สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การรู้จำตัวอักษรลายมือเขียนโดยใช้โครงข่ายประสาทเทียมแบบไร้น้ำหนัก

Handwriting Character Recognition Using Weightless Artificial Neural Network

T042707

จัดทำโดย

นางสาวฤตี จิตต์ปราณี

เลขหมู่.....
เลขทะเบียน 42707
วัน, เดือน, ปี 6 มิ.ย. 2545

.b.....
.i.....

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรู้จำตัวอักษรลายมือเขียนโดยใช้โครงข่ายประสาทเทียมแบบไร้น้ำหนัก

Handwriting Character Recognition Using Weightless Artificial Neural Network

จัดทำโดย

นางสาวฤตี จิตต์ปราณี

อาจารย์ที่ปรึกษา

ผศ.ดร. สุรพันธุ์ เอื้อไพบูลย์

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2543

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การรู้จำตัวอักษรลายมือเขียนโดยใช้โครงข่ายประสาทเทียมแบบไร้น้ำหนัก

Handwriting Character Recognition Using Weightless Artificial Neural Network

ผู้จัดทำ

นางสาวฤตี จิตต์ปราณี

.....อาจารย์ที่ปรึกษา

(.....)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานเรื่อง การรู้จำตัวอักษรลายมือเขียนโดยใช้โครงข่ายประสาทเทียมแบบไร้น้ำหนัก

Handwriting Character Recognition Using Weightless Artificial Neural Network

จัดทำโดย นางสาวฤตี จิตต์ปราณี 40010650

โครงงานนี้ได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

.....

(.....)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรู้จำตัวอักษรลายมือเขียนโดยใช้โครงข่ายประสาทเทียมแบบไร้น้ำหนัก

นางสาวฤตี จิตต์ปราณี
ผศ.ดร. สุรพันธุ์ เอื้อไพบูลย์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2543

บทคัดย่อ

รายงานฉบับนี้ได้นำเสนอเกี่ยวกับวิธีการจดจำตัวอักษรภาษาไทย ซึ่งเป็นตัวอักษรที่ค่อนข้างซับซ้อนและมีความใกล้เคียงกันมากในแต่ละตัวอักษร ซึ่งวิธีการจดจำตัวอักษรนี้จะใช้คุณสมบัติของโครงข่ายประสาทเทียมแบบไร้น้ำหนัก ซึ่งช่วยให้ระบบมีการประมวลผลที่เร็วและง่ายขึ้นกว่าแบบเดิมมาก

การจดจำตัวอักษรของโครงข่ายประสาทเทียมนั้นเกิดจากกระบวนการฝึกสอน ซึ่งข้อมูลที่ใช้ฝึกสอนจะต้องเป็นข้อมูลที่เห็นความแตกต่างของแต่ละตัวอักษรอย่างชัดเจน ในโครงงานนี้จึงได้ใช้ข้อมูลฝึกสอนแบบข้อมูลลายมือของตัวอักษรนั้นๆ และข้อมูลกราฟโครงร่างตัวอักษร ซึ่งเป็นผลให้เปอร์เซ็นต์การรู้จำของระบบสูงเป็นที่น่าพอใจอย่างยิ่ง โดยข้อมูลที่ใช้ฝึกสอนและทดสอบการรู้จำนั้น นำเข้าโดยการเขียนข้อมูลด้วยปากกาอิเล็กทรอนิกส์

นอกจากขบวนการประมวลผลโดยโครงข่ายประสาทเทียมแบบไร้น้ำหนักที่เป็นจุดเด่นของโครงงานนี้แล้ว ยังมีกระบวนการอีกอย่างหนึ่งที่มีความสำคัญไม่แพ้กัน นั้นคือกระบวนการจัดการข้อมูลก่อนการประมวลผล หรือการนอร์มอลไลซ์ข้อมูลนั่นเอง ในโครงงานนี้มีการจัดการข้อมูลหลายขั้นตอนซึ่งมีขั้นตอนหลักๆได้แก่ การปรับขนาดตัวอักษร , การเอียงตัวอักษรจนได้ตำแหน่งที่เสถียร และการปรับความหนาตัวอักษร โดยเฉพาะขั้นตอนในส่วนการเอียงตัวอักษรจนได้ตำแหน่งที่เสถียรนั้น ถือว่าเป็นจุดเด่นอีกอันหนึ่งในโครงงานนี้ ซึ่งตำแหน่งที่เสถียรนี้หาได้โดยใช้หลักการหาความถี่สูงสุดของกราฟกระจายความถี่ที่พล็อตจากจำนวนตำแหน่งของข้อมูลทางด้านนอนจากการเอียงตัวอักษรตามมุมต่างๆนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Handwriting Character Recognition Using Weightless Artificial Neural Network

Ms. Ruetee Chitpranee

Dr. Surapan Airphaiboon Advisor

2000

Abstract

This thesis presents Thai handwriting character recognition. Since, the structure is more complicated and looks so similar with each other, this project is using a well quality processing of Weightless Artificial Neural Networks to recognize the characters, which makes the system advanced in speed and more easier.

The process of character recognition is started from training. The training data must be distinguished or extracted the differences from the characters. So in this project, handwriting and contour tracking graphs are used as input data for the training via an electronic pen.

Another important point of recognition by Weightless Artificial Neural Networks is preprocessing algorithm or data normalization. There are many processes used in this project but the main is scaling process , character slanted until finding the stabilized position and thickening process. Especially, the character slanting plays the key role in preprocessing algorithm that uses a histogram distribution graph to find the position that gives the highest frequency in x axis.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	I
Abstract	II
สารบัญ	III
สารบัญรูป	IV
สารบัญตาราง	V
บทที่ 1 บทนำ	1
บทที่ 2 Preprocessing	2
2.1 Preprocessing ในส่วนรูปแบบการฝึกสอนด้วย Handwriting	2
2.2 Preprocessing ในส่วนรูปแบบการฝึกสอนด้วย Contour tracking graph	8
บทที่ 3 Neural Network	12
3.1 Artificial Neural Network	12
3.2 Weightless Artificial Neural Network	13
3.3 กระบวนการฝึกสอน Neuron	14
3.4 กระบวนการประมวลผลของระบบ Neural Network	19
บทที่ 4 การทดลองการรู้จำตัวอักษร	24
บทที่ 5 วิจารณ์และข้อเสนอแนะ	36
ภาคผนวก ก ตัวอักษรที่ใช้ในการทดสอบ	
ภาคผนวก ข ตัวอักษรที่ใช้ในการฝึกสอน	
ภาคผนวก ค โปรแกรมที่ใช้หาตำแหน่งให้ Neuron จับ	
ภาคผนวก ง โปรแกรมที่ใช้ในการ Training	
ภาคผนวก จ โปรแกรมที่ใช้ทดสอบการรู้จำ	
กิตติกรรมประกาศ	
บรรณานุกรม	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ 2.1 บล็อกไดอะแกรมการเตรียมข้อมูลตัวอักษรก่อนเข้าสู่การประมวลผลของการฝึกสอน ด้วย Handwriting 3
รูปที่ 2.2 ผลจากการปรับขนาด 55×55 pixels 4
รูปที่ 2.3 สมการที่ใช้เป็นหลักในการเอียงตัวอักษร 5
รูปที่ 2.4 Flowchart แสดงขั้นตอนการเอียงตัวอักษรเพื่อหาตำแหน่งที่เสถียรที่สุด 5
รูปที่ 2.5 ภาพแสดงผลจากการเอียงตัวอักษรให้มีความเสถียรที่สุด 6
รูปที่ 2.6 ผลจากการเพิ่มความหนาและการปรับขนาด 7
รูปที่ 2.7 ผลจากการปรับขนาดและการเพิ่มความหนา 7
รูปที่ 2.8 ผลจากการขยายข้อมูล Discrete 7
รูปที่ 2.9 ภาพแสดงการพล็อต Contour tracking graph 8
รูปที่ 2.10 บล็อกไดอะแกรมการเตรียมข้อมูลตัวอักษรก่อนเข้าสู่การประมวลผลของการฝึกสอน ด้วย Contour tracking graph 9
รูปที่ 2.11 Contour tracking graph ที่ได้ทำการ Scaling 100 ×40 pixels 10
รูปที่ 2.12 ภาพแสดงกระบวนการ Preprocessing 11
รูปที่ 3.1 เซลล์ประสาท (Neuron) ทางชีวภาพ 13
รูปที่ 3.2 Artificial Neural Network 13
รูปที่ 3.3 Random Access Memory 14
รูปที่ 3.4 ภาพแสดงการรับข้อมูลของ Synapses 14
รูปที่ 3.5 การฝึกสอน Neuron ใน Neural Network โดยใช้คุณสมบัติของ RAM 15
รูปที่ 3.6 ภาพแสดงการหาตำแหน่งที่ Optimize 17
รูปที่ 3.7 ภาพแสดงตำแหน่งที่ Opimize ตามค่าจากการ Opimize ค่าต่างๆ 18
รูปที่ 3.8 ภาพแสดงตำแหน่งที่ Optimize และตำแหน่งที่ Random ในโครงงานนี้ 18
รูปที่ 3.9 การประมวลผลของ Neuron ใน Neural Network โดยใช้คุณสมบัติของ RAM 19
รูปที่ 3.10 บล็อกไดอะแกรมแสดงการทำงานอย่างคร่าวๆของ Neural Network 21
รูปที่ 3.11 ภาพแสดงตำแหน่งของ Synapses บนข้อมูลต่างๆ 22
รูปที่ 3.12 กราฟแสดงคะแนนของแต่ละตัวอักษรจากการทดสอบตัวอักษร ง 23
รูปที่ 4.1 ภาพแสดงการจับตำแหน่งของ Synapses ในการทดลองที่ 1 24
รูปที่ 4.2 Contour Tracking graph จากการเขียนตัวอักษร ท 26
รูปที่ 4.3 ภาพแสดงการจับตำแหน่ง Synapses ในการทดลองที่ 3 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.4 ภาพแสดงการหาคำตอบของการประมวลผลแบบใช้ Neuron แยกกันในแต่ละข้อมูล 29
รูปที่ 4.5 ภาพแสดงการหาคำตอบของการประมวลผลแบบใช้ข้อมูลร่วมกันในแต่ละ Neuron 30
รูปที่ 4.6 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ง 30
รูปที่ 4.7 ภาพแสดงการจับตำแหน่งของ Synapses ในการทดลองที่ 4 31
รูปที่ 4.8 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ก 32
รูปที่ 4.9 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ถ 33
รูปที่ 4.10 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ภ 33
รูปที่ 4.11 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ข 33
รูปที่ 4.12 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ฆ 34
รูปที่ 4.13 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ฎ 34
รูปที่ 4.14 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ฏ 34
รูปที่ 4.15 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ย 35
รูปที่ 4.16 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ส 35
รูปที่ 4.17 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ฬ 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ 4.1 เปอร์เซ็นต์การรู้จำตัวอักษร ที่มีตัวแปรเป็นจำนวน Synapses ของ Neuron 25

ตารางที่ 4.2 เปอร์เซ็นต์การรู้จำตัวอักษร ที่มีตัวแปรเป็นจำนวนของ Neuron 27

ตารางที่ 4.3 เปอร์เซ็นต์การรู้จำตัวอักษร ที่มี Neuron ใช้ข้อมูลร่วมกันของข้อมูลลายมือเขียน กับข้อมูลภาพของ Contour Tracking graph 29

ตารางที่ 4.4 เปอร์เซ็นต์การรู้จำตัวอักษร ที่มีตัวแปรเป็นตำแหน่ง Synapses ของ Neuron บน Contour Tracking graph 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

การรู้จำตัวอักษรโดยเฉพาะอย่างยิ่งตัวอักษรแบบลายมือเขียน กระบวนการ Preprocessing และกระบวนการทาง Neural Network เป็นกระบวนการที่สำคัญอย่างยิ่งในการเพิ่มประสิทธิภาพของการรู้จำตัวอักษร เพราะ ลายมือเขียนจะมีคุณลักษณะแตกต่างกันไปตามแต่ละบุคคล ซึ่งเป็นการยากที่จะแยกแยะตัวอักษรแบบลายมือเขียนได้ ทำให้เกิดความผิดพลาดในการรู้จำตัวอักษรของ Neural ได้ ดังนั้นจึงต้องหาวิธีที่จะลดข้อแตกต่างของตัวอักษรในแต่ละลายมือและหาวิธีการทาง Neural Network ที่มีประสิทธิภาพเพื่อเพิ่มเปอร์เซ็นต์การรู้จำตัวอักษรดังกล่าว

กระบวนการทาง Neural Network เป็นกระบวนการที่จำลองมาจากกระบวนการทางความคิดของสมองมนุษย์ ซึ่งมีคุณลักษณะพิเศษหลายประการที่เป็นประโยชน์ต่อระบบการประมวลผลอันได้แก่ การรับข้อมูลแบบ Parallel Processing ทำให้สามารถรับ Input ได้หลายทางมาประมวลผลในเวลาเดียวกัน , ใช้หลักการง่ายๆ ในการตัดสินใจ เช่น การมีค่าถึงค่า Threshold ใน Neural แบบ Weightless , มีคุณสมบัติ Noise & Fault Tolerance ที่ดีเยี่ยม กล่าวคือทนทานต่อการเปลี่ยนแปลงของ Input (Input มี noise) และระบบการประมวลผลจะไม่หยุดการทำงานกลางคันแม้ว่าหน่วยประมวลผลบางหน่วยจะได้รับความเสียหายจนไม่อาจทำงานได้ , ไม่ต้องใช้กระบวนการทาง Mathematics ที่ซับซ้อน , มีคุณสมบัติ Generalization คือ Neural ยังคงให้คำตอบที่สมเหตุสมผลแม้ว่าจะมีสัญญาณรบกวนหรือข้อมูลขาดความสมบูรณ์หรือข้อมูลนั้นไม่เคยเห็นมาก่อนก็ตาม , และใช้เวลาในการประมวลผลและตัดสินใจที่รวดเร็ว เป็นต้น แต่ระบบ Neural Network ก็มีข้อเสียบางประการ เช่น ไม่สามารถใช้กับงานที่ต้องการความละเอียดสูงได้ เป็นต้น ในโครงงานนี้ได้มีการจำลองการทำงานของ Neural Network ลงในคอมพิวเตอร์ด้วยการเขียนโปรแกรมการทำงานในขั้นตอนต่างๆของกระบวนการทาง Neural Network ซึ่งผู้จัดทำได้เลือกโปรแกรม Delphi ซึ่งเป็นโปรแกรมที่ใช้ภาษา Pascal เป็นตัวพัฒนา Application ต่างๆบน Window

นอกจากนี้การจัดข้อมูลก่อนการเข้าสู่การประมวลผลของ Neural หรือ Preprocessing ก็เป็นส่วนสำคัญ เราต้องจัดข้อมูลตัวอักษรให้อยู่ในรูปแบบมาตรฐานก่อนการเข้าสู่กระบวนการทาง Neural Network ไม่ว่าจะเป็นการปรับขนาด , ปรับความหนา , และการหาตำแหน่งที่เสถียร จากการทดลองการหารูปแบบข้อมูลในการฝึกสอน Neural ในภาคการเรียนที่1 ได้สรุปออกมาว่า วิธีการพล็อตกราฟโครงร่างตัวอักษร(Contour Tracking) และแบบลายมือ (Handwriting) เป็นวิธีที่ทำให้เปอร์เซ็นต์การรู้จำของ Neural สูง ดังนั้นในภาคการเรียนนี้จึงได้หาวิธีการปรับปรุงและพัฒนาวิธีการดังกล่าวให้ดียิ่งขึ้น ไม่ว่าจะเป็นการหาตำแหน่งที่ Optimize เพื่อให้ขาของ Neural จับ ซึ่งจะกล่าวในรายละเอียดในบทที่กล่าวเกี่ยวกับ Neural Network และการออกแบบโครงสร้างของ Neural Network เองให้ดียิ่งขึ้น เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

Preprocessing

โครงงานนี้เป็นโครงงานที่ทำต่อเนื่องมาจากภาคการเรียนที่ 2 ปีการศึกษา 2542 และภาคการเรียนที่ 1 ปีการศึกษา 2543 เพื่อศึกษาและพัฒนาการทำงานของระบบ Neural Network ที่จำลองการทำงานลงในโปรแกรมคอมพิวเตอร์ ซึ่งในโครงงานนี้ใช้โปรแกรม Delphi 5 ซึ่งเป็นโปรแกรมที่ใช้พัฒนา Application บน Window โดยใช้ภาษา Pascal เป็นหลักในการเขียนโปรแกรม ซึ่งเป็นโครงสร้างภาษาที่ง่ายและ Delphi ยังมีเครื่องมือช่วยเหลือในการนำคำสั่งต่างๆมาใช้งานได้อย่างสะดวกและรวดเร็ว โดยปกติ Delphi จะมีส่วนที่ใช้สำหลับออกแบบและเป็นส่วนที่ใช้ติดต่อกับผู้ใช้ (User) โดยการนำคอมโพเนนต์ (Component) ต่างๆมาวางไว้สำหรับสร้าง Application ต่างๆ เรียกว่า วินโดว์ฟอร์ม (Form Window)

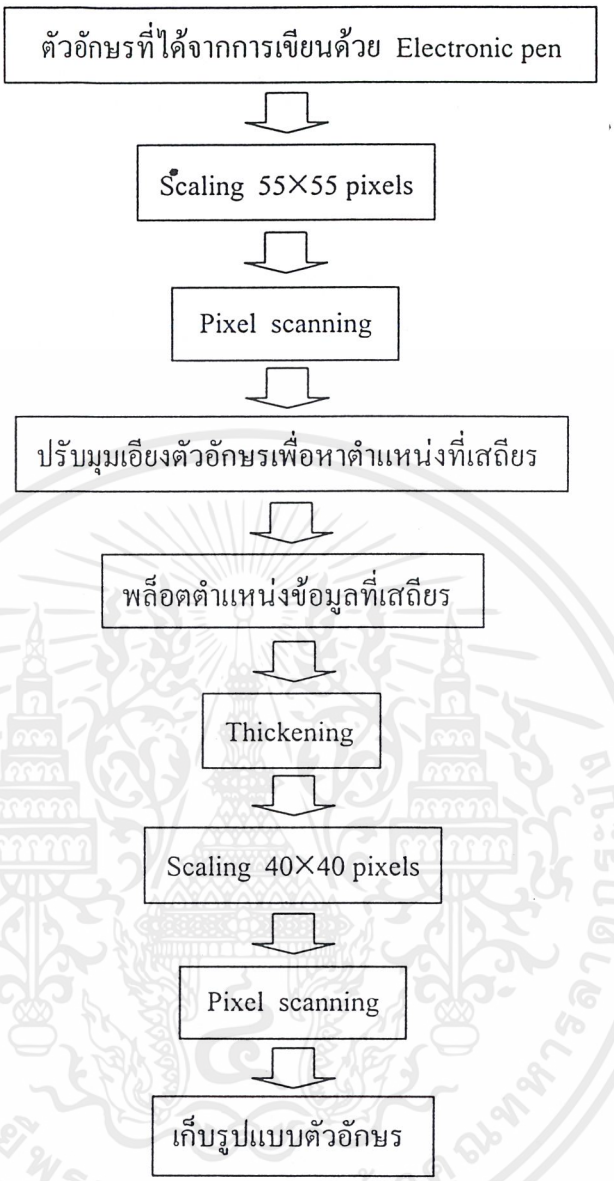
ในภาคการเรียนนี้จะเป็นการพัฒนากระบวนการ Preprocessing และกระบวนการทาง Neural Network ในการจดจำตัวอักษรเขียนภาษาไทยที่รับข้อมูลด้วยปากกาอิเล็กทรอนิกส์ ซึ่งอักษรภาษาไทยจะมีความซับซ้อนและมีความใกล้เคียงกันมากในแต่ละตัวอักษร อีกทั้งข้อมูลที่รับเข้ามานั้นเป็นภาษาเขียนที่มีความแตกต่างกันในแต่ละลายมือของแต่ละบุคคล เราจึงต้องหาวิธีจัดการข้อมูลส่วนนี้ให้มีความใกล้เคียงกันมากที่สุดในแต่ละลายมือ และหาส่วนที่แตกต่างกันในแต่ละตัวอักษรอีกด้วย จากภาคการเรียนที่แล้วได้ทำการทดลองหาวิธีดังกล่าวด้วยการหารูปแบบของข้อมูลเพื่อฝึกสอน Neural และได้ผลสรุปออกมาว่ารูปแบบของข้อมูลแบบลายมือเขียน หรือ Handwriting และข้อมูลจากกราฟโครงร่างตัวอักษร หรือ Contour tracking graph ให้เปอร์เซ็นต์การรู้จำที่สุดเป็นที่น่าพอใจ ดังนั้นในภาคการเรียนนี้จึงได้นำรูปแบบการฝึกสอนดังกล่าวมาพัฒนาและใช้เป็นแบบในการฝึกสอน Neural หรืออาจกล่าวได้ว่าเป็นรูปแบบที่เลือกในการออกแบบและพัฒนากระบวนการ Preprocessing หรือการเตรียมข้อมูลก่อนเข้าสู่กระบวนการประมวลผลนั้นเอง

ในบทนี้จะมีการจัดแบ่งกระบวนการ Preprocessing ออกเป็น 2 ส่วนด้วยกัน คือในส่วนของการจัดรูปแบบลายมือเขียน หรือ Handwriting และการจัดข้อมูลแบบกราฟโครงร่างตัวอักษร หรือ Contour tracking graph

2.1 Preprocessing ในส่วนรูปแบบการฝึกสอนด้วย Handwriting

รูปแบบการฝึกสอน Neural ด้วย Handwriting เป็นการฝึกสอนด้วยตัวอักษรที่ยังไม่ถูกดัดแปลงไปเป็นในรูปแบบอื่น เช่น การดัดแปลงไปเป็นกราฟต่างๆ เป็นต้น เพียงแต่จัดตัวอักษรดังกล่าวให้อยู่ในรูปแบบมาตรฐานเท่านั้น การจัดการข้อมูลให้อยู่ในรูปแบบมาตรฐานสามารถเขียนเป็นบล็อกไดอะแกรมอย่างคร่าวๆได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 บล็อกไดอะแกรมการเตรียมข้อมูลตัวอักษรก่อนเข้าสู่การประมวลผลของการฝึกสอนด้วย Handwriting

จากบล็อกไดอะแกรมสามารถอธิบายในแต่ละส่วนได้ดังนี้

ขั้นที่ 1 การรับข้อมูลตัวอักษรที่ได้จากการเขียนด้วย Electronic pen

การรับข้อมูลนั้นสามารถทำได้หลายวิธี อาจจะได้จากการ Scan ภาพ , การเขียนด้วยเมาส์ผ่านโปรแกรมต่างๆ เป็นต้น โดยปกติแล้วจะสามารถแบ่งวิธีการรับข้อมูลได้ 2 แบบ ได้แก่ การรับข้อมูลแบบ Online และการรับข้อมูลแบบ Offline แต่ในโครงงานนี้จะเป็นการรับข้อมูลแบบ Online เพื่อเป็นประโยชน์ในการเขียน Contour tracking graph ต่อไป

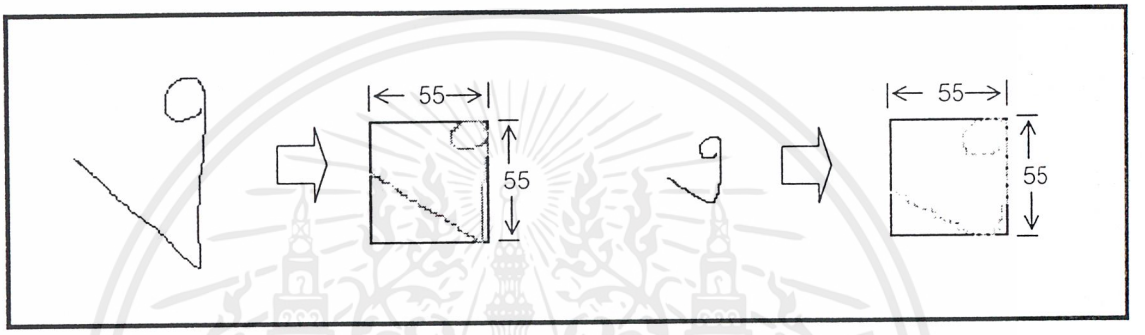
การรับข้อมูลตัวอักษรในโครงงานนี้จะใช้ปากกาอิเล็กทรอนิกส์ หรือ Electronic pen (เมาส์ที่เป็นปากกา) เขียนบนแผ่นรองที่มีมาพร้อมกับปากกาอิเล็กทรอนิกส์ ตัวอักษรที่ได้จากการเขียนก็จะไปปรากฏ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้

บนวินโดว์ฟอร์ม (Form Window) จากนั้นเก็บตำแหน่งของตัวอักษรทางด้านแกน X และแกน Y บนวินโดว์ฟอร์มในอาเรย์ (Array) ของตำแหน่ง X และ Y เพื่อนำไปสู่การจัดการในขั้นต่อไป

ขั้นที่ 2 Scaling 55×55 pixels

Scaling 55×55 pixels หรือ การปรับขนาดตัวอักษร เพื่อทำให้ตัวอักษรมีขนาดเดียวกันหรือเป็นมาตรฐานเดียวกันก่อนการเข้าสู่กระบวนการ Preprocessing ในขั้นตอนต่อไป โดยทำการปรับขนาดตัวอักษรให้มีขนาด 55×55ในหน่วยของ Pixel ผลที่ได้แสดงดังรูป 2.2



รูปที่ 2.2 ผลจากการปรับขนาด 55×55 pixels

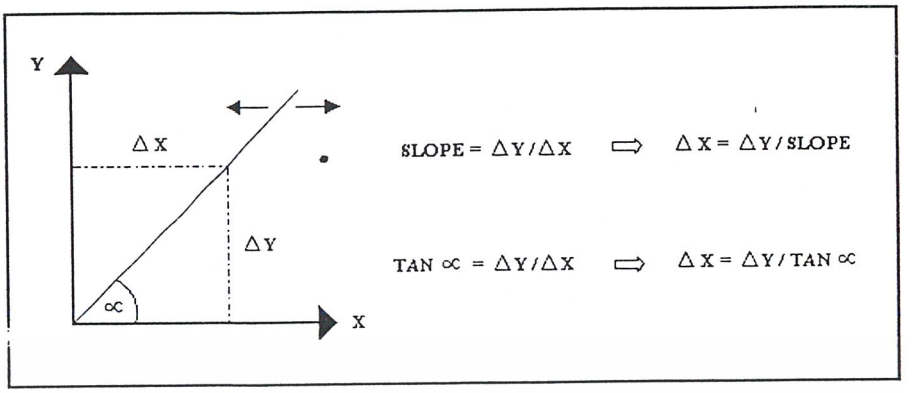
ขั้นที่ 3 Pixel scanning

Pixel scanning คือ การตัดข้อมูลของตำแหน่ง X และ Y ในอาเรย์ (Array) ที่มีการซ้ำซ้อนกันออก ซึ่งข้อมูลที่ซ้ำซ้อนกันนี้เกิดจากการกระทำของการเปลี่ยนแปลงข้อมูลของตำแหน่ง X และ Y เช่น การปรับขนาด หรือ Scaling และการปรับมุมเอียงเพื่อหาตำแหน่งที่เสถียร เป็นต้น

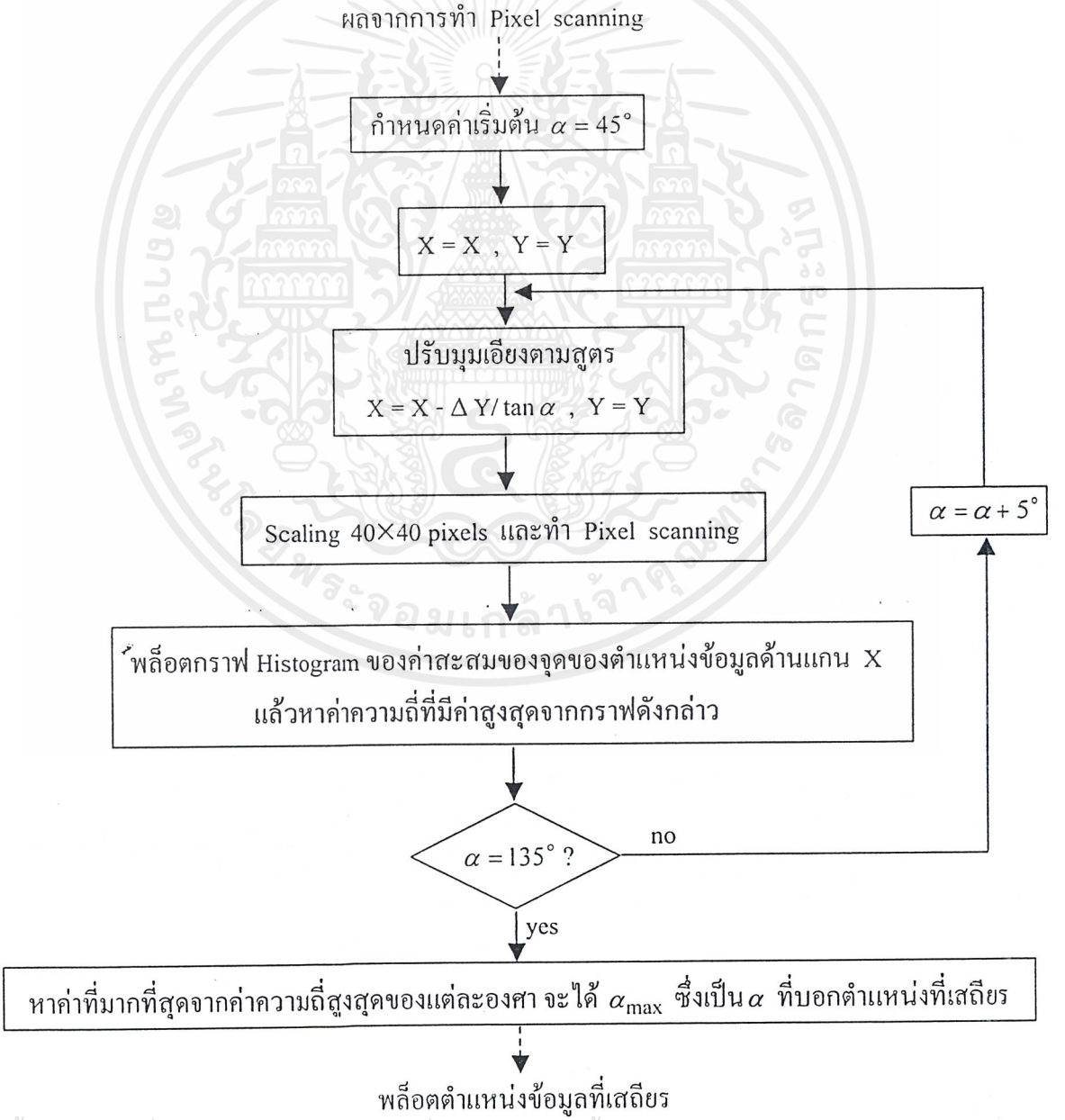
ขั้นที่4 ปรับมุมเอียงตัวอักษรเพื่อหาตำแหน่งที่เสถียร

การทำให้ตัวอักษรมีตำแหน่งที่เสถียรที่สุดไม่ว่าจะด้วยวิธีใดก็ตาม จะสามารถแก้ไขความแตกต่างของลายมือได้ระดับหนึ่ง ในโครงงานนี้เราได้ทำการทดลองหลายวิธีเพื่อที่จะทำให้ตัวอักษรอยู่ในตำแหน่งที่เสถียรที่สุด จนได้วิธีนึงที่ได้ผลเป็นที่น่าพอใจ คือ วิธีการเอียงตัวอักษรให้มีค่าความถี่ของกราฟ Histogram ทางด้านแกน X สูงสุดค่าๆหนึ่ง วิธีนี้จะทำให้ตัวอักษรอยู่ในตำแหน่งที่เสถียร ซึ่งข้อมูลที่เป็นลายมือต่างๆ จะอยู่ตำแหน่งนี้ได้เพียงตำแหน่งเดียว (อาจมีมากกว่า 1 ตำแหน่งแต่ก็จะเป็นตำแหน่งที่คงที่และเสถียร) การเอียงตัวอักษรนี้จะทำได้ โดยจะเลื่อนตำแหน่งของตัวอักษรเฉพาะส่วนที่อยู่ในแกน X ให้เปลี่ยนไปตามองศาต่างๆ (ตำแหน่ง Y คงที่) ซึ่งสามารถพิจารณาได้จากสูตรที่แสดงไว้ในรูปที่ 2.3 จากนั้นนำข้อมูลของตำแหน่งที่ได้จากการเอียงตามองศาต่างๆ มาพล็อตกราฟความถี่ (Histogram) ของข้อมูลทางด้านแกน X แล้วหาค่าความถี่ที่มีค่าสูงสุดตามองศาต่างๆ จากนั้นนำมาเปรียบเทียบหาค่าที่มากที่สุด ซึ่งค่าที่มากที่สุดจะบอกว่าที่องศาใดให้ตำแหน่งที่เสถียรที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 สมการที่ใช้เป็นหลักในการเอียงตัวอักษร

