \cap B}$$
$$HF = \text{span-until (border, B,T)} \cup I$$

Feature vectors can be computed as follows: The image of the character is normalized to the size 64x64 pixels. The feature images are computed from morphological algorithm and count the number of the cavity features in the character image by Euler number method.[7]

3. Segmentation

The Thai language sentences are composed of consonants, vowels and controlled voice tones, on different levels. The segmentation of handprinted Thai characters from the sentence into three different level groups uses the method in reference [1].

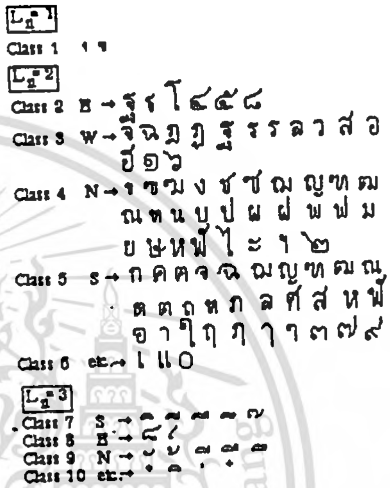
4. Classification

The classification stage consist of two sub stages that are the classification by the cavity features and the classification by neural network. The detail classification will be described in the titles of 4.1 and 4.2 respectively.

4.1 The classification by the cavity features

After the handprinted Thai characters were segmented from each sentence using the different levels into three different level groups, then each handprinted Thai character image is divided into four sub regions by using the center of gravity and detected the cavity features of handprinted Thai characters, and computed the number of cavity feature types. So that the feature vectors composed of the cavity image in each region (H_1... H_4 , C_1... C_4 , N_1...N_4 ,S_1... S_4 , E_1... E_4 and W_1... W_2), the number of the cavity feature types (N_H, N_C, N_N, N_S and N_W) and different levels L_N in which the number of cavity features and different levels represented three bits, and two bits respectively. The example feature codes as shown in Fig. 4.

From the cavity features of the handprinted Thai characters detected the area of the cavity features in each character, and classified these characters according to the area of cavity features of *east*, *west*, *north*, and *south* such as the handprinted Thai characters " ก, ถ, ภ, " were classify in the class of cavity feature of south according to the majority area of the cavity features are south, but " บ, ป, น, ม, ง " the majority area of cavity features were north type. In some of the cases one of the characters may classify as north or south type, in which depend upon the way of writing such as " ฌ, ฒ, ฑ, ห ". From classifying the majority of the area of cavity features of the characters in each group may classify into classes as follows:-



4.2 The classification by neural network

In this sub stage a neural network was utilized to classify handprinted Thai characters by bringing the feature codes of the characters in each class in the title of 4.1 trained by back propagation neural network for learning in classifying of the characters form the group in each class is shown in Fig. 9 [8].

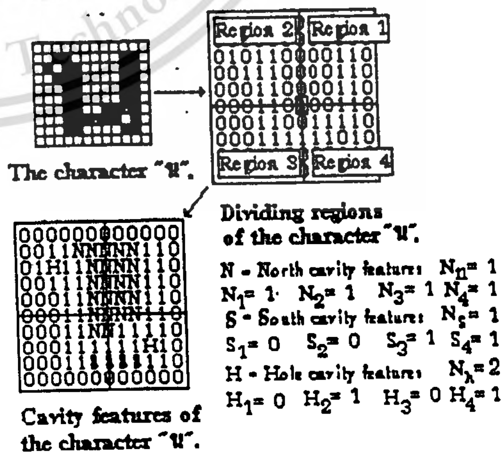


Fig. 4. Dividing regions of Character and the cavity features of Thai character "ม".

5. Heads or loops

The Thai characters characteristics is always written by the head first and then the strokes is written connected from the head. The heads or loops are important to classify by the method in reference [4]. If the heads or loops incomplete the method in reference [4] is fail but in this research do not the problem, and there are the capability to detect the cavity features of these defect head characters as shown in Fig. 7.

6. The special techniques

After training for the classification by "neural network". Some of the Thai characters are similar, and can not be classify any more. So that the researchers found the special techniques for the recognition such as ก ค there are similar, the only difference is that the head of the characters in which lead to the problem of recognition. Any of the methods of recognition is ก there is outer curl, but ค there is inner curl. The experimental of the head of the characters can be scan at the right edge and lowest edge of center cavity feature when expecting the position of the right edge, and the lowest edge, then scan in the horizontal from left to right for finding border of the left center cavity feature as in the fig. 5 a), then can get the segment of the characteristic of the head as in the fig. 5 b). After that will scan specific segment with in the region as in the fig. 5 c), when classified the segment as needed. Finally scanning the specific segment then can get the cavity features as in fig. 6 d) and 6 h), then the researchers can classify the characters that are similar refer to the broken head as shown in fig. 7 d) and 7 h).

In the case of the Thai characters of the similar feature codes ข and บ, then will prove on the width and the height as in the Fig. 8 a), บ and ป, พ and ฟ, ผ and ฝ then will prove on the distance of L and R as in the fig. 8 b), but the characters of and use the cross point can be able to classify as in the fig. 8 c).

7. The experimental results

The handprinted Thai characters set composes of three vertical level groups and totally ten classes. The character image in each class are dividing into 4 regions using of the gravity center and computed the cavity feature vectors, and counted the number of the cavity feature types. The different levels, the number of cavity feature types, and the feature codes of the handprinted Thai characters in each class are inputs into multilayer feedforward neural networks trained

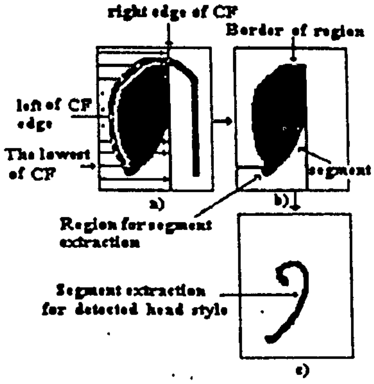


Fig. 5 The process for extracted the specific segment of similar characters.

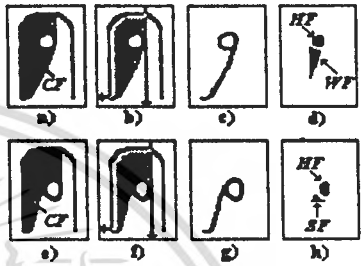


Fig 6 The classification of the similar Thai characters "ค and ก" by the detection of cavity features in the head segment

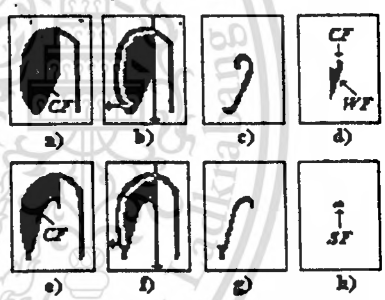


Fig. 7 The classification of the similar Thai characters "ค and ก" by the detection of cavity features in the head region for the case of broken head.

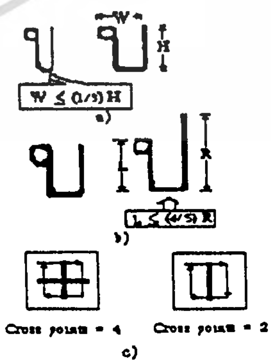


Fig 8.The classification of the Thai characters for the case of the similar feature codes.

using back propagation. The neural networks perform classification and confidence assignment. The example feature codes of handprinted Thai characters are inputs into neural network in the training stage for learning in classifying of the characters as shown in Fig. 9.

In the training stage, the characters in each class has 20 different training patterns, giving a maximum of characters are 600 training patterns for class 5, and a minimum of characters are 40 training patterns for class 1 or 8 in one training cycle. The training patterns of the handprinted Thai characters are written in normal shapes. The experimental results are shown in table 1 as average recognition rate of each different level groups. The comparison of recognition rate of cavity features with other methods are shown in table 2.

Table 1 Results of testing the handprinted Thai character Neural Network Classifiers on the 3,200 characters.

Different level groups	Classes	Recognition rate (%)
1	1	100.00
2	2-6	98.14
3	7-10	96.80
Total recognition rate		98.31

Table 2 The comparison of the recognition rate of the cavity features with other methods.

Recognition methods	Percent correct (%)
Local features[3]	88.9
Head consideration[4]	98.2
Cavity features	98.3

8. Conclusions

This particular paper present a cavity features of character base on neural network for handprinted Thai characters recognition system. The features from the character image data which do not make thinning process and edge detection, then the character image processing time is reduced. The cavity features are inputs into multilayer feedforward neural networks trained using back propagation for classification. The experimental results prove the usefulness of this method with an actual recognition rate of 98.3 percents for 3,200 handprinted Thai characters (80 characters for each copy) from 40 persons.

References

[1] Kimpan C., Itoh A. and Kawanishi K., "Recognition of Printed Thai Characters Using a Matching Method" IEE Proceedings. Vol. 130, Pt.E, No 6, pp. 183-188, November 1983.

[2] Kimpan C., "Printed Thai Characters Recognition Using Topological Properties Method.", INT. J. Electronics. Vol. 60, No.3, pp..303-329, 1986.

[3] Hiranvanichakron P., Agui T. and Nakajima M. " A Recognition Method of Handprinted Thai Characters by Local Features.", The Transaction of the IECE of Japan., vol. E 68, No. 2, pp.83-90, Febuary 1985.

[4] Airphaiboon S., Sangworasil M., Attchoo B., Kimpan C.:" Recognition of Handwritten Thai Character Considering the Head of Character.", IEEE International Conference on Image Processing ICIP'89, pp.326-330, Singapore., 5-8 September 1989.

[5] Dougherty E.R.: Digital Image Processing Methods., Marcel Dekker INC., New York., 1994.

[6] Serra J., Image Analysis and Mathematical Morphology Volume 1, Academic Press. Limited., 1982.

[7] Batchelor B.G.: Intelligent Image Processing in Prolog., Springer-Verlag London Limited., 1991.

[8] Hagan M.T., Demuth H.B., Beale M. Neural Network Design., PWS Publishing Company., 1996.

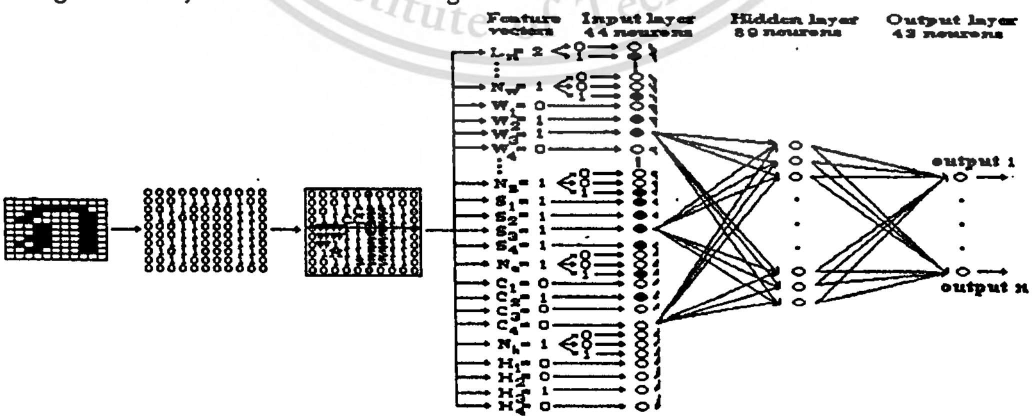


Fig. 9. The example feature codes of the Thai character "ก" for training by neural network.

Recognition of Printed Thai Characters Using Boundary Normalization and Fuzzy Neural Networks

Ukrit Marang, Pisit Phokharatkul, and Chom Kimpan

Faculty of Engineering, Research Center for Communications and Information Technology
King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand.
Tel. (662) 3269969 ext. 125, Fax. (662) 7392400
Email. s0106110@kmitl.ac.th, s0060004@kmitl.ac.th, kkchom@kmitl.ac.th

Abstract

A new approach for recognizing Thai Characters is presented. We employ various methods i.e. Boundary Normalization, Fuzzy Theory and Back-propagation Neural Networks to detecting boundary functions of characters, and attempt to normalize these functions by orientation and size variations of the symbols. Here, Fuzzy logic is used to model the uncertainties in the relationships between the variables of boundary normalization of the characters. Finally, use of feed forward neural network to learn fuzzy rules base. A neural network can implement which this means fuzzy IF-THEN rules. The experimental results shows the recognition rate are high though only a limited number of features have been involved. The result of this approach indicates that the systems can successfully recognized 1600 Thai characters with 98.5% accuracy.

1. Introduction

Many researches for transforming characters images to text in Thai language have been studied for many years. The researchers have attempted to propose several methods to implement in computers and then apply them in various areas. The details of these approaches are illustrated in [1-14]. Although the accuracy rate of the exists systems are accepted, but they have limitations on size and orientation variations. Our approach describes the integration methods among boundary normalization, fuzzy set theory and artificial neural network. Two issues may arise in discussion with the shape of character that are orientation or slant and size variation, which this problem can be solved by finding the boundary functions of the characters. The principle in this research is based on the method developed by Kuhl and Gardina [4], which is efficiently, applied even when the heads of the characters were defected. We continually employ fuzzy modeling technique to extract the essential features and then feed these sets of fuzzy rules into the back propagation neural network, for training and recognition. The paper is organized as follows; Section 2-4 briefly reviews the preprocessing method and the feature extraction. The construction of a Thai word as shown in Fig 5. Those are separated by the elliptic Fourier descriptors. The features of elliptic Fourier descriptors are extracted and fed to Fuzzy-neural system. In section 5-8, we show the application of this algorithm in our work. The application of using fuzzy theory and BPNN are described in section 6 and 7 respectively. Finally, the experiments and their results are in section 8.

2. Thai characters and experimental data

Thai characters, as shown in Fig.1, consists of 42 consonants, 17 vowels, 7 voice tones, 10 Thai numerals and 10 Arabic numerals. A Thai language sentence, such as that in Fig.2, is composed of consonants, vowels and voice tones, on different levels. The vertical level can be divided into four parts as shown in Fig.3. However, as can be seen by inspection, on occasion, depending on the printing machine (see Fig.4), a vowel can be situated above two consonants.

Fig. 1 that character set

สำนักวิจัยการสื่อสารและเทคโนโลยีสารสนเทศ

Fig.2 Example of Thai language sentence

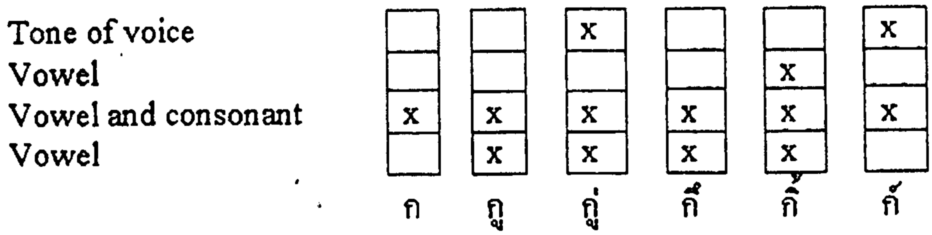


Fig.3 Composition of Thai character word or sentence

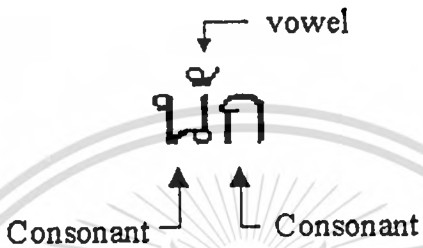


Fig 4 Example of Thai word vowel is situated above both consonants.

3. Segmentation

Because the construction of a Thai word, as shown in Fig.3, is different from an English word, we cannot separate the characters by scanning only in the vertical direction. In order to separate characters from the sentence, it is necessary to scan in both vertical and horizontal directions. The procedure of segmentation as shown in Fig. 5 is as follows:

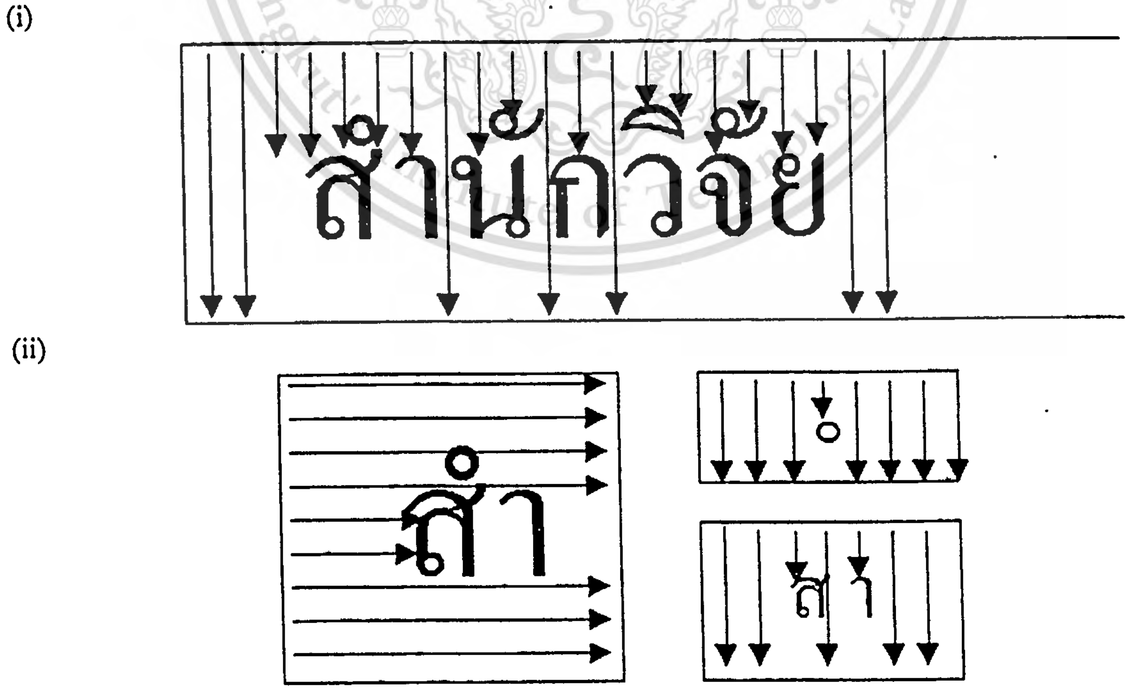


Fig 5 A method of segmentation

(i) Scan along the pattern of the sentence in a vertical direction to find the space, and pull out the data between two spaces.

(ii) Scan again along the data from item (i) in a horizontal direction, the pattern of each character is to be found between two spaces.

(iii) In particular cases, if the pattern of the character is extremely large, it is necessary to scan to find the space once more in a horizontal direction along the pattern.

