

การใช้โครงข่ายรีเลชันนิวรอลเน็ตเวิร์กสำหรับการรู้จำ  
ตัวอักษรพิมพ์ภาษาไทยแบบฟอนต์ไม่มีหัว

CROSS-CORRELATION NEURAL NETWORK FOR  
THAI PRINTED CHARACTERS WITH NO HEAD FONTS RECOGNITION

พฤกษา ดวงphasuk  
PRUEGSA DUANGPHASUK

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2549

ISBN 974-15-2811-4

การใช้โครงข่ายรีเลชันนิวรอลเน็ตเวิร์คสำหรับการรู้จำ  
ตัวอักษรพิมพ์ภาษาไทยแบบฟอนต์ไม่มีหัว

CROSS-CORRELATION NEURAL NETWORK FOR  
THAI PRINTED CHARACTERS WITH NO HEAD FONTS RECOGNITION

พฤกษา ดวงผาสุข  
PRUEGSA DUANGPHASUK

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต  
สาขาวิชาเทคโนโลยีสารสนเทศ  
บัณฑิตวิทยาลัย  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
พ.ศ.2549

ISBN 974-15-2311-4

**CROSS-CORRELATION NEURAL NETWORK FOR  
THAI PRINTED CHARACTERS WITH NO HEAD FONTS RECOGNITION**

**PRUEGSA DUANGPHASUK**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY  
SCHOOL OF GRADUATE STUDIES  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**2006**

**ISBN 974-15-2311-4**

**COPYRIGHT 2006**

**SCHOOL OF GRADUATE STUDIES**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

หัวข้อวิทยานิพนธ์	การใช้โครสคอร์รีเลชันนิเวรอลเน็ตเวิร์คสำหรับการรู้จำตัวอักษรพิมพ์ภาษาไทยแบบฟอนต์ไม่มีหัว
นักศึกษา	นางสาว พกฤษา ดวงผาสุข
รหัสนักศึกษา	44067036
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
พ.ศ.	2549
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ. ดร. อาริต ธรรมโน

### บทคัดย่อ

งานวิจัยที่เกี่ยวข้องกับการรู้จำตัวอักษรภาษาไทยที่ผ่านมานุ่งเน้นในการเพิ่มความสามารถในระบบการรู้จำตัวอักษรภาษาไทยด้วยการพิจารณาโครงสร้างลักษณะเด่นของตัวอักษรที่มีรูปแบบตัวอักษรมาตรฐานตามคุณลักษณะตัวอักษรภาษาไทยแบบดั้งเดิมที่ประกอบไปด้วย เส้นตรง เส้นโค้ง เส้นซิกแซก และการเริ่มต้นตัวอักษรด้วยลูปวงกลมหรือที่เรียกว่าหัวตัวอักษร ส่วนกลุ่มรูปแบบตัวอักษรที่โครงสร้างของตัวอักษรปราศจากส่วนหัวตัวอักษรเช่น “ก” “ง” “ด” “ต” “บ” “ป” “ข” “ช” “ซ” จากตัวอย่างดังกล่าวตัวอักษรปราศจากส่วนหัวตัวอักษรนั้นจะเห็นว่าตัวอักษรมีรูปแบบคล้ายกันมากขึ้น และปัญหาที่ตามมาคือความสามารถในการรู้จำตัวอักษรภาษาไทยรูปแบบตัวอักษรที่ไม่มีหัวยังไม่ถูกต้องเท่าที่ควร

ดังนั้นวิทยานิพนธ์นี้จึงได้นำเสนอแนวคิดวิธีการใหม่โดยใช้การแบ่งแยกตามลำดับชั้นในโครงข่ายประสาทเทียม Hierarchical Normalization Cross-Correlation ARTMAP สำหรับการรู้จำรูปแบบตัวอักษรพิมพ์ภาษาไทยที่ไม่มีหัว และนำเอาหลักการของโครสคอร์รีเลชันมาใช้เป็นฟังก์ชันสำหรับการวัดค่ามาตรฐานความคล้ายคลึงระหว่าง Input Vector กับ Weight Vector ข้อดีของการใช้โครสคอร์รีเลชันคือสามารถวัดค่าความคล้ายคลึงกันของรูปแบบในแต่ละส่วนออกมาได้โดยไม่ขึ้นกับขนาด เพราะเป็นการเปรียบเทียบความเหมือนในแต่ละส่วนของตัวอักษร โดยไม่จำเป็นต้องมีขนาดเริ่มต้นที่เท่ากัน กลุ่มข้อมูลที่ใช้ในการ Train และ Test ประกอบด้วยรูปแบบตัวอักษรพิมพ์ภาษาไทยไม่มีหัว 39 แบบและตัวอักษรบนภาพจากป้ายทะเบียนรถไทย 506 ป้าย ผลการทดลองที่ได้จากการรู้จำด้วยวิธีใหม่ที่น่าเสนอโดยเฉลี่ยเท่ากับ 88.49% นอกจากนี้ยังนำกลุ่มข้อมูลไปทดสอบกับโปรแกรม AmThai, ThaiOCR และ Fuzzy ARTMAP

<b>Thesis Title</b>	Cross-Correlation Neural Network for Thai Printed Characters with No Head Fonts Recognition
<b>Student</b>	Ms. Pruegsa Duangphasuk
<b>Student ID.</b>	44067036
<b>Degree</b>	Master of Science
<b>Programme</b>	Information Technology
<b>Year</b>	2006
<b>Thesis Advisor</b>	Assoc. Prof. Dr. Arit Thammano

## **ABSTRACT**

The relevant researches concerned with Thai Character Recognition in the past have been emphasized in ability enhancement in the recognition rate by considering the structure of the standard character pattern based on the Feature Characteristic. Thai characters which are composed of direct lines, curves, zig-zag lines and the beginning of writing with the head circle loop which is known as the head of character. The examples of Thai Characters without the head circle loop are “ก” “ท” “น” “ด” “ต” “ข” “ช” “ส” “ฮ” “ษ” “ศ”. From these examples, it can be seen that each single character is much more similar to each other. The following problem is that the ability for recognition rate of non-head-loop is still not on the right way.

Therefore, this thesis presents the new methodology utilizing the level classification and Hierarchical Normalization Cross-Correlation ARTMAP for recognizing Thai Characters that have pattern without head circle loop. This research applied with the cross-correlation method which is used to be a criteria function to measure the similarity between the input vector and the weight vector. The advantage of using cross-correlation function is that the similarity of the pattern in each part can be independently measured concerning the size of character because the method is to compare the similarity of each character regardless the same size of starting. The set of data used in training and testing sets compose of non-circle head loop 39 fonts and the characters from 506 vehicle license plates. The result from this new method is shown the average of 88.49 percent recognition rate. Moreover, the data set has been investigated by testing with the program ArnThai, ThaiOCR, and Fuzzy ARTMAP.

## กิตติกรรมประกาศ

การวิจัยนี้เสร็จสมบูรณ์ได้จากความช่วยเหลือของหลายๆ ฝ่าย โดยเฉพาะอย่างยิ่ง อาจารย์ที่ปรึกษา คือ รศ. ดร. อาริต ธรรมโน ผู้ซึ่งให้คำแนะนำ และชี้แนะแนวทางที่เป็นประโยชน์มาโดยตลอดในทุกๆ เรื่อง ผู้จัดทำรู้สึกขอบพระคุณเป็นอย่างสูง

ขอขอบคุณบิดา-มารดา ที่ให้ความอนุเคราะห์ทางด้านทุนทรัพย์ตลอดการศึกษา การทำวิจัย และคอยให้กำลังใจอยู่เสมอ

ขอขอบคุณเจ้าหน้าที่ฝ่ายต่างๆ ของคณะเทคโนโลยีสารสนเทศ และสำนักหอสมุดกลาง ที่ให้ความช่วยเหลือ และอำนวยความสะดวกในหลายๆ ด้าน ระหว่างการทำวิจัย

สุดท้าย ขอขอบคุณเพื่อนๆ ร่วมรุ่น IS 11 ภาคปกติ ที่ให้คำปรึกษาในหลายๆ เรื่อง ซึ่งทำให้ผู้จัดทำรู้สึกกระฉ่าง และแนวคิดของทฤษฎีหลายๆ ทฤษฎี รวมทั้งช่วยให้รู้สึกมีกำลังใจทำวิจัยเมื่อยามที่ต้องการ

พฤษภา ดวงผาสุข

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	2
1.3 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย.....	6
1.4 ขอบเขตการวิจัย.....	6
1.5 ขั้นตอนของการศึกษา.....	7
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	8
บทที่ 2 หลักการและทฤษฎีเบื้องต้นที่นำไปสู่แนวคิดของการวิจัย.....	9
2.1 แนวคิดในการรู้จำตัวอักษรภาษาไทย.....	9
2.2 งานวิจัยที่เกี่ยวข้อง.....	13
2.2.1 HMM Topology Selection for On-line Thai Handwritten Recognition.....	13
2.2.2 Discriminative Training for HMM-Based Offline Handwritten Character Recognition.....	13
2.2.3 The Clustering Technique for Thai Handwritten Recognition.....	14
2.2.4 An Object-Oriented Expert System for Thai Character Recognition.....	16
2.2.5 Recognition of Handprinted Thai Character Using the Cavity Features of Character Based on Neural Network.....	17
2.2.6 Off-line Handwritten Thai Characters from Word Script.....	18
2.2.7 Handwritten Thai Character Recognition Using Fuzzy Membership Function and Fuzzy ARTMAP.....	18

# สารบัญ (ต่อ)

หน้า

2.2.8	Experimental results of Using Rough Sets for printed Thai Character Recognition.....	19
2.2.9	Printed Thai Character Recognition using Fuzzy Rough Sets .....	20
2.2.10	On Thai Character Recognition.....	21
2.2.11	Thai OCR: A Neural Network Application.....	23
2.2.12	On-line Thai-English Handwritten Character Recognition using Distinctive Feature .....	24
2.2.13	On-line Thai Handwritten Character Recognition using Hidden Markov Model and Fuzzy .....	26
2.2.14	Online Thai Handwritten Character Recognition using Hidden Markov Models and Support Vector Machines .....	28
2.2.15	Character Recognition System for Cellular Phone with Camera .....	29
2.2.16	Multi-Modular Architecture Based on Convolutional Neural Network for Online Handwritten Character Recognition .....	32
2.3	ทฤษฎีโครงข่ายประสาทเทียมที่ใช้ในการรู้จำ.....	35
2.3.1	Fuzzy ARTMAP.....	35
2.3.1.1	ART-1.....	36
2.3.1.2	Fuzzy ART .....	37
2.3.1.3	ARTMAP .....	38
2.3.1.4	Fuzzy ARTMAP .....	38
2.3.1.5	Adaptive Hamming Net (AHN) .....	41
2.3.1.6	Gaussian ARTMAP.....	42
2.3.2	Simplified Fuzzy ARTMAP.....	43
2.4	ทฤษฎีเบื้องต้นของงานวิจัยนี้ .....	44
2.4.1	การสอดคล้องรีเลชัน (Cross-Correlation).....	44
บทที่ 3	การรู้จำรูปแบบแนวทางใหม่โดยใช้การสอดคล้องรีเลชัน .....	47
3.1	กระบวนการก่อนการรู้จำ.....	47

## สารบัญ (ต่อ)

	หน้า
3.2 กระบวนการรู้จำ .....	61
3.3 ตัวอย่างการคำนวณ .....	68
3.3.1 ข้อมูลใน Training Set.....	68
3.3.2 ข้อมูลใน Testing Set.....	97
บทที่ 4 ผลการทดลอง .....	104
4.1 ชุดข้อมูลในการทดลอง .....	104
4.1.1 ชุดข้อมูลภาพเอกสาร.....	104
4.1.2 ชุดข้อมูลภาพป้ายทะเบียนรถไทย.....	110
4.2 ผลการทดลอง.....	111
บทที่ 5 สรุปผลการวิจัย และข้อเสนอแนะ .....	119
5.1 สรุปผลการวิจัย.....	119
5.2 ข้อเสนอแนะในการทำวิจัยต่อไป .....	120
บรรณานุกรม .....	121
ภาคผนวก.....	125
ประวัติผู้เขียน .....	131

# สารบัญตาราง

ตารางที่	หน้า
1.1 การเปรียบเทียบลักษณะของตัวอักษรภาษาไทยในรูปแบบที่มีหัวและไม่มีหัว .....	3
1.2 ตัวอักษรภาษาไทยฟอนต์ไม่มีหัว .....	7
2.1 ความสามารถในการจัดกลุ่มของตัวอักษรที่คล้ายคลึงกันในแต่ละกลุ่ม ทั้ง 21 กลุ่ม .....	15
2.2 แสดง Q-Code และรูปแบบแต่ละแบบ .....	16
3.1 กลุ่มระดับตัวอักษรภาษาไทยที่แบ่งออกเป็น 3 กลุ่ม .....	48
3.2 การแทนค่ารหัสแทนเส้นทางของตัวอักษร “อ” .....	58
3.3 การแทนค่ารหัสแทนเส้นทางของตัวอักษร “ท” .....	59
3.4 การแทนค่ารหัสแทนเส้นทางของตัวอักษร “ส” .....	59
3.5 การแทนค่ารหัสแทนเส้นทางของภาพตัวอักษรใน Training Set ตัวที่ 1 .....	69
3.6 การแทนค่ารหัสแทนเส้นทางของภาพตัวอักษรใน Training Set ตัวที่ 2 .....	70
3.7 การแทนค่ารหัสแทนเส้นทางของภาพตัวอักษรใน Training Set ตัวที่ 3 .....	71
3.8 การแทนค่ารหัสแทนเส้นทางของภาพตัวอักษรใน Training Set ตัวที่ 4 .....	71
3.9 การแทนค่ารหัสแทนเส้นทางของภาพตัวอักษรใน Training Set ตัวที่ 5 .....	72
3.10 การแทนค่ารหัสแทนเส้นทางของภาพตัวอักษรใน Training Set ตัวที่ 6 .....	72
3.11 ค่า Weight Vector ที่ชั้น Hidden Layer ในการเรียนรู้ครั้งที่ 1 .....	74
3.12 ค่า Weight Vector ที่ชั้น Cluster Layer ในการเรียนรู้ครั้งที่ 1 .....	75
3.13 ค่า Weight Vector ที่ชั้น Hidden Layer ในการเรียนรู้ครั้งที่ 2 .....	78
3.14 ค่า Weight Vector ที่ชั้น Cluster Layer ในการเรียนรู้ครั้งที่ 2 .....	78
3.15 ค่า Weight Vector ที่ชั้น Hidden Layer ในการเรียนรู้ครั้งที่ 3 .....	82
3.16 ค่า Weight Vector ที่ชั้น Cluster Layer ในการเรียนรู้ครั้งที่ 3 .....	83
3.17 ค่า Weight Vector ที่ชั้น Hidden Layer ในการเรียนรู้ครั้งที่ 4 .....	87
3.18 ค่า Weight Vector ที่ชั้น Cluster Layer ในการเรียนรู้ครั้งที่ 4 .....	87
3.19 ค่า Weight Vector ที่ชั้น Hidden Layer ในการเรียนรู้ครั้งที่ 5 .....	91
3.20 ค่า Weight Vector ที่ชั้น Cluster Layer ในการเรียนรู้ครั้งที่ 5 .....	92
3.21 ค่า Weight Vector ที่ชั้น Hidden Layer ในการเรียนรู้ครั้งที่ 6 .....	96
3.22 ค่า Weight Vector ที่ชั้น Cluster Layer ในการเรียนรู้ครั้งที่ 6 .....	96
3.23 การแทนค่ารหัสแทนเส้นทางของภาพตัวอักษรใน Testing Set ตัวที่ 1 .....	98
3.24 การแทนค่ารหัสแทนเส้นทางของภาพตัวอักษรใน Testing Set ตัวที่ 2 .....	99

## สารบัญตาราง (ต่อ)

ตารางที่	หน้า
3.25 การแทนค่ารหัสแทนเส้นทิศทางของภาพตัวอักษรใน Testing Set ตัวที่ 3.....	99
4.1 ตัวอย่างรูปแบบตัวอักษรไทยที่ไม่มีหัวที่ใช้ในงานวิจัยครั้งนี้ทั้งหมด 39 ฟอนต์.....	104
4.2 ผลการทดสอบตัวอักษร 39 รูปแบบตัวอักษรด้วยโปรแกรม ThaiOCR และ AmThai.....	111
4.3 ผลการทดสอบป้ายทะเบียนรถไทย 506 ป้ายด้วยโปรแกรม ThaiOCR และ AmThai .....	113
4.4 ชุดข้อมูล Training Set และ Testing Set ในการทดลองทั้งหมด 16 ครั้ง .....	114
4.5 การทดลองในการรู้จำชุดข้อมูลภาพเอกสาร (อ้างอิงตามตารางที่ 4.4).....	114
4.6 ผลการทดลองในการรู้จำป้ายทะเบียนรถไทย (อ้างอิง Training Set ตามตารางที่ 4.4).....	115
4.7 ผลการทดลองในการรู้จำชุดข้อมูลภาพเอกสารที่แบ่งย่อยกรอบตัวอักษรออกเป็น 16 กรอบย่อย (อ้างอิง Training Set ตามตารางที่ 4.4).....	117
4.8 ผลการทดลองในการรู้จำชุดข้อมูลภาพเอกสารที่แบ่งย่อยกรอบตัวอักษรออกเป็น 4 กรอบย่อย (อ้างอิง Training Set ตามตารางที่ 4.4).....	118

# สารบัญรูป

รูปที่	หน้า
1.1 ตัวอย่างรูปแบบข้อมูลภาพตัวอักษรที่ใช้ในการทดลอง .....	5
1.2 ตัวอย่างแสดงผลการรู้จำเมื่อผ่านโปรแกรมการรู้จำ AmThai 2.5 Lite .....	5
2.1 ภาพแสดงแนวคิดในการรู้จำตัวอักษรภาษาไทยแบบออฟไลน์.....	9
2.2 ภาพแสดงแนวคิดในการรู้จำตัวอักษรภาษาไทยแบบออนไลน์.....	10
2.3 การแบ่งระดับของตัวอักษรพิมพ์ภาษาไทย .....	11
2.4 การแบ่งกลุ่มระดับโดยการหาจุดศูนย์กลางของกรอบภาพตัวอักษรที่มีส่วนของการเหลื่อมล้ำเกินสองระดับ.....	11
2.5 แผนภาพการทำงานของระบบในการรู้จำตัวอักษร .....	12
2.6 อัลกอริทึมสำหรับการจัดรวมกลุ่มตัวอักษรลายมือเขียนภาษาไทย .....	15
2.7 การแบ่งกรอบตัวอักษรออกเป็น 9 ส่วน .....	19
2.8 The Membership Function of DEPTH .....	19
2.9 โครงสร้างของระบบรู้จำตัวอักษร โดยใช้ Fuzzy-Rough Sets (a) ภาพโดยรวมของระบบ (b) รายละเอียดของการจัดแบ่งกลุ่มลำดับชั้นที่ 1 (c) รายละเอียดของการจัดแบ่งกลุ่มลำดับชั้นที่ 2 .....	22
2.10 กลุ่มทั้ง 5 กลุ่มที่มีสมาชิกประกอบไปด้วยตัวอักษรมากกว่า 1 ตัวอักษรที่แตกต่างกัน.....	22
2.11 ระบบการรู้จำตัวอักษรลายมือเขียนแบบออนไลน์โดยใช้ HMM และ Fuzzy Logic .....	27
2.12 ระบบการรู้จำโดยอาศัยวิธี Support Vector Machines และ Hidden Markov Models .....	29
2.13 รูปแบบตัวอักษรเกาหลีทั้ง 6 รูปแบบ .....	32
2.14 กระบวนการรู้จำโดยอาศัยโครงข่ายประสาทเทียมสำหรับระบบการรู้จำตัวอักษรด้วยกล้องถ่ายภาพที่นำไปประยุกต์ใช้งานบน Mobile Device .....	33
2.15 สถาปัตยกรรม TDNN .....	34
2.16 ขั้นตอนก่อนการรู้จำสำหรับการรู้จำลายมือเขียนแบบออนไลน์ที่เป็นภาษาลาติน .....	35
2.17 วิธีการของ Static และ Dynamic Information Coupling .....	35
2.18 สถาปัตยกรรมของ ART .....	36
2.19 สถาปัตยกรรมของ ARTMAP, Fuzzy ARTMAP .....	38
2.20 สถาปัตยกรรมของ AHN.....	42
2.21 สถาปัตยกรรมของ Simplified Fuzzy ARTMAP (SFAM) .....	43
2.22 (ก) รูปคลื่น x, y แบบต่อเนื่องทางเวลา (ข) ค่าครอสคอร์รีเลชันที่ได้ประมาณ 0.9 แสดงว่ารูปคลื่น x, y มีความสัมพันธ์คล้ายคลึงกันมากที่ delay = 40 จากทั้งหมด 4000 delay .....	45

## สารบัญรูป (ต่อ)

รูปที่	หน้า
2.23 (x) ชุดข้อมูล x และ (y) ชุดข้อมูล y (r) ค่าสหสัมพันธ์รีเลชัน (เมื่อทำการ normalized) ที่ได้ ประมาณ 1 (r=1) แสดงว่าชุดข้อมูล x, y มีความสัมพันธ์เหมือนกันที่ delay = -2 .....	46
3.1 การแยกบรรทัดและตัวอักษร โดยใช้ฮิสโตแกรม และการหาขอบภาพ.....	47
3.2 การแบ่งกลุ่มระดับโดยการหาจุดศูนย์กลางของกรอบภาพตัวอักษร .....	48
3.3 กรอบภาพตัวอักษร “อ” .....	48
3.4 การหาจุดกึ่งกลางของวงกลมที่มีจุดปลายเส้นผ่านศูนย์กลางทั้งสองจุดอยู่บนจุดปลายของขอบ ตัวอักษรสำหรับการทำ Skeleton .....	49
3.5 รูปแบบของภาพลายเส้นตัวอักษรหลังจากการทำ Skeleton .....	49
3.6 ภาพ 8-neighborhood ของพิกเซล P1 .....	50
3.7 แสดงรูปภาพที่จะทำให้เงื่อนไขข้อที่ 1 เป็นเท็จ .....	50
3.8 แสดงรูปภาพที่จะทำให้เงื่อนไขข้อที่ 2 เป็นเท็จ .....	50
3.9 แสดงรูปภาพที่จะทำให้เงื่อนไขข้อที่ 3 เป็นเท็จ .....	51
3.10 แสดงรูปภาพที่จะทำให้เงื่อนไขข้อที่ 4 เป็นเท็จ .....	51
3.11 การแบ่งกรอบภาพตัวอักษรออกเป็น 4 Quadrants .....	51
3.12 เมตริกขนาด 3x3 Pixels ที่ใช้ในการหาจุด Endpoint มีทั้งหมด 8 รูปแบบ .....	52
3.13 กรอบภาพตัวอักษรที่มีจุด Endpoint 2 จุด .....	52
3.14 กรอบภาพตัวอักษรที่มีจุด Endpoint 3 จุด .....	52
3.15 กรอบภาพตัวอักษรที่มีจุด Endpoint 4 จุด .....	53
3.16 แสดงจุดเริ่มต้นและจุดสิ้นสุดของตัวอักษร “อ” .....	54
3.17 แสดงจุดเริ่มต้นและจุดสิ้นสุดของตัวอักษร “ท” .....	54
3.18 แสดงจุดเริ่มต้นและจุดสิ้นสุดของตัวอักษร “ส” .....	54
3.19 รหัสแทนเส้นตัวอักษรทั้งหมด 9 รหัส .....	55
3.20 แสดงทิศทางของเส้นตัวอักษร “อ” ตามการจัดเรียงกรอบย่อยของภาพตัวอักษรตามลำดับของ ทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร .....	57
3.21 แสดงทิศทางของเส้นตัวอักษร “ท” ตามการจัดเรียงกรอบย่อยของภาพตัวอักษรตามลำดับของ ทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร .....	57
3.22 แสดงทิศทางของเส้นตัวอักษร “ส” ตามการจัดเรียงกรอบย่อยของภาพตัวอักษรตามลำดับของ ทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร .....	58

# สารบัญรูป (ต่อ)

รูปที่	หน้า
3.23 โครงสร้างของระบบการรู้จำตัวอักษรที่นำเสนอในงานวิจัยนี้โดยอาศัย Hierarchical Normalized Cross-Correlation (HNCC) Neural Network.....	60
3.24 สถาปัตยกรรมของครอสคอร์รีเลชันนิวรอลเน็ตเวิร์ค.....	61
3.25 รูปแบบข้อมูล Training Set (a) ตัวอักษรตัวที่ 1 คือ “ก” (b) ตัวอักษรตัวที่ 2 คือ “ภ” (c) ตัวอักษรตัวที่ 3 คือ “ถ” (d) ตัวอักษรตัวที่ 4 คือ “ก” (e) ตัวอักษรตัวที่ 5 คือ “ภ” (f) ตัวอักษรตัวที่ 6 คือ “ถ” .....	68
3.26 ข้อมูลภาพตัวอักษรใน Training Set ที่มีการ Resize [29, 25] และทำการ Skeleton แล้ว (a) ตัวอักษรตัวที่ 1 คือ “ก” (b) ตัวอักษรตัวที่ 2 คือ “ภ” (c) ตัวอักษรตัวที่ 3 คือ “ถ” (d) ตัวอักษรตัวที่ 4 คือ “ก” (e) ตัวอักษรตัวที่ 5 คือ “ภ” (f) ตัวอักษรตัวที่ 6 คือ “ถ” .....	69
3.27 แสดงทิศทางของเส้นตัวอักษรในการ Training ตัวที่ 1 ตามการจัดเรียงกรอบย่อยของภาพ ตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร .....	74
3.28 แสดงทิศทางของเส้นตัวอักษรในการ Training ตัวที่ 2 ตามการจัดเรียงกรอบย่อยของภาพ ตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร .....	75
3.29 แสดงทิศทางของเส้นตัวอักษรในการ Training ตัวที่ 3 ตามการจัดเรียงกรอบย่อยของภาพ ตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร .....	79
3.30 แสดงทิศทางของเส้นตัวอักษรในการ Training ตัวที่ 4 ตามการจัดเรียงกรอบย่อยของภาพ ตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร .....	84
3.31 แสดงทิศทางของเส้นตัวอักษรในการ Training ตัวที่ 5 ตามการจัดเรียงกรอบย่อยของภาพ ตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร .....	88
3.32 แสดงทิศทางของเส้นตัวอักษรในการ Training ตัวที่ 6 ตามการจัดเรียงกรอบย่อยของภาพ ตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร .....	93
3.33 รูปแบบข้อมูล Testing Set (a) ตัวอักษรตัวที่ 1 คือ “ก” (b) ตัวอักษรตัวที่ 2 คือ “ภ” (c) ตัวอักษร ตัวที่ 3 คือ “ถ” .....	97
3.34 ข้อมูลภาพตัวอักษรใน Testing Set ที่มีการ Resize [29, 25] และทำการ Skeleton แล้ว (a) ตัวอักษรตัวที่ 1 คือ “ก” (b) ตัวอักษรตัวที่ 2 คือ “ภ” (c) ตัวอักษรตัวที่ 3 คือ “ถ” .....	98
3.35 แสดงทิศทางของเส้นตัวอักษรในการ Testing ตัวที่ 1 ตามการจัดเรียงกรอบย่อยของภาพ ตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร .....	101

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.36 แสดงทิศทางของเส้นตัวอักษรในการ Testing ตัวที่ 2 ตามการจัดเรียงกรอบย่อยของภาพ ตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร .....	102
3.37 แสดงทิศทางของเส้นตัวอักษรในการ Testing ตัวที่ 3 ตามการจัดเรียงกรอบย่อยของภาพ ตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร .....	103
4.1 ตัวอย่างข้อมูลภาพป้ายทะเบียนรถไทยที่ใช้ในการทดลอง.....	110
5.1 แสดงวิธีของการ Thinning ตัวอักษรด้วยวิธีต่าง ๆ (a) กรอบตัวอักษร (b) Hilditch's Algorithm (c) Skeletonization (d) Thinning ลายเส้นตัวอักษร .....	120

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

จากการศึกษาขั้นตอนกระบวนการรู้จำตัวอักษรที่ผ่านมาในงานวิจัยที่เกี่ยวข้องนั้น ได้มีการศึกษาวิจัยกันมาอย่างต่อเนื่องและกว้างขวางเพื่อเพิ่มประสิทธิภาพความสามารถในการรู้จำตัวอักษรให้มีความถูกต้องมากขึ้นอาทิเช่น การรู้จำตัวอักษร โดยอาศัยโครงข่ายประสาทเทียมรูปแบบต่าง ๆ ได้แก่ Back-Propagation, BAM และ CPN หรือจะเป็นการรู้จำโดยอาศัยการใช้ลักษณะเด่นของตัวอักษรซึ่งอาจจะมีการนำเอาทฤษฎีพีชคณิตหรือทฤษฎีอื่น ๆ เข้ามาช่วยในกระบวนการรู้จำ ทั้งนี้ความสามารถในการรู้จำนั้นไม่ได้ขึ้นกับวิธีที่นำมาใช้ในขั้นตอนของกระบวนการรู้จำตัวอักษรเท่านั้น แต่ยังคงอาศัยกระบวนการก่อนการรู้จำที่ดีเพื่อให้ได้ภาพตัวอักษรที่มีคุณภาพดีด้วย ด้วยเหตุนี้จึงทำให้มีข้อดีและข้อเสียแตกต่างกันไป โดยเฉพาะถ้าภาพตัวอักษรก่อนการรู้จำมีคุณภาพที่ไม่ดีก็จะส่งผลให้ประสิทธิภาพในการรู้จำภาพตัวอักษรนั้นไม่มีความถูกต้องเท่าที่ควร[1] ดังนั้นในการรู้จำตัวอักษรก็จะขึ้นอยู่กับกระบวนการก่อนการรู้จำที่จะจัดการข้อมูลภาพตัวอักษรให้อยู่ในรูปแบบที่เหมาะสมก่อนจะเข้าสู่กระบวนการรู้จำ เช่นการปรับให้คุณภาพของข้อมูลภาพตัวอักษรก่อนการรู้จำมีคุณภาพที่ดีโดยจะต้องไม่มีสัญญาณรบกวน และรวมถึงปัญหาต่าง ๆ ที่พบเจอในงานวิจัยที่ผ่านมาเช่น การแก้ไขตัวอักษรที่ติดกัน [2] [3] [4] การลดสัญญาณรบกวน [4] การทำขอบตัวอักษรให้เรียบ [5] การเชื่อมต่อลายเส้นที่ขาดหายไปของภาพอักษรตัวพิมพ์ภาษาไทยในแนวนอน [6] และการเชื่อมต่อลายเส้นที่ขาดหายไปของภาพอักษรตัวพิมพ์ภาษาไทยในแนวตั้ง [7] เหล่านี้ช่วยให้กระบวนการก่อนการรู้จำได้ข้อมูลภาพตัวอักษรที่ดีขึ้น จากนั้นจะพิจารณากระบวนการรู้จำที่มีการออกแบบอัลกอริทึมสำหรับการรู้จำภาษาต่าง ๆ โดยแต่ละรูปแบบภาพตัวอักษรของแต่ละภาษาก็จะมีคุณลักษณะ โครงสร้างที่แตกต่างกันไป สำหรับงานวิจัยที่ผ่านมาจะเป็นการศึกษาในการรู้จำตัวอักษรของแต่ละภาษาโดยเฉพาะ ตัวอย่างเช่นการรู้จำตัวอักษรภาษาอังกฤษ การรู้จำตัวอักษรภาษาจีน การรู้จำตัวอักษรภาษาฝรั่งเศส หรือแม้กระทั่งการรู้จำตัวอักษรภาษาไทย

สำหรับรูปแบบโครงสร้างของตัวอักษรภาษาไทยนั้นปกติแล้วจะประกอบไปด้วยเส้นตัวอักษรที่เป็นเส้นตรง เส้นโค้ง เส้นซิกแซก หรือรูป โดยส่วนมากแล้วตัวอักษรภาษาไทยจะมีจุดเริ่มต้นเป็นรูป หรือที่เราเรียกว่าส่วนที่เป็นหัวตัวอักษร แต่ในปัจจุบันมีการพัฒนารูปแบบตัวอักษรไทยที่เปลี่ยนไปจากเดิมคือที่ส่วนหัวของตัวอักษรไม่ได้เป็นรูปเรขาคณิตแบบตัวอักษรที่ไม่มีหัว ตามตารางที่ 1.1 แสดงการเปรียบเทียบลักษณะของตัวอักษรภาษาไทยในรูปแบบที่มีหัว

และไม่มีหัว ดังนั้นเมื่อมีการพิจารณาลักษณะเด่นของ โครงสร้างตัวอักษรภาษาไทย สิ่งแรกที่จะใช้พิจารณาคือลักษณะของตัวอักษรภาษาไทยในรูปแบบที่มีหัว ซึ่งงานวิจัยที่เกี่ยวกับการรู้จำตัวอักษรภาษาไทยส่วนมากนั้น ไม่ได้เน้นถึงลักษณะของตัวอักษรภาษาไทยในรูปแบบที่ไม่มีหัว ซึ่งสามารถพบเจอได้ตามเอกสารต่าง ๆ เช่นหนังสือพิมพ์ วารสาร สื่อแผ่นพับโฆษณาต่าง ๆ ที่เป็นตัวอักษรภาษาไทยที่มีรูปแบบการเขียนหรือพิมพ์ออกมาแล้วเป็นรูปแบบที่ไม่มีหัวเป็นส่วนใหญ่

ตัวอย่างข้อมูลภาพตัวอักษรตามรูปที่ 1.1 เมื่อผ่านการทดสอบด้วยโปรแกรมอ่านไทย (AmThai 2.5 Lite) พบว่าผลที่ได้ออกมามีความถูกต้องเท่ากับ 34.85% แสดงผลลัพธ์ที่ได้จากการรู้จำตามรูปที่ 1.2 ซึ่งแสดงให้เห็นว่าผลของการรู้จำที่ได้ออกมายังคงมีปัญหาในเรื่องของตัวอักษรพิมพ์ภาษาไทยในรูปแบบที่ไม่มีหัว

ในงานวิจัยครั้งนี้จึงมุ่งเน้นที่กลุ่มข้อมูลตัวอย่างลักษณะของตัวอักษรภาษาไทยในรูปแบบที่ไม่มีหัวซึ่งจะมีทั้งหมด 39 รูปแบบ แต่ละรูปแบบจะประกอบด้วยพยัญชนะ สระ วรรณยุกต์ ตัวเลข อารบิก และสัญลักษณ์อื่น ๆ รวมทั้งหมดรูปแบบละ 80 ตัวอักษร นอกจากนี่ยังพิจารณากลุ่มข้อมูลภาพตัวอักษรที่อยู่บนป้ายทะเบียนรถที่เป็นภาษาไทยและตัวเลขอารบิก

แต่ทั้งนี้จะเห็นว่าตั้งแต่อดีตมาจนถึงปัจจุบัน ได้มีนักวิจัยทำการพัฒนาเทคโนโลยีทางการรู้จำตัวอักษรกันมากขึ้น แต่ในการพัฒนาจะมุ่งเน้นกลุ่มทดลองตามภาษาที่ใช้ของแต่ละประเทศด้วยเหตุนี้การรู้จำของแต่ละภาษาก็มีวิธีในการรู้จำที่ต่างกันออกไปตามโครงสร้างของแต่ละภาษา เนื่องจากตัวอักษรของแต่ละภาษาที่มีรูปแบบต่างกันออกไป และสำหรับงานวิจัยที่เกี่ยวกับการรู้จำภาษาไทยนั้น จะรู้จำในกลุ่มของตัวอักษรมีหัว แต่ส่วนมากในการทำสื่อโฆษณา เผยแพร่ หรือหนังสือต่าง ๆ จะประกอบไปด้วยตัวอักษรภาษาไทยไม่มีหัว ซึ่งปัจจุบันนี้พบอยู่อย่างแพร่หลาย

ดังนั้นงานวิจัยครั้งนี้ได้เล็งเห็นถึงความสำคัญของการรู้จำตัวอักษรตัวพิมพ์ภาษาไทยไม่มีหัว เพื่อเพิ่มประสิทธิภาพในการรู้จำตัวอักษรภาษาไทยให้ครอบคลุมไปถึงตัวอักษรภาษาไทยไม่มีหัว ให้สามารถได้ผลลัพธ์เอกสารสื่อต่าง ๆ ออกมาในรูปแบบที่ยอมรับได้ และมีประสิทธิภาพยิ่งขึ้น

## 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1 เพื่อศึกษาลักษณะพิเศษของตัวพิมพ์อักษรไทยในกลุ่มตัวอักษรไม่มีหัว ที่จะนำมาใช้ในวิธีการรู้จำ โดยศึกษาเปรียบเทียบจากวิธีต่าง ๆ ที่ผ่านการวิจัยมาแล้ว

1.2.2 เพื่อศึกษาวิธีการที่เหมาะสมของการวิเคราะห์หาความสัมพันธ์ของโครงสร้างทางกายภาพมาใช้ในการรู้จำตัวพิมพ์อักษรไทยในกลุ่มตัวอักษรไม่มีหัว

1.2.3 เพื่อศึกษาวิธีการที่เหมาะสมของการนำโครงข่ายประสาทเทียมมาใช้ในการรู้จำตัวพิมพ์อักษรไทย

1.2.4 เพื่อเพิ่มประสิทธิภาพในระบบการรู้จำของตัวอักษรภาษาไทยครอบคลุมในรูปแบบตัวอักษรไม่มีหัว

ตารางที่ 1.1 การเปรียบเทียบลักษณะของตัวอักษรภาษาไทยในรูปแบบที่มีหัวและไม่มีหัว

ตัวอักษรที่	ตัวอักษรภาษาไทยรูปแบบที่มีหัว	ตัวอักษรภาษาไทยรูปแบบที่ไม่มีหัว
1. กอไก่	ก	ก
2. ขอไข่	ข	ข
3. ขอขวด	ข	ข
4. คอควาย	ค	ค
5. คอคน	ค	ค
6. หมอระฆัง	ฃ	ข
7. งองู	ง	ง
8. จอจาน	จ	จ
9. ฉอฉิ่ง	ฉ	ฉ
10. ซอช้าง	ซ	ซ
11. ซอโซ่	ซ	ซ
12. ฌอกะเมือ	ฌ	ฌ
13. ฌอผู้หญิง	ญ	ญ
14. ฎอปลั๊ก	ฎ	ฎ
15. ฎอชฎา	ฎ	ฎ
16. ฐอฐาน	ฐ	ฐ
17. ทอมนโฑ	ท	ท
18. ฒอผู้เฒ่า	ฒ	ฒ
19. ฒอเฒ่า	ฒ	ฒ
20. ดอเด็ก	ด	ด
21. ดอเต่า	ด	ด

## ตารางที่ 1.1 (ต่อ)

ตัวอักษรที่	ตัวอักษรภาษาไทยรูปแบบที่มีหัว	ตัวอักษรภาษาไทยรูปแบบที่ไม่มีหัว
22. ถอถุง	ถ	ท
23. ทอทหาร	ท	ท
24. ธอธง	ธ	ธ
25. นอหนุ	น	น
26. บอไบไม้	บ	บ
27. ปอปลา	ป	ป
28. ผอผี	ผ	ผ
29. ผอฝ	ฝ	ฝ
30. พอพาน	พ	พ
31. ฟอฟิน	ฟ	ฟ
32. ภอสำเภา	ภ	ภ
33. มอม่่า	ม	ม
34. ยอขัณฑ์	ย	ย
35. รออเรือ	ร	ร
36. ลอลิง	ล	ล
37. วอแหวน	ว	ว
38. ศอศาลา	ศ	ศ
39. ฆอบอถุณี	ฆ	ฆ
40. สอเสื่อ	ส	ส
41. หอหีบ	ห	ท
42. พอจุฬา	พ	ฬ
43. อออ่าง	อ	อ
44. ฮอนกฐก	ฮ	ฮ



### 1.3 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

แนวความคิดที่ใช้ในงานวิจัยนี้ประกอบไปด้วย 2 กระบวนการคือกระบวนการก่อนการรู้จำและกระบวนการรู้จำ ในขั้นตอนของกระบวนการก่อนการรู้จำนั้นเมื่อรับเอาภาพตัวอักษรในเอกสารเข้ามาแล้วจะทำการแบ่งแต่ละบรรทัดออกจากกันด้วยวิธีของการทำ Histogram จากนั้นในแต่ละบรรทัดจะแบ่งออกเป็น 3 ระดับซึ่งจะทำการจัดเข้ากลุ่มแต่ละกลุ่มตามระดับของภาพตัวอักษรนั้น ๆ ได้แก่ กลุ่มตัวอักษรระดับบน กลางและล่างตามลำดับ แล้วใช้วิธี Histogram อีกเช่นกันในการแบ่งภาพตัวอักษรให้ได้ออกมาเป็นกรอบภาพตัวอักษรแต่ละตัว จากนั้นจะผ่านกระบวนการ Thinning กรอบภาพตัวอักษร ขั้นตอนต่อไปคือการแบ่งภาพตัวอักษรที่ Skeleton แล้วออกเป็น 9 ส่วนย่อยที่เท่า ๆ กัน แล้วทำการพิจารณาหาจุดปลายของภาพตัวอักษรและมีการเก็บข้อมูลโดยเริ่มจากจุดปลายที่ 1 ไปจนถึงจุดปลายที่ 2 และหากกรอบส่วนย่อยที่ถูกแบ่งนั้นมีจุดปลายที่สามก็จะอยู่ในกลุ่มของตัวอักษรที่มีจุดปลายมากกว่า 2 จุด ซึ่งค่าที่เก็บนั้นจะมีการแทนรหัสแทนทิศทาง 9 รหัสตามทิศทางที่ทำมุมของเส้นตัวอักษร โดยที่แต่ละกรอบที่ถูกแบ่งย่อย 9 กรอบจะเก็บค่าที่อยู่ในช่วงกรอบของตนเอง และทำการเรียงเก็บข้อมูลต่อไป กล่าวคือรูปแบบ Input Vector จะเก็บเฉพาะส่วนข้อมูลที่เป็นลายเส้นภาพตัวอักษร จากนั้นแต่ละกรอบที่ถูกแบ่งออกจะส่งต่อเข้าสู่กระบวนการรู้จำโดยอาศัยโครงข่ายประสาทเทียมแบบใหม่ที่พัฒนาขึ้นมาโดยเน้นที่บริเวณส่วนไม่มีหัวของตัวอักษรภาษาไทยมาเป็นหลักการเบื้องต้นของแนวคิดในการพัฒนากระบวนการเรียนรู้ทั้งหมด และพยายามที่จะสร้างกระบวนการรู้จำใหม่ที่มีการปรับเปลี่ยนเฉพาะบางส่วนย่อยเพื่อให้เกิดประสิทธิภาพในการเรียนรู้บริเวณตัวอักษรไม่มีหัวที่เป็นจุดเริ่มต้นของการเขียนตัวอักษรไม่มีหัว นอกจากนี้งานวิจัยนี้ได้นำเอาวิธีการวัดความเหมือนหรือความคล้ายคลึงกันที่อาศัยฟังก์ชันครอสคอรีเลชัน (Cross-correlation) ที่มีประสิทธิภาพและสามารถนำมาเป็นตัววัดค่าความเหมือนของรูปแบบ ส่วนแนวทางในการรู้จำ Fuzzy Adaptive Resonance Theory (Fuzzy ARTMAP) หรือที่เรียกว่าฟัชชีอาร์ทแมพนั้น มีความสามารถในการเรียนรู้สิ่งใหม่ได้พร้อมกับยังคงความรู้เก่าไว้ได้ ดังนั้นงานวิจัยนี้จึงใช้แนวทางฟัชชีอาร์ทแมพเป็นพื้นฐานของกระบวนการเรียนรู้และพยายามหาวิธีการปรับเอาครอสคอรีเลชันมาใช้ร่วมกับแนวทางฟัชชีอาร์ทแมพให้ได้

### 1.4 ขอบเขตของการวิจัย

งานวิจัยนี้เน้นการพัฒนากระบวนการรู้จำสำหรับรูปแบบภาพตัวอักษรในเอกสารที่มีขอบเขตต่อไปนี้



## 1.6 ประโยชน์ที่คาดว่าจะได้รับ

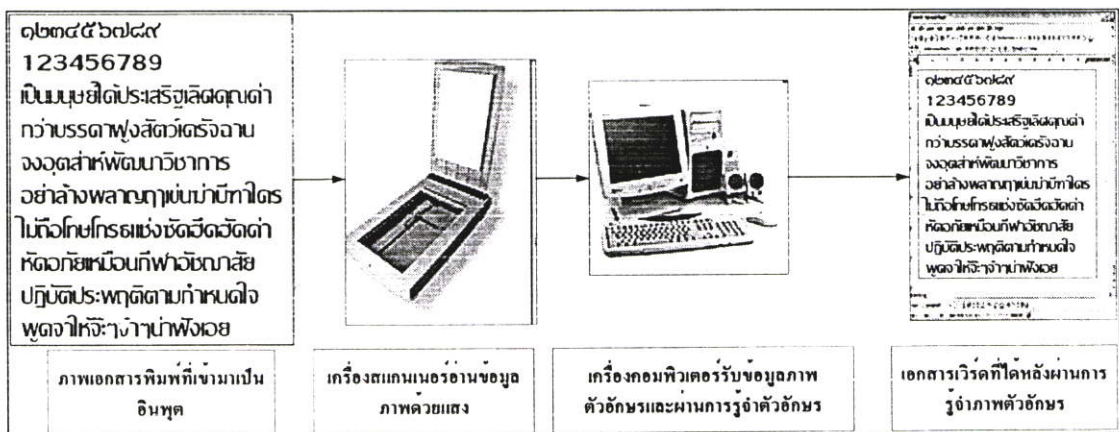
- 1.6.1 ทราบถึงบทความงานวิจัยต่าง ๆ ที่เกี่ยวข้องกับการวิจัยครั้งนี้
- 1.6.2 ทราบถึงลักษณะเด่นของตัวพิมพ์อักษรภาษาไทยแบบฟอนต์ไม่มีส่วนหัวของตัวอักษร
- 1.6.3 เป็นพื้นฐานในการรู้จักตัวอักษรภาษาไทยที่สมบูรณ์แบบ เพื่อใช้ในการพัฒนาต่อไปในอนาคตให้เหมาะกับตัวอักษรภาษาไทยทุกรูปแบบ

## บทที่ 2

# หลักการและทฤษฎีเบื้องต้นที่นำไปสู่แนวคิดของการวิจัย

### 2.1 แนวคิดในการรู้จำตัวอักษรภาษาไทย

ระบบการรู้จำตัวอักษรประกอบไปด้วย Optical Scanner สำหรับอ่านภาพตัวอักษร และซอฟต์แวร์สำหรับวิเคราะห์ภาพตัวอักษร ซึ่งระบบการรู้จำตัวอักษรโดยส่วนมากจะมีการรวมเอาฮาร์ดแวร์และซอฟต์แวร์รวมเข้าด้วยกันเพื่อใช้ในการรู้จำตัวอักษร ดังนั้นหลักการทำงานของการรู้จำตัวอักษรประกอบด้วย 3 กระบวนการหลัก ๆ คือ กระบวนการก่อนการรู้จำ กระบวนการรู้จำ และกระบวนการหลังการรู้จำ ซึ่งมีความสำคัญมากทั้ง 3 กระบวนการ ดังนั้นระบบจะทำงานได้ดีหากมีข้อมูลเข้ามาอย่างถูกต้อง และผ่านการรู้จำที่มีประสิทธิภาพในการเรียนรู้รูปแบบ Pattern และจัดแบ่งประเภทของกลุ่มวัตถุเหล่านั้นได้เหมาะสม พร้อมทั้งสามารถที่จะจัดเรียงเพื่อให้ได้ไฟล์ที่ประกอบไปด้วยตัวอักษรในรูปแบบ ASCII Code ที่สามารถเปลี่ยนแปลงแก้ไขได้ด้วยโปรแกรมจัดการข้อความต่าง ๆ ได้



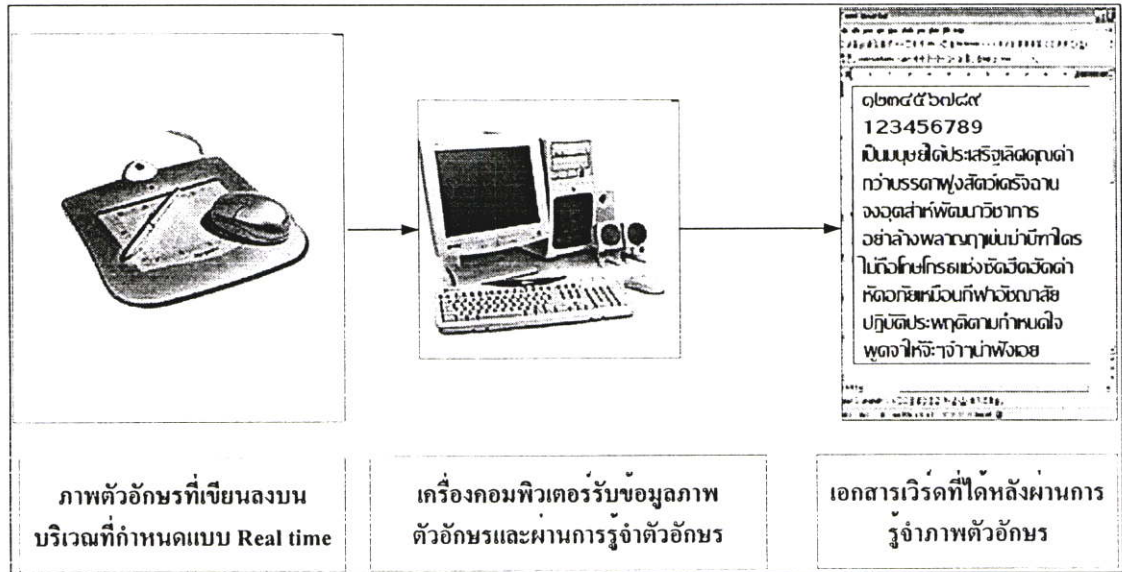
รูปที่ 2.1 ภาพแสดงแนวคิดในการรู้จำตัวอักษรภาษาไทยแบบออฟไลน์

การรู้จำตัวอักษรภาษาไทยแบ่งออกเป็น 2 ประเภทหลักคือ

1. Offline Thai Optical Character Recognition เป็นการรู้จำตัวอักษรภาษาไทยแบบออฟไลน์ของภาพตัวอักษรบนกระดาษที่ออกมาจากเครื่องพิมพ์ชนิดต่าง ๆ เช่นเครื่องพิมพ์อิงค์เจ็ท (Inkjet printer) เครื่องพิมพ์ดอตเมตริกซ์ (Dot metric printer) เครื่องพิมพ์เลเซอร์ (Laser printer) หรือเครื่องพิมพ์ดีด (typewriter) แล้วนำมาผ่านเครื่องสแกนเนอร์ จากนั้นจะได้ภาพเข้าสู่เครื่องคอมพิวเตอร์ออกมาเป็นภาพเอกสารไบนารี (Binary document images) แล้วผ่านเข้าสู่กระบวนการ