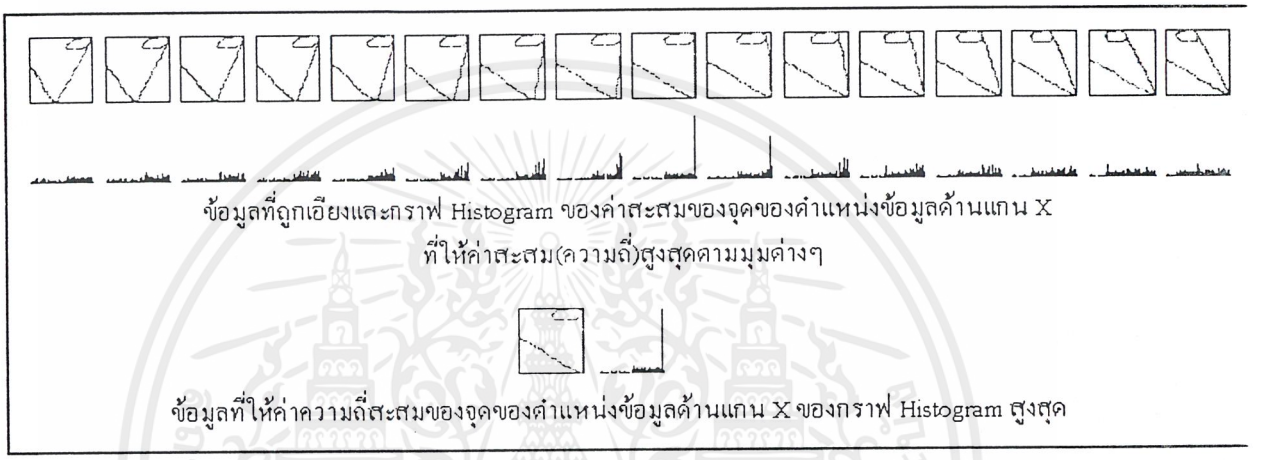


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น รวมทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.4 Flowchart แสดงขั้นตอนการเอียงตัวอักษรเพื่อหาตำแหน่งที่เสถียรที่สุด

ในโครงงานนี้ได้ทำการปรับองศาของมุมเอียงตั้งแต่ 45° - 135° ซึ่งจะให้ค่า tan 45° = 1 และ tan 135° = –1 ซึ่งเป็นค่าที่ครอบคลุมมุมของการเอียงทั้งหมด กล่าวคือ ΔX จะมีค่าเป็นบวกและลบ ได้จากสูตร $\Delta X = \Delta Y / \tan\alpha$ ซึ่ง α มีค่าตั้งแต่ 45° - 135° กระบวนการปรับมุมเอียงตัวอักษรเพื่อให้ได้ตำแหน่งที่เสถียรที่สุดในโครงงานนี้สามารถเขียนเป็น Flowchart อย่างคร่าวๆได้ดังรูป 2.4

จากกระบวนการดังรูปที่ 2.4 ผลที่ได้จากการเอียงตัวอักษรให้อยู่ในตำแหน่งที่เสถียรที่สุดสามารถแสดงได้ดังรูปที่ 2.5 (แสดงไว้สำหรับการเอียงบางองศาเท่านั้น)



รูปที่ 2.5 ภาพแสดงผลจากการเอียงตัวอักษรให้มีความเสถียรที่สุด

ขั้นที่ 5 พล็อตตำแหน่งข้อมูลที่เสถียร

การพล็อตตำแหน่งข้อมูลที่เสถียร ในส่วนกระบวนการ Preprocessing ของการฝึกสอนด้วย Handwriting การพล็อตข้อมูลจะเป็นการพล็อตตำแหน่งของข้อมูลโดยใช้สมการ $X = X - \Delta Y / \tan\alpha_{max}$, $Y = Y$ พร้อมการ Scaling 40×40 pixels

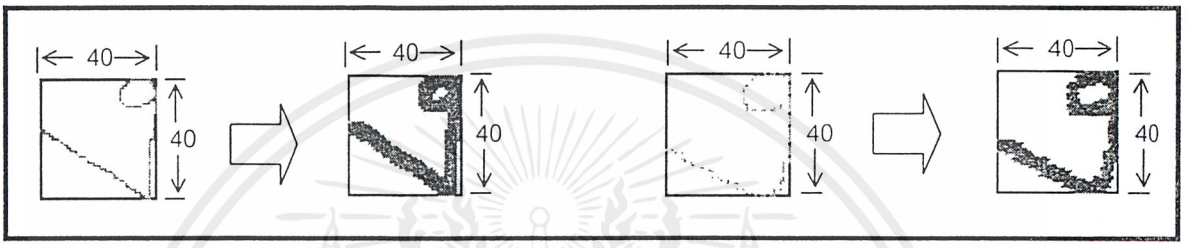
ขั้นที่ 6 Thickening หรือ การปรับความหนาตัวอักษร

การปรับความหนาตัวอักษรจากที่มีลักษณะเป็นโครงร่างบางๆให้หนาขึ้นเพื่อเพิ่มอัตราส่วนจำนวนข้อมูลระหว่างขาวกับดำ (ขาวคือส่วนที่ไม่มีข้อมูล ดำคือส่วนที่มีข้อมูล) ให้มีค่าพอๆกัน จะทำให้ข้อมูลในประสาทเทียมมีค่า 0 , 1 พอๆกัน (ลดการชนกันของข้อมูล [1]) ในแต่ละตัว ซึ่งจะมีประโยชน์มากต่อการประมวลผลของ Neural ที่เป็นแบบ Weightless เพราะ Neural จะจดจำภาพข้อมูลจากความแตกต่าง ดังนั้นการทำข้อมูลให้มีความแตกต่างอย่างเด่นชัดจะเป็นผลดีต่อการการตัดสินใจ (การประมวลผล)ของ Neural อาจกล่าวได้ว่าเป็นการเพิ่มประสิทธิภาพการตัดสินใจให้กับ Neural นั่นเอง ผลที่ได้แสดงดังรูป 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 7 Scaling 40×40 pixels

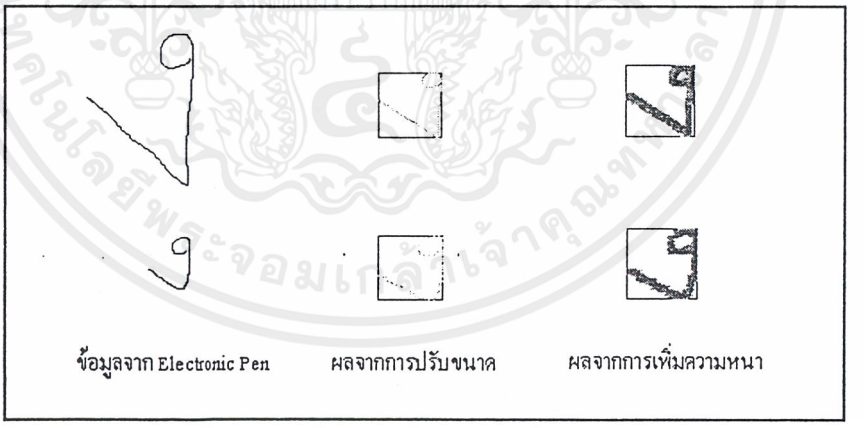
จากการ Thickening หรือ การปรับความหนาตัวอักษร จะเป็นผลให้ตัวอักษรที่มีขนาด 40×40 pixels มีขนาดใหญ่ขึ้น ดังนั้นจึงต้องทำการ Scaling 40×40 pixels หรือ การปรับขนาดตัวอักษรอีกครั้ง เพื่อทำให้ตัวอักษรมีขนาดเท่ากับ 40×40 pixels เพื่อทำให้ตัวอักษรมีขนาดเดียวกันหรือเป็นมาตรฐานเดียวกันก่อนเข้าสู่การประมวลผลของ Neural โดยทำการปรับขนาดตัวอักษรให้มีขนาด 40×40 ในหน่วยของ Pixel ผลที่ได้แสดงดังรูป 2.6



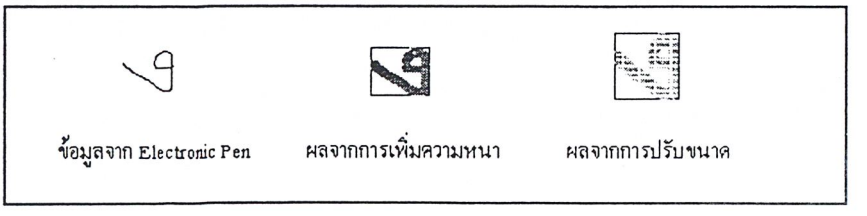
รูปที่ 2.6 ผลจากการเพิ่มความหนาและการปรับขนาด

ข้อสังเกต

การทำการปรับขนาดข้อมูลก่อนการเพิ่มความหนาจะช่วยแก้ปัญหาผลการขยายของข้อมูลที่เป็น Discrete สามารถแสดงดังรูป 2.7 และ 2.8



รูปที่ 2.7 ผลจากการปรับขนาดและการเพิ่มความหนา



รูปที่ 2.8 ผลจากการขยายข้อมูล Discrete

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 8 Pixel scanning

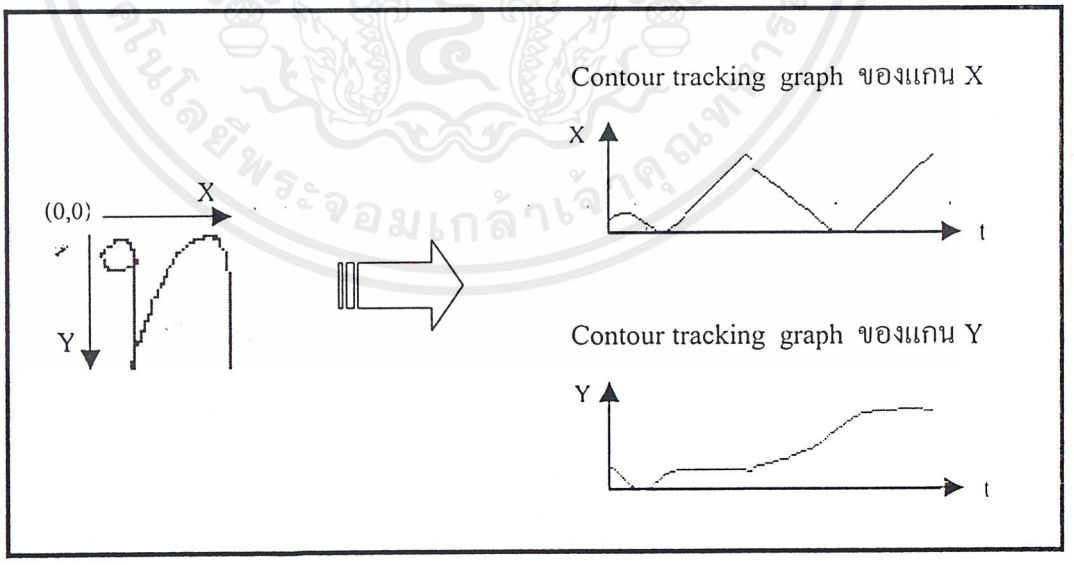
Pixel scanning คือ การตัดข้อมูลของตำแหน่ง X และ Y ในอาเรย์ (Array) ที่มีการซ้ำซ้อนกันออก ซึ่งข้อมูลที่ซ้ำซ้อนกันนี้เกิดจากการกระทำของการเปลี่ยนแปลงข้อมูลของตำแหน่ง X และ Y ในที่นี้เกิดจากการปรับขนาด หรือ Scaling

ขั้นที่ 9 การเก็บรูปแบบตัวอักษร

การเก็บข้อมูลจะเก็บข้อมูลตัวอักษรที่อยู่ในตำแหน่งที่เสถียรที่สุด ซึ่งตำแหน่งที่เสถียรที่สุดนี้ คือ ตำแหน่งที่ให้ค่าสะสมของจุดของตำแหน่งด้านแกน X (ความถี่ทางด้านแกน X) ของกราฟ Histogram สูงสุด โดยการเก็บข้อมูลตัวอักษรจะทำการเก็บข้อมูลอาเรย์ (Array) ของตำแหน่งทางด้านแกน X และตำแหน่งทางด้านแกน Y โดยจะเก็บไว้ใน File เพื่อใช้สำหรับเป็นข้อมูลในการ Training (การฝึกสอนให้กับ Neural) ส่วนข้อมูลที่ใช้ในการทดสอบการรู้จำของ Neural ไม่ต้องจัดเก็บไว้ใน File

2.2 Preprocessing ในส่วนรูปแบบการฝึกสอนด้วย Contour tracking graph

การพล็อต Contour tracking graph คือ การพล็อตค่าของตำแหน่งข้อมูลด้านแกน X หรือแกน Y กับแกนของเวลา หรืออาจกล่าวอีกนัยนึงว่าคือแกนบอกลำดับก่อนหลังของข้อมูลจากการเขียนข้อมูลแบบ Online โดย Electronic Pen ซึ่งจะมีขนาดของแกนเท่ากับจำนวนตำแหน่งของข้อมูลที่ต้องการพล็อต สามารถแสดงได้ดังรูป 2.9

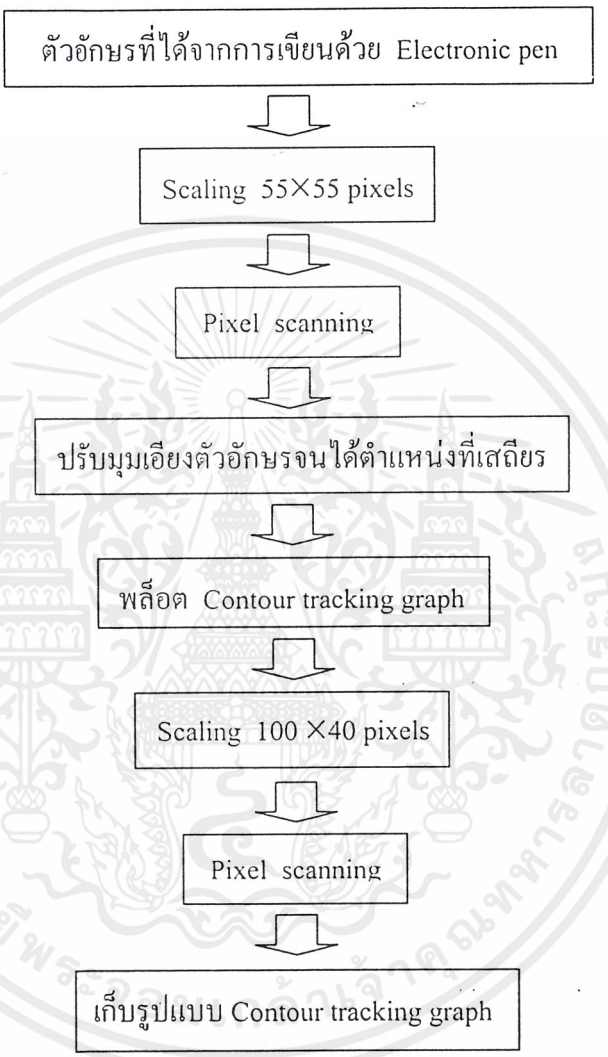


รูปที่ 2.9 ภาพแสดงการพล็อต Contour tracking graph

กระบวนการ Preprocessing ในส่วนรูปแบบการฝึกสอนด้วย Contour tracking graph จะมีขั้นตอนต่างๆเหมือนกับกระบวนการ Preprocessing ในส่วนรูปแบบการฝึกสอนด้วย Handwriting จะต่างกันตรงขั้นตอนการพล็อตข้อมูลที่ได้หลังจากการเอียงข้อมูลตามองศาต่างๆเพื่อหาตำแหน่งที่เสถียร เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Contour tracking graph แทนการพล็อตข้อมูลตามตำแหน่ง X , Y กล่าวคือ กระบวนการ Preprocessing ในขั้นตอนที่ 1 ถึง 4 จะเหมือนกับการฝึกสอนด้วย Handwriting จะต่างกันตั้งแต่ขั้นตอนที่ 5 เป็นต้นไป ซึ่งสามารถเขียนเป็นบล็อกไดอะแกรมอย่างคร่าวๆ ได้ดังนี้



รูปที่ 2.10 บล็อกไดอะแกรมการเตรียมข้อมูลตัวอักษรก่อนเข้าสู่การประมวลผลของการฝึกสอนด้วย Contour tracking graph

จากบล็อกไดอะแกรมสามารถอธิบายในแต่ละส่วนได้ดังนี้

<u>ขั้นที่ 1 – 4</u> ตัวอักษรที่ได้จากการเขียนด้วย Electronic pen , Scaling 55×55 pixels , Pixel scanning , และปรับมุมเอียงตัวอักษรจนได้ตำแหน่งที่เสถียร

ขั้นที่ 1 – 4 จะใช้วิธีการ Preprocessing เดียวกับของการฝึกสอนด้วย Handwriting

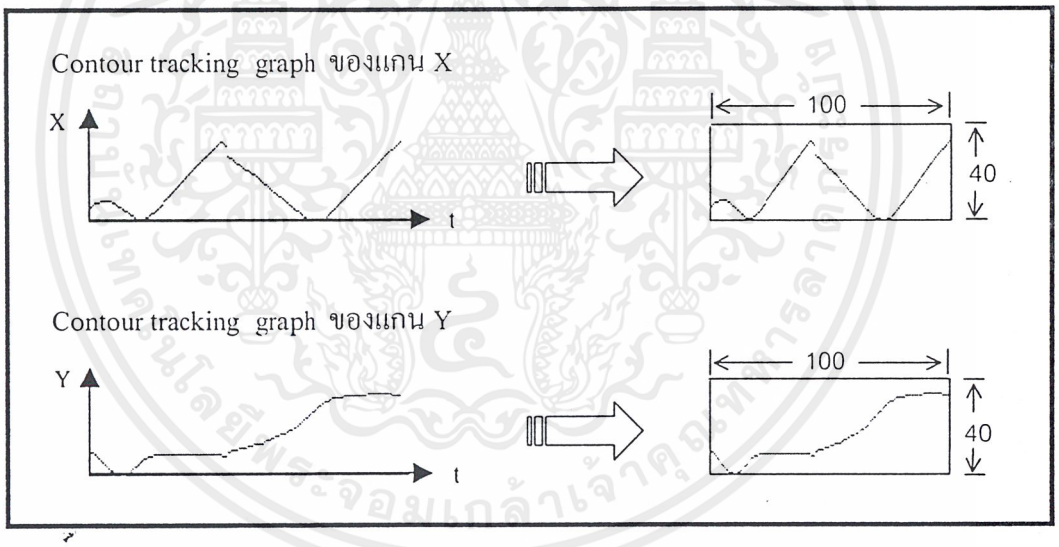
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 5 พล็อต Contour tracking graph

การพล็อต Contour tracking graph จะพล็อตได้หลังจากการหาค่าของผลจากการหาตำแหน่งที่เสถียร ซึ่งหาได้จากสมการ $X = X - \Delta Y / \tan \alpha_{max}$, $Y = Y$ แล้วจึงนำมาพล็อต Contour tracking graph อาจกล่าวได้ว่า การพล็อต Contour tracking graph จะสามารถทำหลังจากการหาและพล็อตค่าของตำแหน่งที่เสถียรจากกระบวนการ Preprocessing ในส่วนของการฝึกสอนด้วย Handwriting

ขั้นที่ 6 Scaling 100 ×40 pixels

การ Scaling 100 ×40 pixels หรือ การปรับขนาดให้มีขนาด 100 ×40 ในหน่วยของ Pixel คือ การปรับขนาด Contour tracking graph จากขนาดด้านแกน X ที่เท่ากับจำนวนข้อมูลทั้งหมด และ จากขนาดด้านแกน Y ที่เท่ากับขนาด 40 pixels ไปเป็นขนาด 100 pixels และ 40 pixels ตามลำดับ เพื่อทำให้ Contour tracking graph มีขนาดเดียวกันหรือเป็นมาตรฐานเดียวกันก่อนเข้าสู่การประมวลผลของ Neural



รูปที่ 2.11 Contour tracking graph ที่ได้ทำการ Scaling 100 ×40 pixels

ขั้นที่ 7 Pixel scanning

Pixel scanning คือ การตัดข้อมูลของแกน X และ Y Contour tracking graph ที่มีการซ้ำซ้อนกันออก ซึ่งข้อมูลที่ซ้ำซ้อนกันนี้เกิดจากการกระทำของการเปลี่ยนแปลงข้อมูลของแกน X และ Y ในที่นี้เกิดจากการปรับขนาด หรือ Scaling

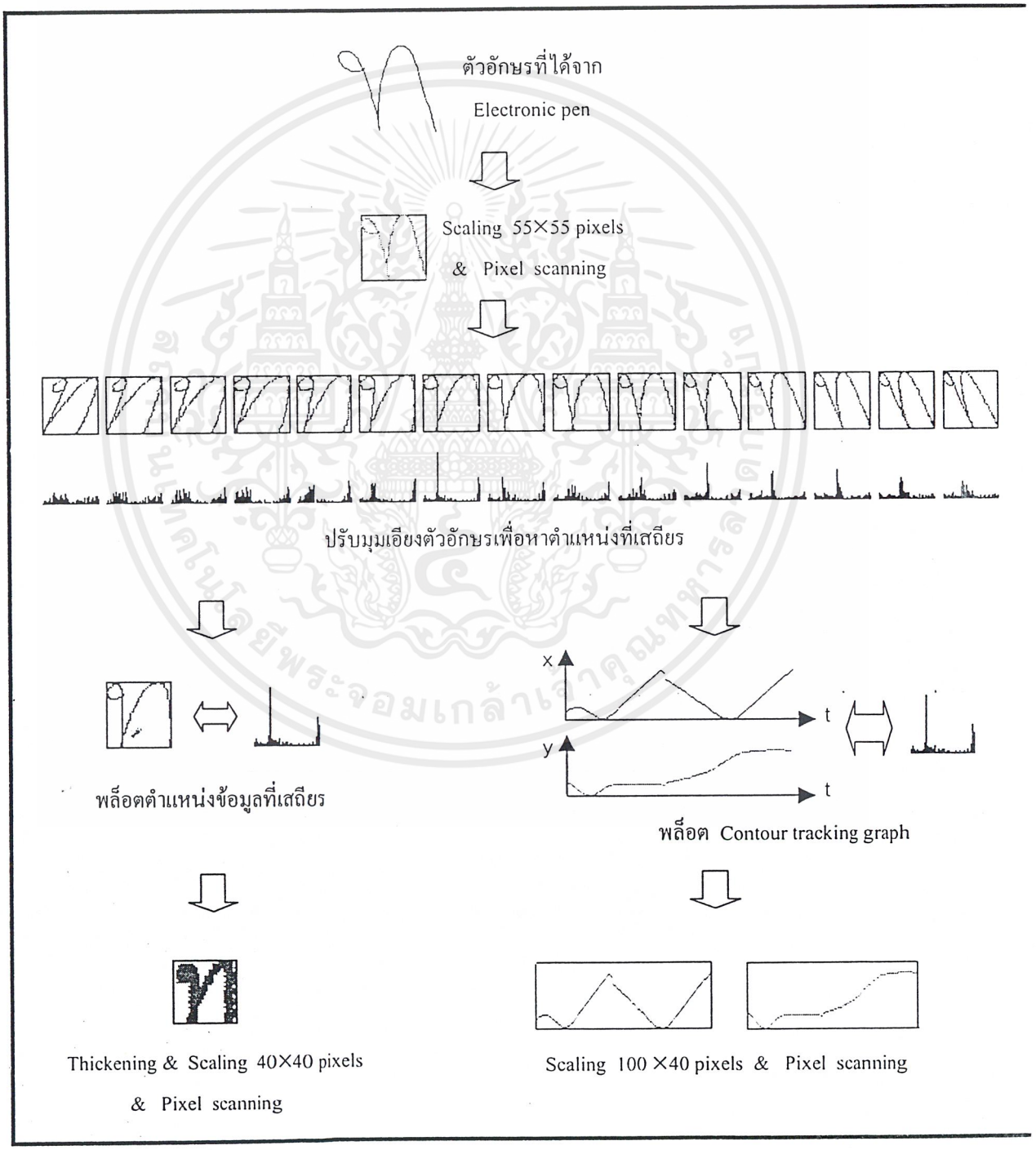
ขั้นที่ 8 เก็บรูปแบบ Contour tracking graph

การเก็บข้อมูลจะเหมือนกับการเก็บข้อมูลจากการฝึกสอนแบบ Handwriting โดยการเก็บข้อมูล Contour tracking graph จะทำการเก็บข้อมูลอาเรย์ (Array) ของตำแหน่งทางด้านแกน Y ของ Contour tracking graph เท่านั้น โดยอาเรย์ (Array) จะมีขนาดเท่ากับ 100 ตำแหน่งทางด้านแกน Y ทั้ง Contour

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

tracking graph ของข้อมูลทางด้านแกน X และแกน Y จะเก็บไว้ใน File สำหรับข้อมูลที่ใช้ในการ Training หรือการฝึกสอนให้กับ Neural และจะนำไปประมวลผลต่อโดย Neural สำหรับข้อมูลที่ต้องการจะทดสอบการรู้จำของ Neural

จากกระบวนการ Preprocessing ในส่วนรูปแบบการฝึกสอนด้วย Handwriting และ Contour tracking graph ที่ได้กล่าวไปแล้วนั้น สามารถนำมาแจกแจงเป็นภาพเพื่อแสดงขั้นตอนต่างๆของทั้ง 2 รูปแบบได้ดังนี้



รูปที่ 2.12 ภาพแสดงกระบวนการ Preprocessing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

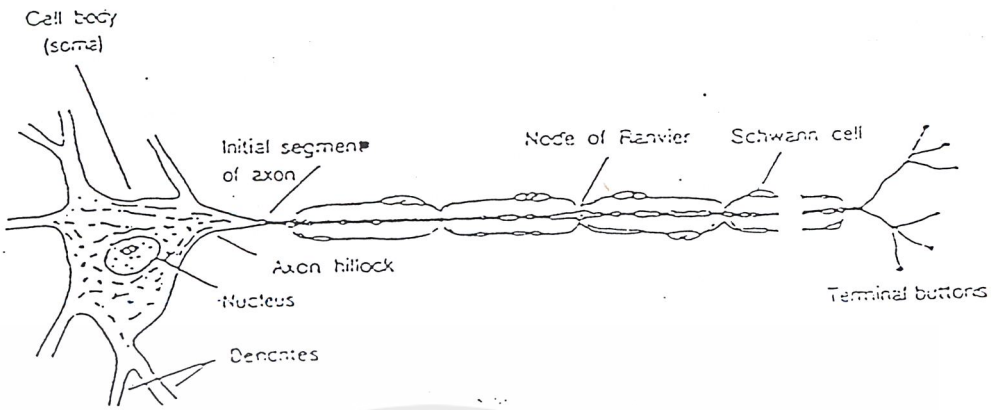
Neural Network

กระบวนการทาง Neural Network หรือ โครงข่ายประสาทเทียมเป็นกระบวนการที่จำลองมาจากกระบวนการทางความคิดของสมองมนุษย์ ซึ่งมีคุณลักษณะพิเศษหลายประการที่เป็นประโยชน์ต่อระบบการประมวลผล ดังนั้นจึงได้มีการพัฒนาและประยุกต์กระบวนการทาง Neural Network ใช้กับงานหลายประเภท ตัวอย่างเช่น งานที่เกี่ยวกับประมวลผลสัญญาณ (Signal Processing) , งานที่เกี่ยวกับการพัฒนาหุ่นยนต์ (Robot) และงานที่เกี่ยวกับการจดจำรูปแบบต่างๆ (Pattern Recognition) เป็นต้น ในโครงงานนี้ เราจึงอาศัยหลักการทำงานของ Neural Network มาใช้ในการรู้จำตัวอักษร หรือ Character Recognition โดยปกติการพัฒนากระบวนการการรู้จำตัวอักษรเราจะใช้โครงข่ายประสาทเทียมแบบทั่วไปที่มีน้ำหนัก หรือ Artificial Neural Network : ANN แต่ในโครงงานนี้ เราจะใช้โครงข่ายประสาทเทียมแบบไร้น้ำหนัก หรือ Weightless Artificial Neural Network : WANN ซึ่งเป็นระบบที่ถูกพัฒนาขึ้นมาใหม่และยังไม่เป็นที่รู้จักกันอย่างแพร่หลาย แต่โครงข่ายประสาทเทียมแบบไร้น้ำหนักนี้มีข้อเด่นหลายประการที่น่าสนใจ อย่างเช่น ความเร็วและกระบวนการที่ง่ายกว่าเมื่อเทียบกับโครงข่ายแบบมีน้ำหนักจึงได้นำมาประยุกต์ใช้ในการรู้จำตัวอักษร ที่ได้จากการเขียนด้วยลายมือ

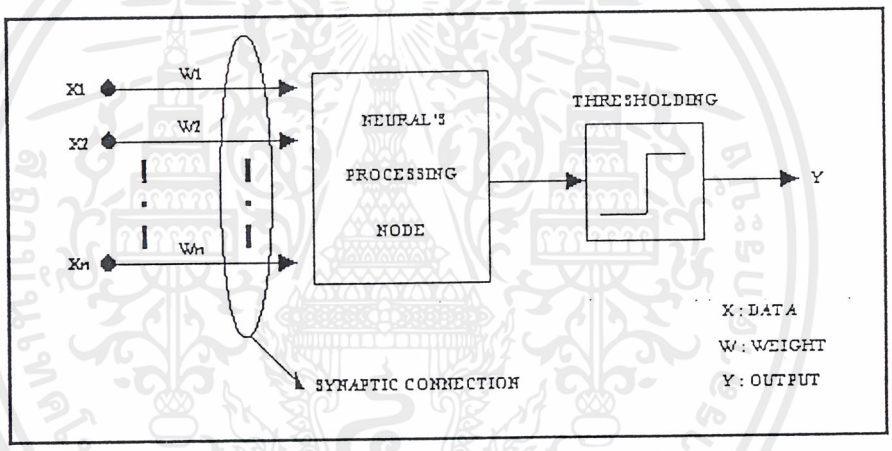
3.1 Artificial Neural Network

โครงสร้างของ Artificial Neural Network นั้นโดยทั่วไปจะประกอบไปด้วยส่วนย่อยๆที่เรียกว่า Neuron หรือ เซลล์ประสาทในทางชีวภาพซึ่งจะมีขาที่ใช้รับส่งข้อมูลหลายขาแล้วแต่การออกแบบ ซึ่งอาจเปรียบเทียบได้กับส่วนของ Synapses ในทางชีวภาพ ที่ทำหน้าที่เป็นทางผ่านให้ข้อมูลหรือข่าวสารสามารถส่งผ่านจาก Neuron ตัวหนึ่งไปยัง Neuron อีกตัวหนึ่ง แต่ละ Synapses จะมีค่าน้ำหนักหรือ Weight ของมันเอง เมื่อรับข้อมูลเข้ามาแต่ละข้อมูลจะต้องนำไปคูณกับค่าน้ำหนักของ Synapses ที่รับข้อมูลนั้นๆ โดยค่าน้ำหนักนั้นอาจได้มาจากการเรียนรู้แบบ Delta Learning Rule [2] คือการเปรียบเทียบกับคำตอบที่ได้จริง กับคำตอบที่ควรจะเป็นแล้วนำข้อผิดพลาดมาเข้าสมการ เพื่อหาค่าน้ำหนักที่เหมาะสม ซึ่งการทำงานของ Artificial Neural Network ในหนึ่งรอบการเรียนรู้นั้นจะต้องผ่านการคำนวณมากมาย ซึ่งแตกต่างจาก Weightless Artificial Neural Network อย่างสิ้นเชิง กล่าวคือ Weightless Artificial Neural Network นั้นจะทำงานในลักษณะของ Boolean Table [3] เหมือนในการสร้างตาราง Logic ซึ่งไม่มีการคูณกับค่าน้ำหนักของ Synapses หรือการเข้าสมการใดๆที่ยุ่งยาก ทำให้ Weightless Artificial Neural Network นั้นมีข้อได้เปรียบในเรื่องความเร็วและกระบวนการที่ง่ายกว่าเมื่อเทียบกับ Artificial Neural Network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 เซลล์ประสาท (Neuron) ทางชีวภาพ