The procedure (ii) is very difficult to use with a pattern such as that in Fig 6, as scanning cannot be horizontal only. In such a case, as scanning traverses the pattern in a horizontal direction and encounters 1's data, the scanning must then scan the vertical direction above or below that data to find 0's data before returning again to horizontal scanning. So a combination of vertical and horizontal scanning is necessary in this case to separate one character from the other. If this fails, then it is treated as another standard pattern for future matching with unknown patterns.

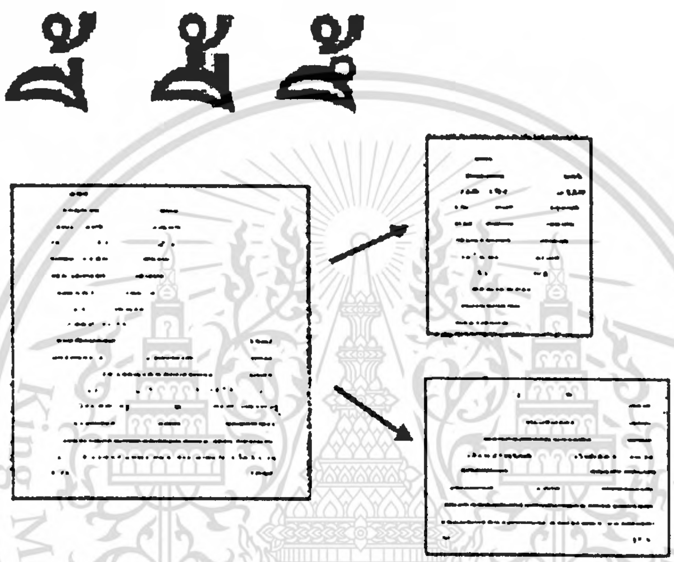


Fig. 6 Special case of pattern in which procedure(ii) is very difficult to use.

4. Elliptic Fourier description

Kuhl and Gardina [4] extracts the Fourier coefficients and from the contour functions $x(l)$ and $y(l)$ in eq.(1) and eq.(2)

$$x(l) = A_0 + \sum_{n=1}^{a} \left(a_n \cos \frac{2n\pi l}{L} + b_n \sin \frac{2n\pi l}{L}\right) \qquad (1)$$

$$y(l) = C_0 + \sum_{n=1}^{a} \left(c_n \cos \frac{2n\pi l}{L} + d_n \sin \frac{2n\pi l}{L}\right) \qquad (2)$$

Where l is the arc length computed from some origin on the contour,
L is the total length of the outer character contour,
a_n, b_n, c_n, d_n are the Fourier coefficients.

The phase angle ψ_1 are obtained from a_n, b_n, c_n, and d_n as in eq.(3).

$$\varphi_1 = \frac{1}{2} \tan^{-1} \left\{ \frac{2(a_1 b_1 + c_1 d_1)}{a_1^2 + b_1^2 - c_1^2 - d_1^2} \right\} \qquad (3)$$

To derive a set of descriptors that are invariant with respect to starting point and rotation, the rotation angle between the positive x-axis and the major axis of the first harmonic ellipse is calculated (4):

$$\phi_1 = \tan^{-1}\left(\frac{b_1}{a_1}\right) = \tan^{-1}\left(\frac{-b_1 \sin\varphi_1 + d_1 \cos\varphi_1}{a_1 \cos\varphi_1 + c_1 \sin\varphi_1}\right) \quad 0 \le \varphi_1 \le 2\pi \qquad (4)$$

In this work, we use ϕ and $\Delta l = \frac{(\varphi_1 L)}{2\pi}$ in order to normalize the contour itself. We rotate the contour by using ϕ and shift the resulting $x_r(l)$ and $y_r(l)$ functions by using Δl. We call $x_{rs}(l)$ and $y_{rs}(l)$ as the resulting boundary functions of normalization to rotation and the shift, respectively.

5. Boundary Normalization Procedure

For characters of different size, the parameter and dynamic ranges of the boundary function are different. To normalize with respect to size variations, the boundary function is performed by scaling and resembling $x_{rs}(l)$ to $\bar{x}(t)$, where $t = 0,1,\ldots T\text{-}1$, and T is a parameter chosen by user, and $\bar{x}(t) \in [0,1]$.

The same procedure is performed on $y_{rs}(l)$. The choice of T depends on the images of the characters that are recognized. If T is chosen too small, some characteristics of the boundary may be lost. If it is chosen too big, for example bigger than the parameter L, the errors due to the digitalization will be accepted.

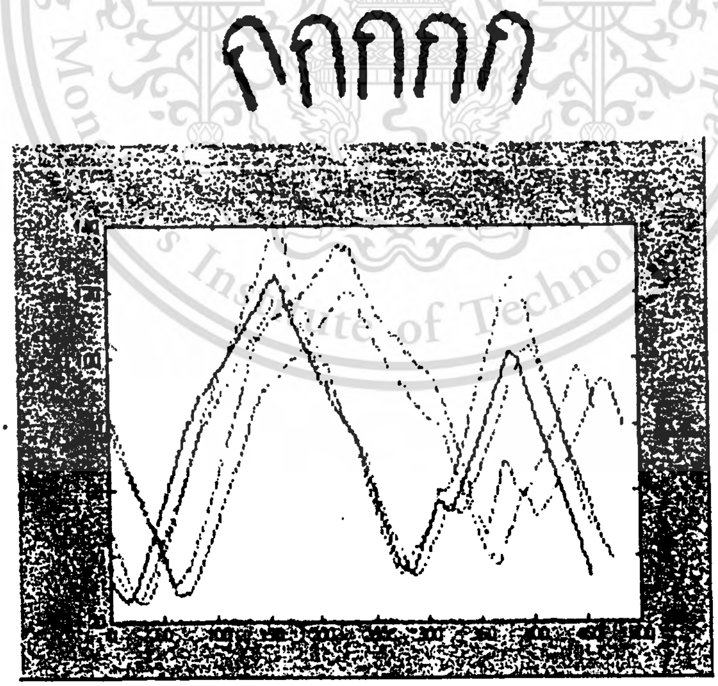


Fig 7. Characters of different orientations and the corresponding x(l) functions.

Fig 7. shows several characters of identical sizes but with different orientation, and shows the plots of the corresponding functions x(i) versus l(i).

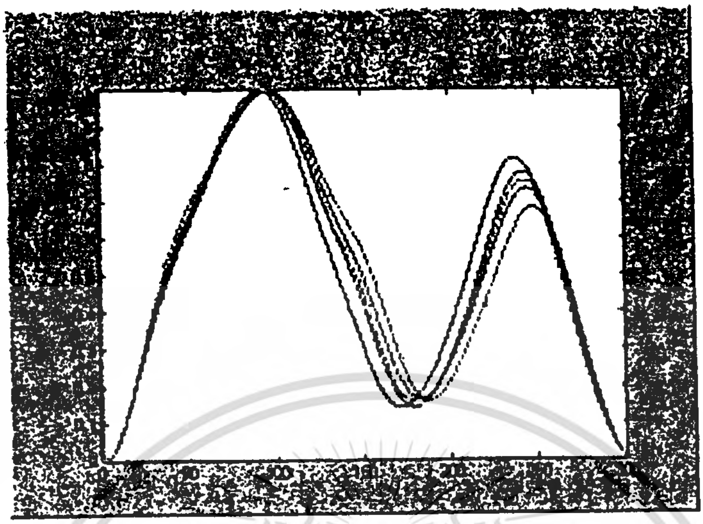
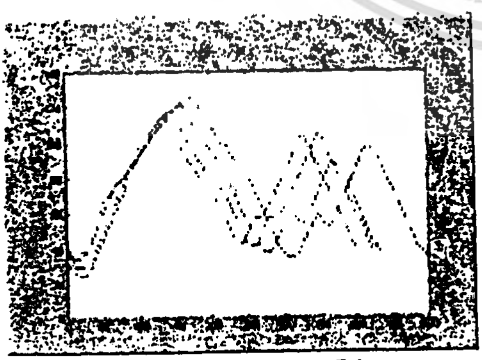


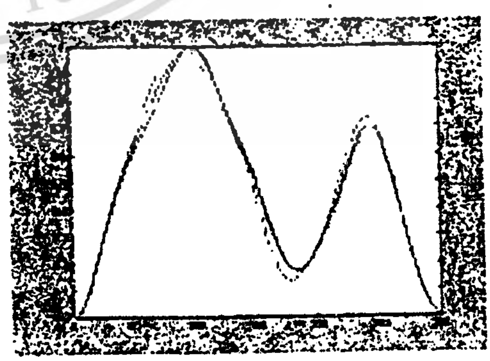
Fig 8. Normalized function $\bar{x}(t)$, T = 64, for characters from figure 7.

Fig 8. illustrates the result of the normalization algorithm of characters from Fig 7. In the case of different size of characters as in Fig 9(a), the function x(l) is shown in Fig 9(b), and $\bar{x}(t)$ for characters with different widths or heights is shown in Fig 9(c). Fig 9 is interesting because it shows that the resulting boundary functions are not only invariant to uniform dilatation, but also directional resizing. This advantage is valuable for recognizing printed characters with different fonts.

(a)



(b)



(c)

Fig 9 The example of printed Thai characters of different size, and boundary normalization where T = 300

6. Fuzzy Model for pattern Recognition

Fuzzy set theory and fuzzy modeling techniques have been applied to various fields such as pattern recognition and data clustering. These techniques allow objects to be assigned to different regions or clusters to different degrees, thereby often improving the model of the data structure. The features extracted from the boundary normalization of an input character are modeled by linguistic variables such as LOW, MEDIUM, HIGH, etc., and then the latter used within a predetermined set of fuzzy rules.

The inputs of the fuzzy network originally consisted of the T_i^p $(i = 1,2,\ldots,30)$ of the boundary normalization (at 30 discrete points of the T_i^p) on characters, and outputs are c_k^p (k = 0,1,2,...,10), corresponding to the list of it alphabet as shown in Fig 10. respectively.

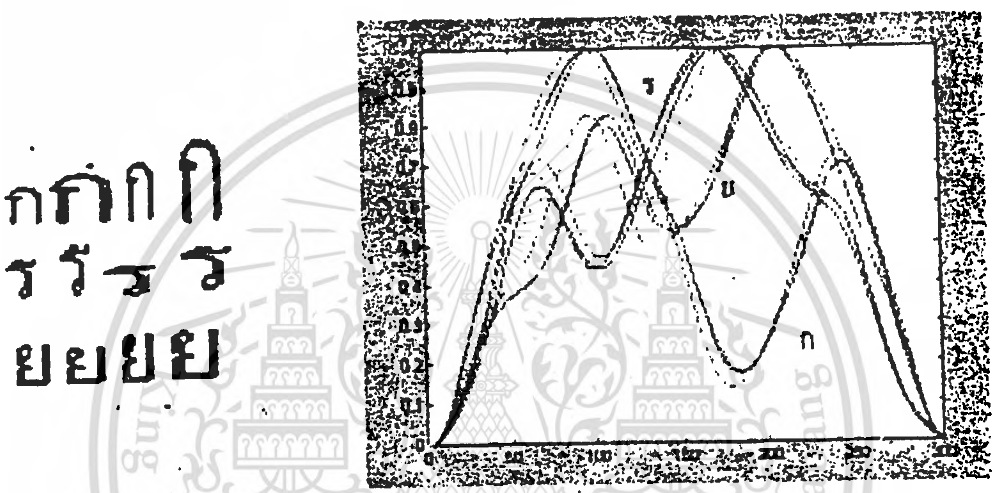


Fig 10. The example of printed Thai characters of different size, and boundary normalization

Each fuzzy inference rule is represented in the neural network by an input vector, which specifies the linguistic states of the input variables T_i^p and an output vector, which defines the corresponding fuzzy number of the output variable c_k^p operation. Fig 11. shows the membership function for the input variables. These two vectors represent one pair input-output. trained in network. The full training set consists of all Thai characters. After training, the neural network responds correctly to each input vector in the training set. For example, each of the input is covered by a set of univariate basic functions, with their size and position being important factors for determining the success of this application.

If the fuzzy system has n input $(x_1,\ldots,x_n)^T$ and m output $(y_1,\ldots,y_m)^T$, the system is composed of fuzzy rules in the form:

$$IF\,(x_1 \; is \; A_1^l) and..and (x_n \; is \; A_n^l) THEN (y \; is \; B^J)$$

Where A^1_k, k = 1,...n be linguistic variables which represent vague terms such as small, medium or large defined on the input and B^j be output variables, respectively.

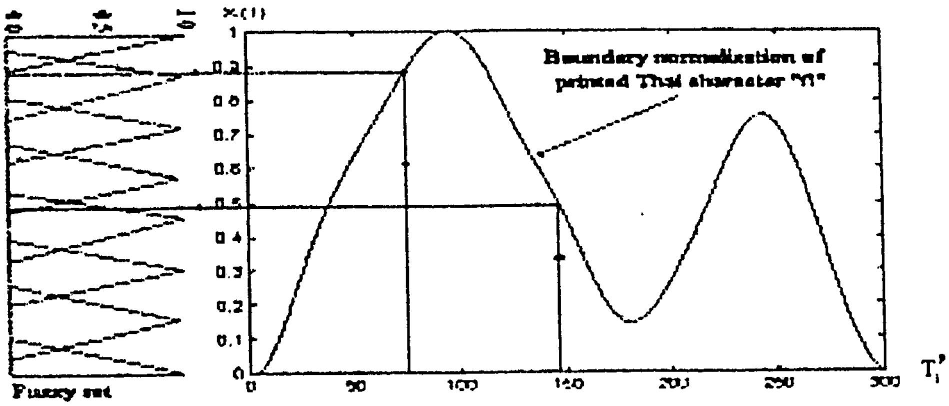


Fig 11. Fuzzy membership function of T_i^p

7.The Fuzzy Neural network system for Classification

Fuzzy neural networks to be used here were proposed by Hayashi, Buckly, and Czogala [15]. They are obtained directly fuzzifying the classical feed-forward neural networks with one more layers. All real numbers that characterize a classical neural network become fuzzy numbers in its fuzzified counterpart. These are numbers that characterize inputs to the network, output of neurons at hidden layers and the output layer, and weights at all layers. Consider, for example, all numbers relevant to a particular output neuron, ON_k, of a single-layer feed-forward neural network. If ON_k is a fuzzy neuron, then the inputs $X_{ko}, X_{k1}, \ldots, X_{kn}$, the weights $W_o, W_1, \ldots, W_n$, and the output Y_k of this neuron are all fuzzy numbers.

The output of each neuron is defined as

$$Y_k = S_\beta\left(\sum_{j=0}^{n} W_j X_{kj}\right) \qquad (5)$$

Where S_β is a sigmoid function. Since symbols W_j and X_{kj} in (6) designate fuzzy members, the sum

$$A_k = \sum_{j=0}^{n} W_j X_{kj} \qquad (6)$$

needs to be calculated by fuzzy arithmetic. The output of the neuron,

$$Y_k = S_\beta(A_k) \qquad (7)$$

is then determined by using the extension principle.

Error function E_p, employed in the Backpropagation learning algorithm in a fuzzy neural network with l output, for each training sample, is

$$E_p = \frac{1}{2}\sum_{k=1}^{l}(T_k^p - Y_k^p)^2 \tag{8}$$

Where T_k^p is the desired output and Y_k^p is the actual output of neuron ON_k for training sample p. Here, again, fuzzy arithmetic is used to calculate E_p. Otherwise, E_p would become exactly the same as its counterpart for classical neural network.

The stopping criterion for fuzzy neural networks must also be properly fuzzified. Assume that $T_k^p = Y_k^p$ for all k, which represents a perfect match of the actual outputs with the target outputs. Then, assuming that the support of $T_k^p\,(and\,Y_k^p)$ is the interval $\left[t_{k1}^p, t_{k2}^p\right]$, the support of E_p is included in the interval$[-\lambda,\lambda]$, where

$$\lambda = \frac{1}{2}\sum_{k=1}^{l}(t_{k2}^p - t_{k1}^p)^2 \tag{9}$$

Choosing some number $\varepsilon > 0$ as an acceptable deviation from the value of E_p when $T_k^p = Y_k^p$ for all k, if is reasonable to stop the learning algorithm. Whenever E_p is induced in the interval $[-\lambda-\varepsilon, \lambda+\varepsilon]$.

Finally, we need to fuzzy the Backpropagation learning algorithm. One way, proposed by Hayashietal. [15] , is to replace the real numbers in the standard formulas with their fuzzy counterparts and apply fuzzy arithmetic to them.

Fuzzy neural network used in this experiment compose of 3 layers and this system can be used for classification: the input layers receives the fuzzy input variables T_i^p and the output layer has as many units as there are classes. The input layer has 30 nodes. The output layer has 20 nodes, corresponding to 89 printed characters. The number of node in hidden layer in this experiment has 62 nodes.

8. The Experimental Results

A set of test images for examples are used in our experiment as shown in Fig 10. Which includes normal, slant and different size characters sampled at a resolution of 300 dpi. The output is simulated using Matlab Version 5.1 on Window95 performed on the Pentium 166 MHz personal microcomputer. We trained fuzzy neural networks with boundary normalization of the character images of the fonts CordiaUPC and AngsanaUPC of size 14, 18, 22, 28. The total numbers of test characters were about 1600 characters.

Sometimes there are some very similar printed Thai Characters that can not be classified, then we used sub region technique by dividing some regions into two parts using a horizontal dividing line, and then fed the sub region into boundary normalization and fuzzy neural network again. By applying this technique, a separation of similar characters were obtained, and increased recognition rate of 98.5%. The experimental results on the test data are shown is Table 1, and the comparison of the recognition rate for boundary normalization and fuzzy neural network with other methods are shown in Table 2.

Table 1. Recognition results

Font Name	Size	Number	Recognition (%)	Miss-Recognition (%)
AngsanaUPC	22	100	98	2.00
	24	100	98	2.00
	26	100	99	1.00
	28	100	100	0.00
JusmineUPC	22	100	97	3.00
	24	100	98	2.00
	26	100	99	1.00
	28	100	100	0.00
EucrosiaUPC	22	100	97	3.00
	24	100	98	2.00
	26	100	99	1.00
	28	100	99	1.00
KodchingUPC	22	100	97	3.00
	24	100	98	2.00
	26	100	99	1.00
	28	100	100	0.00
Total		**1600**	**98.5**	**1.5**

Table 2. The comparison of the recognition rate of boundary normalization and fuzzy neural method with other methods.