ก่อนการรู้จำ โดยจะทำการจัดปรับภาพเอกสารก่อนการรู้จำให้มีคุณภาพที่ดีพร้อมส่งเข้าสู่กระบวนการรู้จำต่อไปเช่น กำจัดสัญญาณรบกวน ปรับภาพเอกสารที่เอียง เป็นต้น ดังรูปที่ 2.1

2. Online Thai Optical Character Recognition เป็นการรู้จำตัวอักษรภาษาไทยแบบออนไลน์ของตัวอักษรในขณะที่ทำการเขียนตัวอักษรนั้น ๆ ลงบนพื้นที่ที่กำหนดไว้แล้วจะทำการอ่านภาพตัวอักษรที่เขียนเข้าสู่กระบวนการรู้จำเป็นต้น ดังรูปที่ 2.2



รูปที่ 2.2 ภาพแสดงแนวคิดในการรู้จำตัวอักษรภาษาไทยแบบออนไลน์

กระบวนการทำงานของการรู้จำตัวอักษรเริ่มต้นจากการวิเคราะห์ภาพและทำการแบ่งภาพออกเป็นแต่ละระดับ (Zone) พิจารณาเฉพาะบริเวณที่เป็นภาพตัวอักษรภาษาไทยประกอบไปด้วย ภาพพยัญชนะ ภาพสระ ภาพวรรณยุกต์ ดังรูปที่ 2.3 จากนั้นจะแบ่งแต่ละระดับออกเป็นกรอบภาพตัวอักษรหรือเรียกว่า Character Block กระบวนการเหล่านี้เรียกว่ากระบวนการก่อนการรู้จำ และทำการส่งข้อมูลแต่ละกรอบภาพตัวอักษรเป็น Pattern เข้าสู่กระบวนการรู้จำโดยอาศัยการแบ่งกลุ่มของวัตถุออกมาเป็นกลุ่มประเภทต่าง ๆ แบ่งออกเป็น 2 วิธีหลัก ๆ คือ

**แบบที่ 1 Pattern Classification** เป็นการจัดแบ่งกลุ่มโดยที่ทราบว่าจะมีข้อมูลแต่ละ Pattern ที่เข้มานั้นสามารถจัดแบ่งกลุ่มได้ตรงตามประเภทของ Pattern ที่กำหนดไว้

**แบบที่ 2 Pattern Clustering** เป็นการจัดแบ่งกลุ่มโดยไม่ทราบประเภทของ Pattern นั้น

สำหรับในขั้นตอนวิธีการของกระบวนการรู้จำของการแบ่งกลุ่มประเภทของ pattern ต่าง ๆ มีการนำเอาโครงข่ายประสาทเทียม (Neural Network) เข้มาใช้ในกระบวนการนี้เรียกว่า กระบวนการรู้จำตัวอักษร ซึ่งเมื่อผ่านขั้นตอนของการรู้จำแล้ว ก็จะมีการจัดเรียงแต่ละภาพตัวอักษร

ที่รู้จำออกมาเป็นข้อมูล ASCII Code เพื่อทำการเรียงตัวอักษรที่ได้ส่วนนี้เรียกว่ากระบวนการหลังการรู้จำ

## ระดับที่ 1 ระดับที่ 2 ระดับที่ 3

# การรู้จำรูปภาพตัวอักษรพิมพ์ภาษาไทยแบบไม่มีหัว

### รูปที่ 2.3 การแบ่งระดับของตัวอักษรพิมพ์ภาษาไทย

สำหรับการหาค่า Histogram ในแนวนอน เพื่อแบ่งภาพตัวอักษรในแนวนอน (แยกบรรทัด)

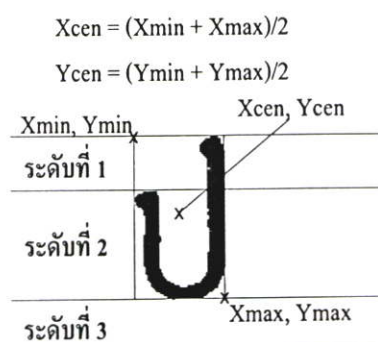
$$\text{HorizontalHistogram}(y) = \sum_x P(x, y) \quad (2.1)$$

สำหรับการหาค่า Histogram ในแนวตั้ง เพื่อแบ่งภาพตัวอักษรในแนวตั้ง (แยกตัวอักษรที่อยู่ในบรรทัดเดียวกัน)

$$\text{VerticalHistogram}(x) = \sum_y P(x, y) \quad (2.2)$$

เมื่อทำการแบ่งระดับของภาพตัวอักษรตามรูปที่ 2.3 แล้ว จะมีการแยกออกเป็นตัวอักษรเดี่ยวหรือเรียกว่ากรอบภาพตัวอักษร (Character Block) ตามตัวอย่างรูปที่ 2.4 โดยมีรายละเอียดดังนี้

- กลุ่มระดับบน คือกลุ่มตัวอักษรที่มีจุดศูนย์กลาง (Xcen, Ycen) อยู่ในบริเวณระดับที่ 1
- กลุ่มระดับกลาง คือกลุ่มตัวอักษรที่มีจุดศูนย์กลาง (Xcen, Ycen) อยู่ในบริเวณระดับที่ 2
- กลุ่มระดับล่าง คือกลุ่มตัวอักษรที่มีจุดศูนย์กลาง (Xcen, Ycen) อยู่ในบริเวณระดับที่ 3

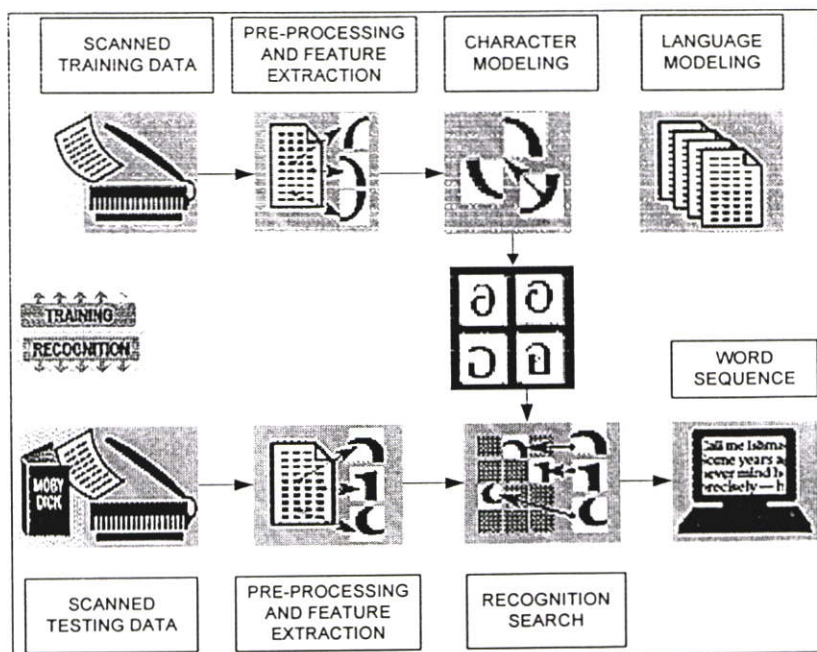


รูปที่ 2.4 การแบ่งกลุ่มระดับโดยการหาจุดศูนย์กลางของกรอบภาพตัวอักษรที่มีส่วนของการเหลื่อมล้ำเกินสองระดับ

ขั้นตอนในการรู้จำตัวอักษรภาษาไทยสามารถที่จะสรุปได้ดังรูปที่ 2.5 มีการประยุกต์ใช้งานโครงข่ายประสาทเทียม (Neural Network) สำหรับงานที่เป็น Characters Classification ซึ่งสามารถแบ่งการทำงานออกเป็น 2 ระยะคือ

1. ระยะของการเรียนรู้ (Training) เริ่มจากการสแกนข้อมูลภาพตัวอักษรจากเอกสารเข้าสู่เครื่องคอมพิวเตอร์ และผ่านเข้าสู่กระบวนการก่อนการรู้จำ (Preprocessing) ในขั้นตอนนี้อาจจะมีการดึงเอาคุณลักษณะพิเศษหรือลักษณะเด่นของวัตถุ (Pattern) ออกมาจากนั้นเข้าสู่กระบวนการรู้จำ (Character Modeling) โดยมีการจัดกลุ่ม Pattern ต่าง ๆ ที่มีความคล้ายคลึงกันมากพอให้อยู่ในกลุ่มประเภทเดียวกัน จากนั้นจะหาตัวแทนของตัวอักษรในกลุ่มนั้นแล้วแปลงเป็นรหัสแทนตัวอักษร ASCII Code เพื่อทำงานในกระบวนการหลังการรู้จำ (Language Modeling) และอาศัยกรรมมาเข้ามาช่วยในการจัดเรียงคำ

2. ระยะของการทดสอบการรู้จำ (Testing) เป็นการทดสอบในการนำไปใช้งานจริง ว่าการแบ่งกลุ่มข้อมูล Pattern ให้จัดอยู่ในประเภทเดียวกันนั้น ได้ผลออกมามีความถูกต้องมากน้อยเพียงใด โดยที่จะเอาชุดข้อมูลภาพตัวอักษรชุดใหม่ที่ยังไม่เคยใส่ในระบบในระยะของการเรียนรู้ มาเป็นข้อมูลในระยะของการทดสอบ แล้วเปรียบเทียบกับกลุ่มโดยดูจากตัวแทนของกลุ่มนั้น โดยจะได้ผลลัพธ์ออกมาเป็นกลุ่มที่ระบบการรู้จำแสดงออกมาว่ามีความคล้ายคลึงกับกลุ่มนั้นมากที่สุด



รูปที่ 2.5 แผนภาพการทำงานของระบบในการรู้จำตัวอักษร

## 2.2 งานวิจัยที่เกี่ยวข้อง

### 2.2.1 HMM Topology Selection for On-line Thai Handwritten Recognition [15]

งานวิจัยนี้นำเสนอเกี่ยวกับการรู้จำลายมือเขียนตัวอักษรภาษาไทยแบบออนไลน์โดยใช้วิธีการของ Chain Code ในการทำ Preprocessing ส่วนขั้นตอนในการรู้จำใช้วิธีการของ Hidden Markov Model (Left-Right-Left) ซึ่งในงานวิจัยที่ผ่านมาวิธีการของ HMM นี้ได้มีการนำไปประยุกต์ใช้ในการรู้จำเสียง แต่สำหรับงานวิจัยนี้ได้ศึกษาการนำเอาวิธี HMM มาใช้ในการรู้จำรูปแบบที่เป็น pattern ลายมือเขียน หากมีการเปรียบเทียบลักษณะอินพุตที่เป็นข้อมูลเสียงและลายมือเขียนนั้น จะพบว่ามีลักษณะที่แตกต่างกันมากคือเสียงนั้นเป็นสัญญาณอนาล็อกที่มีลักษณะอินพุตประกอบไปด้วย Large Dimensional Vector of Continuous Real Values แต่ลายมือเขียนนั้นประกอบไปด้วย Small Dimensional Vector of Discrete Values

สำหรับการทดลองในครั้งนี้ข้อมูลจะถูกเก็บในรูปแบบ (x, y, t) ที่เกิดจากการ Sampling ในแต่ละครั้งโดย Sampling Rate เท่ากัน แต่ทั้งนี้ความเร็วในการเขียนลายมือเขียนมีผลทำให้ความสามารถในการรู้จำออกมาต่างกัน หากว่า Sampling Rate ต่ำกว่าความเร็วในการเขียนก็จะทำให้ลักษณะของลายเส้นบางส่วนที่สำคัญหายไป วิธีในการแทนค่าอินพุตจะใช้ Chain Code ประกอบไปด้วยค่าตั้งแต่ 1 – 8 ปัญหาที่พบต่อมาก็คือเนื่องจากบางครั้งตัวอักษรที่เขียนมีค่า Chain Code ออกมาเหมือนกัน แต่ต่างกันที่ความยาวของเส้นอักษรยกตัวอย่าง “บ” กับ “ป” จึงได้คิดวิธีปรับให้มีการบวกค่า 8 เข้าไป สำหรับการแทนค่าในช่วงกรอบล่างซ้ายและกรอบบนขวาของกรอบภาพตัวอักษร ต่อมาเป็นกระบวนการรู้จำที่อาศัยวิธี HMM โดยในการทดลองประกอบไปด้วย 3 โครงสร้างคือ Fully Connected (FC), Left Right (LR), และ Left-Right-Left (LRL)

กลุ่มข้อมูลที่ใช้ในการทดลองเป็นตัวอักษรภาษาไทย 38 ตัวอักษร ผลลัพธ์ที่ได้จากการทดลองสำหรับโมเดลแบบ LR = 82.56%, โมเดลแบบ FC=85.69% และโมเดลแบบ LRL=95.67%

### 2.2.2 Discriminative Training for HMM-Based Offline Handwritten Character Recognition [16]

งานวิจัยนี้มีการใช้ Discriminative Training ได้แก่ Maximum Mutual Information (MMI) พร้อมทั้งการนำเอา Composite Images ที่ทำการหมุนภาพและปรับเปลี่ยนมุมของภาพตัวอักษรลายมือเขียน และศึกษาวิธีของ Principal Component Analysis (PCA) โดยพัฒนาวิธีใหม่ขึ้นมาและเรียกว่า Block-based PCA ซึ่งจะเป็นการประยุกต์ใช้ PCA จัดการกับบริเวณที่เลื่อมล้ำกันในแนวตั้งสำหรับแต่ละกรอบตัวอักษร ในการทดลองครั้งนี้จะนำมาพัฒนาระบบการรู้จำลายมือเขียนตัวอักษรไทยที่อาศัย HMM-based System

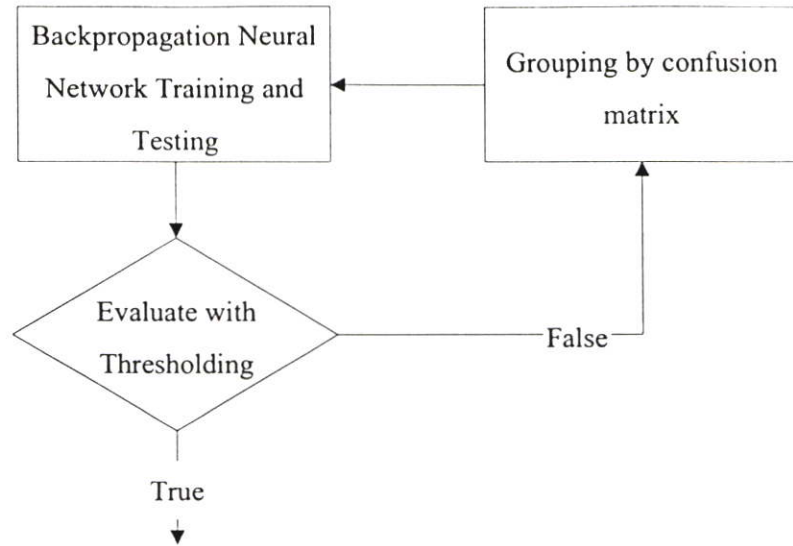
ข้อมูลที่ใช้ในการทดลองประกอบไปด้วย 77 ตัวอักษรที่เป็นพยัญชนะ สระ วรรณยุกต์ และตัวเลข โดยประกอบไปด้วยลายมือเขียนของ 20 คนที่มีวิธีการเขียนแต่ละตัวอักษรที่แตกต่างกันออกไป และ Training Set จะเลือกขึ้นมาตัวอักษรละ 120 รูปแบบจาก 20 คนที่เขียน จำนวนที่ใช้ Train และ Test ก็แบ่งเท่ากัน ซึ่งตัวอักษรจำนวน 64 ตัวอยู่ในระดับ Baseline และ 13 ตัวไม่อยู่ใน Baseline เช่น วรรณยุกต์ สระที่ปรากฏระดับบนและล่าง ผลลัพธ์จากการทดลองจะได้เปอร์เซ็นต์ของการรู้จำเท่ากับ 90.35%

### 2.2.3 The Clustering Technique for Thai Handwritten Recognition [17]

งานวิจัยนี้อธิบายถึงอัลกอริทึมสำหรับการจัดรวมกลุ่มตัวอักษรลายมือเขียนภาษาไทยที่มีลักษณะโครงสร้างคล้ายกันออกเป็นหมวดหมู่โดยเริ่มจากการทำ Vertical Stroke Detection โดยการพิจารณาพบว่าข้อมูลของตัวอักษรจะมีปรากฏในระดับแนวนอนมากกว่าแนวตั้ง จึงแบ่ง Character Area ออกเป็นขนาด 7x10 Blocks และมีการใช้ Global Feature ที่ได้จากการทำ Vertical Stroke ซึ่งจากการศึกษางานวิจัยนี้ ได้กำหนดขอบเขตเอาไว้แต่ในส่วนของการแบ่งกลุ่มเท่านั้น ไม่ได้ทำการรู้จำตัวอักษรออกมา

กลุ่มข้อมูลที่ใช้ทดลองคือตัวอักษรภาษาไทยที่ประกอบไปด้วยพยัญชนะ 44 ตัว สระ 17 ตัว วรรณยุกต์ 4 ตัว และสัญลักษณ์ 2 ตัว วิธีการแบ่งกลุ่มมุ่งไปที่ ลักษณะโครงสร้างตัวอักษรที่อาศัยการพิจารณาและแบ่ง Blocks ที่ไม่เท่ากันขึ้นอยู่กับ Vertical Stroke ที่ได้ จากรูปที่ 2.6 อัลกอริทึมสำหรับการจัดรวมกลุ่มตัวอักษรลายมือเขียนภาษาไทยที่มีลักษณะ โครงสร้างคล้ายกันออกเป็นหมวดหมู่ เริ่มต้นจะมีการใช้ Confusion Matrix ในการวิเคราะห์ผลลัพธ์ที่ได้จาก Back-propagation Neural Network ตัวอักษรที่มีการรู้จำได้ต่ำ จะถูกจัดรวมกลุ่มเข้ากับตัวอักษรที่มีการรู้จำสูงสุด และมีการพิจารณาค่า Threshold ที่ระดับ ต่ำกว่า 70 หรือ ระหว่าง 70 กับ 90 หรือสูงกว่า 90

กลุ่มข้อมูลที่ใช้ในการทดลองได้แก่กลุ่มข้อมูล Isolate Character Set ของ NECTEC Corpus ซึ่งถูกเขียนโดยลายมือเขียนของ 68 คน แต่ละคนเขียนสองครั้ง และสแกนเข้าสู่คอมพิวเตอร์ ด้วยค่าความละเอียด 200 dpi โดยมีจำนวนทั้งหมด 9,020 ตัว และแบ่งเป็น Training Set 5,940 ตัว และ Testing Set 3,080 ตัว โดยความสามารถในการจัดกลุ่มแบ่งออกเป็น 21 กลุ่มที่มีรูปแบบตัวอักษรคล้ายคลึงกันตามตารางที่ 2.1 เปอร์เซ็นต์ความสามารถในการจัดรวมกลุ่มโดยเฉลี่ยเท่ากับ 97.33%



รูปที่ 2.6 อัลกอริทึมสำหรับการจัดรวมกลุ่มตัวอักษรลายมือเขียนภาษาไทย

ตารางที่ 2.1 ความสามารถในการจัดกลุ่มของตัวอักษรที่คล้ายคลึงกันในแต่ละกลุ่ม ทั้ง 21 กลุ่ม

กลุ่มที่	กลุ่มตัวอักษร
1	ก ก ฎ
2	ข ข ฃ บ ม ย ษ
3	ค ต ด ฅ
4	ง จ
5	ฉ ล ว อ ๑
6	ช ฌ ฐ ฑ ร ฮ
7	ณ ณ ญ
8	ฎ ฏ
9	ฑ ท ห
10	น พ
11	ป
12	ผ
13	ฝ
14	พ
15	ฟ
16	ศ ส พ
17	อุ ู

ตารางที่ 2.1 (ต่อ)

กลุ่มที่	กลุ่มตัวอักษร
18	อ อ
19	อิ อี อึ อื
20	อึ อื อึ อื
21	โ ไ

#### 2.2.4 An Object-Oriented Expert System for Thai Character Recognition [18]

งานวิจัยส่วนมากในปัจจุบันมีเทคนิคในการจัดแบ่งโดยใช้ลักษณะ โครงสร้างของตัวอักษร เช่นการแทน Code Value (Q-codes) สำหรับงานวิจัยนี้เสนอ Expert System สำหรับการรู้จำตัวอักษรไทย และใช้เทคนิค Knowledge Acquisition คือ ID3 Induction Algorithm เริ่มต้นพิจารณาที่ Character's center of gravity และ Q-codes ซึ่งเหมาะกับกลุ่มตัวอักษรที่เป็นลายเส้นต่อเนื่องกัน ตั้งแต่เริ่มต้นเขียนจนสิ้นสุดการเขียน 1 ตัวอักษร แล้วทำการกำหนด Basepoint ที่จุด Character's center of gravity จากนั้นจะมีการแบ่งกรอบตัวอักษรออกเป็น 9 ส่วน พิจารณาที่กรอบย่อยตัวอักษร 8 ส่วน โดยจะไม่นับรวมกรอบย่อยตัวอักษรที่มี Basepoint จากนั้นจะหาจุด center of gravity ของแต่ละส่วนที่ถูกแบ่งออกมา แล้วกำหนดค่าตาม Q-codes ดังตารางที่ 2.2 โดย Q-codes จะประกอบไปด้วย 16 ค่า ดังนั้นกรอบที่ถูกแบ่งออกทั้ง 9 กรอบจะมีค่าทั้งหมด 9 ค่าคือ Q1, Q2, Q3, ..., Q9 แล้วใช้ Decision Tree เป็นกฎในการจัดแบ่งอักษรภาษาไทย และระบบการรู้จำพัฒนาด้วย Object Oriented Programming Language Smalltalk โดยที่ Object Oriented System Shell ที่ผ่านการรู้จำ Learning ที่ใช้ในการทดลองนี้มีชื่อเรียกว่า LEX-shell ที่ประกอบไปด้วย 4 ส่วนหลัก ๆ ได้แก่ ส่วนที่ 1 Acquisition Module จะอ่าน Training Set จาก Human Experts เข้ามา ส่วนที่ 2 The Knowledge Base จะจัดเก็บผลที่ได้จากการจัดแบ่งกลุ่มของตัวอักษรภาษาไทย ส่วนที่ 3 The Consulting Module เป็นหน้าจอส่วนติดต่อกับผู้ใช้งานระบบที่จะเชื่อมต่อเข้ากับส่วนที่ 4 The Inference Engine เพื่อผ่านการรู้จำและได้ผลลัพธ์ที่ต้องการออกมา สำหรับค่าความถูกต้องของการรู้จำนั้นขึ้นกับความสมบูรณ์และความถูกต้องในการ Training

ตารางที่ 2.2 แสดง Q-Code และรูปแบบแต่ละแบบ

Q-Code	รูปแบบ	Q-Code	รูปแบบ
0	•	3	└•
1	• —	4	— •
2	•	5	— — •

ตารางที่ 2.2 (ต่อ)

Q-Code	รูปแบบ	Q-Code	รูปแบบ
6		B	
7		C	
8		D	
9		E	
A		F	

### 2.2.5 Recognition of Handprinted Thai Characters Using the Cavity Features of Character Based on Neural Network [19]

งานวิจัยนี้อธิบายวิธี Cavity Feature และ Back-propagation Neural Network ในการรู้จำตัวอักษร ซึ่งจะใช้ Mathematic Morphology ในการหา Cavity Feature และได้ออกมาเป็น Pattern ซึ่ง Cavity Feature จะมี 6 ประเภทคือ East (E), West (W), North (N), South (S), Center (C) และ Hole (H) เมื่อเราได้ภาพตัวอักษรออกมาเป็นกรอบตัวอักษรแล้วจะทำการแบ่งออกเป็น 4 ส่วนย่อย ได้แก่ Region 1 = บนขวา Region 2 = บนซ้าย Region 3 = ล่างซ้าย Region 4 = ล่างขวา โดยที่ Feature vector ประกอบไปด้วย Cavity Image ในแต่ละพื้นที่ย่อย ( $H_1... H_4, C_1... C_4, N_1... N_4, S_1... S_4, E_1... E_4$  and  $W_1... W_4$ ) และจำนวนประเภทของ Cavity Feature คือ ( $N_H, N_C, N_N, N_S, N_E, N_W$ ) และ Different levels คือ  $L_N$  แล้วผ่านเข้าสู่ Neural Network โดยที่พยายามแก้ปัญหาสำหรับตัวอักษรภาษาไทยที่เขียนด้วยตัวบรรจงและมีหัว

กระบวนการรู้จำแบ่งออกเป็น 3 ขั้นตอน ขั้นตอนแรกเริ่มจากแบ่งแยกออกจากประโยคเป็น 3 ระดับกลุ่มที่ต่างกัน จากนั้นพิจารณาที่ Cavity Feature ของแต่ละตัวอักษร แล้วคำนวณนับตัวเลขโดยใช้วิธีของ Euler Number ซึ่งขั้นตอนสุดท้ายจะใช้ Feature Code ของพื้นที่หลักที่หาออกมาส่งเข้าไปในกระบวนการรู้จำ Neural Network เพื่อแบ่งแยกแต่ละตัวอักษรออกมา แต่เนื่องจากตัวอักษรบางตัวก็ยังไม่สามารถบอกได้ว่าเป็นตัวอะไร จึงมีการใช้คุณลักษณะพิเศษแยกย่อยเช่น “ค” กับ “ค” ต่างกันที่หัวเข้า-ออก ก็ใช้ Cavity Feature พิจารณาแยกแยะออกมา และตัว “จ” กับ “บ” ก็พิจารณาระยะจากซ้าย-ขวา ความกว้างที่ไม่เท่ากัน

ผลการทดลองข้อมูลตัวอักษรไทยนั้นแบ่งออกเป็น 3 ระดับและแต่ละระดับรวมแล้วแบ่งเป็น 10 Classes ข้อมูลที่ใช้มาจากคน 40 คน แต่ละคนเขียน 80 ตัวอักษร จึงได้ข้อมูลตัวอักษรทั้งหมด 3,200 ตัว ซึ่งงานวิจัยนี้ไม่ได้ทำ Thinning character และไม่มีการทำ Edge Detection แต่มุ่งเน้นที่การพิจารณา Cavity Feature ผลของการรู้จำที่ได้คือ 98.3%

### 2.2.6 Off-line Handwritten Thai Characters from Word Script [20]

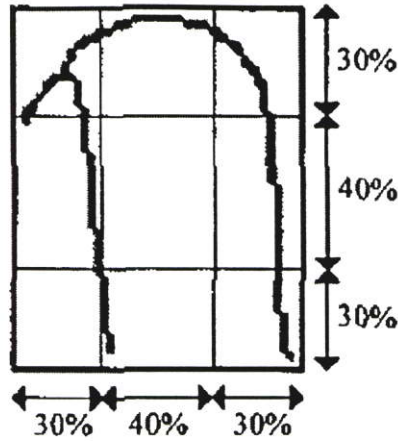
งานวิจัยนี้อาศัยโครงสร้างของ Loop ในการตัวอักษรไทยลายมือเขียนแบ่งออกเป็น 4 กลุ่มตัวอักษร และมีการใช้ Topological Properties of Stroke และลักษณะโครงสร้างของตัวอักษรภาษาไทย จากนั้นได้ออกมาเป็น Decision Tree หากแต่ว่ายังมีปัญหากับกลุ่มลายมือเขียนในรูปแบบที่ไม่มีหัว

### 2.2.7 Handwritten Thai Character Recognition Using Fuzzy Membership Function and Fuzzy ARTMAP [21]

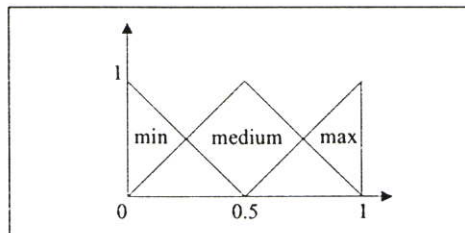
งานวิจัยนี้นำเสนอวิธีที่ใช้ในการรู้จำลายมือเขียนภาษาไทยแบบ Off-line โดยใช้หลักการของ Fuzzy Membership Function ในขั้นตอนของการทำ Feature Extraction เป็นการหาระดับความลึกของเส้นตัวอักษรจากกรอบตัวอักษร แล้วส่งผ่านเข้าสู่กระบวนการรู้จำของ Fuzzy ARTMAP โดยทั่วไปแล้วเอกสารภาษาไทยประกอบไปด้วย 5 ชนิดตัวอักษรได้แก่ พยัญชนะ สระ วรรณยุกต์ เลขอารบิก และสัญลักษณ์อื่น ๆ วิธีการเขียนตัวอักษรจะเขียนจากซ้ายไปขวา บนลงล่าง และแต่ละบรรทัดประกอบไปด้วย 4 ระดับได้แก่ above-upper level, upper level, central level และ lower level

ขั้นตอนการทำงานสำหรับการทดลองครั้งนี้แบ่งออกเป็น 6 ขั้นตอนคือ

1. สแกนเอกสารที่เขียนด้วยลายมือเขียนเข้าสู่คอมพิวเตอร์แล้วทำการแปลงเป็นรูปภาพ
2. กระบวนการก่อนการรู้จำจะแบ่งให้ได้กรอบตัวอักษรออกมา
3. การทำ Thinning Algorithm เพื่อลดขนาดของเส้นตัวอักษรให้บางลง
4. แบ่งกรอบตัวอักษรออกเป็น 9 ส่วน เริ่มจากซ้ายไปขวา และบนลงล่าง ตามรูปที่ 2.7
5. การทำ Feature Extraction ในแต่ละส่วนที่ถูกแบ่งออกจากกรอบตัวอักษรหนึ่ง โดยจะวัดค่าความลึกระหว่างเส้นตัวอักษรถึงขอบกรอบตัวอักษรที่ด้านนั้น ๆ และกำหนดระดับความลึกเป็น 5 ระดับและกำหนดตามค่าเปอร์เซ็นต์ความลึกวัดจากความกว้างของกรอบตัวอักษรได้แก่ ระดับที่ 1 มีค่า 0-20% ระดับที่ 2 มีค่า 21-40% ระดับที่ 3 มีค่า 41-60% ระดับที่ 4 มีค่า 61-80% และระดับที่ 5 มีค่า 81-100% แล้วจะทำการหาค่าเฉลี่ยออกมาสำหรับแต่ละ 9 ส่วนที่ถูกแบ่ง โดยแต่ละส่วนจะประกอบไปด้วย Left view, Top View, Right View และ Bottom View แล้วนำค่ามาพิจารณาใน Fuzzy Set DEPTH ดังแสดงตามรูปที่ 2.8
6. ขั้นตอนการรู้จำโดยใช้ Fuzzy ARTMAP Neural Network



รูปที่ 2.7 การแบ่งกรอบตัวอักษรออกเป็น 9 ส่วน



รูปที่ 2.8 The Membership Function of DEPTH

กลุ่มข้อมูลที่ใช้ในการทดลองมาจาก 2 แหล่งคือกลุ่มแรกมาจาก National Electronic and Computer Technology Center (On-line Thai Handwritten corpus) ซึ่งเป็นการรวบรวมจาก 70 คน โดยที่แต่ละคนเขียน 70 ตัวอักษร สำหรับกลุ่มที่ 2 มาจากการจัดเก็บเพิ่มเติมจากนักวิจัย ซึ่งประกอบไปด้วยลายมือเขียนจาก 4 คน ใน Training Set ประกอบไปด้วยการสุ่มขึ้นมา 50 ตัวอย่าง สำหรับการ Testing แบ่งออกเป็น 2 กลุ่มคือ กลุ่มที่ 1 เป็น 20 ตัวอย่างที่เหลือนอกเหนือจากการ Train ผลลัพธ์ของการรู้จำอยู่ที่ช่วง 39%-80% เฉลี่ยเท่ากับ 62.65% กลุ่มที่ 2 เป็นเอกสารภาษาไทยที่เขียนด้วยคน 4 คนแตกต่างกันไป ผลลัพธ์ของการรู้จำเฉลี่ยเท่ากับ 67.84%

### 2.2.8 Experimental results of Using Rough Sets for printed Thai Character

#### Recognition [22]

งานวิจัยนี้นำเสนอแนวทางในการรู้จำตัวพิมพ์อักษรไทยด้วยการใช้ Rough Set กลุ่มข้อมูลที่ใช้ในการรู้จำประกอบไปด้วยพยัญชนะไทย 42 ตัว ซึ่งไม่นับรวม 2 ตัวที่ไม่มีการนำมาใช้ โดยรูปแบบของตัวอักษรที่ใช้ในการทดลองมีทั้งหมด 7 ฟอนต์ได้แก่ Angsana, Browallia, Cordia, Dillenia, Eucrosia, Freesia, Iris และแต่ละฟอนต์มีขนาดต่างกัน 7 ขนาด จึงทำให้ได้ข้อมูลที่ใช้ทดสอบในการทดลองทั้งสิ้น 2,058 ตัวอักษร สำหรับงานวิจัยที่ผ่านมา ทฤษฎีของ Rough Set ได้มี

การนำไปประยุกต์ใช้ในกลุ่มของ Data Mining แต่ในงานวิจัยนี้ได้นำมาศึกษาและใช้งานร่วมกับการทำ Pattern Recognition เริ่มแรกจะกำหนด Attributes ของสมาชิกที่อยู่ใน Rough Set จากนั้นจะสร้าง Decision Making Rules ขึ้นมาจากค่า Attributes เหล่านั้นเพื่อใช้ในการรู้จำหรือการจัดแบ่ง Object

ภาพรวมของระบบประกอบไปด้วย 4 ขั้นตอนหลักได้แก่ ขั้นตอนที่ 1 การหา Attribute ของวัตถุจะเกิดขึ้นในกระบวนการก่อนการรู้จำมีการแบ่งกรอบภาพตัวอักษรที่มีขนาด 16x32 Pixels ออกเป็นกรอบย่อยขนาด 4x4 Pixels จะได้ทั้งหมด 32 กรอบย่อย ซึ่งจะเก็บค่าได้ทั้งหมด 32 Attributes ต่อหนึ่งภาพตัวอักษร ขั้นตอนที่ 2 เป็นการลดจำนวน Attributes ที่ไม่จำเป็นทิ้งไปด้วยวิธี Boolean Reasoning Algorithm และจะเหลือไว้เฉพาะ Attributes ที่จำเป็น ขั้นตอนที่ 3 เป็นส่วนของการสร้าง Decision rules จาก Attributes ที่ได้จากขั้นตอนที่ 2 ในรูปแบบของ If ... then ... ต่อจากนั้นขั้นตอนที่ 4 จะทำการ Classification โดยแบ่งกลุ่มจาก Rules ที่สร้างขึ้นมา และแก้ปัญหาด้วยวิธีการ Voting Algorithm

ผลการทดลองในงานวิจัยนี้ Testing Set ประกอบไปด้วยภาพตัวอักษรจำนวน 2,058 ตัวอักษรที่มีการพิมพ์จากเครื่อง HP LaserJet 6P ที่ค่าความละเอียด 300 dpi และสแกนเข้าสู่เครื่องคอมพิวเตอร์ด้วยเครื่อง HP Scan Jet 6100c ในรูปของ Binary black / white mode แล้วมีการสร้าง Decision Rules 3 Sets โดยแต่ละ rule สร้างมาจาก Training Set 3 Sets ที่ต่างกันออกไปคือกลุ่มแรกประกอบด้วยรูปแบบตัวอักษร 1 ฟอนต์ที่มีขนาดทั้งหมด 5 ขนาด รวมทั้งหมด 210 ภาพตัวอักษร กลุ่มที่สองประกอบด้วยรูปแบบตัวอักษร 7 ฟอนต์ที่มีขนาด 1 ขนาด รวมทั้งหมด 294 ภาพตัวอักษร กลุ่มที่สามประกอบด้วยรูปแบบตัวอักษร 7 ฟอนต์ที่มีขนาดทั้งหมด 7 ขนาด รวมทั้งหมด 588 ภาพตัวอักษร ผลที่ได้จากการรู้จำในกลุ่มที่ 1, 2 และ 3 คือ 46.20%, 63.15% และ 73.12% ตามลำดับ

### 2.2.9 Printed Thai Character Recognition using Fuzzy-Rough Sets [23]

งานวิจัยนี้นำเสนอวิธีการใช้ Fuzzy-Rough Sets ในการรู้จำตัวอักษรภาษาไทยซึ่งประกอบไปด้วยลำดับชั้นในการจัดแบ่งกลุ่ม 2 ลำดับชั้นคือ Coarse และ Fine Classification โดยขั้นตอนวิธีการที่ใช้แบ่งกลุ่มทั้ง 2 จะใช้วิธีการเดียวกัน แต่จะต่างกันที่ข้อมูลอินพุตที่ใช้ผ่านเข้าสู่กระบวนการในการแบ่งกลุ่ม

จะเห็นได้ว่าในการแบ่งกลุ่มลำดับชั้นที่ 1 (Coarse) จะใช้วิธี Projection ในการ Extract Feature ด้วยการหาค่า Projection ของทั้งกรอบภาพตัวอักษรออกมาทั้งในแนวแกน X และแนวแกน Y หลังจากนั้นจะได้ข้อมูล 2 ชุดสำหรับแต่ละภาพตัวอักษร ส่วนการแบ่งกลุ่มลำดับชั้นที่ 2 (Fine) ก็จะใช้วิธี Projection เช่นกัน แต่จะหาค่าออกมาแค่เพียงบางส่วนของภาพตัวอักษรที่มีลักษณะคล้ายกันมาก ซึ่งจะเป็นค่าชุดข้อมูลที่จัดอยู่ใน Coarse Class ที่ยังไม่สามารถระบุผลลัพธ์ออกมาได้ในขณะเดียวกันจากการทดลองก็มี 40 Coarse Groups ที่ไม่จำเป็นต้องส่งข้อมูลต่อขึ้นไปยังลำดับ

ขั้นที่ 2 ในขั้นตอนต่อมาข้อมูลที่ผ่านการทำ Projection แล้วจะถูก Normalization เพื่อให้มีค่าอยู่ในช่วง 0-1 หลังจากนั้นข้อมูลซึ่งถูก Normalization แล้วจะถูกกำหนดให้เป็น Attributes ของ Object ที่ใช้ใน Rough Set จากนั้นหาค่า Attribute สูงสุดและต่ำสุด

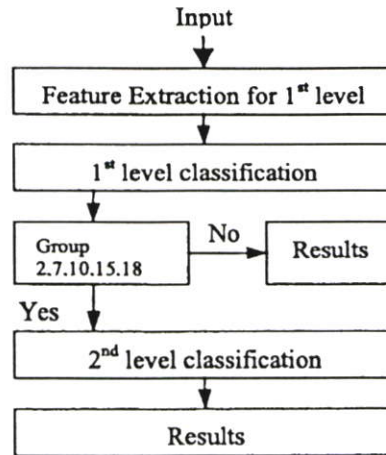
ภาพรวมของระบบแสดงดังรูปที่ 2.9 (a) เมื่อมีข้อมูลอินพุตเข้ามามีการทำ Feature Extraction ของรูปแบบตัวอักษรทั้งตัวอักษรแล้วส่งเข้าสู่ขั้นตอนของการแบ่งกลุ่มลำดับขั้นที่ 1 ซึ่งประกอบด้วยขั้นตอนตามรูปที่ 2.9 (b) คือเมื่อเข้าสู่ "1" Level Classification จะใช้ Rough Set's Lower Approximation ในการกำหนดกลุ่มต่างๆ สำหรับการทดลองครั้งนี้มี 37 กลุ่มในการแบ่งกลุ่มลำดับขั้นแรกและในแต่ละกลุ่มจะมี 1 ตัวอักษรที่กำหนดไว้ แต่จะมีเพียงแค่กลุ่มที่ 2, 7, 10, 15, 18 ที่จะมี 2-3 ตัวอักษรที่ยังไม่สามารถระบุผลลัพธ์ออกมาได้ดังรูปที่ 2.10 โดยทั้ง 5 กลุ่มนี้จะส่งต่อไปยังการแบ่งกลุ่มลำดับขั้นที่ 2 ซึ่งประกอบด้วยขั้นตอนตามรูปที่ 2.9 (c) คือใน 2<sup>nd</sup> Level Classification จะใช้วิธี Rough Set's Lower Approximation ในการกำหนดกลุ่มซึ่งชุดข้อมูลในกลุ่ม 2, 7, 10, 18 นี้จะแบ่งได้ 2 Fine Groups แต่กลุ่มที่ 18 แบ่งได้ 3 Fine Groups และได้ผลลัพธ์ที่ต้องการออกมา สำหรับการทดลองในงานวิจัยนี้ได้กำหนดค่า Threshold เท่ากับ 0.8 หากมีกลุ่มที่ได้ค่ามากกว่าค่า Threshold จะเลือกจัดเข้ากลุ่มที่มีค่าที่ได้สูงสุด

ในการแบ่งกลุ่มจำนวนข้อมูลที่ใช้ในการทดลองคือกลุ่ม Training Set ประกอบไปด้วยรูปแบบตัวอักษร 4 ฟอนต์ แต่ละฟอนต์มีขนาด 4 ขนาด และ Testing Set ประกอบด้วยรูปแบบตัวอักษร 7 ฟอนต์ แต่ละฟอนต์มีขนาด 7 ขนาด และมีการรวมเอากลุ่มตัวอักษรของ Training Set เข้าไปด้วย ค่าความถูกต้องในการรู้จำเท่ากับ 89%

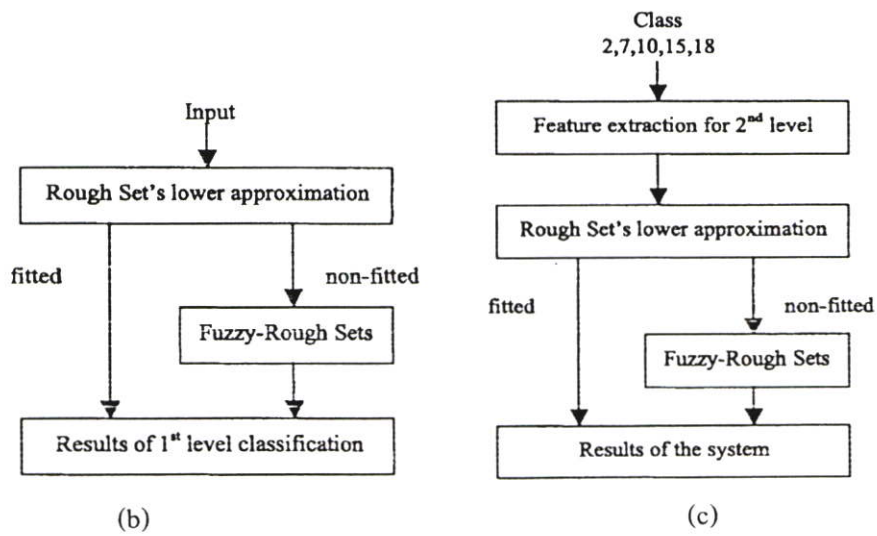
#### 2.2.10 On Thai Character Recognition [24]

งานวิจัยนี้นำเสนอแนวทางการรู้จำตัวอักษรไทยโดยใช้การแบ่งกลุ่ม 2 ขั้นตอน ซึ่งจะใช้ Back-propagation Multi-Layered Perceptron และ Penalized Discriminant Analysis (PDA) กระบวนการก่อนการรู้จำจะนำเอาคุณลักษณะที่ต้องการออกมาจากข้อมูลด้วยการใช้วิธี Simulated Light Sensitive (SLS) โดยมีการกำหนดค่า Threshold และเมื่อทำการ Projection กรอบภาพตัวอักษรแล้วพบว่าค่าที่ได้มีค่ามากกว่าค่า Threshold ก็จะกำหนดให้ Element State On แต่ถ้าไม่ใช่ก็จะกำหนดให้ Element State Off

จากนั้นเข้าสู่การแบ่งกลุ่มซึ่งแบ่งออกเป็น 2 ขั้นตอน ได้แก่ขั้นตอนที่ 1 จะแบ่ง Character Feature Vector ออกเป็นกลุ่มภาพตัวอักษร ทำการเปรียบเทียบกับ Representative Pattern ที่แบ่งออกเป็น 8 กลุ่มได้แก่ กลุ่มที่ 1 ประกอบด้วย "ก" "ด" "ภ" "ห" "อ" "ฮ" กลุ่มที่ 2 ประกอบด้วย "ข" "ฃ" "ร" "ท" กลุ่มที่ 3 ประกอบด้วย "ค" "ด" "ต" "ศ" "ท" "จ" กลุ่มที่ 4 ประกอบด้วย "ฉ" "ม" กลุ่มที่ 5 ประกอบด้วย "ง" กลุ่มที่ 6 ประกอบด้วย "จ" "ว" "ล" "ส" กลุ่มที่ 7 ประกอบด้วย "ฉ" "น" "บ" "ป" "ผ" "ฝ" "พ" "ฟ" "ย" "ย" "พ" กลุ่มที่ 8 ประกอบด้วย "ฉ" "ญ" "ฎ" "ฏ" "ฐ" "ฒ" "ณ"



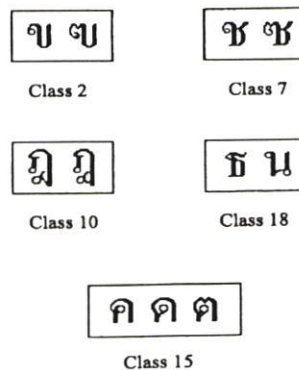
(a)



(b)

(c)

รูปที่ 2.9 โครงสร้างของระบบรู้จำตัวอักษร โดยใช้ Fuzzy-Rough Sets (a) ภาพโดยรวมของระบบ (b) รายละเอียดของการจัดแบ่งกลุ่มลำดับชั้นที่ 1 (c) รายละเอียดของการจัดแบ่งกลุ่มลำดับชั้นที่ 2



รูปที่ 2.10 กลุ่มทั้ง 5 กลุ่มที่มีสมาชิกประกอบไปด้วยตัวอักษรมากกว่า 1 ตัวอักษรที่แตกต่างกัน

แล้วเข้าสู่ขั้นตอนที่ 2 เป็นการพัฒนาประสิทธิภาพของการจัดแบ่งภาพตัวอักษรย่อยในแต่ละกลุ่มจากขั้นตอนแรก หลักการที่ใช้ในการจัดแบ่งกลุ่มทั้ง 2 ขั้นตอนคือ Back-propagation Neural Network และ PDA

กลุ่มข้อมูลที่ใช้ในการทดลองคือภาพตัวอักษร 44 ตัวและมีการใส่สัญญาณรบกวนแบบ Salt and Paper เข้าไปรวมทั้งหมด 4,100 ภาพตัวอักษร และแบ่งอย่างละครึ่งคือ Training Set = 2,050 ภาพตัวอักษรและ Testing Set = 2,050 ภาพตัวอักษร กรอบภาพตัวอักษรที่จัดเก็บมีขนาด 32x32 Pixels พร้อมทั้งในการแบ่งออกเป็นกลุ่ม ๆ นั้นอาศัย Three-Layer Multi-Layer Perceptron (MLP) จากผลการทดลองพบว่าวิธีของ MLP มีการรู้จำดีกว่า PDA

### 2.2.11 Thai OCR: A Neural Network Application [25]

Thai OCR นับว่าเป็นหนึ่งในโปรแกรมคอมพิวเตอร์ของประเทศไทยที่ใช้ในการรู้จำตัวอักษรไทย งานวิจัยนี้นำเสนอระบบการรู้จำทั้งระบบเริ่มตั้งแต่ขั้นตอนก่อนการรู้จำ ขั้นตอนการรู้จำ และขั้นตอนหลังการรู้จำ ในการแก้ปัญหาที่อาจจะเกิดขึ้นในก่อน หรือหลังจากการรู้จำของ Thai OCR ในส่วนของผลจากการทดลองแสดงให้เห็นว่าวิธีที่ใช้ในการพัฒนา Thai OCR ทำให้ความสามารถในการรู้จำดีขึ้นอยู่ในช่วง 90% - 95% ทั้งนี้หลักการของ ANN เป็นวิธีการที่ใช้ในการรู้จำ Pattern และเป็นวิธีที่เรียนรู้การรู้จำคล้ายกับรูปแบบการเรียนรู้การรู้จำของมนุษย์ คุณลักษณะของตัวอักษรภาษาไทยประกอบไปด้วยเส้นโค้ง ซิกแซก และวงกลม รวมทั้งบางตัวอักษรก็มีลักษณะคล้ายกันเช่น “ก” “ถ” “ภ”, “จ” “ช” “ซ” “ฅ” “บ” “ม” “ผ” “ฝ” “ป” “พ” “ฟ”, “ด” “ค” “ต” “ศ” เป็นต้น การทดลองประกอบด้วยตัวอักษรภาษาไทย 78 ตัวอักษรที่มีทิศทางการเขียนจากซ้ายไปขวา และในหน้ากระดาษจะมีทิศทางจากบนลงล่าง ซึ่งจะเขียนในบรรทัดเดียวกันแต่จะประกอบไปด้วย 4 ระดับ แล้วแต่ว่าประโยคนั้น มีตัวอักษรโดยอยู่ข้าง ด้วยเหตุนี้ในการรู้จำจะยากและแตกต่างไปจากการรู้จำภาษาอื่น ๆ ทั่วไป เนื่องจากในหนึ่งบรรทัดแบ่งออกเป็นหลายระดับ นอกจากนี้ Thai OCR ยังสามารถรู้จำ English OCR ได้ด้วย ดังนั้นกลุ่มข้อมูลที่ใช้ในการรู้จำของโปรแกรม Thai OCR ประกอบไปด้วย ตัวอักษรภาษาไทย ตัวอักษรภาษาอังกฤษตัวพิมพ์เล็ก ตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ และตัวเลขอารบิก =  $78 + 26 + 26 + 10 = 140$  ตัวอักษร แต่ในงานวิจัยครั้งนี้มุ่งเน้นไปที่การรู้จำตัวอักษรภาษาไทยเป็นหลัก

กระบวนการ Thai OCR เริ่มต้นจากการรับภาพเอกสารเข้าสู่เครื่องคอมพิวเตอร์ แล้วผลลัพธ์ที่ได้คือเอกสารข้อความที่ผ่านการรู้จำ โปรแกรมจึงไม่ทำงานแค่การรู้จำ แต่รวมไปถึงกระบวนการก่อนการรู้จำที่ทำการปรับภาพอักษรที่เข้ามา กำจัดสัญญาณรบกวน และอักษรเรียงการแบ่งบรรทัด และได้ออกมาเป็น Isolated Thai Character แล้วส่งผ่านต่อเข้าไปในกระบวนการรู้จำที่เป็น ANN รับเอาอินพุตภาพเข้าไปด้วยขนาดที่กำหนดไว้ในขั้นตอนก่อนการรู้จำที่ปรับให้ทุกภาพตัวอักษรที่เข้ามาไม่ว่าขนาดเท่าไรก็ตามให้มีขนาดเท่าที่กำหนดเหมือนกัน จากการศึกษา

พบว่าขนาดที่เหมาะสมสำหรับตัวพิมพ์อักษรไทยคือ 8x23 แต่เนื่องจากบางตัวอักษรมีขนาดกว้าง และบางตัวอักษรก็มีขนาดแคบ จึงมีการปรับขนาดเพิ่มอีก 3 ขนาดคือ 8x8, 8x12 และ 8x16 หลังจากนั้นขนาดของอินพุตจะได้รับการปรับก่อนที่จะส่งเข้าสู่ระบบในการ Train และ Test จากนั้นเมื่อรู้จำตัวอักษรออกมาแล้วก็จะส่งต่อเข้าสู่การทำ Post-processing เพื่อให้ได้เอกสารข้อความออกมาในรูปแบบ Text File ที่สามารถนำไปแก้ไขต่อได้ด้วยโปรแกรม Word Processing