รูปที่ 3.2 Artificial Neural Network

3.2 Weightless Artificial Neural Network

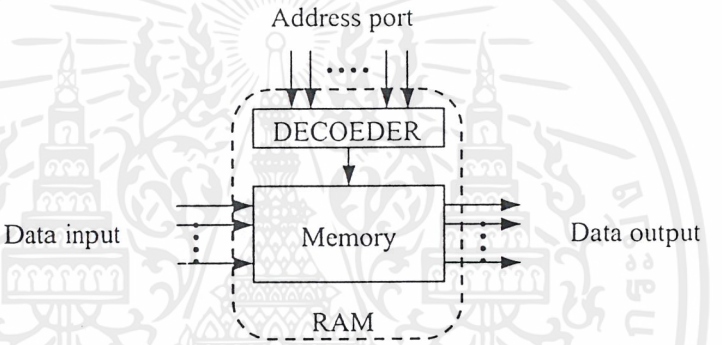
โครงสร้างของ Weightless Artificial Neural Network จะมีลักษณะโครงสร้างเหมือน Artificial Neural Network ต่างกันตรงที่ ไม่มีค่าน้ำหนักหรือ Weight ที่ขา Synapses ซึ่งทำหน้าที่รับส่งข้อมูล ทำให้กระบวนการในการตัดสินใจนั้นง่ายกว่าและเร็วกว่าแบบ Artificial Neural Network มาก จากข้อได้เปรียบที่กล่าวมา เราจึงนำ Weightless Artificial Neural Network มาประยุกต์ใช้ในการจดจำตัวอักษร ซึ่งในการใช้งานนี้ การทำงานของโครงข่ายจะเป็นลักษณะของ Single Layer Feed Forward Neural คือมี 1 ชั้นและไม่มีการป้อนกลับ

ในโครงงานนี้ได้มีการประยุกต์การทำงานของ Weightless Artificial Neural Network ให้เหมาะสมกับรูปแบบการจดจำตัวอักษร จากการทดลองการหารูปแบบของข้อมูลในการฝึกสอน Neural ในภาคการเรียนที่1 ได้สรุปออกมาว่า วิธีการพล็อตกราฟโครงร่างตัวอักษร(Contour Tracking) และแบบลายมือ (Handwriting) เป็นวิธีที่ทำให้เปอร์เซ็นต์การรู้จำของ Neural สูง ดังนั้นในภาคการเรียนนี้จึงได้หาวิธีการปรับปรุงและพัฒนาวิธีการดังกล่าวให้ดียิ่งขึ้น ไม่ว่าจะเป็นการหาตำแหน่งที่ Optimize เพื่อให้ Synapses

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ Neuron จับ และการออกแบบโครงสร้างของ Weightless Artificial Neural Network เองให้ดียิ่งขึ้น เป็นต้น

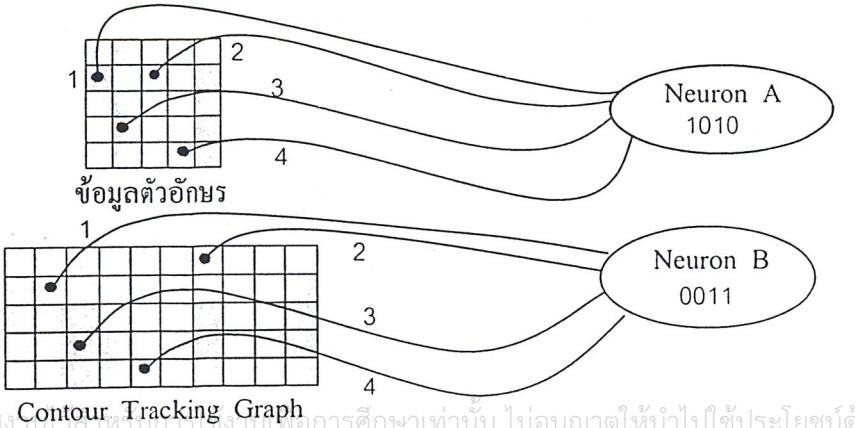
โดยทั่วไปหลักการทำงานของ Weightless Artificial Neural Network ที่ใช้ในโครงงานนี้จะใช้หลักการทำงานเหมือนกับการทำงานของ Random Access Memories (RAMs) กล่าวคือ ในหนึ่งหน่วยของ Neural Network หรือ Neuron จะทำหน้าที่เหมือน RAM ซึ่งเป็นหน่วยความจำที่สามารถอ่านข้อมูลหรือเขียนข้อมูลเข้าไปได้ ซึ่งปกติการเขียนหรือการอ่านข้อมูลจะทำได้โดยการส่ง Address ของข้อมูลที่ต้องการเขียนหรืออ่านไปที่ RAM แล้ว RAM ก็จะการเขียนหรือการอ่านข้อมูลนั้น ซึ่ง Address ของข้อมูลนั้นก็คือ ข้อมูลที่รับเข้ามาจาก Synapses ของ Neuron ซึ่งเป็นข้อมูล Binary ส่วนข้อมูลที่เขียนและอ่านที่ Address นั้นก็คือ ข้อมูลตัวอักษรที่เราต้องฝึกสอน (Training) และข้อมูลที่ได้จากการประมวลผล ตามลำดับ



รูปที่ 3.3 Random Access Memory

3.3 กระบวนการฝึกสอน Neuron

หลักการฝึกสอน Neuron เราจะมองตัวอักษรและ Contour Tracking Graph เป็นภาพๆหนึ่ง และจะกำหนดให้ Pixel ที่มีข้อมูลมีค่าเป็น 1 , Pixel ที่มีไม่มีข้อมูลมีค่าเป็น 0 จากนั้นทำการกำหนดตำแหน่ง Synapses ของแต่ละ Neuron บนภาพตัวอักษรและ Contour Tracking Graph แล้วรับข้อมูลจาก Synapses มาเก็บไว้ที่ Neuron ตัวนั้นๆ สามารถแสดงได้ดังรูปที่ 3.4

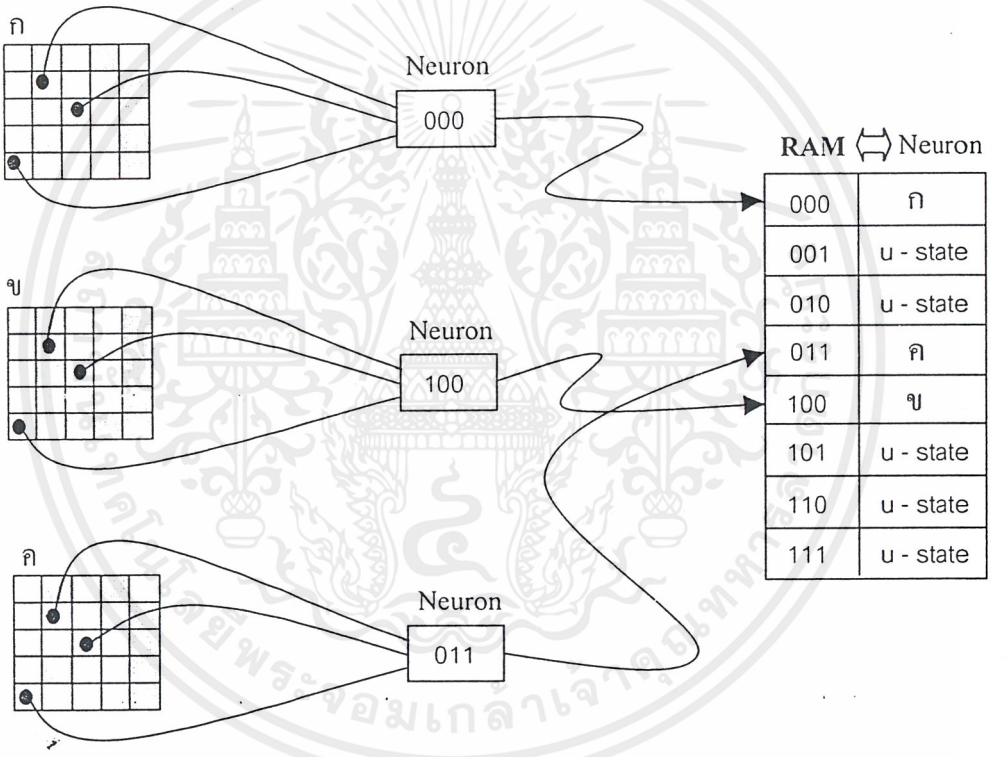


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.4 ภาพแสดงการรับข้อมูลของ Synapses

จากรูปที่ 3.4 Neuron A และ Neuron B สามารถเป็นตัวเดียวกันได้ ซึ่ง Neuron ตัวนั้นจะมี Synapses ทั้งหมด 8 ขา นั่นเอง

จากที่กล่าวมาแล้วว่า การทำงานของ Neuron จะเหมือนการทำงานของ RAM ดังนั้นในการฝึกสอน Neuron จะกำหนดให้ทุกตำแหน่งใน Memory ก่อนการฝึกสอนอยู่ในสภาวะ u – state หรือสภาวะ undefined - state คือ สภาพที่ค่าของข้อมูลใน RAM ยังไม่ถูกกำหนดชัดเจน จากนั้นจึงให้ Synapses ของ Neuron แต่ละตัวจับตามตำแหน่งที่กำหนด แล้วแปลงเป็นรหัส Binary ตามหลักที่กล่าวไปแล้ว เพื่อนำไปเป็นค่าชี้ตำแหน่ง Address ของ RAM ที่จะถูกเขียนด้วยตัวอักษรที่ถูกจับโดย Synapses ของ Neuron นั้นๆ



รูปที่ 3.5 การฝึกสอน Neuron ใน Neural Network โดยใช้คุณสมบัติของ RAM

จากรูป 3.5 Neuron ที่จับภาพของตัวอักษร ก ข ค นั้นคือ Neuron ตัวเดียวกันที่ Synapses จับอยู่ในตำแหน่งเดียวกันทั้ง 3 ตัวอักษร แล้วแปลงตำแหน่งต่างๆเป็นรหัส Binary เมื่อรหัสใดตรงกับ Address ใน RAM ก็จะเก็บตัวอักษรที่จับนั้นตามตำแหน่ง Address ที่ชี้ดังกล่าว สำหรับการจับภาพของ Contour Tracking graph ก็ใช้หลักการเช่นเดียวกัน ซึ่งอาจใช้ Neuron ตัวเดียวกับการจับภาพตัวอักษรก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อสังเกต

จากการที่แต่ละ Synapses ของ Neuron ตัวหนึ่งจับภาพของตัวอักษรหรือภาพของ Contour Tracking graph ซึ่งเป็นไปได้ที่ค่าจากแปลงเป็นรหัส Binary มีค่าซ้ำกัน ทำให้ใน 1 Address ของ RAM มีค่าข้อมูลมากกว่า 1 ค่า ซึ่งไม่เป็นผลดีต่อระบบการประมวลผลเท่าใดนัก ปัญหาเหล่านี้สามารถบรรเทาด้วยการเพิ่มความหนาของข้อมูลตัวอักษร (Thickening) เพื่อให้ Pixel ที่มีข้อมูล (Binary code = 1)และไม่มีข้อมูล (Binary code = 0) มีค่าพอกัน สามารถลดการชนกันของข้อมูล [1] ได้

จากประโยคที่ว่า " แต่ละ Synapses ของ Neuron จะจับตามตำแหน่งที่กำหนดบนภาพของตัวอักษรหรือภาพของ Contour Tracking graph " เราจะมาพิจารณาคำว่า " ตำแหน่งที่กำหนด " ว่ามีความหมายอย่างไรในโครงงานนี้

การกำหนดตำแหน่งขาของ Neural แต่ละตัวสามารถพิจารณาการกำหนดได้หลายวิธี อาจจะอยู่ในตำแหน่งที่ซ้ำกันหรือไม่ซ้ำกันก็ได้ แต่ในโครงงานนี้เราจะกำหนดให้ Synapses ของ Neuron แต่ละตัวจับในตำแหน่งที่แตกต่างกัน และจะใช้วิธีการจับตำแหน่งอยู่ 2 แบบ คือ แบบ Random และแบบวิธีการหาตำแหน่งที่ Optimize ของภาพตัวอักษรและภาพของ Contour Tracking graph ตำแหน่งที่ Optimize คือ ตำแหน่งที่เหมาะสม หรือ ตำแหน่งที่มีการเปลี่ยนแปลงของข้อมูลจากการฝึกสอนมากที่สุด หรือกล่าวอีกนัยนึ่งคือ ตำแหน่งที่ Synapses ของ Neuron มีการเปลี่ยนแปลงข้อมูลเป็น 1 และ 0 พอๆ กันในแต่ละข้อมูลการฝึกสอน หาได้โดยใช้วิธีการดังนี้

1. มองข้อมูลตัวอักษรและข้อมูล Contour Tracking graph เป็นภาพๆหนึ่ง และพิจารณาแยกกัน

2. กำหนดแบบข้อมูลที่จะทำการฝึกสอน Neuron ให้ครบทุกตัวอักษรทุกสระ ซึ่งในโครงงานนี้จะทำการฝึกสอน Neuron ตัวอักษรละประมาณ 2 แบบ

3. นำข้อมูลที่ทำการฝึกสอน Neuron แต่ละตัวมาพิจารณา หา Pixel ของข้อมูลที่มีการเปลี่ยนแปลงของข้อมูล Binary เป็น 1 และ 0 พอๆ กันจากการพิจารณาตัวอักษรทั้งหมด ซึ่งสามารถทำได้โดยให้ทุก Pixel ที่มีข้อมูลภาพมีค่าเป็น +1 และให้ทุก Pixel ที่ไม่มีข้อมูลภาพมีค่าเป็น –1 สำหรับการพิจารณาตัวอักษรแต่ละตัว เรียกว่าการแปลงแบบ Binary เป็น Bipolar จากนั้นนำข้อมูลที่ทำการฝึกสอน Neuron ทุกแบบมาพิจารณาตามหลักดังกล่าว

4. รวมค่าของแต่ละ Pixel ของทุกข้อมูลที่ทำการฝึกสอน Neuron ผลรวมของ Pixel ใดที่มีค่าเท่ากับศูนย์หรือใกล้เคียงศูนย์ คือตำแหน่งที่ Optimize

ขั้นตอนการหาตำแหน่งที่ Optimize สามารถอธิบายได้ดังรูปที่ 3.6 จากรูปจะได้ตำแหน่งที่ Optimize ที่สุดคือตำแหน่งที่ให้ค่าจากการ Optimize = ± 0 ซึ่งได้แก่คู่ลำดับที่ (0,0) และ (0,1) ซึ่งเป็นตำแหน่งที่มีการเปลี่ยนแปลงของค่าของข้อมูล 1 และ 0 (มีข้อมูลและไม่มีข้อมูล) พอๆ กันจากการพิจารณาข้อมูลที่ใช้ฝึกสอน Neuron

ตำแหน่งที่ Optimize เหล่านี้ก็จะเป็นตำแหน่งที่ให้ Synapses จับ ส่วนตำแหน่งที่เหลือก็จะทำการ Random ต่อไป นั่นคือถ้าสมมุติว่า ระบบมี Neuron 2 ตัว และ Neuron แต่ละตัวมี Synapses 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ขา ดังนั้นจะมีตำแหน่งจับทั้งหมด 8 ตำแหน่ง ซึ่งจะต้องไม่มากกว่าจำนวน Pixel ทั้งหมดของกรอบข้อมูล และจากการที่กรอบข้อมูลมีขนาด 4 × 4 นั้นคือจะมีจำนวน Pixel ทั้งหมด 16 Pixel เราก็จะให้ Synapses ทั้ง 8 ขาจับตำแหน่งที่ Optimize ที่สุดก่อนแล้วให้จับตำแหน่งที่ Optimize รองลงมา หรือส่วนที่เหลือจะใช้วิธีการ Random ส่วนการหาตำแหน่งที่ Optimize ของข้อมูล Contour Tracking graph ก็ใช้หลักการเดียวกัน

ตำแหน่ง	ค่าของตำแหน่ง				
	แบบที่ 1	แบบที่ 2	แบบที่ 3	แบบที่ 4	รวม
(0 , 0)	-1	+1	-1	+1	0
(0 , 1)	+1	-1	+1	-1	0
(0 , 2)	-1	-1	-1	+1	-2
(0 , 3)	-1	-1	-1	+1	-2
(1 , 0)	+1	+1	-1	+1	+2
(1 , 1)	-1	+1	-1	-1	-2
(1 , 2)	+1	+1	+1	-1	+2
(1 , 3)	+1	+1	-1	+1	+2
(2 , 0)	+1	-1	+1	+1	+2
(2 , 1)	-1	-1	-1	-1	-4
(2 , 2)	-1	-1	-1	-1	-4
(2 , 3)	-1	+1	+1	+1	+2
(3 , 0)	-1	+1	+1	+1	+2
(3 , 1)	+1	+1	+1	+1	+4
(3 , 2)	+1	+1	+1	+1	+4
(3 , 3)	+1	+1	+1	+1	+4

รูปที่ 3.6 ภาพแสดงการหาตำแหน่งที่ Optimize

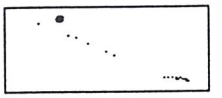
ในโครงงานนี้เราได้ทำการหาตำแหน่งที่ให้ Synapses ของ Neuron จับทั้งแบบการหาตำแหน่งที่ Optimize และจากการหาแบบ Random ซึ่งสามารถแสดงได้ดังรูปที่ 3.7

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

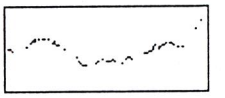
42707

Opimize Value = ±0

ตำแหน่งที่ Opimize ของข้อมูลตัวอักษร



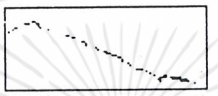
ตำแหน่งที่ Opimize ของข้อมูล Contour Tracking graph ด้านแกน X



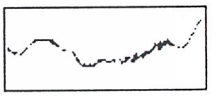
ตำแหน่งที่ Opimize ของข้อมูล Contour Tracking graph ด้านแกน Y

Opimize Value = ±2

ตำแหน่งที่ Opimize ของข้อมูลตัวอักษร



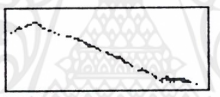
ตำแหน่งที่ Opimize ของข้อมูล Contour Tracking graph ด้านแกน X



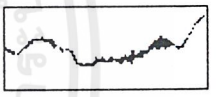
ตำแหน่งที่ Opimize ของข้อมูล Contour Tracking graph ด้านแกน Y

Opimize Value = ±5

ตำแหน่งที่ Opimize ของข้อมูลตัวอักษร



ตำแหน่งที่ Opimize ของข้อมูล Contour Tracking graph ด้านแกน X

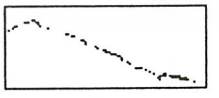


ตำแหน่งที่ Opimize ของข้อมูล Contour Tracking graph ด้านแกน Y

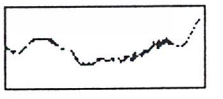
รูปที่ 3.7 ภาพแสดงตำแหน่งที่ Opimize ตามค่าจากการ Opimize ค่าต่างๆ

ในโครงงานนี้เราใช้ตำแหน่งที่ค่าจากการ Optimize = ± 3 และตำแหน่งที่ได้จากการ Random ให้ Synapses ของ Neuron จับ แสดงดังรูปที่ 3.8

ตำแหน่งที่ Optimize ของข้อมูลตัวอักษร



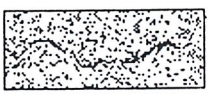
ตำแหน่งที่ Optimize ของข้อมูล Contour Tracking graph ด้านแกน X



ตำแหน่งที่ Optimize ของข้อมูล Contour Tracking graph ด้านแกน Y

ตำแหน่งที่ให้ Synapses จับ ของข้อมูลตัวอักษร

ตำแหน่งที่ให้ Synapses จับ ของข้อมูล Contour Tracking graph ด้านแกน X



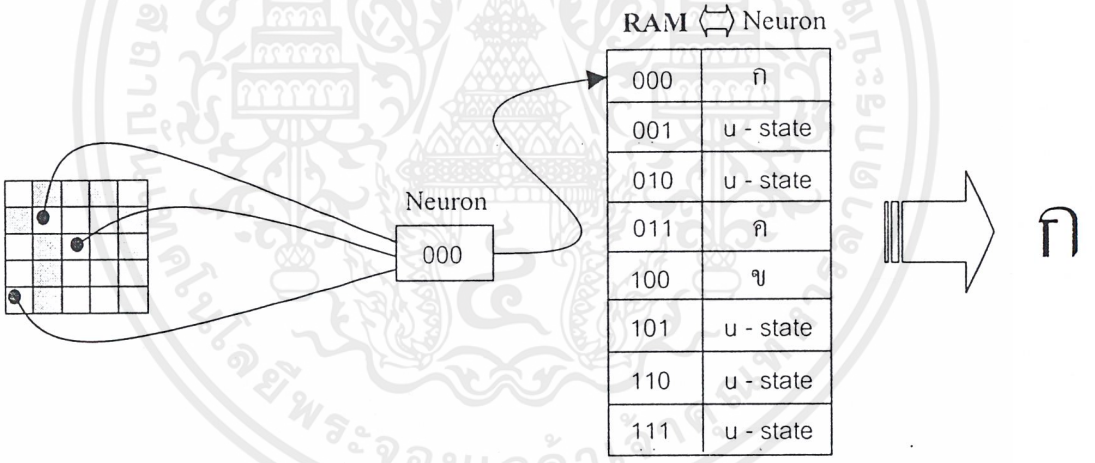
ตำแหน่งที่ให้ Synapses จับ ของข้อมูล Contour Tracking graph ด้านแกน Y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

รูปที่ 3.8 ภาพแสดงตำแหน่งที่ Optimize และตำแหน่งที่ Random ในโครงงานนี้

3.4 กระบวนการประมวลผลของระบบ Neural Network

จากที่ได้กล่าวไปแล้วว่าการทำงานของ Neuron จะเหมือนการทำงานของ RAM ดังนั้นในการกระบวนการประมวลผลของ Neuron สามารถใช้หลักของการอ่านข้อมูลจาก RAM ซึ่งทำได้โดยกำหนดค่า Address ข้อมูลที่ต้องการอ่านไปที่ RAM แล้ว RAM ก็จะส่งข้อมูลที่ต้องการนั้นออกมา จากหลักดังกล่าวสามารถนำมาใช้ในหลักการประมวลผลได้โดย กำหนดให้ตำแหน่ง Address ใน RAM เป็นค่าที่ได้จากตำแหน่งที่ให้ Synapses ของ Neuron แต่ละตัวจับตามตำแหน่งที่กำหนด แล้วแปลงเป็นรหัส Binary ตามหลักที่กล่าวไปแล้ว เพื่อนำไปเป็นค่าชี้ตำแหน่ง Address ของ RAM ที่ต้องการอ่านค่าซึ่งก็คือค่าที่ได้จากการประมวลผลนั้นเอง โดยตำแหน่งที่กำหนดให้ Synapses ของ Neuron แต่ละตัวจับก็คือตำแหน่งเดียวกับตำแหน่งของ Neuron แต่ละตัวจับจากในกระบวนการฝึกสอน (ได้จากการ Optimize และ Random) แต่ถ้าข้อมูลในตำแหน่ง Address ของ RAM ที่ต้องการอ่านอยู่ในสภาวะ u - state ก็จะต้องนำค่า Address นั้นมาเปรียบเทียบหาค่า Classes ที่ใกล้เคียงที่สุดทาง Hamming Distance แล้วจึงนำข้อมูลใน Address ที่ใกล้เคียงนั้นมาเป็นค่าที่ได้การประมวลผล



รูปที่ 3.9 การประมวลผลของ Neuron ใน Neural Network โดยใช้คุณสมบัติของ RAM

จากการฝึกสอนของ Neuron ใน Neural Network จากภาพตัวอย่างในรูปที่ 3.5 เรามาดูการประมวลผลของ Neuron นั้นในรูปที่ 3.9 Neuron ที่ใช้ในการฝึกสอนและการประมวลผล คือ Neuron ตัวเดียวกัน นั่นคือมีตำแหน่งที่ให้ Synapses จับตำแหน่งเดียวกัน เมื่อนำ Synapses ของ Neuron ตัวนั้นไปจับภาพที่ต้องการทดสอบการประมวลผลตามตำแหน่งที่กำหนด แล้วแปลงเป็นรหัส Binary ตามหลักที่กล่าวไปแล้ว ซึ่งจะได้ค่าเท่ากับ 000 เพื่อนำไปเป็นค่าชี้ตำแหน่ง Address ของ RAM ที่ต้องการจะอ่านค่าซึ่งก็คือค่าที่ได้จากการประมวลผลนั่นเอง จากตัวอย่างนี้จะได้ ก เป็นคำตอบที่ได้จากการประมวลผล แต่ถ้าค่าที่ได้จากการแปลงเป็นรหัส Binary มีค่าเท่ากับ 001 ซึ่งข้อมูลในตำแหน่ง Address

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ RAM ที่ต้องการอ่านนั้นอยู่ในสภาวะ u - state ก็จะต้องนำค่า Address นั้นมาเปรียบเทียบหาค่า Classes ที่ใกล้เคียงที่สุด แล้วจึงนำข้อมูลใน Address ที่ใกล้เคียงนั้นมาเป็นค่าที่ได้การประมวลผล ในโครงงานนี้เราใช้วิธีการเปรียบเทียบหาค่า Address ที่ใกล้เคียงที่สุดโดยวิธี Hamming Distance หรือวิธี H.M. Distance ซึ่งจะเป็นวิธีเปรียบเทียบกันบิตต่อบิต โดยจะนำข้อมูลที่ต้องการเปรียบเทียบ (2 ข้อมูล) มา Excusive-or กัน แล้วนำบิตจากคำตอบมีค่าเป็น 1 มาบวกกันก็จะได้ค่า Hamming Distance ของข้อมูลที่เปรียบเทียบนั้น ซึ่งถ้ามีค่ามากก็จะต่างกันมาก จากตัวอย่างที่กำหนดให้ค่าที่ได้จากการแปลงเป็นรหัส Binary มีค่าเท่ากับ 001 เราจะนำมาค่านั้นเปรียบเทียบกับค่า Address ของข้อมูลใน Classes ต่างๆ อีก ได้แก่ตำแหน่ง Address ที่ 000 , 011 , 100

ข้อมูลที่ทดสอบที่มีรหัส Binary เท่ากับ 001 เปรียบเทียบกับข้อมูลที่ใช้ฝึกสอนที่มีค่า Address เท่ากับ 000

ข้อมูลที่ทดสอบ 001

ข้อมูลที่ฝึกสอน 000

Excusive-or.....001

H.M. Distance.....1

ข้อมูลที่ทดสอบที่มีรหัส Binary เท่ากับ 001 เปรียบเทียบกับข้อมูลที่ใช้ฝึกสอนที่มีค่า Address เท่ากับ 011

ข้อมูลที่ทดสอบ 001

ข้อมูลที่ฝึกสอน 011

Excusive-or.....010

H.M. Distance.....1

ข้อมูลที่ทดสอบที่มีรหัส Binary เท่ากับ 001 เปรียบเทียบกับข้อมูลที่ใช้ฝึกสอนที่มีค่า Address เท่ากับ 100

ข้อมูลที่ทดสอบ 001

ข้อมูลที่ฝึกสอน 100

Excusive-or.....101

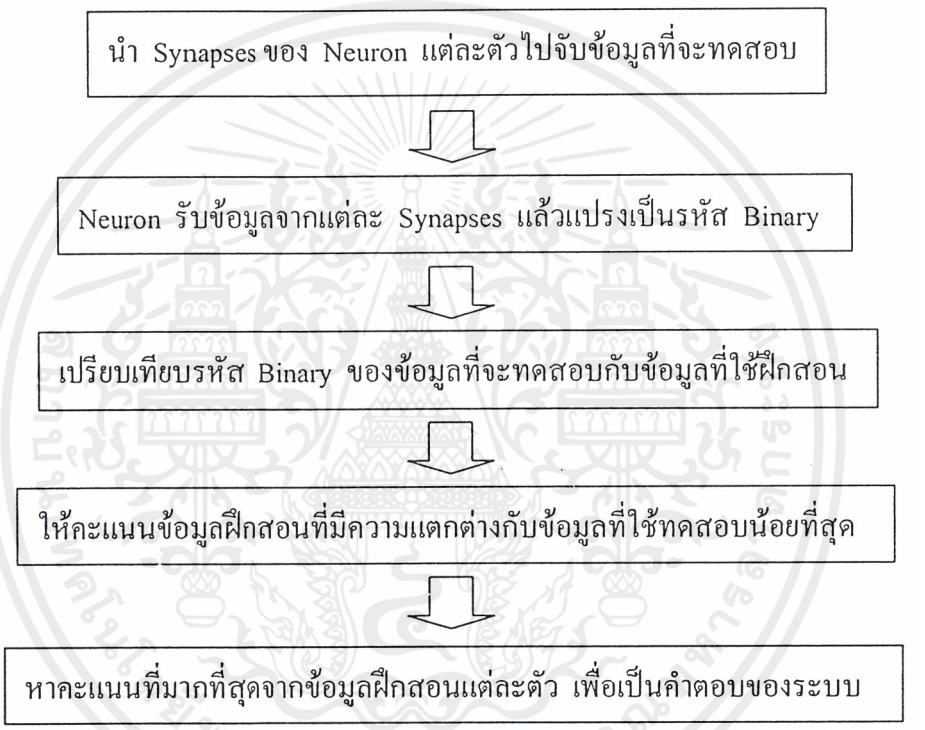
H.M. Distance.....2

จากการเปรียบเทียบด้วยวิธีการหาค่า H.M. Distance จะได้คำตอบที่น้อยที่สุดคือ 1 ซึ่งมีอยู่ 2 ค่า นั้นคือที่ตำแหน่ง Address 011 , 100 ที่ชี้ข้อมูล ก และ ค อยู่ ดังนั้นการประมวลผล Neuron ตัวนี้จะให้คำตอบ 2 ค่า แต่คำตอบนี้ยังไม่ใช้คำตอบของระบบเพราะยังมี Neuron ตัวอื่นๆอีก ที่อาจให้คำตอบเป็นอย่างอื่น ซึ่งการประมวลผลของ Neuron ตัวอื่นๆ ก็ใช้หลักเดียวกับที่แสดงในตัวอย่างข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโครงงานนี้เราจะใช้หลักการตัดสินใจของระบบ Neural Network โดยการให้ 1 Neuron แทนด้วยคะแนน 1 คะแนนของข้อมูลหรือตัวอักษรที่ได้จากการประมวลผลของ Neuron นั้นๆ ดังนั้น ถ้าให้ Neural Network มี Neuron ทั้งหมด x ตัว ก็สามารถกล่าวได้ว่าระบบจะมีคะแนนเต็มทั้งหมด x คะแนน ซึ่งแต่ละคะแนนก็จะกระจายไปตามข้อมูลที่ได้จากการประมวลผลของแต่ละ Neuron และข้อมูลหรืออักษรที่มีคะแนนสูงที่สุดก็จะเป็นคำตอบของระบบนั้นเอง

ขั้นตอนการประมวลผล Weightless Neural Network ที่ได้กล่าวในข้างต้นสามารถเขียนสรุปเป็นขั้นตอนต่างๆ อย่างคร่าวๆได้ดังนี้