Recognition method	Accuracy Rate %
Local Feature	98.3
Matching Model	98.0
Karhunen-Loeve	98.2
Boundary Normalization and Fuzzy Neural Network	98.5

9. Conclusion

We have proposed a new method that combines boundary normalization (Elliptic Fourier Series) of characters and fuzzy neural network for the task of Thai printed character recognition. As a result, we obtained a contour, which is invariant to orientation and dilatation. Fuzzy logic techniques were used to cluster the data and construct the features, which are not so obvious. Feed forward neural networks with the back propagation and learning algorithms were used in the learning and classification of the 89 letters of the Thai alphabet. The recognition rates were high even for noiseless images of the characters.

References

[1] C. Kimpan, A. Itoh, and K. Kawanishi, "Recognition of printed Thai characters using a matching method.", IEE Proc., Vol. 130, Pt. E, No. 6, pp. 138-188, Nov. 1983.

[2] C. Kimpan, "Printed Thai character recognition using topological properties method.", INT. J. Electronics, Vol. 60, No. 3, pp. 303-329, 1986.

[3] C. Kimpan, A. Itoh, and K. Kawanishi, "Fine classification of printed Thai character recognition using Karhunen-Loeve expansion.", IEE Proc., Vol. 134, Pt. E., No.5, pp. 257-264 Sept. 1987.

[4] F.P. Kuhl and C.R. Giardina, "Elliptic Fourier features of closed contour." Computer Graphics and Image Processing, (18): pp. 236-258, 1982.

[5] M. Szmurlo, "Boundary normalization for recognition of non-touching non-degraded characters." IEEE International Conference, pp. 463-466, 1997.

[6] P. Hiranvanichakorn, T. Agui, and M. Nakajima, "A recognition method of Thai characters." IEICE Trans. Vol. E. 65, pp. 737-744, Dec. 1982.

[7] P. Hiranvanichakorn, T. Agui, and M. Nakajima, "Recognition method of Thai characters by using local features." IECE Trans., Vol. E 67, pp. 425-432, Aug 1984.

[8] P. Hiranvanichakorn, T. Agui, and M. Nakajima, "A recognition method of handprinted Thai characters by local features." IECE Trans. Vol. E 68, pp. 83-90, Feb. 1985.

[9] P. Hiranvanichakorn, T. Agui, and M. Nakajima, "An online recognition method of Thai characters." IECE Trans. Vol. E 68, pp. 594-601 Nov.1985.

[10] P. Phokharatkul and C. Kimpan, "Recognition of handprinted Thai characters using the cavity features of character based on neural network." IEEE International Conference Asia-Pacific on Electronic circuit, Chiangmai, Thailand, pp. 149-152, 1998.

[11] S. Airphaiboon, M. Sangwarasil, B. Attachoo, and C. Kimpan, "Recognition of handwritten Thai character considering the head of character." IEEE International Conference on Image Processing. Vol. 2, Singapore, pp. 326-330, 1989.

[12] S. Airphaiboon, S. Kondo, "Recognition of handprinted Thai characters using loop structures." IECE Trans. INF. & SYST. Vol. E 79-D, No. 9, pp. 1296-1304, Sep. 1996.

[13] V. Moungsinh, P. Phokharatkrul and C. Kimpan, "Recognition Handprinted Thai Characters Using Karhunen-Loeve Transform and Fuzzy-Neural Networks", Proceedings EECON22, Thailand, , pp. 64-67, Dec.1999.

[14] P. Lekhachaivorakul. P. Kortungsap, and S. Moadarasmi, "A Performance Comparison of Various Input Features to a Feedforward Neural Network for on-line Handwriting Recognition", Proceedings EECON22, Thailand, pp. 60-63, Dec 1999.

[15] Georage J.Klir and Bo Yuan, Fuzzy sets and Fuzzy logic Theory and Applications, Prentice Hell PTR, New Jersey, pp. 347-349, 1995.

Recognition of Printed Thai Characters Using Elliptic Fourier Descriptors and Genetic Neural Networks

Pisit Phokharatkul, Ukrit Marang, and Chom Kimpan

Faculty of Engineering, Research Center for Communications and Information Technology
King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand.
Tel. (662) 3269969 ext. 125, Fax. (662) 7392400
Email. s0060004@kmitl.ac.th, s1061160@kmitl.ac.th, kkchom@kmitl.ac.th

Abstract

A recognition system for printed Thai character is proposed. We employ various methods i.e. Elliptic Fourier descriptors, **Genetic Algorithms and Back-propagation** Neural Networks to detecting boundary functions of characters, and attempt to normalize these functions by orientation and size variations of the symbols. Here, Genetic Algorithms and Neural Networks used to model the learning between the variables of boundary normalization of the characters. Finally, use of feed forward neural network to learn a boundary normalizes functions. A genetic algorithm is used to adjust the weights of the network interconnection. The experimental result shows the recognition rate are high though only a limited number of features have been involved. The result of this approach indicates that the systems can successfully recognized 3,204 Thai characters with 94% accuracy.

1. Introduction

Many researches for transforming characters images to text in Thai language have been studied for many years. The researchers have attempted to propose several methods to implement in computers and then apply them in various areas. The details of these approaches are illustrated in [1-9]. Although the accuracy rate of the exist systems are accepted, but they have limitations on size and orientation variations. Our approach describes the integration methods among elliptic Fourier descriptors, and genetic algorithm neural networks. Two issues may arise in discussion with the shape of character that are orientation or slant and size variation, which this problem can be solved by finding the boundary function of the characters. The principle in this research is based on the method developed by Kuhl and Gardina [4]. We continually employ a contour function to describe the essential features of character and then feed these values into the back propagation neural network, for training and recognition. A typical character recognition system consists of the following three stages as shown is Fig 1.

The preprocessing stage, involves noise removal, smoothing border of character, image segmentation, and contour following respectively.

In the second stage, the Thai character was classified into 4 rough classes using the number of inner contours. Then, normalize the boundary with respect to initial orientation and size variations. After normalization the Boundary function of characters are trained by the genetic neural network for classify the character from the each rough class.

In the testing stage, testing data is fed through the preprocessing stages and normalizes the orientation and size variation of the character boundary. Finally, the normalize boundary of character activate to the trained neural network

The paper is organized as follows: Section 2-4 briefly reviews the segmentation, elliptic Fourier descriptors, and Boundary normalization procedure. The construction of a Thai word as shown in Fig 2. The elliptic Fourier descriptors use initialize the boundary of the character. The features of boundary function are extracted and feed to genetic neural system. In section 5-6, we show the application of this algorithm in our work. The application of using the genetic neural networks is described in section 5.

2. Segmentation

The Thai characters consist 42 consonants, 17 vowels, 5 voice tones, 10 Thai

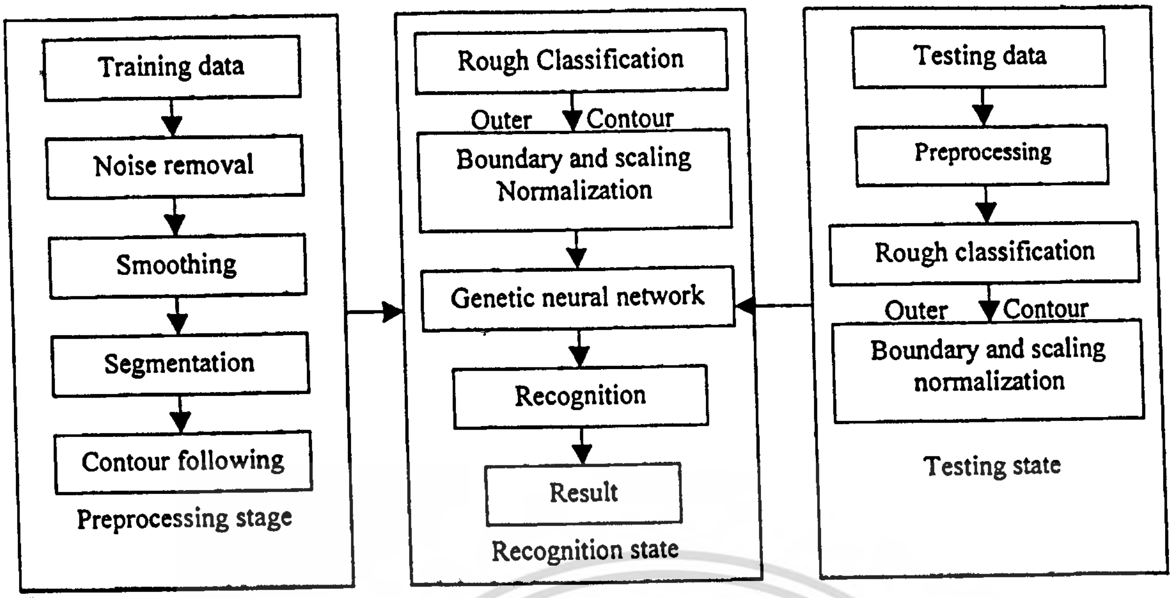


Fig 1. Flowchart of the recognition system.

numerals and 10 Arabic numbers. A Thai language sentence is composed of consonants, vowels and voice tones on different levels. Because the construction of a Thai word, as shown in Fig.2, is different from and English word, we cannot separate the characters by scanning only in the vertical direction. In order to separate characters from the sentence, it is necessary to scan in both vertical and horizontal directions. The procedure of segmentation as shown in Fig. 2 is as follow:

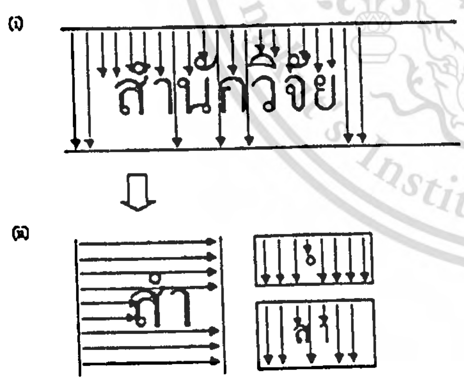


Fig. 2 A method of segmentation

(i) Scan along the pattern of the sentence in a vertical direction to find the space, and pull out the data between two spaces.

(ii) Scan again along the data from item (i) in a horizontal direction, the pattern of each character is to be found between two spaces.

(iii) In particular cases, if the pattern of the character is extremely large, it is necessary to scan to find the space once more in a horizontal direction along the pattern.

The procedure (ii) is very difficult to use with a pattern such as that in Fig 3, as scanning cannot be horizontal only. In such a case, as scanning traverses the pattern in a horizontal direction and encounters 1's data, the scanning must then scan the vertical direction above or below that data to find 0's data before returning again to horizontal scanning. So a combination of vertical and horizontal scanning is necessary in this case to separate one character from the other. If this fails, then it is treated as another standard pattern for future matching with unknown patterns.

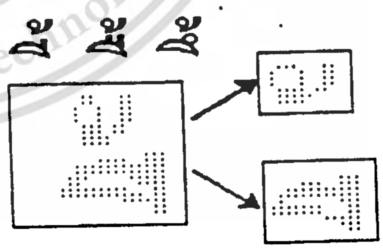


Fig. 3 Special case of pattern in which procedure (ii) is very difficult to use.

3. Elliptic Fourier descriptors

Kuhl and Gradina [4] extracts the Fourier coefficients and from the contour functions $x(l)$ and $y(l)$ in eq.(1) and eq.(2)

$$x(l) = A_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{2n\pi l}{L} + b_n \sin \frac{2n\pi l}{L}\right) \quad (1)$$

$$y(l) = C_0 + \sum_{n=1}^{\infty} \left(c_n \cos \frac{2n\pi l}{L} + d_n \sin \frac{2n\pi l}{L}\right) \quad (2)$$

Where l is the arc length computed from some origin on the contour,

L is the total length of the outer character contour,

a_n, b_n, c_n, d_n are the Fourier coefficients.

The phase angle ψ_1 are obtained from a_n, b_n, c_n and d_n as in eq(3).

$$\varphi_1 = \frac{1}{2} \tan^{-1}\left\{\frac{2(a_1 b_1 + c_1 d_1)}{a_1^2 + b_1^2 - c_1^2 - d_1^2}\right\} \quad (3)$$

To derive a set of descriptors that are invariant with respect to starting point and rotation, the rotation angle between the positive x-axis and the major axis of the first harmonic ellipse is calculated (4):

$$\phi_1 = \tan^{-1}\left(\frac{b_1}{a_1}\right) = \tan^{-1}\left(\frac{-b_1 \sin\varphi_1 + d_1 \cos\varphi_1}{a_1 \cos\varphi_1 + c_1 \sin\varphi_1}\right)$$

$$0 \le \varphi_1 \le 2\pi \quad (4)$$

4. Boundary Normalization Procedure

In this work, we use ϕ and $\Delta l = \frac{(\varphi_1 L)}{2\pi}$ in order to normalize the contour itself. We rotate the contour by using ϕ and shift the resulting $x_r(l)$ and $y_r(l)$ functions by using Δl. We call $x_{rs}(l)$ and $y_{rs}(l)$ as the resulting boundary functions of normalization to rotation and the shift, respectively.

For characters of different size, the parameter and dynamic ranges of the boundary function are different. To normalize with respect to size variations, the boundary function is performed by scaling and resembling $x_{rs}(l)$ to $\bar{x}(t)$, where $t = 0,1,\ldots,T\text{-}1$, and T is a parameter chosen by user, and $\bar{x}(t) \in [0,1]$.

The same procedure is performed on $y_{rs}(l)$. The choice of T depends on the images of the characters that are recognized. If T is chosen too small, some characteristics of the boundary may be lost. If it is chosen too big, for example bigger than the parameter L, the errors due to the digitalization will be accepted.

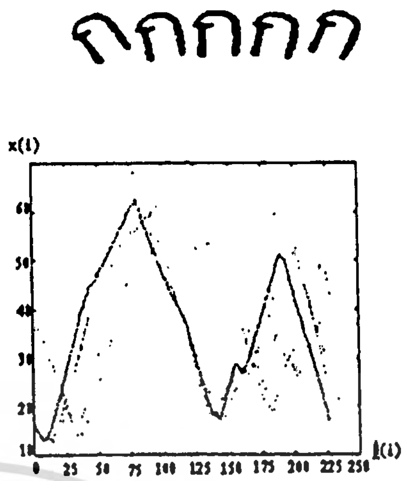


Fig 4. Characters of different orientations and corresponding x(l) function.

Fig 4. shows several characters of identical sizes but with different orientation, and shows the plots of the corresponding functions x(i) versus l(i).

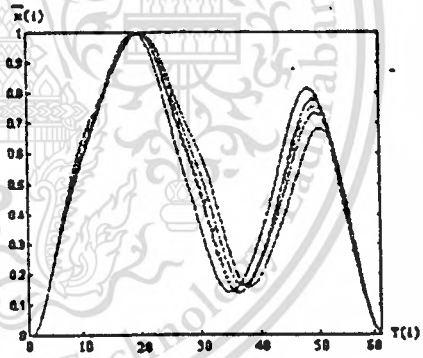
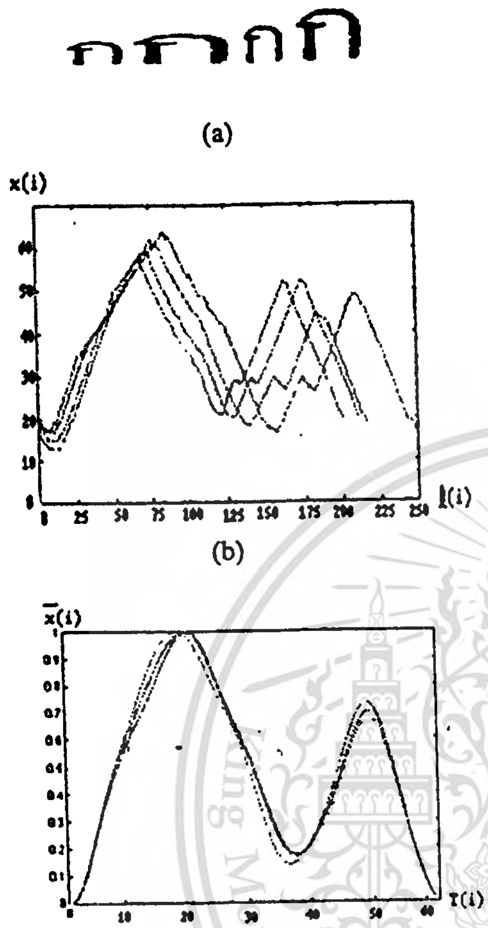


Fig 5. Normalized function $\bar{x}(t)$, T = 64, for characters from Fig. 4

Fig 5. Illustrates the result of the normalization algorithm of characters from Fig 4.

In the case of different size of characters as in Fig 6(a), the function x(l) is shown in Fig 6(b), and $\bar{x}(t)$ for characters with different widths or heights is shown in Fig 6(c). Fig 6 is interesting because it shows that the resulting boundary functions are not only invariant to uniform dilatation, but also directional resizing. This advantage is valuable for recognizing printed characters with different fonts.



(c)

Fig 6 The example of printed Thai characters of different size, and boundary normalization where T = 64

5. Genetic Neural Network Training

In this paper, genetic algorithms have been used to search the weight space of the multi layer feed forward into neural network without the use of any gradient information [10]. The existing GA and NN have been combined to form the new hybrid system. The system model is shown in Fig 7.

The basic concept of this technique is as follows. A complete set of weights in coded in real number string, which has an associated "fitness" indicating its effectiveness. For example, the fitness can be simply given by E, where E is the value of the cost function for that set of weights. Starting with a random population of such strings, successive generations are constructed using genetic operators to construct new strings out of old ones such that fitter strings (weights) are more likely to survive and to participate in crossover operations. The crossover operation can in principle bring together good building blocks such as hidden units that compute certain logical function found by chance in different members of the population. Unlike the backpropagation learning rule, GAs performs a global search and are thus not easily fooled by a local minimum. Furthermore, since the fitness function does not have to be differentiable, we can, for example, start with threshold units in Boolean problems instead of having to use sigmoid that are later trained to saturation. However, for neural networks with continuous activation functions, it is an appropriate compromise to hybridize GAs with gradient methods such as the back-propagation algorithm. An initial genetic search followed by a gradient method or gradient-descent step can be included as one of the genetic operators. The example of encoding a network on a chromosome, is shown in Fig 8. In Fig 8., the weights and biased, the hidden and output node values of a neural network are encoded in order as a list. The interaction between NN and GA is performed by message passing. Both NN and GA instances are created at the beginning of the optimization procedure and last until the end. The GA makes call to NN and passed messages to the fitness function.