สำหรับการทดลอง Thai OCR พัฒนาโดย NECTEC Software Technology Laboratory ภายใต้ Microsoft Windows มีการใช้ SNNS (Stuttgart Neural Network Simulator) บนเครื่อง Pentium 90 MHz ในการจำลองโมเดลระบบเครือข่าย กลุ่มตัวอักษรภาษาไทยมี 78 ตัวอักษรที่ใช้ในการทดลองประกอบไปด้วย 10 fonts ได้แก่ AngsanaUPC, BrowalliaUPC, CordiaUPC, DilleniaUPC, EucrosiaUPC, FreesiaUPC, IrisUPC, JasmineUPC, SB Busaba, TS Burrirum แต่ละฟอนต์มีขนาด 8 ขนาดที่แตกต่างกันตั้งแต่ 8 Point ถึง 22 Point (8, 10, 12, 14, 16, 18, 20, 22) และลักษณะ 4 แบบคือ Normal, Italic, Bold, Italic & Bold รวมทั้งหมด 2,496 ตัวอักษร ความสามารถในการรู้จำอยู่ในช่วง 90% - 95%

#### 2.2.12 On-line Thai-English Handwritten Character Recognition using Distinctive Feature [26]

งานวิจัยนี้นำเสนอการรู้จำการแบ่งกลุ่มลายมือเขียนภาษาไทยและภาษาอังกฤษแบบออนไลน์ด้วยวิธีการดึงเอาคุณลักษณะที่แตกต่างระหว่าง 2 ภาษา (Distinctive Feature Extraction) ในรูปแบบของ Decision Tree Diagram ซึ่งก่อนที่จะเข้าสู่การรู้จำจะมีการแบ่งแยกกลุ่มระหว่างภาษาไทยและภาษาอังกฤษ เพื่อพัฒนาประสิทธิภาพของการรู้จำในการลดความซับซ้อนและความถูกต้องในการรู้จำ เนื่องมาจากการรู้จำตัวอักษรของแต่ละภาษา ก็จะมีโครงสร้างตัวอักษรของแต่ละภาษาที่แตกต่างกันไป ดังนั้นหากต้องการรู้จำภาษาใด ๆ ก็เป็นเรื่องสำคัญที่จะต้องศึกษาลักษณะโครงสร้างพิเศษของภาษานั้น ๆ

สำหรับแนวคิดในงานวิจัยครั้งนี้เน้นที่การรู้จำลายมือเขียนมากกว่าหนึ่งภาษาคือภาษาไทยและภาษาอังกฤษที่เขียนลงบนอุปกรณ์ PDA (Personal Digital Assistant) หลักการพื้นฐานของกระบวนการดึงเอาคุณลักษณะที่แตกต่างกัน มีจุดมุ่งหมายเพื่อดึงเอาคุณลักษณะที่แตกต่างกันออกมาเป็นตัวบ่งบอกความแตกต่างของตัวอักษรสำหรับการกำหนดรูปแบบในการแบ่งกลุ่ม ทั้งนี้ยังไม่มีงานวิจัยที่แบ่งแยกลักษณะของตัวอักษรภาษาไทยและภาษาอังกฤษ

ในงานวิจัยนี้เริ่มต้นศึกษาที่คุณลักษณะหลักที่แบ่งกลุ่มตัวอักษรทุกตัวได้แก่ ข้อที่ 1. Number of Islands คือจำนวนชิ้นส่วนของตัวอักษรเช่น “ด” ประกอบไปด้วย 1 ชิ้นส่วนของตัวอักษรแต่ “จ” ประกอบไปด้วย 2 ชิ้นส่วนของตัวอักษร ข้อที่ 2. Number of Loops คือบริเวณที่เป็นเส้นทางการเขียนที่บรรจบกันเป็นรูป โดยแต่ละตัวอักษรสามารถมีได้มากกว่า 1 ลูปขึ้นไปหรือ

อาจจะไม่มีเลขก็ได้เช่น “C” ไม่มีรูป, “ข” มี 1 รูป, “B” มี 2 รูป และ “จ” มี 3 รูป ข้อที่ 3 Loop Head จะมีการแบ่งกรอบตัวอักษรออกเป็นสามระดับในแนวนอนที่เท่ากัน โดย Loop Head สามารถอยู่ในระดับใดก็ได้เช่น “ก” ไม่มี Loop Head, “P” มี Loop Head ที่ระดับบน, “อ” มี Loop Head ที่ระดับกลาง และ “เ” มี Loop Head ที่ระดับล่าง ข้อที่ 4. Loop Connection Point คือการที่มีเส้นตรงมาสัมผัสกับรูปโดยพิจารณาจุดสัมผัสที่เกิดขึ้นได้แก่ “ก้า” ไม่มีจุดเชื่อมต่อ, “ถ” มีจุดเชื่อมต่อทางซ้าย, “ภ” มีจุดเชื่อมต่อทางขวา และบางตัวอักษรมีจุดเชื่อมต่อซ้าย-ขวา

ต่อจากนั้นจะพิจารณาคูณลักษณะรองสำหรับแต่ละกลุ่มเพื่อแบ่งแยกย่อยในบางกลุ่มตัวอักษรได้แก่ ข้อที่ 1. Active Region Feature จะทำการแบ่งกรอบตัวอักษรออกเป็นขนาด 2 x 2 แล้วทำการหา Endpoint ว่าอยู่ในพื้นที่บริเวณใดในทั้งหมด 4 ส่วน จะเรียกพื้นที่ที่มีจุด Endpoint ว่า Active Region โดยกำหนด Region 1, 2, 3, 4 จากขวาไปซ้าย และบนลงล่างเช่น “Q” มี Active Region 4 ข้อที่ 2. Loop-Width-to-Loop-Height Ratio เช่นการพิจารณาระหว่าง “อ” “อิ” ในขั้นตอนสุดท้ายของการรู้จำ ข้อที่ 3. Character Ripples คือลักษณะเส้นตัวอักษรที่เป็นคลื่นหักซึ่งพบได้ทั้งในภาษาอังกฤษและภาษาไทย โดยที่ภาษาอังกฤษจะพบเฉพาะคลื่นในแนวตั้งเช่น “m” แต่ภาษาไทยส่วนมากจะพบคลื่นในแนวนอนเช่น “ย”

งานวิจัยนี้ได้พิจารณา 6 แบบที่ใช้ในการแบ่งแยกระหว่างตัวอักษรภาษาไทยและตัวอักษรภาษาอังกฤษได้แก่ แบบที่ 1 Level of Character ที่มีการแบ่งออกเป็น 3 ระดับ โดยตัวอักษรภาษาอังกฤษจะอยู่ในระดับกลางเท่านั้น แต่ตัวอักษรภาษาไทยที่ประกอบไปด้วยพยัญชนะส่วนมากจะอยู่ในระดับกลางในขณะที่สระตัวอักษรภาษาไทยจะมีอยู่ปรากฏในทั้ง 3 ระดับ แบบที่ 2 Loop-to-Character-Area Ratio พิจารณาว่าพื้นที่รูปของตัวอักษรไทยเมื่อเทียบกับพื้นที่ตัวอักษรทั้งหมดจะมีอัตราส่วนน้อยกว่าพื้นที่รูปของตัวอักษรอังกฤษเมื่อเทียบกับพื้นที่ตัวอักษรทั้งหมดเช่น “น” กับ “B” แบบที่ 3 Loop-to-Character-Width Ratio เมื่อพิจารณาที่ความกว้างของรูปที่เกิดขึ้นกับความกว้างของตัวอักษรแล้ว ภาษาไทยนั้นมีความกว้างของรูปจะน้อยกว่า 40% ในขณะที่ภาษาอังกฤษมีความกว้างของรูปมากกว่า 50% เช่น “บ” กับ “P” แบบที่ 4 Loop-to-Character-Height Ratio เมื่อพิจารณาที่ความสูงของรูปที่เกิดขึ้นกับความสูงของตัวอักษรแล้ว ภาษาไทยนั้นมีความสูงของรูปจะน้อยกว่า 40% ในขณะที่ภาษาอังกฤษมีความสูงของรูปมากกว่า 50% เช่น “บ” กับ “P” แบบที่ 5 Length-of-Loop-Line-to-Character-Stroke-Length Ratio พิจารณาที่อัตราของความยาวของรูปเมื่อเทียบกับความยาวของเส้นตัวอักษรตั้งแต่จุดเริ่มจนถึงจุดปลายสำหรับตัวอักษรภาษาไทยจะน้อยกว่า 40% แต่ตัวอักษรภาษาอังกฤษจะมากกว่า 50% เช่น “ค” “R” แบบที่ 6 Loop Generated Point สำหรับตัวอักษรไทยรูปแบบข้อมูลจะพบว่าเริ่มต้นด้วยรูปที่เป็นส่วนหัวของตัวอักษร ในขณะที่รูปแบบข้อมูลของตัวอักษรภาษาอังกฤษจะพบรูปที่ส่วนท้ายของตัวอักษร

ข้อมูลที่ใช้ในการทดลองในงานวิจัยนี้ประกอบไปด้วยตัวอักษรภาษาไทย 70 ตัวได้แก่พยัญชนะ 44 ตัวและสระ 26 ตัว และตัวอักษรภาษาอังกฤษ 52 ตัวได้แก่ตัวพิมพ์ใหญ่ 26 ตัวและ

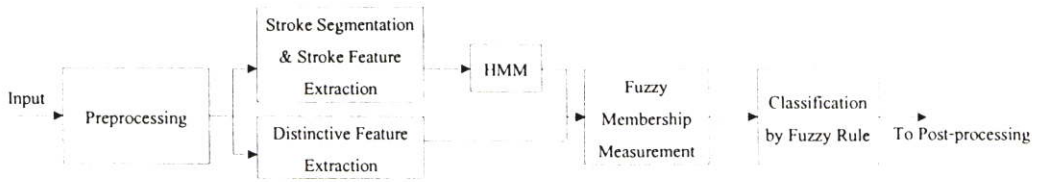
ตัวพิมพ์เล็ก 26 ตัว ดังนั้นรวมทั้งหมด 122 ตัวอักษรที่ใช้ในการทดลอง และมีการสร้าง Decision Tree Diagram โดยกลุ่มแรกพิจารณากลุ่มตัวอักษรที่ไม่มีหัว จะพบว่ามีความคล้ายภาษาไทยอยู่ 5 ตัว ซึ่งจะใช้ Active Region ในการแบ่งแยกย่อยต่อไป กลุ่มที่ 2 คือกลุ่มตัวอักษรที่มีรูปจะสามารถแบ่งต่อไปอีกได้สองกลุ่มคือกลุ่มที่ 3 และกลุ่มที่ 4 โดยที่ใช้ Loop-to-Character-Width Ratio เป็นตัวพิจารณา ซึ่งกลุ่มที่ 3 จะมี Loop-to-Character-Width Ratio น้อยกว่า 70% และพบว่าไม่มีแต่ตัวอักษรภาษาไทย ส่วนกลุ่มที่ 4 จะมี Loop-to-Character-Width Ratio มากกว่า 70% พบว่ามีทั้งตัวอักษรไทยและอังกฤษ ต่อจากนั้นนำกลุ่มที่ 4 มาพิจารณาต่อด้วย Loop-to-Character-Height Ratio จะได้ว่ากลุ่มที่ 5 มี Loop-to-Character-Height Ratio มากกว่า 80% ประกอบไปด้วยเฉพาะตัวอักษรภาษาอังกฤษ ส่วนกลุ่มที่ 6 มี Loop-to-Character-Height Ratio น้อยกว่า 80% นั้นประกอบไปด้วยภาษาไทยและอังกฤษ จากนั้นนำกลุ่มที่ 6 มาพิจารณาด้วย Loop-Width-to-Loop-Height Ratio จะได้ว่ากลุ่มที่ 7 มี Loop-Width-to-Loop-Height Ratio น้อยกว่า 70% ประกอบไปด้วยเฉพาะตัวอักษรภาษาไทย และกลุ่มที่ 8 มี Loop-Width-to-Loop-Height Ratio มากกว่า 70% ประกอบไปด้วยตัวอักษรไทยและอังกฤษ

จากผลการทดลองสามารถรู้จำในการจัดแบ่งกลุ่มตัวอักษรภาษาไทยและตัวอักษรภาษาอังกฤษได้ 86.34% และ 95.42% ตามลำดับ และในการรู้จำทั้งภาษาไทยและภาษาอังกฤษรวมกันได้ 90.21%

### 2.2.13 On-line Thai Handwritten Character Recognition using Hidden Markov

#### Model and Fuzzy Logic [27]

งานวิจัยนี้นำเสนอแนวทางการรู้จำตัวอักษรลายมือเขียนแบบออนไลน์ ที่มีการศึกษาวิธีการรู้จำให้ดีขึ้น โดยการเพิ่ม การจัดแบ่งกลุ่มด้วย Fuzzy Logic เข้าไปก่อนที่จะใช้การรู้จำแบบ HMM เนื่องจากอัตราการรู้จำตัวอักษรมีผลมาจากการแบ่งกลุ่มตัวอักษรไทยก่อนเข้าสู่การรู้จำ ดังนั้นปัญหาหรือข้อบกพร่องที่ตามมาในขั้นตอนการจัดแบ่งกลุ่มคือความยากในการสร้างกลุ่มของกฎให้ครอบคลุมทุกรูปแบบวิธีการเขียนด้วยมือทั้งหมด โดยกฎที่กำหนดขึ้นจากการสังเกตพบว่าในหลายงานวิจัยเมื่อมีการจัดแบ่งกลุ่มแล้ว ยังคงมีตัวอักษรที่มีรูปแบบที่คล้ายคลึงกันอยู่มากเช่น “ค” กับ “ก” พบว่ามีแค่ส่วนที่เป็นรอยหยักที่ด้านบนของตัวอักษรที่ทำให้ตัวอักษรทั้งสองตัวต่างกัน หรือ “ก” “ภ” “ด” ก็มีเพียงแค่จุดเริ่มต้นของตัวอักษรที่ต่างกัน เป็นเหตุให้งานวิจัยหลาย ๆ งานวิจัยที่ผ่านมาแก้ปัญหาด้วยการอาศัยคุณลักษณะพิเศษเฉพาะในการจัดแบ่งกลุ่มย่อย ซึ่งในทางปฏิบัติ นั้นเป็นเรื่องที่ยากในการสร้างกฎสำหรับการแบ่งกลุ่มเหล่านี้ขึ้นมาเพื่อที่จะให้ครอบคลุมทุกรูปแบบวิธีการเขียน ฉะนั้นในงานวิจัยนี้จึงนำเอา HMM มาใช้งานร่วมกับ Fuzzy เพื่อทำให้ได้ผลจากการรู้จำที่ดีขึ้น งานวิจัยนี้อธิบายแนวทางการใช้ Block Diagram ดังรูปที่ 2.11



รูปที่ 2.11 ระบบการรู้จำตัวอักษรลายมือเขียนแบบออนไลน์โดยใช้ HMM และ Fuzzy Logic

จากรูปที่ 2.11 เมื่อมีการรับค่าอินพุตที่เป็นลายมือเขียนเข้ามาแล้ว เริ่มต้นจะผ่านกระบวนการก่อนการรู้จำ หลังจากนั้นจะส่งข้อมูลของเส้นตัวอักษรลายมือเขียนที่ได้ไปยัง 2 กระบวนการได้แก่

- กระบวนการที่ 1 จะดึงเอาข้อมูลที่เกิดจากเส้นตัวอักษรทั้งตัวมาเข้าสู่การใช้ Stroke Segmentation & Stroke Feature Extraction ซึ่งจะพิจารณาระหว่างจุดข้อมูลบนเส้นตัวอักษรในการเขียนหากมีค่ามุมมากกว่าค่ามุม Threshold ที่กำหนดไว้ก็จะถือเอาจุดนั้นตัดออกมาจากเส้นตัวอักษรเดิมมาเป็นส่วนย่อยของตัวอักษร โดยที่หากกำหนดค่ามุม Threshold ต่ำ เส้นที่อยู่ในส่วนเดียวกันจะออกมาใกล้เคียงกับเส้นตรง ในขั้นตอนนี้ส่วนย่อยของเส้นตัวอักษรที่มีการตัดแบ่งแยกออกมาจะเรียกแต่ละส่วนว่า Stroke Segmentation

ต่อจากนั้นจะเก็บค่าลักษณะพิเศษของแต่ละส่วนแบ่งย่อย 4 ค่าได้แก่ จุดพิกัดศูนย์กลางของส่วนแบ่งย่อย มุมจากจุดเริ่มต้นถึงจุดปลายของส่วนแบ่งย่อย ค่าความแตกต่างของมุมสำหรับส่วนแบ่งย่อยนั้นกับส่วนแบ่งย่อยก่อนหน้า และความยาวของส่วนแบ่งย่อย ต่อจากนั้นใช้วิธี HMM

- กระบวนการที่ 2 จะมีการดึงเอาลักษณะพิเศษออกมาจากตัวอักษรเพื่อจัดแบ่งกลุ่มตัวอักษรภาษาไทย นับว่าเป็นวิธีการที่งานวิจัยนี้นำเสนอขึ้นมาให้มีการทำงานควบคู่กับ HMM เพิ่มไปด้วย ซึ่งจะมีคุณลักษณะพิเศษอยู่ 3 อย่างที่ใช้ในการพิจารณาคืออย่างแรก Character Head เป็นส่วนหัวมีลักษณะเป็นรูปที่เกิดขึ้นที่จุดเริ่มต้นของตัวอักษร สังเกตได้ว่าเป็นสิ่งที่แบ่งแยก “ก” เป็นตัวที่ไม่มีหัวออกจาก “ด” ที่มีรูปเกิดขึ้นในทิศทางตามเข็มนาฬิกา และ “ภ” ที่มีรูปเกิดขึ้นในทิศทางทวนเข็มนาฬิกา อย่างที่สอง Starting and End-point Locations เช่น “ร” กับ “ว” หรือ “อ” กับ “ฮ” พบว่าแม้จะมีจุดเริ่มแบบเดียวกัน แต่มีจุดปลายของตัวอักษรอยู่ที่บริเวณต่างกัน อย่างที่สาม Curl and Notch of Character เช่น “ค” กับ “ศ” หรือ “ม” กับ “ฉ” หรือ “ฎ” กับ “ฏ” เป็นต้น จากนั้นจะพิจารณา Fuzzy Membership Measures จากค่า Posteriori Probability สำหรับ Output ที่ออกมาจาก HMM และค่า Character Head Measure จะมาจากการแบ่งตัวอักษรแต่ละตัวออกเป็นสองส่วนคือ ส่วนหัวและตัวอักษร โดยพิจารณาจากการใช้การแบ่ง Sub-Stroke และค่า Starting and End-point Locations Measure ซึ่งจะกำหนดแบ่งกรอบตัวอักษรออกเป็น 9 ส่วน คือกรอบที่มีขนาด 3x3 จากนั้นพิจารณาเป็น 2 ส่วนคือแนวนอนและแนวตั้ง โดยพื้นที่แนวนอนประกอบด้วย Left Variable, Center Variable, Right Variable ส่วนพื้นที่แนวตั้งประกอบด้วย Top Variable, Middle

Variable, Bottom Variable และค่า Curl and Notch Measure จะพิจารณาจากรอยหยักที่เกิดบนแต่ละส่วนของทั้งตัวอักษรและสามารถที่จะแบ่งกลุ่มได้สามกลุ่มคือกลุ่มที่ 1 รอยหยักที่ด้านบนตัวอักษร ได้แก่ “จ”, “ด”, “ม”, “ต”, “ฒ”, “ช” และ “ต” กลุ่มที่ 2 รอยหยักที่ด้านล่างตัวอักษร ได้แก่ “ฎ”, “ฐ”, “ผ”, “ฝ”, “พ”, “ฟ” และ “พ” กลุ่มที่ 3 รอยหยักที่ด้านซ้ายของตัวอักษร ได้แก่ “ย” ต่อจากนั้นเป็นขั้นตอนของการจัดแบ่งกลุ่มตาม Fuzzy Rule โดยพิจารณารวมทั้ง 3 ค่าข้างต้นที่ได้ของแต่ละตัวอักษรค่าอินพุตที่เข้ามา และพิจารณาจากค่าที่น้อยที่สุดที่ได้ออกมาเพื่อหาผลลัพธ์ที่ได้จากการรู้จำตัวอักษรนั้น

ผลการทดลองตัวอักษรที่ใช้ในการทดลองมาจาก 44 ตัวอักษรไทย โดยกลุ่มข้อมูลใน Training Set ประกอบด้วยจำนวนตัวอักษรทั้งสิ้น 13,608 ตัวอักษรจากคนที่เขียนลายมือเขียน 37 คน สำหรับข้อมูลที่ใช้ใน Testing Set ประกอบด้วย 7,664 ตัวอักษรจากคนที่เขียนลายมือเขียน 21 คน ผลจากการทดลองเมื่อเทียบกับ HMM มีความสามารถในการรู้จำได้ 89.3% แต่วิธีที่ศึกษาและนำเสนอด้วย HMM + Fuzzy นั้นมีความสามารถในการรู้จำได้ถึง 92.1%

#### 2.2.14 Online Thai Handwritten Character Recognition using Hidden Markov

##### Models and Support Vector Machines [28]

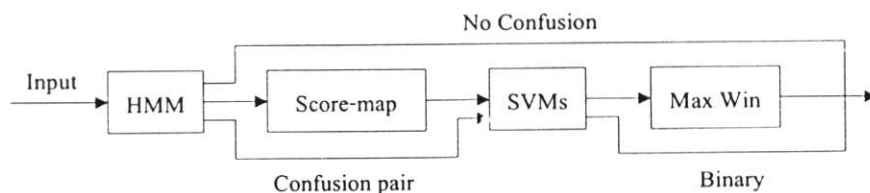
งานวิจัยนี้นำเสนอแนวทางของวิธี Support Vector Machines (SVM) เข้ามาใช้งานร่วมกับ Hidden Markov Models (HMM) โดยพิจารณาจาก Confusion Matrix ของ HMM หากพบว่ามี Confused Candidates มีค่ามากกว่าหนึ่งค่าแสดงว่ายังไม่สามารถรู้จำกลุ่มนั้นออกมาได้ ก็จะใช้วิธี SVM เข้าทำงานร่วมด้วยเพื่อการเพิ่มประสิทธิภาพในการรู้จำ

ระบบประกอบไปด้วยขั้นตอนก่อนการรู้จำ ลักษณะของข้อมูลที่ได้มาจากลายมือเขียนแบบออนไลน์นั้นจะมีข้อมูลที่ได้มาคือลำดับของจุดพิกัดที่เกิดจากการเคลื่อนที่ของปากกาเขียนที่รับเข้ามาจากอุปกรณ์รับข้อมูลเช่น Tablet อย่างไรก็ตามไม่ว่าทุกจุดจะถูกเก็บและส่งเข้ามา แต่จะพยายามลดจำนวนจุดที่รับเข้ามา ดังนั้นในกระบวนการนี้จะลดจุดพิกัดที่ซ้ำกันบางส่วนออกไปหรือนับได้ว่าเป็นจุดที่เปลี่ยนแปลงน้อยที่สุดออก เพื่อที่จะหาระยะทางของเส้นตัวอักษรจากลายมือเขียนให้ได้ระยะที่สั้นที่สุด จากนั้นจะเข้าสู่กระบวนการ Feature Extraction ลำดับข้อมูลของแต่ละตัวอักษรจะถูกแบ่งออกเป็น Sub-Stroke ด้วยวิธี Stroke Segmentation และจะจัดเก็บค่า 4 ค่าออกมา ได้แก่ จุดพิกัดศูนย์กลางของส่วนแบ่งย่อย มุมจากจุดเริ่มต้นถึงจุดปลายของส่วนแบ่งย่อย ค่าความแตกต่างของมุมสำหรับส่วนแบ่งย่อยนั้นกับส่วนแบ่งย่อยก่อนหน้า และความยาวของส่วนแบ่งย่อย แล้วส่งต่อเข้าสู่ HMM ที่มีโครงสร้างโมเดล Left-to-Right with No Skip และแต่ละ State Output ที่แบ่งแยกออกมาได้คือ Gaussian Components ของ Score-map เนื่องจาก SVM เป็น Binary Classifier ดังนั้นจึงมีการประยุกต์เอา SVM มาใช้ใน Multi-Classification โดยเลือกใช้ Max Win Algorithm เพราะจำนวนของกลุ่มที่ได้มีขนาดไม่ใหญ่ พิจารณาจาก HMMs Confusion matrix และ

ผลลัพธ์จากการรู้จำจะจัดอยู่ในกลุ่มที่มีค่า Maximum Number of Votes ทั้งนี้ความยาวของขนาดอินพุตที่รับเข้ามามีค่าไม่คงที่แน่นอน และวิธีการใช้ SVM จะต้องกำหนดให้ค่าอินพุตที่เข้ามานั้นมีขนาดที่คงที่เท่ากัน

งานวิจัยนี้จึงนำเสนอการทำ Normalization of the Feature Vectors ที่ประกอบด้วย 4 ขั้นตอนได้แก่ Sequence Length Normalization, Normalizing the Score-maps for Training and Testing, Score-space Whitening และ Spherical Normalization ในระบบที่นำเสนอในงานวิจัยนี้เป็นไปตามรูปที่ 2.12 HMMs ใช้ในการจัดแบ่งลำดับข้อมูลที่ได้รับเข้ามาจากอินพุตในขั้นตอนแรก และวิเคราะห์คาดคะเน Model Parameter สำหรับ Score-map ออกมา จากนั้น SVMs with Score-space Kernel จะเข้ามาช่วยในการแบ่งกลุ่มที่ยังไม่สามารถระบุผลลัพธ์ในการรู้จำในขั้นตอนที่สอง โดยมีการใช้ Max Win Algorithm ใน Multi-class ถ้าจำนวน Input Candidate of SVMs มีระบุมากกว่า 2 Classes

สำหรับผลการทดลองครั้งนี้ข้อมูลที่ใช้คือตัวอักษรภาษาไทย 42 ตัวอักษร โดยไม่นับรวม “จ” และ “ค” เพราะปัจจุบันไม่มีการนำมาใช้แล้ว ในกลุ่ม Training Set ประกอบด้วยลายมือเขียนจาก 31 คน รวมทั้งหมด 14,557 ตัวอักษร และกลุ่ม Testing Set ประกอบด้วยลายมือเขียนจากใน Training Set จำนวน 10,844 ตัวอักษร และรวมกับ 7,812 ตัวอักษรจากลายมือเขียนของ 62 คน ผลจากการรู้จำโดยใช้วิธีการของ HMM = 89.9% แต่สำหรับการทดลองครั้งนี้มี 2 วิธีคือ HMM/SVM/LR (HMM/SVM with Likelihood Ratio Score-space) = 88.1% และ HMM/SVM/SLR (HMM/SVM with Symmetric Likelihood Ratio Score-space) = 92.5%



รูปที่ 2.12 ระบบการรู้จำโดยอาศัยวิธี Support Vector Machines และ Hidden Markov Models

### 2.2.15 Character Recognition System for Cellular Phone with Camera [29]

งานวิจัยนี้นำเสนอระบบการรู้จำตัวอักษรด้วยกล้องถ่ายรูปที่นำไปประยุกต์ใช้งานบน Mobile Device อย่างเช่น PDA และ โทรศัพท์มือถือที่มีกล้องดิจิทัล โดยเริ่มจากการพัฒนาระบบการรู้จำตัวอักษรด้วยกล้องถ่ายรูปสำหรับเครื่อง PC โดยอาศัยเทคโนโลยีต่าง ๆ เช่น Image Enhancement, Local Adaptive Binarization และ Blob Coloring ในการดึงภาพพื้นที่ตัวอักษรและกำจัดสัญญาณรบกวนที่อาจจะเกิดจากการจับภาพจากกล้องถ่ายรูป หลังจากนั้นจะแปลงระบบ OCR

System บนเครื่อง PC ให้อยู่ในรูปแบบ Embedded OCR System สำหรับโทรศัพท์มือถือ โดยที่มี Function การทำงานมากมายที่พัฒนาขึ้นมาโดยเฉพาะที่ใช้งานบนอุปกรณ์สื่อสารเคลื่อนที่ ที่ไม่ได้มีการคำนวณเกิดขึ้นจริงเนื่องจากพื้นที่ของหน่วยความจำและทรัพยากรของเครื่องที่มีอยู่อย่างจำกัด ในงานวิจัยนี้มุ่งเน้นไปที่ปัญหาต่าง ๆ เหล่านี้และกลไกการทำงานของวิธีการรู้จำที่นำเสนอในอดีตการใช้งานของโทรศัพท์คือการพูดคุยสนทนาและต่อมาการสื่อสารทางมือถือได้รับความนิยมมากขึ้น จุดมุ่งหมายกลายมาเป็นการจัดการและจัดเก็บข้อมูลส่วนตัวบนโทรศัพท์มือถือ รวมไปถึงการพัฒนาประมวลผลดิจิทัลเข้ากับโทรศัพท์มือถือหรืออุปกรณ์สื่อสารแบบเคลื่อนที่ไปมาได้ อย่างเช่น PDA และสิ่งที่พัฒนาตามมาก็คือโปรแกรมประยุกต์ต่าง ๆ เช่น Navigation System, Smart Tour Guide, Robot Automatic Traveling เป็นต้น อย่างไรก็ตามในการจับภาพ (Capture) ของกล้องยังคงมีสัญญาณรบกวนที่เกิดจากความสว่างและความคมชัดต่ำอันเนื่องมาจากระดับแสงที่ไม่คงที่ จึงทำให้ยากในการหาบริเวณของพื้นที่ตัวอักษร และรู้จำตัวอักษรนั้น ๆ

ในงานวิจัยนี้เป็นการรู้จำตัวอักษรจากการถ่ายภาพเอกสารด้วยกล้องดิจิทัล และนำเสนอวิธี Locally Adaptive Binarization รวมทั้งการทำ Image Enhancement Algorithm ก่อนที่จะทำ Binarization เพื่อช่วยในการปรับสภาพแวดล้อมของแสงที่เกิดให้คงที่สม่ำเสมอ ระบบการรู้จำตัวอักษรสำหรับภาพจากกล้องถ่ายรูปบนมือถือนั้นประกอบไปด้วย 3 ขั้นตอนหลักได้แก่ Preprocessing, Segmentation และ Recognition โดยที่กระบวนการ Preprocessing นั้นจะแตกต่างจากการจัดการกับภาพที่สแกนเข้ามาสู่คอมพิวเตอร์ตรงที่ภาพที่ถ่ายจากกล้องนั้นค่อนข้างมีความแตกต่างและความสว่างของภาพที่ได้ไม่เท่ากันทำให้สิ่งเหล่านั้นเกิดเป็นสัญญาณรบกวนที่ทำให้ยากต่อการค้นหาพื้นที่บริเวณของตัวอักษรออกมาได้ ประสิทธิภาพของกระบวนการก่อนการรู้จำจึงขึ้นอยู่กับความสามารถในการกำจัดเอาสัญญาณรบกวนเหล่านี้ออกไปได้มากน้อยเพียงใด ดังนั้นในขั้นตอนของกระบวนการก่อนการรู้จำนี้แบ่งออกเป็นการแปลงภาพที่ถ่ายมาให้เป็น Gray Level Image แล้วมีการทำ Image Enhancement จากนั้นเข้าสู่การทำ Binarization โดยใช้วิธี Locally Adaptive Threshold เพื่อที่จะสามารถแยกภาพตัวอักษรออกจากพื้นที่ข้างหลังได้ส่วนหนึ่ง ต่อจากนั้นจะทำการกำจัดสัญญาณรบกวนด้วยกระบวนการ Blob Coloring โดยวิเคราะห์จากตำแหน่งและขนาดของ Blob ที่เกิดขึ้น และเมื่อผ่านกระบวนการ Preprocessing แล้วก็จะเป็นการ Segmentation ซึ่งจะดึงเอาเฉพาะกรอบภาพตัวอักษรที่ใช้ในการรู้จำตัวอักษรแต่ละตัวออกมา โดยจะแบ่ง Binary Image ที่ได้ออกเป็นบรรทัด จากนั้นแบ่งออกเป็นแต่ละคำและแบ่งออกเป็นแต่ละตัวอักษร โดยในงานวิจัยนี้ใช้วิธี Projection ในการทำ Segmentation โดยเริ่มจากการทำ Projection ในแนวนอนเพื่อให้ได้แต่ละบรรทัดแยกออกมาจากกัน จากนั้นแบ่งแต่ละคำออกจากกันโดยอาศัยการทำ Projection ในแนวตั้งจะได้แต่ละคำออกมา และใช้การทำ Projection แนวตั้งในแต่ละคำก็จะได้เป็นกรอบตัวอักษรแต่ละตัวออกมา

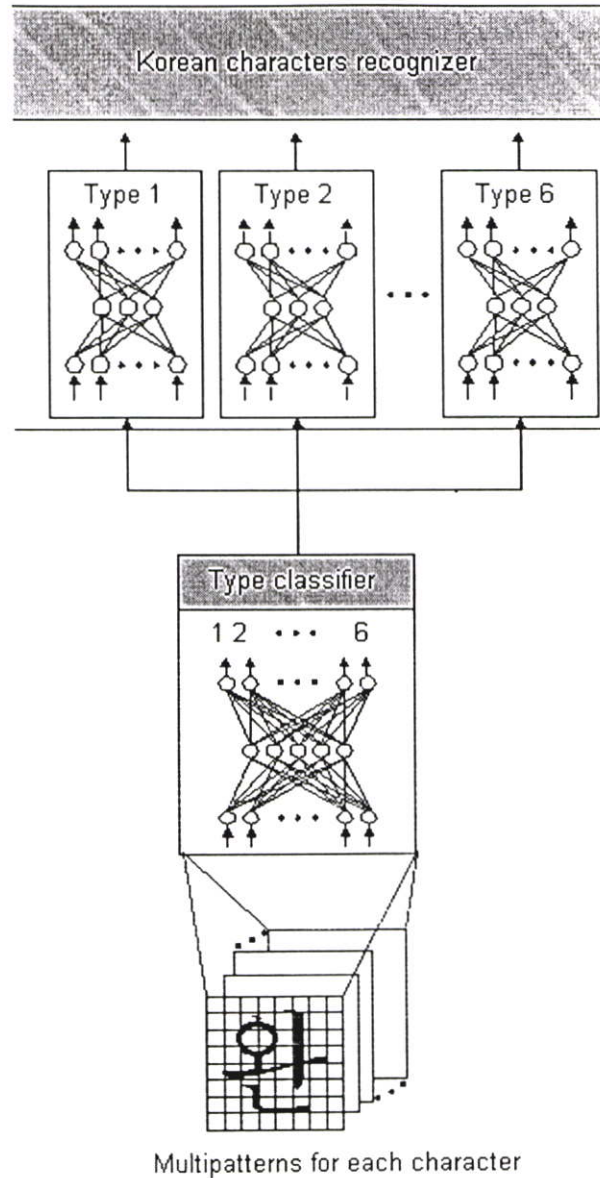
ตัวอักษรที่ใช้ในการรู้จำสำหรับงานวิจัยนี้คือตัวอักษรภาษาเกาหลีที่มีชื่อเรียกว่า “Hangul” โดยโครงสร้างของภาษาประกอบไปด้วยพยัญชนะและสระ แบ่งแยกลักษณะโครงสร้างของตัวอักษรออกเป็น 6 รูปแบบตามรูปที่ 2.13

ขั้นตอนของกระบวนการรู้จำ แต่ละกรอบภาพตัวอักษรที่ได้จากการทำ Segmentation จะมีการดึงเอา Feature ออกมาแล้วส่งเข้าสู่ระบบการรู้จำที่อาศัยวิธี Neural Network โดยที่จะมีการทำ Normalization เพื่อให้กรอบภาพตัวอักษรแต่ละภาพมีขนาดเท่ากัน จากนั้นทำ Feature Extraction ด้วยการใช้ Mesh Feature คือการรวมพิกเซลของแต่ละส่วนของตัวอักษรและใช้ Chain Code ที่ประกอบด้วย 8 ทิศทาง โครงข่ายประสาทเทียมที่ใช้ในการทำ Type Classification นั้นประกอบไปด้วย Input Node จำนวน 256 โหนด Hidden Node จำนวน 100 โหนด และ Output Node จำนวน 6 โหนด ซึ่งเป็นกลุ่มโครงสร้างของตัวอักษรเกาหลีทั้ง 6 รูปแบบ จากนั้นจะส่งกรอบภาพตัวอักษรในแต่ละกลุ่มเข้าไปยังระบบการรู้จำตัวอักษรโดยอาศัยโครงข่ายประสาทเทียม ทั้งนี้ขนาดของ Input Vector ทั้ง 6 กลุ่มนั้นจะแตกต่างกันไปดังนี้ 314, 120, 76, 54, 520, 301 โหนดตามลำดับและ Output ที่ได้จากการรู้จำมีจำนวน 55 Neurons ในแต่ละกลุ่มตามรูปที่ 2.14

งานวิจัยนี้มุ่งเน้นที่จะพัฒนาและประยุกต์วิธีการรู้จำตัวอักษรบนเครื่อง PC ไปใช้ในการรู้จำบนอุปกรณ์สื่อสาร มือถือ โดยภาพรับเข้ามาจากกล้องถ่ายภาพที่มีอยู่บนตัวอุปกรณ์เหล่านั้น แต่เนื่องด้วยหน่วยความจำและขีดความสามารถในการคำนวณของอุปกรณ์เหล่านั้นมีขีดจำกัด อาทิ เช่นในการคำนวณ Sigmoid Function นั้นเป็น Exponential Function ซึ่งอุปกรณ์มือถือไม่สามารถที่จะคำนวณได้ จึงต้องทำการพัฒนาให้อยู่ในรูปแบบ Embedded OCR System โดยการแปลง Real Number Operation ไปเป็น Integer Operation พิจารณาที่การลดอัตราการรู้จำที่ไม่ถูกต้อง และมีการเปลี่ยน 64 Bits Decimal Data เป็น 32 Bits Integral Data นอกจากนี้ยังต้องมีการทำ Approximation of Sigmoid Function

จากการทดลองนี้มีการใช้ระบบการรู้จำตัวอักษรบน WIPI (Wireless Internet Platform for Interoperability) ภายใต้การพัฒนาและได้รับความร่วมมือจาก SK Telecom Korean Company กลุ่มข้อมูลที่ใช้ในการทดลองมาจาก ETRI (Electronics and Telecommunications Research Institute) Database โครงข่ายประสาทเทียม MLP (Multi-layer perceptron) ประกอบด้วย 256 Input Neurons, 100 Hidden Neurons และ 6 Output Neurons ในการแบ่งกลุ่มโครงสร้างออกเป็น 6 กลุ่ม Training Set ประกอบไปด้วย 11,260 ตัวอักษร Testing Set ประกอบไปด้วย 6,756 ตัวอักษร ค่าความละเอียดของกล้องดิจิทัลเท่ากับ 1,280 x 1,024 Pixels จากผลการทดลองในการแบ่งกลุ่มตามรูปแบบ 6 รูปแบบนั้นมีค่าความถูกต้องโดยเฉลี่ย 99.19% และค่าเฉลี่ยของการรู้จำตัวอักษรของทั้ง 6 รูปแบบเท่ากับ 96.82%





รูปที่ 2.14 กระบวนการรู้จำโดยอาศัยโครงข่ายประสาทเทียมสำหรับระบบการรู้จำตัวอักษรด้วยกล้องถ่ายรูปที่นำไปประยุกต์ใช้งานบน Mobile Device

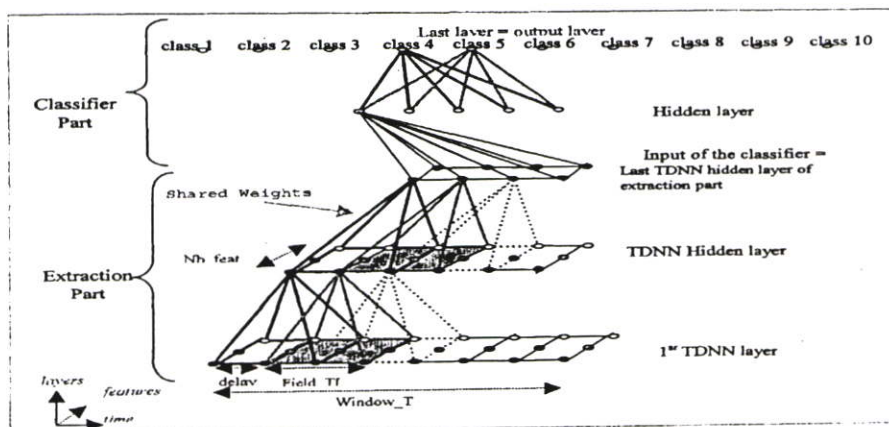
สำหรับโครงสร้างการรู้จำแบบ CNN นั้นมีพื้นฐานสถาปัตยกรรมมาจาก Multi-Layer Perceptron with Back-propagation Learning ที่เป็น Fully-Connection โดยที่โครงสร้างของ Input นั้นสามารถเป็นรูปแบบอะไรก็ได้ โดยจะไม่มีผลกระทบกับผลลัพธ์ที่ได้ออกมาของการ Training ส่วน CNN ในชั้นของ Hidden Neuron จะเชื่อมต่อกับ Subset of Neuron ที่มาจาก Preceding Layer ซึ่งในงานวิจัยนี้นำเอารูปแบบ 2 อย่างของ CNN คือ TDNN มาใช้กับข้อมูลแบบออนไลน์และ SDNN มาใช้จัดการข้อมูลแบบออฟไลน์ จากรูปที่ 2.15 เป็น TDNN Architecture เป็น Neural Network with Temporal Shift จะมีการพิจารณาสิ่งต่างๆ เช่น Size of the Receptive Fields, Number of Layers, Constraint on the Weight Sharing และ Learning Algorithm ซึ่งจากรูปประกอบ

ไปด้วย 2 ส่วนหลักคือส่วนล่าง Lower Layer เรียกว่า Extraction Part โดยจะแปลงลำดับของ Feature Vectors ให้เป็นลำดับของ Higher Order Feature Vectors ส่วนที่สองคือ ส่วนบน เรียกว่า Classifier Part เป็นโครงสร้าง MLP ที่รับเอา Input ที่เป็น Output ออกมาจาก Extraction Part

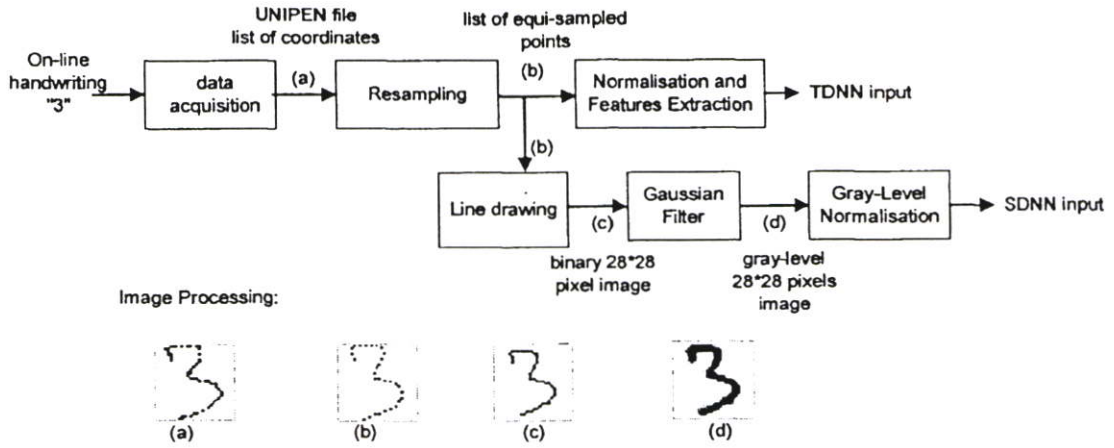
ข้อมูลที่ใช้ในการทดลองแบบออนไลน์มาจากฐานข้อมูลของ Unipen และ IRONOFF โดยที่ข้อมูลของ Input ได้มีการกำหนดรับเข้ามา 50 จุดคงที่สำหรับแต่ละลายมือเขียนที่รับมาจากความเร็วของการเคลื่อนที่ของปากกา จากนั้นผ่านเข้าสู่ Preprocessing ตามรูปที่ 2.16 เริ่มจากรับเอาลายมือเขียนแบบออนไลน์เข้ามา จากนั้นทำ Resampling จำนวน 50 จุดสำหรับแต่ละ Input แล้วแบ่งข้อมูลเป็น 2 ชุดเพื่อส่งต่อเข้าไป 2 วิธีต่อไปนี้ได้แก่วิธีที่ 1 ทำการ Normalized Features จากแต่ละจุดจะได้ค่า Position 2 ค่า, Direction 2 ค่า, Curvature 1 ค่า, Pen Status 1 ค่า แล้วส่งต่อเข้าไปให้เป็น Input ของ TDNN วิธีที่ 2 คือส่งเข้าสู่การทำ Line Drawing ได้รูปภาพแบบไบนารีขนาด 28x28 Pixels เข้าสู่ Gaussian Filter จะได้รูปภาพแบบ Gray-Level ขนาด 28x28 Pixels ออกมา จากนั้นจะทำ Gray-Level Normalization เพื่อส่งต่อเป็น SDNN Input ในขณะเดียวกันก่อนที่จะส่งข้อมูล Input ต่อเข้าไปยัง TDNN ก็มีการจัดแบ่งออกเป็น 3 ประเภทคือ Digits, Lowercase และ Uppercase ซึ่งจะช่วยให้ประสิทธิภาพในการรู้จำดีขึ้น 16% สำหรับ SDNN Input นั้นจะมีค่าระหว่าง -1 ถึง 1

ฐานข้อมูลที่ใช้เป็นข้อมูลในการทดลองการรู้จำแบบออฟไลน์สำหรับ SDNN คือฐานข้อมูลเดียวกันกับที่ใช้ใน TDNN แต่อยู่ในรูปแบบ Offline Image ผลการทดลองที่ได้ TDNN มีอัตราการเรียนรู้จำสูงกว่า MLP โดยเฉลี่ย 1-0.8% ในขณะเดียวกันก็มีการลดความซ้ำซ้อนของจำนวน Weights และขนาดของระบบนั้นเป็น Low Storage Capacities

สำหรับผลการทดลองที่ได้ SDNN มีอัตราการเรียนรู้จำสูงกว่า MLP โดยเฉลี่ย 1-3% ในการศึกษาครั้งนี้ได้รวมวิธี TDNN เข้ากับ SDNN ตามรูปแบบสถาปัตยกรรมรูปที่ 2.17 โดยที่ข้อมูล Training TDNN จะแยกจากการ Training SDNN และจากการทดลองผลจากการรู้จำของ TDNN/SDNN เพิ่มขึ้นจาก 97.9% เป็น 98.2%



รูปที่ 2.15 สถาปัตยกรรม TDNN



รูปที่ 2.16 ขั้นตอนก่อนการรู้จำสำหรับใช้ในการรู้จำลายมือเขียนแบบออนไลน์ที่เป็นภาษาลาดิน

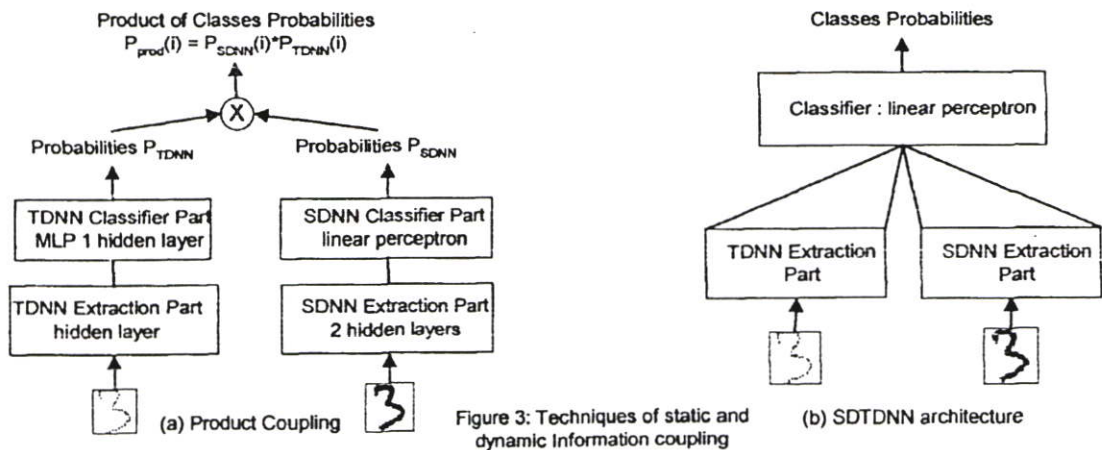


Figure 3: Techniques of static and dynamic information coupling

รูปที่ 2.17 วิธีการของ Static และ Dynamic Information Coupling

### 2.3 ทฤษฎีโครงข่ายประสาทเทียมที่ใช้ในการรู้จำ

#### 2.3.1 Fuzzy ARTMAP

เริ่มแรกของการรู้จำของ Neural Network มีการใช้อัลกอริทึมแบบ Back Propagation ซึ่งพบว่ามีปัญหาที่พบในเรื่องของระยะเวลาในการเรียนรู้และทดสอบนั้นใช้เวลานานรวมทั้งขีดความสามารถในการเรียนรู้ Pattern Input กล่าวคือในการ Trained จะมีการกำหนดจำนวน Neurons และค่าของ Weight ไว้ไม่มีการเปลี่ยนแปลง จึงไม่สามารถที่จะเรียนรู้ Pattern ใหม่ที่ผ่านเข้ามาได้ และหากให้มีการเรียนรู้ Pattern รูปแบบใหม่ก็จะลืมการรู้จำ Pattern แบบเก่าไป ปัญหาตรงนี้เรียกว่า “Plasticity/stability dilemma”

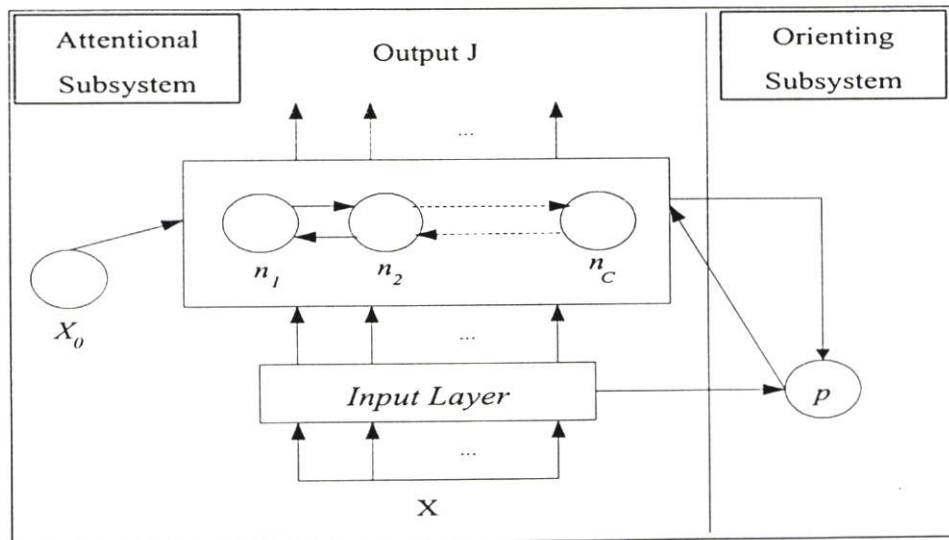
ในระยะเวลาต่อมาได้มีการคิดค้น ART เป็น Neural Network ชนิดหนึ่งที่ออกแบบโดย Grossberg ในปี ค.ศ. 1976 เพื่อนำมาใช้แก้ปัญหา “Plasticity/stability dilemma” เวอร์ชันแรกของ