รูปที่ 3.10 บล็อกไดอะแกรมแสดงการทำงานอย่างคร่าวๆของ Neural Network

จากบล็อกไดอะแกรมสามารถอธิบายได้ดังนี้

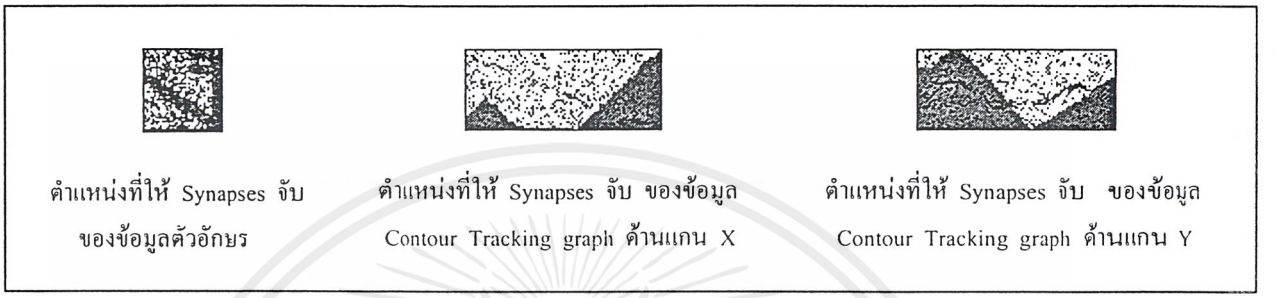
ขั้นที่ 1 นำ Synapses ของ Neuron แต่ละตัวไปจับข้อมูลที่จะทดสอบ

ตำแหน่งที่กำหนดที่ให้ Synapses ของ Neuron แต่ละตัวจับก็คือตำแหน่งเดียวกับตำแหน่งของ Neuron แต่ละตัวจับจากการฝึกสอน ซึ่งได้จากการหาตำแหน่งที่ Optimize และได้จากการ Random ตำแหน่งดังกล่าวแสดงดังรูปที่ 3.8 ซึ่งโครงงานนี้เราจะใช้ Neuron ทั้งหมด 100 ตัว ตัวหนึ่งมี Synapses 24 ขา สำหรับจับภาพข้อมูลตัวอักษร 8 ขา สำหรับจับภาพข้อมูล Contour Tracking graph ในแต่ละแกนของการพล็อต แกนละ 8 ขา นั่นคือระบบจะมีคะแนนเต็มจากการประมวลผลทั้งหมด 100 คะแนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 2 Neuron รับข้อมูลจากแต่ละ Synapses แล้วแปลงเป็นรหัส Binary

การแปลงเป็นรหัส Binary จากการรับข้อมูลของ Synapses แต่ละ Synapses ของ Neuron คือ เราจะมองข้อมูลตัวอักษรและข้อมูล Contour Tracking Graph เป็นภาพๆหนึ่ง และจะกำหนดให้ Pixel ที่มีข้อมูลอยู่มีค่าเป็น 1 , Pixel ที่มีไม่มีข้อมูลอยู่มีค่าเป็น 0



รูปที่ 3.11 ภาพแสดงตำแหน่งของ Synapses บนข้อมูลต่างๆ

จากรูปที่ 3.11 ตำแหน่งจุดต่างๆที่อยู่บนข้อมูล เราจะกำหนดให้ตำแหน่งตรงจุดนั้นมีค่าเป็น 1 และตำแหน่งจุดที่ไม่ได้อยู่บนข้อมูล เราจะกำหนดให้ตำแหน่งตรงจุดนั้นมีค่าเป็น 0

ขั้นที่ 3 เปรียบเทียบรหัส Binary ของข้อมูลที่จะทดสอบกับข้อมูลที่ใช้ฝึกสอน

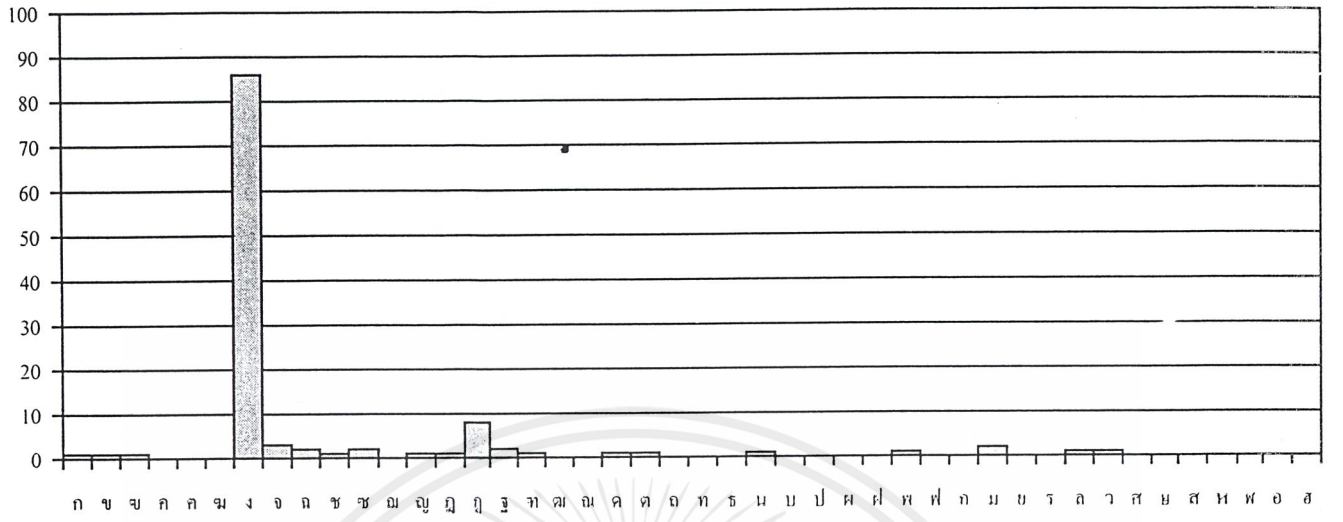
การเปรียบเทียบรหัส Binary ของข้อมูลที่จะทดสอบกับข้อมูลที่ใช้ฝึกสอนจะใช้วิธีหา Hamming Distance หรือวิธี H.M. Distance ซึ่งจะเป็นวิธีเปรียบเทียบกันบิตต่อบิต โดยจะนำข้อมูลที่ต้องการเปรียบเทียบมา Excusive-or กัน แล้วนำบิตจากคำตอบมีค่าเป็น 1 มาบวกกันก็จะได้ค่า Hamming Distance ของข้อมูลที่เปรียบเทียบนั้น ซึ่งถ้ามีค่ามากก็จะต่างกันมาก

การเปรียบเทียบนี้จะเปรียบเทียบข้อมูลที่จะทดสอบกับข้อมูลที่ใช้ฝึกสอนที่ถูกจับด้วย Synapses ที่ตำแหน่งเดียวกัน หรือใช้ Neuron ตัวเดียวกันนั่นเอง

ขั้นที่ 4 ให้คะแนนข้อมูลฝึกสอนที่มีความแตกต่างกับข้อมูลที่ใช้ทดสอบน้อยที่สุด

จากการเปรียบเทียบด้วยวิธีการหาค่า Hamming Distance ซึ่งค่า Hamming Distance ที่มีค่าน้อยที่สุดก็จะเป็นค่าที่บอกว่าข้อมูลที่ใช้ทดสอบมีค่าใกล้เคียงกับข้อมูลที่ใช้ฝึกสอนตัวนั้นมากที่สุด เราก็จะให้คะแนนข้อมูลฝึกสอนตัวนั้น 1 คะแนน ซึ่งข้อมูลฝึกสอนที่ได้จากการเปรียบเทียบนั้นอาจมีมากกว่าหนึ่งข้อมูล เราก็จะให้คะแนนข้อมูลฝึกสอนทั้งหมดที่ได้นั้นข้อมูลละ 1 คะแนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 กราฟแสดงคะแนนของแต่ละตัวอักษรจากการทดสอบตัวอักษร ง

ขั้นที่ 5 หาคะแนนที่มากที่สุดจากข้อมูลฝึกสอนแต่ละตัว เพื่อเป็นคำตอบของระบบ

โครงงานนี้เรากำหนดให้ Neuron 1 ตัวแทนด้วยคะแนน 1 คะแนน ดังนั้นถ้าให้ระบบ Neural Network มี Neuron ทั้งหมด x ตัว ก็สามารถกล่าวได้ว่าระบบจะมีคะแนนเต็มทั้งหมด x คะแนน จากการประมวลผลของ Neuron ทุกตัวในระบบ แต่ละตัวก็จะให้ผลการประมวลผลแตกต่างกัน เป็นคะแนน 1 คะแนนให้กับข้อมูลฝึกสอนที่มีความใกล้เคียงกับข้อมูลที่ใช้ทดสอบมากที่สุด ดังนั้นแต่ละคะแนนของระบบก็จะกระจายไปตามผลจากการประมวลผลของ Neuron แต่ละตัว และคำตอบของระบบก็คือข้อมูลฝึกสอนที่มีคะแนนสูงที่สุด จากการรวมคะแนนที่ให้โดย Neuron แต่ละตัวใน

ในโครงงานนี้เราใช้ Neural Network จดจำตัวอักษร ก – ฮ พร้อมด้วยสระต่างๆ อันได้แก่ อะ , อา , อิ , อี , อึ , อื , โอ , อำ , ไอ , ใอ , เอ , แอ , อุ และ อู วรรณยุกต์ต่างๆ ได้แก่ อ่ , อ้ , อ๊ และ อ๋ เครื่องหมายต่างๆ ได้แก่ อ์ , อั , อ็ อักขระพิเศษ ได้แก่ ฯ , ๆ , ฤ และ ฦ ซึ่งประสิทธิภาพหรือเปอร์เซ็นต์การจดจำจะแสดงในบทต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองการรู้จำตัวอักษร

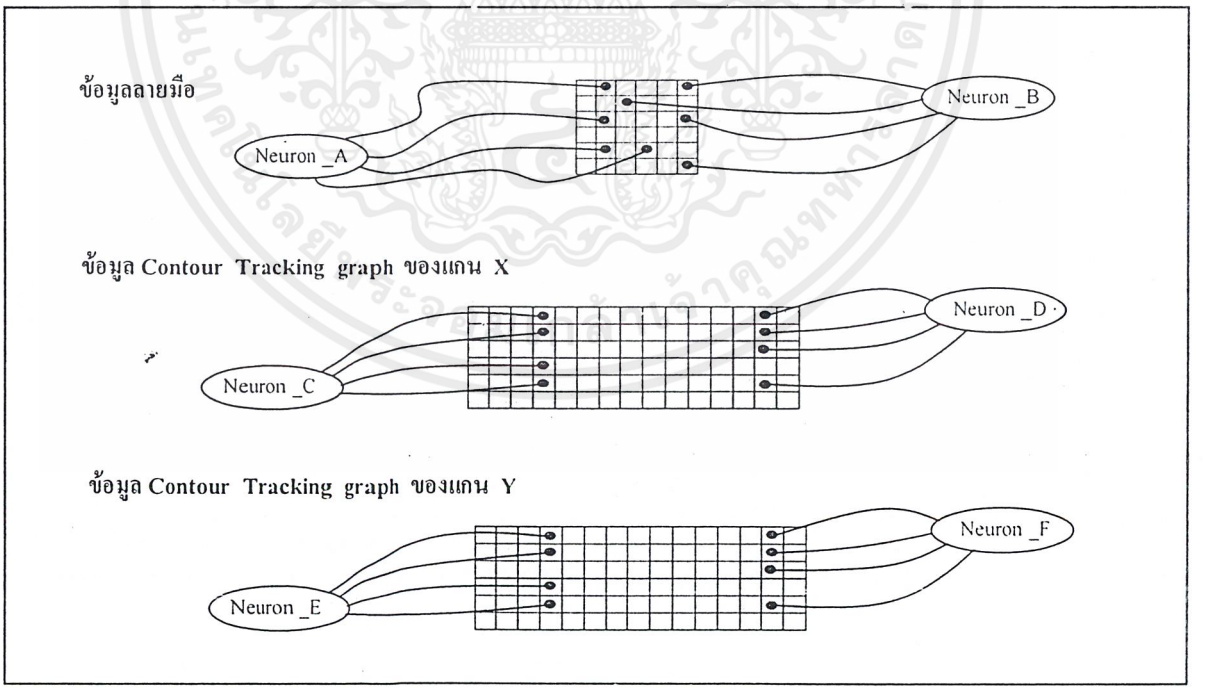
ในโครงงานนี้เราได้ทำการทดลองหาคุณสมบัติของ Neuron ที่เหมาะสมที่ใช้ในการจดจำภาพของตัวอักษรและภาพ Contour Tracking graph โดยทดสอบจำนวน Synapses ที่เหมาะสมของ Neuron ตลอดจนจำนวน Neuron ทั้งหมดที่ใช้ในระบบ เพื่อหารูปแบบที่ให้ประสิทธิภาพการรู้จำที่สูงที่สุด ในบทนี้จะแสดงผลการทดลองจากวิธีการต่างๆ พร้อมทั้งสรุปวิธีที่ดีที่สุดที่ได้

ในโครงงานนี้เราแบ่งรูปแบบการทดลองออกเป็นดังนี้

<u>การทดลองที่ 1</u> ทำการเปรียบเทียบจำนวน Synapses ของ Neuron แต่ละตัว

โดยกำหนดให้ Neuron ที่ใช้ในการจดจำภาพของตัวอักษร 100 ตัวและภาพของ Contour Tracking graph ของแกน X ,Y อย่างละ 100 ตัว และกำหนดตำแหน่ง Synapses ของ Neuron หนึ่งตัวที่จับภาพ Contour Tracking graph จับในตำแหน่งข้อมูล ณ. เวลา (t) หนึ่งเท่านั้น

การทดลองจะกระทำการฝึกสอน Neuron โดยใช้ตัวอักษร ก - ฮ ต่างกัน 2 รูปแบบ แล้วหาเปอร์เซ็นต์การรู้จำจากการทดสอบจำนวน 20 ครั้งในแต่ละตัวอักษร โดยตัวอักษรที่ทดสอบนั้นจะประกอบด้วยหลายลายมือจากผู้ทดสอบหลายคนด้วยกัน



รูปที่ 4.1 ภาพแสดงการจับตำแหน่งของ Synapses ในการทดลองที่ 1

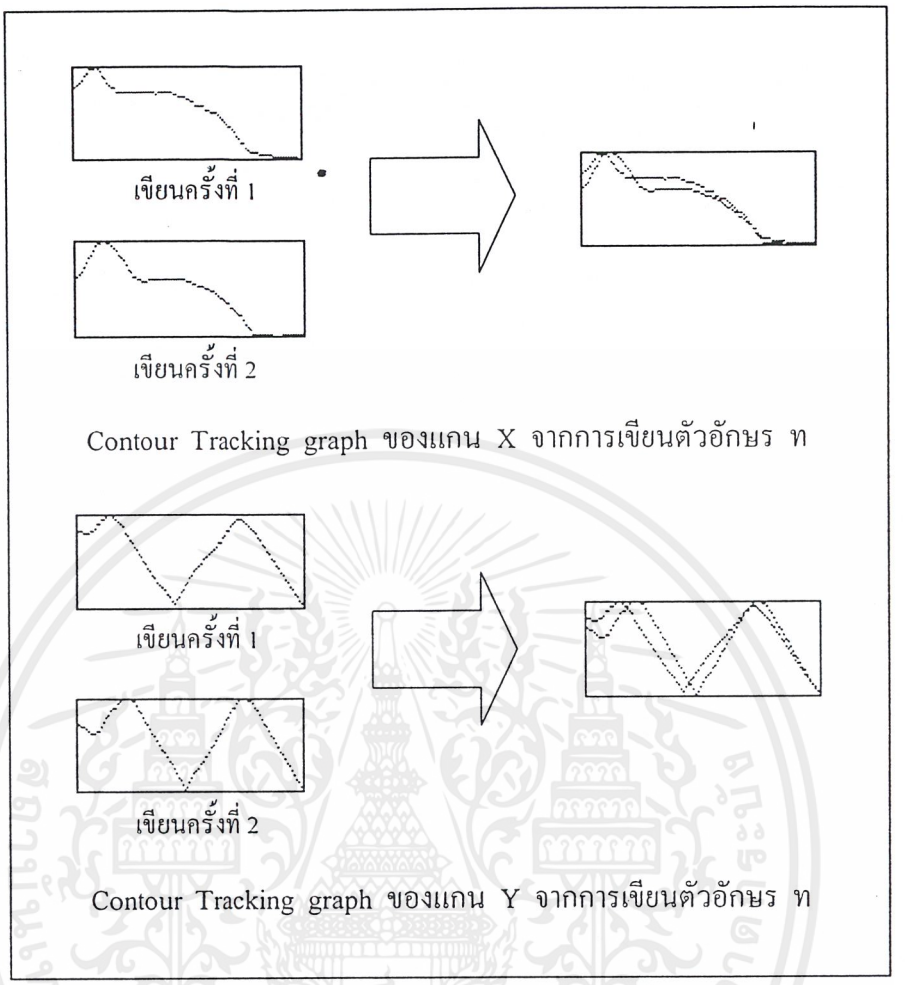
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 เปอร์เซ็นต์การรู้จำตัวอักษร ที่มีตัวแปรเป็นจำนวน Synapses ของ Neuron

จำนวน Synapses ต่อ Neuron 1 ตัว ภาพตัวอักษร/ภาพ Contour Tracking graph	เปอร์เซ็นต์การรู้จำตัวอักษร (%)ของระบบ
4 / 8	82.50
4 / 16	73.32
4 / 20	67.56
8 / 8	92.20
8 / 16	89.50
8 / 20	88.40
16 / 8	85.23
16 / 16	81.98
16 / 20	78.30

จากตารางที่ 4.1 จำนวน Synapses ต่อ Neuron ของภาพตัวอักษร/ภาพ Contour Tracking graph ที่มีค่าเท่ากับ 8 / 8 ให้ค่าเปอร์เซ็นต์การรู้จำตัวอักษรสูงที่สุดคือ 92.20% จากคะแนนของภาพ Contour Tracking graph ที่มีคะแนนรวมทั้งหมด 200 คะแนนจากคะแนนเต็ม 300 คะแนน ซึ่งเป็นคะแนนส่วนใหญ่ของระบบนั้น จะเป็นตัวกำหนดเปอร์เซ็นต์การรู้จำของตัวอักษร จากตาราง ภาพ Contour Tracking graph ที่มีค่าเท่ากับ 8 จะให้ค่าเปอร์เซ็นต์การรู้จำอยู่ในช่วง 82.50 – 92.20 ซึ่งถือว่าเป็นช่วงที่สูง สามารถอธิบายได้จากการที่กำหนดให้ตำแหน่ง Synapses ของ Neuron จับในตำแหน่งข้อมูล ณ. เวลา (t) หนึ่งเท่านั้นจากกรอบข้อมูลที่มีขนาด 40 Pixels นั้นคือมีการจับตำแหน่งข้อมูล 20 % ของข้อมูลทั้งหมดซึ่งเป็นค่าที่เหมาะสมกับการเปลี่ยนแปลงของกราฟในช่วงเวลาหนึ่งๆ สามารถแสดงดังรูป 4.2 ดังนั้นการจับตำแหน่งข้อมูลในเปอร์เซ็นต์ที่สูงเกินไปจึงไม่เป็นผลดีกับการรู้จำของระบบนัก ส่วนคะแนนจากการภาพข้อมูลตัวอักษรที่มี 100 คะแนนจะเป็นตัวช่วยเพิ่มความมั่นใจให้กับระบบ โดยจะเป็นตัวบอกรายละเอียดส่วนย่อยของข้อมูลที่มีความคล้ายกันมาก เช่น ตัว ข กับ บ , ถ กับ ภ เป็นต้น ดังนั้นเปอร์เซ็นต์ในการจับตำแหน่งข้อมูลลายมือจึงต้องมีค่าสูงกว่าเปอร์เซ็นต์ในการจับตำแหน่งข้อมูล Contour Tracking graph พอสมควร ซึ่งจำนวน Synapses ต่อ Neuron ของภาพตัวอักษรที่มีค่าเท่ากับ 8 ให้เปอร์เซ็นต์ที่เหมาะสม นั้นคือมีค่าเท่ากับ 50 % ซึ่งทำให้ค่าเปอร์เซ็นต์การรู้จำของระบบมีค่าสูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 Contour Tracking graph จากการเขียนตัวอักษร ท

<u>การทดลองที่ 2</u> ทำการเปรียบเทียบจำนวน Neuron ที่ใช้ในระบบ

กำหนดโครงสร้างของ Neuron เหมือนกับการทดลองที่ 1 เพียงแต่ใช้จำนวน Neuron ต่างกันในระบบ

การทดลองจะกระทำการฝึกสอน Neuron โดยใช้ตัวอักษร ก - ฮ ต่างกัน 2 รูปแบบ แล้วหาเปอร์เซ็นต์การรู้จำจากการทดสอบจำนวน 20 ครั้งในแต่ละตัวอักษร โดยตัวอักษรที่ทดสอบนั้นจะประกอบด้วยหลายลายมือจากผู้ทดสอบหลายคนด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 เปอร์เซ็นต์การรู้จำตัวอักษร ที่มีตัวแปรเป็นจำนวนของ Neuron

จำนวน Neuron ของ ภาพตัวอักษร/ภาพ Contour Tracking graph	เปอร์เซ็นต์การรู้จำตัวอักษร (%)
25 / 50	32.78
50 / 50	39.65
100 / 50	62.55
25 / 100	72.30
50 / 100	78.50
100 / 100	85.50
25 / 200	82.89
50 / 200	83.98
100 / 200	92.20

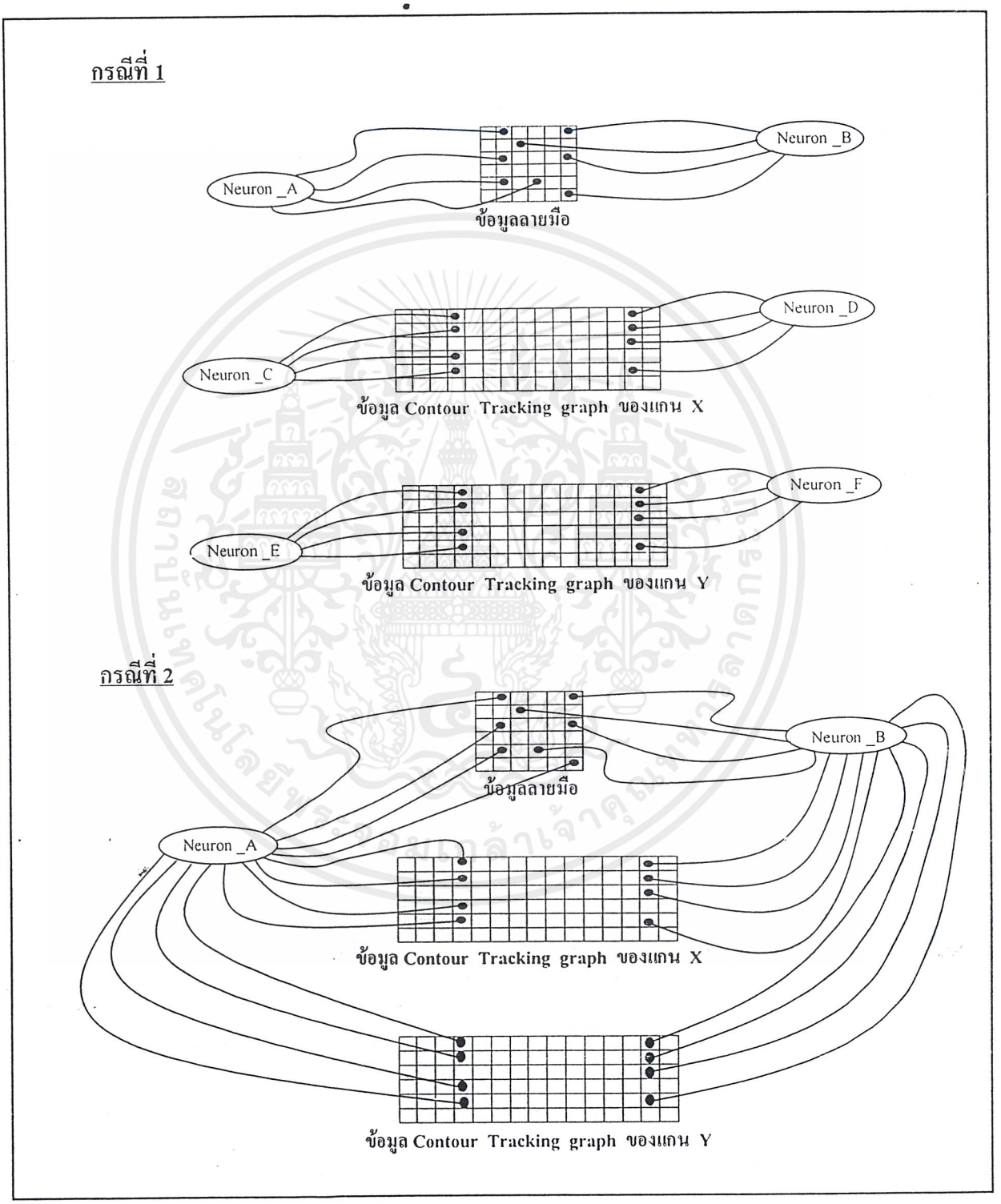
จากตาราง 4.2 ผลที่ได้สามารถอธิบายด้วยเหตุผลเดียวกับการทดลองที่ 1 นั่นคือ เปอร์เซ็นต์การจับตำแหน่งข้อมูลต่างๆ และอีกเหตุผลหนึ่งคือภาพของ Contour Tracking graph แสดงให้เห็นภาพความแตกต่างของแต่ละข้อมูลตัวอักษรมากกว่าข้อมูลลายมือเขียน จึงเป็นผลดีกับระบบ ถ้ากำหนดให้จำนวน Neuron หรือ คะแนนที่ให้กับข้อมูลภาพ Contour Tracking graph มีจำนวนมากกว่าข้อมูลลายมือเขียน

<u>การทดลองที่ 3</u> ทำการเปรียบเทียบการใช้ Neuron ร่วมกันของข้อมูลลายมือเขียนกับข้อมูลภาพของ Contour Tracking graph

กรณีที่1 กำหนดโครงสร้างของ Neuron เหมือนกับการทดลองที่ 1 และ 2 ที่ผ่านมาเพื่อจะใช้เป็นข้อมูลเปรียบเทียบ ส่วนกรณีที่ 2 กำหนดให้ Neuron ที่ใช้ในการจดจำภาพของตัวอักษรและภาพ Contour Tracking graph ของแกน X ,Y เป็นตัวเดียวกันโดยใช้ทั้งหมด 100 ตัว ตัวหนึ่งมี Synapses ทั้งหมด 24 ขา โดยแบ่งขาทั้งหมดเป็น 3 ส่วน ส่วนละ 8 ขา โดยแต่ละส่วนจะจับภาพข้อมูลแต่ละประเภท และกำหนดให้ตำแหน่ง Synapses ของ Neuron ที่จับภาพ Contour Tracking graph จับในตำแหน่งข้อมูล ณ. เวลา (t) หนึ่งเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองจะกระทำการฝึกสอน Neuron โดยใช้ตัวอักษร ก - ฮ ต่างกัน 2 รูปแบบ แล้วหาเปอร์เซ็นต์การรู้จำจากการทดสอบจำนวน 20 ครั้งในแต่ละตัวอักษร โดยตัวอักษรที่ทดสอบนั้นจะประกอบด้วยหลายลายมือจากผู้ทดสอบหลายคนด้วยกัน



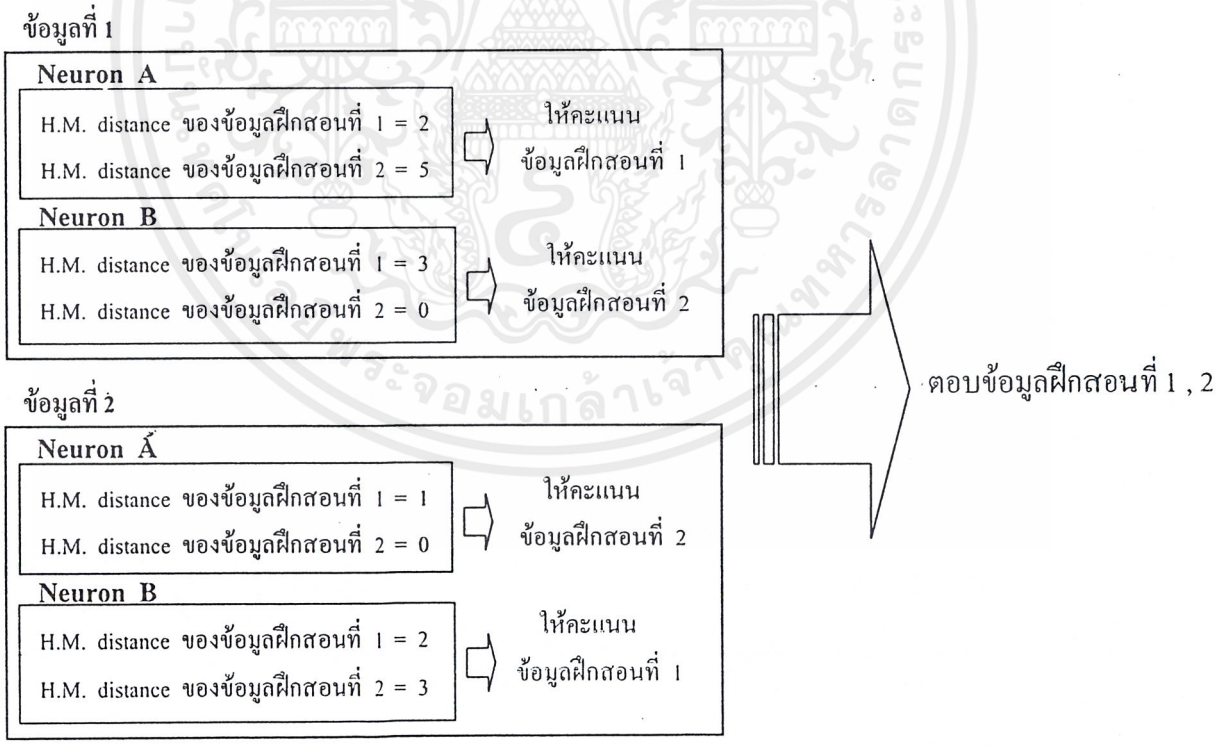
รูปที่ 4.3 ภาพแสดงการจับตำแหน่ง Synapses ในการทดลองที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 เปอร์เซ็นต์การรู้จำตัวอักษร ที่มี Neuron ใช้ข้อมูลร่วมกันของข้อมูลลายมือเขียนกับข้อมูลภาพของ Contour Tracking graph

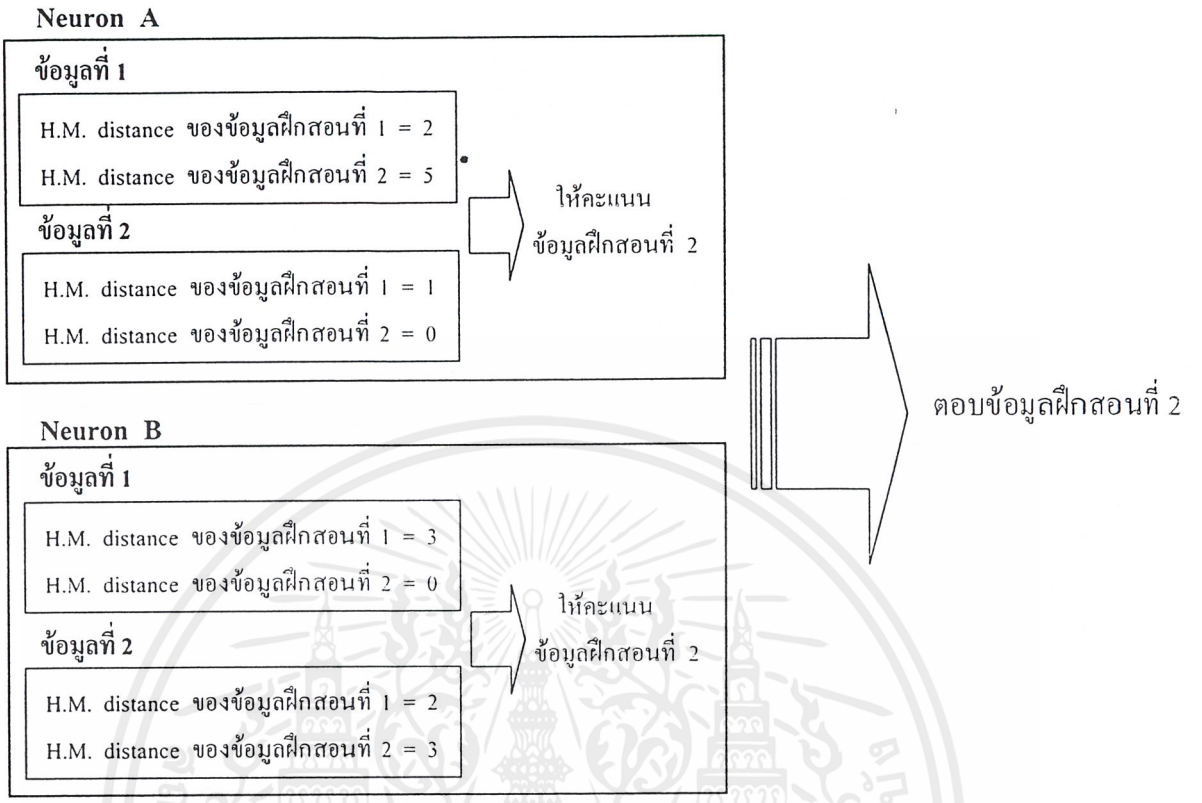
ประเภทของ Neuron	เปอร์เซ็นต์การรู้จำตัวอักษร (%)
ใช้ Neuron ร่วมกัน	94.12
ไม่ใช้ Neuron ร่วมกัน	92.20