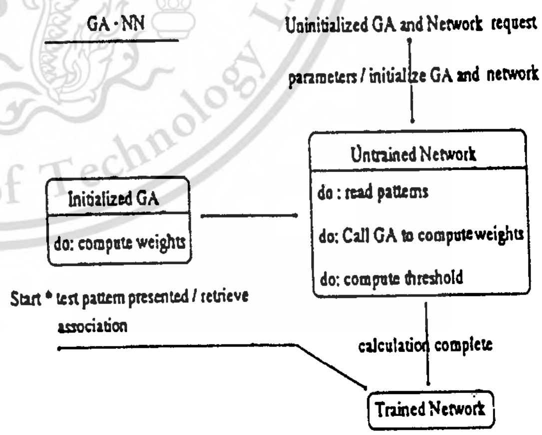


Fig 7. The GA-NN

In the experiment, the neural networks are operated in two modes: training, and testing. In the training mode, a set of training data is used to adjust the weights of the network interconnection. The training neural networks are determined by these weights. In testing mode, the trained neural network is activated by the testing data.

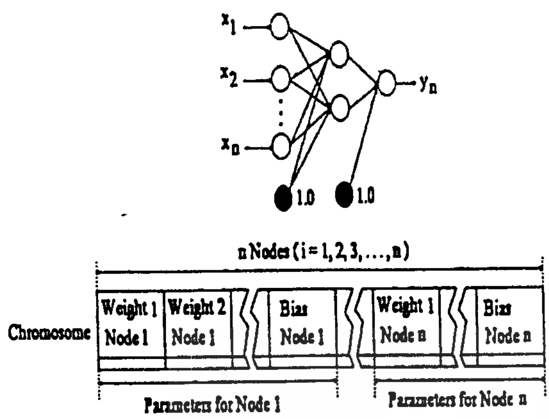


Fig 8. Encoding a network on a chromosome.

Genetic neural network used in this experiment compose of 3 layers and this system can be used for classification: the input layers receives the boundary moralized input variables $T_i^p\,(i = 2,4,\ldots,64)$ and the output layer has as many units as there are classed. The input layer has 32 nodes. The output layer has 8 nodes, corresponding to printed characters in each class. The number of node in hidden layer in this experiment has 20 nodes.

6. The Experimental Results

A set of test images for examples are used in our experiment as shown in Fig 9. Which includes normal, slant and different size characters sampled at a resolution of 300 dpi. The output is simulated using Matlab Version 5.1 on Window95 performed on the Pentium 166 MHz personal microcomputer. We trained genetic neural networks with boundary normalization of the character images of the fonts AngsanaUPC, CordiaUPC and EucrosiaUPC of size 14, 18, 22, 28. The total numbers of training character set were about 1,068 characters.

Sometimes there are some very similar printed Thai characters that can not be classified, then we used sub-region technique by dividing some region into two parts using a horizontal dividing line, and then fed the sub region into boundary normalization and genetic neural network again. By applying this technique, the separation of similar characters were obtained, and increased recognition rate of 98.25% for training set data. The experimental results on the test data are shown is Table 1, and the comparison of the recognition rate for boundary normalization and genetic neural network with other methods are shown in Table 2.

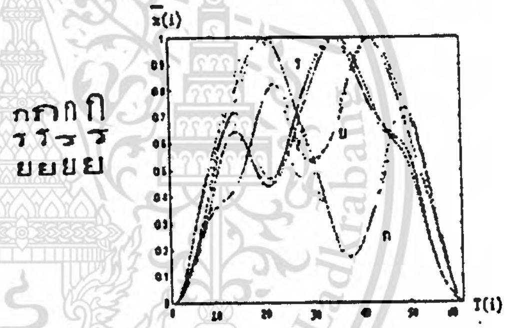


Fig. 9 The example of printed Thai characters of different size, and boundary normalization.

Table 1. Recognition results

Font Name	Size	Training set		Test set	
		No. of char.	Recognition(%)	No. of char.	Recognition(%)
AngsanaUPC	14	89	98	267	92
	18	89	99	267	93
	22	89	99	267	94
	28	89	99	267	97
CordiaUPC	14	89	97	267	92
	18	89	97	267	93
	22	89	98	267	95
	28	89	99	267	97
EucrosiaUPC	14	89	97	267	92
	18	89	98	267	93
	22	89	99	267	94
	28	89	99	267	96
Total		1,068	98.25	3,204	94

Table 2. The comparison of the recognition rate of elliptic Fourier descriptors and genetic neural method with other methods.

Recognition method	Accuracy Rate %
Local Feature[5]	88.9
Matching Model[1]	98.0
Karhunen-Loeve[3]	98.2
Elliptic Fourier Descriptors and Genetic NN	94.0

7. Conclusion

We have proposed a new method that combines boundary normalization (shift the starting point by Elliptic Fourier Series) of characters and genetic neural network for the task of Thai printed character recognition. As a result, we obtained a contour, which is invariant to orientation and dilatation. Genetic algorithms were used to search the weight space of the neural network. The feed forward neural networks with the back propagation and learning algorithms were used in the learning and classification of the 89 letters of the Thai alphabet. The recognition rates were high even for noiseless images of the characters.

References

[1] C. Kimpan, A. Itoh, and K. Kawanishi, "Recognition of printed Thai characters using a matching method.", IEE Proc., Vol. 130, Pt. E, No. 6, pp. 138-188, Nov. 1983.

[2] C. Kimpan, "Printed Thai character recognition using topological properties method.", INT. J. Electronics, Vol. 60, No. 3, pp. 303-329, 1986.

[3] C. Kimpan, A. Itoh, and K. Kawanishi, "Fine classification of printed Thai character recognition using Karhunen-Loeve expansion.", IEE Proc., Vol. 134, Pt. E., No.5, pp. 257-264 Sept. 1987.

[4] M.Szmurlo, "Boundary normalization for recognition of non-degraded characters." IEEE International Conference, pp. 463-466, 1997.

[5] P. Hiranvanichakorn, T. Agui, and M. Nakajima, "A recognition method of handprinted Thai characters by using local features." IECE Trans., Vol. E 68, pp. 83-90, Feb 1985.

[6] P. Phokharatkul and C. Kimpan, "Recognition of handprinted Thai characters using the cavity features of character based on neural network." IEEE International Conference Asia-Pacific on Electronic circuit, Chiangmai, Thailand, pp. 149-152, 1998.

[7] S. Airphaiboon, M. Sangwarasil, B. Attachoo, and C. Kimpan, "Recognition of handwritten Thai character considering the head of character." IEEE International Conference on Image Processing. Vol. 2, Singapore, pp. 326-330, 1989.

[8] S. Airphaiboon, S. Kondo, "Recognition of handprinted Thai characters using loop structures." IECE Trans. INF. & SYST. Vol. E 79-D, No. 9, pp. 1296-1304, Sep. 1996.

[9] V. Moungsinh, P. Phokharatkul and C.Kimpan, "Recognition Handprinted Thai characters Using Karhunen-Loeve Transform and Fuzzy-Neural Networks", Proceeding EECON22, Thailand, . pp. 64-67, Dec.1999.

[10] Chin-Teng Lin and C.S. George Lee, Neural Fuzzy System., Printice-Hall International, Inc., USA, pp.399-404, 1996.

SCI 2001/ ISAS 2001 Best Papers

Sessions Title	Authors' Names	Papers Title
Applications of Computer Arithmetic: Multimedia, Cryptography, DSP	Abdul Shakoor, Abdul Rafeeq; Szwarc, Valek; Kwasniewski, Tad A. (Canada)	Low Power Viterbi Decoder Implementation for CDMA Applications
Wireless computing tools and softwares	Abid, M. R.; Rachidi, T.; Bensaid, A. (Morocco)	An Adaptive Fuzzy Call Admission Controller for UMTS
Intra-, Extra- and Internet	Abreu, Jorge T. Ferraz De; Almeida, Pedro Alexandre F. S. (Portugal)	2BeOn integrated multimedia communication environment towards telework and leisure activities
User Interface Design	Acosta, Eleonora Alecia; Zambrano Nancy (Venezuela)	Patterns and Web Interface Design
Issues in SOC Design	Ahmadi, Shahrokh; Zaghloul, Mona (USA)	System on Chip; a Fabry-Perot Sensor
Management Information Systems I	Akomode, Joseph (UK)	Measuring the Satisfaction of Customers to Minimise Business Risks and Improve Quality in New Products
Image and Multidimensional Signal Processing I	Al-Asmari, Awad Kh. (Saudi Arabia)	Video Compression Using A Semi-Hexagonal Shaped Absolute Moment Block Truncation Coding
Theoretical Methods in Cryptology	Alvarez, G.; Montoya, F.; Romera, M.; Pastor, G. (Spain)	Cryptanalytic Methods in Chaotic Cryptosystems
Mobile / Wireless Computing II	Alves de Barros, M. *; Naviner, L. ** (* Brazil, ** France)	A Genetic Based Processor for Predictive Coding in Mobile Communications
Conceptual Infraestructure of SCI	Allgood, Glenn O.; Manges, Wayne W. (USA)	Assessing Risk of Innovation and Technology Adoption Using the Value-Derivative Model
Digital Signal Ptocessing I	Anderson, David; Hasler, Paul (USA)	Cooperative analog/digital signal processing
DWDM Telecommunication Systems: Enabling techniques	Andr, P. S.; Pinto, J.; Teixeira, A. J.; Almeida, T.; Nolasko Pinto, A.; Pinto, J. L.; Morgado, F.; Pousa, M. (Portugal)	Strictly Non-Blocking Optical Path Cross - Connect System for DWDM Wavelength Path Networks
Control Systems V	Areibi, Shawki M. (Canada)	Iterative Improvement Heuristics for Standard Cell Placement: A

Computing I	(USA)	Filtering for Dispersive CDMA Signals
Speech Processing II	zhan, Orhan; Pastaci, Halit (Turkey)	Glottal Contribution To Speaker Identification
Network Architectures, Topologies and Protocols V	Pahl, Claus (Ireland)	Analysis and Test Case Generation for Security Protocols using Abstraction
Information Systems Development, Methods and Methodologies V	Panda, Brajendra; Ragothaman, Prahalad (USA)	Alternate Methods of Storing Committed Transactions in the Log for Their Future Re-execution
Evolutionary Computation	Pandolfi, D.; Vilanova, G.; De San Pedro, M.; Villagra, A.; Gallard, R. (Argentina)	Adaptability of Multirecombined Evolutionary Algorithms in the Single-Machine Common Due Date Problem
Evolutionary Computation	Pandolfi, D.; Vilanova, G.; De San Pedro, M.; Villagra, A.; Gallard, R. (Argentina)	Solving the Single-Machine Common Due Date Problem Via Studs and Inmigrants in Evolutionary Algorithms
Information Systems Development, Methods and Methodologies I	Park, Jeongho (USA)	Distributed Denial of Service Issues and Defense Strategies
ABR to the Internet	Park, Sangkyu; Li, Na; Li, San-Qi (USA)	Threshold-based Binary Feedback for Congestion Avoidance
SCI in Biology and Medicine II	Park, Seon Hee; Lee, Sooyeul; Kim, Seunghwan; Jeong, Ji-Wook (Korea)	Analysis of Bone Trabecular Patterns by Fractal Dimension
Creating the Image for New Systems	Parker, James W. (USA)	Training to Enhance Design Team Performance: A Cure for Tunnel Vision
Special Methodologies for Specific Kinds of Applications (DSS, EIS, etc.) II	Periyasamy, K. *; Alagar, V. A. ** (* USA, ** Canada)	Separating Business Rules from Application Domain
Internet Video Streaming	Pesquet-Popescu, Batrice; Bottreau, Vincent (France)	Lifting Schemes In Scalable Video Coding
Robotics and Control Systems II	Phelps, John A.; Wagner, Thomas A. (USA)	Knowledge-Based Task Structure Generation for a Real-Time Information Gathering Agent
Document and Natural Language Processing	Phokharatkul, Pisit *; Kimpan,Chom *; Sato, Makoto ** (* Thailand, ** Japan)	A Rough Ring Projection Method for Invariant Thai Character Recognition
Genetic and Agent: From		

A Rough Ring Projection Method for Invariant Thai Character Recognition

Pisit Phokharatkul, Chom Kimpan
Research Center for Communication and Information Technology, Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand.
Email: egpph@mahidol.ac.th, kkchom@kmitl.ac.th

and

Makoto Sato
Precision and Intelligence Laboratory, Tokyo Institute of Technology,
4259 Nagatsuta, Midori-ku, Yokohama, 226-8503, Japan.
Email.: msato@pi.titech.ac.jp

ABSTRACT

This paper presents a method to solve the problems of translation, rotation, and scale in invariant recognition of Thai characters. The first step of the method (coarse classification) takes into account the invariant properties of the normalized ring projection values. The ring projection is obtained using the total number of foreground pixels as distribution along circular rings. These have a radius dependent on a centroid of the shape, but independent of the shape's position, orientation, and scale. After obtaining a ring projection, we get data sets to set up the object's attributes. The rough sets receive the appropriate attributes of each object, and generate the decision-making rules for coarse classification. In the second step (fine classification), we use the same method as in the first step to classify similar characters in the same class. We apply the two-stage classification method to the Thai character recognition system. The proposed method in this research can solve the problem of invariant characters and the defect of the heads of Thai characters. The experimental results show the usefulness of the proposed method.

Keywords: Character Recognition, Ring Projection, Rough Sets, Invariant Recognition

1. INTRODUCTION

The rough set and its application have become the fastest growing research area in recent years. Advanced research and development in rough sets has found numerous applications, such as in data mining, image processing, and hybridization with fuzzy sets or genetic algorithms. During this fast growth the rough set development of invariant character recognition has lagged behind compared to the other areas. An increasing number of research groups have proposed a variety of invariant character recognition methods. Among them, we can find boundary-based analysis via Fourier descriptors [1,2], wavelet transformation [3], neural network models [4], and invariant moment [5]. However, most of these methods are not new disciplines or too computationally intensive.

In this contribution, we attempt to apply the rough set to the invariant character recognition that achieved excellent invariance under translation, rotation, and scaling. The application of rough set theory to invariant Thai character recognition is a new development. There had been an attempt to apply other techniques to Thai character recognition [9-12] but this cannot be used for the invariant problem. The recognition method has two steps: coarse classification and fine classification. The first step takes into account the ring projection vectors and applies the rough sets to create the rules base for classification. The second step is done by the fine classifier into recognized characters. This step has been made to use the same concept of rough sets, as in the first step, to classify the characters in each coarse class.

In order to apply the rough set to this paper, we must find the appropriate attributes for each object or pattern in this instance. We use the ring projection vectors and the rough rule bases for coarse grouping. The 2nd level classification is used to classify the characters in each coarse class. The recognition procedure will be illustrated in next section.

2. FEATURE EXTRACTION

In this paper, we use an invariant projection method to extract features from the Thai characters. Ring projection [3] is defined as the total member of foreground pixels whose distances (or radii) to the centroid of character pattern are r_i as shown in Fig .I. All ring projections in a character pattern constitute a ring projection vector that is invariant to rotation. We can derive the centroid of the character, as denoted by m (x_0, y_0) and, subsequently, translate the origin of our reference frame to this centroid. Next, we let

$$R_{max} = \max |N(x,y) - m(x_0, y_0)| \qquad (1)$$

where $|N(x,y) - m(x_0, y_0)|$ represent the Euclidean distance between two points, N and m on the plane. Further, we transform the original reference Cartesian frame into a polar frame based on the following relations.

$$\left.\begin{array}{l} x = r\cos\theta \\ y = r\sin\theta \end{array}\right\} \qquad (2)$$

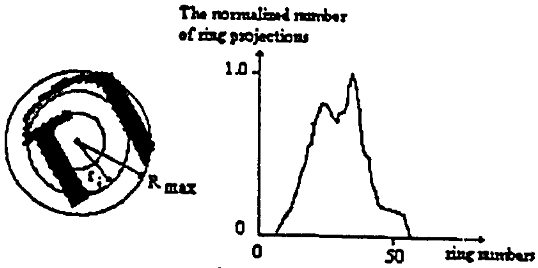


Fig. 1. An illustration of the ring projection for the Thai character "ก" rotated 30°

Hence

$$p(x,y) = P(r\cos\theta, r\cos\theta)$$

where $r \in [0, \infty]$, $\theta \in [\theta, 2\pi]$. For any fixed $r \in [0, R_{max}]$, we then compute the following integral:

$$f(x,y) = \int_0^{2\pi} p(r\cos\theta), r\sin\theta) d\theta \quad (3)$$

The resulting f(x,y) is equal to the total mass as distribution along circular rings, as shown in Fig. 1. From a concept of ring projection as mentioned above, the ring projection is calculated by

$$\text{ring projection (I)} = f(0,0) + \sum_{l=1}(x,y),$$

$$\text{ring projection (l)} = \sum_{l=r}(x,y), \quad \text{if } 2 \le l \le L$$

$$\text{ring projection (L)} = \sum_{L=r}(x,y),$$

where $r = \text{int}(\sqrt{x^2 + y^2})$, $L = R_{max}$ is the largest radius, $l = 1,2,3,\ldots,L$ is the ring number, and int ( ) means taking the integer part from a real number. We use these values to set up the object attributes as the sets, and to obtain the decision rules for classification by the rough set as the detail in section IV.

3. ROUGH SETS

Rough set theory [6-8] is a mathematical tool using the basic concepts of logic in traditional set theory. The idea of the rough set consists of the lower and upper approximation of this set as shown Fig. 2. These approximations are interior, and closure operations in a certain topology. They are generated by available data about the elements of the set. In other words, the rough set approach is based on the ability to classify data obtained from observation and from measurements.

3.1 Information Systems

An information system can be expressed as a pair S = (U, A). Where U is nonempty, finite set of objects known as the universe, and A is a finite of attributes. For every

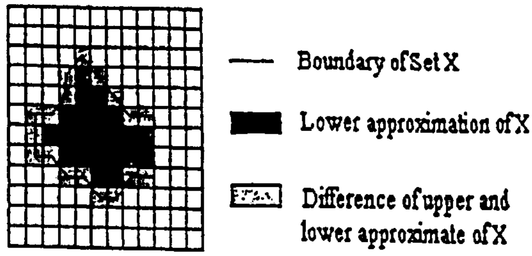


Fig 2. Rough approximation.

their exists a set of its values known as a domain of a, denoted as V_a. For every attribute $a \in A$ (a belongs to A), there exists a function $u : U \to V_a$, which assigns a unique attribute value from V_a to every object $x \in U$. Now consider subsets of attributes $B \subseteq A$, where B uniquely defines equivalence relations

$$IND(B) = \{(x,y) \in U^2 : a(x) = a(y) \text{ for every } a \in B)\} \quad (4)$$

The family of all equivalence classes of the equivalence relation IND (B) is denoted U/IND(B). With every $X \subseteq U$, and $B \subseteq A$ we associate two sets defined as follows:

$$\underline{B}X = U\{Y \in U/IND(B): Y \subseteq X\} \quad (5)$$

is the lower approximation or positive region of X, and

$$\overline{B}X = U\{Y \in U/IND(B): Y \cap X \neq \phi\} \quad (6)$$

is the upper approximation or possible region of X.