ART คือ ART-1 คิดค้นโดย Carpenter และ Grossberg ในปี ค.ศ. 1987 ใช้งานในการจัดแบ่งข้อมูลไบนารี จากนั้นไม่นานก็มี ART-2 เกิดขึ้นมาโดยมีการขยายขีดความสามารถคือจัดการกับข้อมูลอนาล็อกได้ด้วย โดยทั้ง ART-1 และ ART-2 มีการรู้จำข้อมูลแบบไม่มีการเรียนรู้ (unsupervised clustering data) ต่อมามีการพัฒนามาเป็น ARTMAP ซึ่งจัดว่าเป็นอัลกอริทึมที่มีการเรียนรู้ใช้งานกับข้อมูลไบนารี หลังจากนั้นก็มี Fuzzy ARTMAP ที่สามารถพัฒนาให้ใช้งานกับข้อมูลอนาล็อกได้ โดยที่ทั้ง ARTMAP และ Fuzzy ARTMAP สามารถใช้วิธีในการรู้จำแบบมีการเรียนรู้ (Supervised Learning Classification)

ปัจจุบันนักวิจัยต่าง ๆ ได้ทำการพัฒนาเพิ่มประสิทธิภาพของรูปแบบการรู้จำของ Neural Network อาทิเช่น Adaptive Hamming Net (AHN), Gaussian ART (GA), Simplified Fuzzy ARTMAP (SFAM) และ Simplified ART (SFART)

### 2.3.1.1 ART-1

จัดว่าเป็นเวอร์ชันแรกของ ART-based Network ที่ได้รับการคิดค้นโดย Carpenter และ Grossberg โดยมุ่งไปที่การรู้จำข้อมูลไบนารีแบบไม่มีการเรียนรู้ (unsupervised clustering of binary data) โดยแบ่งออกเป็นสองส่วนหลักคือส่วนแรกเรียกว่า Attentional subsystem จะประกอบด้วย 1 Layer และส่วนที่สองเรียกว่า Orienting Subsystem ซึ่งใช้ในการตรวจสอบพิจารณาจากค่า Match Function เปรียบเทียบกับค่า Vigilance เพื่อหา Node ที่ชนะ โดยมีสถาปัตยกรรมของ ART ดังรูปที่ 2.18



รูปที่ 2.18 สถาปัตยกรรมของ ART

กำหนดให้ ค่า  $T(X, W_j)$  คือ *Choice Function*

$$T(X, W_j) = \frac{\|X \cap W_j\|}{\alpha + \|W_j\|} \quad (2.3)$$

และค่า  $M(X, W_j)$  คือ *Match Function*

$$M(X, W_j) = \frac{\|X \cap W_j\|}{\|X\|} \quad (2.4)$$

สมการในการเปรียบเทียบหา Match Criterion กับค่า Vigilance  $\rho$

$$M(X, W_j) \geq \rho \quad (2.5)$$

จากสมการที่ 2.5 ถ้า Match Function มีค่ามากกว่าหรือเท่ากับค่า Vigilance แสดงว่าโหนดนั้นเป็น Winner Node จะมีการยอมให้ Input จัดเข้ากลุ่ม Node นั้น แต่หาก Match Function มีค่าน้อยกว่าค่า Vigilance จะไม่มีการจัดให้ Input เข้ากลุ่ม Node นั้น

$$a \cap b = (a_1 \text{ AND } b_1, a_2 \text{ AND } b_2, \dots, a_c \text{ AND } b_c) \quad (2.6)$$

$$\|a\| = \sum_{i=1}^D a_i \quad (2.7)$$

### 2.3.1.2 Fuzzy ART

ต่อมา Carpenter และ Grossberg ได้พัฒนาให้ ART-1 สามารถทำงานกับชุดข้อมูลอินพุตที่เป็นข้อมูลอะนาล็อกได้ด้วย โดยอาศัยทฤษฎีของ Fuzzy Set และมีการใช้ Complement Coding กับชุดข้อมูลอินพุต กำหนดให้

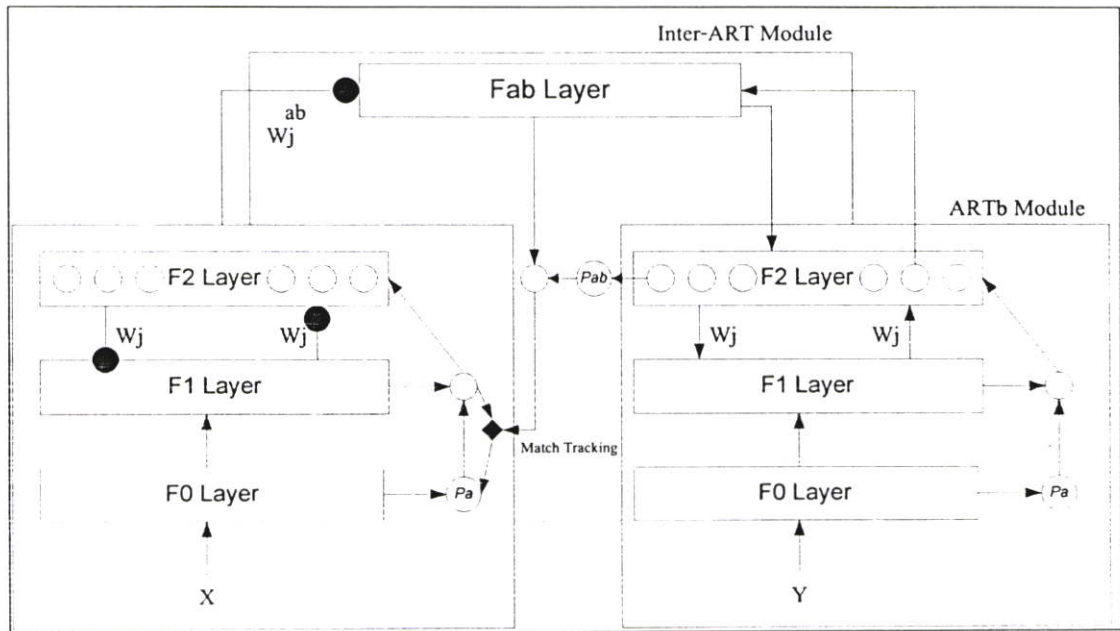
$$\text{Fuzzy AND operation } \wedge : a \wedge b = (\min(a_1, b_1), \min(a_2, b_2), \dots, \min(a_D, b_D)) \quad (2.8)$$

$$\|a\| \text{ คือ } \|a\| = \sum_{i=1}^D a_i \quad (2.9)$$

$$X = (X, X^C) = (x_1, x_1, \dots, x_D, 1 - x_1, 1 - x_2, \dots, 1 - x_D) \quad (2.10)$$

### 2.3.1.3 ARTMAP

เนื่องจาก ART และ Fuzzy ART เป็นวิธีการรู้จำแบบไม่มีการเรียนรู้ (Unsupervised Clustering Methods) จึงทำให้มีการพัฒนา ARTMAP ขึ้นมาเพื่อใช้สำหรับวิธีการรู้จำแบบมีการเรียนรู้ (Supervised Learning Classification) สถาปัตยกรรมของ ARTMAP ประกอบไปด้วย 2 ART คือ ART<sub>a</sub> และ ART<sub>b</sub> หลักการทำงานคือ Pattern Input จะถูกส่งเข้าไปยัง ART<sub>a</sub> และ Label Input จะถูกส่งไปยัง ART<sub>b</sub> ดังรูปที่ 2.19



รูปที่ 2.19 สถาปัตยกรรมของ ARTMAP, Fuzzy ARTMAP

### 2.3.1.4 Fuzzy ARTMAP

พัฒนาต่อเนื่องมาจาก ARTMAP เพื่อให้สามารถรับค่าข้อมูลอินพุตที่เป็นอนาล็อกเข้าไปได้ โดยที่ค่าที่ใช้ในการพิจารณาจะคล้ายกับแนววิธีพิจารณาเหมือน Fuzzy ART และมีโครงสร้างสถาปัตยกรรมเหมือนกับ ARTMAP ดังรูปที่ 2.19

หลักการทำงานของ Fuzzy ARTMAP มีขั้นตอนดังต่อไปนี้

#### ขั้นตอนที่ 1

Input Vector: ในแต่ละ Input Vector จะมีขนาด n-dimensional vector และมีการรวมเอา Complement Coding เข้าไปด้วย

$$x = (A, A^c) = \{A_1 A_2 A_3 \dots A_n \ 1-A_1 \ 1-A_2 \ 1-A_3 \dots 1-A_n\} \quad (2.11)$$

$$|x| = \left| (A, A^C) \right| = \sum_{i=1}^n A_i + \left( n - \sum_{i=1}^n A_i \right) = n \quad (2.12)$$

กำหนดให้  $x$  แทน Input Vector ของกระบวนการรู้จำ ART<sub>s</sub>  
 $A = A_1 \dots A_n$  แทน ค่า Input Pattern  
 $A^C = 1 - A_1 \ 1 - A_2 \ 1 - A_3 \dots 1 - A_n$  แทน ค่า Complement Coding ของค่า Input Pattern

∴ ขนาดของ 1 Vector Input สำหรับ 1 Pattern (ภาพตัวอักษร) = 2n

### ขั้นตอนที่ 2

Weight Vector: ในแต่ละ Category จะมีการกำหนดค่า Weight Vector

$$W_1 = W_2 = W_3 = \dots = W_n = 1 \quad (2.13)$$

กำหนดให้  $W_1 \ W_2 \ W_3 \dots W_n$  แทน Weight Vector ของชั้น F<sub>12</sub> Layer

### ขั้นตอนที่ 3

Parameters:

- A Choice Parameter ( $\alpha$ ) จะมีค่าเข้าใกล้ 0 ∴ กำหนดให้  $\alpha = 0$
- A Learning Parameter ( $\beta$ ) จะมีค่าได้ 3 ค่าตามความเร็วของการเรียนรู้ดังนี้
  - $\beta = 1$  เรียกว่า Fast Learning ส่งผลให้ค่าของ Weight Vector มีการลดลงเปลี่ยนแปลงไปอย่างรวดเร็วขึ้นอยู่กับ Input และ Weight Vector
  - $0 < \beta < 1$  แสดงว่ามีการเรียนรู้แบบค่อยเป็นค่อยไป ส่งผลให้ค่าของ Weight Vector มีการลดลงอย่างเรื่อย ๆ เป็นสัดส่วนกับค่า  $\beta$  ที่เลือกใช้
  - $\beta = 0$  แสดงว่าไม่มีการเรียนรู้เลย ส่งผลให้ค่าของ Weight Vector ไม่มีการเปลี่ยนแปลงตลอดการเรียนรู้

สำหรับการเลือก Neuron ที่ชนะของ ART ได้มาจากการคำนวณค่า Choice Function เพื่อวัดค่าความคล้ายคลึงกันระหว่าง Input และ Weight Vector หาได้จากสมการต่อไปนี้

$$T_j(x) = \frac{|x \wedge W|}{|W|} \quad (2.14)$$

$$Winner = T_j = \max \{T_j(x)\} \quad (2.15)$$

กำหนดให้	$T_j(x)$	แทนค่าความคล้ายคลึงกันระหว่าง Input (x) และ Weight (W) จะมีค่าอยู่ตั้งแต่ 1 ถึง 0
	$ W $	คือค่า Norm ของ Weight Vector หาได้จาก $ W  = \sum_{i=1}^n W_i$
	$T_j$	คือค่าที่ใช้ในการเลือก Neuron ที่เหมาะสมและชนะของ ART ซึ่งถ้ามีค่ามากกว่า 1 Neuron ที่ชนะจะเลือก โหนด Neuron ที่มีค่า j น้อยที่สุด
	x	แทน Input Vector ของกระบวนการรู้จำ ART <sub>a</sub>
	j	คือ Neuron แต่ละตัวที่ Layer นั้น
	n	แทนจำนวนของ Element ใน Weight Vector

#### ขั้นตอนที่ 4

- A Vigilance Parameter ( $\rho$ ) คือค่าความคล้ายคลึงกันระหว่าง Input และ Weight Vector ว่าจะมีความคล้ายคลึงกันมากพอที่จะจัดให้ Input นั้นไปรวมอยู่ในกลุ่ม Category นั้นหรือไม่ โดยจะกำหนดค่า = 1 เมื่อเหมาะสมที่มีความคล้ายคลึงกันมากที่สุดที่จะจัดไว้ในกลุ่มเดียวกัน มีการตรวจสอบตามสมการต่อไปนี้

$$\frac{|x \wedge W_j|}{|x|} \geq \rho \quad (2.16)$$

จากนั้นตรวจสอบ Vigilance Criterion ของส่วน Map Field ตามสมการดังต่อไปนี้

$$\frac{|y_{ART_b} \wedge W_{ART_b}^j|}{|y_{ART_b}|} \geq \rho_{ab} \quad (2.17)$$

และถ้าเป็นสมการเป็นเท็จ จะทำการปรับค่า Vigilance ให้  $\rho > \frac{|x \wedge W_j|}{|x|}$

#### ขั้นตอนที่ 5

ในขณะที่มีการเรียนรู้จะคำนวณหาค่าจาก Category Choice Function ระหว่าง Input กับทุก ๆ Weight ของ Neurons จากนั้นจะทำการนำค่าที่ได้ที่มีค่ามากที่สุดมาทำการเปรียบเทียบกับค่า Vigilance Parameter ( $\rho$ ) ซึ่งจะเกิดเหตุการณ์ที่เป็นไปได้ 2 กรณีคือ

### 1. Resonance

*“If match function of the chosen node meets (equal or more than) the vigilance criterion”*

$$f(x,W) \geq \rho \quad (2.18)$$

หากพบว่า มี Neuron ที่มีค่าที่ได้ออกมาจาก Category Choice Function มีค่ามากกว่าค่า Vigilance อยู่มากกว่า 1 Neuron จะทำการเลือก Neuron ลำดับแรกสุดที่ชนะ โดยทำการตรวจสอบ Vigilance Criterion

### 2. Reset

*“If match function of the chosen node meets (less than) the vigilance criterion”*

$$f(x,W) < \rho \quad (2.19)$$

จากนั้นจะมีการสร้าง Neuron ขึ้นมาใหม่สำหรับ Input Node ใหม่ที่รับเข้าสู่กระบวนการ รู้จำและกำหนดค่า Weight ขึ้นมาเริ่มต้นตามขั้นตอนที่ 2

#### ขั้นตอนที่ 6

Learning: ส่วนของการเรียนรู้นี้จะเกิดขึ้นเมื่อขั้นตอนที่ 4 มีการเกิดกรณี Resonance หลังจากนั้นจะทำการปรับค่าของ Weight Vector สำหรับ Neuron ที่ชนะ (Winner) และยอมให้จัดเอา Input ที่เข้ามารวมไว้ในกลุ่มเดียวกันได้ ดังสมการต่อไปนี้

$$W^{new} = \beta(x \wedge W^{old}) + (1 - \beta)W^{old} \quad (2.20)$$

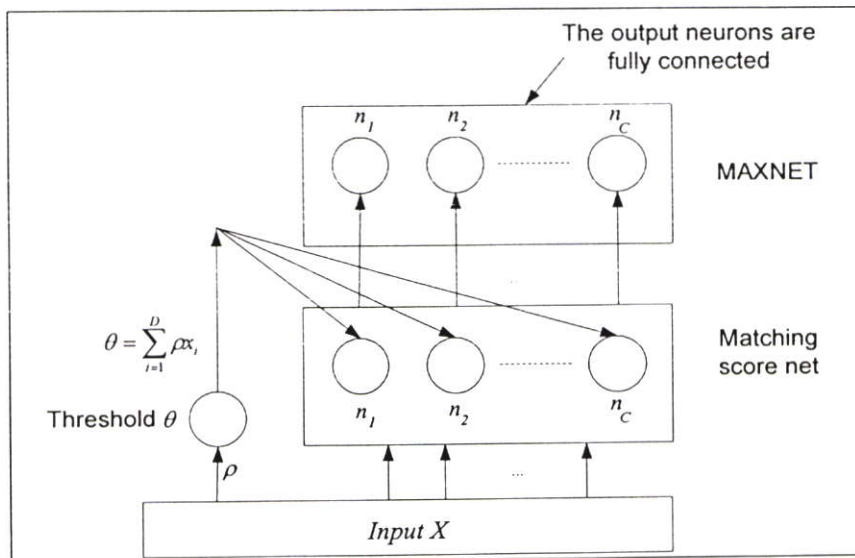
#### 2.3.1.5 Adaptive Hamming Net (AHN)

พัฒนามาจาก ART Models ออกแบบโดย Cheng-An Hung และ Sheng-Fuu สถาปัตยกรรมของ AHN แสดงดังรูปที่ 2.20 ประกอบไปด้วย 2 เลเยอร์ จำนวนของ Neurons ของ Hidden Layer จะเท่ากับจำนวน Neurons ของ Output Layer ซึ่ง ทุก Input จะเชื่อมต่อเข้ากับทุก Neurons ใน Hidden Layer แต่สำหรับในแต่ละ Neuron ที่  $j$  ใน Hidden Layer จะเชื่อมต่อไปที่ Neuron ที่  $j$  ใน Output Layer เท่านั้น ซึ่งใน Hidden Layer จะพิจารณาค่า Matching Score Net ว่า Input Pattern มีความเหมือนพอที่จะจัดเข้ากลุ่ม Neuron นั้น ๆ ได้หรือไม่ โดยที่อาจจะมี Neurons ที่คล้ายกับ Input เกิด Activated มากกว่า 1 Neuron ซึ่งหลังจากนั้นใน Output Layer จะใช้การหา

MAXNET ในการคัดเลือกว่า Neuron ไหนจะชนะซึ่งจะเป็น Neuron ที่มีค่า Matching Score Net มากที่สุด

### 2.3.1.6 Gaussian ARTMAP

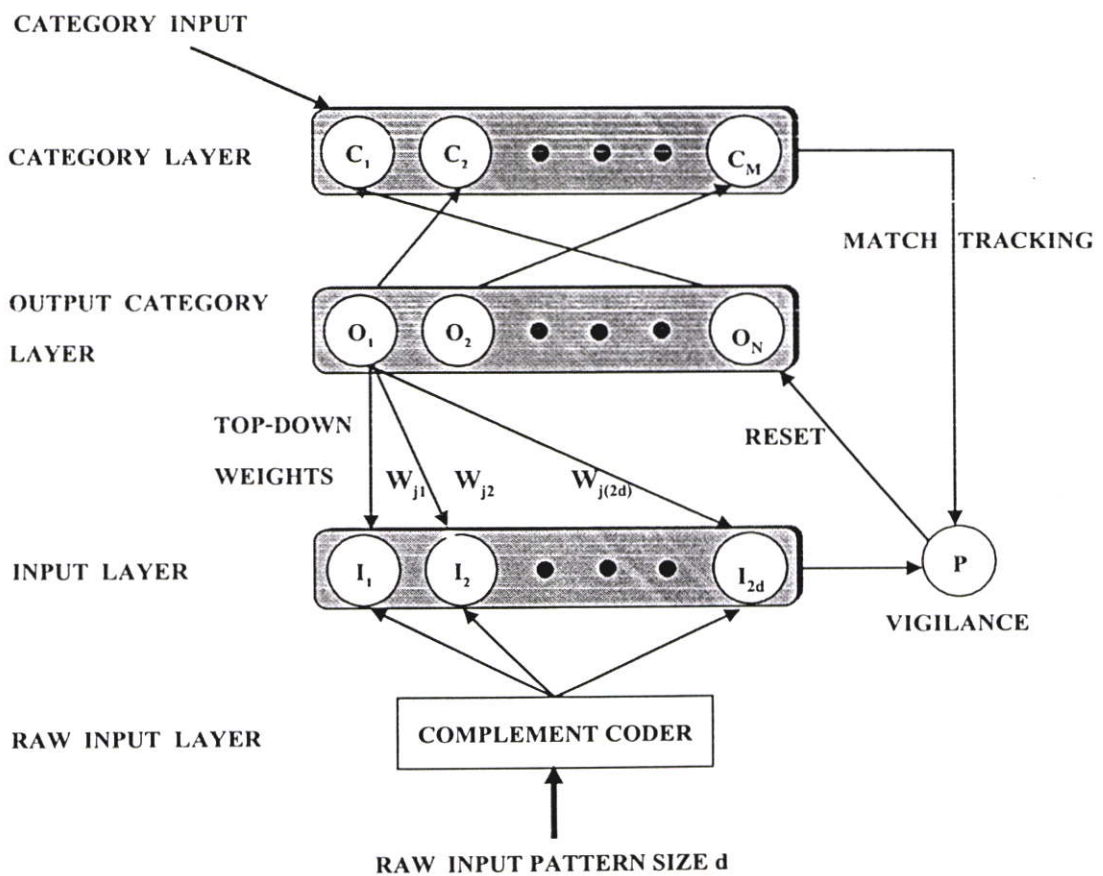
Williamson ได้คิดวิธีในการแก้ปัญหาที่พบใน Fuzzy ARTMAP 2 อย่างคือ ข้อมูลสัญญาณรบกวนที่เกิดขึ้นในการรู้จำ และการแบ่งกลุ่มนำเสนอแต่ละประเภทอย่างไม่มีประสิทธิภาพ ด้วยปัญหาสองอย่างนี้ จึงนำเอาการกระจายแบบ Gaussian เข้ามาใช้ในพิจารณา Choice Function, Match Function, และ Classification Function ข้อดีคือการเพิ่มค่าของ Choice Function และ Matching Function จะมีค่าที่สามารถแบ่งกลุ่มให้รูปแบบของ Input เข้าจัดกลุ่มได้ตรงตามแต่ละประเภทได้อย่างเหมาะสม โดยไม่มีผลมาจากสัญญาณรบกวนที่อาจจะเกิดขึ้นและทำให้แบ่งกลุ่มผิด นอกจากนี้การกระจายแบบ Gaussian เป็นรูปแบบของการกระจายที่มีการนำมาใช้งานส่วนใหญ่ แต่สำหรับขอบเขตของ GA ก็คือการใช้งานจะมีค่าการกระจายแบบ Gaussian อย่างอิสระดังนั้นจึงจำเป็นต้องมี Diagonal Ellipsoids ครอบคลุมช่วงของข้อมูลอินพุต ข้อดีที่เห็นได้ชัดว่ามีประสิทธิภาพมากกว่า Fuzzy ARTMAP ก็คือการรู้จำแบ่งกลุ่มได้มีจำนวน Nodes ที่รู้จำได้ถูกต้องมากขึ้นตรงตามค่า Output ที่ต้องการและทำการกำหนดไว้



รูปที่ 2.20 สถาปัตยกรรมของ AHN

### 2.3.2 Simplified Fuzzy ARTMAP (SFAM)

เป็น ART ชนิดหนึ่งที่มีการปรับวิธีในการรู้จำให้ง่ายขึ้นจาก Fuzzy ARTMAP โดยจะมีการลดส่วนของ ART<sub>b</sub> ออกไป คงเหลือไว้เพียงแต่ ART<sub>a</sub> ที่มีลักษณะการทำงานคล้ายกับใน Fuzzy ARTMAP ในการเข้าใจการทำงานโครงสร้างภายในเป็นไปตามพีชชีลอจิก (Fuzzy Logic) สถาปัตยกรรมของ SFAM ดังรูปที่ 2.21 โครงสร้างของ SFAM จะคล้ายกับการเชื่อมต่อ Weight, Input Layer และ Output Layer นอกจากนี้จะมี Category Layer ที่มีการรับเอา Category Input เข้ามาไว้ ลักษณะพิเศษนี้ทำให้ SFAM สามารถแก้ปัญหากรณีที่ข้อมูลอินพุตที่เข้ามาเป็น Nonlinear Decision Region คือการจัดประเภทกลุ่มชุดข้อมูลโดยไม่มีรูปแบบที่แน่นอน ทั้งนี้จะมีการทำ Complement Coding ให้กับ Input Vector ก่อนที่จะเข้าสู่ SFAM เนื่องจากเป็นการทำการ Normalize และเพิ่มขยายข้อมูล Input ให้มีขนาดเป็นสองเท่าของชุดข้อมูลเดิม เพื่อช่วยให้ SFAM ได้ทำการตัดสินใจได้ดีขึ้น



รูปที่ 2.21 สถาปัตยกรรมของ Simplified Fuzzy ARTMAP (SFAM)

## 2.4 ทฤษฎีเบื้องต้นของงานวิจัยนี้

### 2.4.1 กรอสคอรีเลชัน (Cross-Correlation)

ในการจัดแบ่งกลุ่มของรูปแบบต่าง ๆ ออกเป็นกลุ่มเดียวกัน จะอาศัยการวัดความคล้ายคลึงกันระหว่างรูปแบบของชุดข้อมูลต่าง ๆ เหล่านั้น กรอสคอรีเลชันคือวิธีมาตรฐานอีกวิธีหนึ่งที่ใช้ในการวัดและประเมินความคล้ายคลึงกันว่าชุดข้อมูลต่าง ๆ มีความสัมพันธ์กันมากน้อยเพียงใด ซึ่งถ้ากล่าวถึงค่าคอรีเลชันก็คือค่ากรอสคอรีเลชันที่มีค่า Delay = 0 โดยค่าความสัมพันธ์ที่ได้เมื่อทำการ Normalize ค่ากรอสคอรีเลชันแล้ว จะได้ค่าความสัมพันธ์อยู่ระหว่าง 0 ถึง 1 พิจารณาจากสมการต่อไปนี้

$$r(x, y) = \sum_i x_i y_i \quad (2.21)$$

$$r_p(x, y) = \frac{\sum_i x_i y_{i-p}}{\sqrt{\sum_i x_i^2 \sum_i y_i^2}} \quad (2.22)$$

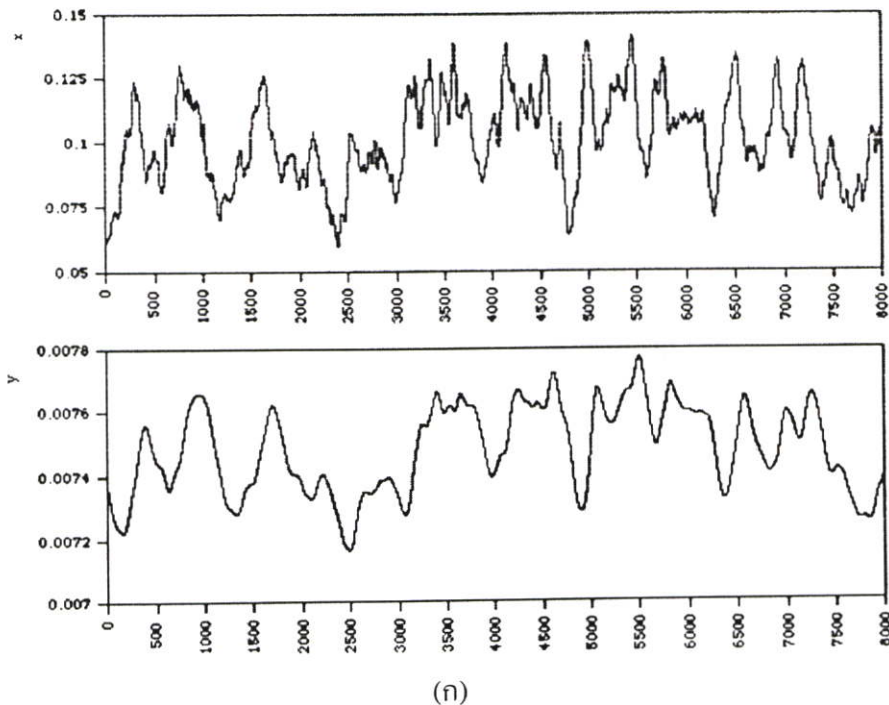
$$R(x, y) = \max_p (r_p(x, y)) \quad (2.23)$$

กำหนดให้	$r(x, y)$	แทนค่า Correlation (at elay $p = 0$ ) without normalized
	$r_p(x, y)$	แทนค่า Normalized Cross-correlation at delay = $p$
	$R(x, y)$	แทนค่า Normalized Cross-correlation function ระหว่างสองชุดข้อมูล $x, y$
	$x_i$	แทนชุดข้อมูลรูปแบบ $x_i$
	$y_i$	แทนชุดข้อมูลรูปแบบ $y_i$
	$i = 0, 1, 2, 3, \dots, N-1$	
	$N$	คือจำนวนของ element ที่อยู่ใน Pattern ชุดข้อมูล $x_i$ และ $y_i$
	$p$	คือค่า delay ที่มีการเลื่อนในแต่ละตำแหน่งของการหาค่าความสัมพันธ์ออกมา โดย $p = 0, 1, 2, 3, \dots, N-1$
	max	คือการคำนวณหาค่ามากที่สุดที่ได้

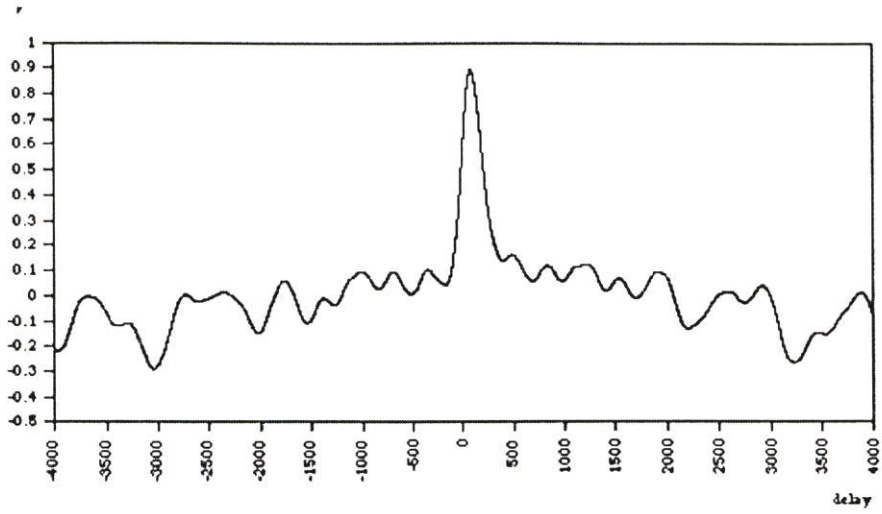
ในการประยุกต์ใช้งานในการจัดกลุ่มของวัตถุที่เป็นสัญญาณคลื่นอนาล็อกหรือเรียกว่ารูปคลื่นแบบต่อเนื่องทางเวลา (Continuous-time Waveform) ดังรูปที่ 2.22 หรืออาจจะเป็นชุดข้อมูลแบบลำดับไม่ต่อเนื่อง (Discrete-time Sequence) หรือชุดข้อมูลแบบเวกเตอร์ดังรูปที่ 2.23

สำหรับการพิจารณารูปคลื่นแบบต่อเนื่องทางเวลาจะคำนวณตามสมการ ซึ่งเป็นการหาค่าความคล้ายคลึงกันของคลื่นตามช่วงเวลา หรือเรียกว่าเป็นการหาค่าครอสคอร์รีเลชันที่ค่า Delay ต่าง ๆ และสำหรับการหาค่าคอร์รีเลชันก็คือการหาค่าครอสคอร์รีเลชันที่ค่า delay = 0 จากรูปที่ 2.10 มีการทำ Normalized โดยค่าครอสคอร์รีเลชันจะมีค่าอยู่ระหว่าง -1, 1 โดยที่ถ้า  $r = 1$  คือมีความคล้ายคลึงกันมาก แต่ถ้า  $r = 0$  คือไม่มีความคล้ายคลึงกัน และถ้า  $r = -1$  คือมีความแตกต่างกันระหว่างชุดข้อมูลทั้งสอง

จากรูปที่ 2.23 ยังไม่มีการทำ Normalized ดังนั้นค่าที่ได้ก็จะมีความสูงไปด้วย ตามขนาดของเวกเตอร์ และพบว่าสมาชิกของเวกเตอร์ที่มีค่าเป็น 0 นั้นจะเข้ามารบกวน ส่งผลกระทบต่อความคล้ายคลึงกัน แต่การวัดค่าด้วยครอสคอร์รีเลชันจะได้ผลออกมาเป็น 1 แสดงว่าชุดข้อมูลสองเวกเตอร์มีความเหมือนกัน แต่หากพิจารณาค่าคอร์รีเลชันจะพบว่ามีความเป็น  $3/5 = 0.6$  เมื่อทำการ Normalized ซึ่งจะเห็นว่าการพิจารณาเฉพาะค่าคอร์รีเลชันจะได้ค่าความคล้ายกันออกมา ณ ค่าของ delay = 0 ซึ่งจะเห็นว่าการใช้การคำนวณค่าครอสคอร์รีเลชันจะได้ผลลัพธ์ของการแสดงค่าความเหมือนหรือคล้ายคลึงกันได้ออกมาอย่างถูกต้องและเหมาะสม

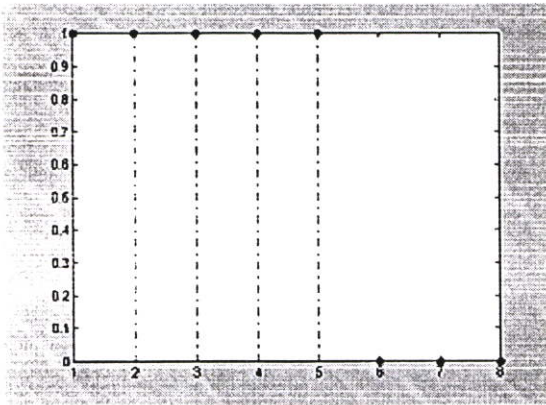


รูปที่ 2.22 (ก) รูปคลื่น  $x, y$  แบบต่อเนื่องทางเวลา (ข) ค่าครอสคอร์รีเลชันที่ได้ประมาณ 0.9 แสดงว่ารูปคลื่น  $x, y$  มีความสัมพันธ์คล้ายคลึงกันมากที่ delay = 40 จากทั้งหมด 4000 delay

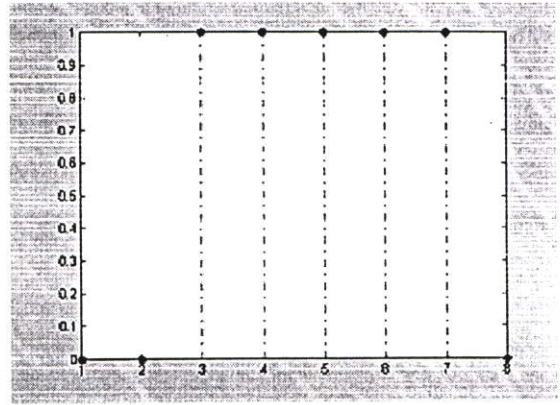


(จ)

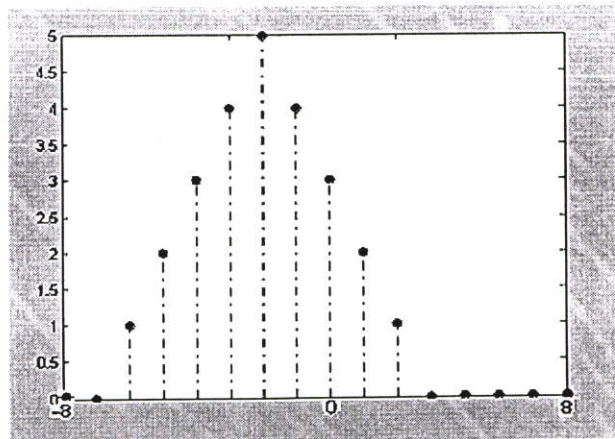
รูปที่ 2.22 (ต่อ)



(x)



(y)



(r)

รูปที่ 2.23 (x) ชุดข้อมูล x และ (y) ชุดข้อมูล y (r) ค่าคออสคอร์รีเลชัน (เมื่อทำการ normalized) ที่ได้ประมาณ 1 ( $r=1$ ) แสดงว่าชุดข้อมูล x, y มีความสัมพันธ์เหมือนกันที่ delay = -2

### บทที่ 3

## การรู้จำรูปแบบแนวทางใหม่โดยใช้โครงสกอรีรีเลชัน

### 3.1 กระบวนการก่อนการรู้จำ

กระบวนการก่อนการรู้จำนับว่าเป็นกระบวนการแรกในระบบการรู้จำที่ทำการประมวลผลเบื้องต้นเกี่ยวกับภาพตัวอักษรเพื่อให้ได้ข้อมูลเป็นอินพุตส่งต่อเข้าสู่กระบวนการรู้จำ โดยกระบวนการก่อนการรู้จำนี้จะประกอบไปด้วย 5 ขั้นตอนดังต่อไปนี้

#### ขั้นตอนที่ 1

เริ่มต้นจะมีการสแกนภาพเอกสารตัวอักษรเข้ามาเป็นข้อมูลอินพุตของระบบการรู้จำ ซึ่งจะ เป็นภาพเอกสารที่ประกอบไปด้วยบรรทัดหลายบรรทัด ในขั้นตอนนี้จะทำการแยกบรรทัดและ แยกตัวอักษรแต่ละตัวออกจากบรรทัด โดยใช้วิธีการของฮิสโตแกรม (Histogram) และการหา ขอบภาพ (Contour Algorithm) ตามรูปที่ 3.1 ซึ่งในขั้นตอนนี้จะได้ตำแหน่งและขนาดของภาพ ตัวอักษรแต่ละตัวออกมาเรียกว่า “กรอบตัวอักษร”

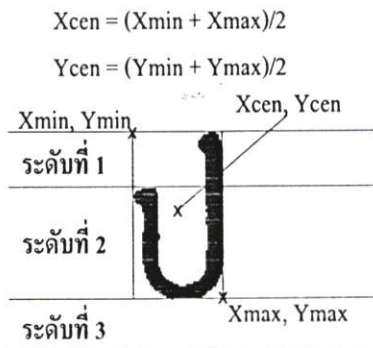


รูปที่ 3.1 การแยกบรรทัดและตัวอักษรโดยใช้ฮิสโตแกรม และการหาขอบภาพ

จากนั้นจะทำการจัดกลุ่มของกรอบตัวอักษรที่ได้เป็น 3 กลุ่มคือกลุ่มตัวอักษรที่มีจุด ศูนย์กลางของกรอบตัวอักษรอยู่ในระดับที่ 1, 2, และ 3 ตามลำดับยกตัวอย่างเช่นรูปที่ 3.2 เป็นการ แบ่งกลุ่มระดับโดยการหาจุดศูนย์กลางของกรอบภาพตัวอักษร “ป” และจัดอยู่ในกลุ่มระดับที่ 2 สำหรับกลุ่มตัวอักษรระดับที่ 1, 2, และ 3 นี้จะเรียกว่าเป็นกลุ่มระดับบน กลาง และล่างตามลำดับ ตามตารางที่ 3.1 พบว่าพยัญชนะไทยและตัวเลขอารบิกนั้นอยู่ในระดับกลางเหมือนกันทั้งหมด ส่วน วรรณยุกต์ทั้งหมดก็อยู่ในระดับบน แต่สำหรับสระไทยนั้นมีปรากฏอยู่ที่ 3 ระดับ

ขั้นตอนที่ 2

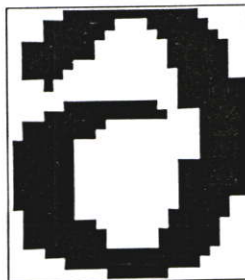
จากนั้นจะนำเอากรอบภาพตัวอักษรที่ได้มาดังรูปที่ 3.3 มาทำการปรับขนาดให้เท่ากับ 25x29 Pixels ด้วยวิธีการ Resize โดยอาศัย Interpolation/resampling filter แบบ Hermite Filter แล้วทำการลดขนาดของเส้นตัวอักษรจากการหาจุดกึ่งกลางของวงกลมที่มีจุดปลายเส้นผ่านศูนย์กลางทั้งสองจุดอยู่บนจุดปลายของขอบตัวอักษรตามรูปที่ 3.4 ทั้งนี้เพื่อปรับภาพตัวอักษรให้อยู่ในรูปแบบของภาพเส้นของตัวอักษรดังรูปที่ 3.5 โดยจะไม่มีมีการเปลี่ยนแปลงโครงสร้างพิเศษต่าง ๆ ของภาพตัวอักษรด้วยวิธีการ Skeletonization โดยใช้ Hilditch's Algorithm



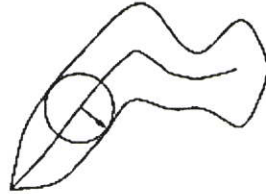
รูปที่ 3.2 การแบ่งกลุ่มระดับโดยการหาจุดศูนย์กลางของกรอบภาพตัวอักษร

ตารางที่ 3.1 กลุ่มระดับตัวอักษรภาษาไทยที่แบ่งออกเป็น 3 กลุ่ม

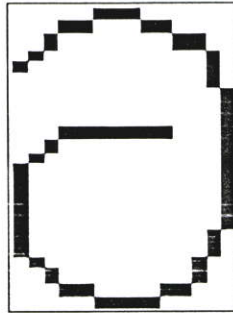
กลุ่มระดับบน 12 ตัว	ค ส ๕ ๕ ๖ ๖ ๗ + ๘ ๙ ๐
กลุ่มระดับกลาง 66 ตัว	ก ข ช ด ต ข ง จ ฉ ช ฌ ญ ฎ ฏ ฐ ท ฒ ณ ด ต ก ท ร บ บ ป ฝ ฝ พ ฝ ม ย ร ล ว ต ษ ส ห ฬ อ ฮ ก ก ๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ ะ ำ เ แ โ ใ ไ ๗ ๘
กลุ่มระดับล่าง 2 ตัว	จ ๖



รูปที่ 3.3 กรอบภาพตัวอักษร “อ”



รูปที่ 3.4 การหาจุดกึ่งกลางของวงกลมที่มีจุดปลายเส้นผ่านศูนย์กลางทั้งสองจุดอยู่บนจุดปลายของขอบตัวอักษรสำหรับการทำ Skeleton



รูปที่ 3.5 รูปแบบของภาพลายเส้นตัวอักษรหลังจากการทำ Skeleton

ประโยชน์ของการทำ Skeleton ตัวอักษรเพื่อจุดประสงค์ของการได้มาซึ่งตำแหน่งของพิกเซลที่ต่อเนื่องกันในรูปแบบของตัวอักษรหนึ่ง ๆ ซึ่งยังคงความเป็นรูปแบบลายเส้นของตัวอักษรหลังจากที่มีการทำ Skeleton และยังสามารถที่จะส่งเข้าสู่กระบวนการรู้จำได้ หรืออาจจะกล่าวได้ว่ายังคงสามารถรู้จำออกมาได้ว่าเป็นรูปแบบของตัวอักษรอะไร อัลกอริทึมที่นำมาใช้ก็คือ Hilditch's Algorithm เริ่มจากการพิจารณา 8-neighborhood ของพิกเซล P1 ตามรูปที่ 3.6 จากนั้นจะเลือกที่จะพิจารณาให้ P1 นั้นเป็นพิกเซลที่ผ่านการ Skeleton หรือไม่ หรือว่าจะลบพิกเซลนั้นทิ้งไปถ้าพิกเซลนั้นไม่นับว่าเป็น Skeleton Pixel จะมีการกำหนดฟังก์ชัน 2 ฟังก์ชันที่ใช้ในการพิจารณาคือ

1.  $B(P1)$  เป็นฟังก์ชันที่ใช้นับจำนวนของ non-zero neighbors ของพิกเซล P1 คือ  $B(P1) = P2 + P3 + P4 + P5 + P6 + P7 + P8 + P9$
2.  $A(P1)$  เป็นฟังก์ชันที่ใช้นับจำนวนของการเปลี่ยนแปลงค่า 0-1 Patterns ในลำดับของ P2, P3, P4, P5, P6, P7, P8, P9, P2

Hilditch's Algorithm นั้นจะมีการตรวจสอบเงื่อนไขในการนำมาพิจารณาและหากพบว่า เป็นไปตามเงื่อนไขก็จะสามารถทำการลบพิกเซลที่ตำแหน่ง P1 นั้นทิ้งไปได้ แต่หากไม่ผ่านเงื่อนไขข้อใดข้อหนึ่งหรือมากกว่าหนึ่งเงื่อนไขแสดงว่าผลลัพธ์เงื่อนไขไม่เป็นจริง ก็จะกำหนดให้พิกเซลที่ตำแหน่ง P1 เป็น Skeleton Pixel และพิจารณาต่อไปเรื่อย ๆ จนกระทั่งไม่มีพิกเซลที่สามารถจะลบทิ้งออกไปได้อีกแล้ว ยกตัวอย่างผลลัพธ์ที่ได้ออกมาหลังจากที่ผ่านการทำ Skeleton ดังรูปที่ 3.5 ข้างต้น เงื่อนไขดังกล่าวมีทั้งหมด 4 ข้อ ได้แก่

1. เงื่อนไขข้อที่ 1:  $2 \leq B(P1) \leq 6$  จะเป็นการทดสอบจุดปลายและจุดบริเวณขอบเขตของรูปแบบตัวอักษร ซึ่งจากรูปที่ 3.7 ไม่เป็นไปตามเงื่อนไขข้อที่ 1 แสดงให้เห็นว่าถ้า  $B(P1) = 1$  แล้วจุดพิกเซล P1 จะเป็นจุดปลายของการทำ Skeleton และหาก  $B(P1) = 0$  แสดงว่าจุดพิกเซล P1 นั้นเป็นจุดแยกออกมาจุดหนึ่งซึ่งไม่ควรที่จะลบทิ้ง และหาก  $B(P1) = 7$  แสดงว่าจุดพิกเซล P1 นั้นเป็นจุดที่เส้นขอบของตัวอักษรที่ไม่ควรที่จะลบทิ้งด้วยเช่นกัน

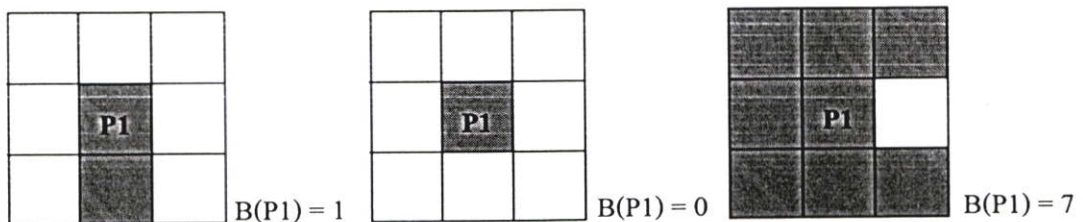
2. เงื่อนไขข้อที่ 2:  $A(P1) = 1$  จะเป็นการทดสอบความต่อเนื่องของพิกเซลในรูปแบบตัวอักษร ซึ่งจากรูปที่ 3.8 ไม่เป็นไปตามเงื่อนไขข้อที่ 2 แสดงให้เห็นว่าถ้า  $A(P1) > 1$  แล้วหากทำการลบจุดพิกเซล P1 ทิ้งไป จะทำให้รูปแบบของเส้นตัวอักษรไม่ต่อเนื่องกัน

3. เงื่อนไขข้อที่ 3:  $P2 \times P4 \times P8 = 0$  หรือ  $A(P2) \neq 1$  จะเป็นการทดสอบเพื่อให้แน่ใจว่า Pattern ที่กว้าง 2 พิกเซลติดกันในแนวตั้งจะไม่ถูกตัดทิ้งหายไป ซึ่งจากรูปที่ 3.9 ไม่เป็นไปตามเงื่อนไขข้อที่ 3

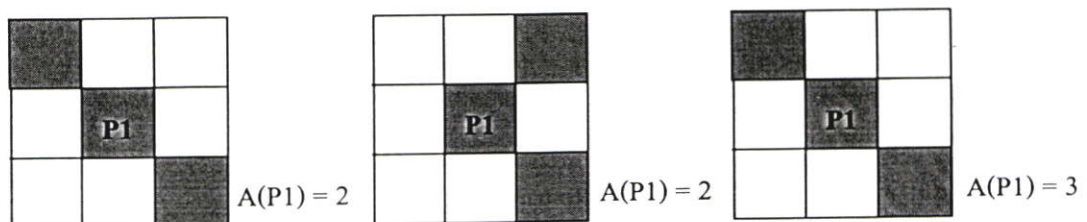
4. เงื่อนไขข้อที่ 4:  $P2 \times P4 \times P6 = 0$  หรือ  $A(P4) \neq 1$  จะเป็นการทดสอบเพื่อให้แน่ใจว่า Pattern ที่กว้าง 2 พิกเซลติดกันในแนวนอนจะไม่ถูกตัดทิ้งหายไป ซึ่งจากรูปที่ 3.10 ไม่เป็นไปตามเงื่อนไขข้อที่ 4

P9	P2	P3
P8	P1	P4
P7	P6	P5

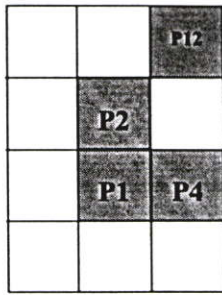
รูปที่ 3.6 ภาพ 8-neighborhood ของพิกเซล P1



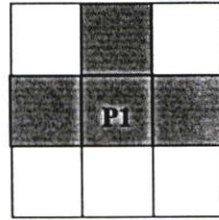
รูปที่ 3.7 แสดงรูปภาพที่จะทำให้เงื่อนไขข้อที่ 1 เป็นเท็จ