จากตาราง 4.3 จะเห็นว่าการใช้ Neuron ตัวเดียวกันในการจับข้อมูลลายมือเขียนและข้อมูลภาพ Contour Tracking graph จะทำให้ค่าเปอร์เซ็นต์การรู้จำตัวอักษรสูงกว่าการใช้ Neuron คนละตัว ก่อนอื่นเรามาย้อนไปในบทที่ 3 ที่กล่าวเกี่ยวกับเรื่อง กระบวนการประมวลผลของ Neuron เราใช้ค่า Hamming Distance ในการประมวลผลของ Neuron แต่ละตัว ซึ่งสามารถสรุปได้ว่า Classes ในกรณีของ Neuron ที่มีการใช้ข้อมูลร่วมกันซึ่งมี 24 Input จะมีระยะห่างของแต่ละ Classes มากกว่าระยะห่างของข้อมูลที่มีการการใช้ Neuron แยกในข้อมูลแต่ละชนิด ได้แก่ Neuron ซึ่งมีแค่ 8 Input

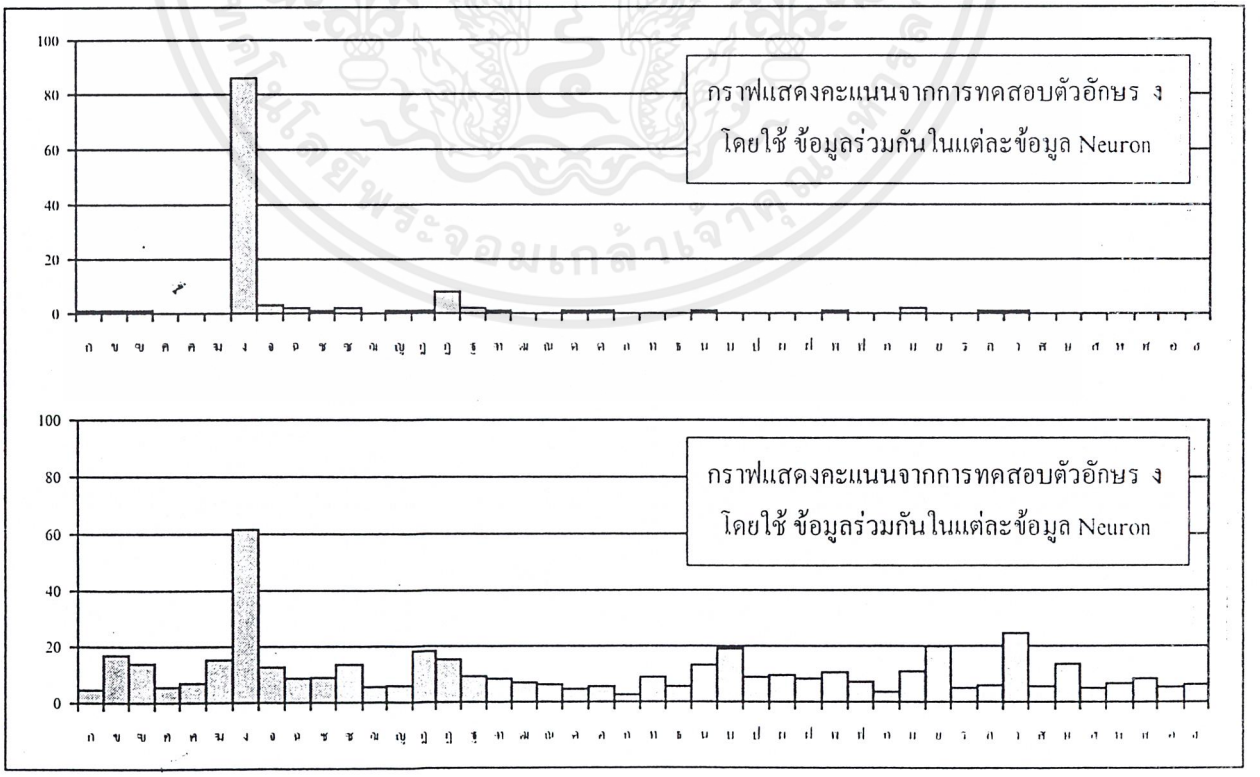


รูปที่ 4.4 ภาพแสดงการหาคำตอบของการประมวลผลแบบใช้ Neuron แยกกันในแต่ละข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 ภาพแสดงการหาคำตอบของการประมวลผลแบบใช้ข้อมูลร่วมกันในแต่ละ Neuron



รูปที่ 4.6 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ง

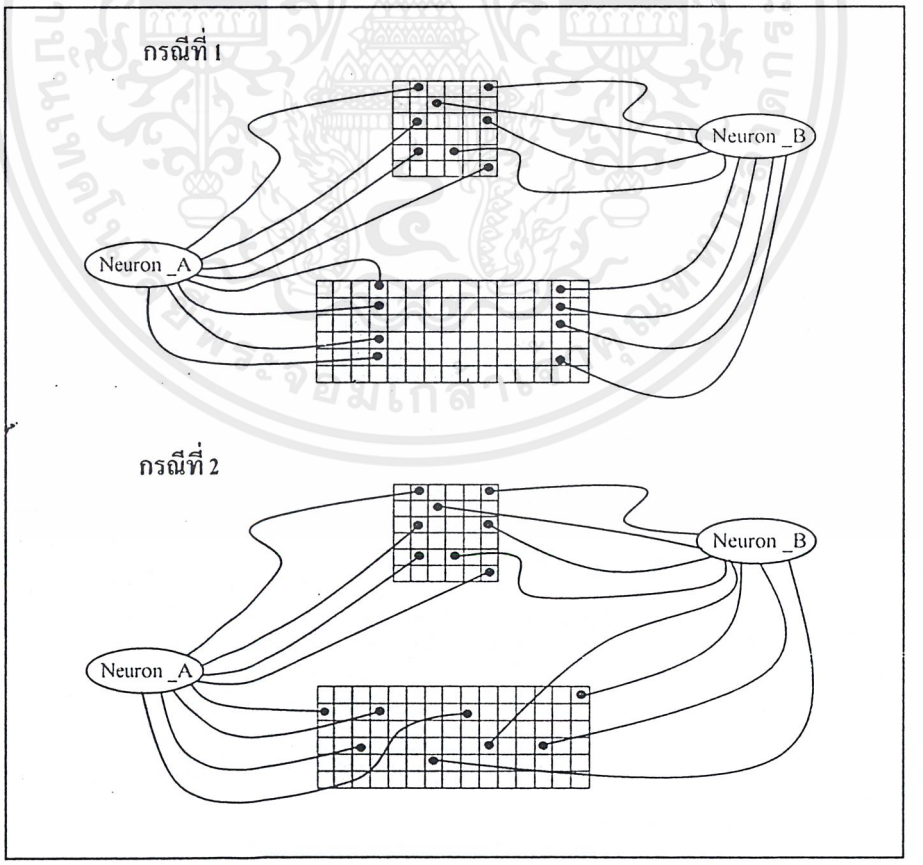
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างที่แสดงในรูป 4.4 และ 4.5 จะเห็นว่าผลการประมวลผลจากการใช้ Neuron แยกตัวกันในแต่ละตัวอักษรจะทำให้คำตอบที่ได้จะไม่เห็นเด่นชัดเท่าที่ควร ซึ่งสามารถแสดงให้เห็นชัดยิ่งขึ้นจากตัวอย่างรูปกราฟในรูปที่ 4.6

<u>การทดลองที่ 4</u> **ทำการเปรียบเทียบตำแหน่ง Synapses ของ Neuron ที่จับภาพของ Contour Tracking graph**

กำหนดให้ Neuron ที่ใช้ในการจดจำภาพของตัวอักษรและภาพของ Contour Tracking graph ของแกน X ,Y เป็นตัวเดียวกันโดยใช้ทั้งหมด 100 ตัว ตัวหนึ่งมี Synapses ทั้งหมด 24 ขา โดยแบ่งขาทั้งหมดเป็น 3 ส่วน ส่วนละ 8 ขา โดยแต่ละส่วนจะจับภาพข้อมูลแต่ละประเภท โดยกรณีที่ 1 ให้ Synapses ของ Neuron ตัวหนึ่งที่จับข้อมูล Contour Tracking graph ที่ตำแหน่งข้อมูล ณ. เวลา (t) หนึ่งๆ และกรณีที่ 2 ให้จับตำแหน่งข้อมูลไม่กำหนดช่วงเวลา (t)

การทดลองจะกระทำการฝึกสอน Neuron โดยใช้ตัวอักษร ก - ฮ ต่างกัน 2 รูปแบบ แล้วหาเปอร์เซ็นต์การรู้จำจากการทดสอบจำนวน 20 ครั้งในแต่ละตัวอักษร โดยตัวอักษรที่ทดสอบนั้นจะประกอบด้วยหลายลายมือจากผู้ทดสอบหลายคนด้วยกัน



รูปที่ 4.7 ภาพแสดงการจับตำแหน่งของ Synapses ในการทดลองที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

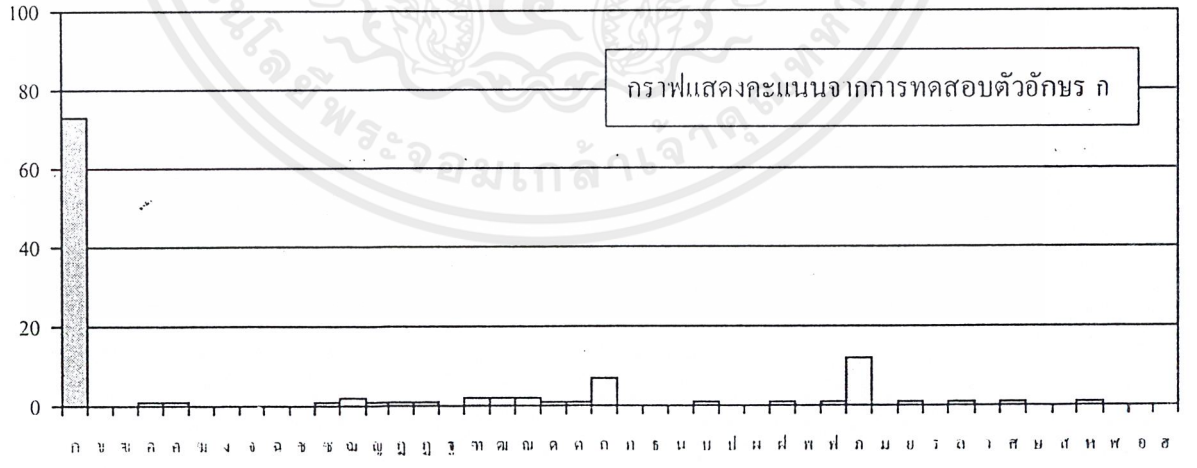
ตารางที่ 4.4 เปอร์เซ็นต์การรู้จำตัวอักษร ที่มีตัวแปรเป็นตำแหน่ง Synapses ของ Neuron บน Contour Tracking graph

ประเภทการจับของ Synapses ของ Neuron	เปอร์เซ็นต์การรู้จำตัวอักษร (%)
จับเฉพาะช่วงเวลาหนึ่ง	94.12
จับแบบสุ่มทุกช่วงเวลา	94.98

จากตารางที่ 4.4 จะเห็นว่าเปอร์เซ็นต์การรู้จำตัวอักษรของการทดลองทั้ง 2 แบบมีค่าใกล้เคียงกันมากซึ่งอาจเป็นเพราะว่าผลมาจากการจับตำแหน่งของ Neuron เป็นแบบ Random ซึ่งไม่มีความแน่นอน ทำให้ไม่ค่อยมีผลใดๆต่อเปอร์เซ็นต์การรู้จำของระบบ

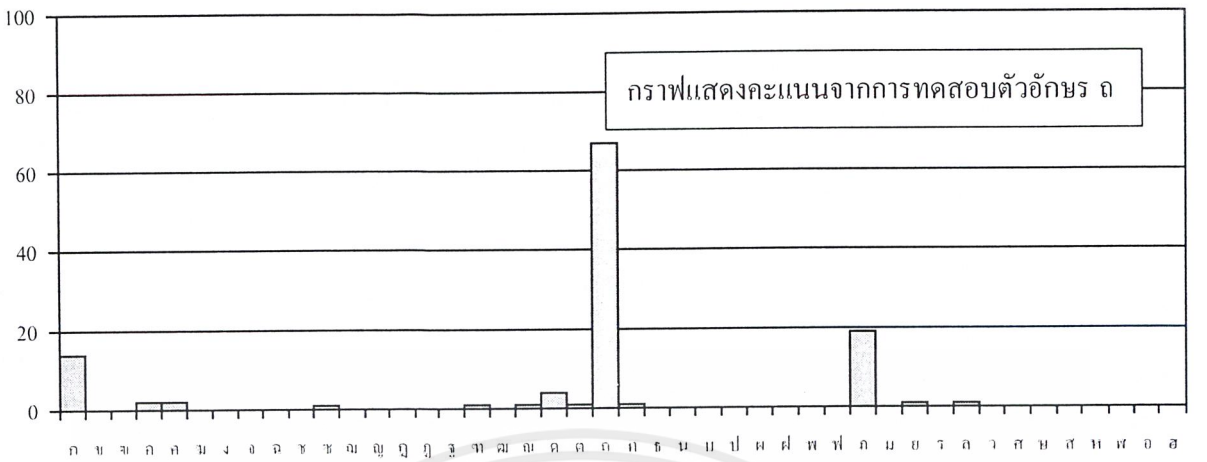
จากการทดลองทั้งหมดทำให้ได้วิธีการที่ดีที่สุดในการนำมาใช้ในการรู้จำตัวอักษรภาษาไทย นั่นคือ การกำหนดให้ Neuron จดจำทั้งภาพของตัวอักษรและภาพของ Contour Tracking graph ของแกน X ,Y โดยใช้ทั้งหมด 100 ตัว ตัวหนึ่งมี Synapses 24 ขา โดยแบ่งขาเป็น 3 ส่วน ส่วนละ 8 ขา แต่ละส่วนจะจับภาพข้อมูลแต่ละประเภท โดยให้ Synapses ที่จับข้อมูล Contour Tracking graph จับตำแหน่งข้อมูล ณ. เวลา (t) หนึ่งๆ หรือจับตำแหน่งข้อมูลไม่กำหนดช่วงเวลา (t)

ต่อไปนี้จะเป็นตัวอย่างจากการทดสอบตัวอักษรบางตัวโดยใช้วิธีข้างต้น

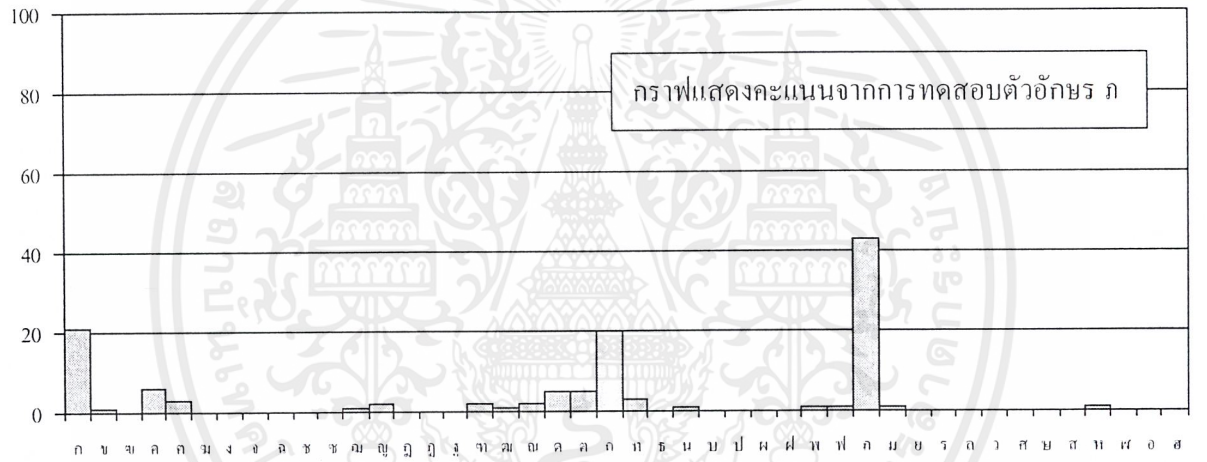


รูปที่ 4.8 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ก

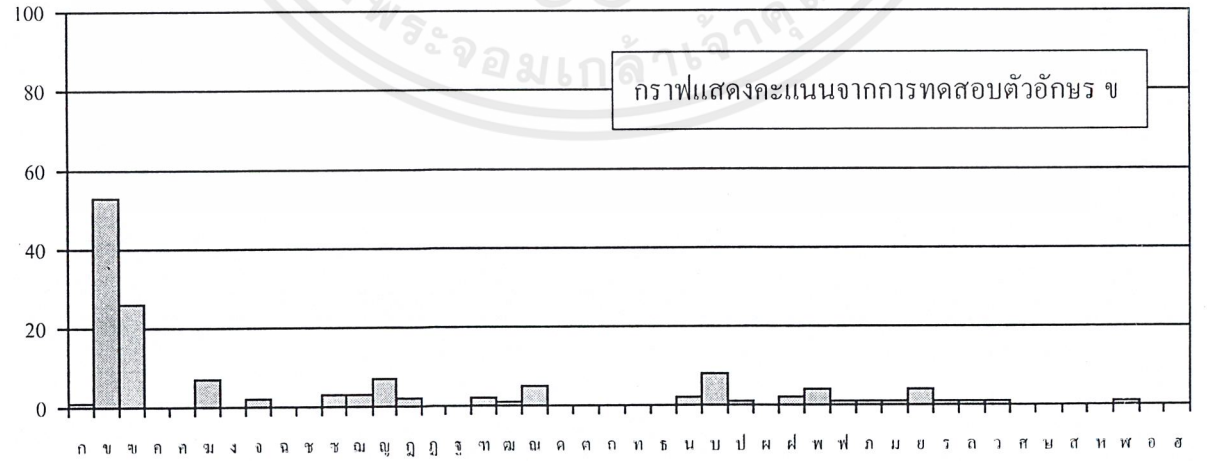
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ถ

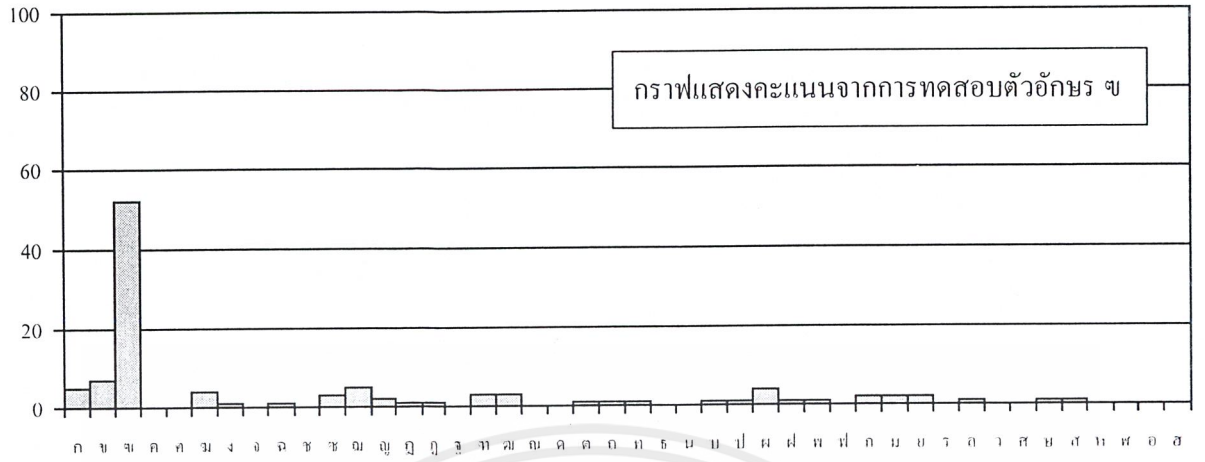


รูปที่ 4.10 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ภ

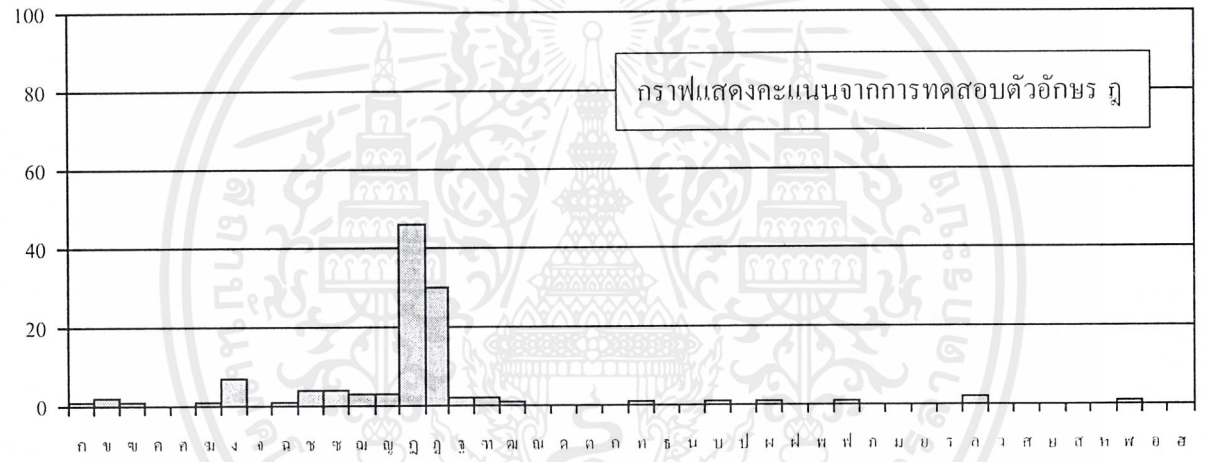


รูปที่ 4.11 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ข

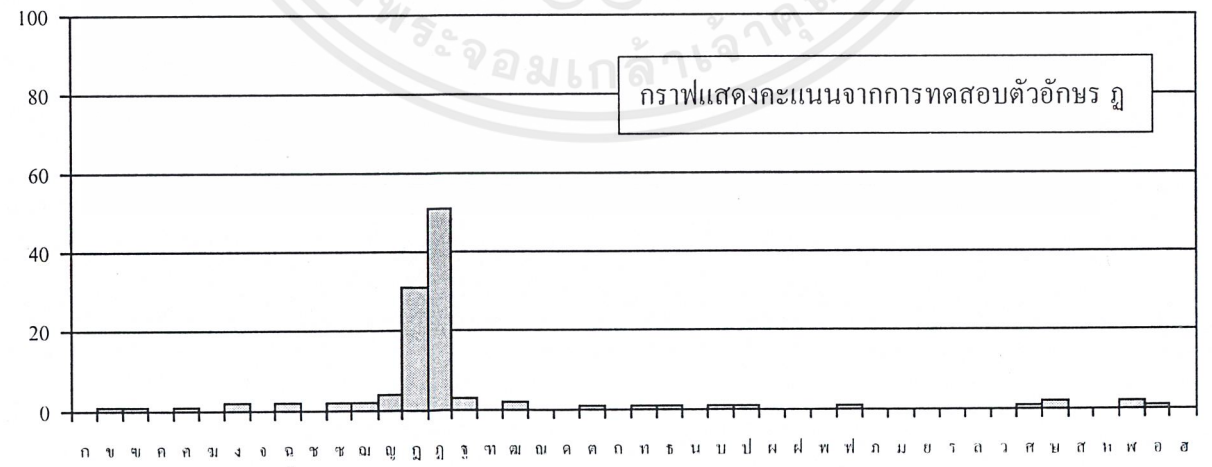
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ฃ

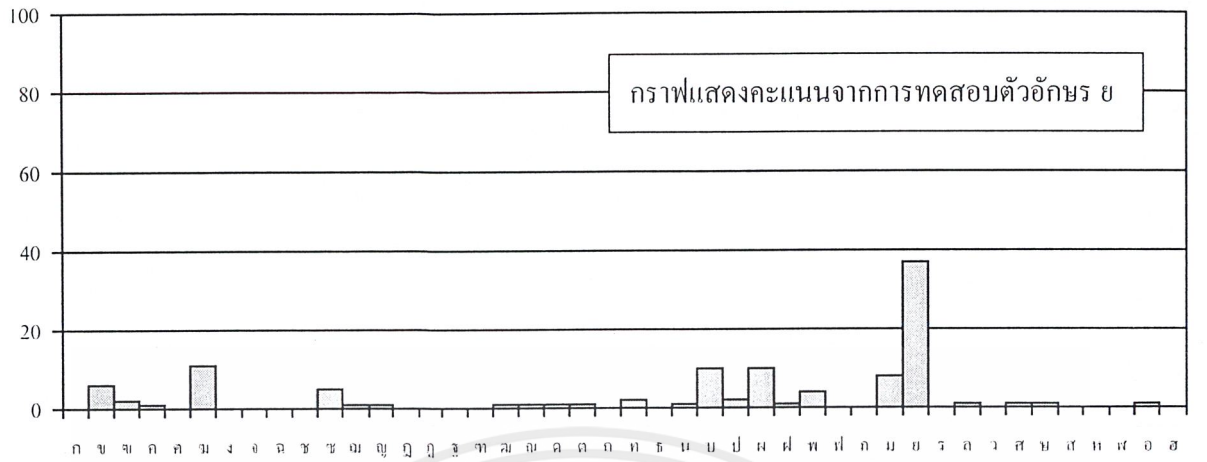


รูปที่ 4.13 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ฎ

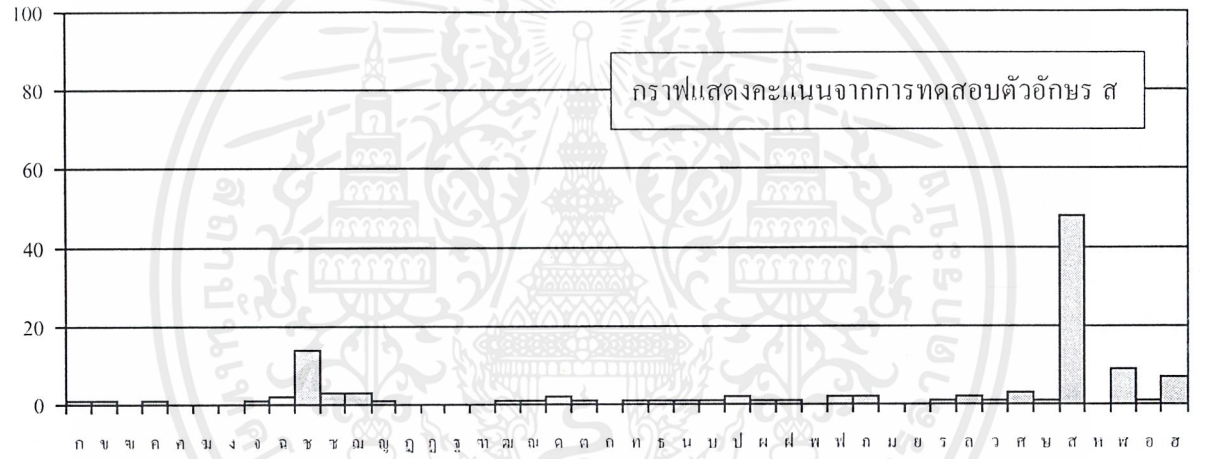


รูปที่ 4.14 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ฏ

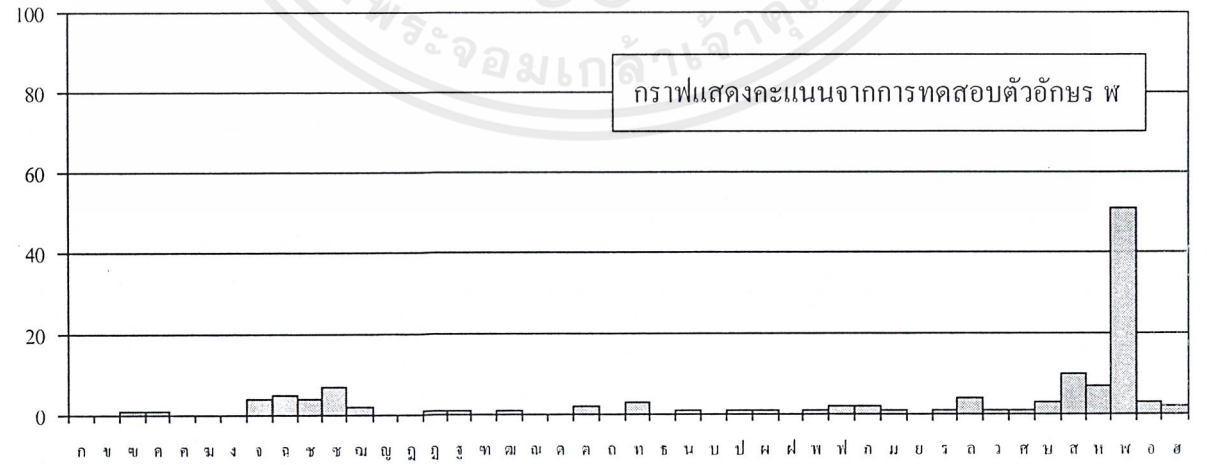
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ย



รูปที่ 4.16 กราฟแสดงคะแนนจากการทดสอบตัวอักษร ส



รูปที่ 4.17 กราฟแสดงคะแนนจากการทดสอบตัวอักษร พ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
วิจารณ์และข้อเสนอแนะ

จากที่ได้กล่าวไปแล้วว่า ในโครงงานนี้มีข้อมูลเป็นลายมือเขียน ความแตกต่างของลายมือจะขึ้นอยู่กับวิธีการเขียนของแต่ละบุคคล ทำให้เกิดความผิดพลาดในการรู้จำของ Neural ดังนั้นการกำจัดลักษณะเฉพาะของลายมืออาจเป็นวิธีหนึ่งที่สามารถทำให้การรู้จำของ Neural ดีขึ้น ซึ่งในโครงงานนี้จึงได้ใช้ข้อมูลฝึกสอนแบบลายมือเขียนของตัวอักษรนั้นๆ และข้อมูลกราฟโครงร่าง ซึ่งเป็นผลให้เปอร์เซ็นต์การรู้จำของระบบสูงเป็นที่น่าพอใจอย่างยิ่ง

ในโครงงานนี้ได้มีการทดสอบหลายรูปแบบ เพื่อจะหาวิธีที่ดีที่สุดที่จะทำให้ประสิทธิภาพการรู้จำของ Neural สูงที่สุด และจากการทดลองในบทที่ 4 สรุปได้ว่าการใช้ Neuron ในการจดจำทั้งภาพของตัวอักษรและภาพของ Contour Tracking graph ของแกน X ,Y จำนวน 100 ตัว ที่มี Synapses 24 ขา โดยแบ่งขาทั้งหมดเป็น 3 ส่วน ส่วนละ 8 ขา แต่ละส่วนจะจับภาพข้อมูลแต่ละประเภท แต่ก็ยังมีปัญหาที่เกิดจากความผิดพลาดที่เกิดขึ้นกับบางตัวอักษร ไม่ว่าจะเกิดจากสาเหตุของลายมือของแต่ละบุคคลที่แตกต่างกันหรือความคล้ายคลึงกันของตัวอักษรแต่ละตัว