A set $BN_B(x) = \overline{B}X - \underline{B}X$ will be called B-boundary of X.

If $X \subseteq U$ is given by a predicate P and $x \in U$, then

3.1.1. $x \in \underline{B}X$ means that x certainly has property P,

3.1.2 $x \in \overline{B}X$ means that x possible has property P,

3.1.3. $x \in U \backslash \overline{B}X$ means that x definitely does not have property P.

The area of uncertainly extends over $\overline{B}X \backslash \underline{B}X$, and the area of certainty is $\underline{B}XU - \overline{B}X$, respectively.

3.2 Rough Membership Function

The rough membership function quantifies the degree of relation overlap between the set X and the equivalence $[x_B]$ class to which x belongs. It is defined as follows:

$$\mu_X^S : U \to [0,1] \text{ and } \mu_X^B(x) = \frac{|[x]_B \cap X|}{|[x]_B|} \quad (7)$$

where S = (U, B, V, f) is an information system and $\phi \neq X \subseteq U$.

3.3 Core and Reduction of Attributes

The set of attributes in the information system describes objects in the system. We use the set of attributes to classify the objects into classes that are determined by anywhere from one to several attributes. Sometimes it happens that the set of attributes provides more information about objects that is needed to distinguish them. In such a situation some of the attributes may be reduced without loosing the ability to group the objects.

So we can say that attributes $a \in B$ is superfluous in B, if $IND(B - \{a\}) = IND(B)$, otherwise it is indispensable.

If all attributes $a \in B$ are indispensable in B. So the set of all indispensable attributes in B will be called the core of B, and will be denote CORE(B). The core is a set of attribute such that the removal of a single attribute impacts on the original ability to distinguish the objects. The core of attributes is the intersection of all reductions.

3.4 Decision Rules

We can be obtained the decision rules from an information system or a reduction of condition attributes. Some condition values may be unnecessary in the decision rules, so we can remove them from the rules or change them into don't care values. The process of conditional values removed is called values reduction.

4. SET UP OF OBJECT ATTRIBUTES

To apply the rough sets to the character recognition process, we have to set up the object's attributes from the ring projection vectors. In this paper we get data from the point that $r = 1,2,3,\ldots,50$ (total of 50 values) for attributed $a_1, a_2, a_3,\ldots, a_{50}$ respectively. So, we have 50 attributes for each character at this stage.

5. CLASSIFICATION USING ROUGH SETS

We propose a two-stage pattern classification method based on the rough ring projection. The rough ring projection values are computed from the set of invariant Thai characters. Fig. 3 shows the procedure of our proposed two-stage classification method.

5.1 The Coarse Classification

From all sets of attributes using the same characters, we find the maximum value of attributes $a_1, a_2, a_3,\ldots, a_{50}$ and also to find the minimum value of the attributes. We use these sets of maximum and minimum values of attributes as the boundaries for the course groups of those characters are shown in Fig. 4. There are some characters that can not be completely separated, using these sets of conditional attributes, so we must combine them together into a same class. From grouping, we have 27 course invariant character groups. In the decision rules, the answer is considered from the matching of the number attributes which are on the boundaries of an attributes set. After the first step classification, the groups have only one member. This classification is the final recognition. Most of the results in the first step are not good enough for the aim of clustering because this cannot separate the some character in each coarse class. It is necessary to separate these characters by the 2[nd] level classification.

5.2 The Fine Classification

In the second step we must classify characters in the same class. We still use the same method as in section 5.1, instead of applying the projection to an explicit ring area of similar characters in the same class as shown in Fig. 5.

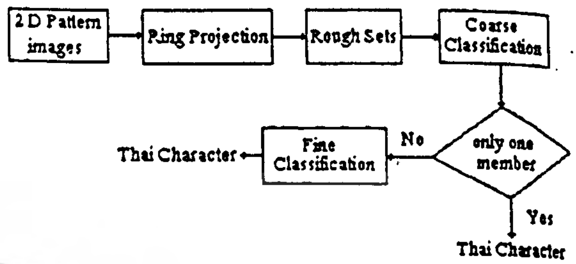


Fig. 3. Diagram of the two-stage classification method.

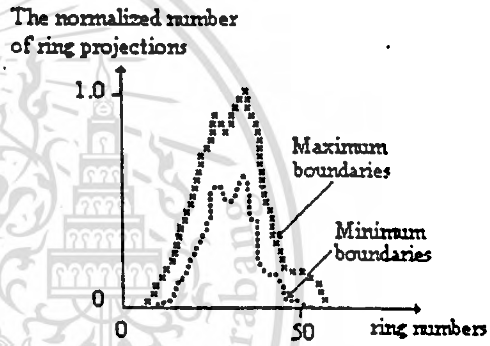


Fig. 4.. The attribute boundaries of course group 1 which compose the Thai characters " ฑ, ฒ, and ฌ " respectively.

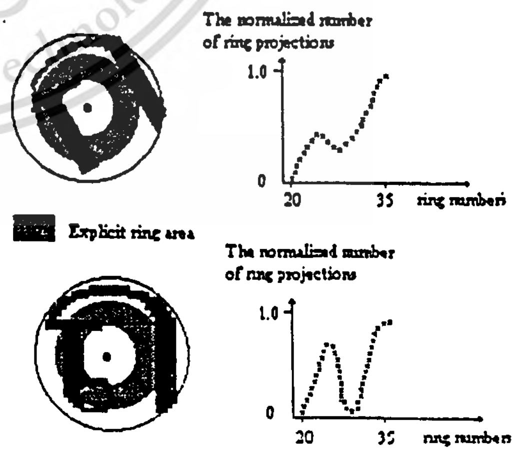


Fig. 5. Explicit rings of ring projections

6. EXPERIMENTAL STUDY AND RESULTS

In our experiments the learning sets consist of 19,936 training characters (four fonts, four sizes in each font, and 14 different orientations in each size). The largest characters have 28x28 pixels. The smallest characters have 14x14 pixels. The decision rules can be obtained directly from a set of attributes, and consist of fifty condition attributes based on the total pixel distribution in a ring projection of each character. We use 27 rules to classify the unknowns into course groups and use the 2nd classification to classify the similar characters in the second step. The unknowns used in the experiment are 14,952 Thai characters tested with 3 fonts that have 4 different sizes, and 14 rotations for each size.

The results obtained are shown in Table 1. From the results we can see, invariant character recognition is obtained with above 95 % accuracy from images with size between 28x28 and 22x22 pixels. The performance of the model decreases slightly for smaller characters. Fig. 6 shows the example rotated image of letters "ก", and "ข" in the character data set of experiment.

The comparison of the recognition rate of rough ring projection method with the other methods is shown in Table 2. From the results in Table 2, the recognition rate of the matching method, and Karhunen-Loeve method is very high, but they cannot to solve the invariant character problem. The elliptic Fourier descriptors and genetic neural networks, and rough ring projection use to solve the invariant character problem. The recognition rate of rough ring projection was slightly greater than that in the case of using the elliptic Fourier descriptors and genetic neural networks.

The noise tolerance and sensitivity of ring projection are studies experimentally. The random noises are generated by changing the value of pixels in the image in a uniform random manner as shown in Fig. 7. Noisy images are recognized correctly by our method as well. Furthermore, if the head of characters were defect (such as broken head or solid head), this particular method of analysis can be used. The defect head is an important problem of Thai character recognition. The method in references [1, 11, 12] can not be use in situations with defect head. Fig. 8 shows the Thai characters that have the defect head or solid head.

7. CONCLUSION

In this experiment, we use the set of attributes to generate the decision rules about the characters on the basis of training examples. A rough rule bases system is used to classify the characters into coarse classes. The 2nd classification is used to classify the characters in each coarse class from the other. A new development method is used to recognize the character invariance to size, translation, and rotation respectively. This method takes the advantage of properties of ring projection, and the rough rule bases. We evaluate the system accuracy with

Table 1. Recognition results.

Font Name	Size	Training sets: Recognition rate (%)	Test sets: Recognition rate (%)
AngsanaUPC	14	100.0	93.3
	18	100.0	94.4
	22	100.0	96.6
	28	100.0	98.9
CordiaUPC	14	100.0	93.3
	18	100.0	93.3
	22	100.0	95.5
	28	100.0	97.8
EucrosiaUPC	14	100.0	94.4
	18	100.0	95.5
	22	100.0	96.6
	28	100.0	98.9
Average recognition		100.0	95.7

Table 2. The comparison of the recognition rate of rough ring projection with other methods.

Recognition methods	Accuracy rate (%)
Matching Model [9]	98.0
Karhunen-Loeve [10]	98.2
Elliptic FD and GNN (invariant) [2]	94.0
Rough ring projection (invariant)	95.7

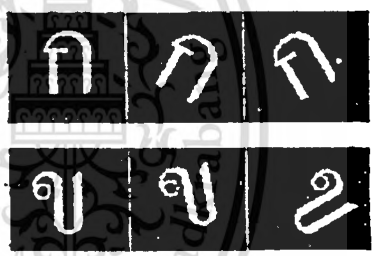


Fig. 6. The example of testing images of Thai characters "ก", and "ข".

Fig. 7. A noisy image of Thai character "ก".

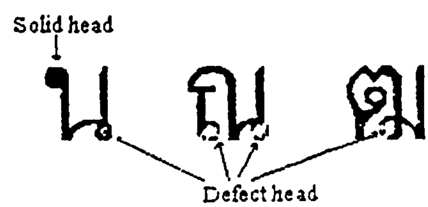


Fig. 8 The example of the defect head or solid head of Thai characters.

the Thai characters that have different fonts, different sizes, different orientations, and including some noise added randomly or a defect head of character. When testing the system with training samples, it results in 100 % accuracy.

The results when applied the system to the unknown characters show an average recognition rate of 95.7 % accuracy. This correspondence shows that the rough sets can be applied to invariant Thai character recognition with good results.

8. REFERENCES

[1] Pisit Phokharatkul, and Chom Kimpan., "Handwritten Thai character Recognition Using Fourier Descriptors and Genetic Neural Networks.", The Fourth Symposium on Natural Language Processing 2000., May 10-12, 2000, Thailand ,pp. 108-123.

[2] Pisit Phokharatkul, Ukrit Marang, and Chom Kimpan, "Recognition of Printed Thai Characters Using Elliptic Fourier Descriptors and Genetic Neural Networks.", Proceedings 2000 Asia-Pacific Symposium on broadcasting and communication December 21-23, 2000, pp. 406-411.

[3] Yuan Y. Tang, Bing F. Li, Hong Ma, and Jimming Liu, "Ring-Projection-Wavelet-Fractal Signatures: A novel Approach to Feature Extraction.", IEEE Transactions on Circuit and Systems-II: Analog and Digital Signal Processing., Vol. 45, No. 8, August 1998, pp. 1130 – 1134.

[4] L. A. Torres - Mendez, J. C. Ruiz - Suarez, Luis E. Sucar, and G.Gomez., "Translation, Rotation, and Scale-Invariant Object Recognition.", IEEE Transactions on Systems, Man, and Cybernetics-Part c: Applications and Reviews, Vol.30, No.1 , February 2000, pp. 125-130.

[5] Alireze Khotanzad, and Yaw Hau Hong, "Invariant Image recognition by Zernike Moments." IEEE Transactions on Pattern Analysis and Machine Intelligence., Vol. 12, No. 5, May 1990, pp. 489-497.

[6] S.K. Pal, and A. Skowron., Rough Fuzzy Hybridization A New Trend in Decision-Making., Springer, 1999.

[7] Dorota Nejman. "Rough Sets in Handwritten Numerals Recognition", ICA WUT Reports on Rough set, 1995.

[8] Daijin Kim, and Suan-Yamg Bamg, "A Handwritten Numeral Character Classification Using Tolerant Rough set.", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.22 , No.9. September 2000, pp.923-937.

[9] C. Kimpan, A. Itoh, and K. Kawanishi, "Recognition of printed Thai characters using a matching method", Proc. IEE, vol.130, Pt.E, No.6, Nov. 1983, pp. 183-188.

[10] C. Kimpan, A. Itoh, and K. Kawanishi, "Fine classification of printed Thai characters using the Karhunen-Loeve expansion", Proc. IEE, vol.134, Pt.E, No.5, Nov. 1987, pp. 257-264.

[11] S. Airphaiboon, M. Sangworasil, B. Attachoo, and C. Kimpan, "Recognition of handwritten Thai characters considering the head of character.", IEEE international Conference on Image Processing, Vol.2, Singapore, pp. 326-330, 1989.

[12] S. Airphaiboon and S. Kondo, "Recognition of handprinted Thai Characters Using Loop structures.", IEICE Trans. INF.&SYST., Vol.E79-D, No.9, pp. 1296-1304, Sep 1996.

Computational Intelligence
An International Journal

Intelligence informatique
Une revue internationale

11 October 2001

Pisit Phokharatkul, Chom Kimpan
Department of Computer Engineering
Faculty of Engineering
Research Center for Communications and Information Technology
King Monkut's Institute of Technology
Ladkrabang, Bangkok 10520
Thailand

Dear Drs. Phokharatkul and Kimpan:

I am happy to inform you that your paper entitled "Handwritten Thai Character Recognition using Fourier Descriptors and Genetic Neural Networks" has been accepted for publication in *Computational Intelligence: an International Journal.* We will send you the copyright forms to complete from the editorial office soon.

Your paper will be one of three papers selected from over 40 papers presented at SNLP'2000 to form a special issue of *Computational Intelligence*. There will undoubtedly be one or two regular papers in the special issue as well. I believe your paper should appear in early 2002 and will let you know as soon as possible the exact issue.

Congratulations and thank you for submitting your manuscript to *Computational Intelligence*. If we have any problems with the electronic version of the paper we will inform you immediately. Best wishes.

Yours Sincerely,

Nick Cercone, co-Editor, *Computational Intelligence*

University of Waterloo

Nick Cercone, Professor & Past Chair
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

email: ncercone@uwaterloo.ca
tel: 519-888-4567 x6816
fax: 519-885-1208

Computational Intelligence

HANDWRITTEN THAI CHARACTER RECOGNITION USING FOURIER DESCRIPTORS AND GENETIC NEURAL NETWORKS

PISIT PHOKHARATKUL [1] AND CHOM KIMPAN [1]
Department of Computer Engineering, Faculty of Engineering,
Research Center for Communications and Information Technology,
King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand.

This paper presents a method to solve the rolated and scaling character recognition problem using Fourier descriptors and genetic neural networks. The contours of character image are extracted, and separated between the outer contour and inner or loop contours. The loop contours are a special characteristic of Thai characters: called the head of the character. The special features of Thai characters (loop contours) are used in the rough classification stage, and Fourier descriptors with genetic neural networks are used in the fine classification stage. The Fourier descriptors detect the outer contour of a character and it is fed to network. These features are recognized by a multilayer neural network. Genetic algorithms (GAs) are utilized to help compute the weights of the neural network optimally and reduce uncertain states in the neural networks output. Experimental results have shown that the combination of the Fourier descriptors with genetic neural networks, loop features and local curvature charateristics of similar characters are powerful tools in successfully classifying Thai characters. The recognition rate by this method is 99.12% for 1,200 examples of handwritten Thai words (a total of 13,500 characters) written by 60 persons.

Key words: character recognition, Fourier descriptors, genetic neural networks, contour invariant, concavity, convexity.

1. INTRODUCTION

In research of the topic of Thai character recognition in the past seventeen years, many approaches have been tried to make computers recognize the characters. The preliminary stage of this field, which concerns the use of the matching method was described by Kimpan, Itoh, and Kawanishi (1983). Subsequently, the topological properties (Kimpan 1986) are used to improve recognition accuracy. In 1987, the same author proposed the recognition method of printed Thai charaters using the Karhunen-Loève expansion (Kimpan et al. 1987). Hiranvanichakorn, Agui and Nakajima (1985) used local features of Thai characters, such as convexity and concavity of stroke in their recognition system. Aipaiboon et at. (1989, 1996) used the heads of Thai characters to recognize the handwritten Thai characters. Phokharatkul and Kimpan (1998) used the cavity features of each character based on neural networks for solving the problem in the recognition by considering the head of the Thai characters. Although, the recognition rates of these methods were satisfactory, they have the rotated problem (except for the local features method which does not have the rotated character problem: it has low accuracy rate).

Handwritten character recognition is a difficult problem due to the great variations of writing styles, and the different size and orientation angle of the characters. In this reserch proposes a method of recognition of Thai characters using the Fourier descriptors and genetic neural networks for solving the problem of rotated and scaling character recognition. Persoon and Fu (1977), Mahmoud (1994), Jeong (1999) used the Fourier descriptors for solving the

[1]Address correspondence to the authors at the Research Center for Communications and Information Technology, King Mongkut's Institue of technology Ladkrabang, Bangkok, 10520, Thailand., Email: egpph@mahidol.ac.th, kkchom@kmitl.ac.th, chom@cpe.rsu.ac.th

problem of retated characters, but this cannot be used for the scaling problem. The Fourier descriptior can also be used for detcction of boundary features from a character image. The boundary representation is essential in shape description, and recognition. The high quality boundary shape boundary representations obtained using the low frequency coefficients are the ones of greatest interest. The boundary information Fourier descriptors are classified by genetic neural networks. The motivation of embedding genetic algorithms in the proposed hybrids has two goals:

- The network learning stagnation can be avoided.
- Uncertain states in the neural network outputs can be reduced.

The recognition procedure will be illustrated in the following section.

2. THAI CHARACTERS AND WORDS

Thai characters consist of 42 consonants. 18 vowels, 4 controled voice tones, and 3 special symbols as shown in Figure 1. As a result, 87 fundamental characters (including Thai numerals and Arabic numerals) are obtained.

The structure of most Thai characters consists of small loops (heads of character) combined with curves and lines. The heads appear in various locations of each character. By counting the number of Thai characters, we can categorize three groups of Thai characters.

- No-head characters: eg. "ก", "ธ", "า".
- One-head characters: eg. "ง", "ค", "ว", "ภ".
- Two-headed characters: eg. "น", "ม", "ฉ", "ห".

Writing a Thai character always starts with a head (if the character has a head or loop) followed by the other component. Most Thai characters consist of only one stroke, except "ศ", "ต", and "+".