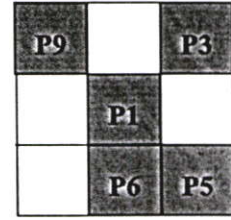
รูปที่ 3.8 แสดงรูปภาพที่จะทำให้เงื่อนไขข้อที่ 2 เป็นเท็จ



$$A(P2) \neq 1$$

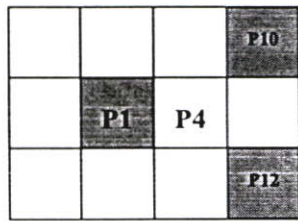


$$P2 \times P4 \times P8 \neq 0 \text{ และ } A(P2) = 1$$

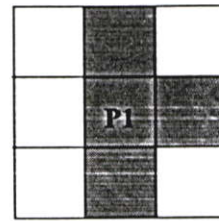


$$P2 \times P4 \times P8 = 0$$

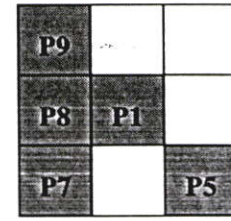
รูปที่ 3.9 แสดงรูปภาพที่จะทำให้เงื่อนไขข้อที่ 3 เป็นเท็จ



$$A(P4) \neq 1$$



$$P2 \times P4 \times P6 \neq 0 \text{ และ } A(P4) = 1$$



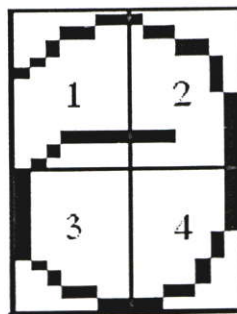
$$P2 \times P4 \times P6 = 0$$

รูปที่ 3.10 แสดงรูปภาพที่จะทำให้เงื่อนไขข้อที่ 4 เป็นเท็จ

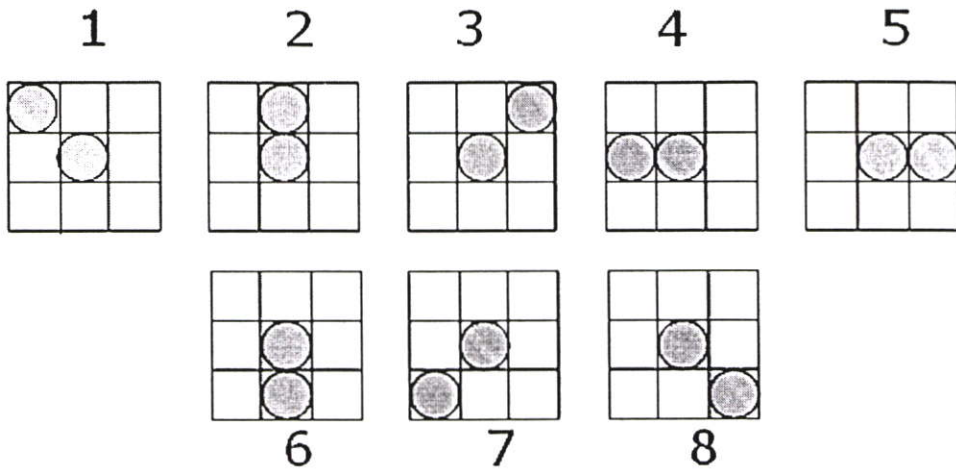
### ขั้นตอนที่ 3

ทำการหาจุด Endpoint ของแต่ละกรอบภาพตัวอักษรและกำหนดจุดเริ่มต้นและจุดสิ้นสุดของเส้นภาพตัวอักษรซึ่งประกอบไปด้วย 4 ขั้นตอนดังต่อไปนี้

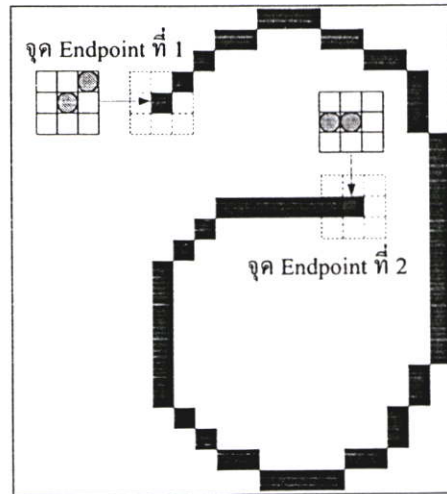
ขั้นตอนที่ 3.1 แบ่งกรอบภาพตัวอักษรออกเป็น 4 Quadrants แล้วกำหนดลำดับของกรอบภาพตัวอักษรที่ถูกแบ่งจากซ้ายไปขวา และบนลงล่างคือ 1, 2, 3 และ 4 ตามรูปที่ 3.11



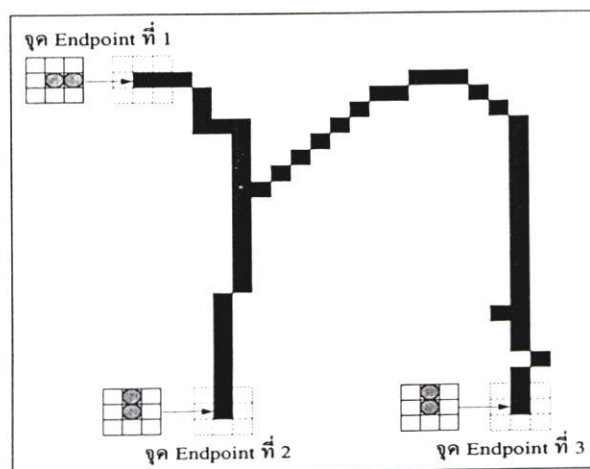
รูปที่ 3.11 การแบ่งกรอบภาพตัวอักษรออกเป็น 4 Quadrants



รูปที่ 3.12 เมตริกขนาด 3x3 Pixels ที่ใช้ในการหาจุด Endpoint มีทั้งหมด 8 รูปแบบ



รูปที่ 3.13 กรอบภาพตัวอักษรที่มีจุด Endpoint 2 จุด

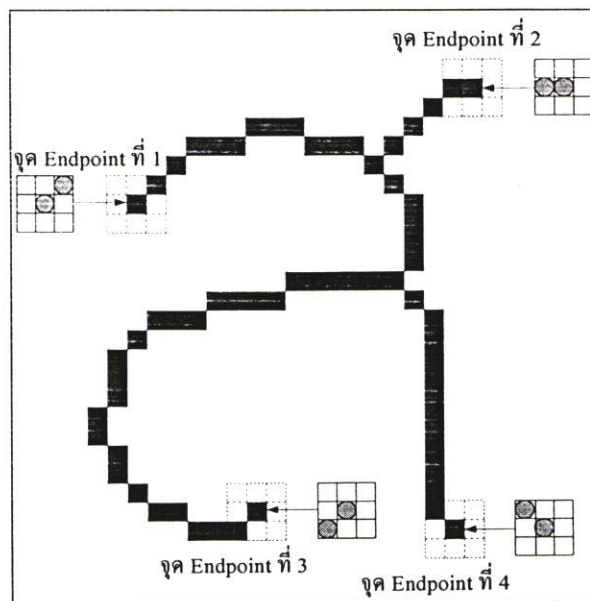


รูปที่ 3.14 กรอบภาพตัวอักษรที่มีจุด Endpoint 3 จุด

ขั้นตอนที่ 3.2 กำหนดเมตริกขนาด 3x3 Pixels สำหรับใช้ในการหาจุด Endpoint ดังรูปที่ 3.12 ซึ่งจะมีอยู่ทั้งหมด 8 รูปแบบ

ขั้นตอนที่ 3.3 นำเอารูปแบบของเส้นภาพตัวอักษรในกรอบภาพที่ผ่านการทำ Skeleton แล้ว มาเปรียบเทียบกับจุดศูนย์กลางของรูปแบบเมตริก 3x3 ที่มีอยู่ทั้งหมด 8 รูปแบบที่ใช้ในการหาจุด Endpoint และเมื่อได้ Pattern ที่ตรงกันแล้วจะกำหนดให้ตำแหน่งจุดที่อยู่บนเส้นภาพตัวอักษรนั้น เป็นจุด Endpoint ของลายเส้นตัวอักษร ยกตัวอย่างเช่นรูปที่ 3.13 แสดงกรอบภาพตัวอักษรที่มีจุด Endpoint 2 จุด หรือรูปที่ 3.14 แสดงกรอบภาพตัวอักษรที่มีจุด Endpoint 3 จุด หรือรูปที่ 3.15 แสดงกรอบภาพตัวอักษรที่มีจุด Endpoint 4 จุด

ขั้นตอนที่ 3.4 จากนั้นจะทำการกำหนดจุดเริ่มต้นและจุดสิ้นสุดตามลำดับความสำคัญ โดยเริ่มจากจุด Endpoint ที่อยู่ที่จุดศูนย์กลางของกรอบภาพตัวอักษร จากนั้นตามด้วยจุด Endpoint ที่อยู่ใน Quadrant ที่ 1, 2, 3 และ 4 ตามลำดับดังตัวอย่างรูปที่ 3.16 แสดงจุดเริ่มต้นและจุดสิ้นสุดของตัวอักษร “อ” รูปที่ 3.17 แสดงจุดเริ่มต้นและจุดสิ้นสุดของตัวอักษร “ท” และรูปที่ 3.18 แสดงจุดเริ่มต้นและจุดสิ้นสุดของตัวอักษร “ส”

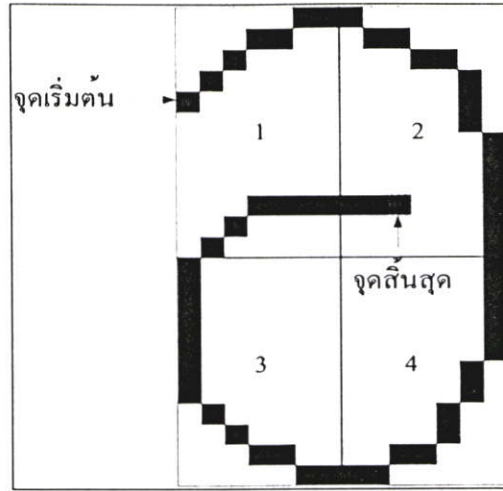


รูปที่ 3.15 กรอบภาพตัวอักษรที่มีจุด Endpoint 4 จุด

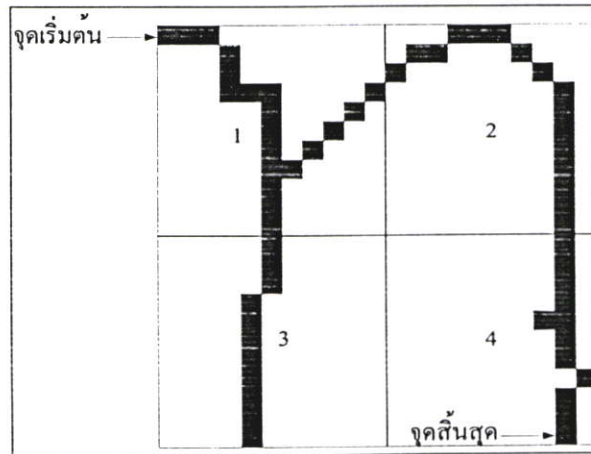
#### ขั้นตอนที่ 4

ทำการกำหนดรหัสแทนเส้นตัวอักษรซึ่งประกอบไปด้วย 4 ขั้นตอนดังต่อไปนี้

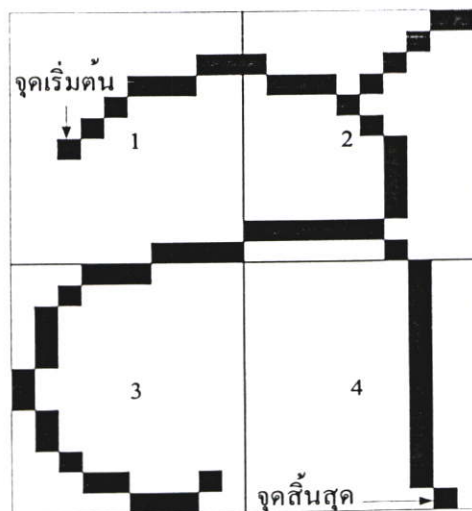
ขั้นตอนที่ 4.1 กำหนดรหัสแทนเส้นตัวอักษรทั้งหมด 9 รหัสตามรูปที่ 3.19 และรหัสแทนเส้นตัวอักษรหรือเรียกว่า “Directional Code” นี้จะแสดงในสมการที่ 3.1 โดยจะพิจารณามุมระหว่างเส้นตัวอักษรกับเส้น Base line  $\theta = 0$



รูปที่ 3.16 แสดงจุดเริ่มต้นและจุดสิ้นสุดของตัวอักษร “อ”



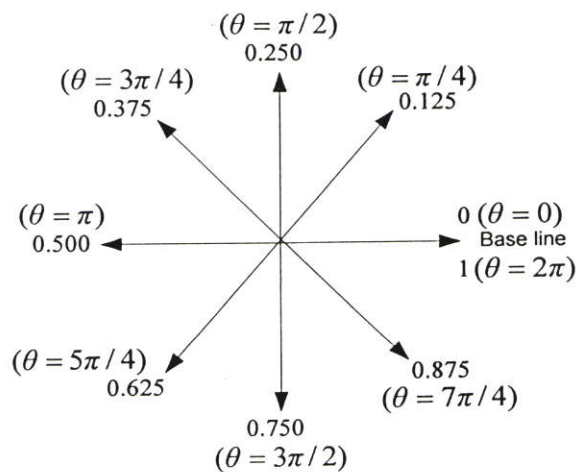
รูปที่ 3.17 แสดงจุดเริ่มต้นและจุดสิ้นสุดของตัวอักษร “ท”



รูปที่ 3.18 แสดงจุดเริ่มต้นและจุดสิ้นสุดของตัวอักษร “ส”

ขั้นตอนที่ 4.2 แบ่งภาพกรอบตัวอักษรที่มีการกำหนดจุดเริ่มต้นและจุดสิ้นสุดของเส้นตัวอักษรออกเป็นสามส่วนในแนวตั้งและสามส่วนในแนวนอนก็จะได้กรอบย่อยที่ถูกแบ่งออกเป็น 9 กรอบย่อย จากนั้นทำการกำหนดให้กรอบย่อยที่มีจุดเริ่มต้นของเส้นตัวอักษรนั้นเป็นกรอบย่อยที่ 1 และกำหนดกรอบย่อยต่อไปที่มีเส้นตัวอักษรเชื่อมต่อในแนวทิศทางของตัวอักษรให้เป็นกรอบย่อยที่ 2, 3, 4, 5, 6, 7, 8 และ 9 ตามลำดับ ยกตัวอย่างตามรูปที่ 3.20 แสดงทิศทางของเส้นตัวอักษร “อ” ตามการจัดเรียงกรอบย่อยของภาพตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร นอกจากนี้หากกรอบตัวอักษรนั้นมีจุดเริ่มต้นและจุดสิ้นสุดแล้วยังมีจุด Endpoint อื่น ๆ อยู่อีกด้วย จะทำการกำหนดลำดับกรอบย่อยเริ่มจากจุดเริ่มต้นไปยังจุดสิ้นสุด จากนั้นจะพิจารณาจุด Endpoint ต่อไปให้เป็นจุดเริ่มต้นในการกำหนดทิศทางตัวอักษรและกำหนดลำดับกรอบย่อยตัวอักษรต่อไปเรื่อย ๆ จนครบ 9 กรอบย่อยยกตัวอย่างตามรูปที่ 3.21 แสดงทิศทางของเส้นตัวอักษร “ท” ตามการจัดเรียงกรอบย่อยของภาพตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร และรูปที่ 3.22 แสดงทิศทางของเส้นตัวอักษร “ส” ตามการจัดเรียงกรอบย่อยของภาพตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร

ขั้นตอนที่ 4.3 ค่าข้อมูลสำหรับในแต่ละกรอบย่อยตัวอักษรที่แบ่งออกเป็นทั้งหมด 9 กรอบย่อย จะแทนด้วยค่ารหัสแทนเส้นตัวอักษร โดยเริ่มจากจุดเริ่มต้นของเส้นทางการเดินไปตามเส้นตัวอักษรที่อ้างอิงทิศทางแต่ละพิคเซลหนึ่งไปยังอีกพิคเซลหนึ่งที่มีการเชื่อมต่อของเส้นตัวอักษรไปเรื่อย ๆ ในกรอบย่อยแต่ละกรอบตามลำดับของกรอบย่อยที่กำหนดไว้ ยกตัวอย่างเช่นการแทนค่ารหัสแทนเส้นทิศทางของตัวอักษร “อ” แสดงตามตารางที่ 3.2 และการแทนค่ารหัสแทนเส้นทิศทางของตัวอักษร “ท” แสดงตามตารางที่ 3.3 และการแทนค่ารหัสแทนเส้นทิศทางของตัวอักษร “ส” แสดงตามตารางที่ 3.4



รูปที่ 3.19 รหัสแทนเส้นตัวอักษรทั้งหมด 9 รหัส

$$\text{Directional Code} = \begin{cases} 0 & \rightarrow \theta = 0 \\ 0.125 & \rightarrow \theta = \pi/4 \\ 0.250 & \rightarrow \theta = \pi/2 \\ 0.375 & \rightarrow \theta = 3\pi/4 \\ 0.5 & \rightarrow \theta = \pi \\ 0.675 & \rightarrow \theta = 5\pi/4 \\ 0.750 & \rightarrow \theta = 3\pi/2 \\ 0.875 & \rightarrow \theta = 7\pi/4 \\ 1 & \rightarrow \theta = 2\pi \end{cases} \quad (3.1)$$

กำหนดให้ Directional Code แทนรหัสแทนเส้นตัวอักษร  
 $\theta$  แทนมุมระหว่างเส้นตัวอักษรกับเส้น Base line  $\theta = 0$

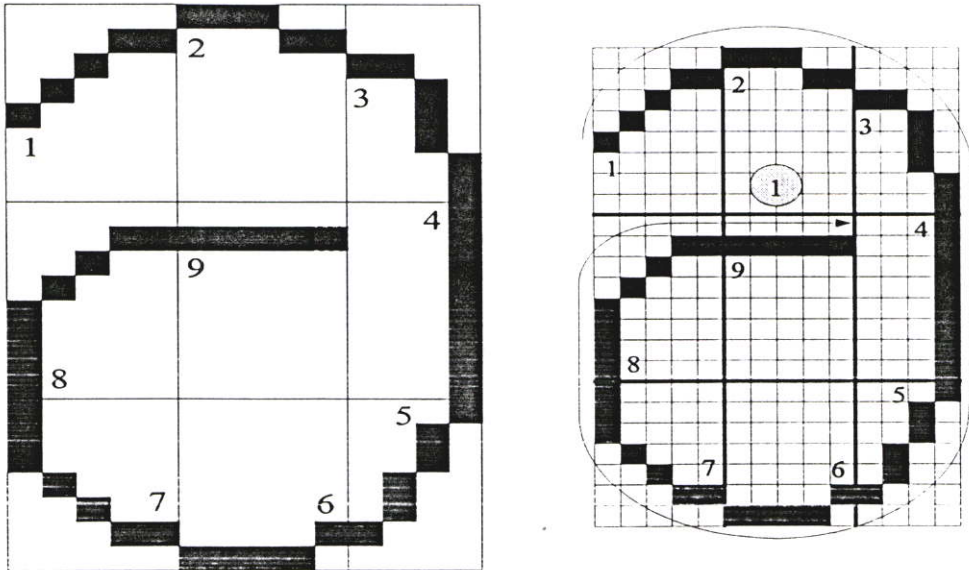
ขั้นตอนที่ 4.4 เป็นขั้นตอนการกำหนดค่าข้อมูลโดยพิจารณาค่ารหัสแทนเส้นตัวอักษรที่จะนำมาใช้คำนวณในฟังก์ชันครอสคอร์รีเลชัน เพื่อแก้ปัญหาที่อาจจะเกิดขึ้นโดยมีสาเหตุมาจากการเปรียบเทียบค่าความคล้ายคลึงกันระหว่างสองค่าชุดข้อมูลนั้นจำเป็นต้องกำหนดให้ชุดข้อมูลทั้งสองมีขนาดความยาวเท่ากัน และค่ารหัสแทนเส้นตัวอักษรที่มีค่าเป็น 0 ในทางทฤษฎีครอสคอร์รีเลชันนั้นจะหมายความว่าค่าข้อมูลที่เป็น 0 นั้นไม่มีความหมาย จึงต้องมีการแปลงค่าข้อมูลที่เป็นค่า 0 ดิค ๆ กัน จากนั้นก็จะได้ค่าข้อมูลที่ได้ออกมาเป็นค่า Input Vector สำหรับส่งต่อเข้าสู่กระบวนการรู้จำที่อาศัยฟังก์ชันครอสคอร์รีเลชัน จึงมีหลักในการพิจารณาดังต่อไปนี้

1. หากค่าความยาวของค่าข้อมูลไม่เท่ากันจะแทนค่าด้วยค่าสุดท้ายของชุดข้อมูลนั้น ๆ โดยที่ขนาดของ Input Vector จะมีขนาดเท่ากับขนาดของชุดข้อมูลย่อยที่มีขนาดยาวที่สุด
2. พิจารณาชุดข้อมูลที่มีค่า 0 ดิคกันมากกว่า 2 ตัวขึ้นไป เช่น [... 0 0 0 ...] , [... 0 0 0 0 ...] และ [... 0 0 0 0 ...] จะทำการแปลงค่า 0 ระหว่างนั้นให้เป็น 1 จะทำให้ได้ Input Vector คือ [... 0 1 0 ...] , [... 0 1 1 0 ...] และ [... 0 1 1 1 0 ...] ตามลำดับเป็นต้น
3. ถ้าค่าข้อมูลมีค่าเป็น [0] หรือ [0 0] จะแทนค่าข้อมูลด้วย [0 1 0]
4. หากไม่มีค่าข้อมูลในบางกรอบส่วนย่อย จะกำหนดให้ Input Vector ที่กรอบย่อยนั้นมีค่าเป็น 0 ทั้งหมด โดยที่ขนาดของ Input Vector จะมีขนาดเท่ากับขนาดของชุดข้อมูลย่อยที่มีขนาดยาวที่สุด (ซึ่งจะไม่มีมีการแปลงค่า 0 ที่ติดกันไปเป็นค่า 1 ตามข้อที่ 2 ข้างต้น)

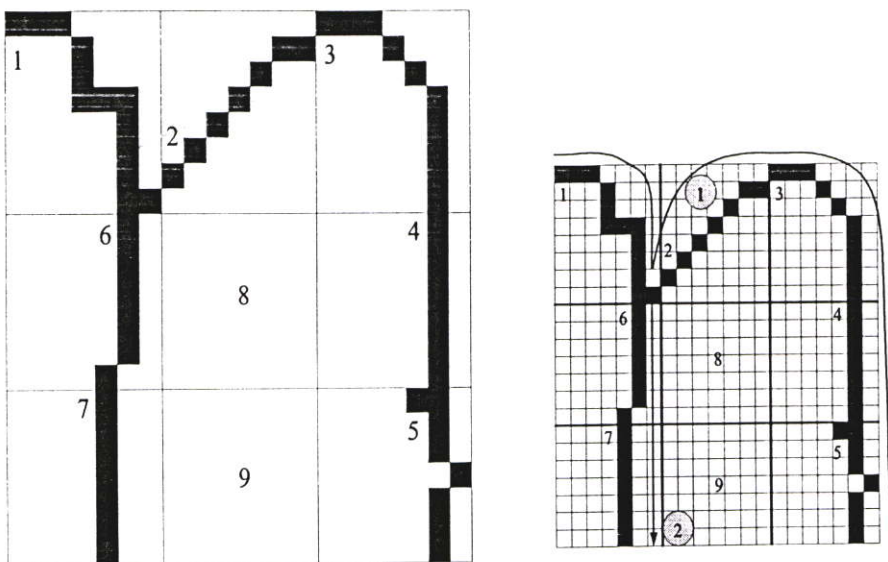
#### ขั้นตอนที่ 5

แนวคิดของวิธีใหม่ที่น่าเสนอเริ่มต้นด้วยการจัดแบ่งแยกแยะข้อมูลภาพตัวอักษรออกเป็น 3 กลุ่มระดับตัวอักษร ได้แก่ กลุ่มระดับตัวอักษรบน กลุ่มระดับตัวอักษรกลาง และกลุ่มระดับตัว

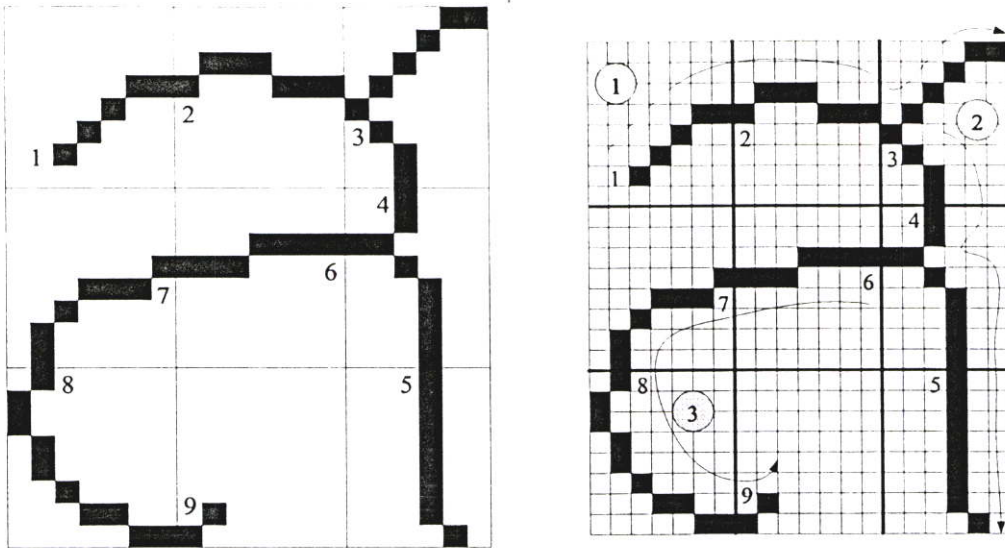
อักษรล่าง โดยแต่ละกลุ่มจะมีประเภทที่กำหนดไว้ตามกลุ่มตัวอักษรระดับบน กลาง และล่าง ตามลำดับแสดงดังรูปที่ 3.23 จากนั้นในแต่ละกลุ่มที่ถูกแบ่งจะส่งรูปแบบตัวอักษรเข้าไปสู่กระบวนการรู้จำแบบโครงข่ายประสาทเทียมตามวิธีการของพีคฟอร์ดเวิร์ดแบบสี่ชั้น (Four Layers Feedforward Neural Network) ตามที่แสดงในรูปที่ 3.24



รูปที่ 3.20 แสดงทิศทางของเส้นตัวอักษร “อ” ตามการจัดเรียงกรอบย่อยของภาพตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร



รูปที่ 3.21 แสดงทิศทางของเส้นตัวอักษร “ท” ตามการจัดเรียงกรอบย่อยของภาพตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร



รูปที่ 3.22 แสดงทิศทางของเส้นตัวอักษร “ส” ตามการจัดเรียงกรอบย่อยของภาพตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร

ตารางที่ 3.2 การแทนค่ารหัสแทนเส้นทิศทางของตัวอักษร “อ”

กรอบย่อยที่	ค่าข้อมูล	Input Vector มีขนาดเท่ากับ 8
1 ( $I^1$ )	0.125 0.125 0.125 0 0.125	0.125 0.125 0.125 0 0.125 0.125 0.125 0.125
2 ( $I^2$ )	0 0 0.875 1 0.875	0 0 0.875 1 0.875 0.875 0.875 0.875
3 ( $I^3$ )	1 0.875 0.75 0.75 0.875 0.75 0.75	1 0.875 0.75 0.75 0.875 0.75 0.75 0.75
4 ( $I^4$ )	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
5 ( $I^5$ )	0.625 0.75 0.625 0.75 0.625 0.5	0.625 0.75 0.625 0.75 0.625 0.5 0.5 0.5
6 ( $I^6$ )	0.625 0.5 0.5 0.5 0.375	0.625 0.5 0.5 0.5 0.375 0.375 0.375 0.375
7 ( $I^7$ )	0.5 0.375 0.375 0.375 0.25 0.25 0.25	0.5 0.375 0.375 0.375 0.25 0.25 0.25 0.25
8 ( $I^8$ )	0.25 0.25 0.25 0.125 0.125 0.125 0 0	0.25 0.25 0.25 0.125 0.125 0.125 0 0
9 ( $I^9$ )	0 0 0 0	0 1 1 1 1 1 1 0

ตารางที่ 3.3 การแทนค่ารหัสแทนเส้นทางของตัวอักษร “ท”

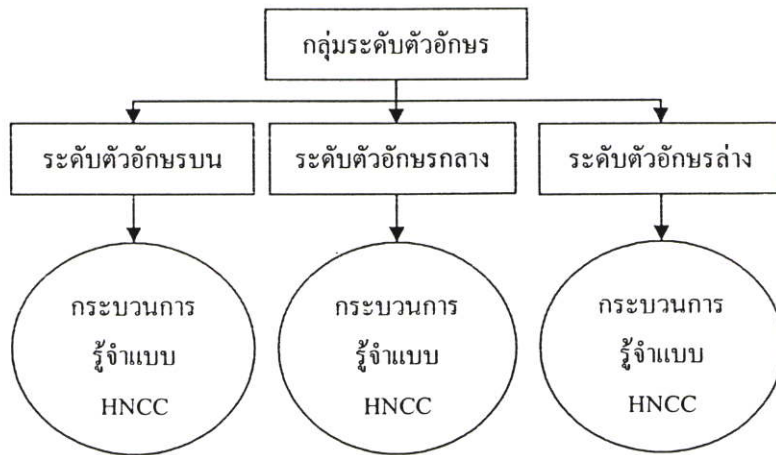
กรอบย่อยที่	ค่าข้อมูล	Input Vector มีขนาดเท่ากับ 13
1 ( $I^1$ )	0 0 0.875 0.75 0.75 1 1 0.75 0.75 0.75 0.75 1 0.125	0 0 0.875 0.75 0.75 1 1 0.75 0.75 0.75 0.75 1 0.125
2 ( $I^2$ )	0.125 0.125 0.125 0.125 0.125 0 0.125	0.125 0.125 0.125 0.125 0.125 0 0.125 0.125 0.125 0.125 0.125 0.125 0.125
3 ( $I^3$ )	0 0 0.875 0.875 0.875 0.75 0.75 0.75 0.75 0.75	0 0 0.875 0.875 0.875 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
4 ( $I^4$ )	0.75 0.75 0.75 0.75 0.75 0.75 0.625 0.625	0.75 0.75 0.75 0.75 0.75 0.75 0.625 0.625 0.625 0.625 0.625 0.625 0.625
5 ( $I^5$ )	0.875 0.75 0.875 0.625 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75	0.875 0.75 0.875 0.625 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
6 ( $I^6$ )	0.75 0.75 0.75 0.75 0.75 0.625 0.75 0.75 0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.625 0.75 0.75 0.75 0.75 0.75 0.75 0.75
7 ( $I^7$ )	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
8 ( $I^8$ )	-	0 0 0 0 0 0 0 0 0 0 0 0 0
9 ( $I^9$ )	-	0 0 0 0 0 0 0 0 0 0 0 0 0

ตารางที่ 3.4 การแทนค่ารหัสแทนเส้นทางของตัวอักษร “ส”

กรอบย่อยที่	ค่าข้อมูล	Input Vector มีขนาดเท่ากับ 10
1 ( $I^1$ )	0.125 0.125 0.125 0 0	0.125 0.125 0.125 0 1 1 1 1 1 0
2 ( $I^2$ )	0.125 0 0 0.875 1 1 0.875 0.875	0.125 0 0 0.875 1 1 0.875 0.875 0.875 0.875
3 ( $I^3$ )	0.125 0.125 0.125 0.125 0 0.875 0.75 0.75	0.125 0.125 0.125 0.125 0 0.875 0.75 0.75 0.75 0.75
4 ( $I^4$ )	0.75 0.625 0.5 0.875 0.75 0.75 0.75 0.75	0.75 0.625 0.5 0.875 0.75 0.75 0.75 0.75 0.75 0.75
5 ( $I^5$ )	0.75 0.75 0.75 0.75 0.75 0.75 0.875 0.875 0.875 0.875	0.75 0.75 0.75 0.75 0.75 0.75 0.875 0.875 0.875 0.875

ตารางที่ 3.4 (ต่อ)

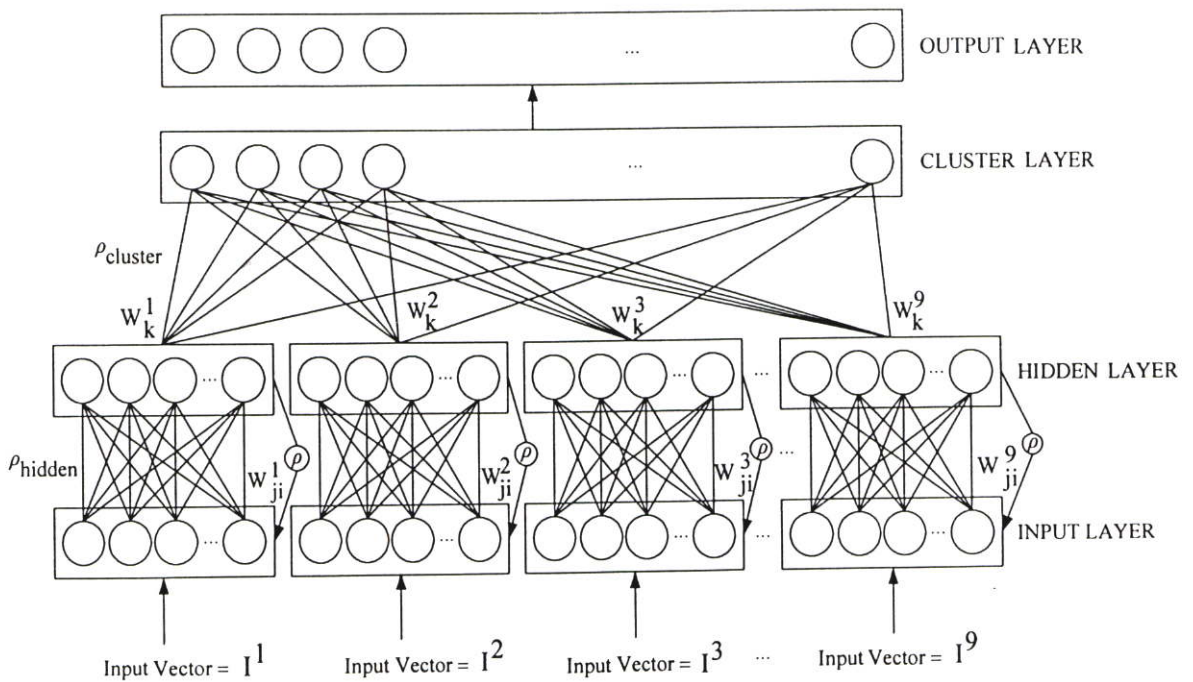
กรอบย่อยที่	ค่าข้อมูล	Input Vector มีขนาดเท่ากับ 10
6 ( $1^6$ )	0.5 0.5 0.5 0.625 0.5 0.5 0.5	0.5 0.5 0.5 0.625 0.5 0.5 0.5 0.5 0.5
7 ( $1^7$ )	0.625 0.5 0.5 0.625 0.625 0.75 0.75	0.625 0.5 0.5 0.625 0.625 0.75 0.75 0.75 0.75 0.75
8 ( $1^8$ )	0.625 0.75 0.875 0.75 0.875 0.875 1 0.875 1 1	0.625 0.75 0.875 0.75 0.875 0.875 1 0.875 1 1
9 ( $1^9$ )	0.125	0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125



รูปที่ 3.23 โครงสร้างของระบบการรู้จำตัวอักษรที่นำเสนอในงานวิจัยนี้โดยอาศัย Hierarchical Normalized Cross-Correlation (HNCC) Neural Network

### 3.2 กระบวนการรู้จำ

ระบบในการรู้จำที่นำเสนอใหม่นี้เป็นแนวทางการรู้จำโดยมีการออกแบบสถาปัตยกรรมของโครงข่ายประสาทเทียมตามวิธีการของฟีดฟอร์เวิร์ดแบบสี่ชั้น (Four Layers Feedforward Neural Network) ตามที่แสดงในรูปที่ 3.24 ชั้นเลเยอร์แรกเรียกว่าชั้นอินพุตประกอบไปด้วย 9 ส่วน และแต่ละส่วนจะรับข้อมูลอินพุตเข้ามาเป็นอินพุตเวกเตอร์ที่มีขนาด  $N$  โหนดเท่า ๆ กันในแต่ละส่วนตามค่าข้อมูลอินพุตเวกเตอร์สำหรับแต่ละกรอบย่อยที่มีอยู่ทั้งหมด 9 กรอบย่อยที่ถูกแบ่งออก และจากนั้นก็ทำการกำหนดด้วยรหัสแทนเส้นตัวอักษรนั้น ๆ ตามขั้นตอนที่ 3 ในกระบวนการก่อนการรู้จำที่กล่าวถึงขั้นต้น ต่อมาเป็นชั้นเลเยอร์ที่สองเรียกว่าชั้นฮิดเดน (Hidden Layer) ทุกโหนดใน Hidden Layer จะเชื่อมต่อกับทุกโหนดเฉพาะในแต่ละส่วนของชั้นอินพุตเดียวกันตามที่แบ่งออกไว้ 9 ส่วน ในการเชื่อมต่อจะอาศัยเวกเตอร์ค่าน้ำหนัก (Weight Vector) ที่เชื่อมต่อผ่านทาง  $w_{ji}^h$  ระหว่างชั้น  $i$  กับชั้น  $j$  หรือ Hidden Layer สำหรับแต่ละส่วนย่อย  $h$  จากนั้นชั้นเลเยอร์ที่สามเรียกว่าชั้นคลัสเตอร์ (Cluster Layer) แต่ละโหนดในชั้นนี้จะเชื่อมต่อกับโหนดใน Hidden Layer ที่สร้างขึ้นมาในช่วงของการเรียนรู้ และจะมีค่า Weight Vector ในชั้นนี้เป็นตัวเชื่อมต่อกับโหนดที่เชื่อมกันคือ  $w_k^h$  สำหรับชั้นเลเยอร์ที่สี่เรียกว่าชั้นเอาต์พุต (Output Layer) แต่ละโหนดในชั้นนี้จะนำเสนอประเภทของตัวอักษรที่เป็น Target Output Vector สำหรับในแต่ละอินพุตเวกเตอร์ในช่วงของการเรียนรู้แบบ Supervised Learning



รูปที่ 3.24 สถาปัตยกรรมของโครงข่ายประสาทเทียมสี่ชั้น

## อัลกอริทึมในการเรียนรู้ประกอบไปด้วย 7 ขั้นตอนดังต่อไปนี้

### ขั้นตอนที่ 1

กำหนดค่า Input vector (I): ในแต่ละ Input vector ได้มีการแบ่งช่วงย่อยของ Input vector ออกเป็น 9 ส่วนย่อยตามทิศทางของเส้นตัวอักษรที่อยู่ในแต่ละกรอบย่อยที่ถูกแบ่งออกเป็น 9 กรอบย่อย (ตามตัวอย่างที่แสดงในตารางที่ 3.2, 3.3 และ 3.4 เป็นต้น)

$$I = \{I^1 \ I^2 \ I^3 \ I^4 \ I^5 \ I^6 \ I^7 \ I^8 \ I^9\} \quad (3.2)$$

$$I^1 = \{I_1^1 \ I_2^1 \ I_3^1 \ \dots \ I_N^1\} \quad (3.3)$$

$$I^2 = \{I_1^2 \ I_2^2 \ I_3^2 \ \dots \ I_N^2\} \quad (3.4)$$

$$I^3 = \{I_1^3 \ I_2^3 \ I_3^3 \ \dots \ I_N^3\} \quad (3.5)$$

$$I^4 = \{I_1^4 \ I_2^4 \ I_3^4 \ \dots \ I_N^4\} \quad (3.6)$$

$$I^5 = \{I_1^5 \ I_2^5 \ I_3^5 \ \dots \ I_N^5\} \quad (3.7)$$

$$I^6 = \{I_1^6 \ I_2^6 \ I_3^6 \ \dots \ I_N^6\} \quad (3.8)$$

$$I^7 = \{I_1^7 \ I_2^7 \ I_3^7 \ \dots \ I_N^7\} \quad (3.9)$$

$$I^8 = \{I_1^8 \ I_2^8 \ I_3^8 \ \dots \ I_N^8\} \quad (3.10)$$

$$I^9 = \{I_1^9 \ I_2^9 \ I_3^9 \ \dots \ I_N^9\} \quad (3.11)$$

กำหนดให้	I	แทน Input vector ของกระบวนการรู้จำ
	$I^1 \ \dots \ I^h \ \dots \ I^9$	แทน Input vector ที่มีการกำหนดในขอบเขตย่อยในแต่ละกรอ ย่อยที่ 1 ถึง 9 ของกระบวนการรู้จำ
	N	คือจำนวน โหนดอินพุตในแต่ละส่วนย่อย (Feature Components)
	h	คือลำดับกรอย่อยที่ส่งค่า Input Vector เข้ามาในระบบการรู้จำ

### ขั้นตอนที่ 2

ครั้งแรกของกระบวนการเรียนรู้ จะกำหนดค่าเริ่มต้นให้กับ Weight Vector ที่ชั้น Hidden Layer และ Cluster Layer โดยที่ Weight vector ( $W_{ji}^h$ ): ในชั้นของ Hidden Layer จะมีค่าเท่ากับค่าของ Input Vector แรกที่เริ่มต้นส่งเข้ามาในกระบวนการรู้จำ สำหรับ Weight Vector ( $W_{1h}$ ): ในชั้นของ Cluster Layer ก็จะมีการกำหนดค่าเริ่มต้นให้เท่ากับค่า Weight Vector ( $W_1^h$ ) ตามสมการต่อไปนี้

$$W_{li}^h = I_i^h \quad (3.12)$$

$$W_{lh} = W_l^h \quad (3.13)$$

กำหนดให้	$W_{li}^h$	คือค่า Weight vector อ้างอิงของ โหนด $j=1$ ซึ่งเป็น โหนดแรก ที่กำหนดขึ้นมาในชั้น Hidden Layer สำหรับการเรียนรู้ครั้งแรก
	$W_{lh}$	คือค่า Weight vector อ้างอิงของ โหนด $k=1$ ซึ่งเป็น โหนดแรก ที่สร้างขึ้นมาในชั้น Cluster Layer สำหรับการเรียนรู้ครั้งแรก
	$I_i^h$	แทน Input vector แรกที่ส่งเข้ามาใน Input Layer ในแต่ละกรอบย่อยที่ $h$
	$i$	คือ จำนวน element ใน Input vector หรือจำนวน Input Nodes ที่ส่งเข้ามา

จำนวน neurons ในชั้น Hidden Layer = กรอบย่อยของแต่ละส่วนของตัวอักษร

### ขั้นตอนที่ 3

จากนั้นเมื่อรับเอา Input Pattern ของภาพตัวอักษรตัวต่อไปเข้ามาจะมีการคำนวณค่า Choice Function ( $T_j^h(I^h)$ ) ตามสมการที่ 3.14 โดยสามารถหาได้จากค่า Maximum of The Normalized Cross-correlation ระหว่าง Input Vector ( $I^h$ ) ของกรอบย่อยที่  $h$  กับค่า Weight Vector ( $W_j^h$ ) ที่อ้างอิงของกรอบย่อยที่  $h$  เดียวกัน ในทุก ๆ ค่า  $p$  ที่กำหนดตามสมการที่ 3.15 แล้วจะได้ โหนด  $J$  ที่มีค่า  $T_j^h(I^h)$  มากที่สุดออกมา ค่าที่ได้จะมีค่า  $0 \leq f(I^h, W_j^h) \leq 1$  ซึ่งถ้าค่าที่ได้ออกมามีค่า เข้าใกล้ 0 นั้นแสดงว่าความคล้ายคลึงกันระหว่าง Input Vector กับ Weight Vector นั้นมีความ คล้ายคลึงกันน้อยที่สุดหรือแทบจะไม่คล้ายคลึงกันเลย แต่หากค่าที่ได้มีค่าเข้าใกล้ 1 จะหมายถึงค่า Input Vector และค่า Weight Vector มีความคล้ายคลึงกันมาก โดยที่จะพิจารณาคำนวณค่า  $T_j^h(I^h)$  ออกมาในทุก ๆ กรอบย่อย  $h$  และสำหรับแต่ละ โหนด  $J$  ใน Hidden Layer ที่มีค่า  $T_j^h(I^h)$  สูงสุดออกมาจะถูกเลือกให้เป็น โหนดที่ชนะ (Winning Node) ตามสมการที่ 3.16 ในกรณีที่หากมีค่า มากที่สุดมากกว่า 1 ค่า จะเลือกค่าที่มากที่สุดจากลำดับ โหนด  $J$  ที่น้อยที่สุดออกมา

$$T_j^h(I^h) = f(I^h, W_j^h) \quad (3.14)$$

$$f(x, y) = \max_p \left( \frac{\sum_{i=1}^N x_i y_{i-p}}{\sqrt{\sum_{i=1}^N x_i^2 \sum_{i=1}^N y_i^2}} \right) \quad (3.15)$$

$$J = \arg \max_j \{T_j^h(I^h)\} \quad (3.16)$$

กำหนดให้	$T_j^h(I^h)$	แทนค่า Choice Function ที่ใช้วัดความคล้ายคลึงกันระหว่าง Input Vector กับ Weight Vector ของโหนด $j$ ใน Hidden Layer
	$f(x, y)$	แทนค่า Maximum of The Normalized Cross-correlation at any shifted period $p$
	$p$	คือค่า Lag Variable ที่มีค่าระหว่าง $-(N-1)$ ถึง $(N-1)$
	$N$	คือจำนวน Feature components ของ Input vector ในแต่ละ $h^{\text{th}}$ segment
	$I$	คือ Input Vector
	$W$	คือ Weight Vector
	$h$	คือลำดับกรอบย่อยที่ส่งค่า Input Vector เข้ามาในการรู้จำ มีค่าเท่ากับ 1, 2, 3, ..., 9
	$i$	มีค่าเท่ากับ 1, 2, 3, ..., $N$
	$j$	มีค่าเท่ากับ 1, 2, 3, ..., $M^h$
	$J$	เป็น Winning Node

#### ขั้นตอนที่ 4

ตรวจสอบค่า Vigilance Parameter ( $\rho$ ) ซึ่งเป็นค่าความคล้ายคลึงกันระหว่าง Input Vector และ Weight Vector ของโหนดที่ชนะ  $J$  นั้นว่ามีความคล้ายคลึงกันมากพอที่จะจัดให้ Input นั้นไปรวมอยู่ในกลุ่ม Category ของโหนด  $J$  นั้นหรือไม่ จาก Vigilance Criterion โดยจะนำเอาค่า  $T_j^h(I^h)$  มาเปรียบเทียบกับค่า  $\rho_{\text{hidden}}$  ตามสมการที่ 3.17 หากค่าที่ได้จากสมการนั้นเป็นจริงแสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะ  $J$  นั้นได้ ต่อจากนั้นจะทำการปรับค่า Weight Vector ( $W_j^h$ ) สำหรับโหนดที่ชนะ  $J$  ตามสมการที่ 3.18 จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$T_j^h(I^h) \geq \rho_{\text{hidden}} \quad (3.17)$$

$$W_J^h(t+1) = \beta(I^h \wedge W_J^h(t)) + (1 - \beta)W_J^h(t) \quad (3.18)$$

กำหนดให้	$T_J^h(I^h)$	แทนค่า Choice Function ที่ใช้วัดความคล้ายคลึงกันระหว่าง Input Vector กับ Weight Vector ของ โหนด J ใน Hidden Layer
	$\rho_{\text{hidden}}$	คือค่า Vigilance Parameter ที่ Hidden Layer มีค่าระหว่าง 0 - 1
	$W_J^h(t+1)$	คือค่า Weight Vector ใหม่ที่ปรับเปลี่ยนสำหรับ โหนด J
	$W_J^h(t)$	คือค่า Weight Vector เก่าของ โหนด J
	$I^h$	คือค่า Input Vector ที่เป็น Pattern นำมาเปรียบเทียบกับค่า Weight Vector ของ โหนด J ใน Hidden Layer
	$\beta$	คือค่า Learning Parameter จะมีค่าได้ 3 ค่าตามความเร็วของการเรียนรู้ดังนี้ <ul style="list-style-type: none"> <li>○ <math>\beta=1</math> เรียกว่า fast learning ส่งผลให้ค่าของ Weight Vector มีการลดลงเปลี่ยนแปลงไปอย่างรวดเร็วขึ้นอยู่กับ Input และ Weight Vector</li> <li>○ <math>0 &lt; \beta &lt; 1</math> แสดงว่ามีการเรียนรู้แบบค่อยเป็นค่อยไป ส่งผลให้ค่าของ Weight Vector มีการลดลงอย่างเรื่อย ๆ เป็นสัดส่วนกับค่า <math>\beta</math> ที่เลือกใช้</li> <li>○ <math>\beta=0</math> แสดงว่าไม่มีการเรียนรู้เลย ส่งผลให้ค่าของ Weight Vector ไม่มีการเปลี่ยนแปลงตลอดการเรียนรู้</li> </ul>

แต่ในกรณีที่ค่าที่ได้จากสมการที่ 3.17 นั้นเป็นเท็จ แสดงว่า Input นั้นมีความคล้ายคลึงกันไม่มากพอที่จะจัดเข้ารวมอยู่กับ โหนดที่ชนะ J นั้นได้ ต่อจากนั้นจะทำการสร้าง โหนดใหม่ใน Hidden Layer และกำหนดให้มี Weight Vector ของ โหนดใหม่ที่สร้างขึ้นมาเท่ากับค่า Input Vector ของช่วงกรอบส่วนย่อย h นั้นตามสมการที่ 3.19 จะเรียกกรณีนี้ว่าการเกิด “Reset”

$$W_J^h = I^h \quad (3.19)$$

กำหนดให้	$W_J^h$	คือค่า Weight Vector ใหม่ที่กำหนดให้กับ โหนด J ใน Hidden Layer ใหม่ที่สร้างขึ้นมา
	$I^h$	คือค่า Input Vector ที่เป็น Pattern ที่กำหนดให้กับ Hidden Node ใหม่ที่สร้างขึ้นมา

$$W_J^h(t+1) = \beta(I^h \wedge W_J^h(t)) + (1 - \beta)W_J^h(t) \quad (3.18)$$

กำหนดให้	$T_J^h(I^h)$	แทนค่า Choice Function ที่ใช้วัดความคล้ายคลึงกันระหว่าง Input Vector กับ Weight Vector ของโหนด J ใน Hidden Layer
	$\rho_{\text{hidden}}$	คือค่า Vigilance Parameter ที่ Hidden Layer มีค่าระหว่าง 0 - 1
	$W_J^h(t+1)$	คือค่า Weight Vector ใหม่ที่ปรับเปลี่ยนสำหรับโหนด J
	$W_J^h(t)$	คือค่า Weight Vector เก่าของโหนด J
	$I^h$	คือค่า Input Vector ที่เป็น Pattern นำมาเปรียบเทียบกับค่า Weight Vector ของโหนด J ใน Hidden Layer
	$\beta$	คือค่า Learning Parameter จะมีค่าได้ 3 ค่าตามความเร็วของการเรียนรู้ดังนี้ <ul style="list-style-type: none"> <li>○ <math>\beta=1</math> เรียกว่า fast learning ส่งผลให้ค่าของ Weight Vector มีการลดลงเปลี่ยนแปลงไปอย่างรวดเร็วขึ้นอยู่กับ Input และ Weight Vector</li> <li>○ <math>0 &lt; \beta &lt; 1</math> แสดงว่ามีการเรียนรู้แบบค่อยเป็นค่อยไป ส่งผลให้ค่าของ Weight Vector มีการลดลงอย่างเรื่อย ๆ เป็นสัดส่วนกับค่า <math>\beta</math> ที่เลือกใช้</li> <li>○ <math>\beta=0</math> แสดงว่าไม่มีการเรียนรู้เลย ส่งผลให้ค่าของ Weight Vector ไม่มีการเปลี่ยนแปลงตลอดการเรียนรู้</li> </ul>

แต่ในกรณีที่ค่าที่ได้จากสมการที่ 3.17 นั้นเป็นเท็จ แสดงว่า Input นั้นมีความคล้ายคลึงกันไม่มากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะ J นั้นได้ ต่อจากนั้นจะทำการสร้างโหนดใหม่ใน Hidden Layer และกำหนดให้มี Weight Vector ของโหนดใหม่ที่สร้างขึ้นมาเท่ากับค่า Input Vector ของช่วงกรอบส่วนย่อย h นั้นตามสมการที่ 3.19 จะเรียกกรณีนี้ว่าการเกิด “Reset”

$$W_J^h = I^h \quad (3.19)$$

กำหนดให้	$W_J^h$	คือค่า Weight Vector ใหม่ที่กำหนดให้กับโหนด J ใน Hidden Layer ใหม่ที่สร้างขึ้นมา
	$I^h$	คือค่า Input Vector ที่เป็น Pattern ที่กำหนดให้กับ Hidden Node ใหม่ที่สร้างขึ้นมา