เมื่อการรู้จำของ Neural มีประสิทธิภาพดีเพียงพอแล้ว การนำข้อมูลที่ได้ไปประยุกต์ก็มีหลายแบบหลายวิธี ตัวอย่างเช่น การนำข้อมูลจากข้อมูลแบบต่างๆ , การเชื่อมต่อผลที่ได้กับการทำงานของ Microsoft Office , Auto-Cad หรือโปรแกรมอื่นๆ การประยุกต์ใช้กับอุปกรณ์ประเภท Hand-Held เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

ตัวอักษรที่ใช้ในการทดสอบ

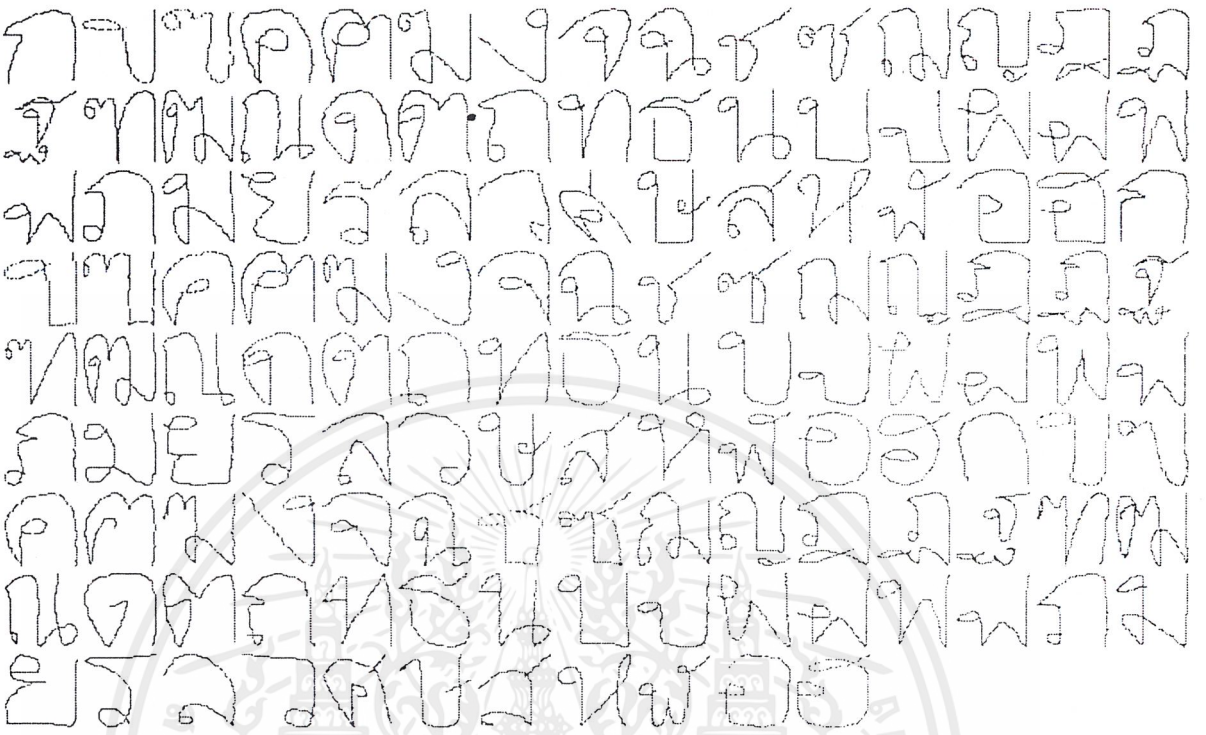
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 1ก ตัวอย่างตัวอักษรที่ใช้ในการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

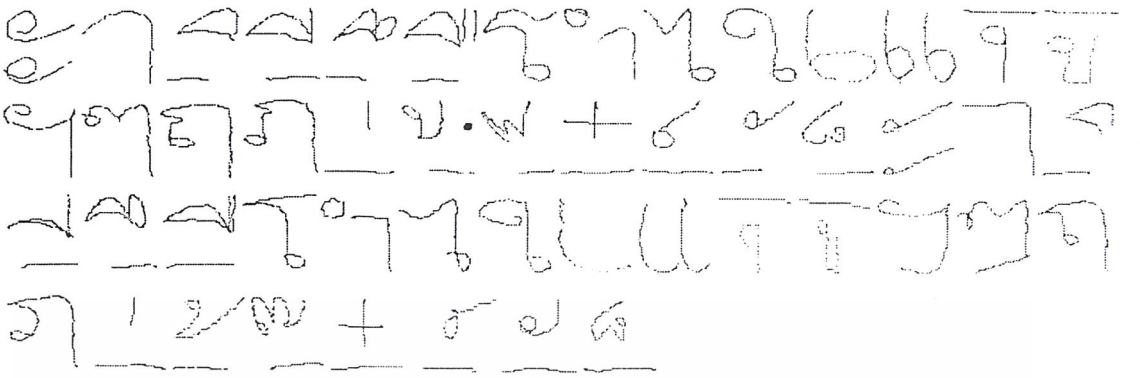
ภาคผนวก ข
ตัวอักษรที่ใช้ในการฝึกสอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1ข ตัวอักษรที่ใช้ในการ Training

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2ข สระที่ใช้ในการ Training

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

โปรแกรมที่ใช้หาตำแหน่งให้ Neuron จับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unit thai_Optimize_Random;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs,
  StdCtrls;

type
  TForm1 = class(TForm)
    Button1: TButton;
    Edit1: TEdit;
    Label11: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Edit2: TEdit;
    Edit3: TEdit;
    Label6: TLabel;
    Edit4: TEdit;
    Label7: TLabel;
    Edit5: TEdit;

    procedure aa(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
  array_dot=array[0..6000]of Smallint;
  array_dot_dot=array[0..1,0..2000]of word;
  function Find_xmin(count:word; pointx:array_dot):word;
  function Find_xmax(count:word; pointx:array_dot):word;
  function Find_ymin(count:word; pointy:array_dot):word;
  function Find_ymax(count:word; pointy:array_dot):word;
  function scaling(rate:real; count:word; point:array_dot;
        num_min,num_max:integer):array_dot;
  function scan_pixel(count:word; pointx:array_dot;
        pointy:array_dot):array_dot_dot;
const
  amount_of_pattern=88; //84
  random_point=40;
  leg_of_xy=8;
  leg_of_axis=8;
  total_random_xy=800;
  total_random_axis=800;   //use in random_mode=2
  optimiz_xy=4;
  ////////////////////////////////////////////
  //   optimizing mode of axis   //
  //   0. no optimizing          //
  //   1. optimize by point      //
  //   2. optimize by area       //
  ////////////////////////////////////////////
  optimizing_mode=2;
  opimize_axis=3;   // use in optimizing mode of axis=2
  ////////////////////////////////////////////
  //    random mode of axis     //
  //    1. random per pixel     //
  //    2. random from area     //
  ////////////////////////////////////////////
  random_mode=2;


var
  Form1: TForm1;
  neuralx:file;
  neuraly:file;
  patx:file;  pataxisx:file;
  paty:file;  pataxisy:file;
  amount:array[1..amount_of_pattern]of word;
  i,ii,j,jj,kk,cc:word;
  dotx,doty:array[1..amount_of_pattern,1..1500]of byte;
  dotaxisx,dotaxisy:array[1..amount_of_pattern,1..100]of
byte;
  score:array[0..random_point,0..random_point]of Shortint;
  scoreaxisx:array[1..100,1..random_point]of Shortint;
  scoreaxisy:array[1..100,1..random_point]of Shortint;
  temp_dotx,temp_doty:array_dot;
implementation
uses thai_ruetee2;
{$R *.DFM}

procedure TForm1.aa(Sender: TObject);
      var
      ii,j,c,Status:integer;
      temp:array_dot_dot;
      a,ax:array[1..total_random_xy]of Byte;
      b,by:array[1..total_random_xy]of Byte;

axisx,axisy,axis_xx,axis_yy:array[1..100,1..leg_of_axis]of
Shortint;

axisx_area,axisy_area,axis_area_xx,axis_area_yy:array[1..
total_random_axis*2]of shortint;
      his_x,his_y,his_xx,his_yy:array[1..800]of byte;
      xmin,ymin,xmax,ymax:word;

neuralx,neuraly,neuralaxisx,neuralaxisy,neuralhisx,neuralh
isy:file;
      begin
      form1.hide;
      form2.show;
      //recieve pattern from file//
      assignfile(patx,'thai_training_x.dat');
      reset(patx,4);
      assignfile(paty,'thai_training_y.dat');
      reset(paty,4);
      for ii:=1 to amount_of_pattern do
            begin
            seek(patx,(ii-1)*1000);
            blockread(patx,amount[ii],1);
            for i:=1 to amount[ii] do
                  begin
                  seek(patx,(ii-1)*1000+i);
                  blockread(patx,dotx[ii,i],1);
                  seek(paty,(ii-1)*1000+i);
                  blockread(paty,doty[ii,i],1);
                  end;
            end;
      closefile(patx);
      closefile(paty);
      //test recieve//
      c:=0;
      for ii:=1 to amount_of_pattern do
            begin
            if ((ii-1) mod 13 = 0) then c:=0;
            for i:=1 to amount[ii] do

form2.Canvas.pixels[dotx[ii,i]+(c*50),doty[ii,i]+((ii-1) div
13)*50]:=clGreen;
            c:=c+1;
            end;