ก ข ค 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

consonants

ะ า ิ ี ึ ื ุ ู เ แ โ ํ ใ ไ ฤ ฦ ั ็

vowels

่ ้ ๊ ๋

controled voice tones

์ ๆ ฯ

special letters

FIGURE 1. Thai character set consists of 42 consonants, 18 vowels, 4 controled voice tones, and 3 special letters.

A Thai language sentence, such as that in Figure 2, is composed of consonants, vowels,and tonal symbols on different levels. The vertical level can be divided into four parts as shown in Figure 3.

In general, a Thai word is noncursive in writing (printing). This mean that there is no connected part between two consecutive characters in a word. This suggests easy segmen-

นับตั้งแต่โครงการจัดตั้งสำนักวิจัยการสื่อสารและ
เทคโนโลยี่สารสนเทศได้ก่อตั้งขึ้นเมื่อปี พ.ศ. 2540

FIGURE 2. Example of Thai language sentence. Translation: Since research center for communications and information technology has been in B.E. 2540.

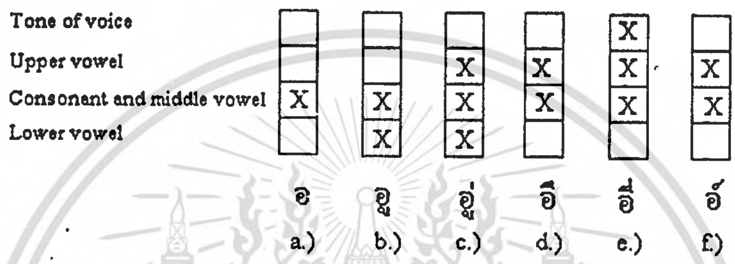


FIGURE 3. Composition of Thai character word or sentence. The Thai language sentences are composed of consonants, vowels and controlled voice tones, on different levels.

tation of each character from a word. Unfortunately, the Thai word has a complicated line level structuring, as shown in Figure 2. Overlapping of characters will occur when a word consists of consonants, vowels, and tonal symbols, as shown in Figure 2. From Figure 2, the vowel " ั" overlaps with both consonants: eg. “น” and “บ”, “ต” and “ง”, “จ” and “ค, “น” and “ก”, “จ” and “ย”.

Tonal symbols are located on the tonal line level. However, when a word has no upper vowel characters, the location of tonal symbols will be shifted down to the position of the upper vowel line level.

The difficulty of Thai character recognition is the problem of similar characters. The characters “ท,ฑ”, “ฤ,ภ”, “ถ,ภ,ย,”, “ด,ค,ต”, “ฆ,ณ”, and “ฉ,ฉ” are examples of similar characters. The differences are the stroke of head, and the fork point of the characters, which lead to the problem of recognition, this will be briefly described in the SPECIAL TECHNIQUES section.

3. THAI CHARACTER RECOGNITION SYSTEM

A typical character recognition system consists of the following three stages as shown in Figure 4.

The preprocessing stage involves noise removal, smoothing borders of character, stroke thickness normalization, skew correction, slant normalization, character image segmentation, and contour-following respectively.

In the second stage, the Thai character was classified into three rough classes using the number of inner contours. Then, the Fourier coefficients of the outer contour are computed using Fourier descriptors and trained by the genetic neural networks.

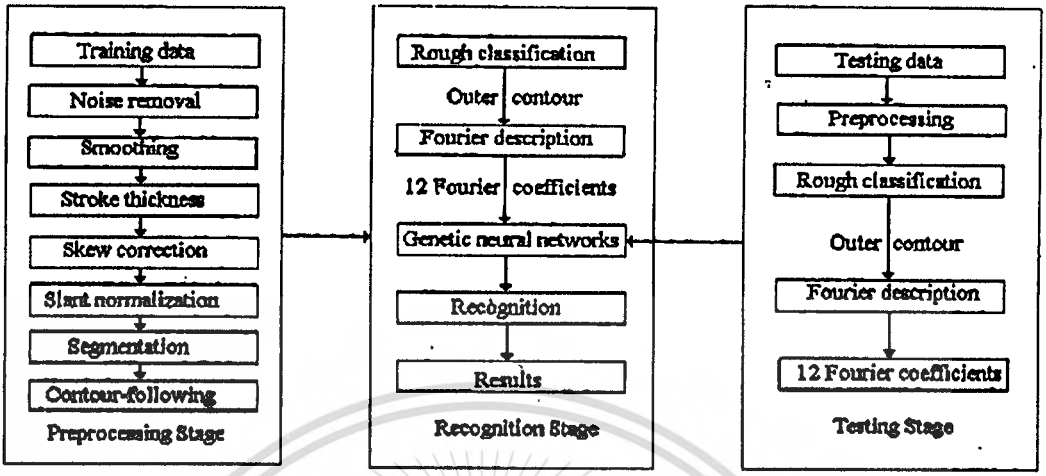


FIGURE 4. Flowchart of the recognition system.

In the testing stage, testing data is fed through the preprocessing stage and contours of the character are extracted. Then, we classify the characters using the inner contour and compute the Fourier coefficient of the outer contour. Finally, the twelve Fourier coefficients activate the trained neural network.

4. CONTOUR INVARIANT IN DIGITAL IMAGES

We will describe the property of invariance to digital images (XIA 1995). Discussing the contour invariant provides an universal theory for 2D digital images which is completely independent of neighborhood used. The basic definition on discrete geometry is defined as follows.

Definition 4.1: Let P_{i-1} , P_i , and P_{i+1} be three consecutive contour pixels and separate its j -neighbors into object side and background side. Let $\angle P_{i-1}P_iP_{i+1}$ be the interior angle (in objects side) at P_i. Local curvature at P_i $LC(P_i)$ represents the complementary angle of $\angle P_{i-1}P_iP_{i+1}$ as shown in Figure 5. Note that X pixels are interior points.

The value of local curvature $LC(P_i) \geq 0$ if $\angle P_{i-1}P_iP_{i+1}$ is convex (Figure 5 (a)) or $LC(P_i) < 0$ when $\angle P_{i-1}P_iP_{i+1}$ is concave (Figure 5 (b)). As local curvature in Figure 6 is an example of different local curvature in 8-neighbors. By the definition of contour, LC may have discrete values {-2, -1, 0, 1, 2, 3, 4}. In the Figure 6, X pixels designate object/contour pixels, a circle pixel stands for the current edge points, and the numeric in center represents LC at these points.

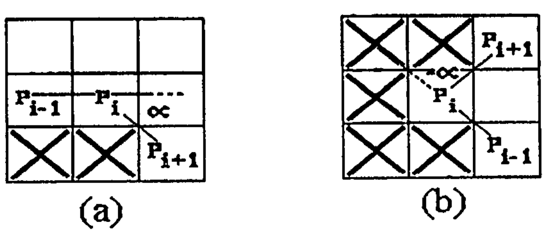


FIGURE 5. Local curvature in 8 - neighbors.

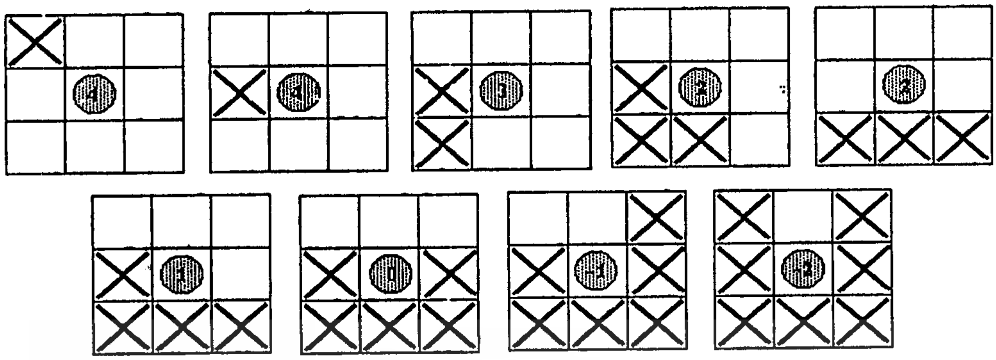


FIGURE 6. Configuration of different local curvatures.

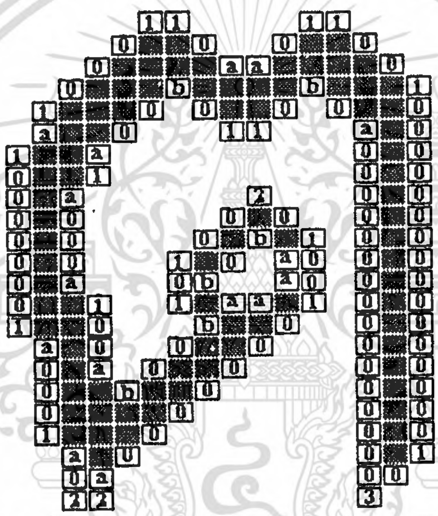


FIGURE 7. Component with distinguished outer and inner contours. The contour codes 0, 1, 2, 3, 4, a, and b are coded by the definition of contour invariant.

Definition 4.2: Total curvature $TC(\partial C)$ is the sum of local curvature LC on a contour ∂C and denoted by $\sum_i LC(P_i | P_i \in \partial C)$.

Corollary 4.3: In digital images, $TC(\partial C^+) = -TC(\partial C^-) = N$, where N equals 8 for rectangular tessellation and 6 for hexagonal tessellation.

In this research, digital boundary tracing of characters is made and the traced boundary is coded by the definition of contour invariant. The definition of the contour invariant is used to obtain the local curvature codes of each Thai character. The definition of contour invariant is independent of contour-tracing orientation. Figure 7 illustrates the resulting images in which a and b represent local curvature values -1 and -2 on contour pixels. The local curvature of contour invariant is used for smoothing and to compute the concavity and convexity of boundary in the next section respectively.

5. PREPROCESSING

5.1. Isolated Points Removal

Isolated points (Figure 8) in a binary image are most probably caused by noise, so any 3x3 mask to apply to any image is one that would remove such points. A mask is passed over the image to decide whether pixels should be removed or added. A black pixel is removed when the sum of all eight neighbors have to be 0. Similarity a white pixel is removed (that is a black pixel is added) when it matches with a mask in Figure 8(b). Note that X pixels are black points and 0 pixels are white points.

O	O	O
O	X	O
O	O	O

(a) Isolated black pixel.

?	X	?
X	O	X
?	X	?

(b) Isolated white pixel.

FIGURE 8. A 3x3 mask for the removal of isolated pixel. In the mask, the letter '0' stands for a white or background pixel, 'X' stands for a black or foreground pixel and '?' can take either value black or white.

5.2. local Smoothing of Character Image Borders

In some cases, we might want to remove or add one or two pixels to the character image borders. The algorithm that smoothes the border of characters performs two operations: removing and filling. Removing is performed when the border is locally convex. It clears some border pixels in order to remove the local convexity. Filling copes with local convexity. In Figure 9 (a), we wish to remove and add the consecutive pixels that are circled. The smoothing operations on the border of a character image are performed by applying the contour-following algorithm (Richard and Peter 1973) and the contour invariant of character images traces the external boundary points. We use the invariant codes of the image to detect local curvature noise: such as convex (01b10, and 01aa10 codes) and concave (0a2a0, 0a11a0 codes) in the character image borders. Local curvature noise is removed or added when it matches one of the situations described in Figure 10. Thus, we define a simple smoothing operation, to remove or add one or two pixels to external boundary contours of character image, using the contour invariant codes.

5.3. Stroke Thickness Normalization

Stroke thickness normalization is computed as defined in (Schürmann 1992). First, it is assumed that each character is written with a fixed stroke width, or the stroke width is the same throughout the entire image. Second, we assume that each character shape can be modeled as a line with length l and width w. Then the number of black pixels P is equal to l x w and the circumference C of the shape is equal to 2 (l + w). From the preprocessing stage, the values of P and C are already known for all connected components. Solving these equations, the stroke width, w, of the character model can be readily determined.

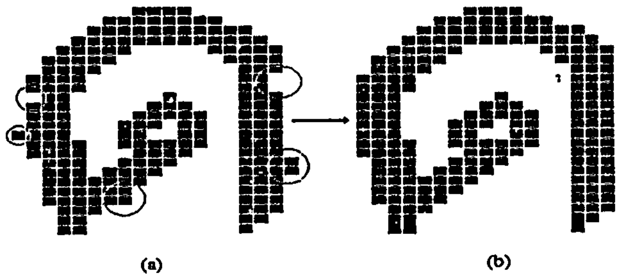


FIGURE 9. (a) Original image. (b) Smoothed image.

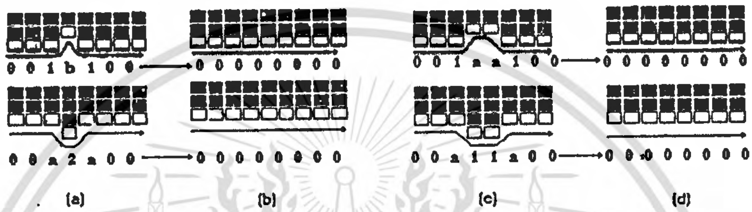


FIGURE 10. (a) and (c) original, (b) and (d) smoothed.

5.4. Skew Correction for Handwritten Thai Character Recognition

Skew correction is one of the most essential preprocessing steps in character recognition. The base line of a handwritten document is irregular and varies. Depending on the writer, it is difficult to find the base line. The problem that we consider in this section is how to know x-y coordinate information. We use these coordinates to rotate the handwritten document image to its correct angle. The skew correction procedure has two stages as follows.

Finding the lowest pixel in each character.. The Thai word is noncursive in writing or there is no connected part between two consecutive characters in a word, as shown in Figure 2. From Figure 11, the characters ม, เ, and ก have the lowest location of pixels at positions a, b, and c respectively. Then, we can find angle θ_1 and θ_2, and use them to compute the mean skew correction angle $(\theta_1 + \theta_2)/2$.

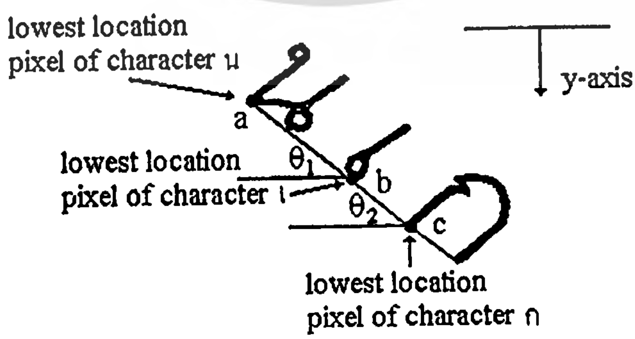


FIGURE 11. The pictorial meaning of the skew correction concept.

There are two search sequences used in finding the lowest point in each character:

- giving the priority to the left position
- giving the priority to the right position.

And there are two cases to consider when finding the lowest point on the base line for each character (Figure 12):

- the image is tilted clockwise
- the image is tilted counterclockwise.

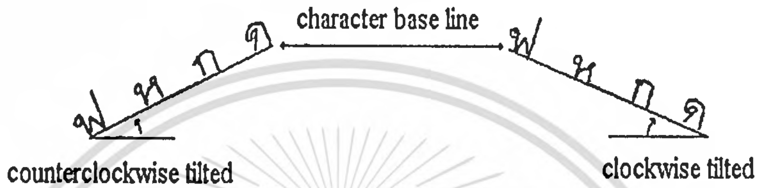


FIGURE 12. Different tiltings of the character base line.

The process of finding lowest point is defining a point-searching sequence. The sequence uses the masks as shown in Figure 13 (a). The X in the middle is the starting point of a border contour which we will call the initial black pixel (1 pixel), and the point number 1, 2, 3, 4, and 5 are searched in that order to find the next black point of the border contour.

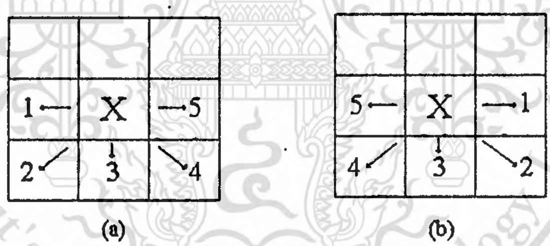


FIGURE 13. The searching sequences. (a) For giving priority to the left position. (b) For giving priority to the right position.

The example of the searching algorithm for finding the lowest point, by giving priority to the left position, can be summarized as follows.

Repeat

check for a black point at the next position
if found and it is not past the initial point of the before state,
move to the next position
else check for a black point at the southwestern position
if found move to the southwestern position
else check for a black point at the next lower point
if found move to the next lower position
else check for a black point at the southeastern position
if found move to the southeastern position
else check for a black point at the eastern position
if found move to the eastern position

until there is no black point at the next left (except past the initial point of the before state), southwestern, lower, southeastern, or eastern position.

Similarly, we use the process in Figure 13 (a) to modify finding the lowest point by giving priority to the right position. The searching sequence in this case is shown in Figure 13 (b).

Computing the rotating angle from the lowest pixels. From Figure 14, the highest point of the image is located by scanning for black points from left to right and top to bottom. Then, the lowest points on the base line for each character procedure are computed as the following:

1. Locate the highest black point of the image.
2. Locate the lowest point by giving priority to the left position and by giving priority to the right position, then choose the higher of these two points.
3. If a black point in the next left position is found then move to the left position (which is still in the same character)
   else move to next right position (which is still in the same character).
4. To locate the lowest point for the next character do
   - Step 1: Find the first black point to the right (to the left for the counterclockwise tilted image case) on the same horizontal line, store its position, and locate the lowest point from there.
   - Step 2: From the position stored in step 1, move to the next black point to the right (to the left for the counterclockwise tilted image case) and locate the lowest point from there.

   The lowest base line point for the next character is whichever lowest point found in step 1 and 2 that has the higher y-coordinate (the lower y-coordinate for the counterclockwise tilted image case).
5. repeat 4 until the end of line.

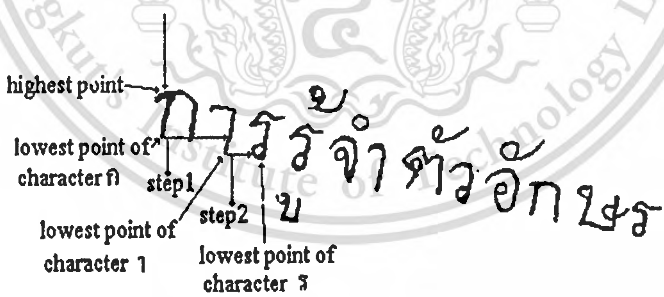


FIGURE 14. An example to find base line of clockwise tilted image.

We can compute the rotating angle from the base line, which depends on the lowest points of each character in the Thai words. After rotation normalization, we can extract each character from the words using the segmentation method.