### ขั้นตอนที่ 5

จากนั้นจะส่งต่อค่า Weight Vector ของโหนดที่ชนะ  $J$  (กรณีการเกิด Resonance) หรือค่า Weight Vector ของโหนดใหม่ที่สร้างขึ้นมาจากโหนด  $J$  (กรณีการเกิด Reset) ใน Hidden Layer ไปยัง Cluster Layer แล้วมีการคำนวณหาค่า Choice Function กับแต่ละโหนด  $k^h$  ใน Cluster Layer ซึ่งจะคำนวณตามสมการที่ 3.20

$$C_k = f(W_J, W_k) \quad (3.20)$$

กำหนดให้	$W_J$	คือค่า Weight Vector ที่ส่งต่อขึ้นมาจาก Hidden Layer โดยจะมีค่า $W_J = \{W_J^1, W_J^2, W_J^3, \dots, W_J^9\}$
	$W_k$	คือค่า Reference Weight Vector ใน Cluster Layer โดยจะมีค่า $W_k = \{W_{k1}, W_{k2}, W_{k3}, \dots, W_{k9}\}$
	$f(W_J, W_k)$	เป็นฟังก์ชันที่คำนวณได้ตามสมการที่ 3.14
	$C_k$	คือค่า Choice Function ที่ใช้วัดความคล้ายคลึงกันระหว่าง Weight Vector ของ โหนด $J$ ใน Hidden Layer กับ Weight Vector ของโหนด $k$ ใน Cluster Layer

### ขั้นตอนที่ 6

ต่อจากนั้นจะทำการเลือกโหนดที่ชนะ ( $K$ ) ใน Cluster Layer ที่มีค่า Maximum Choice Function เมื่อเปรียบเทียบกับทุกโหนดใน Cluster Layer ตามสมการที่ 3.21 โดยที่หากมีโหนดมากกว่า 1 โหนดที่มีค่า Maximum Choice Function จะทำการเลือกลำดับโหนดที่ชนะ  $K$  ที่มีลำดับน้อยที่สุดออกมา

$$K = \arg \max_k \{C_k\} \quad (3.21)$$

กำหนดให้	$C_k$	คือค่า Choice Function ที่ใช้วัดความคล้ายคลึงกันระหว่าง Weight Vector ของ โหนด $J$ ใน Hidden Layer กับ Weight Vector ของโหนด $k$ ใน Cluster Layer
	$K$	คือโหนดที่ถูกเลือกให้เป็นโหนดที่ชนะใน Cluster Layer ที่มีค่า Maximum Choice Function

## ขั้นตอนที่ 7

ทำการเปรียบเทียบ  $C_K$  กับ Vigilance Criterion ที่มีการกำหนดค่าไว้ใน Cluster Layer โดยพิจารณา Vigilance Parameter ( $\rho_{cluster}$ ) ที่ Cluster Layer และทำการเปรียบเทียบกันตามสมการที่ 3.22 เพื่อตรวจสอบว่าโหนด K ที่เลือกนั้นจัดเป็นกลุ่ม Output Class ที่ถูกต้องที่สามารถจัด Input Pattern นั้นเข้ากลุ่มได้หรือไม่ หากสมการที่ 3.22 เป็นจริงหรือเรียกว่าเกิดกรณี “Resonance” แสดงว่า Weight Vector ( $W_J$ ) นั้นมีความคล้ายคลึงกับ Weight Vector ( $W_k$ ) ของโหนดที่เลือกให้เป็นโหนดที่ชนะคือโหนด K ใน Cluster Layer และตรงกับกลุ่มผลลัพธ์ของ Input Pattern นอกจากนี้จะพิจารณาต่อว่าหาก  $W_J$  นั้นเป็นค่า Weight Vector ที่ส่งค่ามาจากโหนด J ที่มีการสร้างขึ้นใหม่ใน Hidden Layer ก็จะมีการปรับค่า Weight Vector ( $W_{kh}(t+1)$ ) ตามสมการที่ 3.23 แต่หาก  $W_J$  นั้นเป็นค่า Weight Vector ที่ส่งค่ามาจากโหนด J ที่เป็นโหนดเดิมอยู่แล้วใน Hidden Layer ก็จะมีการกำหนดค่า Weight Vector ( $W_{kh}(t+1)$ ) ตามสมการที่ 3.24

$$C_K \geq \rho_{cluster} \quad (3.22)$$

$$W_{kh}(t+1) = \beta(W_J^h \wedge W_{kh}(t)) + (1-\beta)W_{kh}(t) \quad (3.23)$$

$$W_{kh}(t+1) = W_{kh}(t) \quad (3.24)$$

กำหนดให้	$C_K$	คือค่า Maximum Choice Function ของโหนดที่เลือกให้ชนะ K
	$\rho_{cluster}$	คือค่า Vigilance Parameter ที่ Cluster Layer โดยจะมีค่าระหว่าง 0 และ 1
	$W_J^h$	คือค่า Weight Vector ที่ส่งต่อมาจากโหนด J ที่สร้างขึ้นมาใหม่ใน Hidden Layer
	$W_{kh}(t+1)$	คือค่า Weight Vector ใหม่ที่ปรับเปลี่ยนสำหรับโหนด k ที่เชื่อมกับกรอบย่อยที่ h
	$W_{kh}(t)$	คือค่า Weight Vector เก่าของโหนด k ที่เชื่อมกับกรอบย่อยที่ h

ทั้งนี้หากสมการที่ 3.22 เป็นเท็จหรือเรียกว่าเป็นกรณี “Reset” แสดงว่าโหนดที่เลือกให้เป็นโหนดชนะ K ที่ Cluster Layer นั้นมีการจัดเข้ากลุ่มที่ไม่ถูกต้องตรงตาม กลุ่ม Output Class ของ Input Pattern จะมีการสร้างโหนดใหม่ขึ้นมาใน Cluster Layer และทำการกำหนด Weight Vector ของโหนดใหม่ k ( $W_k$ ) ให้มีค่าเท่ากับค่า Weight Vector ( $W_J$ ) ที่ส่งต่อขึ้นมาจาก Hidden Layer ตามสมการที่ 3.25 หลังจากนั้นจะมีการสร้างการเชื่อมต่อจากโหนดใหม่ใน Cluster Layer ไปยังโหนดใน Output Layer ที่ตรงตาม Target Output ของ Input Pattern ที่ทำการเรียนรู้ครั้งนั้น

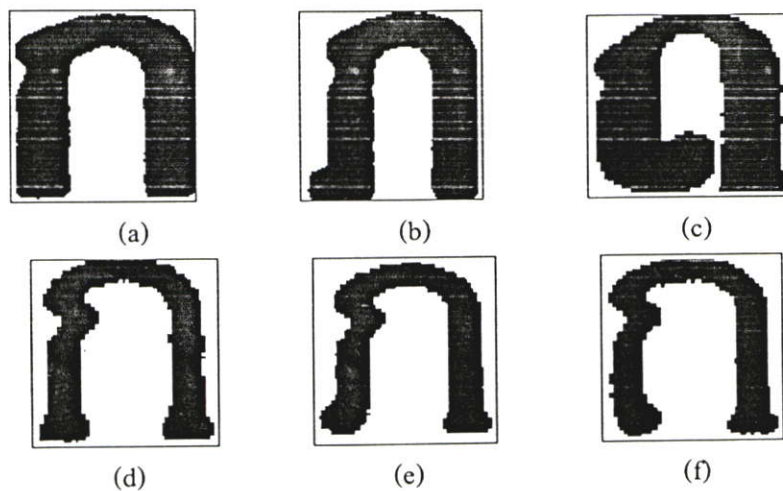
$$W_k = W_J \quad (3.25)$$

กำหนดให้	$k$	คือโหนดใหม่ที่ทำกรสร้างขึ้นมาใน Cluster Layer
	$W_k$	คือค่า Weight Vector ที่กำหนดขึ้นมาใหม่ให้กับโหนด $k$ ที่สร้างขึ้นมาใหม่ใน Cluster Layer
	$W_J$	คือค่า Weight Vector ที่ส่งต่อมาจากโหนด $J$ ใน Hidden Layer

ช่วงของการทดสอบระบบ รูปแบบอินพุตที่เข้ามาจะเริ่มจากการแบ่งกลุ่มในการแยกแยะกลุ่ม 3 กลุ่มได้แก่ระดับบน กลาง และล่างตามลำดับ จากนั้นจะทำการทดสอบส่งผ่านค่าเข้ามาแล้วคาดคะเนตาม Cluster Node ที่ได้ออกมาจากค่าความคล้ายคลึงกันมากที่สุดที่ระบบได้ทำการคำนวณออกมาว่ามีการจัดเข้าประเภท Class ไหนใน Output Node ก็จะได้คำตอบของแต่ละอินพุตที่ใส่เข้าไปออกมาจากกระบวนการรู้จำแบบใหม่นี้

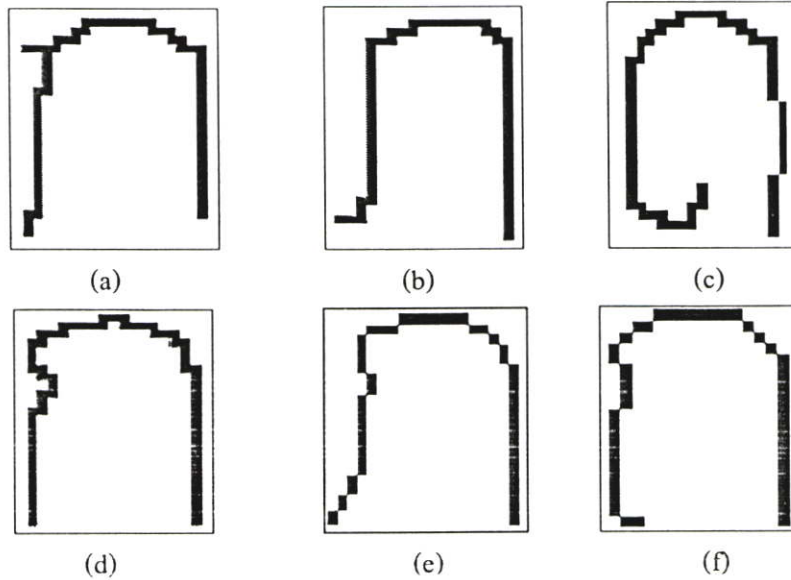
### 3.3 ตัวอย่างการคำนวณ

3.3.1 ข้อมูลใน Training Set มีทั้งหมด 6 รูปแบบได้แก่ตัวอักษร “ก” “ภ” “ด” “ก” “ภ” และ “ด” ตามลำดับดังรูปที่ 3.25



รูปที่ 3.25 รูปแบบข้อมูล Training Set (a) ตัวอักษรตัวที่ 1 คือ “ก” (b) ตัวอักษรตัวที่ 2 คือ “ภ” (c) ตัวอักษรตัวที่ 3 คือ “ด” (d) ตัวอักษรตัวที่ 4 คือ “ก” (e) ตัวอักษรตัวที่ 5 คือ “ภ” (f) ตัวอักษรตัวที่ 6 คือ “ด”

ต่อจากนั้นเข้าสู่กระบวนการก่อนการรู้จำเริ่มจากการแบ่งออกเป็นกรอบตัวอักษรแต่ละตัว ดังรูปที่ 3.25 แล้วทำการลดขนาดให้เท่ากับ 25x29 Pixels จากนั้นทำการ Skeleton ตัวอักษรแต่ละตัว ที่รับเข้ามา หลังจากนั้นจะได้กรอบตัวอักษรที่มีการ Skeleton แล้วดังรูปที่ 3.26



รูปที่ 3.26 ข้อมูลภาพตัวอักษรใน Training Set ที่มีการ Resize [29, 25] และทำการ Skeleton แล้ว

(a) ตัวอักษรตัวที่ 1 คือ “ก” (b) ตัวอักษรตัวที่ 2 คือ “ภ” (c) ตัวอักษรตัวที่ 3 คือ “จ”

(d) ตัวอักษรตัวที่ 4 คือ “ก” (e) ตัวอักษรตัวที่ 5 คือ “ภ” (f) ตัวอักษรตัวที่ 6 คือ “จ”

ขั้นตอนต่อไปคือการแทนค่ารหัสแทนตัวอักษรของภาพตัวอักษรใน Training Set ตัวที่ 1, 2, 3, 4, 5, และ 6 ตามตารางที่ 3.5, 3.6, 3.7, 3.8, 3.9, และ 3.10 ตามลำดับ

ตารางที่ 3.5 การแทนค่ารหัสแทนเส้นทิศทางของภาพตัวอักษรใน Training Set ตัวที่ 1

กรอบย่อยที่	ค่าข้อมูล	Input Vector มีขนาดเท่ากับ 12
1 ( $I^1$ )	0.25 0.25 0 0.25 0.25 0.25 0.25 0.25	0.25 0.25 0 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
2 ( $I^2$ )	0.25 0.25 0.25 0.25 0.25 0.25 0 0.25	0.25 0.25 0.25 0.25 0.25 0.25 0 0.25 0.25 0.25 0.25 0.25
3 ( $I^3$ )	0.25 0.25 0.25 0.5 0.5 0 0.25 0 0 0.25 0	0.25 0.25 0.25 0.5 0.5 0 0.25 0 0 0.25 0 0
4 ( $I^4$ )	0.25 0 0 0 0 0 0 0	0.25 0 1 1 1 1 1 1 1 1 1 0

ตารางที่ 3.5 (ต่อ)

กรอบย่อยที่	ค่าข้อมูล	Input Vector มีขนาดเท่ากับ 12
5 ( $I^5$ )	0.75 1 1 0.75 1 0.75 1 1 0.75 0.75 0.75 0.75	0.75 1 1 0.75 1 0.75 1 1 0.75 0.75 0.75 0.75
6 ( $I^6$ )	0.75 0.75 0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
7 ( $I^7$ )	0.75 0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
8 ( $I^8$ )	-	0 0 0 0 0 0 0 0 0 0 0 0
9 ( $I^9$ )	-	0 0 0 0 0 0 0 0 0 0 0 0

ตารางที่ 3.6 การแทนค่ารหัสแทนเส้นทางของภาพตัวอักษรใน Training Set ตัวที่ 2

กรอบย่อยที่	ค่าข้อมูล	Input Vector มีขนาดเท่ากับ 11
1 ( $I^1$ )	0 0 0 0.25 0.25 0 0.25 0.25 0.25 0.25	0 1 0 0.25 0.25 0 0.25 0.25 0.25 0.25 0.25
2 ( $I^2$ )	0.25 0.25 0.25 0.25 0.25 0.25	0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
3 ( $I^3$ )	0.25 0.25 0.25 0.25 0 0	0.25 0.25 0.25 0.25 0 1 1 1 1 1 0
4 ( $I^4$ )	0.25 0 0 0.25 0 0 0 0	0.25 0 0 0.25 0 1 1 1 1 1 0
5 ( $I^5$ )	0 0 0.75 1 0.75 1 0.75 0.75 0.75 0.75 0.75	0 0 0.75 1 0.75 1 0.75 0.75 0.75 0.75 0.75
6 ( $I^6$ )	0.75 0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
7 ( $I^7$ )	0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
8 ( $I^8$ )	-	0 0 0 0 0 0 0 0 0 0 0
9 ( $I^9$ )	-	0 0 0 0 0 0 0 0 0 0 0

ตารางที่ 3.7 การแทนค่ารหัสแทนเส้นทางของภาพตัวอักษรใน Training Set ตัวที่ 3

กรอบย่อยที่	ค่าข้อมูล	Input Vector มีขนาดเท่ากับ 12
1 ( $I^1$ )	0.5 0.5 0.625 0.5 0.5 0.5	0.5 0.5 0.625 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
2 ( $I^2$ )	0.5 0.5 0.375 0.375 0.25 0.375 0.25 0.25 0.25	0.5 0.5 0.375 0.375 0.25 0.375 0.25 0.25 0.25 0.25 0.25 0.25
3 ( $I^3$ )	0.25 0.125 0.25 0.25 0.25 0.375 0.25	0.25 0.125 0.25 0.25 0.25 0.375 0.25 0.25 0.25 0.25 0.25 0.25
4 ( $I^4$ )	0.25 0.125 0.125 0.125 0.125 0.125 0	0.25 0.125 0.125 0.125 0.125 0.125 0 1 1 1 1 0
5 ( $I^5$ )	0 0 0 0 0 0.875	0 1 1 1 0 0.875 0.875 0.875 0.875 0.875 0.875 0.875
6 ( $I^6$ )	1 0.875 1 0.875 0.75 0.75 0.75 0.75	1 0.875 1 0.875 0.75 0.75 0.75 0.75 0.75
7 ( $I^7$ )	0.75 0.75 0.875 0.625 0.75 0.75 0.875	0.75 0.75 0.875 0.625 0.75 0.75 0.875 0.875 0.875
8 ( $I^8$ )	0.625 0.75 0.75 0.75 0.75 0.75	0.625 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
9 ( $I^9$ )	-	0 0 0 0 0 0 0 0 0 0

ตารางที่ 3.8 การแทนค่ารหัสแทนเส้นทางของภาพตัวอักษรใน Training Set ตัวที่ 4

กรอบย่อยที่	ค่าข้อมูล	Input Vector มีขนาดเท่ากับ 13
1 ( $I^1$ )	0.25 0.25 0.25 0.25 0.25 0.25 0.25	0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
2 ( $I^2$ )	0.25 0.25 0.25 0.25 0 0.25 0.25 0 0.25 0.25	0.25 0.25 0.25 0.25 0 0.25 0.25 0 0.25 0.25 0.25 0.25 0.25
3 ( $I^3$ )	0.5 0.25 0.5 0.25 0.25 0.25 0 0.25 0 0 0.25 0 0	0.5 0.25 0.5 0.25 0.25 0.25 0 0.25 0 0 0.25 0 0
4 ( $I^4$ )	0 0.25 0 0 0.75 1 1 1	0 0.25 0 0 0.75 1 1 1 1 1 1 1 1
5 ( $I^5$ )	0.75 1 1 0.75 1 0.75 0.75 0.75 1 0.75 0.75	0.75 1 1 0.75 1 0.75 0.75 0.75 1 0.75 0.75 0.75 0.75

ตารางที่ 3.8 (ต่อ)

กรอบย่อยที่	ค่าข้อมูล	Input Vector มีขนาดเท่ากับ 13
6 ( $I^6$ )	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
7 ( $I^7$ )	0.75 0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
8 ( $I^8$ )	-	0 0 0 0 0 0 0 0 0 0 0 0 0
9 ( $I^9$ )	-	0 0 0 0 0 0 0 0 0 0 0 0 0

ตารางที่ 3.9 การแทนค่ารหัสแทนเส้นทางของภาพตัวอักษรใน Training Set ตัวที่ 5

กรอบย่อยที่	ค่าข้อมูล	Input Vector มีขนาดเท่ากับ 8
1 ( $I^1$ )	0.125 0.25 0.125 0.25 0.125 0.25 0.25	0.125 0.25 0.125 0.25 0.125 0.25 0.25 0.25
2 ( $I^2$ )	0.25 0.25 0.25 0.25 0.25 0.125 0.25	0.25 0.25 0.25 0.25 0.25 0.125 0.25 0.25
3 ( $I^3$ )	0.375 0.25 0.25 0.25 0.125 0 0	0.375 0.25 0.25 0.25 0.125 0 1 0
4 ( $I^4$ )	0.125 0 0 0 0 0	0.125 0 1 1 1 1 1 0
5 ( $I^5$ )	0.875 1 0.875 0.875 0.75 0.875 0.75 0.75	0.875 1 0.875 0.875 0.75 0.875 0.75 0.75
6 ( $I^6$ )	0.75 0.75 0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
7 ( $I^7$ )	0.75 0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
8 ( $I^8$ )	-	0 0 0 0 0 0 0 0
9 ( $I^9$ )	-	0 0 0 0 0 0 0 0

ตารางที่ 3.10 การแทนค่ารหัสแทนเส้นทางของภาพตัวอักษรใน Training Set ตัวที่ 6

กรอบย่อยที่	ค่าข้อมูล	Input Vector มีขนาดเท่ากับ 9
1 ( $I^1$ )	0.5 0.375 0.25 0.25 0.25 0.25 0.25 0.25	0.5 0.375 0.25 0.25 0.25 0.25 0.25 0.25 0.25
2 ( $I^2$ )	0.25 0.25 0.25 0.25 0.125 0.25 0.25	0.25 0.25 0.25 0.25 0.125 0.25 0.25 0.25 0.25

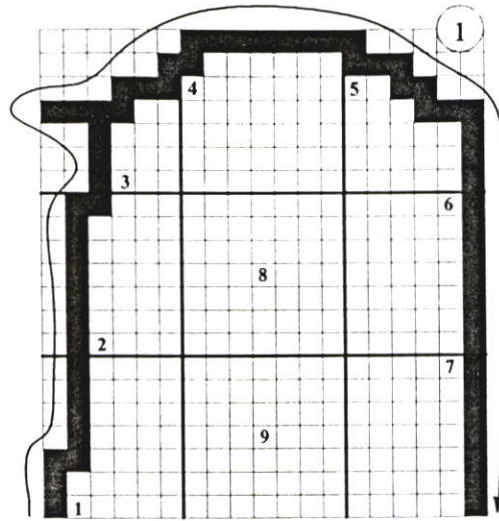
ตารางที่ 3.10 (ต่อ)

กรอบย่อยที่	ค่าข้อมูล	Input Vector มีขนาดเท่ากับ 9
3 ( $1^3$ )	0.25 0.375 0.25 0.125 0.125 0 0 0.125 0	0.25 0.375 0.25 0.125 0.125 0 0 0.125 0
4 ( $1^4$ )	0 0 0 0 0 0	0 1 1 1 1 1 1 1 0
5 ( $1^5$ )	0 0.875 0.875 0.875 0.875 0.75 0.75 0.75	0 0.875 0.875 0.875 0.875 0.75 0.75 0.75 0.75
6 ( $1^6$ )	0.75 0.75 0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
7 ( $1^7$ )	0.75 0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
8 ( $1^8$ )	-	0 0 0 0 0 0 0 0 0
9 ( $1^9$ )	-	0 0 0 0 0 0 0 0 0

จะสังเกตเห็นว่าทั้ง 6 รูปแบบภาพตัวอักษรใน Training Set ที่รับเข้ามาเมื่อหาจุดศูนย์กลางของกรอบตัวอักษรแล้วอยู่ในกลุ่มระดับกลาง จากนั้นจะส่งรูปแบบตัวอักษรแต่ละตัวเข้าสู่ระบบการรู้จำแนวทางใหม่ในกระบวนการรู้จำตัวอักษรภาษาไทยด้วยวิธีใหม่ที่น่าสนใจ

### กระบวนการรู้จำในขั้นตอนของการ Training

การเรียนรู้ครั้งที่ 1 รับค่ารูปแบบตัวอักษรที่มีการกำหนด Input Vector ตามตารางที่ 3.5 โดยรูปแบบของตัวอักษรมีการแบ่งออกเป็น 9 กรอบย่อยแสดงดังรูปที่ 3.27 แล้วทำการกำหนดค่า Vigilance Parameter ( $\rho$ ) ได้แก่ ค่า  $\rho_{\text{hidden}} = 0.75$  และค่า  $\rho_{\text{cluster}} = 0.85$  สำหรับค่า Learning Rate ( $\beta$ ) กำหนดให้เท่ากับ 0.05 จากนั้นกำหนดค่าเริ่มต้นให้กับ Weight Vector โดยที่ Weight vector ( $W_{ji}^h$ ): ในชั้นของ Hidden Layer มีค่าเท่ากับ Input Vector แรกที่เริ่มต้นส่งเข้ามาในกระบวนการรู้จำ ค่า Weight Vector ที่ชั้น Hidden Layer ในการเรียนรู้ครั้งที่ 1 แสดงตามตารางที่ 3.11



รูปที่ 3.27 แสดงทิศทางของเส้นตัวอักษรในการ Training ตัวที่ 1 ตามการจัดเรียงกรอบย่อยของ ภาพตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร

ตารางที่ 3.11 ค่า Weight Vector ที่ชั้น Hidden Layer ในการเรียนรู้ครั้งที่ 1

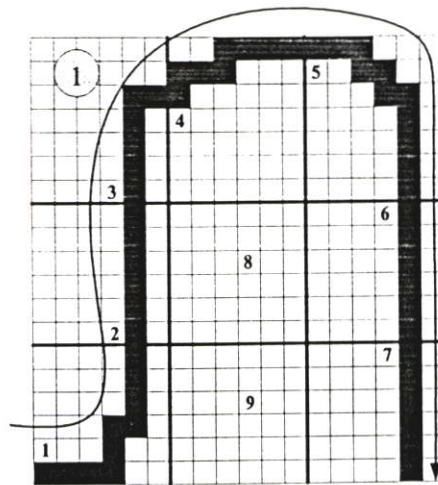
กรอบย่อยที่	Weight Vector ที่ชั้น Hidden Layer
1 ( $W_{li}^1$ )	0.25 0.25 0 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
2 ( $W_{li}^2$ )	0.25 0.25 0.25 0.25 0.25 0.25 0 0.25 0.25 0.25 0.25 0.25 0.25
3 ( $W_{li}^3$ )	0.25 0.25 0.25 0.5 0.5 0 0.25 0 0 0.25 0 0
4 ( $W_{li}^4$ )	0.25 0 1 1 1 1 1 1 1 1 1 0
5 ( $W_{li}^5$ )	0.75 1 1 0.75 1 0.75 1 1 0.75 0.75 0.75 0.75
6 ( $W_{li}^6$ )	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
7 ( $W_{li}^7$ )	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
8 ( $W_{li}^8$ )	0 0 0 0 0 0 0 0 0 0 0 0
9 ( $W_{li}^9$ )	0 0 0 0 0 0 0 0 0 0 0 0

จากนั้นจะส่งต่อค่า Weight Vector ของโหนด  $J$  ใน Hidden Layer ต่อไปให้เป็นค่า Input Vector ของ Cluster Layer ส่วน Weight Vector ( $W_{lh}$ ): ในชั้นของ Cluster Layer ก็จะมีการกำหนดค่าเริ่มต้นให้เท่ากับค่า Weight Vector ( $W_l^h$ ) ที่ส่งขึ้นมาเป็น Input ตามสมการที่ 3.19 ดังนั้นค่า Weight Vector ที่ชั้น Cluster Layer ในการเรียนรู้ครั้งที่ 1 แสดงตามตารางที่ 3.12 จากนั้นเปรียบเทียบกับคำตอบที่แท้จริงจะได้ Output Node ที่ 1 คือ Target Output = “ก”

ตารางที่ 3.12 ค่า Weight Vector ที่ชั้น Cluster Layer ในการเรียนรู้ครั้งที่ 1

กรอบย่อยที่	Weight Vector ที่ชั้น Cluster Layer
1 ( $W_{11}$ )	0.25 0.25 0 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
2 ( $W_{12}$ )	0.25 0.25 0.25 0.25 0.25 0.25 0 0.25 0.25 0.25 0.25 0.25
3 ( $W_{13}$ )	0.25 0.25 0.25 0.5 0.5 0 0.25 0 0 0.25 0 0
4 ( $W_{14}$ )	0.25 0 1 1 1 1 1 1 1 1 1 0
5 ( $W_{15}$ )	0.75 1 1 0.75 1 0.75 1 1 0.75 0.75 0.75 0.75
6 ( $W_{16}$ )	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
7 ( $W_{17}$ )	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
8 ( $W_{18}$ )	0 0 0 0 0 0 0 0 0 0 0 0
9 ( $W_{19}$ )	0 0 0 0 0 0 0 0 0 0 0 0

การเรียนรู้ครั้งที่ 2 รับค่ารูปแบบตัวอักษรตัวที่ 2 ที่มีการกำหนด Input Vector ตามตารางที่ 3.6 โดยรูปแบบของตัวอักษรที่มีการแบ่งออกเป็น 9 กรอบย่อยแสดงดังรูปที่ 3.28



รูปที่ 3.28 แสดงทิศทางของเส้นตัวอักษรในการ Training ตัวที่ 2 ตามการจัดเรียงกรอบย่อยของภาพตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร

ทำการคำนวณหาค่า Choice Function ( $T_j^h(I^h)$ ) ตามสมการที่ 3.14 แล้วนำเอาค่า  $T_j^h(I^h)$  มาเปรียบเทียบกับค่า  $\rho_{\text{hidden}} = 0.75$  ที่กำหนดไว้ แล้วทำการพิจารณาตรวจสอบค่า Vigilance Parameter ที่ Hidden Layer ( $\rho_{\text{hidden}}$ ) ตามสมการที่ 3.17 ดังนี้

กรอบย่อยที่ 1: คำนวณหาค่า Choice Function ( $T_1^1(I^1)$ ) = 0.7385

$$W_{li}^1 = 0.2500 \ 0.2500 \ 0 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500$$

เนื่องจาก  $T_1^1(I^1) < \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันไม่มากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการสร้างโหนดใหม่ใน Hidden Layer และกำหนดให้มี Weight Vector ของโหนดใหม่ที่สร้างขึ้นมามีค่าเท่ากับค่า Input Vector ของช่วงกรอบย่อย h นั้น จะเรียกกรณีนี้ว่าการเกิด “Reset”

$$W_{2i}^1 = 0 \ 1.0000 \ 0 \ 0.2500 \ 0.2500 \ 0 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500$$

กรอบย่อยที่ 2: คำนวณหาค่า Choice Function ( $T_1^2(I^2)$ ) = 0.9574

$$W_{li}^2 = 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500$$

เนื่องจาก  $T_1^2(I^2) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^2(t+1) = 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500$$

กรอบย่อยที่ 3: คำนวณหาค่า Choice Function ( $T_1^3(I^3)$ ) = 0.8473

$$W_{li}^3 = 0.2500 \ 0.2500 \ 0.2500 \ 0.5000 \ 0.5000 \ 0 \ 0.2500 \ 0 \ 0 \ 0.2500 \ 0 \ 0$$

เนื่องจาก  $T_1^3(I^3) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^3(t+1) = 0.2500 \ 0.2500 \ 0.2500 \ 0.4875 \ 0.4750 \ 0 \ 0.2500 \ 0 \ 0 \ 0.2500 \ 0 \ 0$$

กรอบย่อยที่ 4: คำนวณหาค่า Choice Function ( $T_1^4(I^4)$ ) = 0.7795

$$W_{li}^4 = 0.2500 \ 0 \ 1.0000 \ 1.0000 \ 1.0000 \ 1.0000 \ 1.0000 \ 1.0000 \ 1.0000 \ 1.0000 \ 1.0000 \ 0$$

เนื่องจาก  $T_1^4(I^4) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^4(t+1) = 0.2500 \ 0 \ 0.9500 \ 0.9625 \ 0.9500 \ 1.0000 \ 1.0000 \ 1.0000 \ 1.0000 \ 1.0000 \ 0.9500 \ 0$$

กรอบย่อยที่ 5: คำนวณหาค่า Choice Function ( $T_1^5(I^5)$ ) = 0.9266

$$W_{li}^5 = 0.7500 \ 1.0000 \ 1.0000 \ 0.7500 \ 1.0000 \ 0.7500 \ 1.0000 \ 1.0000 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500$$

เนื่องจาก  $T_1^5(I^5) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^5(t+1) = 0.7125 \ 0.9500 \ 0.9875 \ 0.7500 \ 0.9875 \ 0.7500 \ 0.9875 \ 0.9875 \ 0.7500 \ 0.7500 \ 0.7500 \\ 0.7500$$

กรอบย่อยที่ 6: คำนวณหาค่า Choice Function ( $T_1^6(I^6)$ ) = 1

$$W_{li}^6 = 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500$$

เนื่องจาก  $T_1^6(I^6) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^6(t+1) = 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \\ 0.7500$$

กรอบย่อยที่ 7: คำนวณหาค่า Choice Function ( $T_1^7(I^7)$ ) = 1

$$W_{li}^7 = 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500$$

เนื่องจาก  $T_1^7(I^7) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^7(t+1) = 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \\ 0.7500$$

กรอบย่อยที่ 8: คำนวณหาค่า Choice Function ( $T_1^8(I^8)$ ) = 0

$$W_{li}^8 = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$$

เนื่องจากขนาดของ  $W_{li}^8 = 0$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^8(t+1) = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$$

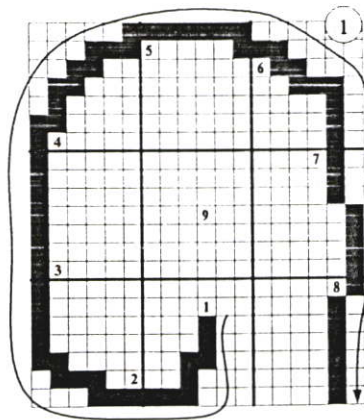


ตารางที่ 3.14 (ต่อ)

กรอบย่อยที่	Weight Vector ที่ชั้น Cluster Layer
8 ( $W_{28}$ )	0 0 0 0 0 0 0 0 0 0 0 0
9 ( $W_{29}$ )	0 0 0 0 0 0 0 0 0 0 0 0

เมื่อทำการพิจารณา Choice Function ตามสมการที่ 3.20 จะได้ค่า  $C_K$  ออกมา จากนั้นทำการเปรียบเทียบ  $C_K \geq \rho_{\text{cluster}}$  หรือไม่ ตามสมการที่ 3.22 ทั้งนี้  $C_1 = 0.7896$  ซึ่งทำให้สมการที่ 3.22 เป็นเท็จหรือเรียกว่าเป็นกรณี “Reset” แสดงว่าโหนดที่เลือกให้เป็นโหนดขณะ  $K=1$  ที่ Cluster Layer นั้นมีการจัดเข้ากลุ่มที่ไม่ถูกต้องตรงตาม กลุ่ม Output Class ของ Input Pattern จะมีการสร้างโหนดใหม่ขึ้นมาใน Cluster Layer โดยที่  $K = 2$  และทำการกำหนด Weight Vector ของโหนดใหม่ ( $W_{kh}$ ) ให้มีค่าเท่ากับค่า Weight Vector ( $W_j$ ) ที่ส่งต่อขึ้นมาจาก Hidden Layer ตามสมการที่ 3.25 หลังจากนั้นจะมีการสร้างการเชื่อมต่อจากโหนดใหม่ใน Cluster Layer ไปยังโหนดใน Output Layer ที่ตรงตาม Target Output ของ Input Pattern ที่ทำการเรียนรู้ครั้งนั้นคือ Target Output = “ภ”

การเรียนรู้ครั้งที่ 3 รับค่ารูปแบบตัวอักษรที่มีการกำหนด Input Vector ตามตารางที่ 3.7 โดยรูปแบบของตัวอักษรที่มีการแบ่งออกเป็น 9 กรอบย่อยแสดงดังรูปที่ 3.29



รูปที่ 3.29 แสดงทิศทางของเส้นตัวอักษรในการ Training ตัวที่ 3 ตามการจัดเรียงกรอบย่อยของภาพตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร

ทำการคำนวณหาค่า Choice Function ( $T_j^h(I^h)$ ) ตามสมการที่ 3.14 แล้วนำเอาค่า  $T_j^h(I^h)$  มาเปรียบเทียบกับค่า  $\rho_{\text{hidden}} = 0.75$  ที่กำหนดไว้แล้วทำการพิจารณาตรวจสอบค่า Vigilance Parameter ที่ Hidden Layer ( $\rho_{\text{hidden}}$ ) ตามสมการที่ 3.17 ดังนี้

กรอบย่อยที่ 1: คำนวณหาค่า Choice Function ( $T_2^1(I^1)$ ) = 0.7055

$$W_{2i}^1 = 0 \ 1.0000 \ 0 \ 0.2500 \ 0.2500 \ 0 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500$$

เนื่องจาก  $T_2^1(I^1) < \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันไม่มากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการสร้างโหนดใหม่ใน Hidden Layer และกำหนดให้มี Weight Vector ของโหนดใหม่ที่สร้างขึ้นมาเท่ากับค่า Input Vector ของช่วงกรอบส่วนย่อย h นั้น จะเรียกกรณีนี้ว่าการเกิด “Reset”

$$W_{3i}^1 = 0.5000 \ 0.5000 \ 0.6250 \ 0.5000 \ 0.5000 \ 0.5000 \ 0.5000 \ 0.5000 \ 0.5000 \ 0.5000 \ 0.5000 \ 0.5000$$

กรอบย่อยที่ 2: คำนวณหาค่า Choice Function ( $T_1^2(I^2)$ ) = 0.9374

$$W_{1i}^2 = 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500$$

เนื่องจาก  $T_1^2(I^2) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{1i}^2(t+1) = 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \\ 0.2500$$

กรอบย่อยที่ 3: คำนวณหาค่า Choice Function ( $T_1^3(I^3)$ ) = 0.7888

$$W_{1i}^3 = 0.2500 \ 0.2500 \ 0.2500 \ 0.4875 \ 0.4750 \ 0 \ 0.2500 \ 0 \ 0 \ 0.2500 \ 0 \ 0$$

เนื่องจาก  $T_1^3(I^3) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{1i}^3(t+1) = 0.2500 \ 0.2437 \ 0.2500 \ 0.4756 \ 0.4638 \ 0 \ 0.2500 \ 0 \ 0 \ 0.2500 \ 0 \ 0$$

กรอบย่อยที่ 4: คำนวณหาค่า Choice Function ( $T_1^4(I^4)$ ) = 0.7491

$$W_{1i}^4 = 0.2500 \ 0 \ 0.9500 \ 0.9625 \ 0.9500 \ 1.0000 \ 1.0000 \ 1.0000 \ 1.0000 \ 1.0000 \ 0.9500 \ 0$$

เนื่องจาก  $T_1^4(I^4) < \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันไม่มากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการสร้างโหนดใหม่ใน Hidden Layer และ

กำหนดให้มี Weight Vector ของโหนดใหม่ที่สร้างขึ้นมาเท่ากับค่า Input Vector ของช่วงกรอบ ส่วนย่อย h นั้น จะเรียกกรณีนี้ว่าการเกิด “Reset”

$$W_{2i}^4 = 0.2500 \ 0.1250 \ 0.1250 \ 0.1250 \ 0.1250 \ 0.1250 \ 0 \ 1.0000 \ 1.0000 \ 1.0000 \ 1.0000 \ 0$$

กรอบย่อยที่ 5: คำนวณหาค่า Choice Function ( $T_1^5(I^5)$ ) = 0.9233

$$W_{li}^5 = 0.7125 \ 0.9500 \ 0.9875 \ 0.7500 \ 0.9875 \ 0.7500 \ 0.9875 \ 0.9875 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500$$

เนื่องจาก  $T_1^5(I^5) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^5(t+1) = 0.6769 \ 0.9500 \ 0.9875 \ 0.7500 \ 0.9381 \ 0.7500 \ 0.9819 \ 0.9819 \ 0.7500 \ 0.7500 \ 0.7500 \\ 0.7500$$

กรอบย่อยที่ 6: คำนวณหาค่า Choice Function ( $T_1^6(I^6)$ ) = 0.9932

$$W_{li}^6 = 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500$$

เนื่องจาก  $T_1^6(I^6) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^6(t+1) = 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \\ 0.7500$$

กรอบย่อยที่ 7: คำนวณหาค่า Choice Function ( $T_1^7(I^7)$ ) = 0.9951

$$W_{li}^7 = 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500$$

เนื่องจาก  $T_1^7(I^7) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^7(t+1) = 0.7500 \ 0.7500 \ 0.7500 \ 0.7437 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \\ 0.7500$$

กรอบย่อยที่ 8: คำนวณหาค่า Choice Function ( $T_1^8(I^8)$ ) = 0

$$W_{li}^8 = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$$

เนื่องจาก  $W_{li}^8 = 0$  แต่ค่า Input Vector มีค่าไม่เท่ากับ 0 ดังนั้นแสดงว่า Input นั้นจะต้องทำการสร้างโหนดใหม่ใน Hidden Layer และกำหนดให้มี Weight Vector ของโหนดใหม่ที่สร้างขึ้นมามีค่าเท่ากับค่า Input Vector ของช่วงกรอบส่วนย่อย  $h$  นั้น จะเรียกกรณีนี้ว่าการเกิด “Reset”

$$W_{2i}^8 = 0.6250 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500$$

กรอบย่อยที่ 9: กำหนดค่า Choice Function ( $T_1^9(I^9)$ ) = 0

$$W_{li}^9 = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$$

เนื่องจากขนาดของ  $W_{li}^9 = 0$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชะนั้น จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^9(t+1) = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$$

จากตารางที่ 3.15 แสดงค่า Weight Vector ที่ชั้น Hidden Layer ในการเรียนรู้ครั้งที่ 3 และตารางที่ 3.16 แสดงค่า Weight Vector ที่ชั้น Cluster Layer ในการเรียนรู้ครั้งที่ 3

ตารางที่ 3.15 ค่า Weight Vector ที่ชั้น Hidden Layer ในการเรียนรู้ครั้งที่ 3

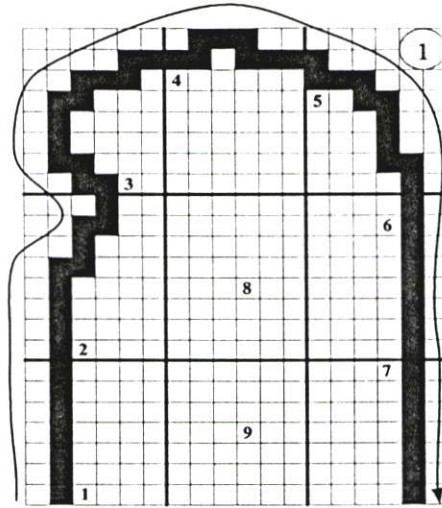
กรอบย่อยที่	Weight Vector ที่ชั้น Hidden Layer
1 ( $W_{3i}^1$ )	0.5000 0.5000 0.6250 0.5000 0.5000 0.5000 0.5000 0.5000 0.5000 0.5000 0.5000 0.5000
2 ( $W_{li}^2$ )	0.2500 0.2500 0.2500 0.2500 0.2500 0.2500 0 0.2500 0.2500 0.2500 0.2500 0.2500
3 ( $W_{li}^3$ )	0.2500 0.2437 0.2500 0.4756 0.4638 0 0.2500 0 0 0.2500 0 0
4 ( $W_{2i}^4$ )	0.2500 0.1250 0.1250 0.1250 0.1250 0.1250 0 1.0000 1.0000 1.0000 1.0000 0
5 ( $W_{li}^5$ )	0.6769 0.9500 0.9875 0.7500 0.9381 0.7500 0.9819 0.9819 0.7500 0.7500 0.7500 0.7500
6 ( $W_{li}^6$ )	0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500
7 ( $W_{li}^7$ )	0.7500 0.7500 0.7500 0.7437 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500
8 ( $W_{2i}^8$ )	0.6250 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.75 0.75 0.75
9 ( $W_{li}^9$ )	0 0 0 0 0 0 0 0 0 0 0 0

ตารางที่ 3.16 ค่า Weight Vector ที่ชั้น Cluster Layer ในการเรียนรู้ครั้งที่ 3

กรอบย่อยที่	Weight Vector ที่ชั้น Cluster Layer
1 ( $W_{31}$ )	0.5000 0.5000 0.6250 0.5000 0.5000 0.5000 0.5000 0.5000 0.5000 0.5000 0.5000 0.5000
2 ( $W_{32}$ )	0.2500 0.2500 0.2500 0.2500 0.2500 0.2500 0 0.2500 0.2500 0.2500 0.2500 0.2500
3 ( $W_{33}$ )	0.2500 0.2437 0.2500 0.4756 0.4638 0 0.2500 0 0 0.2500 0 0
4 ( $W_{34}$ )	0.2500 0.1250 0.1250 0.1250 0.1250 0.1250 0 1.0000 1.0000 1.0000 1.0000 0
5 ( $W_{35}$ )	0.6769 0.9500 0.9875 0.7500 0.9381 0.7500 0.9819 0.9819 0.7500 0.7500 0.7500 0.7500
6 ( $W_{36}$ )	0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500
7 ( $W_{37}$ )	0.7500 0.7500 0.7500 0.7437 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500
8 ( $W_{38}$ )	0.6250 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500
9 ( $W_{39}$ )	0 0 0 0 0 0 0 0 0 0 0

เมื่อทำการพิจารณา Choice Function ตามสมการที่ 3.20 จะได้ค่า  $C_K$  ออกมา จากนั้นทำการเปรียบเทียบ  $C_K \geq \rho_{\text{cluster}}$  หรือไม่ ตามสมการที่ 3.22 ทั้งนี้  $C_1 = 0.8134$  ซึ่งทำให้สมการที่ 3.22 เป็นเท็จหรือเรียกว่าเป็นกรณี “Reset” แสดงว่าโหนดที่เลือกให้เป็นโหนดชนะ  $K=1$  ที่ Cluster Layer นั้นมีการจัดเข้ากลุ่มที่ไม่ถูกต้องตรงตาม กลุ่ม Output Class ของ Input Pattern จะมีการสร้างโหนดใหม่ขึ้นมาใน Cluster Layer โดยที่  $K = 3$  และทำการกำหนด Weight Vector ของโหนดใหม่ ( $W_{kh}$ ) ให้มีค่าเท่ากับค่า Weight Vector ( $W_j$ ) ที่ส่งต่อขึ้นมาจาก Hidden Layer ตามสมการที่ 3.25 หลังจากนั้นจะมีการสร้างการเชื่อมต่อจากโหนดใหม่ใน Cluster Layer ไปยังโหนดใน Output Layer ที่ตรงตาม Target Output ของ Input Pattern ที่ทำการเรียนรู้ครั้งนั้นคือ Target Output = “ถ”

การเรียนรู้ครั้งที่ 4 รับค่ารูปแบบตัวอักษรที่มีการกำหนด Input Vector ตามตารางที่ 3.8 โดยรูปแบบของตัวอักษรที่มีการแบ่งออกเป็น 9 กรอบย่อยแสดงดังรูปที่ 3.30



รูปที่ 3.30 แสดงทิศทางของเส้นตัวอักษรในการ Training ตัวที่ 4 ตามการจัดเรียงกรอบย่อยของ ภาพตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร

ทำการคำนวณหาค่า Choice Function ( $T_j^h(I^h)$ ) ตามสมการที่ 3.14 แล้วนำเอาค่า  $T_j^h(I^h)$  มาเปรียบเทียบกับค่า  $\rho_{\text{hidden}} = 0.75$  ที่กำหนดไว้ แล้วทำการพิจารณาตรวจสอบค่า Vigilance Parameter ที่ Hidden Layer ( $\rho_{\text{hidden}}$ ) ตามสมการที่ 3.17 ดังนี้

กรอบย่อยที่ 1: คำนวณหาค่า Choice Function ( $T_1^1(I^1)$ ) = 0.9608

$$W_{li}^1 = 0.2500 \ 0.2500 \ 0 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500$$

เนื่องจาก  $T_1^1(I^1) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^1(t+1) = 0.2500 \ 0.2500 \ 0 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.25 \ 0.25 \ 0.25$$

กรอบย่อยที่ 2: คำนวณหาค่า Choice Function ( $T_1^2(I^2)$ ) = 0.8704

$$W_{li}^2 = 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500$$

เนื่องจาก  $T_1^2(I^2) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”



เนื่องจาก  $T_1^6(I^6) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{ii}^6(t+1) = 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500$$

กรอบย่อยที่ 7: คำนวณค่า Choice Function ( $T_1^7(I^7)$ ) = 1

$$W_{ii}^7 = 0.7500 \ 0.7500 \ 0.7500 \ 0.7437 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500$$

เนื่องจาก  $T_1^7(I^7) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{ii}^7(t+1) = 0.7500 \ 0.7500 \ 0.7500 \ 0.7437 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500$$

กรอบย่อยที่ 8: คำนวณค่า Choice Function ( $T_1^8(I^8)$ ) = 0

$$W_{ii}^8 = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$$

เนื่องจากขนาดของ  $W_{ii}^8 = 0$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{ii}^8(t+1) = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$$

กรอบย่อยที่ 9: คำนวณค่า Choice Function ( $T_1^9(I^9)$ ) = 0

$$W_{ii}^9 = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$$

เนื่องจากขนาดของ  $W_{ii}^9 = 0$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{ii}^9(t+1) = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$$

จากตารางที่ 3.17 แสดงค่า Weight Vector ที่ชั้น Hidden Layer ในการเรียนรู้ครั้งที่ 4 และ ตารางที่ 3.18 แสดงค่า Weight Vector ที่ชั้น Cluster Layer ในการเรียนรู้ครั้งที่ 4

ตารางที่ 3.17 ค่า Weight Vector ที่ชั้น Hidden Layer ในการเรียนรู้ครั้งที่ 4

กรอบย่อยที่	Weight Vector ที่ชั้น Hidden Layer
1 ( $W_{li}^1$ )	0.2500 0.2500 0 0.2500 0.2500 0.2500 0.2500 0.2500 0.2500 0.2500 0.2500 0.2500 0.2500
2 ( $W_{li}^2$ )	0.2500 0.2500 0.2500 0.2500 0.2375 0.2500 0 0.2375 0.2500 0.2500 0.2500 0.2500 0.2500
3 ( $W_{2i}^3$ )	0.5000 0.2500 0.5000 0.2500 0.2500 0.2500 0 0.2500 0 0 0.2500 0 0
4 ( $W_{li}^4$ )	0.2375 0 0.9025 0.9144 0.9400 1.0000 1.0000 1.0000 1.0000 1.0000 0.9500 0 0
5 ( $W_{li}^5$ )	0.6769 0.9500 0.9875 0.7500 0.9381 0.7500 0.9703 0.9703 0.7500 0.7500 0.7500 0.7500 0.7500
6 ( $W_{li}^6$ )	0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500
7 ( $W_{li}^7$ )	0.7500 0.7500 0.7500 0.7437 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500
8 ( $W_{li}^8$ )	0 0 0 0 0 0 0 0 0 0 0 0 0
9 ( $W_{li}^9$ )	0 0 0 0 0 0 0 0 0 0 0 0 0

ตารางที่ 3.18 ค่า Weight Vector ที่ชั้น Cluster Layer ในการเรียนรู้ครั้งที่ 4

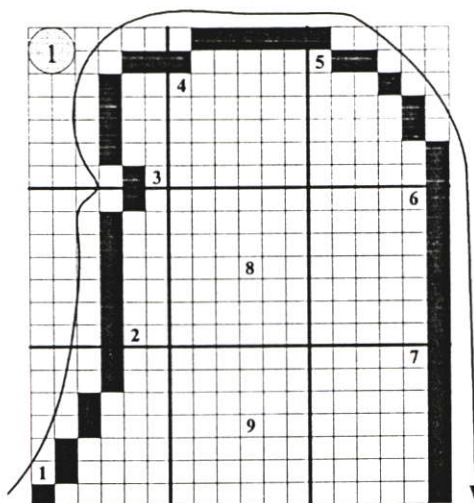
กรอบย่อยที่	Weight Vector ที่ชั้น Cluster Layer
1 ( $W_{11}$ )	0.2500 0.2500 0 0.2500 0.2500 0.2500 0.2500 0.2500 0.2500 0.2500 0.2500 0.2500 0.2500
2 ( $W_{12}$ )	0.2500 0.2500 0.2500 0.2500 0.2375 0.2500 0 0.2375 0.2500 0.2500 0.2500 0.2500 0.2500
3 ( $W_{13}$ )	0.2500 0.2500 0.2500 0.4875 0.4875 0 0.2375 0 0 0.2375 0 0 0
4 ( $W_{14}$ )	0.2375 0 0.9025 0.9144 0.9400 1.0000 1.0000 1.0000 1.0000 1.0000 0.9500 0 0
5 ( $W_{15}$ )	0.6769 0.9500 0.9875 0.7500 0.9381 0.7500 0.9703 0.9703 0.7500 0.7500 0.7500 0.7500 0.7500
6 ( $W_{16}$ )	0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500
7 ( $W_{17}$ )	0.7500 0.7500 0.7500 0.7437 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500