      //recieve pattern from file//
      assignfile(pataxisx,'thai_training_axisx.dat');
      reset(pataxisx,4);
      assignfile(pataxisy,'thai_training_axisy.dat');
      reset(pataxisy,4);
      for ii:=1 to amount_of_pattern do
      for i:=1 to 100 do
                  begin
                  seek(pataxisx,(ii-1)*100+i);
                  blockread(pataxisx,dotaxisx[ii,i],1);
                  dotaxisx[ii,i]:=round(dotaxisx[ii,i]*40/45);
                  seek(pataxisy,(ii-1)*100+i);
                  blockread(pataxisy,dotaxisy[ii,i],1);
                  dotaxisy[ii,i]:=round(dotaxisy[ii,i]*40/45);
```

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            end;
        closefile(pataxisx);
        closefile(pataxisy);

for kk:=1 to  amount_of_pattern do
        begin
        cc:=amount[kk];
        for i:=1 to cc do
            begin
            temp_dotx[i-1]:=dotx[kk,i];
            temp_doty[i-1]:=doty[kk,i];
            end;
        //extened line//
        for i:=0 to cc-1 do
            begin
            temp_dotx[1*cc+i]:=temp_dotx[i]-1;
            temp_doty[1*cc+i]:=temp_doty[i]-1;
            temp_dotx[2*cc+i]:=temp_dotx[i]-0;
            temp_doty[2*cc+i]:=temp_doty[i]-1;
            temp_dotx[3*cc+i]:=temp_dotx[i]+1;
            temp_doty[3*cc+i]:=temp_doty[i]-1;

            temp_dotx[4*cc+i]:=temp_dotx[i]-1;
            temp_doty[4*cc+i]:=temp_doty[i]-0;
            temp_dotx[5*cc+i]:=temp_dotx[i]+1;
            temp_doty[5*cc+i]:=temp_doty[i]-0;

            temp_dotx[6*cc+i]:=temp_dotx[i]-1;
            temp_doty[6*cc+i]:=temp_doty[i]+1;
            temp_dotx[7*cc+i]:=temp_dotx[i]-0;
            temp_doty[7*cc+i]:=temp_doty[i]+1;
            temp_dotx[8*cc+i]:=temp_dotx[i]+1;
            temp_doty[8*cc+i]:=temp_doty[i]+1;

            temp_dotx[9*cc+i]:=temp_dotx[i]-2;
            temp_doty[9*cc+i]:=temp_doty[i]-2;
            temp_dotx[10*cc+i]:=temp_dotx[i]-1;
            temp_doty[10*cc+i]:=temp_doty[i]-2;
            temp_dotx[11*cc+i]:=temp_dotx[i]-0;
            temp_doty[11*cc+i]:=temp_doty[i]-2;
            temp_dotx[12*cc+i]:=temp_dotx[i]+1;
            temp_doty[12*cc+i]:=temp_doty[i]-2;
            temp_dotx[13*cc+i]:=temp_dotx[i]+2;
            temp_doty[13*cc+i]:=temp_doty[i]-2;

            temp_dotx[14*cc+i]:=temp_dotx[i]-2;
            temp_doty[14*cc+i]:=temp_doty[i]+2;
            temp_dotx[15*cc+i]:=temp_dotx[i]-1;
            temp_doty[15*cc+i]:=temp_doty[i]+2;
            temp_dotx[16*cc+i]:=temp_dotx[i]-0;
            temp_doty[16*cc+i]:=temp_doty[i]+2;
            temp_dotx[17*cc+i]:=temp_dotx[i]+1;
            temp_doty[17*cc+i]:=temp_doty[i]+2;
            temp_dotx[18*cc+i]:=temp_dotx[i]+2;
            temp_doty[18*cc+i]:=temp_doty[i]+2;
            end;

        //scan pixels//
        temp:=scan_pixel(19*cc-1,temp_dotx,temp_doty);
        cc:=temp[0,0];
        for i:=1 to cc-1  do
            begin
            temp_dotx[i]:=temp[0,i];
            temp_doty[i]:=temp[1,i];
            end;

        //find xmax,xmin//
        xmin:=Find_xmin(cc,temp_dotx);
        xmax:=Find_xmax(cc,temp_dotx);
        ymin:=Find_ymin(cc,temp_doty);
        ymax:=Find_ymax(cc,temp_doty);
        //scaling//
        temp_dotx:=scaling(random_point,cc,temp_dotx,xmin,xmax);
        temp_doty:=scaling(random_point,cc,temp_doty,ymin,ymax);
            //scan pixels//
            temp:=scan_pixel(cc,temp_dotx,temp_doty);
            amount[kk]:=temp[0,0];
            for i:=1 to amount[kk]-1  do
                begin
                dotx[kk,i]:=temp[0,i]-xmin;
                doty[kk,i]:=temp[1,i]-ymin;
                end;
            end;

            //test recieve//
            Form2.canvas.brush:=brush;
            Form2.Canvas.FillRect(Form2.ClientRect);
            c:=0;
            for ii:=1 to amount_of_pattern do
                begin
                if ((ii-1) mod 13 = 0) then c:=0;
                for i:=1 to amount[ii] do

form2.Canvas.pixels[dotx[ii,i]+(c*50),doty[ii,i]+((ii-1) div 13)*50]:=clolive;
                c:=c+1;
                end;

            for jj:=0 to random_point do
            for  j:=0 to random_point do
                score[jj,j]:=0;
            for ii:=1 to amount_of_pattern do
                begin
                for jj:=0 to random_point do
                for  j:=0 to random_point do
                    score[jj,j]:=score[jj,j]-1;
                for i:=1 to amount[ii] do

score[dotx[ii,i],doty[ii,i]]:=score[dotx[ii,i],doty[ii,i]]+2;
                end;

            form2.hide;
            form1.show;
            form1.canvas.Brush.color:=clwhite;
            form1.canvas.pen.color:=clwhite;

form1.Canvas.Rectangle(0,0,random_point+1,random_point+1);

            i:=1;
            for jj:=0 to random_point do
            for  j:=0 to random_point do
                if (score[jj,j]>=-
optimiz_xy)and(score[jj,j]<=optimiz_xy)then
                    begin
                    form1.Canvas.Pixels[jj,j]:=clblack;
                    a[i]:=jj;
                    b[i]:=j;
                    i:=i+1;
                    end;
            //random for xy//
            j:=i;
            for i:=j to total_random_xy do
                begin
                status:=0;
                while status=0 do
                    begin
                    status:=1;
                    a[i]:=random(random_point);
                    b[i]:=random(random_point);
                    for ii:=1 to i-1 do
                    if (a[i]=a[ii]) and (b[i]=b[ii]) then
                    status:=0;
                    end;
                end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      ////////////////////////////optimize ////////////////////////////////

form1.Canvas.Rectangle(50,0,150+1,random_point+1);

form1.Canvas.Rectangle(160,0,260+1,random_point+1);
      case optimizing_mode of
      ////////////////////////optimize by
area//////////////////////////////
      2:
      begin


      //axisx//
      for jj:=1 to 100 do
      for  j:=1 to random_point do
            scoreaxisx[jj,j]:=0;

      for ii:=1 to amount_of_pattern do
            begin
            for i:=1 to 100 do
            for j:=1 to dotaxisx[ii,i] do
                  scoreaxisx[i,j]:=scoreaxisx[i,j]-1;
            for i:=1 to 100 do
            for j:=dotaxisx[ii,i]+1 to 40 do
                  scoreaxisx[i,j]:=scoreaxisx[i,j]+1;
            end;
      i:=0;
      for jj:=1 to 100 do
      for  j:=1 to random_point do
            if (scoreaxisx[jj,j]>=-
opimize_axis)and(scoreaxisx[jj,j]<=opimize_axis)then
                  begin
                  form1.Canvas.Pixels[jj+50,j]:=clblack;
                  i:=i+1;
                  scoreaxisx[jj,j]:=j;
                  end
            else
                  scoreaxisx[jj,j]:=-1;
      //axisy//
      for jj:=1 to 100 do
      for  j:=1 to random_point do
            scoreaxisy[jj,j]:=0;

      for ii:=1 to amount_of_pattern do
            begin
            for i:=1 to 100 do
            for j:=1 to dotaxisy[ii,i] do
                  scoreaxisy[i,j]:=scoreaxisy[i,j]-1;
            for i:=1 to 100 do
            for j:=dotaxisy[ii,i]+1 to 40 do
                  scoreaxisy[i,j]:=scoreaxisy[i,j]+1;
            end;
      i:=0;
      for jj:=1 to 100 do
      for  j:=1 to random_point do
            if (scoreaxisy[jj,j]>=-
opimize_axis)and(scoreaxisy[jj,j]<=opimize_axis)then
                  begin
                  form1.Canvas.Pixels[jj+160,j]:=clblack;
                  i:=i+1;
                  scoreaxisy[jj,j]:=j;
                  end
            else
                  scoreaxisy[jj,j]:=-1;



      end;

      ////////////////////////////optimize by
point//////////////////////////
      1:
      begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      //axis x//
      for jj:=1 to 100 do
      for  j:=1 to random_point do
            scoreaxisx[jj,j]:=0;
      for ii:=1 to amount_of_pattern do
            begin
            for jj:=1 to 100 do
            for  j:=1 to random_point do
                  scoreaxisx[jj,j]:=scoreaxisx[jj,j]-1;
            for i:=1 to 100 do

scoreaxisx[i,dotaxisx[ii,i]]:=scoreaxisx[i,dotaxisx[ii,i]]+2;
            end;

      for jj:=1 to 100 do
      for  j:=1 to random_point do
            if (scoreaxisx[jj,j]>=-60) then
                  begin
                  form1.Canvas.Pixels[jj+50,j]:=clblack;
                  scoreaxisx[jj,j]:=j;
                  end
            else
                  scoreaxisx[jj,j]:=-1;

      //axis y//
      for jj:=1 to 100 do
      for  j:=1 to random_point do
            scoreaxisy[jj,j]:=0;
      for ii:=1 to amount_of_pattern do
            begin
            for jj:=1 to 100 do
            for  j:=1 to random_point do
                  scoreaxisy[jj,j]:=scoreaxisy[jj,j]-1;
            for i:=1 to 100 do

scoreaxisy[i,dotaxisy[ii,i]]:=scoreaxisy[i,dotaxisy[ii,i]]+2;
            end;
      i:=1;
      for jj:=1 to 100 do
      for  j:=1 to random_point do
            if (scoreaxisy[jj,j]>=-60) then
                  begin
                  form1.Canvas.Pixels[jj+160,j]:=clblack;
                  scoreaxisy[jj,j]:=j;
                  end
            else
                  scoreaxisy[jj,j]:=-1;



      end;


      0:
      begin

      for ii:=1 to 100 do
      for  i:=1 to random_point do
            begin
            scoreaxisx[ii,i]:=-1;
            scoreaxisy[ii,i]:=-1;
            end;

      end;
      end;

      ////////////////////////////random ////////////////////////////////
      case random_mode of
      ////////////////////////random per
pixel//////////////////////////
      1:
      begin
```

```
    //random for axisx//
    for ii:=1 to 100 do
    for  i:=1 to 8 do
         axisx[ii,i]:=-1;
    for i:=1 to 100 do
         begin
         jj:=1;
         for j:=1 to random_point do
              begin
              if (scoreaxisx[i,j]<>-1) and
(jj<=leg_of_axis) then
                   begin
                   axisx[i,jj]:=scoreaxisx[i,j];
                   jj:=jj+1;
                   end;
              end;
         for ii:=1 to leg_of_axis do
              begin
              status:=0;

              while status=0 do
                   begin
                   status:=1;
                   if (axisx[i,ii]=-1)then
                   axisx[i,ii]:=random(random_point);
                   for j:=1 to ii-1 do
                        begin
                        if axisx[i,ii]=axisx[i,j]  then
                             begin
                             status:=0;
                             axisx[i,ii]:=-1;
                             end;
                        end;
                   end;

              end;
         end;
    //random for axisy//
    for ii:=1 to 100 do
    for  i:=1 to 8 do
         axisy[ii,i]:=-1;
    for i:=1 to 100 do
         begin
         jj:=1;
         for j:=1 to random_point do
              begin
              if (scoreaxisy[i,j]<>-1) and
(jj<=leg_of_axis) then
                   begin
                   axisy[i,jj]:=scoreaxisy[i,j];
                   jj:=jj+1;
                   end;
              end;
         for ii:=1 to leg_of_axis do
              begin
              status:=0;

              while status=0 do
                   begin
                   status:=1;
                   if (axisy[i,ii]=-1)then
                   axisy[i,ii]:=random(random_point);
                   for j:=1 to ii-1 do
                        begin
                        if axisy[i,ii]=axisy[i,j]  then
                             begin
                             status:=0;
                             axisy[i,ii]:=-1;
                             end;
                        end;
                   end;
              end;
         end;

    end;

    ////////////////////////////random from
area//////////////////////////
    2:
    begin

    //axisx_area//
    for i:=1 to (total_random_axis*2) do
    axisx_area[i]:=-1;
    jj:=1;
    for i:=1 to 100 do
         begin
         for j:=1 to random_point do
              begin
              if (scoreaxisx[i,j]<>-1) then
                   begin
                   axisx_area[jj]:=scoreaxisx[i,j];
                   axisx_area[jj+total_random_axis]:=i;
                   jj:=jj+1;
                   end;
              end;
         end;

    for i:=jj to total_random_axis do
         begin
         status:=0;
         while status=0 do
              begin
              status:=1;
              if (axisx_area[i]=-1)then
                   begin
                   axisx_area[i]:=random(random_point);

axisx_area[i+total_random_axis]:=random(100);
                   end;
              for j:=1 to i-1 do
                   begin
                   if (axisx_area[i]=axisx_area[j]) and

(axisx_area[i+total_random_axis]=axisx_area[j+total_rand
om_axis]) then
                        begin
                        status:=0;
                        axisx_area[i]:=-1;
                        end;
                   end;
              end;
         end;

    //axisy_area//
    for i:=1 to (total_random_axis*2) do
    axisy_area[i]:=-1;
    jj:=1;
    for i:=1 to 100 do
         begin
         for j:=1 to random_point do
              begin
              if (scoreaxisy[i,j]<>-1) then
                   begin
                   axisy_area[jj]:=scoreaxisy[i,j];
                   axisy_area[jj+total_random_axis]:=i;
                   jj:=jj+1;
                   end;
              end;
         end;

    for i:=jj to total_random_axis do
         begin
         status:=0;
         while status=0 do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            begin
            status:=1;
            if (axisy_area[i]=-1)then
                begin
                axisy_area[i]:=random(random_point);

axisy_area[i+total_random_axis]:=random(100);
                end;
            for j:=1 to i-1 do
                begin
                if (axisy_area[i]=axisy_area[j]) and

(axisy_area[i+total_random_axis]=axisy_area[j+total_rand
om_axis]) then
                    begin
                    status:=0;
                    axisy_area[i]:=-1;
                    end;
                end;
            end;
        end;




    end;
    end;



    ////////////////////////////////test//////////////////////////////////
    form1.canvas.Brush.color:=clwhite;
    form1.canvas.pen.color:=clwhite;

form1.Canvas.rectangle(80,50,80+random_point,50+rando
m_point);
    form1.Canvas.rectangle(160,50,260,90);
    form1.Canvas.rectangle(300,50,400,90);
    //test xy//
    for i:=1 to total_random_xy do
    form1.Canvas.pixels[a[i]+80,b[i]+50]:=clblack;
    c:=0;
    for i:=80 to 80+random_point do
    for j:=50 to 50+random_point do
    if form1.Canvas.pixels[i,j]=clblack then
    inc(c);
    form1.Edit1.Text:=inttostr(c);

    //test axis//
    case random_mode of
    1:
    begin


    for ii:=1 to 100 do
    for  i:=1 to leg_of_axis do
        begin
        form1.Canvas.pixels[(ii-
1)+160,axisx[ii,i]+50]:=clblack;
        form1.Canvas.pixels[(ii-
1)+300,axisy[ii,i]+50]:=clblack;
        end;


    end;

    2:
    begin

    for i:=1 to total_random_axis do
        begin

form1.Canvas.pixels[axisx_area[i+total_random_axis]+16
0,axisx_area[i]+50]:=clblack;

form1.Canvas.pixels[axisy_area[i+total_random_axis]+30
0,axisy_area[i]+50]:=clblack;
        end;

    end;
    end;


    c:=0;
    for ii:=160 to 260 do
    for  i:=50 to 50+random_point do
    if form1.Canvas.pixels[ii,i]=clblack then
    inc(c);
    form1.Edit2.Text:=inttostr(c);
    c:=0;
    for ii:=300 to 400 do
    for  i:=50 to 50+random_point do
    if form1.Canvas.pixels[ii,i]=clblack then
    inc(c);
    form1.Edit3.Text:=inttostr(c);


    //test after save xy//
    form1.canvas.Brush.color:=clwhite;
    form1.canvas.pen.color:=clwhite;

form1.Canvas.rectangle(80,100,80+random_point,100+ran
dom_point);
    form1.Canvas.rectangle(160,100,260,140);
    form1.Canvas.rectangle(300,100,400,140);

form1.Canvas.rectangle(450,100,450+random_point,100+r
andom_point);

form1.Canvas.rectangle(550,100,550+random_point,100+r
andom_point);
    ////save to file////
    //open file//
    assignfile(neuralx,'ran_x.dat');
    rewrite(neuralx,4);
    assignfile(neuraly,'ran_y.dat');
    rewrite(neuraly,4);
    assignfile(neuralaxisx,'ran_axisx.dat');
    rewrite(neuralaxisx,4);
    assignfile(neuralaxisy,'ran_axisy.dat');
    rewrite(neuralaxisy,4);

    //save xy//
    for i:=1 to total_random_xy do
        begin
        seek(neuralx,i);
        blockwrite(neuralx,a[i],1);
        seek(neuraly,i);
        blockwrite(neuraly,b[i],1);
        end;

    //test receive file//
    for i:=1 to total_random_xy do
        begin
        seek(neuralx,i);
        blockread(neuralx,ax[i],1);
        seek(neuraly,i);
        blockread(neuraly,by[i],1);

form1.Canvas.pixels[ax[i]+80,by[i]+100]:=clblack;
        end;

    //save axisxy//
    case random_mode of
    1:
    begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for ii:=1 to 100 do
    for  i:=1 to leg_of_axis do
         begin
         seek(neuralaxisx,(ii-1)*leg_of_axis+i);
         blockwrite(neuralaxisx.axisx[ii,i],1);
         seek(neuralaxisy,(ii-1)*leg_of_axis+i);
         blockwrite(neuralaxisy.axisy[ii,i],1);
         end;
    //test receive file//
    for ii:=1 to 100 do
    for  i:=1 to leg_of_axis do
         begin
         seek(neuralaxisx,(ii-1)*leg_of_axis+i);
         blockread(neuralaxisx.axis_xx[ii,i],1);
         form1.Canvas.pixels[(ii-
1)+160,axis_xx[ii,i]+100]:=clblack;
         seek(neuralaxisy,(ii-1)*leg_of_axis+i);
         blockread(neuralaxisy.axis_yy[ii,i],1);
         form1.Canvas.pixels[(ii-
1)+300,axis_yy[ii,i]+100]:=clblack;
         end;

    end;

    2:
    begin

    for  i:=1 to total_random_axis*2 do
         begin
         seek(neuralaxisx,i);
         blockwrite(neuralaxisx.axisx_area[i],1);
         seek(neuralaxisy,i);
         blockwrite(neuralaxisy.axisy_area[i],1);
         end;

    //test receive file//
    for  i:=1 to total_random_axis*2 do
         begin
         seek(neuralaxisx,i);
         blockread(neuralaxisx.axis_area_xx[i],1);
         seek(neuralaxisy,i);
         blockread(neuralaxisy.axis_area_yy[i],1);
         end;
    for  i:=1 to total_random_axis do
         begin

form1.Canvas.Pixels[axis_area_xx[i+total_random_axis]+
160,axis_area_xx[i]+100]:=clblack;

form1.Canvas.Pixels[axis_area_yy[i+total_random_axis]+
300,axis_area_yy[i]+100]:=clblack;
         end;

    end;
    end;

    closefile(neuralx);
    closefile(neuraly);
    closefile(neuralaxisx);
    closefile(neuralaxisy);
    end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

โปรแกรมที่ใช้ในการ Training

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unit thai_training;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls,math;

type
  TForm1 = class(TForm)
    Button1: TButton;
    Timer1: TTimer;
    Button2: TButton;
    Edit1: TF :it;
    procedure clear(Sender: TObject);
    procedure Line_from_mousemove(Sender: TObject;
Shift: TShiftState; X,Y: Integer);
    procedure setup(Sender: TObject);
    procedure timer(Sender: TObject);
    procedure count_clear(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
  zero_one=set of 0..1;
  array_dot=array[0..3000]of Smallint;
  array_dot_dot=array[0..1,0..2500]of word;
  function Find_xmin(count:word; pointx:array_dot):word;
  function Find_xmax(count:word; pointx:array_dot):word;
  function Find_ymin(count:word; pointy:array_dot):word;
  function Find_ymax(count:word; pointy:array_dot):word;
  function scaling(rate:real; count:word; point:array_dot;
        num_min,num_max:integer):array_dot;
  function scan_pixel(count:word; pointx:array_dot;
        pointy:array_dot):array_dot_dot;
var
  Form1:TForm1;
  c,cc,j,xxx,yyy:word;
  press:zero_one;
  positionx,positiony:array_dot;
  jj:array[0..10]of word;
  pos_xy:longword;
implementation

{$R *.DFM}

procedure TForm1.setup(Sender: TObject);
      begin
      timer1.Enabled:=false;
      press:=[0];
      c:=0;
      pos_xy:= 113000  ;
      end;

procedure TForm1.Line_from_mousemove(Sender:
TObject; Shift: TShiftState; X,Y: Integer);
      begin
      if not((ssleft in shift)or(ssright in shift)or(ssmiddle in
shift)or
      (ssdouble in shift) or(ssalt in shift) or(ssctrl in shift)
or
      (ssshift in shift))then
            begin
            if 1 in press then
                  begin
                  press:=[0];
                  timer1.Enabled:=true;
                  end;
            end;
      if ssleft in shift then
            begin
            if 0 in press then
                  begin
                  jj[j]:=c; j:=j+1;
                  press:=[1];
                  form1.canvas.moveto(x,y);
                  timer1.Enabled:=false;
                  end;
            positionx[c]:=x;
            positiony[c]:=y;
            form1.canvas.LineTo(x,y);
            c:=c+1;
            end;
      end;

procedure TForm1.clear(Sender: TObject);
      begin
      Form1.canvas.brush:=brush;
      Form1.Canvas.FillRect(Form1.ClientRect);
      form1.Edit1.text:=(inttostr((pos_xy div 1000)+1));
      end;

procedure TForm1.count_clear(Sender: TObject);
      begin
      Form1.canvas.brush:=brush;
      Form1.Canvas.FillRect(Form1.ClientRect);
      pos_xy:=pos_xy+1000;
      form1.Edit1.text:=(inttostr((pos_xy div 1000)+1));
      end;


procedure TForm1.timer(Sender: TObject);
      var
      i,ii,ccx,ccy:word;

dotx,doty,axis,axisx,axisy,contourx,contoury,temp_axisx,t
emp_axisy:array_dot;
      xx,yy:array[0..500]of byte;
      axis_xx,axis_yy:array[1..100]of byte;
      xmin,xmax,ymin,ymax:integer;
      temp:array_dot_dot;
      temp_dotx,temp_doty:array_dot;
      temp_cc:word;
      temp_ymin,temp_ymax:word;
      slope_max,angle:single;
      temp_angle:extended;
      his_value_max:word;
      Hisx,Hisy:array_dot;
      Hisx_max:word;
      file_x:file;
      file_y:file;
      file_axisx:file;
      file_axisy:file;
      begin
      //test recieve positionx,y//
            //for i:=0 to c-1 do
            //begin

//form1.Canvas.Pixels[positionx[i]+100,positiony[i]]:=claq
ua;
            //end;


      ii:=0; j:=0;
      repeat
            begin
            if  jj[j+1]<>0 then
            begin
            for i:=jj[j] to jj[j+1]-2 do
                  begin
                  if
(positionx[i]<=positionx[i+1])and(positiony[i]<=positiony
[i+1])then
                        begin
                        for xxx:=positionx[i] to
positionx[i+1]do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    for yyy:=positiony[i] to
positiony[i+1]do
                        if
form1.canvas.Pixels[xxx,yyy]=clblack then
                            begin
                            dotx[ii]:=xxx;
                            doty[ii]:=yyy;
                            ii:=ii+1;
                            end;
                    end;
                if
(positionx[i]>=positionx[i+1])and(positiony[i]>=positiony
[i+1])then
                    begin
                    for xxx:=positionx[i] downto
positionx[i+1]do
                    for yyy:=positiony[i] downto
positiony[i+1]do
                        if
form1.canvas.Pixels[xxx,yyy]=clblack then
                            begin
                            dotx[ii]:=xxx;
                            doty[ii]:=yyy;
                            ii:=ii+1;
                            end;
                    end;
                if
(positionx[i]<=positionx[i+1])and(positiony[i]>=positiony
[i+1])then
                    begin
                    for xxx:=positionx[i] to
positionx[i+1]do
                    for yyy:=positiony[i] downto
positiony[i+1]do
                        if
form1.canvas.Pixels[xxx,yyy]=clblack then
                            begin
                            dotx[ii]:=xxx;
                            doty[ii]:=yyy;
                            ii:=ii+1;
                            end;
                    end;
                if
(positionx[i]>=positionx[i+1])and(positiony[i]<=positiony
[i+1])then
                    begin
                    for xxx:=positionx[i] downto
positionx[i+1]do
                    for yyy:=positiony[i] to
positiony[i+1]do
                        if
form1.canvas.Pixels[xxx,yyy]=clblack then
                            begin
                            dotx[ii]:=xxx;
                            doty[ii]:=yyy;
                            ii:=ii+1;
                            end;
                    end;
                end;
            j:=j+1;
            end;
            end;
        until jj[j+1]=0;

            for i:=jj[j] to c-2 do
                begin
                if
(positionx[i]<=positionx[i+1])and(positiony[i]<=positiony
[i+1])then
                    begin
                    for xxx:=positionx[i] to
positionx[i+1]do
                    for yyy:=positiony[i] to
positiony[i+1]do
                        if
form1.canvas.Pixels[xxx,yyy]=clblack then
                            begin
                            dotx[ii]:=xxx;
                            doty[ii]:=yyy;
                            ii:=ii+1;
                            end;
                    end;
                if
(positionx[i]>=positionx[i+1])and(positiony[i]>=positiony
[i+1])then
                    begin
                    for xxx:=positionx[i] downto
positionx[i+1]do
                    for yyy:=positiony[i] downto
positiony[i+1]do
                        if
form1.canvas.Pixels[xxx,yyy]=clblack then
                            begin
                            dotx[ii]:=xxx;
                            doty[ii]:=yyy;
                            ii:=ii+1;
                            end;
                    end;
                if
(positionx[i]<=positionx[i+1])and(positiony[i]>=positiony
[i+1])then
                    begin
                    for xxx:=positionx[i] to
positionx[i+1]do
                    for yyy:=positiony[i] downto
positiony[i+1]do
                        if
form1.canvas.Pixels[xxx,yyy]=clblack then
                            begin
                            dotx[ii]:=xxx;
                            doty[ii]:=yyy;
                            ii:=ii+1;
                            end;
                    end;
                if
(positionx[i]>=positionx[i+1])and(positiony[i]<=positiony
[i+1])then
                    begin
                    for xxx:=positionx[i] downto
positionx[i+1]do
                    for yyy:=positiony[i] to
positiony[i+1]do
                        if
form1.canvas.Pixels[xxx,yyy]=clblack then
                            begin
                            dotx[ii]:=xxx;
                            doty[ii]:=yyy;
                            ii:=ii+1;
                            end;
                    end;
                end;
        cc:=ii;
        //find xmax,xmin,ymax,ymin//
        xmin:=Find_xmin(cc,dotx);
        xmax:=Find_xmax(cc,dotx);
        ymin:=Find_ymin(cc,doty);
        ymax:=Find_ymax(cc,doty);

        //scaling//
        dotx:=scaling(55,cc,dotx,xmin,xmax);
        doty:=scaling(55,cc,doty,ymin,ymax);

        //scan pixels//
        temp:=scan_pixel(cc,dotx,doty);
        cc:=temp[0,0];
        for i:=1 to cc-1  do
            begin
            dotx[i]:=temp[0,i];
            doty[i]:=temp[1,i];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        end;

    //make bounding //
    xmin:=Find_xmin(cc,dotx);
    xmax:=Find_xmax(cc,dotx);
    ymin:=Find_ymin(cc,doty);
    ymax:=Find_ymax(cc,doty);
    //slant normalization//
    for i:=0 to cc-1 do
        begin
        temp_dotx[i]:= dotx[i];  temp_doty[i]:= doty[i];
        end;
    temp_ymin:=ymin;
    temp_ymax:=ymax;
    temp_cc:=cc;
    his_value_max:=0;
    //change slope//
    angle:=50.0;
    repeat
        begin
            temp_angle:=(angle*pi)/180.0;
            temp_angle:=tan(temp_angle);
            ymin:=temp_ymin;
            ymax:=temp_ymax;
            cc:=temp_cc;
            for i:=0 to cc-1 do
                begin
                dotx[i]:= temp_dotx[i];
                doty[i]:= temp_doty[i];
                end;
            //chang_slope//
            for i:=0 to cc-1 do
            dotx[i]:= dotx[i]-round((doty[i]-
ymin)/temp_angle);
            //find xmax,xmin//
             xmin:=Find_xmin(cc,dotx);
             xmax:=Find_xmax(cc,dotx);
            //scaling//
            dotx:=scaling(45,cc,dotx,xmin,xmax);
            doty:=scaling(45,cc,doty,ymin,ymax);
            //scan pixels//
            temp:=scan_pixel(cc,dotx,doty);
            cc:=temp[0,0];
            for i:=1 to cc-1  do
                begin
                dotx[i]:=temp[0,i];
                doty[i]:=temp[1,i];
                end;
            //find xmax,xmin//
            xmin:=Find_xmin(cc,dotx);
            xmax:=Find_xmax(cc,dotx);
            //find ymax,ymin//
            ymin:=Find_ymin(cc,doty);
            ymax:=Find_ymax(cc,doty);

            //plot histogram//
            for i:=0 to 2000 do
                begin
                Hisx[i]:=0;
                end;
            for i:=0 to cc-1 do
                begin
                Hisx[dotx[i]]:=Hisx[dotx[i]]+1;
                end;
            //find histogram_x max//
            Hisx_max:=0;
            for i:=xmin to xmax do
            if Hisx_max<Hisx[i] then
                begin
                Hisx_max:=Hisx[i];
                ii:=i;
                end;
            //save slope max//
            if his_value_max<Hisx_max then
                begin
                his_value_max:=Hisx_max;
                slope_max:=angle;
                end;
        angle:=angle+5.0;
        end;
    until angle>135.0;

    //plot the slope_max//
    temp_angle:=(slope_max*pi)/180.0;
    temp_angle:=tan(temp_angle);
    ymin:=temp_ymin;
    ymax:=temp_ymax;
    cc:=temp_cc;
    for i:=0 to cc-1 do
        begin
        dotx[i]:= temp_dotx[i];
        doty[i]:= temp_doty[i];
        end;
    //chang_slope//
    for i:=0 to cc-1 do
    dotx[i]:= dotx[i]-round((doty[i]-ymin)/temp_angle);
    //find xmax,xmin//
    xmin:=Find_xmin(cc,dotx);
    xmax:=Find_xmax(cc,dotx);
    //scaling//
    dotx:=scaling(45,cc,dotx,xmin,xmax);
    doty:=scaling(45,cc,doty,ymin,ymax);
    //scan pixels//
    temp:=scan_pixel(cc,dotx,doty);
    cc:=temp[0,0];
    for i:=1 to cc-1  do
        begin
        dotx[i]:=temp[0,i];
        doty[i]:=temp[1,i];
        end;
    //find xmax,xmin,ymax,ymin//
    xmin:=Find_xmin(cc,dotx);
    xmax:=Find_xmax(cc,dotx);
    ymin:=Find_ymin(cc,doty);
    ymax:=Find_ymax(cc,doty);

    //reset dotx,doty before neural process//
    for i:=0 to cc-1 do
        begin
        dotx[i]:=dotx[i]-xmin;
        doty[i]:=doty[i]-ymin;
        end;

    //scaling contour_tracking//
    for i:=0 to cc-1 do axis[i]:=i;
    axis:=scaling(100,cc,axis,0,cc);

    for i:=1 to 100 do
    axisx[i]:=-1;
    for i:=1 to temp[0,0]-1 do
    axisx[axis[i]]:=dotx[i];
    for i:=1 to 100 do
        begin
        if axisx[i]=-1 then
            begin
            ii:=i;
            repeat ii:=ii-1;
            until (axisx[ii]<>-1)or(ii=0);
            if (axisx[ii]<>-1)then axisx[i]:=axisx[ii]
            else
                begin
                ii:=i;
                repeat ii:=ii+1;
                until (axisx[ii]<>-1)or(ii=temp[0,0]-1);
                axisx[i]:=axisx[ii];
                end;
            end;
        end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for i:=1 to 100 do
axisy[i]:=-1;
for i:=1 to temp[0,0]-1 do
axisy[axis[i]]:=doty[i];
for i:=1 to 100 do
     begin
     if axisy[i]=-1 then
          begin
          ii:=i;
          repeat ii:=ii-1;
          until (axisy[ii]<>-1)or(ii=0);
          if (axisy[ii]<>-1)then axisy[i]:=axisy[ii]
          else
               begin
               ii:=i;
               repeat ii:=ii+1;
               until (axisy[ii]<>-1)or(ii=temp[0,0]-1);
               axisy[i]:=axisy[ii];
               end;
          end;
     end;

{ //test after scan pixels//
for i:=1 to 100 do
form1.canvas.Pixels[i,axisx[i]+300]:=clred;
for i:=1 to 100 do
form1.canvas.Pixels[i+110,axisy[i]+300]:=clblue;
for i:=0 to cc-1 do
     begin
     dotx[i]:=dotx[i]+30;
     doty[i]:=doty[i]+30;
     form1.canvas.Pixels[dotx[i]+50,doty[i]]:=clred;
     end; }
////save the  pattern for neural////
assignfile(file_x,'thai_training_x.dat');
assignfile(file_y,'thai_training_y.dat');
assignfile(file_axisx,'thai_training_axisx.dat');
assignfile(file_axisy,'thai_training_axisy.dat');

reset(file_x,4);
reset(file_y,4);
reset(file_axisx,4);
reset(file_axisy,4);
//position of record//
pos_xy:=  176000   ; //44 88 132 137 176

//save xy//
seek (file_x,pos_xy);
blockwrite(file_x,cc,1);
for i:=1 to cc do
     begin
     seek(file_x,i+pos_xy);
     blockwrite(file_x,dotx[i-1],1);
     seek( file_y,i+pos_xy);
     blockwrite(file_y,doty[i-1],1);
     end;
//save axisx axisy//
for i:=1 to 100 do
     begin
     seek( file_axisx,i+pos_xy div 10 );
     blockwrite(file_axisx,axisx[i],1);
     seek( file_axisy,i+pos_xy div 10 );
     blockwrite(file_axisy,axisy[i],1);
     end;

//test recieve//
c:=0;
for ii:=0 to (pos_xy div 1000) do
     begin
     if (ii mod 15 = 0) then c:=0;
     seek( file_x,(ii*1000));  blockread(file_x,cc,1);
     for i:=1 to cc do
          begin
          seek( file_x,i+(ii*1000));
          blockread(file_x,xx[i],1);
          seek( file_y,i+(ii*1000));
          blockread(file_y,yy[i],1);
          form1.Canvas.pixels[xx[i]+(c*50),yy[i]+(ii
div 15)*50]:=clGreen;
          end;
     c:=c+1;
     end;

//close file//
closefile(file_x);
closefile(file_y);
closefile(file_axisx);
closefile(file_axisy);

//reset//
for i:=0 to 10 do jj[i]:=0;
j:=0;
timer1.Enabled:=false;
for i:=0 to c do
     begin
     positionx[i]:=0;
     positiony[i]:=0;
     end;
c:=0;

end; //end timer
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ

โปรแกรมที่ใช้ในการทดสอบการรู้จำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unit thai_movemouse;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls, Psock, NMHttp,math, ComCtrls,
OleServer, Word97;

type
  TForm1 = class(TForm)
    Timer1: TTimer;
    Timer2: TTimer;
    Button1: TButton;
    PaintBox1: TPaintBox;

    Label1: TLabel;
    Edit45: TEdit;

    RichEdit1: TRichEdit;
    WordApplication1: TWordApplication;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    Button5: TButton;
    Button6: TButton;

    procedure setup(Sender: TObject);
    procedure clear(Sender: TObject);
    procedure timer(Sender: TObject);
    procedure Line_from_mousemove(Sender: TObject;
Shift: TShiftState;
    X,Y: Integer);
    procedure auto_clear(Sender: TObject);
    procedure open_word(Sender: TObject);
    procedure document(Sender: TObject);
    procedure exit_word(Sender: TObject);
    procedure change(Sender: TObject);
    procedure clear_richedit(Sender: TObject);
    procedure a_clear_click(Sender: TObject);


  private
  myapp:variant;
    { Private declarations }
  public
    { Public declarations }
  end;

  zero_one=set of 0..1;
  random_xy=array[1..100,1..8]of
Byte;//[1..neuralxy,1..legxy]
  random_axisxy=array[1..100,1..8]of
Byte;//[1..100,1..legaxis]
  array_dot=array[0..2000]of Smallint;
  array_dot_max=array[0..6000]of Smallint;
  array_dot_dot=array[0..1,0..1500]of word;
  array_temp_xy=array[0..1000]of word;

  function Find_xmin(count:word; pointx:array_dot):word;
  function Find_xmax(count:word; pointx:array_dot):word;
  function Find_ymin(count:word; pointy:array_dot):word;
  function Find_ymax(count:word; pointy:array_dot):word;
  function scaling(rate:real; count:word; point:array_dot;
        num_min,num_max:integer):array_dot;
  function scan_pixel(count:word; pointx:array_dot;
        pointy:array_dot):array_dot_dot;
  function scan_pixel_max(count:word;
pointx:array_dot_max;
        pointy:array_dot_max):array_dot_dot;

const
    amount_of_pattern=88;
    random_point=40;
    neuralxy=100;
    legxy=8;
    legaxis=8;
    totalaxis=800;//for 2 axis
    ////////////////////////////////////
    //  mode of random (axis) //
    //   1.random 1 axis          //
    //   2.random 2 axis          //
    //   3.random in window   //
    ////////////////////////////////////
    mode_of_random=2;
    ////////////////////////////////////
    //      score count             //
    //   1. 300 score               //
    //   2. 100 score               //
    ////////////////////////////////////
    score_count=2;

var
    Form1:TForm1;

    c,cc,ccaxis:word;
    press:zero_one;
    positionx,positiony:array_dot;
    jj:array[0..10]of byte;
    j:byte;

    xy:array[1..amount_of_pattern,1..neuralxy,1..legxy]of
Byte;

axis_x,axis_y:array[1..amount_of_pattern,1..100,1..legaxis
]of byte;

    ran_x,ran_y:random_xy;
    ran_axisx,ran_axisy:random_axisxy;

ran_axisx_x,ran_axisx_y,ran_axisy_x,ran_axisy_y:random
_axisxy;

    //test and temp for UNDELETE//
    ax,ay:array[1..amount_of_pattern,1..100]of byte;

    answer:byte;
    text:string;
    Template,NewTemplate:OLEvariant;
    doc_on:zero_one;
    on_clear:zero_one;

    //on_off_autoclear//
    count_no_off:byte;

    implementation
    uses thai_movemouse_test;
    {$R *.DFM}


procedure TForm1.setup(Sender: TObject);
    type
    pattern_xy=array[1..1500]of Byte;
    var
    randomx:file; randomaxisx:file; patx:file; pataxisx:file;
    randomy:file; randomaxisy:file; paty:file; pataxisy:file;

    i,ii,iii,iiii,no,kk,c,cc:word;
    xmin,xmax,ymin,ymax:integer;

    temp_x,temp_y:random_xy;
    patt_x,patt_y,temp_patt_x,temp_patt_y:pattern_xy;
    amountt:array[1..amount_of_pattern]of word;

dottx,dotty,temp_dottx,temp_dotty:array[1..amount_of_pat
tern,1..1000]of byte;
    temp:array_dot_dot;
    temp_dotx,temp_doty:array_dot;
    temp_dotx_max,temp_doty_max:array_dot_max;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  temp_axisx,temp_axisy:random_axisxy;

temp_axisx_x,temp_axisx_y,temp_axisy_x,temp_axisy_y:
random_axisxy;
  axissx,axissy:array[1..100]of byte;

      begin
      //reset//
      form1.timer1.Enabled:=false;
      form1.Timer2.Enabled:=false;
      count_no_off:=2;
      press:=[0];
      c:=0;
      j:=0;
      text:='';
      form1.Button3.Enabled:=false;
      form1.Button4.Enabled:=false;
      doc_on:=[0];

////////////////////////////////////////////////////////////////////////////


      //recieve random's position from file//
      assignfile(randomx,'ran_x.dat');FileMode :=
0;reset(randomx,4);
      for ii:=1 to neuralxy do
      for  i:=1 to  legxy do
            begin
            seek( randomx,i+(ii-1)*legxy);
            blockread(randomx,temp_x[ii,i],1);
            ran_x[ii,i]:=temp_x[ii,i];
            end;
      closefile(randomx);

      assignfile(randomy,'ran_y.dat');FileMode :=
0;reset(randomy,4);
      for ii:=1 to neuralxy do
      for  i:=1 to  legxy do
            begin
            seek( randomy,i+(ii-1)*legxy);
            blockread(randomy,temp_y[ii,i],1);
            ran_y[ii,i]:=temp_y[ii,i];
            end;
      closefile(randomy);

      //recieve pattern by trianing from file//
      assignfile(patx,'thai_training_x.dat'); reset(patx,4);