5.5. Slant Normalization

Slant correction is the process which tries to normalize the slant of the handwriting to the vertical. Some people write with the words slanted to the right of the vertical, while others write more or less slanted to the left. So, each pixel of the original image is corrected by a slant normalization algorithm (Bozinovic 1989). The slant-corrected image is obtained by applying the following transformation to all foreground pixels with coordinate points x, y in the original image:

$$x' = x - y \times \tan(b - def), \qquad y' = y \tag{1}$$

where def is a parameter specifying the default (normal) slant.

5.6. Segmentation of Thai Characters From the Sentence.

The segmentation of Thai characters from the sentence (Airphaiboon and Kondo 1996) must be done before the decision of the unknown pattern process, because the construction of the Thai word, as shown in Figure 15 (a), is different from an English word: it is more complex. But the Thai word is noncursive and has no connected part between two characters in a word. Then, it is easy to produce segments each character from a word. The procedure of segmentation is applied to extract each character (CS, UV, MV, LV or TS) from a given word. The algorithm is based on histogram analysis and contour tracing. An example of this algorithm is shown in Figure 15.

FIGURE 15. Single-character segmentation is base on histogram analysis and contour tracing.

Figure 15 (a) shows a complicated Thai word which consists of three CSs ("ว", "ม", "ร"), one UV ("ิ"), one MV ("ไ"), one LV ("ุ"), and two TS ("่", "้"). Using the average value of the horizontal projection profile of this word, the component line level is obtained as shown in Figure 15 (b). In this step, the upper part of the middle vowel "ไ" is deleted because it is located in the upper vowel line (the horizontal projection profile value is less than the average). To restore the deleted parts, a contour tracing algorithm is applied, starting from one pixel (X_s, Y_s) and going to a pixel (X_e, Y_e) of each deleted part (Figure15 (b)).

For each character in the consonant line level, UV, TS, and LV are searched, and their locations are determined in the upper vowel, tonal and lower vowel line levels, respectively (Figure15 (c)).

In an orientation case, we cannot use a segmented character as mentioned above, because some characters may be digital images shifted or rotated from the horizontal or vertical line as shown in Figure 16. So, we must scan digital images in a left to right and top to bottom direction. If the pattern of each character is found, then we pull out one character from the other. To obtain a full scan we continue to separate characters until the end of the digital images. With each character from the orientation case, it is not necessary to normalize the rotation and scaling of the character, because the Fourier descriptors can solve these problems, as shown in equation (4) and (5) respectively.

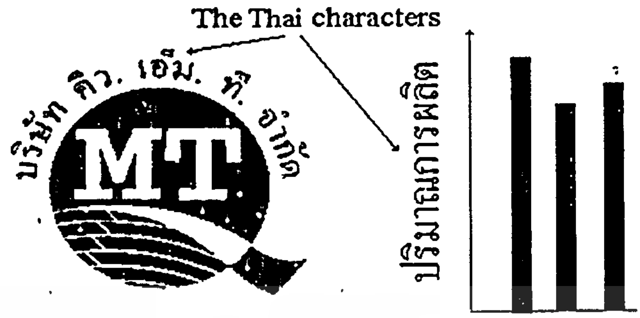


FIGURE 16. An example of the orientation character.

6. FEATURES EXTRACTION AND CLASSIFICATION

6.1. Features Extraction

Inner contours (loop contours) extraction.. We applied an edge detection algorithm (Anil 1989) and a contour-following algorithm (Richard and Peter 1973), and obtained the detection of a Thai character's head. Figure 17 (a) shows a normal pattern for inner contour (loop contours) extraction strategies. Using an edge detection algorithm, an edge of the character appears as shown in Figure 17 (b). Later, an external edge is erased by the contour-following algorithm. The internal edge appears, which we refer to as the " head " of the character, as shown in Figure 17 (d).

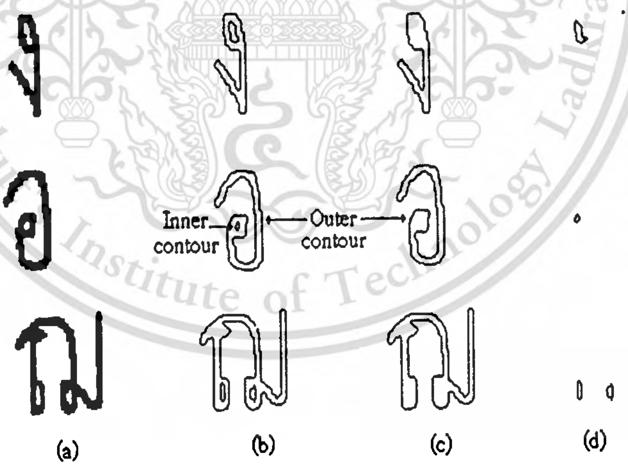


FIGURE 17. Examples of feature extraction.
(a) Original image of handwritten Thai characters "ง", "อ", and "ฌ".
(b) Outer and inner contour of handwritten Thai characters "ง", "อ", and "ฌ".
(c) Outer contour of handwritten Thai characters "ง", "อ", and "ฌ".
(d) Inner contour of handwritten Thai characters "ง", "อ", and "ฌ"

Outer contour extraction based on Fourier descriptor.. The Fourier descriptor has been used as the popular curve descriptor in many applications (Persoon and Fu 1977, Mahmoud 1994, Jeong 1999). It is the method of describing a closed curve by a set of Fourier coefficients.

Because a character contour is also a closed curve, the character contour can be represented by Fourier descriptors. The detail of the Fourier descriptors is discussed in Section 7.

6.2. Classification

The recognition system was separated into two stages. First, in the rough classification stage, the loop contours of a character from the previous section were classified into 3 classes using the number of loop contours such as class 1, class 2, and class 3. These are the groups of characters as shown in Table 1, which consist of no loop, one loop, and two loops respectively. Second, in the fine classification stage, the outer contour information from Fourier descriptors is classified by a genetic neural network.

Table 1. Three classes in Thai alphabet

Level / Class	Consonant	Vowel			Tonal
		Upper	Middle	Lower	
1	ก ธ		า		+ ่
2	ข ค ง จ ซ ช ฑ ด บ ป ผ ฝ ห ฟ ภ ย ต ถ ท ร ว ศ ส อ ล ฤ ฦ ๆ ฯ ญ ฐ	ั ิ ี ํ ็	เ แ ไ โ	ุ ู	๋ ๊ ่ ์
3	ฆ ฉ ณ ฌ ฎ ฏ ฒ น ม ษ ฬ พ ธ	ื	ะ ฯ		

7. FOURIER DESCRIPTORS

Fourier descriptors in this system as shown in Figure 18. The character boundary was presented by co-ordinates $P_m(X_m, Y_m)$ and computed

$$a[k] = \frac{1}{L}\sum_{m=1}^{L} x[m]e^{-jk(2\pi/L)m} \tag{2}$$

$$b[k] = \frac{1}{L}\sum_{m=1}^{L} y[m]e^{-jk(2\pi/L)m} \tag{3}$$

Where $a[k]$, $b[k]$ are the coefficient of Fourier descriptor,
L is the curve length,
m is 1 to L, and
k is 0 to $L-1$.

Then, to cut the DC component of the Fourier descriptor $a[0]$ and $b[0]$, the other data can be used to independently perform the position of the character boundary. Computed by

$$r(k) = \sqrt{|a[k]|^2 + |b[k]|^2} \tag{4}$$

Where k is 1 to $L-1$, $|a[k]|$ and $|b[k]|$ are absolute value, and complex numbers. The $r[k]$ are independent on both the translation, and rotation of the character. In the case of the application for independent scaling work, the Fourier descriptors $S[k]$ are developed to be equation 5,

$$S[k] = r[k]/r[1] \tag{5}$$

The first 10 - 15 descriptors $s[k]$ are sufficient for character description.

Fourier descriptors $s[k]$ are symmetrical around position $(L\text{-}1)/2$. The $s[l]$ of every character is closely matched. For an application to be recognized we neglected $s[l]$, and used the first half of the Fourier descriptors $s[2]$ to $s[(L-1)/2]$. Because of this consideration, Fourier descriptors of character graphs, $s[k]$ are reduced to near zero when k increases to more than 15 as shown in Figure 18. The lower order coefficients are the high quality boundary shape representation. We can select the first order coefficients 10 - 15 descriptors for sufficient character recognition. However, since some Thai characters are very close in shape, the use of 10 Fourier descriptors gives better results.

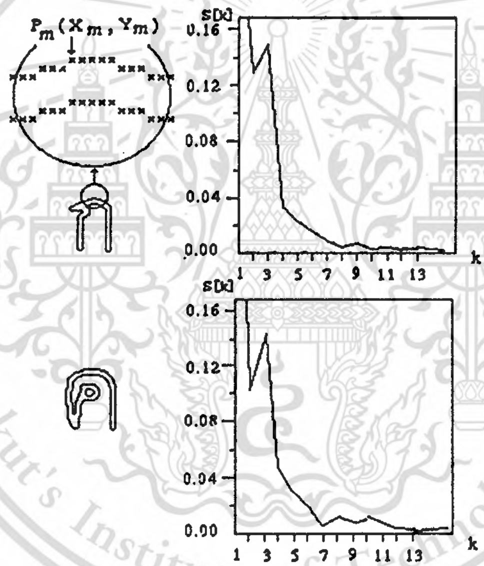


FIGURE 18. Fourier descriptors of 'ก' and 'ค'.

8. GENETIC NEURAL NETWORKS

In this paper, genetic neural networks are utilized to classify the handwritten Thai characters in each group. The details of this approach are given in the following subsections.

8.1. Genetic Algorithms

Genetic algorithms (GAs) (Goldberg 1989) are general-purpose search techniques based on principles inspired from the genetic and evolution mechanisms observed in natural systems and populations of living beings. Their basic principle is the maintenance of a population of solutions to a problem (genotypes) in the form of encoded information individuals that evolve in time. The evolution is based on the laws of natural selection (survival of the fittest) and genetic information recombination within the population. The evolving population samples the search space and accumulates knowledge about good and bad quality areas recombining the knowledge to form solutions with optimal performance to the specific problem. Figure 19 illustrates the schematic diagram of the genetic search process.

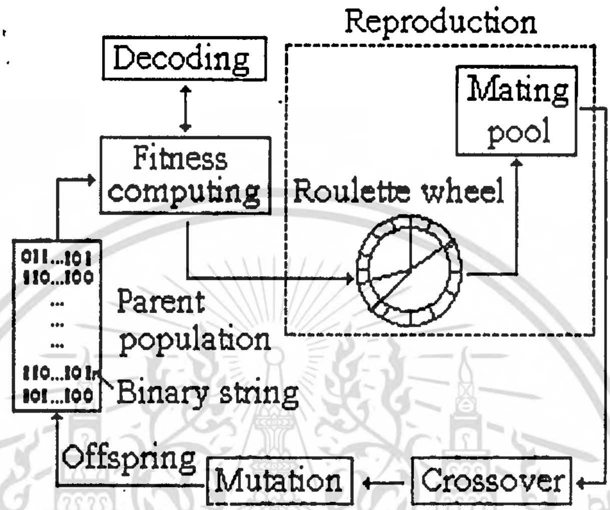


FIGURE 19. Schematic diagram of the genetic search process.

In genetic algorithms, each parameter is represented by a string structure. This is similar to the chromosome structure in natural genes. A group of strings is called a population. Strings in a given population when mutated and/or crossover evolve a new generation of strings. For each generation, all the populations are evaluated based on the fitness. The individual (string) with higher fitness has more chances of evolving to the next generation. The procedure is repeated many times until it arrives at a predetermined ending condition. A GA is usually implemented according to the following step:

- *Step 1* Coding of Searching Space
  The searching space is coded into a finite-length bit string. Different strings represent different candidates for optimizing. The length of the string will determine the range to be searched by GA.
- *Step 2* Initial Generation
  An initial population of N strings with length n is randomly generated, where n is selected by step 1. Large N will increase the possibility of including the optimal solution in less generation but will decrease the computation speed of GAs.
- *Step 3* Fitness Evolution
  In each generation, every string is decoded according to its meaning in the original system. An objective function is used to evaluate them and assign them with fitness values.
- *Step 4* Reproduction
  Individual strings are copied according to their fitness values: those with a higher probability of contributing one or more offspring in the next generation. Once a string has

been reproduced, it will enter into a mating pool as a tentative new population for further generation action.

- *Step 5* Crossover
  After reproduction, members of the newly reproduced string in the mating pool are mated at random. Each pair of strings exchanges their corresponding portion of binary strings at a randomly selected position between 1 and n - 1
- *Step 6* Mutation
  This is a complementary operator that increases the variability of the population. After the crossover operation, each bit position of each string in the new population undergoes a random change with a probability equal to the mutation rate. A low level of mutation will prevent loss of important natures or of some potentially useful generation materials, and also allow a search close to a point in the search space.
- *Step 7* Iteration
  Step 3 to 6 are run iteratively until a predetermined ending condition is satisfied.

The acceptable solution is obtained and decoded into its original pattern from the resulting binary string.

8.2. Neural Networks

A neural network is modeled as massively parallel-interconnected networks of elementary processors or neurons. Various software algorithms are crafted to synthesise a mapping between input and output variables by learning a set of the weights and neuron thresholds from training samples. Supervised training has mostly been formulated as a weight training process. Effort is made to find an optimal or near optimal set of connection weights for a network according to some criteria. The total mean square error (MSE) between the actual output and the target output is used to guide BP search in the weight space. There have been some successful applications of BP algorithms in various areas. However, drawbacks with the BP algorithm do exist owing to its gradient descent nature. It often gets trapped in a local minimum of a function which is vast, multimodal and non-differentiable. One way to overcome the shortcomings of BP as well as other gradient descent search-based training algorithms, is to consider the training process as the evolution of connection weights towards an optimal or near optimal set defined by a fitness function. From such a point of view, global search procedures like GAs can be used effectively to train a GA-NN. The fitness of a GA-NN can be defined by the aforementioned total MSE.

8.3. Genetic Neural Network Training

Genetic algorithms have been used to search the weight space of the multilayer feedforward neural network without the use of any gradient in formation (Lin and Lee 1996). The existing GA and NN have been combined to form the hybrid system. The system model is shown in Figure 20.

The basic concept of GA-NN is as follows. A complete set of weights is coded in a number string, which has an associated "fitness" indicating its effectiveness. The GA object makes calls to NN objects and passes messages to the fitness function. The procedure will start with a random population of such strings. Successive generations are constructed using genetic operators to construct new strings out of old ones such that fitter strings (weights) are more likely to survive and to participate in crossover operations. The crossover operation can in principle bring together good building blocks such as hidden units that compute certain logical functions found by chance in different members of the population. Both NN and GA instances are created at the beginning of the optimization procedure and last until the end, which optimizes the NN parameters including weights and bias. The example of encoding a network on a chromosome, is shown in Figure 21. The weights and biases in a neural network are encoded in order, as a list.

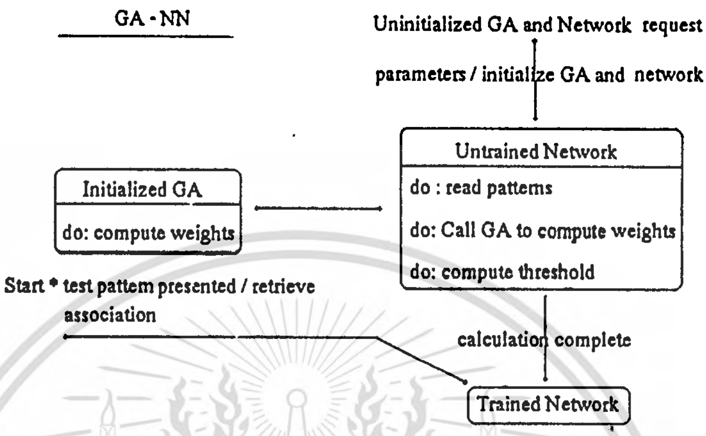


FIGURE 20. The GA-NN model.

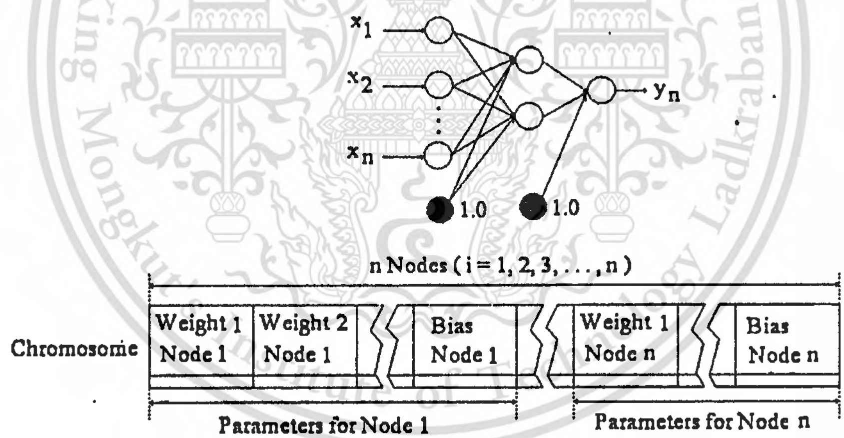


FIGURE 21. Encoding a network on a chromosome.

In Thai character recognition, the multilayer neural networks are fed by 12 coefficients of Fourier descriptors to the input nodes. The hidden layer of the network has a maximum of 25 neurons and the output layer has 8 neurons, corresponding to the number of recognized characters. The number of parameters is 533, with 4 bytes to represent each parameter. Each individual will take up 2132 bytes. With a population size of 150, the memory size is 319,800 bytes. The required computational time in the training stage is much greater than that for conventional BP neural networks. In the experiment, the neural networks are operated in two modes: training, and testing. In the training mode, a set of training data is used to adjust the weights of the network interconnections. The training neural networks are determined by these weights. In testing mode, the trained network is activated by the testing data.

9. SPECIAL TECHNIQUES

After training for the classification by genetic neural networks, some of the Thai characters are similar, and cannot be classified any more at all. It is necessary to use special techniques for recognizing some characters such ค,ด ,which are similar. The only difference is with the heads of the characters, which leads to the problem of recognition. The difference is that in ค there is an outer curl, but in ด there is an inner curl. In the case of character ค, ต, the only difference is in the upper part of a character. In classes 2 and 3 of Table 1, several characters cannot be separated by genetic neural networks, because these characters are similar as mentioned above (see in Figure 22). To classify the similar character in these groups, some local characteristics of Thai characters are required. The local characteristics of Thai characters are defined by a new concept for finding concavity and convexity from contour invariants as follows.

FIGURE 22. The example of similar Thai characters. (a) The similar character in class 2. (b) The similar character in class 3.