ตารางที่ 3.18 (ต่อ)

กรอบย่อยที่	Weight Vector ที่ชั้น Cluster Layer
8 ( $W_{18}$ )	0 0 0 0 0 0 0 0 0 0 0 0
9 ( $W_{19}$ )	0 0 0 0 0 0 0 0 0 0 0 0

เมื่อทำการพิจารณา Choice Function ตามสมการที่ 3.20 จะได้ค่า  $C_K$  ออกมา จากนั้นทำการเปรียบเทียบ  $C_K \geq \rho_{cluster}$  หรือไม่ ตามสมการที่ 3.22 ทั้งนี้  $C_1 = 0.9448$  ซึ่งทำให้สมการที่ 3.22 เป็นจริงหรือเรียกว่าเกิดกรณี “Resonance” แสดงว่า Weight Vector ( $W_J$ ) นั้นมีความคล้ายคลึงกับ Weight Vector ( $W_k$ ) ของโหนดที่เลือกให้เป็นโหนดที่ชนะคือโหนด  $K=1$  ใน Cluster Layer และตรงกับกลุ่มผลลัพธ์ของ Input Pattern นอกจากนี้จะพิจารณาว่า  $W_{2i}^3$  นั้นเป็นค่า Weight Vector ที่ส่งค่ามาจากโหนด  $J=2$  ที่มีการสร้างขึ้นใหม่ใน Hidden Layer ก็จะมีการปรับค่า Weight Vector ( $W_{13}(t+1)$ ) ตามสมการที่ 3.23 หลังจากนั้นจะมีการสร้างการเชื่อมต่อจากโหนด  $K=1$  ใน Cluster Layer ไปยังโหนดใน Output Layer ที่ตรงตาม Target Output ของ Input Pattern ที่ทำการเรียนรู้ครั้งนั้นคือ Target Output = “ก”

การเรียนรู้ครั้งที่ 5 รับค่ารูปแบบตัวอักษรที่มีการกำหนด Input Vector ตามตารางที่ 3.9 โดยรูปแบบของตัวอักษรที่มีการแบ่งออกเป็น 9 กรอบย่อยแสดงดังรูปที่ 3.31



รูปที่ 3.31 แสดงทิศทางของเส้นตัวอักษรในการ Training ตัวที่ 5 ตามการจัดเรียงกรอบย่อยของภาพตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร

ทำการคำนวณหาค่า Choice Function ( $T_j^h(I^h)$ ) ตามสมการที่ 3.14 แล้วนำเอาค่า  $T_j^h(I^h)$  มาเปรียบเทียบกับค่า  $\rho_{\text{hidden}} = 0.75$  ที่กำหนดไว้แล้วทำการพิจารณาตรวจสอบค่า Vigilance Parameter ที่ Hidden Layer ( $\rho_{\text{hidden}}$ ) ตามสมการที่ 3.17 ดังนี้

กรอบย่อยที่ 1: คำนวณหาค่า Choice Function ( $T_2^1(I^1)$ ) = 0.9685

$$W_{2i}^1 = 0 \ 1.0000 \ 0 \ 0.2500 \ 0.2500 \ 0 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500$$

เนื่องจาก  $T_2^1(I^1) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{2i}^1(t+1) = 0 \ 0.9625 \ 0 \ 0.2500 \ 0.2437 \ 0 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500$$

กรอบย่อยที่ 2: คำนวณหาค่า Choice Function ( $T_1^2(I^2)$ ) = 0.9480

$$W_{1i}^2 = 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2375 \ 0.2500 \ 0 \ 0.2375 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \\ 0.2500$$

เนื่องจาก  $T_1^2(I^2) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{1i}^2(t+1) = 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2375 \ 0.2437 \ 0 \ 0.2375 \ 0.2500 \ 0.2500 \ 0.2500 \\ 0.2500 \ 0.2500$$

กรอบย่อยที่ 3: คำนวณหาค่า Choice Function ( $T_2^3(I^3)$ ) = 0.8093

$$W_{2i}^3 = 0.5000 \ 0.2500 \ 0.5000 \ 0.2500 \ 0.2500 \ 0.2500 \ 0 \ 0.2500 \ 0 \ 0 \ 0.2500 \ 0 \ 0$$

เนื่องจาก  $T_2^3(I^3) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{2i}^3(t+1) = 0.4937 \ 0.2500 \ 0.4875 \ 0.2500 \ 0.2437 \ 0.2375 \ 0 \ 0.2375 \ 0 \ 0 \ 0.2500 \ 0 \ 0$$

กรอบย่อยที่ 4: คำนวณหาค่า Choice Function ( $T_1^4(I^4)$ ) = 0.9473

$$W_{1i}^4 = 0.2375 \ 0 \ 0.9025 \ 0.9144 \ 0.9400 \ 1.0000 \ 1.0000 \ 1.0000 \ 1.0000 \ 1.0000 \ 0.9500 \ 0 \ 0$$

เนื่องจาก  $T_1^4(I^4) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^4(t+1) = 0.2319 \ 0 \ 0.9025 \ 0.9144 \ 0.9400 \ 1.0000 \ 1.0000 \ 1.0000 \ 1.0000 \ 1.0000 \ 0.9500 \ 0 \ 0$$

กรอบย่อยที่ 5: คำนวณหาค่า Choice Function ( $T_1^5(I^5)$ ) = 0.9880

$$W_{li}^5 = 0.6769 \ 0.9500 \ 0.9875 \ 0.7500 \ 0.9381 \ 0.7500 \ 0.9703 \ 0.9703 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \\ 0.7500$$

เนื่องจาก  $T_1^5(I^5) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^5(t+1) = 0.6769 \ 0.9500 \ 0.9819 \ 0.7500 \ 0.9287 \ 0.7500 \ 0.9593 \ 0.9593 \ 0.7500 \ 0.7500 \ 0.7500 \\ 0.7500 \ 0.7500$$

กรอบย่อยที่ 6: คำนวณหาค่า Choice Function ( $T_1^6(I^6)$ ) = 1

$$W_{li}^6 = 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \\ 0.7500$$

เนื่องจาก  $T_1^6(I^6) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^6(t+1) = 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \\ 0.7500 \ 0.7500$$

กรอบย่อยที่ 7: คำนวณหาค่า Choice Function ( $T_1^7(I^7)$ ) = 1

$$W_{li}^7 = 0.7500 \ 0.7500 \ 0.7500 \ 0.7437 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \\ 0.7500$$

เนื่องจาก  $T_1^7(I^7) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^7(t+1) = 0.7500 \ 0.7500 \ 0.7500 \ 0.7437 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \\ 0.7500 \ 0.7500$$

กรอบย่อยที่ 8: กำหนดค่า Choice Function ( $T_1^8(I^8) = 0$ )

$$W_{li}^8 = 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$

เนื่องจากขนาดของ  $W_{li}^8 = 0$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^8(t+1) = 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$

กรอบย่อยที่ 9: กำหนดค่า Choice Function ( $T_1^9(I^9) = 0$ )

$$W_{li}^9 = 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$

เนื่องจากขนาดของ  $W_{li}^9 = 0$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^9(t+1) = 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$

จากตารางที่ 3.19 แสดงค่า Weight Vector ที่ชั้น Hidden Layer ในการเรียนรู้ครั้งที่ 5 และตารางที่ 3.20 แสดงค่า Weight Vector ที่ชั้น Cluster Layer ในการเรียนรู้ครั้งที่ 5

ตารางที่ 3.19 ค่า Weight Vector ที่ชั้น Hidden Layer ในการเรียนรู้ครั้งที่ 5

กรอบย่อยที่	Weight Vector ที่ชั้น Hidden Layer
1 ( $W_{21}^1$ )	0 0.9625 0 0.2500 0.2437 0 0.2500 0.2500 0.2500 0.2500 0.2500 0.2500
2 ( $W_{li}^2$ )	0.2500 0.2500 0.2500 0.2500 0.2375 0.2437 0 0.2375 0.2500 0.2500 0.2500 0.2500 0.2500
3 ( $W_{21}^3$ )	0.4937 0.2500 0.4875 0.2500 0.2437 0.2375 0 0.2375 0 0 0.2500 0 0
4 ( $W_{li}^4$ )	0.2319 0 0.9025 0.9144 0.9400 1.0000 1.0000 1.0000 1.0000 1.0000 0.9500 0 0
5 ( $W_{li}^5$ )	0.6769 0.9500 0.9819 0.7500 0.9287 0.7500 0.9593 0.9593 0.7500 0.7500 0.7500 0.7500 0.7500
6 ( $W_{li}^6$ )	0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500
7 ( $W_{li}^7$ )	0.7500 0.7500 0.7500 0.7437 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500
8 ( $W_{li}^8$ )	0 0 0 0 0 0 0 0 0 0 0 0 0

ตารางที่ 3.19 (ต่อ)

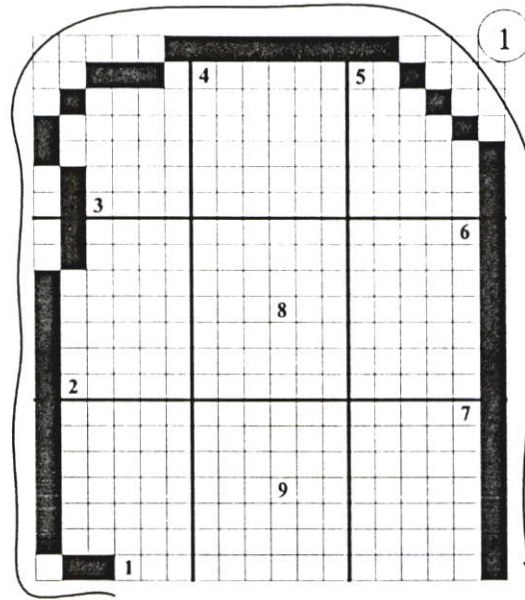
กรอบย่อยที่	Weight Vector ที่ชั้น Hidden Layer
9 ( $W_{li}^9$ )	0 0 0 0 0 0 0 0 0 0 0 0 0

ตารางที่ 3.20 ค่า Weight Vector ที่ชั้น Cluster Layer ในการเรียนรู้ครั้งที่ 5

กรอบย่อยที่	Weight Vector ที่ชั้น Cluster Layer
1 ( $W_{41}$ )	0 0.9625 0 0.2500 0.2437 0 0.2500 0.2500 0.2500 0.2500 0.2500 0.2500
2 ( $W_{42}$ )	0.2500 0.2500 0.2500 0.2500 0.2375 0.2437 0 0.2375 0.2500 0.2500 0.2500 0.2500 0.2500
3 ( $W_{43}$ )	0.4937 0.2500 0.4875 0.2500 0.2437 0.2375 0 0.2375 0 0 0.2500 0 0
4 ( $W_{44}$ )	0.2319 0 0.9025 0.9144 0.9400 1.0000 1.0000 1.0000 1.0000 1.0000 0.9500 0 0
5 ( $W_{45}$ )	0.6769 0.9500 0.9819 0.7500 0.9287 0.7500 0.9593 0.9593 0.7500 0.7500 0.7500 0.7500 0.7500
6 ( $W_{46}$ )	0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500
7 ( $W_{47}$ )	0.7500 0.7500 0.7500 0.7437 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500
8 ( $W_{48}$ )	0 0 0 0 0 0 0 0 0 0 0 0 0
9 ( $W_{49}$ )	0 0 0 0 0 0 0 0 0 0 0 0 0

เมื่อทำการพิจารณา Choice Function ตามสมการที่ 3.20 จะได้ค่า  $C_K$  ออกมาจากนั้นทำการเปรียบเทียบ  $C_K \geq \rho_{cluster}$  หรือไม่ ตามสมการที่ 3.22 ทั้งนี้  $C_1 = 0.8319$  ซึ่งทำให้สมการที่ 3.22 เป็นเท็จหรือเรียกว่าเป็นกรณี “Reset” แสดงว่าโหนดที่เลือกให้เป็นโหนดชนะ  $K=1$  ที่ Cluster Layer นั้นมีการจัดเข้ากลุ่มที่ไม่ถูกต้องตรงตาม กลุ่ม Output Class ของ Input Pattern จะมีการสร้างโหนดใหม่ขึ้นมาใน Cluster Layer โดยที่  $K = 4$  และทำการกำหนด Weight Vector ของโหนดใหม่ ( $W_{kh}$ ) ให้มีค่าเท่ากับค่า Weight Vector ( $W_j$ ) ที่ส่งต่อขึ้นมาจาก Hidden Layer ตามสมการที่ 3.25 หลังจากนั้นจะมีการสร้างการเชื่อมต่อจากโหนดใหม่ใน Cluster Layer ไปยังโหนดใน Output Layer ที่ตรงตาม Target Output ของ Input Pattern ที่ทำการเรียนรู้ครั้งนั้นคือ Target Output = “ภ”

การเรียนรู้ครั้งที่ 6 รับค่ารูปแบบตัวอักษรที่มีการกำหนด Input Vector ตามตารางที่ 3.10 โดยรูปแบบของตัวอักษรที่มีการแบ่งออกเป็น 9 กรอบย่อยแสดงดังรูปที่ 3.32



รูปที่ 3.32 แสดงทิศทางของเส้นตัวอักษรในการ Training ตัวที่ 6 ตามการจัดเรียงกรอบย่อยของ ภาพตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร

ทำการคำนวณหาค่า Choice Function ( $T_j^h(I^h)$ ) ตามสมการที่ 3.14 แล้วนำเอาค่า  $T_j^h(I^h)$  มาเปรียบเทียบกับค่า  $\rho_{\text{hidden}} = 0.75$  ที่กำหนดไว้ แล้วทำการพิจารณาตรวจสอบค่า Vigilance Parameter ที่ Hidden Layer ( $\rho_{\text{hidden}}$ ) ตามสมการที่ 3.17 ดังนี้

กรอบย่อยที่ 1: คำนวณหาค่า Choice Function ( $T_3^1(I^1)$ ) = 0.9624

$$W_{3i}^1 = 0.5000 \ 0.5000 \ 0.6250 \ 0.5000 \ 0.5000 \ 0.5000 \ 0.5000 \ 0.5000 \ 0.5000 \ 0.5000 \ 0.5000 \ 0.5000$$

เนื่องจาก  $T_3^1(I^1) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชะนั้น จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{3i}^1(t+1) = 0.5000 \ 0.4937 \ 0.6062 \ 0.4875 \ 0.4875 \ 0.4875 \ 0.4875 \ 0.4875 \ 0.4875 \ 0.4875 \ 0.4875 \ 0.4875$$

กรอบย่อยที่ 2: คำนวณหาค่า Choice Function ( $T_1^2(I^2)$ ) = 0.9500

$$W_{1i}^2 = 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2375 \ 0.2437 \ 0 \ 0.2375 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2500$$

$$0.2500$$

เนื่องจาก  $T_1^2(I^2) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{ji}^2(t+1) = 0.2500 \ 0.2500 \ 0.2500 \ 0.2500 \ 0.2319 \ 0.2437 \ 0 \ 0.2375 \ 0.2500 \ 0.2500 \ 0.2500 \\ 0.2500 \ 0.2500$$

กรอบย่อยที่ 3: คำนวณหาค่า Choice Function ( $T_2^3(I^3)$ ) = 0.8396

$$W_{2i}^3 = 0.4937 \ 0.2500 \ 0.4875 \ 0.2500 \ 0.2437 \ 0.2375 \ 0 \ 0.2375 \ 0 \ 0 \ 0.2500 \ 0 \ 0$$

เนื่องจาก  $T_2^3(I^3) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{2i}^3(t+1) = 0.4816 \ 0.2500 \ 0.4756 \ 0.2437 \ 0.2378 \ 0.2256 \ 0 \ 0.2319 \ 0 \ 0 \ 0.2375 \ 0 \ 0$$

กรอบย่อยที่ 4: คำนวณหาค่า Choice Function ( $T_1^4(I^4)$ ) = 0.8936

$$W_{1i}^4 = 0.2375 \ 0 \ 0.9025 \ 0.9144 \ 0.9400 \ 1.0000 \ 1.0000 \ 1.0000 \ 1.0000 \ 1.0000 \ 0.9500 \ 0 \ 0$$

เนื่องจาก  $T_1^4(I^4) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{1i}^4(t+1) = 0.2256 \ 0 \ 0.9025 \ 0.9144 \ 0.9400 \ 1.0000 \ 1.0000 \ 1.0000 \ 0.9500 \ 0.9500 \ 0.9025 \ 0 \ 0$$

กรอบย่อยที่ 5: คำนวณหาค่า Choice Function ( $T_1^5(I^5)$ ) = 0.9683

$$W_{1i}^5 = 0.6769 \ 0.9500 \ 0.9819 \ 0.7500 \ 0.9287 \ 0.7500 \ 0.9593 \ 0.9593 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \\ 0.7500$$

เนื่องจาก  $T_1^5(I^5) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{1i}^5(t+1) = 0.6430 \ 0.9462 \ 0.9765 \ 0.7500 \ 0.9260 \ 0.7500 \ 0.9488 \ 0.9488 \ 0.7500 \ 0.7500 \ 0.7500 \\ 0.7500 \ 0.7500$$

กรอบย่อยที่ 6: คำนวณหาค่า Choice Function ( $T_1^6(I^6)$ ) = 1

$$W_{li}^6 = 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500$$

เนื่องจาก  $T_1^6(I^6) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^6(t+1) = 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500$$

กรอบย่อยที่ 7: คำนวณหาค่า Choice Function ( $T_1^7(I^7) = 1$ )

$$W_{li}^7 = 0.7500 \ 0.7500 \ 0.7500 \ 0.7437 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500$$

เนื่องจาก  $T_1^7(I^7) > \rho_{\text{hidden}}$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^7(t+1) = 0.7500 \ 0.7500 \ 0.7500 \ 0.7437 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500 \ 0.7500$$

กรอบย่อยที่ 8: คำนวณหาค่า Choice Function ( $T_1^8(I^8) = 0$ )

$$W_{li}^8 = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$$

เนื่องจากขนาดของ  $W_{li}^8 = 0$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^8(t+1) = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$$

กรอบย่อยที่ 9: คำนวณหาค่า Choice Function ( $T_1^9(I^9) = 0$ )

$$W_{li}^9 = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$$

เนื่องจากขนาดของ  $W_{li}^9 = 0$  แสดงว่า Input นั้นมีความคล้ายคลึงกันมากพอที่จะจัดเข้ารวมอยู่กับโหนดที่ชนะนั้นได้ ต่อจากนั้นจะทำการปรับ Weight Vector ของโหนดที่ชนะ จะเรียกกรณีนี้ว่าการเกิด “Resonance”

$$W_{li}^9(t+1) = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$$

จากตารางที่ 3.21 แสดงค่า Weight Vector ที่ชั้น Hidden Layer ในการเรียนรู้ครั้งที่ 6 และ ตารางที่ 3.22 แสดงค่า Weight Vector ที่ชั้น Cluster Layer ในการเรียนรู้ครั้งที่ 6

ตารางที่ 3.21 ค่า Weight Vector ที่ชั้น Hidden Layer ในการเรียนรู้ครั้งที่ 6

กรอบย่อยที่	Weight Vector ที่ชั้น Hidden Layer
1 ( $W_{3i}^1$ )	0.5000 0.4937 0.6062 0.4875 0.4875 0.4875 0.4875 0.4875 0.4875 0.4875 0.4875 0.4875
2 ( $W_{li}^2$ )	0.2500 0.2500 0.2500 0.2500 0.2319 0.2437 0 0.2375 0.2500 0.2500 0.2500 0.2500 0.2500
3 ( $W_{2i}^3$ )	0.4816 0.2500 0.4756 0.2437 0.2378 0.2256 0 0.2319 0 0 0.2375 0 0
4 ( $W_{li}^4$ )	0.2256 0 0.9025 0.9144 0.9400 1.0000 1.0000 1.0000 0.9500 0.9500 0.9025 0 0
5 ( $W_{li}^5$ )	0.6430 0.9462 0.9765 0.7500 0.9260 0.7500 0.9488 0.9488 0.7500 0.7500 0.7500 0.7500 0.7500
6 ( $W_{li}^6$ )	0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500
7 ( $W_{li}^7$ )	0.7500 0.7500 0.7500 0.7437 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500
8 ( $W_{li}^8$ )	0 0 0 0 0 0 0 0 0 0 0 0 0
9 ( $W_{li}^9$ )	0 0 0 0 0 0 0 0 0 0 0 0 0

ตารางที่ 3.22 ค่า Weight Vector ที่ชั้น Cluster Layer ในการเรียนรู้ครั้งที่ 6

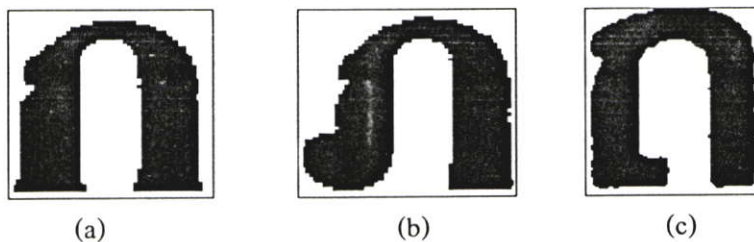
กรอบย่อยที่	Weight Vector ที่ชั้น Cluster Layer
1 ( $W_{31}$ )	0.5000 0.4937 0.6062 0.4875 0.4875 0.4875 0.4875 0.4875 0.4875 0.4875 0.4875 0.4875
2 ( $W_{32}$ )	0.2500 0.2500 0.2500 0.2500 0.2319 0.2437 0 0.2375 0.2500 0.2500 0.2500 0.2500 0.2500
3 ( $W_{33}$ )	0.2500 0.2437 0.2500 0.4640 0.4525 0 0.2375 0 0 0.2375 0 0 0
4 ( $W_{34}$ )	0.2488 0.1187 0.1250 0.1250 0.1250 0.1250 0 1.0000 0.9975 0.9975 0.9951 0 0
5 ( $W_{35}$ )	0.6430 0.9462 0.9765 0.7500 0.9260 0.7500 0.9488 0.9488 0.7500 0.7500 0.7500 0.7500 0.7500

ตารางที่ 3.22 (ต่อ)

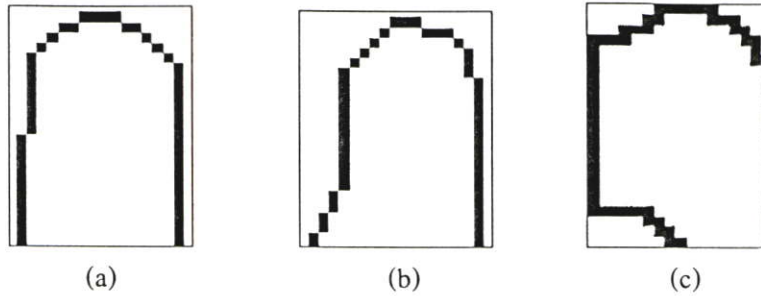
กรอบย่อยที่	Weight Vector ที่ชั้น Cluster Layer
6 ( $W_{36}$ )	0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500
7 ( $W_{37}$ )	0.7500 0.7500 0.7500 0.7437 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500 0.7500
8 ( $W_{38}$ )	0 0 0 0 0 0 0 0 0 0 0 0
9 ( $W_{39}$ )	0 0 0 0 0 0 0 0 0 0 0 0

เมื่อทำการพิจารณา Choice Function ตามสมการที่ 3.20 จะได้ค่า  $C_K$  ออกมา จากนั้นทำการเปรียบเทียบ  $C_K \geq \rho_{\text{cluster}}$  หรือไม่ ตามสมการที่ 3.22 ทั้งนี้  $C_3 = 0.9148$  ซึ่งทำให้สมการที่ 3.22 เป็นจริงหรือเรียกว่าเกิดกรณี “Resonance” แสดงว่า Weight Vector ( $W_j$ ) นั้นมีความคล้ายคลึงกับ Weight Vector ( $W_k$ ) ของโหนดที่เลือกให้เป็นโหนดที่ชนะคือโหนด  $K=3$  ใน Cluster Layer และตรงกับกลุ่มผลลัพธ์ของ Input Pattern นอกจากนี้จะพิจารณาว่า  $W_{2i}^3, W_{hi}^4$  นั้นเป็นค่า Weight Vector ที่ส่งค่ามาจากโหนดใหม่ใน Hidden Layer ก็จะมีการปรับค่า Weight Vector ( $W_{33}(t+1)$  และ  $W_{34}(t+1)$ ) ตามสมการที่ 3.23 หลังจากนั้นจะมีการสร้างการเชื่อมต่อจากโหนด  $K=3$  ใน Cluster Layer ไปยังโหนดใน Output Layer ที่ตรงตาม Target Output ของ Input Pattern ที่ทำการเรียนรู้ครั้งนั้นคือ Target Output = “ด”

3.3.2 ข้อมูลใน Testing Set มีทั้งหมด 3 รูปแบบ ได้แก่ ตัวอักษร “ก” “ภ” และ “ด” ตามลำดับดังรูปที่ 3.33



รูปที่ 3.33 รูปแบบข้อมูล Testing Set (a) ตัวอักษรตัวที่ 1 คือ “ก” (b) ตัวอักษรตัวที่ 2 คือ “ภ”  
(c) ตัวอักษรตัวที่ 3 คือ “ด”



รูปที่ 3.34 ข้อมูลภาพตัวอักษรใน Testing Set ที่มีการ Resize [29, 25] และทำการ Skeleton แล้ว  
(a) ตัวอักษรตัวที่ 1 คือ “ก” (b) ตัวอักษรตัวที่ 2 คือ “ภ” (c) ตัวอักษรตัวที่ 3 คือ “ถ”

ต่อจากนั้นเข้าสู่กระบวนการก่อนการรู้จำเริ่มจากการแบ่งออกเป็นกรอบตัวอักษรแต่ละตัว ดังรูปที่ 3.33 แล้วทำการลดขนาดให้เท่ากับ 25x29 Pixels จากนั้นทำการ Skeleton ตัวอักษรแต่ละตัว ที่รับเข้ามา หลังจากนั้นจะได้กรอบตัวอักษรที่มีการ Skeleton แล้วดังรูปที่ 3.34

ขั้นตอนต่อไปคือการแทนค่ารหัสแทนเส้นทางของภาพตัวอักษรใน Testing Set ตัวที่ 1, 2 และ 3 ตามตารางที่ 3.23, 3.24 และ 3.25 ตามลำดับ

ตารางที่ 3.23 การแทนค่ารหัสแทนเส้นทางของภาพตัวอักษรใน Testing Set ตัวที่ 1

กรอบย่อยที่	ค่าข้อมูล	Input Vector มีขนาดเท่ากับ 7
1 ( $I^1$ )	0.25 0.25 0.25 0.25 0.25 0.25 0.25	0.25 0.25 0.25 0.25 0.25 0.25 0.25
2 ( $I^2$ )	0.25 0.25 0.25 0.125 0.25 0.25 0.25	0.25 0.25 0.25 0.125 0.25 0.25 0.25
3 ( $I^3$ )	0.25 0.25 0.125 0.125 0.125 0 0.125	0.25 0.25 0.125 0.125 0.125 0 0.125
4 ( $I^4$ )	0 0 0 0.875 1	0 1 0 0.875 1 1 1
5 ( $I^5$ )	0.875 0.875 0.875 0.875 0.75 0.75	0.875 0.875 0.875 0.875 0.75 0.75 0.75
6 ( $I^6$ )	0.75 0.75 0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75
7 ( $I^7$ )	0.75 0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75
8 ( $I^8$ )	-	0 0 0 0 0 0 0
9 ( $I^9$ )	-	0 0 0 0 0 0 0

ตารางที่ 3.24 การแทนค่ารหัสแทนเส้นทางของภาพตัวอักษรใน Testing Set ตัวที่ 2

กรอบย่อยที่	ค่าข้อมูล	Input Vector มีขนาดเท่ากับ 8
1 ( $I^1$ )	0.125 0.25 0.125 0.25 0.125 0.25	0.125 0.25 0.125 0.25 0.125 0.25 0.25 0.25
2 ( $I^2$ )	0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25	0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
3 ( $I^3$ )	0.25 0.125 0.125 0.125	0.25 0.125 0.125 0.125 0.125 0.125 0.125 0.125
4 ( $I^4$ )	0.125 0.125 0 0 0.875 1	0.125 0.125 0 0 0.875 1 1 1
5 ( $I^5$ )	1 1 0.875 0.875 0.75 0.75 0.75 0.875	1 1 0.875 0.875 0.75 0.75 0.75 0.875
6 ( $I^6$ )	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
7 ( $I^7$ )	0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
8 ( $I^8$ )	-	0 0 0 0 0 0 0 0
9 ( $I^9$ )	-	0 0 0 0 0 0 0 0

ตารางที่ 3.25 การแทนค่ารหัสแทนเส้นทางของภาพตัวอักษรใน Testing Set ตัวที่ 3

กรอบย่อยที่	ค่าข้อมูล	Input Vector มีขนาดเท่ากับ 13
1 ( $I^1$ )	0.5 0.25 0.5 0.25 0.5 0.25 0.5	0.5 0.25 0.5 0.25 0.5 0.25 0.5 0.5 0.5 0.5 0.5 0.5 0.5
2 ( $I^2$ )	0.5 0.5 0.5 0.5 0.25 0.25 0.25 0.25 0.25	0.5 0.5 0.5 0.5 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
3 ( $I^3$ )	0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25	0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
4 ( $I^4$ )	0.25 0.25 0.25 0.25 0 0 0 0.25 0 0	0.25 0.25 0.25 0.25 0 1 0 0.25 0 1 1 1 0
5 ( $I^5$ )	0.25 0 0.25 0 0 0 0 0	0.25 0 0.25 0 1 1 1 1 1 1 1 1 0

ตารางที่ 3.25 (ต่อ)

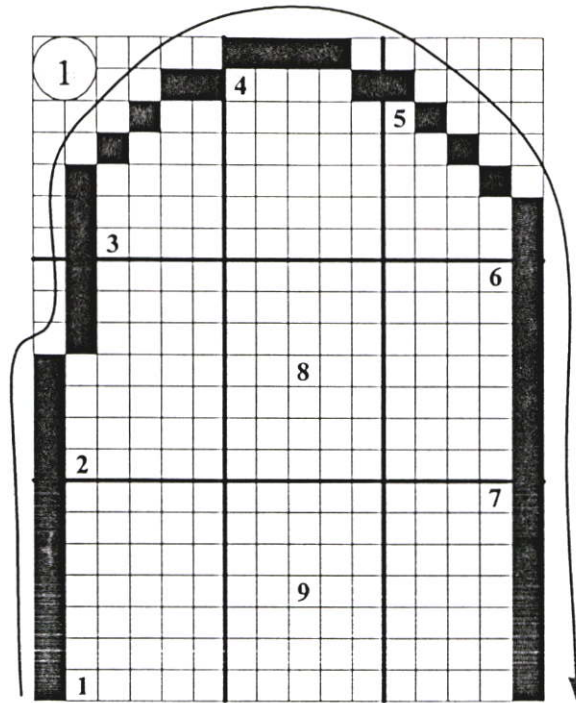
กรอบย่อยที่	ค่าข้อมูล	Input Vector มีขนาดเท่ากับ 13
6 ( $1^6$ )	0.75 1 1 0.75 1 0.75 1 0.75 0.75 1 0.75 0.75 0.75	0.75 1 1 0.75 1 0.75 1 0.75 0.75 1 0.75 0.75 0.75
7 ( $1^7$ )	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
8 ( $1^8$ )	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
9 ( $1^9$ )	-	0 0 0 0 0 0 0 0 0 0 0 0 0

จะสังเกตเห็นว่าทั้ง 3 รูปแบบภาพตัวอักษรใน Testing Set ที่รับเข้ามาเมื่อหาจุดศูนย์กลางของกรอบตัวอักษรแล้วอยู่ในกลุ่มระดับกลาง จากนั้นจะส่งรูปแบบตัวอักษรแต่ละตัวเข้าสู่ระบบการรู้จำแนวทางใหม่ในกระบวนการรู้จำตัวอักษรภาษาไทยด้วยวิธีใหม่ที่น่าสนใจเพื่อทดสอบระบบ

#### กระบวนการรู้จำในขั้นตอนของการ Testing

ในกระบวนการทดสอบระบบการรู้จำตัวอักษรที่ผ่านกระบวนการเรียนรู้ข้างต้นนี้เป็นการตรวจสอบว่าระบบที่ผ่านการเรียนรู้มาแล้วนั้นสามารถที่จะรู้จำผลลัพธ์ออกมาได้มากน้อยเพียงใด ซึ่งในการหาคำตอบของระบบทำได้โดยการคำนวณค่า Choice Function ( $T_j^h(I^h)$ ) ตามสมการที่ 3.14 แล้วนำเอาค่า  $T_j^h(I^h)$  ของโหนด  $J$  ที่ให้ค่ามากที่สุดออกมา และจะเลือกให้โหนด  $J$  นั้นเป็นโหนดที่ชนะ จากนั้นจะส่งต่อค่า Weight Vector ของโหนดที่ชนะใน Hidden Layer ต่อไปให้เป็นค่า Input Vector สำหรับ Cluster Layer แล้วก็ทำการคำนวณค่า Choice Function ( $C_k$ ) ตามสมการที่ 3.20 แล้วนำเอาค่า  $C_k$  ของโหนด  $K$  ที่ให้ค่ามากที่สุดออกมา และจะเลือกให้โหนด  $K$  นั้นเป็นโหนดที่ชนะ แล้วเปรียบเทียบกับคำตอบที่แท้จริงกับผลลัพธ์ที่ได้จากระบบ

การทดสอบครั้งที่ 1 รับค่ารูปแบบตัวอักษรที่มีการกำหนด Input Vector ตามตารางที่ 3.23 โดยรูปแบบของตัวอักษรที่มีการแบ่งออกเป็น 9 กรอบย่อยแสดงดังรูปที่ 3.35

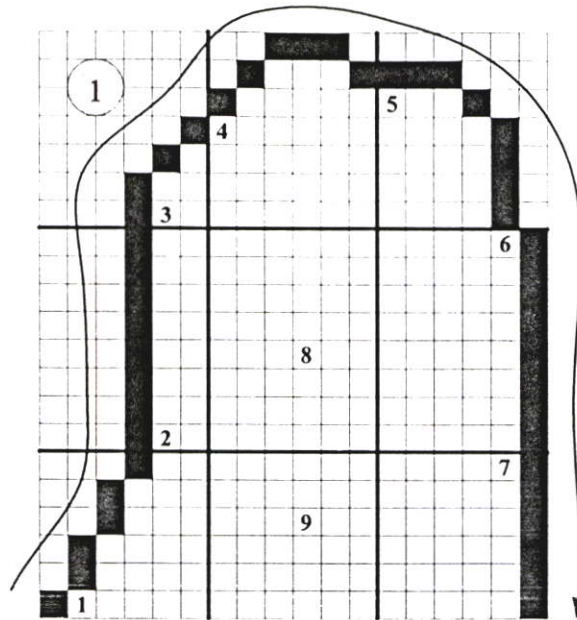


รูปที่ 3.35 แสดงทิศทางของเส้นตัวอักษรในการ Testing ตัวที่ 1 ตามการจัดเรียงกรอบย่อยของภาพ ตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร

- กรอบย่อยที่ 1: มีค่า Maximum Choice Function = 0.9979 ที่  $J = 1$   
 กรอบย่อยที่ 2: มีค่า Maximum Choice Function = 0.9477 ที่  $J = 1$   
 กรอบย่อยที่ 3: มีค่า Maximum Choice Function = 0.7626 ที่  $J = 2$   
 กรอบย่อยที่ 4: มีค่า Maximum Choice Function = 0.7464 ที่  $J = 1$   
 กรอบย่อยที่ 5: มีค่า Maximum Choice Function = 0.9894 ที่  $J = 1$   
 กรอบย่อยที่ 6: มีค่า Maximum Choice Function = 1 ที่  $J = 1$   
 กรอบย่อยที่ 7: มีค่า Maximum Choice Function = 1 ที่  $J = 1$   
 กรอบย่อยที่ 8: มีค่า Maximum Choice Function = 0 ที่  $J = 1$   
 กรอบย่อยที่ 9: มีค่า Maximum Choice Function = 0 ที่  $J = 1$

เมื่อทำการหาค่า Choice Function ที่ Cluster Layer จะพบว่า มีค่ามากที่สุดที่ได้คือ 0.9246 ที่  $K = 1$  ซึ่งทำให้ผลจากการรู้จำคือตัวอักษร “ก” ซึ่งระบบสามารถทำการรู้จำตัวอักษรได้ถูกต้อง

การทดสอบครั้งที่ 2 รับค่ารูปแบบตัวอักษรที่มีการกำหนด Input Vector ตามตารางที่ 3.24 โดยรูปแบบของตัวอักษรที่มีการแบ่งออกเป็น 9 กรอบย่อยแสดงดังรูปที่ 3.36



รูปที่ 3.36 แสดงทิศทางของเส้นตัวอักษรในการ Testing ตัวที่ 2 ตามการจัดเรียงกรอบย่อยของภาพ ตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร

กรอบย่อยที่ 1: มีค่า Maximum Choice Function = 0.9685 ที่  $J = 3$

กรอบย่อยที่ 2: มีค่า Maximum Choice Function = 0.9605 ที่  $J = 1$

กรอบย่อยที่ 3: มีค่า Maximum Choice Function = 0.8020 ที่  $J = 2$

กรอบย่อยที่ 4: มีค่า Maximum Choice Function = 0.7745 ที่  $J = 2$

กรอบย่อยที่ 5: มีค่า Maximum Choice Function = 0.9847 ที่  $J = 1$

กรอบย่อยที่ 6: มีค่า Maximum Choice Function = 1 ที่  $J = 1$

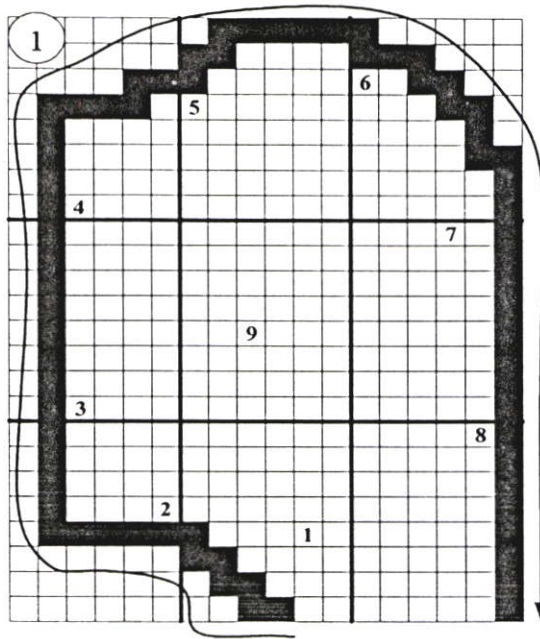
กรอบย่อยที่ 7: มีค่า Maximum Choice Function = 1 ที่  $J = 1$

กรอบย่อยที่ 8: มีค่า Maximum Choice Function = 0 ที่  $J = 1$

กรอบย่อยที่ 9: มีค่า Maximum Choice Function = 0 ที่  $J = 1$

เมื่อทำการหาค่า Choice Function ที่ Cluster Layer จะพบว่า มีค่ามากที่สุดที่ได้คือ 0.9190 ที่  $K = 4$  ซึ่งทำให้ผลจากการรู้จำคือตัวอักษร “ก” ซึ่งระบบสามารถทำการรู้จำตัวอักษรได้ถูกต้อง

การทดสอบครั้งที่ 3 รับค่ารูปแบบตัวอักษรที่มีการกำหนด Input Vector ตามตารางที่ 3.25 โดยรูปแบบของตัวอักษรที่มีการแบ่งออกเป็น 9 กรอบย่อยแสดงดังรูปที่ 3.37



รูปที่ 3.37 แสดงทิศทางของเส้นตัวอักษรในการ Testing ตัวที่ 3 ตามการจัดเรียงกรอบย่อยของภาพ  
ตัวอักษรตามลำดับของทิศทางเส้นตัวอักษรเพื่อกำหนดรหัสแทนเส้นตัวอักษร

กรอบย่อยที่ 1: มีค่า Maximum Choice Function = 0.9730 ที่ J = 3

กรอบย่อยที่ 2: มีค่า Maximum Choice Function = 0.4439 ที่ J = 1

กรอบย่อยที่ 3: มีค่า Maximum Choice Function = 0.7402 ที่ J = 2

กรอบย่อยที่ 4: มีค่า Maximum Choice Function = 0.7731 ที่ J = 1

กรอบย่อยที่ 5: มีค่า Maximum Choice Function = 0.8247 ที่ J = 1

กรอบย่อยที่ 6: มีค่า Maximum Choice Function = 0.9898 ที่ J = 1

กรอบย่อยที่ 7: มีค่า Maximum Choice Function = 1 ที่ J = 1

กรอบย่อยที่ 8: มีค่า Maximum Choice Function = 0.9990 ที่ J = 2

กรอบย่อยที่ 9: มีค่า Maximum Choice Function = 0 ที่ J = 1

เมื่อทำการหาค่า Choice Function ที่ Cluster Layer จะพบว่า มีค่ามากที่สุดที่ได้คือ 0.9639 ที่  
K = 3 ซึ่งทำให้ผลจากการรู้จำคือตัวอักษร "ถ" ซึ่งระบบสามารถทำการรู้จำตัวอักษรได้ถูกต้อง

## บทที่ 4

### ผลการทดลอง

บทนี้จะกล่าวถึงชุดข้อมูลที่ใช้ในการทดลอง และผลการทดลอง โดยจะเป็นการเปรียบเทียบวิธีใหม่ที่น่าเสนอกับ Fuzzy ARTMAP และซอฟต์แวร์ในการรู้จำที่ใช้งานกันอย่างแพร่หลายคือ AmThai และ ThaiOCR เพื่อทำการเปรียบเทียบประสิทธิภาพในการรู้จำตัวอักษรภาษาไทยในกลุ่มของตัวอักษรที่ปราศจากหัว ซึ่งเป็นงานวิจัยที่ทำการศึกษานี้

ในงานวิจัยครั้งนี้พัฒนาระบบการรู้จำตัวอักษรภาษาไทยโดยใช้โปรแกรม MATLAB 6.5 ในการทดลองทั้งหมด และมีการใช้โปรแกรม Adobe Photoshop CS2 ในการเตรียมข้อมูลก่อนการทดลอง ทั้งนี้ชุดข้อมูลที่ได้มาจากการสแกนภาพเอกสารและรูปถ่ายป้ายทะเบียนรถไทยด้วยเครื่อง XDA II และได้ทำการแบ่งกลุ่มการทดลองเป็น 2 ส่วนคือส่วนที่ 1 การทดลองกับชุดข้อมูลภาพเอกสาร และส่วนที่ 2 การทดลองกับชุดข้อมูลภาพป้ายทะเบียนรถไทย

#### 4.1 ชุดข้อมูลที่ใช้ในการทดลอง

ชุดข้อมูลที่ใช้ในงานวิจัยครั้งนี้แบ่งออกเป็น 2 กลุ่มชุดข้อมูลได้แก่

4.1.1 ชุดข้อมูลภาพเอกสาร เป็นภาพเอกสารที่ประกอบไปด้วยรูปแบบตัวอักษรทั้งหมด 39 รูปแบบตัวอักษร และแต่ละรูปแบบตัวอักษรจะมีทั้งหมด 80 อักขระได้แก่ พยัญชนะ สระวรรณยุกต์ และตัวเลขอารบิก แสดงตามตารางที่ 4.1

ตารางที่ 4.1 ตัวอย่างรูปแบบตัวอักษรไทยที่ไม่มีหัวที่ใช้ในงานวิจัยครั้งนี้มีทั้งหมด 39 ฟอนต์

ลำดับ	ชื่อรูปแบบตัวอักษร	ตัวอย่างรูปแบบตัวอักษร
1	Js Boaboon	ก ข ค ด ต บ ว จ ช ช ฅ ญ ฎ ฏ ร ก ฌ ฎ ฏ ต ฏ ก ร น บ ป พ ฟ พ ฟ ก ม ย ร ล ว ศ ส ห ฟ อ ฮ ฤ ฤ อะ อา อิ อี อี อี อู อุ เอ แอ อ่า โอ ใ โอ ไอ “ ‘ ๋ อี อี อี อี อี ำ ๐ 1 2 3 4 5 6 7 8 9
2	Js Chalit	ก ข ค ค ค ข ง จ ฅ ช ช ฅ ฌ ฎ ฏ ฌ ก ฌ ฌ ค ค ค ก ร บ บ ป ฝ ฝ พ ฝ ฝ ฝ ก บ ย ร ล ว ศ ส ห ฟ อ ส ฤ ฤ อะ อา อิ อี อี อี อู อุ เอ แอ อ่า ไอ ใ โอ ไอ “ ‘ ๋ อี อี อี อี อี อี ำ ๐ 1 2 3 4 5 6 7 8 9
3	Js Chanok	ก ข ค ค ค ม ง ฅ ฌ ฌ ฌ ฌ ฌ ฌ ฎ ฏ ฐ ฌ ฌ ฌ ฌ ฌ ฌ ฌ ป ฝ ฝ พ ฝ พ ม ย ร ล ว ศ ส ฌ ฌ ฌ ฌ ฌ ฌ ฌ ฌ ฌ ฌ ฌ ฌ อู เอ แอ อ่า โอ ใ โอ ไอ “ ‘ ๋ อี อี อี อี อี อี ำ ๐ 1 2 3 4 5 6 7 8 9





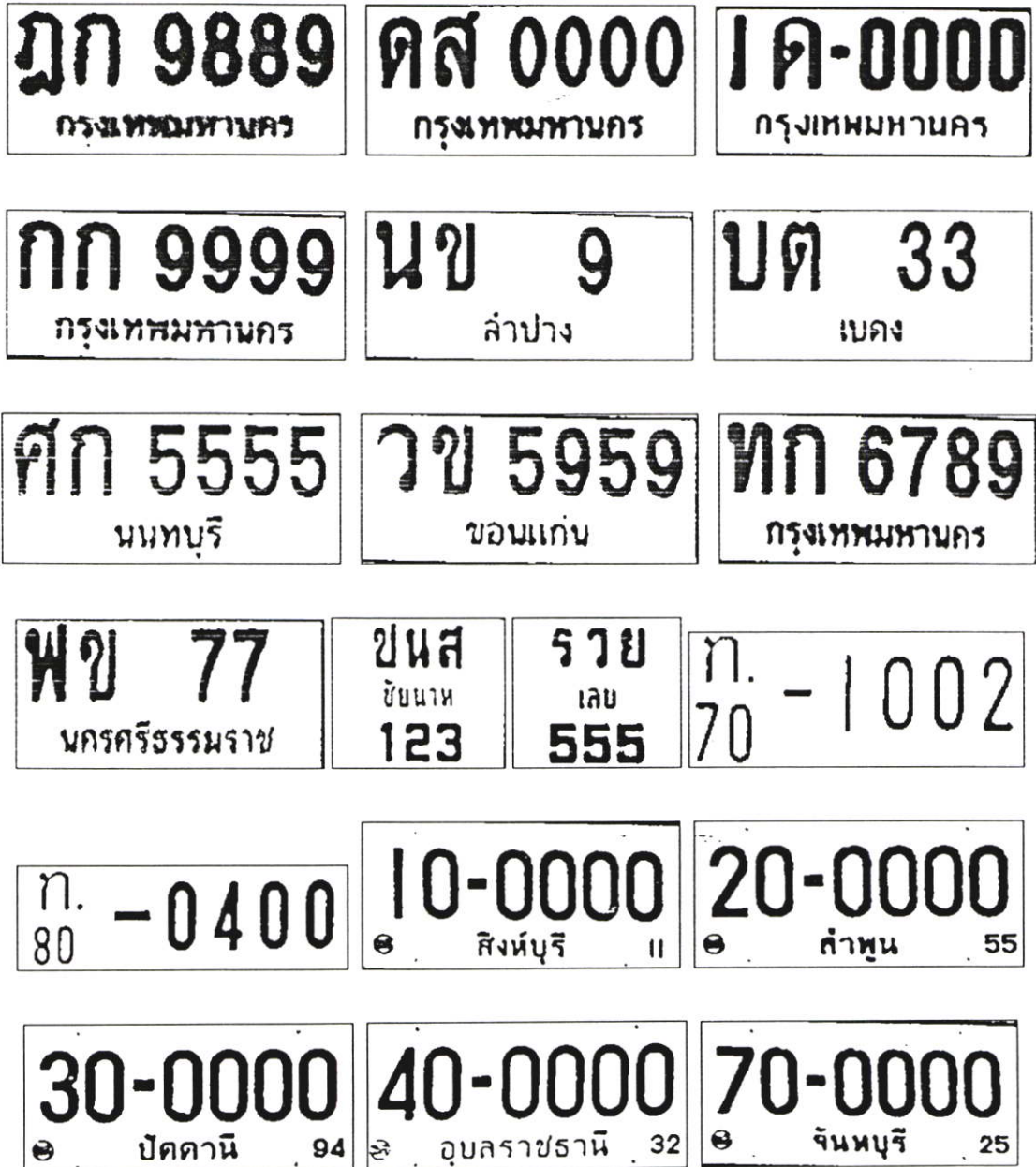






โดยชุดข้อมูลภาพเอกสารตัวอักษรพิมพ์นั้นจะได้รับการสแกนเข้าสู่เครื่องคอมพิวเตอร์ที่มีค่าความละเอียด 600 dpi จากนั้นเข้าสู่กระบวนการก่อนการรู้จำเพื่อให้ได้กรอบตัวอักษรออกมา แล้วจึงเข้าสู่กระบวนการรู้จำ

4.1.2 ชุดข้อมูลภาพป้ายทะเบียนรถไทย เป็นภาพป้ายทะเบียนรถไทยที่เก็บข้อมูลด้วยกล้องบนอุปกรณ์ XDA II โดยภาพป้ายทะเบียนประกอบไปด้วยป้ายทะเบียนรถยนต์ รถจักรยานยนต์ รถบรรทุก รถประจำทาง ทั้งหมด 506 ป้ายทะเบียน ยกตัวอย่างดังรูปที่ 4.1



รูปที่ 4.1 ตัวอย่างข้อมูลภาพป้ายทะเบียนรถไทยที่ใช้ในการทดลอง



รูปที่ 4.1 (ต่อ)