      assignfile(paty,'thai_training_y.dat'); reset(paty,4);
      for ii:=1 to amount_of_pattern do
            begin
            seek(patx,(ii-1)*1000);
            blockread(patx,amountt[ii],1);
            for i:=1 to amountt[ii] do
                  begin
                  seek(patx,(ii-1)*1000+i);
                  blockread(patx,temp_dottx[ii,i],1);
                  dottx[ii,i]:=temp_dottx[ii,i];

                  seek(paty,(ii-1)*1000+i);
                  blockread(paty,temp_dotty[ii,i],1);
                  dotty[ii,i]:=temp_dotty[ii,i];
                  end;
            end;
      closefile(patx);
      closefile(paty);


      for kk:=1 to  amount_of_pattern do
            begin
            cc:=amountt[kk];
            for i:=1 to cc do
                  begin
                  temp_dotx_max[i-1]:=temp_dottx[kk,i];
                  temp_doty_max[i-1]:=temp_dotty[kk,i];
                  end;
            //extened line//
            for i:=0 to cc-1 do
                  begin

      temp_dotx_max[1*cc+i]:=temp_dotx_max[i]-1;

      temp_doty_max[1*cc+i]:=temp_doty_max[i]-1;

      temp_dotx_max[2*cc+i]:=temp_dotx_max[i]-0;

      temp_doty_max[2*cc+i]:=temp_doty_max[i]-1;

      temp_dotx_max[3*cc+i]:=temp_dotx_max[i]+1;

      temp_doty_max[3*cc+i]:=temp_doty_max[i]-1;


      temp_dotx_max[4*cc+i]:=temp_dotx_max[i]-1;

      temp_doty_max[4*cc+i]:=temp_doty_max[i]-0;

      temp_dotx_max[5*cc+i]:=temp_dotx_max[i]+1;

      temp_doty_max[5*cc+i]:=temp_doty_max[i]-0;


      temp_dotx_max[6*cc+i]:=temp_dotx_max[i]-1;

      temp_doty_max[6*cc+i]:=temp_doty_max[i]+1;

      temp_dotx_max[7*cc+i]:=temp_dotx_max[i]-0;

      temp_doty_max[7*cc+i]:=temp_doty_max[i]+1;

      temp_dotx_max[8*cc+i]:=temp_dotx_max[i]+1;

      temp_doty_max[8*cc+i]:=temp_doty_max[i]+1;


      temp_dotx_max[9*cc+i]:=temp_dotx_max[i]-2;

      temp_doty_max[9*cc+i]:=temp_doty_max[i]-2;

      temp_dotx_max[10*cc+i]:=temp_dotx_max[i]-1;

      temp_doty_max[10*cc+i]:=temp_doty_max[i]-2;

      temp_dotx_max[11*cc+i]:=temp_dotx_max[i]-0;

      temp_doty_max[11*cc+i]:=temp_doty_max[i]-2;

      temp_dotx_max[12*cc+i]:=temp_dotx_max[i]+1;

      temp_doty_max[12*cc+i]:=temp_doty_max[i]-2;

      temp_dotx_max[13*cc+i]:=temp_dotx_max[i]+2;

      temp_doty_max[13*cc+i]:=temp_doty_max[i]-2;


      temp_dotx_max[14*cc+i]:=temp_dotx_max[i]-2;

      temp_doty_max[14*cc+i]:=temp_doty_max[i]+2;

      temp_dotx_max[15*cc+i]:=temp_dotx_max[i]-1;

      temp_doty_max[15*cc+i]:=temp_doty_max[i]+2;

      temp_dotx_max[16*cc+i]:=temp_dotx_max[i]-0;

      temp_doty_max[16*cc+i]:=temp_doty_max[i]+2;

      temp_dotx_max[17*cc+i]:=temp_dotx_max[i]+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp_doty_max[17*cc+i]:=temp_doty_max[i]+2;

temp_dotx_max[18*cc+i]:=temp_dotx_max[i]+2;

temp_doty_max[18*cc+i]:=temp_doty_max[i]+2;
            end;

        //scan pixels//
        temp:=scan_pixel_max(19*cc-
1,temp_dotx_max,temp_doty_max);
        cc:=temp[0.0];
        for i:=0 to cc-2  do
            begin
            temp_dotx[i]:=temp[0,i+1];
            temp_doty[i]:=temp[1,i+1];
            end;
        cc:=cc-1;
        //find xmax,xmin//
        xmin:=Find_xmin(cc,temp_dotx);
        xmax:=Find_xmax(cc,temp_dotx);
        ymin:=Find_ymin(cc,temp_doty);
        ymax:=Find_ymax(cc,temp_doty);
        //scaling//

temp_dotx:=scaling(random_point,cc,temp_dotx,xmin,xm
ax);

temp_doty:=scaling(random_point,cc,temp_doty,ymin,ym
ax);

        xmin:=Find_xmin(cc,temp_dotx);
        ymin:=Find_ymin(cc,temp_doty);

        //scan pixels//
        temp:=scan_pixel(cc,temp_dotx,temp_doty);
        cc:=temp[0.0];
        for i:=0 to cc-2  do
            begin
            temp_dotx[i]:=temp[0,i+1];
            temp_doty[i]:=temp[1,i+1];
            end;
        cc:=cc-1;
        for i:=0 to cc-1 do
            begin
            temp_dottx[kk,i+1]:=temp_dotx[i]-xmin;
            temp_dotty[kk,i+1]:=temp_doty[i]-ymin;
            end;
        amountt[kk]:=cc;
        end;

    //reset and make code xy//
    for iii:=1 to amount_of_pattern do
    for  ii:=1 to neuralxy do
    for   i:=1 to legxy do
    xy[iii][ii][i]:=0;

    for iiii:=1 to amount_of_pattern do
    for  iii:=1 to neuralxy do
    for   ii:=1 to legxy do
    for    i:=1 to amountt[iiii] do
        if (temp_dottx[iiii,i]=ran_x[iii,ii])and
          (temp_dotty[iiii,i]=ran_y[iii,ii])then
          xy[iiii,iii,ii]:=1;


////////////////////////////////////////////////////////////////////////////


    case mode_of_random of
    1:
    begin

        //receive axis's random from file//
        assignfile(randomaxisx,'ran_axisx.dat');
reset(randomaxisx,4);
        for ii:=1 to 100 do
        for  i:=1 to legaxis do
            begin
            seek(randomaxisx,(ii-1)*legaxis+i);
            blockread(randomaxisx,temp_axisx[ii,i],1);
            ran_axisx[ii,i]:=temp_axisx[ii,i];
            end;
        closefile(randomaxisx);

        assignfile(randomaxisy,'ran_axisy.dat');
reset(randomaxisy,4);
        for ii:=1 to 100 do
        for  i:=1 to legaxis do
            begin
            seek(randomaxisy,(ii-1)*legaxis+i);
            blockread(randomaxisy,temp_axisy[ii,i],1);
            ran_axisy[ii,i]:=temp_axisy[ii,i];
            end;
        closefile(randomaxisy);

        assignfile(pataxisx,'thai_training_axisx.dat');
        FileMode := 0; reset(pataxisx,4);
        assignfile(pataxisy,'thai_training_axisy.dat');
        FileMode := 0; reset(pataxisy,4);
        for iiii:=1 to amount_of_pattern do
            begin
            for no:=1 to 100 do
                begin
                seek(pataxisx,no+ (iiii-1)*100 );
                blockread(pataxisx,axissx[no],1);
                axissx[no]:=round((axissx[no]*40)/45);
                ax[iiii,no]:=axissx[no];

                seek(pataxisy,no+ (iiii-1)*100 );
                blockread(pataxisy,axissy[no],1);
                axissy[no]:=round((axissy[no]*40)/45);
                ay[iiii,no]:=axissy[no];
                end;
            for no:=1 to 100 do
                begin
                axissx[no]:=ax[iiii,no];
                axissy[no]:=ay[iiii,no];
                end;
            //make code axis_xy//
            for iii:=1 to 100 do
            for  ii:=1 to legaxis do
                begin
                if axissx[iii]<=ran_axisx[iii,ii] then
                axis_x[iiii,iii,ii]:=1
                else
                axis_x[iiii,iii,ii]:=0;
                if axissy[iii]<=ran_axisy[iii,ii] then
                axis_y[iiii,iii,ii]:=1
                else
                axis_y[iiii,iii,ii]:=0;
                end;
            end;
        closefile(pataxisx);
        closefile(pataxisy);


    end;

    2:
    begin


        //receive axisx's random from file//
        assignfile(randomaxisx,'ran_axisx.dat');
reset(randomaxisx,4);
        for ii:=1 to 100 do
        for  i:=1 to legaxis do
            begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            seek(randomaxisx,(ii-1)*legaxis+i);
            blockread(randomaxisx,temp_axisx_x[ii,i],1);
            ran_axisx_x[ii,i]:=temp_axisx_x[ii,i];

            seek(randomaxisx,(ii-1)*legaxis+i+totalaxis);
            blockread(randomaxisx,temp_axisx_y[ii,i],1);
            ran_axisx_y[ii,i]:=temp_axisx_y[ii,i];
            end;
        closefile(randomaxisx);

        //receive axisy's random from file//
        assignfile(randomaxisy,'ran_axisy.dat');
reset(randomaxisy,4);
        for ii:=1 to 100 do
        for  i:=1 to legaxis do
            begin
            seek(randomaxisy,(ii-1)*legaxis+i);
            blockread(randomaxisy,temp_axisy_x[ii,i],1);
            ran_axisy_x[ii,i]:=temp_axisy_x[ii,i];

            seek(randomaxisy,(ii-1)*legaxis+i+totalaxis);
            blockread(randomaxisy,temp_axisy_y[ii,i],1);
            ran_axisy_y[ii,i]:=temp_axisy_y[ii,i];
            end;
        closefile(randomaxisy);

        assignfile(pataxisx,'thai_training_axisx.dat');
        FileMode := 0; reset(pataxisx,4);
        assignfile(pataxisy,'thai_training_axisy.dat');
        FileMode := 0; reset(pataxisy,4);
        for iiii:=1 to amount_of_pattern do
            begin
            for no:=1 to 100 do
                begin
                seek(pataxisx,no+ (iiii-1)*100 );
                blockread(pataxisx,axissx[no],1);
                axissx[no]:=round((axissx[no]*40)/45);
                ax[iiii,no]:=axissx[no];

                seek(pataxisy,no+ (iiii-1)*100 );
                blockread(pataxisy,axissy[no],1);
                axissy[no]:=round((axissy[no]*40)/45);
                ay[iiii,no]:=axissy[no];
                end;
            for no:=1 to 100 do
                begin
                axissx[no]:=ax[iiii,no];
                axissy[no]:=ay[iiii,no];
                end;
            //make code axis_xy//
            for iii:=1 to 100 do
            for  ii:=1 to legaxis do
                begin
                if
axissx[ran_axisx_y[iii,ii]]<=ran_axisx_x[iii,ii] then
                axis_x[iiii,iii,ii]:=1
                else
                axis_x[iiii,iii,ii]:=0;
                if
axissy[ran_axisy_y[iii,ii]]<=ran_axisy_x[iii,ii] then
                axis_y[iiii,iii,ii]:=1
                else
                axis_y[iiii,iii,ii]:=0;
                end;
            end;
        closefile(pataxisx);
        closefile(pataxisy);


        end;

        3:
        begin
            end;
            end;


        end;

procedure TForm1.Line_from_mousemove(Sender:
TObject; Shift: TShiftState;
        X, Y: Integer);
        begin
        if not((ssleft in shift)or(ssright in shift)or(ssmiddle in
shift)or
        (ssdouble in shift) or(ssalt in shift) or(ssctrl in shift)
or
        (ssshift in shift))then
            begin
            if 1 in press then
                begin
                press:=[0];
                timer1.Enabled:=true;
                end;
            end;
        if ssleft in shift then
            begin
            if 0 in press then
                begin
                jj[j]:=c; j:=j+1;
                press:=[1];
                form1.canvas.moveto(x,y);
                timer1.Enabled:=false;
                end;
            positionx[c]:=x;
            positiony[c]:=y;
            form1.canvas.LineTo(x,y);
            c:=c+1;
            end;
        end;


procedure TForm1.timer(Sender: TObject);
  var
  xxx,yyy:word;
  i,ii,iii:word;
  temp_cc:word;

  xmin,xmax,ymin,ymax:integer;
  temp_ymin,temp_ymax:word;

  slope_max,angle:single;
  temp_angle:extended;
  his_value_max:word;
  Hisx,Hisy:array_dot;
  Hisx_max:word;

  dotx,doty:array_dot;
  dotx_max,doty_max:array_dot_max;
  temp_dotx,temp_doty:array_dot;
  temp:array_dot_dot;

  axis,dotaxisx,dotaxisy:array_dot;
  axisx,axisy:array[0..250]of Shortint;

  code_xy:array[1..neuralxy,1..legxy]of Byte;
  code_axisx,code_axisy:array[1..100,1..legaxis]of byte;
  count_xy:array[1..amount_of_pattern,1..neuralxy]of
Byte;

count_axisx,count_axisy:array[1..amount_of_pattern,1..10
0]of Byte;
  count_xy_axis:array[1..amount_of_pattern,1..100]of
Byte;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  score,score_xy,score_axis:array[1..amount_of_pattern]of
Byte;

Min_HM_distance,Minx_HM_distance.Miny_HM_distanc
e:word;
  Max_score:word;

    begin
    ii:=0; j:=0;
    repeat
        begin
        if  jj[j+1]<>0 then
        begin
        for i:=jj[j] to jj[j+1]-2 do
            begin
            if (positionx[i]<=positionx[i+1])and
              (positiony[i]<=positiony[i+1])then
                begin
                for xxx:=positionx[i] to
positionx[i+1]do
                for yyy:=positiony[i] to
positiony[i+1]do
                if
form1.canvas.Pixels[xxx,yyy]=clblack then
                    begin
                    dotx[ii]:=xxx;
                    doty[ii]:=yyy;
                    ii:=ii+1;
                    end;
                end;
            if (positionx[i]>=positionx[i+1])and
              (positiony[i]>=positiony[i+1])then
                begin
                for xxx:=positionx[i] downto
positionx[i+1]do
                for yyy:=positiony[i] downto
positiony[i+1]do
                if
form1.canvas.Pixels[xxx,yyy]=clblack then
                    begin
                    dotx[ii]:=xxx;
                    doty[ii]:=yyy;
                    ii:=ii+1;
                    end;
                end;
            if (positionx[i]<=positionx[i+1])and
              (positiony[i]>=positiony[i+1])then
                begin
                for xxx:=positionx[i] to
positionx[i+1]do
                for yyy:=positiony[i] downto
positiony[i+1]do
                if
form1.canvas.Pixels[xxx,yyy]=clblack then
                    begin
                    dotx[ii]:=xxx;
                    doty[ii]:=yyy;
                    ii:=ii+1;
                    end;
                end;
            if (positionx[i]>=positionx[i+1])and
              (positiony[i]<=positiony[i+1])then
                begin
                for xxx:=positionx[i] downto
positionx[i+1]do
                for yyy:=positiony[i] to
positiony[i+1]do
                if
form1.canvas.Pixels[xxx,yyy]=clblack then
                    begin
                    dotx[ii]:=xxx;
                    doty[ii]:=yyy;
                    ii:=ii+1;
                    end;
                end;
        j:=j+1;
        end;
        end;
    until jj[j+1]=0;

        for i:=jj[j] to c-2 do
            begin
            if (positionx[i]<=positionx[i+1])and
              (positiony[i]<=positiony[i+1])then
                begin
                for xxx:=positionx[i] to
positionx[i+1]do
                for yyy:=positiony[i] to
positiony[i+1]do
                if
form1.canvas.Pixels[xxx,yyy]=clblack then
                    begin
                    dotx[ii]:=xxx;
                    doty[ii]:=yyy;
                    ii:=ii+1;
                    end;
                end;
            if (positionx[i]>=positionx[i+1])and
              (positiony[i]>=positiony[i+1])then
                begin
                for xxx:=positionx[i] downto
positionx[i+1]do
                for yyy:=positiony[i] downto
positiony[i+1]do
                if
form1.canvas.Pixels[xxx,yyy]=clblack then
                    begin
                    dotx[ii]:=xxx;
                    doty[ii]:=yyy;
                    ii:=ii+1;
                    end;
                end;
            if (positionx[i]<=positionx[i+1])and
              (positiony[i]>=positiony[i+1])then
                begin
                for xxx:=positionx[i] to
positionx[i+1]do
                for yyy:=positiony[i] downto
positiony[i+1]do
                if
form1.canvas.Pixels[xxx,yyy]=clblack then
                    begin
                    dotx[ii]:=xxx;
                    doty[ii]:=yyy;
                    ii:=ii+1;
                    end;
                end;
            if (positionx[i]>=positionx[i+1])and
              (positiony[i]<=positiony[i+1])then
                begin
                for xxx:=positionx[i] downto
positionx[i+1]do
                for yyy:=positiony[i] to
positiony[i+1]do
                if
form1.canvas.Pixels[xxx,yyy]=clblack then
                    begin
                    dotx[ii]:=xxx;
                    doty[ii]:=yyy;
                    ii:=ii+1;
                    end;
                end;
            end;
    cc:=ii;
    //find xmax,xmin,ymax,ymin//
    xmin:=Find_xmin(cc,dotx);
    xmax:=Find_xmax(cc,dotx);
    ymin:=Find_ymin(cc,doty);
    ymax:=Find_ymax(cc,doty);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//scaling//
dotx:=scaling(55,cc,dotx,xmin,xmax);
doty:=scaling(55,cc,doty,ymin,ymax);

//scan pixels//
temp:=scan_pixel(cc,dotx,doty);
cc:=temp[0,0];
for i:=1 to cc-1  do
      begin
      dotx[i]:=temp[0,i];
      doty[i]:=temp[1,i];
      end;

//make bounding //
xmin:=Find_xmin(cc,dotx);
xmax:=Find_xmax(cc,dotx);
ymin:=Find_ymin(cc,doty);
ymax:=Find_ymax(cc,doty);

//slant normalization//
for i:=0 to cc-1 do
      begin
      temp_dotx[i]:= dotx[i];  temp_doty[i]:= doty[i];
      end;
temp_ymin:=ymin;
temp_ymax:=ymax;
temp_cc:=cc;
his_value_max:=0;
//change slope//
angle:=50.0;
repeat
      begin
            temp_angle:=(angle*pi)/180.0;
            temp_angle:=tan(temp_angle);
            ymin:=temp_ymin;
            ymax:=temp_ymax;
            cc:=temp_cc;
            for i:=0 to cc-1 do
                  begin
                  dotx[i]:= temp_dotx[i];
                  doty[i]:= temp_doty[i];
                  end;
            //chang_slope//
            for i:=0 to cc-1 do
            dotx[i]:= dotx[i]-round((doty[i]-
ymin)/temp_angle);
            //find xmax,xmin//
             xmin:=Find_xmin(cc,dotx);
             xmax:=Find_xmax(cc,dotx);
            //scaling//
            dotx:=scaling(40,cc,dotx,xmin,xmax);
            doty:=scaling(40,cc,doty,ymin,ymax);
            //scan pixels//
            temp:=scan_pixel(cc,dotx,doty);
            cc:=temp[0,0];
            for i:=1 to cc-1  do
                  begin
                  dotx[i]:=temp[0,i];
                  doty[i]:=temp[1,i];
                  end;

            //plot histogram//
            for i:=0 to 2000 do
                  begin
                  Hisx[i]:=0;
                  end;
            for i:=0 to cc-1 do
                  begin
                  Hisx[dotx[i]]:=Hisx[dotx[i]]+1;
                  end;
            //find histogram_x max//
            Hisx_max:=0;
            for i:=xmin to xmax do
            if Hisx_max<Hisx[i] then
                  Hisx_max:=Hisx[i];

                  //save slope max//
                  if his_value_max<Hisx_max then
                        begin
                        his_value_max:=Hisx_max;
                        slope_max:=angle;
                        end;
            angle:=angle+5.0;
            end;
until angle>135.0;

//plot the slope_max//
temp_angle:=(slope_max*pi)/180.0;
temp_angle:=tan(temp_angle);
ymin:=temp_ymin;
ymax:=temp_ymax;
cc:=temp_cc;
for i:=0 to cc-1 do
      begin
      dotx[i]:= temp_dotx[i];
      doty[i]:= temp_doty[i];
      end;
//chang_slope//
for i:=0 to cc-1 do
dotx[i]:= dotx[i]-round((doty[i]-ymin)/temp_angle);
//find xmax,xmin//
xmin:=Find_xmin(cc,dotx);
xmax:=Find_xmax(cc,dotx);
//scaling//
dotx:=scaling(40,cc,dotx,xmin,xmax);
doty:=scaling(40,cc,doty,ymin,ymax);
//scan pixels//
temp:=scan_pixel(cc,dotx,doty);
cc:=temp[0,0];
ccaxis:=temp[0,0]-1;
for i:=1 to cc-1  do
      begin
      dotx_max[i-1]:=temp[0,i];
      doty_max[i-1]:=temp[1,i];
      dotaxisx[i-1]:=temp[0,i];
      dotaxisy[i-1]:=temp[1,i];
      end;
cc:=cc-1;

////////////////////////////////////xy process//////////////////////////////

//extened line//
 for i:=0 to cc-1 do
      begin
      dotx_max[1*cc+i]:=dotx_max[i]-1;
      doty_max[1*cc+i]:=doty_max[i]-1;
      dotx_max[2*cc+i]:=dotx_max[i]-0;
      doty_max[2*cc+i]:=doty_max[i]-1;
      dotx_max[3*cc+i]:=dotx_max[i]+1;
      doty_max[3*cc+i]:=doty_max[i]-1;

      dotx_max[4*cc+i]:=dotx_max[i]-1;
      doty_max[4*cc+i]:=doty_max[i]-0;
      dotx_max[5*cc+i]:=dotx_max[i]+1;
      doty_max[5*cc+i]:=doty_max[i]-0;

      dotx_max[6*cc+i]:=dotx_max[i]-1;
      doty_max[6*cc+i]:=doty_max[i]+1;
      dotx_max[7*cc+i]:=dotx_max[i]-0;
      doty_max[7*cc+i]:=doty_max[i]+1;
      dotx_max[8*cc+i]:=dotx_max[i]+1;
      doty_max[8*cc+i]:=doty_max[i]+1;

      dotx_max[9*cc+i]:=dotx_max[i]-2;
      doty_max[9*cc+i]:=doty_max[i]-2;
      dotx_max[10*cc+i]:=dotx_max[i]-1;
      doty_max[10*cc+i]:=doty_max[i]-2;
      dotx_max[11*cc+i]:=dotx_max[i]-0;
      doty_max[11*cc+i]:=doty_max[i]-2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            dotx_max[12*cc+i]:=dotx_max[i]+1;
            doty_max[12*cc+i]:=doty_max[i]-2;
            dotx_max[13*cc+i]:=dotx_max[i]+2;
            doty_max[13*cc+i]:=doty_max[i]-2;

            dotx_max[14*cc+i]:=dotx_max[i]-2;
            doty_max[14*cc+i]:=doty_max[i]+2;
            dotx_max[15*cc+i]:=dotx_max[i]-1;
            doty_max[15*cc+i]:=doty_max[i]+2;
            dotx_max[16*cc+i]:=dotx_max[i]-0;
            doty_max[16*cc+i]:=doty_max[i]+2;
            dotx_max[17*cc+i]:=dotx_max[i]+1;
            doty_max[17*cc+i]:=doty_max[i]+2;
            dotx_max[18*cc+i]:=dotx_max[i]+2;
            doty_max[18*cc+i]:=doty_max[i]+2;
            end;

      //scan pixels//
      temp:=scan_pixel_max(19*cc-
1,dotx_max,doty_max);
      cc:=temp[0,0];
      for i:=1 to cc-1  do
            begin
            dotx[i]:=temp[0,i];
            doty[i]:=temp[1,i];
            end;

      //find xmax,xmin,ymax,ymin//
      xmin:=Find_xmin(cc,dotx);
      xmax:=Find_xmax(cc,dotx);
      ymin:=Find_ymin(cc,doty);
      ymax:=Find_ymax(cc,doty);
      //scaling//
      dotx:=scaling(random_point,cc,dotx,xmin,xmax);
      doty:=scaling(random_point,cc,doty,ymin,ymax);

      //scan pixels//
      temp:=scan_pixel(cc,dotx,doty);
      cc:=temp[0,0];
      for i:=1 to cc-1 do
            begin
            dotx[i]:=temp[0,i];
            doty[i]:=temp[1,i];
            end;
      //reset//
      xmin:=Find_xmin(cc,dotx);
      ymin:=Find_ymin(cc,doty);
      for i:=0 to cc-1 do
            begin
            dotx[i]:=dotx[i]-xmin;
            doty[i]:=doty[i]-ymin;
            end;

////////////////////////////////axis
process////////////////////////////////

      xmin:=Find_xmin(ccaxis,dotaxisx);
      ymin:=Find_ymin(ccaxis,dotaxisy);
      for i:=0 to ccaxis-1 do
            begin
            dotaxisx[i]:=dotaxisx[i]-xmin;
            dotaxisy[i]:=dotaxisy[i]-ymin;
            end;
      //for i:=0 to ccaxis-1 do

//form1.canvas.Pixels[dotaxisx[i]+5,dotaxisy[i]+50]:=clred
;

      //scaling contour_tracking//
      for i:=0 to ccaxis-1 do axis[i]:=i;
      axis:=scaling(100,ccaxis,axis,0,ccaxis);

      //scan pixels axis_x//
      for i:=1 to 100 do
      axisx[i]:=-1;
      for i:=0 to ccaxis-1 do
      axisx[axis[i]]:=dotaxisx[i];
      for i:=1 to 100 do
            begin
            if axisx[i]=-1 then
                  begin
                  ii:=i;
                  repeat ii:=ii-1;
                  until (axisx[ii]<>-1)or(ii=0);
                  if (axisx[ii]<>-1)then axisx[i]:=axisx[ii]
                  else
                        begin
                        ii:=i;
                        repeat ii:=ii+1;
                        until (axisx[ii]<>-1)or(ii=iii-1);
                        axisx[i]:=axisx[ii];
                        end;
                  end;
            end;

      //scan pixels axis_y//
      for i:=1 to 100 do
      axisy[i]:=-1;
      for i:=0 to ccaxis-1 do
      axisy[axis[i]]:=dotaxisy[i];
      for i:=1 to 100 do
            begin
            if axisy[i]=-1 then
                  begin
                  ii:=i;
                  repeat ii:=ii-1;
                  until (axisy[ii]<>-1)or(ii=0);
                  if (axisy[ii]<>-1)then axisy[i]:=axisy[ii]
                  else
                        begin
                        ii:=i;
                        repeat ii:=ii+1;
                        until (axisy[ii]<>-1)or(ii=iii-1);
                        axisy[i]:=axisy[ii];
                        end;
                  end;
            end;

//////////////////////////////neural
processing////////////////////////////////

      //make code 01 with dotx,doty//
      for ii:=1 to neuralxy do
      for  i:=1 to legxy do
      code_xy[ii][i]:=0;
      for iii:=1 to neuralxy do
      for  ii:=1 to legxy do
      for   i:=0 to cc do
            if (dotx[i]=ran_x[iii,ii])and
(doty[i]=ran_y[iii,ii])then
            code_xy[iii,ii]:=1;

      //H.M. distance of xy//
      for ii:=1 to amount_of_pattern do
      for  i:=1 to neuralxy do
      count_xy[ii][i]:=0;

      //compare code_xy with xy//
      for iii:=1 to amount_of_pattern do
      for  ii:=1 to neuralxy do
      for   i:=1 to legxy do
      if code_xy[ii][i]<>xy[iii][ii][i]then
      count_xy[iii][ii]:=count_xy[iii][ii]+1;

      case score_count of
      1:
      begin
      //find the min count//
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for i:=1 to amount_of_pattern do
    score_xy[i]:=0;
    Min_HM_distance:=10000;
    for ii:=1 to neuralxy do
          begin
          for i:=1 to amount_of_pattern do
          if count_xy[i][ii] < Min_HM_distance then
          Min_HM_distance:=count_xy[i][ii];

          for i:=1 to amount_of_pattern do
          if count_xy[i][ii] = Min_HM_distance then
          score_xy[i]:=score_xy[i]+1;
          Min_HM_distance:=10000;
          end;


    end;
    end;

////////////////////////////////////////////////////////////////////////

    //make code 01 with axisx axisy//
    for ii:=1 to 100 do
    for  i:=1 to legaxis do
          begin
          code_axisx[ii][i]:=0;  code_axisy[ii][i]:=0;
          end;

    case mode_of_random of
    1:
    begin

    for ii:=1 to 100 do
    for  i:=1 to legaxis do
          begin
          if axisx[ii]<=ran_axisx[ii,i] then
          code_axisx[ii][i]:=1
          else
          code_axisx[ii,i]:=0;
          if axisy[ii]<=ran_axisy[ii,i] then
          code_axisy[ii,i]:=1
          else
          code_axisy[ii,i]:=0;
          end;

    end;

    2:
    begin

    for ii:=1 to 100 do
    for  i:=1 to legaxis do
          begin
          if axisx[ran_axisx_y[ii,i]]<=ran_axisx_x[ii,i]
then
          code_axisx[ii][i]:=1
          else
          code_axisx[ii,i]:=0;
          if axisy[ran_axisy_y[ii,i]]<=ran_axisy_x[ii,i]
then
          code_axisy[ii,i]:=1
          else
          code_axisy[ii,i]:=0;
          end;


    end;

    3:
    begin
    end;
    end;

    //H.M. distance of axisxy//
    for ii:=1 to amount_of_pattern do
    for  i:=1 to 100 do
          begin
          count_axisx[ii,i]:=0;
          count_axisy[ii,i]:=0;
          end;
    for iii:=1 to amount_of_pattern do
    for  ii:=1 to 100 do
    for  i:=1 to legaxis do
          begin
          if axis_x[iii,ii,i]<>code_axisx[ii,i]then
          count_axisx[iii,ii]:=count_axisx[iii,ii]+1;
          if axis_y[iii,ii,i]<>code_axisy[ii,i]then
          count_axisy[iii,ii]:=count_axisy[iii,ii]+1;
          end;

    case score_count of
    1:
    begin


    //find the min count//
    Minx_HM_distance:=10000;
    Miny_HM_distance:=10000;

    //find the min count//
    for i:=1 to amount_of_pattern do
    score_axis[i]:=0;

    for ii:=1 to 100 do
          begin
          for i:=1 to amount_of_pattern do
                begin
                if Minx_HM_distance>count_axisx[i][ii]
then
                Minx_HM_distance:=count_axisx[i][ii];
                if Miny_HM_distance>count_axisy[i][ii]
then
                Miny_HM_distance:=count_axisy[i][ii];
                end;
          for i:=1 to amount_of_pattern do
                begin
                if count_axisx[i][ii]=Minx_HM_distance
then
                score_axis[i]:=score_axis[i]+1;
                if count_axisy[i][ii]=Miny_HM_distance
then
                score_axis[i]:=score_axis[i]+1;
                end;
          Minx_HM_distance:=10000;
          Miny_HM_distance:=10000;
          end;

    end;
    end;

    case score_count of
    2:
    begin

    for ii:=1 to amount_of_pattern do
    for  i:=1 to neuralxy do  //100
    count_xy_axis[ii][i]:=count_xy[ii][i]+
//xy
                count_axisx[ii,i]+count_axisy[ii,i];
//axis

    //find the min count//
    for i:=1 to amount_of_pattern do
    score[i]:=0;
    Min_HM_distance:=10000;
    for ii:=1 to neuralxy do
          begin
          for i:=1 to amount_of_pattern do
          if count_xy_axis[i][ii] < Min_HM_distance then
          Min_HM_distance:=count_xy_axis[i][ii];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        for i:=1 to amount_of_pattern do
        if count_xy_axis[i][ii] = Min_HM_distance then
        score[i]:=score[i]+1;
        Min_HM_distance:=10000;
        end;

    Max_score:=0;
    for i:=1 to amount_of_pattern do
    if score[i]>Max_score then
        begin
        Max_score:=score[i];
        answer:=i;
        end;


    end;
    end;


    case score_count of
    1:
    begin

    //find the max score//
    for i:=1 to amount_of_pattern do
    score[i]:=score_xy[i]+score_axis[i];

    Max_score:=0;
    for i:=1 to amount_of_pattern do
    if score[i]>Max_score then
        begin
        Max_score:=score[i];
        answer:=i;
        end;

    end;
    end;

////////////////////////////////////////////////////////////////////////
    on_clear:=[1];
    //display answer//
    case answer of
        1,2:form1.Edit45.Text:='¡';
        3,4:form1.Edit45.Text:='¢';
        5,6:form1.Edit45.Text:='£';
        7,8:form1.Edit45.Text:='¤';
        9,10:form1.Edit45.Text:='¥';
        11,12:form1.Edit45.Text:='¦';
        13,14:form1.Edit45.Text:='§';
        15,16:form1.Edit45.Text:='¨';
        17,18:form1.Edit45.Text:='©';
        19,20:form1.Edit45.Text:='ª';
        21,22:form1.Edit45.Text:='«';
        23,24:form1.Edit45.Text:='¬';
        25,26:form1.Edit45.Text:='-';
        27,28:form1.Edit45.Text:='®';
        29,30:form1.Edit45.Text:='¯';
        31,32:form1.Edit45.Text:='°';
        33,34:form1.Edit45.Text:='±';
        35,84:form1.Edit45.Text:='²';
        36,37:form1.Edit45.Text:='³';
        38,39:form1.Edit45.Text:='´';
        40,41:form1.Edit45.Text:='µ';
        42,43:form1.Edit45.Text:='¶';
        44,45:form1.Edit45.Text:='·';
        46,47:form1.Edit45.Text:='¸';
        48,85:form1.Edit45.Text:='¹';
        49,86:form1.Edit45.Text:='º';
        50,87:form1.Edit45.Text:='»';
        51,52:form1.Edit45.Text:='¼';
        53,54:form1.Edit45.Text:='½';
        55,56:form1.Edit45.Text:='¾';
        57,58:form1.Edit45.Text:='¿';
        59,60:form1.Edit45.Text:='À';
        61,62:form1.Edit45.Text:='Á';
        63,64:form1.Edit45.Text:='Â';
        65,66:form1.Edit45.Text:='Ã';
        67,68:form1.Edit45.Text:='Å';
        69,70:form1.Edit45.Text:='Ç';
        71,72:form1.Edit45.Text:='È';
        73,88:form1.Edit45.Text:='É';
        74,75:form1.Edit45.Text:='Ê';
        76,77:form1.Edit45.Text:='Ë';
        78,79:form1.Edit45.Text:='Ì';
        80,81:form1.Edit45.Text:='Í';
        82,83:form1.Edit45.Text:='Î';
        end;


    case answer of
        1,2:Text:='¡';
        3,4:Text:='¢';
        5,6:Text:='£';
        7,8:Text:='¤';
        9,10:Text:='¥';
        11,12:Text:='¦';
        13,14:Text:='§';
        15,16:Text:='¨';
        17,18:Text:='©';
        19,20:Text:='ª';
        21,22:Text:='«';
        23,24:Text:='¬';
        25,26:Text:='-';
        27,28:Text:='®';
        29,30:Text:='¯';
        31,32:Text:='°';
        33,34:Text:='±';
        35,84:Text:='²';
        36,37:Text:='³';
        38,39:Text:='´';
        40,41:Text:='µ';
        42,43:Text:='¶';
        44,45:Text:='·';
        46,47:Text:='¸';
        48,85:Text:='¹';
        49,86:Text:='º';
        50,87:Text:='»';
        51,52:Text:='¼';
        53,54:Text:='½';
        55,56:Text:='¾';
        57,58:Text:='¿';
        59,60:Text:='À';
        61,62:Text:='Á';
        63,64:Text:='Â';
        65,66:Text:='Ã';
        67,68:Text:='Å';
        69,70:Text:='Ç';
        71,72:Text:='È';
        73,88:Text:='É';
        74,75:Text:='Ê';
        76,77:Text:='Ë';
        78,79:Text:='Ì';
        80,81:Text:='Í';
        82,83:Text:='Î';
        end;


    //display answer for richtext//
    case answer of
        1,2:form1.RichEdit1.SelText:='¡';
        3,4:form1.RichEdit1.SelText:='¢';
        5,6:form1.RichEdit1.SelText:='£';
        7,8:form1.RichEdit1.SelText:='¤';
        9,10:form1.RichEdit1.SelText:='¥';
        11,12:form1.RichEdit1.SelText:='¦';
        13,14:form1.RichEdit1.SelText:='§';
        15,16:form1.RichEdit1.SelText:='¨';
        17,18:form1.RichEdit1.SelText:='©';
        19,20:form1.RichEdit1.SelText:='ª';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        21,22:form1.RichEdit1.SelText:='«';
        23,24:form1.RichEdit1.SelText:='¬';
        25,26:form1.RichEdit1.SelText:='-';
        27,28:form1.RichEdit1.SelText:='®';
        29,30:form1.RichEdit1.SelText:='¯';
        31,32:form1.RichEdit1.SelText:='°';
        33,34:form1.RichEdit1.SelText:='±';
        35,84:form1.RichEdit1.SelText:='²';
        36,37:form1.RichEdit1.SelText:='³';
        38,39:form1.RichEdit1.SelText:='´';
        40,41:form1.RichEdit1.SelText:='µ';
        42,43:form1.RichEdit1.SelText:='¶';
        44,45:form1.RichEdit1.SelText:='·';
        46,47:form1.RichEdit1.SelText:='¸';
        48,85:form1.RichEdit1.SelText:='¹';
        49,86:form1.RichEdit1.SelText:='º';
        50,87:form1.RichEdit1.SelText:='»';
        51,52:form1.RichEdit1.SelText:='¼';
        53,54:form1.RichEdit1.SelText:='½';
        55,56:form1.RichEdit1.SelText:='¾';
        57,58:form1.RichEdit1.SelText:='¿';
        59,60:form1.RichEdit1.SelText:='À';
        61,62:form1.RichEdit1.SelText:='Á';
        63,64:form1.RichEdit1.SelText:='Â';
        65,66:form1.RichEdit1.SelText:='Ã';
        67,68:form1.RichEdit1.SelText:='Ä';
        69,70:form1.RichEdit1.SelText:='Ç';
        71,72:form1.RichEdit1.SelText:='È';
        73,88:form1.RichEdit1.SelText:='É';
        74,75:form1.RichEdit1.SelText:='Ê';
        76,77:form1.RichEdit1.SelText:='Ë';
        78,79:form1.RichEdit1.SelText:='Ì';
        80,81:form1.RichEdit1.SelText:='Í';
        82,83:form1.RichEdit1.SelText:='Î';
        end;

    //auto clear//
    if (count_no_off mod 2 = 0) then
    form1.Timer2.Enabled:=true
    else
    form1.Timer2.Enabled:=false;

    //reset timer//
    for i:=0 to 10 do jj[i]:=0;
    j:=0;
    timer1.Enabled:=false;
    for i:=0 to c do
        begin
        positionx[i]:=0;
        positiony[i]:=0;
        end;
    c:=0;
    end; //end timer


procedure TForm1.clear(Sender: TObject);
    begin
    Form1.PaintBox1.Canvas.brush:=brush;

Form1.PaintBox1.Canvas.FillRect(Form1.ClientRect);
    form1.Edit45.Text:=(' ');
    end;

procedure TForm1.auto_clear(Sender: TObject);
    begin
    form1.PaintBox1.Canvas.brush:=brush;

Form1.PaintBox1.Canvas.FillRect(Form1.ClientRect);
    form1.Timer2.Enabled:=false;
    end;

procedure TForm1.a_clear_click(Sender: TObject);
    begin
        count_no_off:=count_no_off+1;
        if count_no_off mod 2 = 0 then
            begin
            form1.PaintBox1.Canvas.brush:=brush;

    Form1.PaintBox1.Canvas.FillRect(Form1.ClientRect);
            end;
        end;

    function Find_xmin;
        var
        temp,num:word;
        begin
        temp:=10000;
        for num:=0 to count-1 do
            begin
            if temp>pointx[num] then temp:=pointx[num];
            end;
        result:=temp;
        end;

    function Find_xmax;
        var
        temp,num:word;
        begin
        temp:=0;
        for num:=0 to count-1 do
            begin
            if temp<pointx[num] then temp:=pointx[num];
            end;
        result:=temp;
        end;

    function Find_ymin;
        var
        temp,num:word;
        begin
        temp:=10000;
        for num:=0 to count-1 do
            begin
            if temp>pointy[num] then temp:=pointy[num];
            end;
        result:=temp;
        end;

    function Find_ymax;
        var
        temp,num:word;
        begin
        temp:=0;
        for num:=0 to count-1 do
            begin
            if temp<pointy[num] then temp:=pointy[num];
            end;
        result:=temp;
        end;

    function scaling;
        var
        Find_rate:real;
        num:word;
        begin
        if(num_max-num_min)<>0then
            begin
            Find_rate:=rate/(num_max-num_min);
            for num:=0 to count-1 do
            point[num]:=round((point[num]-
    num_min)*Find_rate)+num_min;
            end;
        result:=point;
        end;

    function scan_pixel;
        var
        temp:array_dot_dot;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    num,i,ii:word;
    begin
    num:=1;
    for ii:=0 to count-1 do
        begin
        if(pointx[ii]<>-1)and(pointy[ii]<>-1)then
            begin
            for  i:=ii+1 to count-1 do
                begin

if(pointx[ii]=pointx[i])and(pointy[ii]=pointy[i]) then
                    begin
                    pointx[i]:=-1;
                    pointy[i]:=-1;
                    end;
                end;
            temp[0,num]:=pointx[ii];
            temp[1,num]:=pointy[ii];
            num:=num+1;
            end;
        end;
    temp[0.0]:=num;
    result:=temp
    end;

function scan_pixel_max;
    var
    temp:array_dot_dot;
    num,i,ii:word;
    begin
    num:=1;
    for ii:=0 to count-1 do
        begin
        if(pointx[ii]<>-1)and(pointy[ii]<>-1)then
            begin
            for  i:=ii+1 to count-1 do
                begin

if(pointx[ii]=pointx[i])and(pointy[ii]=pointy[i]) then
                    begin
                    pointx[i]:=-1;
                    pointy[i]:=-1;
                    end;
                end;
            temp[0,num]:=pointx[ii];
            temp[1,num]:=pointy[ii];
            num:=num+1;
            end;
        end;
    temp[0.0]:=num;
    result:=temp
    end;

procedure TForm1.open_word(Sender: TObject);
    begin
    //form1.WordApplication1.Connect;
    form1.WordApplication1.Visible:=true;
    form1.Button3.Enabled:=true;
    form1.Button4.Enabled:=true;
    doc_on:=[0];
    end;

procedure TForm1.document(Sender: TObject);
    begin
    Template := EmptyParam;
    NewTemplate := True;

form1.WordApplication1.Documents.add(Template,NewT
emplate);

form1.WordApplication1.Options.CheckSpellingAsYouTy
pe:=false;

form1.WordApplication1.Options.CheckGrammarAsYouT
ype:=false;
    doc_on:=[1];
    end;

procedure TForm1.exit_word(Sender: TObject);
    begin
    form1.WordApplication1.visible:=false;
    //form1.WordApplication1.Quit;
    //form1.WordApplication1.Free;
    form1.Button3.Enabled:=false;
    form1.Button4.Enabled:=false;
    doc_on:=[0];
    end;

procedure TForm1.change(Sender: TObject);
    begin
    if (1 in doc_on)and(1 in on_clear)then
    form1.WordApplication1.Selection.TypeText(text);
    end;

procedure TForm1.clear_richedit(Sender: TObject);
    begin
    on_clear:=[0];
    form1.RichEdit1.Clear;
    end;


end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จได้ด้วยดีจากความช่วยเหลือของ ผศ.ดร. สุรพันธุ์ เอื้อไพบูลย์ และ ดร. ยุทธนา คิดใจเดียว ที่ได้ให้ความกรุณาเสนอแนะวิธีการใหม่ๆ ที่เป็นประโยชน์และสร้างความกระจ่างให้กับปัญญาที่เกิดขึ้นระหว่างการดำเนินงาน ผู้จัดทำรู้สึกซาบซึ้งและขอกราบพระคุณเป็นอย่างสูง

สุดท้ายผู้จัดทำขอขอบคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และ อาจารย์ท่านอื่นๆ ที่มีส่วนส่งเสริมและประสิทธิประสาทวิชาให้ผู้จัดทำสามารถทำรายงานฉบับนี้ขึ้นมาอย่างเสร็จสมบูรณ์

ฤตี จิตต์ปราณี

(นางสาวฤตี จิตต์ปราณี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

[1] Yuttana Kitjidure (1998). The General Neural Unit : Aspect of sequence recognition. , Ph.D. Thesis, University of London.

[2] Zurada, J.M. (1992). Introduction to Artificial Neural Systems , West Publishing Company.

[3] Wong,K.Y.M. and D. Sherrington (1989). Theory of associative memory in randomly connected Boolean neural networks. , J.Phy. A : Math Gen.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้