We applied a contour-following algorithm to detect the external contour of the similar characters and compute the invariant codes by using the definition of contour invariant in Section 4. From a contour invariant code of each point Ci may have discrete values { -2, -1, 0, 1, 2, 3, 4} as illustrated in Figure 23 (a).

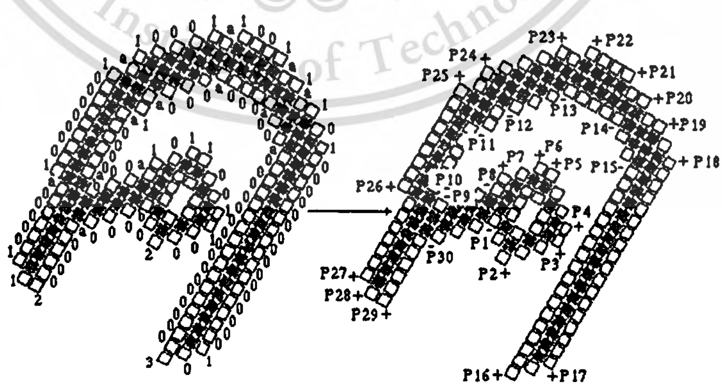


FIGURE 23. (a) The contour invariant codes of a Thai character "ค" rotated to 30° and (b) An example of assignment a Thai character in figure 20(a) with + and -.

Then, S_i is represented as in equation (6).

$$S_i = \{b, a, 0, 1, 2, 3, 4\} \tag{6}$$

Where

S_i is a sign of an each point C_i ,
a and b represent local curvature values -1 and -2 on contour pixels respectively.

Accordingly, each C_i is given a value of the seven signs S_i. The points are then utilized to detect + and − vertices of a contour as follows.

Definition 9.1: When a point C_i satisfies either condition (a) or (b), a label + is assigned to C_i

(a) $(S_i > 0)$ & $(S_{i-1} > 0$ or $S_{i+1} > 0)$
(b) $(S_i > 0)$ & $(S_{i-1} \leq 0)$ & $(S_{i+1} \leq 0)$ & $(D_{i,i+1} \geq k_1)$ & $(D_{i-1,i} \geq k_2)$

where $D_{i,j} = \sqrt{(C_{xi} - C_{yj})^2 + (C_{yi} - C_{yj})^2}$ and k_1 and k_2 are thresholds.

Definition 9.2: When a point C_i satisfies either condition (a) or (b), a label - is assigned to C_i .

(a) $(S_i < 0)$ & $(S_{i-1} < 0$ or $S_{i+1} < 0)$
(b) $(S_i < 0)$ & $(S_{i-1} \geq 0)$ & $(S_{i+1} \geq 0)$ & $(D_{i,i+1} \geq k_1)$ & $(D_{i-1,i} \geq k_2)$.

An example of assignment for a Thai character with + and - according to the definitions is shown in Figure 23 (b).

When there are successive + vertices between two - vertices, the successive + vertices are called a convex arc, and a concave arc is the successive - vertices put between two + vertices.

As shown in Figure 23 (b), the segment passing points P_1, P_2, $\cdots$, P_8 and P_{15}, P_{16}, $\cdots$, P_{30} are regarded as two convex arcs having as the start points P_1 and P_{15}, P_8 and P_{30} as the end points respectively. Further, the segment passing points P_7, P_8, $\cdots$, P_{16} and P_{29}, P_{30}, $\cdots$, P_2 are two concave arcs having P_7 and P_{29} as the start points, P_{16} and P_2 as the end points respectively.

Let $P_j = \{P_{xj}, P_{yj}\}$; $j = 1$ - J be a sequence of + and - vertices belonging to an arc in the x - y plane. Then, length of arc $D_{j,j+1}$ is defined as follows.

$$D_{j,j+1} = \sqrt{(P_{x,j+1} - P_{x,j})^2 + (P_{y,j+1} - P_{y,j})^2} \tag{7}$$

The convex and concave arcs of a similar one-head character is calculated in three steps as shown in Figure 24 (a).

1. Obtain the centroid of a head of a Thai character.
2. From the centroid scan around the head, find the - vertices or concave arc neighbouring the outer contour, and measure the distance r_i between the - vertices and centroid.
3. Use the - vertices point, which has a shortest distance r_i as a starting point for contours following in a contour counterclockwise direction, detect the concavity and convexity of contours for classification.

The convex and concave arcs of the similar two-head character is calculated in four steps as shown in Figure 24 (b)

1. Calculate the centroid of character.

2. Calculate the distance R_1 and R_2.
3. Calculate an angle θ (R_1 is the first distance arm of angle always in a counterclockwise direction)
4. Apply the contour-following algorithms between the points r_1 and r_2 to find the convex and concave arcs in counterclockwise direction.

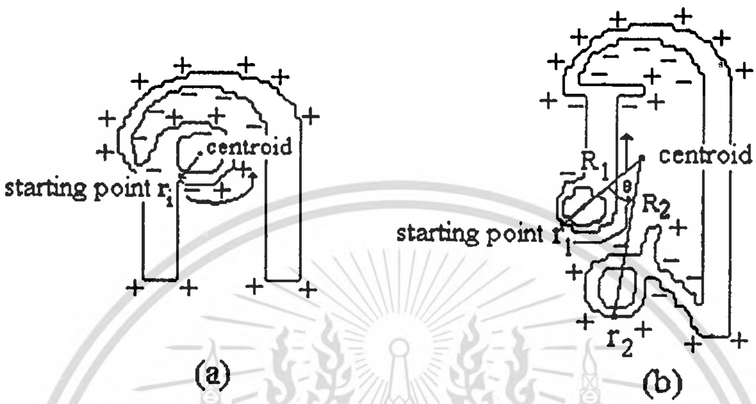


FIGURE 24. An example of extracting convexities and concavities.
(a) A similar one-head character "ค".
(b) A similar two-head character "ฌ".

By this method, we can classify the characters that are similar as shown in Figure 25. In this figure, the Thai characters "ค", "ด", and "ฅ" are similar, but they have the different sequence of convexity and concavity. Note that the convex arcs and concave arcs of a character are independent of contour rotation.

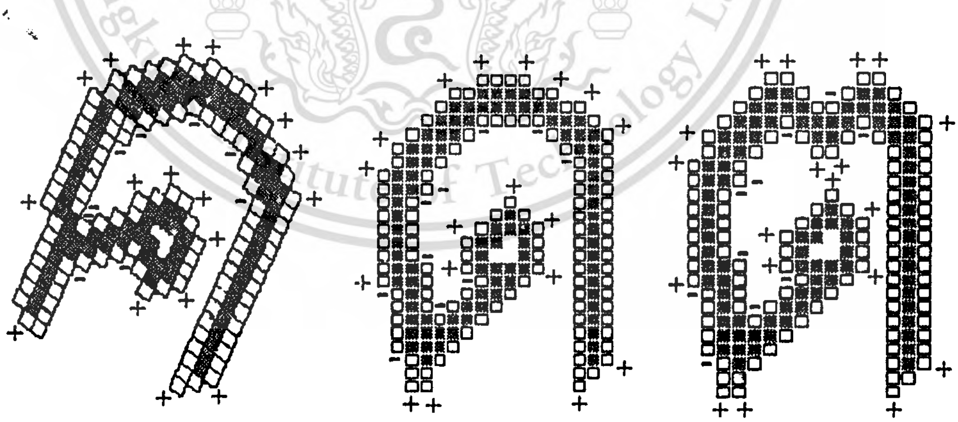


FIGURE 25. An example of extracting convexities and concavities in similar Thai characters.
(a) The handwritten Thai character "ค" rotated 30°.
(b) The normal handwritten Thai character "ด".
(c) The normal handwritten Thai character "ฅ".

10. EXPERIMENTAL RESULTS

The character classifiers were trained on a set of 1,350 Thai word samples written by 70 persons (a total of 14,550 characters) extracted from training image. A subject was requested to write a Thai word on the given sample form. In the sample, two writing conditions are given:

- Please use a black ball point pen or 2b pencil.
- Please write a Thai word in clear writing.

FIGURE 26. Correctly recognized sample of Thai words.

The training and experimental images were scanned in binary form at a resolution of 300 dpi. The experimental process was applied to 1,200 Thai words written by 60 persons (a total 13,500 characters). The Thai words, which have various scripts such as clear writing, slant writing, rotated characters (skewed image), and scaling characters are shown in Figure 26. Sometimes, a character image might be skewed with respect to the horizontal axis (depending on handwriting). We need to separate each character from the word by the segmentation method in Section 5. We detect the outer contour of the character image, and the inner contours according to the method in Section 6. After that the characters are divided into three groups by rough classification using the number of inner contours (the head of the character) of the characters. Then the outer contour of the character image is transformed using Fourier descriptors. We found that the graphs of the Fourier coefficients do not depend on the rotation of the character: as rotated in different angles 10°, 45°, and 90°. All are similar as shown in Figure 27.

Graphs of the different sized characters are shown in Figure 28. In this figure, the graphs of the Fourier descriptors of the contour of character "ร" is almost the same, because the contour of the character image is still the same size and is not a changeable ratio. We brought the contour information of the character transformation using Fourier descriptors into the input nodes of the GA-NN. We used learning by genetic neural networks for finding the weight of the networks for the recognition system.

In the case of the similar characters, we used the SPECIAL TECHNIQUES in Section 9 for helping to improve the recognition rate. We tested the word recognizer on a clear writing character set, some samples of which are shown in Figure 29. For classification purposes, it should be noted that the people who wrote the characters in the testing database are the same who wrote the characters in the training database. The recognition rate is nearly 99% (assuming the segmentation of the characters are corrected) as shown in Table 2. The comparison of the recognition rate of Fourier descriptors and the genetic neural networks

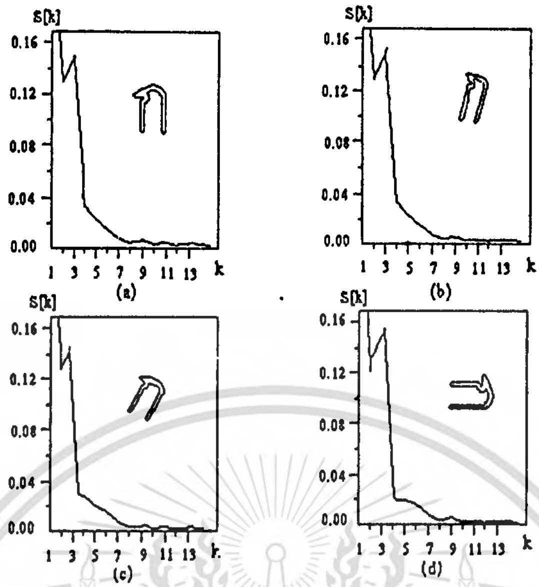


FIGURE 27. The results of Fourier descriptors from the character "".
(a) The normal character.
(b) The character "ก" rotated to 10°.
(c) The character "ก" rotated to 45°.
(d) The character "ก" rotated to 90°.

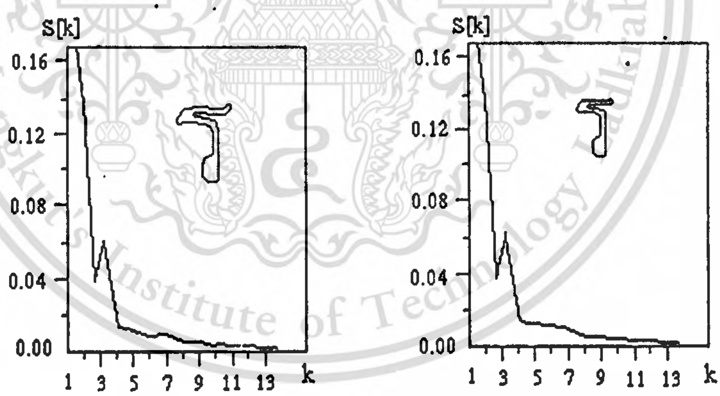


FIGURE 28. The result of the Fourier descriptors from Thai character "ร" with different sizes.

with the other methods is shown in Table 3. In Table 3, the recognition rate of the cavity features based on the neural network, Karhunen-Loève, and loop structure is very high. If the characters are rotated and changed in size, this particular method gives a low recognition rate. The local features have no rotation problem, but this method gives a recognition rate which is very low. Comparing the Fourier descriptors and genetic neural networks is better, because this method can solve the rotation (not using skew correction) and scaling problems (Figure 16), with nearly 99% correct recognition.

ตัวอย่างตัวอักษรลายมือเขียนภาษาไทย
ตัวบรรจงที่ใช้ในการทดลองกับระบบการรู้จำ

คณะวิศวกรรมศาสตร์ สำนักวิจัยการสื่อสารและ
เทคโนโลยีสารสนเทศ

FIGURE 29. Some examples of handwritten Thai character (clear writing) from testing database.

Table 2. Recognition accuracy.

Sample	Recognized	Mis-recognized	Rejected
1200	99.12%	0.23%	0.65%

Table 3. Comparison of the recognition rate between the Fourier descriptors and genetic neural networks with the other methods.

Recognition method	Percent correct
Local feature	88.9%
Cavity feature & neural net	98.3%
Loop structure	99.0%
Karhunen-Loève (printed characters)	98.8%
Fourier descriptors & genetic NN	99.12%

11. CONCLUSIONS

In this paper, we described an efficient approach to invariant character recognition that utilizes the number of head and a Fourier descriptor with genetic neural network for the classification phase, and uses the local curvature characteristic in the similar character case. If a character is rotated we cannot use the position of head and shape to recognize it, but Fourier descriptors can describe the shape of these rotated characters. From the experimental results, the graphs of Fourier descriptors are independent with rotation, translation position and the scaling of characters. The advantage of this method gives a high recognition rate. The data of Fourier descriptors from contour information of characters are learned by the genetic neural network. The genetic algorithms optimize the weight interconnections, and reduce uncertain states in the neural network output. After the fine classification stage, some characters are completely recognized, but some similar Thai characters cannot be separated from each other at this stage. It is necessary to use local curvature characteristics for recognizing. These have different sequences of concavity and convexity, so we can separate them. Furthermore, we can add various styles of Fourier descriptors into the recognition system. Script writing and defects of inner contours give a very low recognition rate because characters can be mis-grouped. Some mis-recognized characters are caused by preprocessing errors or writing errors, which causes the total rate of recognition to be low. The mis-recognized samples are shown in Figure 30. The disadvantage of this method are that if the inner loop or head of character cannot be detected, if there is unclear writing, and solid head or blurring, and this particular method of analysis cannot be used.

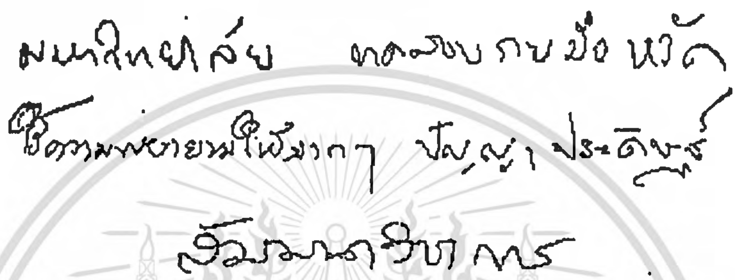


FIGURE 30. Mis-recognized samples.

REFERENCES

Airphaiboon, S. and Kondo, S. 1996. Recognition of handprinted Thai Characters Using Loop structures. IEICE Trans. INF.& SYST., **E79-D**(9): pp. 1296-1304.

Airphaiboon, S. Sangworasil, M. Attachoo, B. and Kimpan C., 1989. Recognition of handwritten Thai character considering the head of character. IEEE international Conference on Image Processing, **2**: pp. 326-330.

Anil K, 1989. Fundamentals of digital image processing, Prentice-Hall international editions, pp.342-425.

Bozinovic, R. M. and Srihari, S. N. 1989 Off-line cursive script word recognition. IEEE trans. PAMI, 11(11): pp. 68-83.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Company.

Hiranvanichakorn, P. Agui, T. and Nakajima, M. 1985. A recognition method of handprinted Thai characters by local features. IEICE Trans., **E68**(2): E68, pp. 83-90.

Jeong, S.C. and Jeoug, D.S. 1999. Hand-written digit Recognition Using Fourier Descriptors and contour Information. The 1999 IEEE TENCON., **II**: pp. 1283-1286.

Kimpan, C. 1986. Printed Thai character recognition using topological properties method. INT. J. Electronics, **60**(3): pp.303-329.

Kimpan, C. Itoh, A. and Kawanishi, K. 1986. Recognition of printed Thai characters using a matching method. Proc. IEE.,**130**(Pt.E.6): pp. 183-188.

Kimpan; C. Itoh, A. and Kawanishi, K. 1987. Fine classification of printed Thai character recognition using the Karhunen-Loève expansion. Proc. IEE, **134**(Pt. E.5): pp. 257-264.

Lin, C. T. and Lee, C. S. G. 1996. Neural Fuzzy Systems. Printice-Hall International, Inc., pp. 399-404.

Mahmoud, S. A. 1994. Arabic Character Recognition Using Fourier Descriptors and Character Contour Encoding., Pattern Recognition, **27**(6): pp.815-824.

Persoon, E and Fu, K. S. 1977. Shape Descrimi-nation using Fourier Descriptors., IEEE Transactions on systems, Man, and Cybernetics., SMC-7(3): pp.170-179.

Phokharatkul, P and Kimpan, C. 1998. Recognition of handprinted Thai characters Using the cavity features of character based on neural network.,The 1998 IEEE Asia-Pacific Conference on Circuit and Systems., pp. 149-152.

Richard, D. D. and Peter, E. H. 1973. Contour Following, Pattern Classification and scene Analysis., A Miley-Interscience Publication, pp. 290-293.

Schürmann, J. 1992. Document analysis-from pixels to contents. Preceeding of the IEEE,80(7): pp. 1101-1119.

XIA, F. 1995. On a New Basic Concept and Topological Invariant. Image Analysis and Processing 8^{th} International Conference, ICIAP' 95, pp.341-346.

AUTHOR BIOGRAPHY

Mr. Pisit Phokharatkul was born in Uttaradit, Thailand. He received Bachelor Degree in Physics from Srinakharinwirot University in 1981, B. Econ. (Financial Economics) from Ramkhamheang University in 1984, Grad. Dip. (Nuclear) and M.Eng. (Nuclear) from Chulalongkorn University in 1982 and 1985, and M.Eng. (Electrical) from King Mongkut's Institue of Technology North Bangkok in 1990.

He is currently an Assistance Professor at Department of Electrical Engineering, Faculty of Engineering, Mahidol University Salaya, Nakhorn Pathom. His research interests in Pattern recognition, Computer vision, Fuzzy and Rough sets, and Intelligence control system.