## 4.2 ผลการทดลอง

ผลการทดลองจากงานวิจัยนี้ได้ทำการเปรียบเทียบความสามารถของการรู้จำตัวอักษรพิมพ์ภาษาไทยโดยทดสอบชุดข้อมูลภาพเอกสารที่ประกอบไปด้วยรูปแบบตัวอักษร 39 รูปแบบรวมทั้งหมด 3,120 ตัวอักษร ด้วยโปรแกรม AmThai และ ThaiOCR และแสดงผลการทดสอบตามตารางที่ 4.2 นอกจากนี้ยังทำการทดสอบชุดข้อมูลภาพป้ายทะเบียนรถไทยที่ประกอบไปด้วยตัวอักษรไทย และตัวเลขอารบิกทั้งหมด 506 ป้ายทะเบียนซึ่งมีทั้งหมด 8,121 ตัว ด้วยโปรแกรม AmThai และ ThaiOCR และแสดงผลการทดสอบตามตารางที่ 4.3

ตารางที่ 4.2 ผลการทดสอบตัวอักษร 39 รูปแบบตัวอักษรด้วยโปรแกรม ThaiOCR และ AmThai

ชุดข้อมูล	จำนวนรูปแบบ ตัวอักษรที่ ทดสอบ	AmThai 2.5 (lite version)		ThaiOCR version 1.5b	
		Number of Correct Pattern	Accuracy (%)	Number of Correct Pattern	Accuracy (%)
1	80	11	13.92	13	16.46
2	80	25	31.65	22	27.85
3	80	23	29.11	17	21.52
4	80	18	22.78	21	26.58
5	80	70	88.61	48	60.76
6	80	31	39.24	26	32.91
7	80	51	64.56	40	50.63
8	80	22	27.85	30	37.97
9	80	16	20.25	12	15.19
10	80	13	16.46	19	24.05
11	80	37	46.84	29	36.71
12	80	7	8.86	10	12.66

ตารางที่ 4.2 (ต่อ)

ชุดข้อมูล	จำนวนรูปแบบ ตัวอักษรที่ ทดสอบ	ArnThai 2.5 (lite version)		ThaiOCR version 1.5b	
		Number of Correct Pattern	Accuracy (%)	Number of Correct Pattern	Accuracy (%)
13	80	9	11.25	12	15.00
14	80	59	73.75	42	52.50
15	80	17	21.25	16	20.00
16	80	31	38.75	29	36.25
17	80	29	36.25	34	42.50
18	80	22	27.5	20	25.00
19	80	64	80.00	55	68.75
20	80	32	40.00	31	38.75
21	80	45	56.25	39	48.75
22	80	45	56.25	32	40.00
23	80	48	60.00	41	51.25
24	80	29	36.25	30	37.50
25	80	18	22.50	20	25.00
26	80	16	20.00	10	12.50
27	80	21	26.25	17	21.25
28	80	50	62.50	43	53.75
29	80	71	88.75	58	72.50
30	80	72	90.00	53	66.25
31	80	25	31.25	18	22.50
32	80	14	17.50	15	18.75
33	80	13	16.25	17	21.25
34	80	43	53.75	36	45.00
35	80	45	56.25	40	50.00
36	80	28	35.00	20	25.00
37	80	26	32.50	23	28.75

ตารางที่ 4.2 (ต่อ)

ชุดข้อมูล	จำนวนรูปแบบ ตัวอักษรที่ ทดสอบ	AmThai 2.5 (lite version)		ThaiOCR version 1.5b	
		Number of Correct Pattern	Accuracy (%)	Number of Correct Pattern	Accuracy (%)
38	80	21	26.25	17	21.25
39	80	35	43.75	30	37.50
รวม	3,120	1,252	40.13	1,085	34.78

จากตารางที่ 4.2 ลำดับชุดข้อมูลที่ 1 – 39 อ้างอิงตามตารางที่ 4.1 ลำดับที่ 1 – 39 ตามลำดับ

ตารางที่ 4.3 ผลการทดสอบป้ายทะเบียนรถไทย 506 ป้ายด้วยโปรแกรม ThaiOCR และ AmThai

ชุดข้อมูล	จำนวนรูปแบบ ตัวอักษรที่ ทดสอบ	AmThai 2.5 (lite version)		Thai OCR version 1.5b	
		Number of Correct Pattern	Accuracy (%)	Number of Correct Pattern	Accuracy (%)
ป้ายทะเบียน รถไทย 506 ป้ายทะเบียน	8,121	3,972	48.91	3,359	41.36

ผลจากการตรวจสอบประสิทธิภาพความสามารถในการรู้จำของโปรแกรม AmThai และ ThaiOCR แสดงให้เห็นว่าโปรแกรมทั้งสองยังมีความสามารถในการรู้จำไม่ดีเท่าที่ควร งานวิจัยนี้จึงทำการทดสอบกับระบบใหม่ที่นำเสนอ และเปรียบเทียบกับการรู้จำวิธีของ Fuzzy ARTMAP โดยแบ่งชุดข้อมูลภาพเอกสารออกเป็นสองส่วนคือ Training Set และ Testing Set อ้างอิงลำดับชุดข้อมูลรูปแบบภาพเอกสารตามตารางที่ 4.1 ทั้งนี้ได้แบ่ง Training Set และ Testing Set ตามตารางที่ 4.4 และได้มีการทำการทดลองทั้งหมด 16 ครั้ง แสดงผลการทดลองที่ได้ตามตารางที่ 4.5

สำหรับการทดลองอีกส่วนคือการทดสอบชุดข้อมูลภาพป้ายทะเบียนจำนวน 506 ป้ายทะเบียนที่มีการกำหนดให้เป็น Testing Set และมีการกำหนด Training Set ตามตารางที่ 4.4 ซึ่งมีการทำการทดลองทั้งหมด 16 ครั้ง แสดงผลการทดลองที่ได้ตามตารางที่ 4.6

จากตารางที่ 4.4 ลำดับชุดข้อมูลที่ 1 – 39 อ้างอิงตามตารางที่ 4.1 ลำดับที่ 1 – 39 ตามลำดับ

ตารางที่ 4.4 ชุดข้อมูล Training Set และ Testing Set ในการทดลองทั้งหมด 16 ครั้ง

การทดลอง ครั้งที่	ชุดข้อมูล Training Set	ชุดข้อมูล Testing Set
1	1-10	11-39
2	11-20	1-10,21-39
3	21-30	1-20,31-39
4	1, 31-39	2-30
5	1-5, 11-15	6-10,16-39
6	16-20, 26-30	1-15,21-25,31-39
7	21-25, 31-35	1-20,26-30,36-39
8	1,5,7,10,15,20,25,30,35,39	2-4,6,8,9,11-14,16-19,21-24,26-29, 31-34,36-38
9	5,10,15,20,25,30,35-38	1-4,6-9,11-14,16-19,21-24,26-29, 31-34,39
10	2,4,6,8,10,12,14,16,18,20	1,3,5,7,9,11,13,15,17,19,21-39
11	3,6,9,12,15,18,21,24,27,30	1,2,4,5,7,8,10,11,13,14,16,17,19,20,22, 23,25,26,28,29,31-39
12	4,8,12,16,20,24,28,32,36,39	1-3,5-7,9-11,13-15, 17-19,21-23,35-27, 29-31,33-35,37,38
13	6,12,18,24,30,7,14,21,28,35	1-5,8-11,13,15-17,19,20,22,23,25-27, 29,31-34,36-39
14	1,2,3,11,12,13,21,22,23,31	4-10,14-20,24-30,32-39
15	6,7,8,16,17,18,26,27,28,36	1-5,9-15,19-25,29-35,37-39
16	8,9,10,18,19,20,28,29,30,39	1-7,11-17,21-27,31-38

ตารางที่ 4.5 ผลการทดลองในการรู้จำชุดข้อมูลภาพเอกสาร (อ้างอิงตามตารางที่ 4.4)

การ ทดลอง ครั้งที่	จำนวนรูปแบบ ตัวอักษรที่ ทดสอบ (Testing Set)	Cross-correlation Neural Network (9 Block-segments)		Fuzzy ARTMAP	
		Number of Correct Pattern	Accuracy (%)	Number of Correct Pattern	Accuracy (%)
1	2,320	2,004	86.38	1,820	78.45

ตารางที่ 4.5 (ต่อ)

การทดลองครั้งที่	จำนวนรูปแบบตัวอักษรที่ทดสอบ (Testing Set)	Cross-correlation Neural Network (9 Block-segments)		Fuzzy ARTMAP	
		Number of Correct Pattern	Accuracy (%)	Number of Correct Pattern	Accuracy (%)
2	2,320	2,167	93.41	1,865	80.39
3	2,320	2,022	87.16	1,742	75.09
4	2,320	2,173	93.66	1,867	80.47
5	2,320	2,135	92.03	1,874	80.78
6	2,320	1,948	83.97	1,751	75.47
7	2,320	2,203	94.96	1,962	84.57
8	2,320	2,046	88.19	1,773	76.42
9	2,320	2,102	90.60	1,769	76.25
10	2,320	2,180	93.97	1,903	82.03
11	2,320	2,159	93.06	1,896	81.72
12	2,320	2,130	91.81	1,878	80.95
13	2,320	2,086	89.91	1,833	79.01
14	2,320	2,032	87.59	1,811	78.06
15	2,320	1,995	85.99	1,798	77.50
16	2,320	2,190	94.40	1,913	82.46

ตารางที่ 4.6 ผลการทดลองในการรู้จำป้ายทะเบียนรถไทย (อ้างอิง Training Set ตามตารางที่ 4.4)

การทดลองครั้งที่	จำนวนตัวอักษรบนป้ายทะเบียน (Testing Set)	Cross-correlation Neural Network (9 Block-segments)		Fuzzy ARTMAP	
		Number of Correct Pattern	Accuracy (%)	Number of Correct Pattern	Accuracy (%)
1	8,121	6,913	85.12	6,047	74.46
2	8,121	6,972	85.85	6,139	75.59
3	8,121	6,941	85.47	6,026	74.20
4	8,121	6,967	85.79	6,103	75.15
5	8,121	6,991	86.09	6,180	76.10

ตารางที่ 4.6 (ต่อ)

การทดลองครั้งที่	จำนวนตัวอักษรบนป้ายทะเบียน (Testing Set)	Cross-correlation Neural Network (9 Block-segments)		Fuzzy ARTMAP	
		Number of Correct Pattern	Accuracy (%)	Number of Correct Pattern	Accuracy (%)
6	8,121	7,021	86.45	6,215	76.53
7	8,121	7,110	87.55	6,308	77.68
8	8,121	7,231	89.04	6,434	79.23
9	8,121	6,849	84.34	6,024	74.18
10	8,121	6,925	85.27	6,059	74.61
11	8,121	6,997	86.16	6,136	75.56
12	8,121	7,042	86.71	6,341	78.08
13	8,121	7,003	86.23	6,195	76.28
14	8,121	7,181	88.43	6,392	78.71
15	8,121	7,106	87.50	6,317	77.79
16	8,121	7,204	88.71	6,376	78.51

จากผลการทดลองข้างต้นเป็นการแบ่งกรอบตัวอักษรออกเป็น 9 กรอบย่อยซึ่งจากตารางที่ 4.5 ได้ผลการรู้จำชุดข้อมูลภาพตัวอักษรเฉลี่ยเท่ากับ 90.44% แต่เมื่อพิจารณาแบ่งกรอบย่อยตัวอักษรออกเป็นจำนวนที่มากกว่า 9 กรอบย่อย ผลที่ได้จะมีข้อดีคือแนวโน้มของผลการรู้จำกรอบตัวอักษรที่ได้ออกมาจะดีขึ้นประมาณ 3.14% ตามตารางที่ 4.7 แสดงผลการรู้จำของกรอบตัวอักษรที่มีการแบ่งย่อยออกเป็น 16 กรอบย่อยซึ่งได้ผลการรู้จำชุดข้อมูลภาพตัวอักษรเฉลี่ยเท่ากับ 93.58% เนื่องจากแต่ละกรอบก็จะประกอบด้วยทิศทางหลายเส้นของตัวอักษรตามวิธีการเขียนของตัวอักษรนั้น ๆ ในขณะที่เดียวกันก็มีข้อเสียก็คือจำนวนโหนดที่เกิดในระดับ Hidden Layer จะมีจำนวนมากทำให้ระยะเวลาที่ใช้ในการเรียนรู้ใช้เวลานานขึ้น นอกจากนี้หาก Training Set ไม่มีกลุ่มข้อมูลภาพตัวอักษรตัวเดียวกันที่มีความกว้างไม่เท่ากันตามแต่ละฟอนต์ก็จะส่งผลให้ Testing Set ที่ส่งข้อมูลเข้าไปไม่สามารถรู้จำได้อย่างถูกต้อง และเมื่อพิจารณากรณีที่กรอบย่อยตัวอักษรที่ถูกแบ่งออกเป็นจำนวนที่น้อยกว่า 9 กรอบย่อย ผลที่ได้จะมีข้อเสียคือแนวโน้มของผลการรู้จำกรอบตัวอักษรที่ได้ออกมาจะน้อยลง 4.58% ตามตารางที่ 4.8 แสดงผลการรู้จำของกรอบตัวอักษรที่มีการแบ่งย่อยออกเป็น 4 กรอบย่อยซึ่งได้ผลการรู้จำชุดข้อมูลภาพตัวอักษรเฉลี่ยเท่ากับ 85.86% ทั้งนี้เนื่องจากแต่ละกรอบก็จะประกอบไปด้วยทิศทางหลายเส้นของตัวอักษรตามวิธีการเขียนของตัวอักษรนั้นมากกว่า 1 ทิศทางที่รวมอยู่ใน 1 กรอบย่อยตัวอักษร แต่ข้อดีก็คือจำนวนโหนดที่เกิดในระดับ

Hidden Layer จะมีจำนวนน้อยลงทำให้ระยะเวลาที่ใช้ในการเรียนรู้ใช้เวลาเร็วขึ้น ดังนั้นการแบ่งกรอบย่อยตัวอักษร ถ้ามีการแบ่งย่อยกรอบมากขึ้นก็อาจจะทำให้ข้อมูลบางส่วนที่ควรจะอยู่ในเส้นทางกรอบเดียวกันนั้นกลายเป็นคนละกรอบ และส่งผลให้การรู้จำตัวอักษรบางตัวที่ไม่ได้รับการเรียนรู้ในช่วง Training ก็จะไม่สามารถรู้จำออกมาได้

ตารางที่ 4.7 ผลการทดลองในการรู้จำชุดข้อมูลภาพเอกสารที่แบ่งย่อยกรอบตัวอักษรออกเป็น 16 กรอบย่อย (อ้างอิงตามตารางที่ 4.4)

การทดลองครั้งที่	จำนวนรูปแบบตัวอักษรที่ทดสอบ (Testing Set)	Cross-correlation Neural Network (16 Block-segments)		Fuzzy ARTMAP	
		Number of Correct Pattern	Accuracy (%)	Number of Correct Pattern	Accuracy (%)
1	2,320	2,191	94.44	1,820	78.45
2	2,320	2,213	95.39	1,865	80.39
3	2,320	2,187	94.27	1,742	75.09
4	2,320	2,217	95.56	1,867	80.47
5	2,320	2,210	95.26	1,874	80.78
6	2,320	2,030	87.50	1,751	75.47
7	2,320	2,289	98.66	1,962	84.57
8	2,320	2,148	92.59	1,773	76.42
9	2,320	2,122	91.47	1,769	76.25
10	2,320	2,210	95.26	1,903	82.03
11	2,320	2,192	94.48	1,896	81.72
12	2,320	2,195	94.61	1,878	80.95
13	2,320	2,141	92.28	1,833	79.01
14	2,320	2,097	90.39	1,811	78.06
15	2,320	2,036	87.76	1,798	77.50
16	2,320	2,260	97.41	1,913	82.46

ตารางที่ 4.8 ผลการทดลองในการรู้จำชุดข้อมูลภาพเอกสารที่แบ่งย่อยกรอบตัวอักษรออกเป็น 4  
กรอบย่อย (อ้างอิงตามตารางที่ 4.4)

การ ทดลอง ครั้งที่	จำนวนรูปแบบ ตัวอักษรที่ ทดสอบ (Testing Set)	Cross-correlation Neural Network (4 Block-segments)		Fuzzy ARTMAP	
		Number of Correct Pattern	Accuracy (%)	Number of Correct Pattern	Accuracy (%)
1	2,320	1,904	82.07	1,820	78.45
2	2,320	2,036	87.76	1,865	80.39
3	2,320	1,961	84.53	1,742	75.09
4	2,320	2,008	86.55	1,867	80.47
5	2,320	2,025	87.28	1,874	80.78
6	2,320	1,872	80.69	1,751	75.47
7	2,320	2,117	91.25	1,962	84.57
8	2,320	1,931	83.23	1,773	76.42
9	2,320	2,007	86.51	1,769	76.25
10	2,320	2,010	86.64	1,903	82.03
11	2,320	2,071	89.27	1,896	81.72
12	2,320	2,003	86.34	1,878	80.95
13	2,320	2,011	86.68	1,833	79.01
14	2,320	1,983	85.47	1,811	78.06
15	2,320	1,903	82.03	1,798	77.50
16	2,320	2,029	87.46	1,913	82.46

## บทที่ 5

# สรุปผลการวิจัยและข้อเสนอแนะ

งานวิจัยนี้ได้นำเสนอแนวทางวิธีใหม่สำหรับกระบวนการรู้จำตัวอักษรพิมพ์ภาษาไทยเพื่อแก้ปัญหาภาพตัวอักษรพิมพ์ภาษาไทยที่ปราศจากส่วนหัวที่มีลักษณะเป็นรูป โดยจะเป็นสาเหตุหนึ่งที่ทำให้กระบวนการรู้จำตัวอักษรภาษาไทยผิดพลาดหรือไม่สามารถรู้จำได้ งานวิจัยนี้ครอบคลุมทั้งกระบวนการก่อนการรู้จำและกระบวนการรู้จำ โดยขั้นตอนแรกจะต้องทำการหากรอบตัวอักษรโดยใช้วิธีการหาโครงร่างฮิสโตแกรมและการหาขอบภาพ ทำให้ได้ตำแหน่งและขอบเขตของภาพข้อมูลตัวอักษรแต่ละตัว หลังจากหากรอบตัวอักษรได้แล้ว จะต้องนำกรอบตัวอักษรไปวิเคราะห์หาจุดศูนย์กลางของภาพตัวอักษรเพื่อแยกระดับของตัวอักษรซึ่งในการทดลองแบ่งออกเป็น 3 ระดับคือระดับบน ระดับกลาง และระดับล่าง โดยกลุ่มข้อมูลที่ใช้ในการทดลองคือภาพตัวอักษรพิมพ์ภาษาไทยที่ประกอบไปด้วย พญชชนะ สระ วรรณยุกต์ และตัวเลขอารบิก จากนั้นก็จะทำการ Skeleton ตัวอักษรแล้วมีการกำหนดรหัสแทนทิศทางเส้นตัวอักษร 9 รหัส โดยจะทำการกำหนดจุดเริ่มต้นและจุดปลาย ทั้งนี้ในการเก็บข้อมูลนั้นจะแบ่งกรอบตัวอักษรออกเป็น 9 ส่วน เพื่อใช้เป็นอินพุตในการส่งเข้าสู่โครงข่ายประสาทเทียมในกระบวนการรู้จำแนววิธีใหม่คือ Hierarchical Normalize Cross Correlation Neural Network

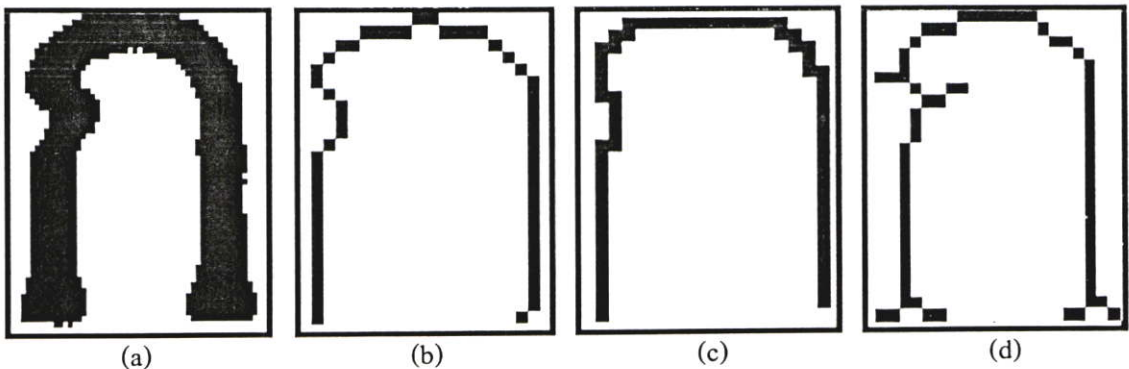
### 5.1 สรุปผลการวิจัย

เนื่องจากงานวิจัยนี้พิจารณาที่กลุ่มตัวอักษรพิมพ์ภาษาไทยที่มีรูปแบบวิธีการเขียนที่แตกต่างไปจากแต่ก่อนคือบริเวณที่เป็นส่วนหัวตัวอักษรนั้นถูกตัดทิ้งไปหรือ ปราศจากส่วนหัวตัวอักษรที่มีลักษณะเป็นรูป โดยจากงานวิจัยที่ผ่านมายังไม่มียงานวิจัยที่มุ่งเน้นแก้ปัญหาเหล่านี้ โดยนับว่าเป็นปัญหาสำคัญที่ทำให้ระบบการรู้จำนั้นไม่สามารถรู้จำได้อย่างถูกต้อง ในการทดลองทำการเปรียบเทียบกับซอฟต์แวร์ทางการค้าคือ AmThai และ ThaiOCR ซึ่งผลที่ได้จากการรู้จำของโปรแกรม AmThai และ ThaiOCR ยังไม่ดีเท่าที่ควร นอกจากนี้ได้ทำการทดลองเปรียบเทียบระหว่างวิธีในการรู้จำแบบ Fuzzy ARTMAP กับแนวคิดใหม่ในการรู้จำแบบ Cross Correlation Neural Network ที่นำเสนอในงานวิจัยครั้งนี้ โดยผลการทดลองนั้นสรุปได้ว่าแนวคิดวิธีใหม่ที่น่าเสนอนี้มีความสามารถในการรู้จำได้ดีกว่าการรู้จำแบบ Fuzzy ARTMAP เพราะแนวคิดวิธีใหม่ที่น่าเสนอนี้มีการแยกส่วนของรูปแบบข้อมูลออกเป็น 9 ส่วนย่อยและมีการปรับค่า Weight Vector เพียงบางส่วนของรูปแบบข้อมูลตัวอักษรที่เข้าสู่ระบบการรู้จำ นอกจากนี้ แนวคิดวิธีใหม่ที่น่าเสนอนี้ใช้เวลาในการรู้จำตัวอักษรแต่ละตัวได้เร็วกว่าการรู้จำแบบ Fuzzy ARTMAP เพราะจำนวนโหนด

ในแต่ละลำดับชั้นนั้นมีจำนวนน้อยกว่าจำนวนโหนดใน Fuzzy ARTMAP เพราะค่าข้อมูลอินพุตที่รับเข้าไปจะเป็นค่าข้อมูลที่เป็นเส้นตัวอักษรของรูปแบบที่ต้องการรู้จำ แต่ Fuzzy ARTMAP นั้นจะส่งทุก ๆ ค่าข้อมูลในกรอบตัวอักษรเข้าไปในกระบวนการรู้จำ ทั้งนี้ลักษณะของชุดข้อมูลภาพตัวอักษรที่ใช้ในการทดลองเมื่อเป็นกลุ่มที่ปราศจากส่วนหัวที่เป็นรูป จึงส่งผลให้ภาพตัวอักษรมีลักษณะคล้ายคลึงกันมาก หากมองในลักษณะของความสัมพันธ์ในแต่ละจุดที่ประกอบเป็นเส้นทางเดินของตัวอักษรนั้น พบว่าแต่ละภาพตัวอักษรมีความสัมพันธ์กัน แม้ว่าขนาดของตัวอักษรจะมีขนาดเล็กหรือใหญ่ แนวคิดวิธีใหม่ในงานวิจัยนี้จึงนำเอาฟังก์ชันครอสคอร์รีเลชันเข้ามาใช้เป็น Choice Function ในการวัดค่าความคล้ายคลึงกัน และในกระบวนการก่อนการรู้จำได้มีการจัดแบ่งแยกแยะข้อมูลภาพตัวอักษรออกเป็น 3 กลุ่มระดับตัวอักษร ได้แก่ กลุ่มระดับตัวอักษรบน กลุ่มระดับตัวอักษรกลาง และกลุ่มระดับตัวอักษรล่าง จากนั้นในแต่ละกลุ่มที่ถูกแบ่งจะส่งรูปแบบตัวอักษรเข้าไปสู่กระบวนการรู้จำแบบโครงข่ายประสาทเทียมตามวิธีการของฟีดฟอร์เวิร์ดแบบสี่ชั้น (Four Layers Feedforward Neural Network) โดยอาศัย Hierarchical Normalized Cross-Correlation (HNCC) Neural Network

## 5.2 ข้อเสนอแนะในการทำวิจัยต่อไป

1. ปัญหาของรูปแบบตัวอักษรภาษาไทยที่เชื่อมติดกันภายในตัวอักษรส่งผลให้ประสิทธิภาพของการรู้จำลดลง ด้วยเหตุนี้ถ้ามีการพัฒนาวิธีการพิจารณาเส้นตัวอักษรที่ติดกันภายในตัวอักษรได้ ก็จะส่งผลให้กระบวนการรู้จำได้ผลออกมาถูกต้องมากขึ้น
2. วิธีการในการ Thinning ตัวอักษรมีผลในการรู้จำกลุ่มตัวอักษรที่ไม่มีหัว ดังนั้นหากมีการทำ Thinning ตัวอักษรแล้วยังคงโครงสร้างเดิมได้มากก็จะทำให้กระบวนการรู้จำได้ผลออกมาดีขึ้นตามรูปที่ 5.1 แสดงวิธีการ Thinning ตัวอักษรด้วยวิธีต่าง ๆ



รูปที่ 5.1 แสดงวิธีการ Thinning ตัวอักษรด้วยวิธีต่าง ๆ (a) กรอบตัวอักษร (b) Hilditch's Algorithm (c) Skeletonization (d) Thinning ลายเส้นตัวอักษร

## บรรณานุกรม

- [1] Tanspraset C. and Koanantakool T., "Thai OCR : A Neural Network Application", **IEEE Transection on Digital Signal Processing Application**. pp.90-95, 1996.
- [2] Nucharee P., Wichian P., and Seinosuke N., "Segmentation of Horizontal and Vertical Touching Thai Character." **International Technical Conference on Circuit Systems, Computers and Communications. ITC-CSCC'99**. Niigata, Japan. 1999.
- [3] Wicha P., Somchai J., Prasert C. "Segmentation of Connected Characters Using Distinctive Feature of The Character in Thai Character Recognition System." **Electrical Engineering Conference on Circuits and Systems**. pp.338-342, 1997.
- [4] Shunji, Ching Y. Suen, Kazuhiko Y. "Historical Review of OCR Research and Development" **Proceedings of IEEE**. vol. 80, 7 Jul 1992.
- [5] D.G. Elliman and I.T. Lancaster. "A Review of Segmentation and Contextual Analysis Techniques for Text Recognition." **Pattern Recognition**. Vol. 23. No. 3/4, pp.337-376, 1990.
- [6] Pongsuree L., Wichian P., Nucharee P. "Repairing Broken Thai Printed Characters Using Feature Extraction", **The National computer Science and Engineering Conference**. 2000.
- [7] อุบลรัตน์ พาศิขานุกูล, วิเชียร เปรมชัยสวัสดิ์, นุชรี เปรมชัยสวัสดิ์. "การซ่อมแซมตัวอักษรตัวพิมพ์ภาษาไทยที่ขาด." การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 25 (EECON-25), 21-22 พฤศจิกายน 2545. หน้า 36-40.
- [8] Dimitrios Charalampidis, Georgios C. Anagnostopoulos, Michael Georgiopoulos, Takis Kasparis. "Fuzzy ART and Fuzzy ARTMAP with Adaptively Weighted Distances" EE Dept., University of New Orleans, School of EE & CS, University of Central Florida.
- [9] P.R. Innocent, R.I. John, J. King. "A Modified ART algorithm for processing Fuzzy Granules" Centre for Computational Intelligence, De Montfort University, Leicester UK.
- [10] YI LU. "Machine printed character segmentation – An overview" Department of Electrical and Computer Engineering, The University of Michigan – Dearborn, Dearborn, MI 48128-1491, U.S.A. **Pattern Recognition**. Vol. 28, No. 1. pp. 67-80, 1995.

- [11] Hirobumi Nishida. "Structural Feature Extraction Using Multiple Bases" School of Computer Science and Engineering, University of Aizu, Aizu Wakamatsu, Fukushima 965-80, Japan. **Computer vision and image understanding**. Vol. 62, No. 1. July, pp. 78-89, 1995.
- [12] Sargur N. Srihari, Tao Hong and Geetha Srikantan. "Machine-Printed Japanese Document Recognition" Center of Excellence for Document Analysis and Recognition, State University of New York at Buffalo, The UB Commons, Suite 202, 520 Lee Entrance, Amherst, NY 14228-2567, U.S.A. **Pattern Recognition**. Vol. 30, No. 8. pp. 1301-1313, 1997.
- [13] J. P. Lewis. "**Fast Normalized Cross-Correlation**" Industrial Light & Magic
- [14] Paul Bourke. "**Cross Correlation**" [Online]. Available:  
<http://astronomy.swin.edu.au/~pbourke/analysis/correlate/>
- [15] Kritawan Siriboon, Apirak Jirayusakul, Boontee Kruatrachue. "HMM Topology Selection for On-line Thai Handwritten Recognition" **Proceedings of the First International Symposium on Cyber Worlds. CW'02**. pp. 0142 – 0145, 2002.
- [16] Roongroj Nopsuwanchai, Dan Povey. "Discriminative Training for HMM-Based Offline Handwritten Character Recognition" **Proceedings of the Seventh International Conference on Document Analysis and Recognition. ICDAR'03**. pp. 114 – 118, 2003.
- [17] Ithipan Methasate, Sutat Sae-tang. "The Clustering Technique for Thai Handwritten Recognition" **Proceedings of the 9<sup>th</sup> Int'l Workshop on Frontiers in Handwriting Recognition. IWFHR-9**. pp. 450 – 454, 2004.
- [18] Pitak Thumwarin, Suphamit Chittayasothorn. "An Object-oriented Expert System for Thai Character Recognition" **The 1998 IEEE Asia-Pacific conference on Digital Object Identifier. IEEE APCCAS**. pp 153 – 156, 1998.
- [19] Pisit Phokharatkul, Chom Kimpan. "Recognition of Handprinted Thai Character Using the Cavity Features of Character Based on Neural Network" **The 1998 IEEE Asia-Pacific conference on Digital Object Identifier. IEEE APCCAS**. pp 149 – 152, 1998.
- [20] Suraphun Airphaiboon, Manas Sangworasil, Shozo Kondo. "Off-line Handwritten Thai Characters from Word Script." **Proceedings of the 12<sup>th</sup> IAPR International Conference on Volume 2**. pp 445 – 449.

- [21] Arrak Pornchaikajornsak, Arit Thammano. "Handwritten Thai Character Recognition Using Fuzzy Membership Function And Fuzzy ARTMAP" **Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics And Automation**. July 16 – 20, 2003. pp 40 – 44, 2003.
- [22] S.Mitatha, K.Dejharn, F.Chevasuvit, B.Chankuang and W.Kasemsiri. "Experimental results of Using Rough Sets for printed Thai Characters Recognition" **Electrical and Electronic Technology, 2001 TENCON. Proceedings of IEEE Region 10 International**. Vol 1, 19 – 22 Aug. 2001. pp 331 – 334. 2001.
- [23] Watjanapong Kasemsiri and Chom Kimpan. "Printed Thai Character Recognition using Fuzzy-Rough Sets" **Electrical and Electronic Technology, 2001 TENCON. Proceedings of IEEE Region 10 International**. Vol 1, 19 – 22 Aug. 2001. pp 326 – 330. 2001.
- [24] Olivier de vel, Sujint Wangsuya and Danny Coomans. "On Thai Character Recognition" **Neural Networks, Proceedings of IEEE International**. Vol 4. 27 Nov – 1 Dec, 1995. pp 2095 – 2098, 1995.
- [25] Chularat Tanprasert and Thaweesak Koanantakool. "Thai OCR: A Neural Network Application" **IEEE TENCON – Digital Signal Processing Applications**. Vol 1. 26 – 29 Nov, 1996. pp. 90 – 95, 1996.
- [26] Thirapiroon Thongkamwitoon, Widhyakorn Asdornwised, S upavadee Aramvith, and Somchai Jitapunkul. "On-line Thai-English Handwritten Character Recognition using Distinctive Features" **Circuits and Systems, 2002. APCCAS'02. 2002 Asia-Pacific**. Vol 2. 28 – 31 Oct. 2002. pp. 259 – 264, 2002.
- [27] Rud Budsayaplakorn, Widhayakorn Asdornwised, and Somchai Jitapunkul. "On-line Thai Handwritten Character Recognition using Hidden Markov Model and Fuzzy Logic" **IEEE 13<sup>th</sup> Workshop on Neural Networks for Signal Processing (NNSP'03)**. 17 – 19 Sept. 2003. pp. 537 – 546, 2003.
- [28] Parinya Sanguansat, Widhayakorn Asdornwised, and Somchai Jitapunkul. "Online Thai Handwritten Character Recognition Using Hidden Markov Models and Support Vector Machines" **International Symposium on Communications and Information Technologies 2004. ISCIT 2004**. Sapporo, Japan. October 26 – 29, 2004. pp. 492 – 497, 2004.

- [29] K.S.Bae, K.K.Kim, Y.G.Chung, W.P.Yu. "Character Recognition System for Cellular Phone with Camera" **Proceedings of the 29<sup>th</sup> Annual International Computer Software and Applications conference. COMPSAC'05.** pp. 539 – 544, 2005.
- [30] Emilie Poisson, Christian Viard Gaudin, Pierre-Michel Lallican. "Multi-Modular Architecture based on Convolutional Neural Networks for Online Handwritten Character Recognition" **Proceedings of the 9<sup>th</sup> International Conference on Neural Information Processing. ICONIP'02.** Vol 5. pp. 2444 – 2448, 2002.

ภาคผนวก

## ภาคผนวก ก.

### ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

1. Arit Thammano and Pruegsa Duangphasuk, “**Printed Thai Character Recognition Using the Hierarchical Cross-correlation ARTMAP**”, The 17<sup>th</sup> IEEE International Conference on Tool with Artificial Intelligence (ICTAI05), Hong Kong, November 14-16 2005, pp.695-698

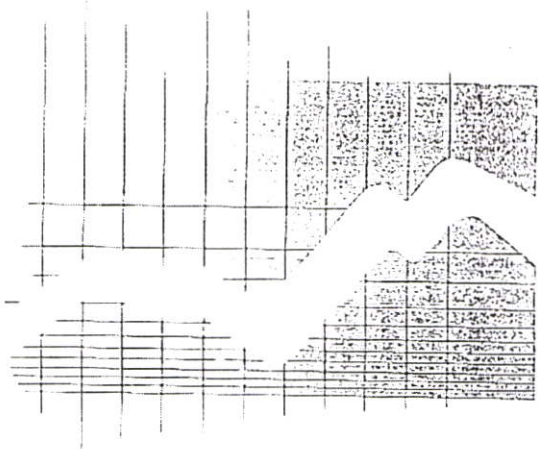
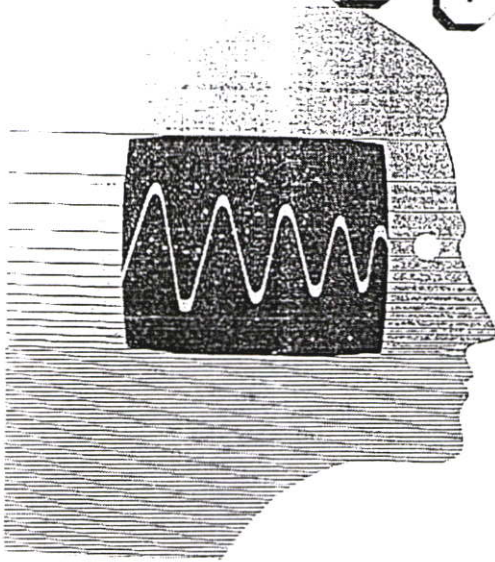
Proceedings

Seventeenth International Conference on

# TOOLS WITH ARTIFICIAL INTELLIGENCE

# ICTAI 2005

14-16 November 2005  
Hong Kong, China



Sponsored by  
IEEE Computer Society  
Co-Sponsored by  
Information Technology Research Institute,  
Wright State University  
The Hong Kong University of Science and Technology

Edited by  
Andrew Lim

  
IEEE  
COMPUTER  
SOCIETY

 **IEEE**

## Printed Thai Character Recognition Using the Hierarchical Cross-correlation ARTMAP

Arit Thammano and Pruegsa Duangphasuk  
 Faculty of Information Technology  
 King Mongkut's Institute of Technology Ladkrabang  
 Bangkok, 10520 Thailand  
 Phone: (662)737-2551 to 4 ext. 529, Fax: (662)326-4332  
 E-mail: arit@it.kmitl.ac.th and pruegsa@mut.ac.th

### Abstract

Traditionally, Thai characters are composed of circle, zigzag line, curve, and head. However, many new Thai fonts, which are now gaining in popularity, do not follow the traditional writing rule: the head has been omitted from the characters. Without the head, it is very difficult to segregate the characters. Even the best commercial Thai OCR software has difficulty in recognizing this kind of character. Therefore, the hierarchical cross-correlation ARTMAP is proposed in this paper to recognize the no-head Thai characters.

### 1. Introduction

Thai character recognition is a very complex problem. Even though many researches have been conducted on the recognition of Thai characters since the past decade, the success in recognizing Thai characters is still limited, compared to English characters. This is due to the following reasons: (a) there are a number of characters in Thai language, each of which has a complicated structure; (b) many Thai characters look very alike, e. g. ก-ก, ข-ข, ฃ-ฃ, ฅ-ฅ, ง-ง, ฉ-ฉ, ซ-ซ, ฌ-ฌ; (c) many new Thai fonts violate the traditional writing rule. Traditionally, Thai characters are composed of circle, zigzag line, curve, and head. However, many new fonts omit the head of the characters for the sake of its beauty. The head is the loop at the beginning of the characters. It is one of the important features used to differentiate one character from another. Without the head, the characters are even more similar as shown in Figure 1. The situation gets even worse when these font styles are growing in popularity, but researchers do not pay much attention to them.

Standard Thai Characters	No-head Characters
ก ก ก	ก ก ก
ข ข ข	ข ข ข

Figure 1. The similarity of Thai characters when the head of the characters are left out

Since the past decade, there has been a considerable growth in the Thai character recognition area. However, most of the researches put a lot of effort into the recognition of standard Thai characters. The methods widely used are the statistical approach [1, 2], the neural network based approach [3, 4, 5], and the hybrid approach [6, 7]. In this paper, the hierarchical cross-correlation ARTMAP is proposed to recognize the no-head Thai characters.

Following this introduction, section 2 describes the architecture of the proposed model and its learning algorithm. The preprocessing of the character images is described in section 3. In section 4, the experimental results are demonstrated and discussed. Finally, section 5 is the conclusion.

### 2. The proposed model

The proposed model is a four-layer feedforward neural network as shown in Figure 2. The first layer is the input layer, which consists of 9 segments. Each segment contains N input nodes. The number of input nodes in a segment is associated with the number of feature components extracted from each segment of the

character image. The second layer is the hidden layer. Each node in the  $s^{\text{th}}$  segment of the second layer is fully connected to the input nodes of the same segment via the connections  $w_{ji}^s$ . The weight vector  $w_j^s$  of dimension  $N$  represents the reference pattern of the  $j^{\text{th}}$  node in the  $s^{\text{th}}$  segment of the hidden layer. The third layer is the cluster layer. The nodes in the second and the third layers are constructed during the training process. The fourth layer is the output layer. Each node in the output layer represents a class of character. During the supervised learning, the input vector is presented to the model, together with its respective target output vector. The input vector is denoted by  $I = \{I^1, \dots, I^s, \dots, I^9\} = \{(I_1^1, \dots, I_N^1), \dots, (I_1^9, \dots, I_N^9)\}$ , where  $s$  is the  $s^{\text{th}}$  segment in the input layer, and  $N$  is the number of feature components in each segment. Once the model receives the input and its associated target output (I, O), the maximum of the normalized cross-correlation between the input vector of the  $s^{\text{th}}$  segment ( $I^s$ ) and each weight vector of the same segment ( $w_j^s$ ) is computed and the outputs of the  $j^{\text{th}}$  node,  $T_j^s(I^s)$ , in the  $s^{\text{th}}$  segment of the hidden layer are then determined.

$$f(x, y) = \max_p \left\{ \frac{\sum_i x_i y_{i-p}}{\sqrt{\left(\sum_i x_i^2\right) \left(\sum_i y_i^2\right)}} \right\} \quad (2)$$

where  $i = 1, \dots, N$ ,  
 $j = 1, \dots, M^s$ ,  
 $s = 1, \dots, 9$ .  
 $N$  is the number of feature components in each segment.

$M^s$  is the number of hidden nodes in the  $s^{\text{th}}$  segment.

$f(x, y)$  is the maximum of the normalized cross-correlation at any shifted period  $p$ .

$p$  is the lag variable, which is between  $-(N-1)$  to  $(N-1)$ .

For each segment, the hidden node with the highest output value,  $T_j^s(I^s)$ , is selected as the winning node. Next, the vigilance criterion is evaluated to check whether the degree of matching between the input pattern and the reference pattern of the chosen node is within an acceptable level.

$$T_j^s(I^s) = f(I^s, w_j^s) \quad (1)$$

$$T_j^s(I^s) \geq \rho_{\text{hidden}} \quad (3)$$

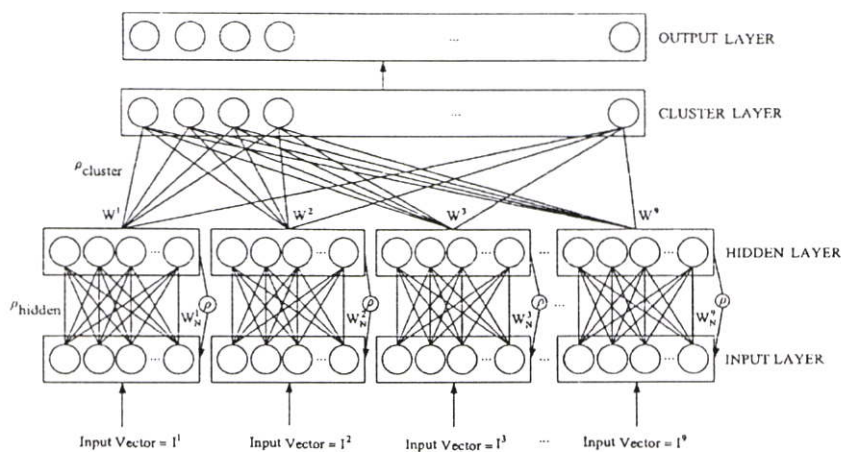


Figure 2. Architecture of the hierarchical cross-correlation ARTMAP

where  $\rho_{\text{hidden}}$  is the vigilance parameter at the hidden layer. It has a value between 0 and 1. If the winning node meets the above criterion, the weight vector of the winning node ( $w_j^*$ ) will be updated according to the following equation:

$$w_j^*(t+1) = \beta(1^* \wedge w_j^*(t)) + (1-\beta)w_j^*(t) \quad (4)$$

However, if the condition in (3) is not satisfied, a new hidden node is recruited to code the input pattern. The weight of this new node is initialized to be equal to the input pattern.

$$w_j^* = 1^* \quad (5)$$

Next, the weight vector of the winning node is transmitted to the next layer. The choice function of each  $k^{\text{th}}$  node in the cluster layer is then evaluated as follows:

$$C_k = f(w_j, w_k) \quad (6)$$

where  $w_j = \{w_j^1, w_j^2, \dots, w_j^m\}$ ,  
 $w_k = \{w_{k1}, w_{k2}, \dots, w_{kn}\}$ .

$f(w_j, w_k)$  is the maximum of the normalized cross correlation between the weight vector transmitted from the hidden layer and the weights of the connections from the hidden layer to the  $k^{\text{th}}$  node of the cluster layer.

The system then makes a cluster choice by selecting the winning node  $K$  with maximum choice function value, among all the nodes  $k$  in the cluster layer.

$$C_K = \max_k \{C_k\} \quad (7)$$

Next,  $C_K$  is compared to the vigilance parameter ( $\rho_{\text{cluster}}$ ).  $\rho_{\text{cluster}}$  is the vigilance parameter at the cluster layer. It has a value between 0 and 1. If  $C_K$  is greater than or equal to  $\rho_{\text{cluster}}$  and the winning cluster node  $K$  belongs to the correct class defined by the target output vector, the weight vector  $w_{k^*}$  will be updated according to where the weight vector  $w_j^*$  is originated:

(a) The weight vector  $w_j^*$  is transmitted from the newly recruited  $J$  node,

$$w_{k^*}(t+1) = \beta(w_j^* \wedge w_{k^*}(t)) + (1-\beta)w_{k^*}(t) \quad (8)$$

(b) Otherwise,

$$w_{k^*}(t+1) = w_{k^*}(t) \quad (9)$$

However, if  $C_K$  is less than  $\rho_{\text{cluster}}$  or the winning cluster node  $K$  does not belong to the correct class defined by the target output vector, a new cluster node is recruited and its weight is initialized to be equal to the weight vector transmitted from the hidden layer ( $w_j$ ). Then the connection between a new cluster node and the target output is created.

During testing, each testing vector is applied in turn and its class is predicted. The class whose cluster node returns the maximum output value is the result of the prediction.

### 3. Preprocessing

An optical scanner scans a printed document and converts it into an image. The segmentation process then segments the entire image into individual character blocks. Once the image is turned into character blocks, the thinning algorithm is applied to each character image to reduce the thickness of the character image to its skeleton. Next, each character block is divided into 9 equal segments and the feature is extracted from the thinned character image. In this paper, the merely feature is a list of directional codes. There are 8 directional codes, labelled "1" to "8". "1" represents the line whose angle is between  $15\pi/8$  and  $\pi/8$ , "2" represents the line whose angle is between  $\pi/8$  and  $3\pi/8$ , "3" represents the line whose angle is between  $3\pi/8$  and  $5\pi/8$  and so on.

In the feature extraction process, the starting point of a stroke must be identified first, then the directional code is used to traverse along the contour of the character. Figure 4 shows the directional codes for the character "a."

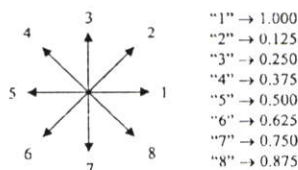


Figure 3. Eight directional codes

### 4. Experimental results

The performance of the proposed network is evaluated against the fuzzy ARTMAP neural network and two of the best commercial software available in the market. The commercial products used in this

research are "ArnThai version 2.5" and "Thai OCR version 1.5b." ArnThai is a commercial software developed by National Electronic and Computer Technology Center (NECTEC). It is the most widely used and the most precise character recognition software in Thailand.



[0.125 1 0.125 1 0.125 1 1 0.125],[1 1 1 1 0.875 1 1 1],  
 [0.875 1 1 1 1 0.875 0.75 0.75 0.75 0.875 0.75 0.75 0.75],  
 [0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75],  
 [0.75 0.75 0.625 0.75 0.625 0.75 0.625 0.5 0.5 0.5 0.5],  
 [0.625 0.5 0.5 0.5 0.5 0.5 0.375],[0.5 0.5 0.5 0.375 0.5  
 0.375 0.5 0.375 0.25 0.25 0.25 0.25 0.25],[0.25 0.25 0.25  
 0.25 0.125 1 0.125 1 0.125 1 1 1],[1 1 1 1 1]

Figure 4. Directional codes for the character "a"

To compare the performance of the selected systems, eight experiments have been conducted on 12 no-head fonts. The twelve no-head fonts used in this research are Boaboon, Chalit, Chanok, DB Private, Jasmine, Js 75 Pumpu, Kodchiang, Lily, Pat Ex, Patpong, Sathorn, and Silom. In each experiment, six fonts are randomly chosen to use as the training data, while the remaining fonts are used as the testing data.

The experimental results (Table 1) demonstrate a very strong performance of the proposed approach. It outperforms fuzzy ARTMAP and two of the best commercial products by a wide margin. However, it should be noted here that the inability of the available commercial products to retrain the models with new data sets is probably the cause of their failure to recognize the above no-head fonts.

Table 1. The experimental results

	Recognition Rate (%)			
	Proposed Model	Fuzzy ARTMAP	ArnThai	Thai OCR
1	76.74	55.04	27.13	22.87
2	86.11	59.52	21.83	19.05
3	89.29	59.92	23.02	25.00
4	81.40	54.26	31.01	29.46
5	82.56	55.04	24.42	20.16
6	85.71	58.33	19.05	21.83
7	85.94	56.64	22.27	23.44
8	82.42	55.86	25.00	22.66

5. Conclusion

The experiments show that the proposed approach can be used successfully to solve the printed Thai character recognition problem. It can recognize the no-head Thai characters, which are problematic even for the best commercial Thai OCR software. With its ability to separately train each segment of the network, the hierarchical cross-correlation ARTMAP can achieve much higher performance on the experimental data in comparison to the fuzzy ARTMAP neural network and two of the best commercial software available in the market.

6. References

[1] R. Nopsuwanchai and D. Povey, "Discriminative Training for HMM-based Offline Handwritten Character Recognition," *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, Vol. 1, August 2003, pp. 114-118.  
 [2] T. Theeramunkong and C. Wongtapan, "Off-line Isolated Handwritten Thai OCR Using Island-based Projection with N-gram Model and Hidden Markov Models," *Information Processing and Management*, Vol. 41, Issue 1, January 2005, pp. 139-160.  
 [3] C. Tanprasert and T. Koanantakool, "Thai OCR: A Neural Network Application," *Proceedings of the 1996 IEEE TENCON*, Vol. 1, 1996, pp. 90-95.  
 [4] B. Kijisirikul, S. Sinthupinyu, and A. Supanwansa, "Thai Printed Character Recognition by Combining Inductive Logic Programming with Backpropagation Neural Network," *Proceedings of the 1998 IEEE Asia-Pacific Conference on Circuits and Systems*, 1998, pp. 539-542.  
 [5] P. Phokharatkul and C. Kimpan, "Recognition of Handprinted Thai Characters Using the Cavity Features of Character Based on Neural Network," *Proceedings of the 1998 IEEE Asia-Pacific Conference on Circuits and Systems*, 1998, pp. 149-152.  
 [6] N. Premchaiswadi, "A Study on Printed Thai Character Recognition," *Doctoral Dissertation*, Waseda University, Tokyo, March 2001.  
 [7] A. Thammano and P. Ruxpakawong, "Printed Thai Character Recognition Using the Hybrid Approach," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E85-A, No. 6, June 2002, pp. 1236-1241.

## ประวัติผู้เขียน

ชื่อ	นางสาวพุกษา ดวงผาสุข
วัน เดือน ปีเกิด	14 พฤษภาคม 2523
ที่อยู่	เลขที่ 99/376 หมู่บ้านปาริชาติ ถ.คุ้มเกล้า แขวงลำปลาทิว เขตลาดกระบัง กรุงเทพมหานคร 10520
ประวัติการศึกษา	2543 วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น
ประวัติการทำงาน	2545-ปัจจุบัน อาจารย์ประจำ สังกัดคณะวิทยาการและเทคโนโลยีสารสนเทศ ภาควิชาเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีมหานคร