สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การรู้จำตัวอักษรลายมือเขียนภาษาไทย

Thai Handwriting Character Recognition

T050371

โดย

นายสุพัฒน์ ธนาศรีวิไล

นายโสภณ ภัทรวงศ์หิรัญ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน 50371
วัน,เดือน,ปี 13 พ.ค. 2547
b.....
i.....

สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรู้จำตัวอักษรลายมือเขียนภาษาไทย

Thai Handwriting Character Recognition

โดย

นายสุพัฒน์ ธนาศรีวิไล 42010396

นายโสภณ ภัทรวงศ์หิรัญ 42010415



อาจารย์ที่ปรึกษา

ดร.ยุทธนา คิดใจเดียว

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2545

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การรู้จำตัวอักษรลายมือเขียนภาษาไทย

ผู้จัดทำ

1. นายสุพัฒน์ ธนาศรีวิไล เลขประจำตัว 42010396
2. นายโสภณ ภัทรวงศ์หิรัญ เลขประจำตัว 42010415



.....อาจารย์ที่ปรึกษา

(ดร.ยุทธนา คิดใจเดียว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์เรื่อง(ภาษาไทย)	**การรู้จำตัวอักษรลายมือเขียนภาษาไทย**
(ภาษาอังกฤษ)	**Thai handwriting character Regconition**
จัดทำโดย	**นายสุพัฒน์ ธนาศรีวิไล เลขประจำตัว 42010396**
	นายโสภณ ภัทรวงศ์หิรัญ เลขประจำตัว 42010415
อาจารย์ที่ปรึกษา	**ดร.ยุทธนา คิดใจเดียว**



ปริญญานิพนธ์ฉบับนี้ได้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ.....อาจารย์ที่ปรึกษา

(ดร.ยุทธนา คิดใจเดียว)

วันที่...../...../.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรู้จำตัวอักษรลายมือเขียนภาษาไทย

นายสุพัฒน์ ธนาศรีวิไล
นายโสภณ ภัทรวงศ์หิรัญ
ดร.ยุทธนา คิดใจเดียว อาจารย์ที่ปรึกษา
ภาคเรียนที่ 2 ปีการศึกษา 2545

บทคัดย่อ

รายงานฉบับนี้นำเสนอการรู้จำตัวอักษรลายมือเขียนภาษาไทยด้วยโครงข่ายประสาทเทียมแบบไร้น้ำหนัก โดยรับข้อมูลผ่านทางปากกาอิเล็กทรอนิกส์ ซึ่งขั้นตอนของการรู้จำแบ่งออกเป็น 2 ขั้น คือขั้นลดความแตกต่างของตัวอักษรและขั้นประมวลผล หลังจากนั้นทำการแปลงข้อมูลภาษาไทยที่ได้เป็นอักษรเบรลล์

ขั้นลดความแตกต่างของข้อมูล สามารถแบ่งออกได้เป็น การปรับขนาด,การหาตำแหน่งที่เสถียรของตัวอักษร,การเพิ่มความหนา,การฝึกสอนด้วยข้อมูลลายมือเขียนและข้อมูลกราฟโครงร่างตัวอักษร ขั้นตอนนี้เป็นขั้นที่มีความสำคัญอย่างมากต่อความถูกต้องของการประมวลผลด้วยโครงข่ายประสาทเทียมแบบไร้น้ำหนัก

ขั้นประมวลผลด้วยโครงข่ายประสาทเทียมแบบไร้น้ำหนัก ใช้หลักการประมวลผลโดยเลียนแบบการทำงานของหน่วยความจำชั่วคราว

ขั้นตอนการแปลงข้อมูลภาษาไทยเป็นอักษรเบรลล์ ใช้หลักการของสระแบบพิเศษและไม่ใช่สระพิเศษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thai Handwriting Character Recognition

Mr. Supat Thanasriwilai

Mr. Sophon Phattharawonghiran

Dr.Yuttana Kidjaideaw(Adviser)

2nd Semester Education Year 2002

ABSTRACT

This report present Thai handwriting character recognition using weightless Artificial neural network which used input via an electronic pen. Recognition can divide as 2 step which are preprocessing and processing after that translate to Braille Characters.

Preprocessing have many step but the main step is scaling process, finding the stabilized position, thickening ,handwriting and contour tracking graph . This step is important for processing using weightless artificial neural network used principle of random access memory (RAM).

Translating Thai's data to Braille use 2 theories that are special vowels and not special vowels.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้าที่
บทคัดย่อ	I
Abstract	II
สารบัญ	III
สารบัญรูป	V
สารบัญตาราง	V
บทที่ 1 บทนำ	1
บทที่ 2 Preprocessing	2
2.1 Preprocessing ในส่วนรูปแบบการฝึกสอนด้วย Handwriting	2
2.2 Preprocessing ในส่วนรูปแบบการฝึกสอนด้วย Contour tracking graph	7
บทที่ 3 Neural Networks	11
3.1 Artificial Neural Network	11
3.2 Weightless Artificial Neural Network	12
3.3 กระบวนการฝึกสอน Neuron	13
3.4 กระบวนการประมวลผลของ Neural Network	18
บทที่ 4 ความเป็นมาและลักษณะของอักษรเบรลล์	23
4.1 ความเป็นมาของอักษรเบรลล์	23
4.2 ลักษณะของอักษรเบรลล์	23
4.2.1 กรณีข้อมูลเป็นตัวอักษรภาษาไทย	24
4.3 การใช้งานโปรแกรม Fontographer ในการสร้างแบบตัวอักษรเบรลล์	26
4.4 หลักการในการแปลงข้อมูลภาษาไทยให้เป็นอักษรเบรลล์	29
บทที่ 5 การทดลองการทำงานของโปรแกรมและผลการทดลอง	31
การทดลองการรู้จำตัวอักษร	31
การทดลองแปลงข้อมูลจากภาษาไทยเป็นอักษรเบรลล์	34
การทดลองที่ 1 การทดลองการแปลงข้อมูลจากภาษาไทยเป็นอักษรเบรลล์	34
การทดลองที่ 2 การทดลองการแปลงข้อมูลจากภาษาไทยเป็นอักษรเบรลล์	36
บทที่ 6 วิจารณ์และข้อเสนอแนะ	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง หน้าที่

ภาคผนวก

1. ตารางแสดงพยัญชนะ , สระ , วรรณยุกต์ ที่ใช้ในโครงงานนี้
1. โปรแกรมที่ใช้ฝึกสอน Neuron
3. โปรแกรมที่ใช้ทดสอบการรู้จำ
4. ตารางการเทียบเคียงอักษรเบรลล์กับอักษรไทย

กิตติกรรมประกาศ

บรรณานุกรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปและตาราง

หน้าที่

รูปที่ 2.1 บล็อกไดอะแกรมแสดงขั้นตอนการ Preprocessing ด้วยวิธี Handwriting 3

รูปที่ 2.2 สมการที่ใช้เป็นหลักในการเอียงตัวอักษร 5

รูปที่ 2.3 Flowchart แสดงขั้นตอนการหาตำแหน่งที่เสถียรของตัวอักษร 5

รูปที่ 2.4 ภาพแสดงการพล็อต Contour tracking graph 7

รูปที่ 2.5 บล็อกไดอะแกรมแสดง Preprocessing โดยวิธี Contour tracking graph 8

รูปที่ 2.6 การ Scaling ของ Contour tracking graph เป็นขนาด 100 × 40 9

รูปที่ 3.1 เซลล์ประสาททางชีวภาพ 12

รูปที่ 3.2 โครงข่ายประสาทเทียมแบบมีน้ำหนัก 12

รูปที่ 3.3 Random Access Memory 13

รูปที่ 3.4 ภาพแสดงการรับข้อมูลของ Synapses 14

รูปที่ 3.5 การฝึกสอน Neuron ใน Neural Network โดยใช้คุณสมบัติของ RAM 14

รูปที่ 3.6 ภาพและตารางแสดงการหาตำแหน่งที่ Optimize 17

รูปที่ 3.7 ภาพแสดงตำแหน่งที่ optimize และตำแหน่งที่ Random ในโครงงานนี้ 18

รูปที่ 3.8 การประมวลผลของ Neuron ใน Neural Network โดยใช้คุณสมบัติของ RAM 19

รูปที่ 3.9 บล็อกไดอะแกรมแสดงการทำงานของ Neural Network 21

รูปที่ 4.1 แสดงการสร้างแบบตัวอักษรใหม่ของโปรแกรม Fontographer 27

รูปที่ 4.2 แสดงการสร้างแบบตัวอักษรบนโปรแกรม Fontographer 27

รูปที่ 4.3 แสดงแบบอักษรที่สร้างเสร็จแล้วของโปรแกรม Fontographer 28

รูปที่ 4.4 แสดงแบบตัวอักษรที่แปลงจากโปรแกรม Fontographer 28

รูปที่ 4.5 Flowchart แสดงการทำงานหลักของการแปลงข้อมูลภาษาไทยเป็นอักษรเบรลล์ 29

รูปที่ 4.6 Flowchart แสดงการทำงานของโปรแกรมในส่วนการแปลงข้อมูลแบบไม่ใช่สระพิเศษ 30

รูปที่ 4.7 Flowchart แสดงการทำงานของโปรแกรมในส่วนการแปลงข้อมูลแบบสระพิเศษ 30

ตารางที่ 1 ตารางแสดงผลการทดลองการรู้จำ 31

รูปที่ 5.1 แสดงการทดลองที่ 1 การแปลงข้อมูลภาษาไทยเป็นข้อมูลอักษรเบรลล์ 34

รูปที่ 5.2 แสดงการทดลองที่ 2 การแปลงข้อมูลภาษาไทยเป็นข้อมูลอักษรเบรลล์ 36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

การรู้จำตัวอักษรโดยเฉพาะอย่างยิ่งตัวอักษรแบบลายมือเขียน กระบวนการ Preprocessing และกระบวนการทาง Neural Network เป็นกระบวนการที่สำคัญอย่างยิ่งในการเพิ่มประสิทธิภาพการรู้จำตัวอักษร เพราะลายมือเขียนจะมีลักษณะแตกต่างกันไปตามแต่บุคคล ซึ่งเป็นการยากที่จะแยกแยะตัวอักษรแบบลายมือเขียนได้ ทำให้เกิดความผิดพลาดในการรู้จำตัวอักษรของ Neural Network ได้ ดังนั้นจึงต้องหาวิธีที่จะลดข้อแตกต่างของตัวอักษรในแต่ละลายมือและหาวิธีการทาง Neural Network ที่มีประสิทธิภาพเพื่อเพิ่มเปอร์เซ็นต์การรู้จำตัวอักษร

กระบวนการทาง Neural Network เป็นกระบวนการที่จำลองมาจากกระบวนการทางความคิดของมนุษย์ ซึ่งมีลักษณะพิเศษหลายประการที่เป็นประโยชน์ต่อระบบประมวลผลอันได้แก่ การรับข้อมูลแบบ Parallel Processing ทำให้สามารถรับ input ได้หลายทางมาประมวลผลในเวลาเดียวกัน ใช้หลักการง่ายๆในการตัดสินใจ 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 ยังคงให้คำตอบที่สมเหตุสมผลแม้ว่ามีสัญญาณ รบกวนหรือข้อมูลขาดความสมบูรณ์หรือข้อมูลนั้นไม่เคยเห็นมาก่อนก็ตาม และใช้เวลาในการประมวลผลและตัดสินใจที่รวดเร็ว เป็นต้น แต่ระบบ Neural Network ก็มีข้อเสียบางประการ เช่น ไม่สามารถใช้กับงานที่ต้องการความละเอียดสูงได้ ในโครงงานนี้ได้มีการจำลองการทำงานของ Neural Network ลงในคอมพิวเตอร์ด้วยการเขียนโปรแกรมการทำงานในขั้นตอนต่างๆของกระบวนการทาง Neural Network ซึ่งผู้จัดทำได้เลือกโปรแกรม Delphi ซึ่งเป็นโปรแกรมที่ใช้ภาษา Pascal เป็นตัวพัฒนา Application ต่างๆบน Window

นอกจากนี้การจัดข้อมูลก่อนการเข้าสู่การประมวลผลของ Neural หรือ preprocessing ก็เป็นส่วนสำคัญ เราต้องจัดข้อมูลให้อยู่ในรูปแบบมาตรฐานก่อนเข้าสู่กระบวนการทาง Neural Network ไม่ว่าจะเป็นการปรับขนาด ปรับความหนาและการหาตำแหน่งที่เสถียร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

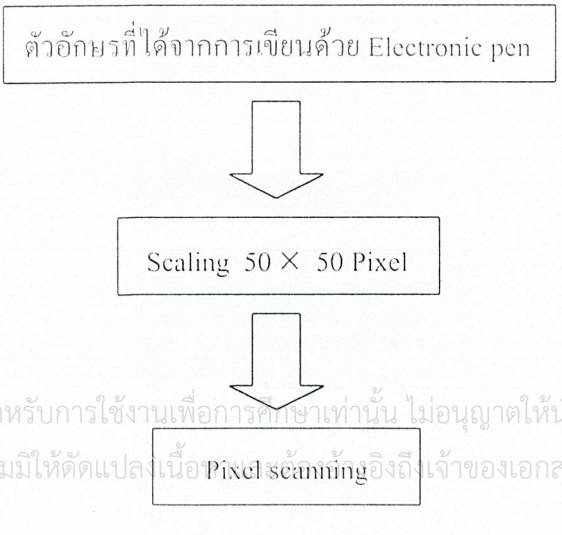
Preprocessing

ในภาคเรียนนี้ได้ทำโครงงานนี้ในช่วงเริ่มต้น โดยทำการศึกษาและพัฒนาการทำงานของระบบ Neural Network จำลองลงในคอมพิวเตอร์โดยใช้โปรแกรม Delphi เป็นโปรแกรมในการพัฒนา Application บน Window โปรแกรมนี้ใช้ภาษา Pascal เป็นหลักในการเขียน ซึ่งเป็นภาษาที่ง่ายต่อการทำความเข้าใจ พร้อมทั้งมีอุปกรณ์ในการช่วยเขียนโปรแกรมทำให้การเขียนง่ายขึ้น

โครงงานนี้เป็นการจดจำตัวอักษรภาษาไทย โดยรับข้อมูลผ่านทางปากกาอิเล็กทรอนิกส์ ซึ่งตัวอักษรภาษาไทยเป็นตัวอักษรที่มีความซับซ้อน อีกทั้งลายมือเขียนของแต่ละคนมีความแตกต่างกัน ดังนั้นจำเป็นที่จะต้องจัดการข้อมูลให้มีความเหมาะสมมากที่สุด เพื่อง่ายต่อการประมวลผลตัวอักษร ขั้นตอนก่อนการนำข้อมูลไปประมวลผลเรียกว่า "ขั้น Preprocessing" ซึ่งในภาคเรียนนี้ได้เลือกการ Preprocessing แบบลายมือเขียน (Handwriting) และแบบกราฟโครงร่างตัวอักษร (Contour tracking graph) ซึ่งจาก ข้อมูลที่ค้นคว้ากล่าวว่าทั้งสองแบบเป็น Preprocessing ในแบบที่จะให้เปอร์เซ็นต์การรู้จำที่สูงสุด ดังนั้นผู้จัดทำจึงเลือกการจัดการข้อมูลทั้งสองแบบมาทำการศึกษาและพัฒนาต่อไปในบทนี้จะกล่าวถึงขั้นตอนการ Preprocessing

2.1 Preprocessing แบบ Handwriting

รูปแบบการฝึกสอน Neural ด้วย Handwriting เป็นการฝึกสอนด้วยตัวอักษรที่ยังไม่ถูกดัดแปลงไปเป็นรูปแบบอื่น เพียงแต่จัดตัวอักษรดังกล่าวให้อยู่ในรูปแบบมาตรฐานเท่านั้น การจัดการข้อมูลให้อยู่ในรูปแบบมาตรฐานสามารถเขียนเป็นบล็อกไดอะแกรมดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.1 บล็อกไดอะแกรมแสดงขั้นตอนของ Preprocessing ด้วยวิธี Handwriting

จากบล็อกไดอะแกรมสามารถอธิบายในแต่ละส่วนได้ดังนี้

<u>ขั้นที่1</u> การรับข้อมูลตัวอักษรที่ได้จากการเขียนด้วย Electronic pen

การรับข้อมูลนั้นสามารถทำได้หลายวิธีอาจจะแบ่งวิธีการรับข้อมูลได้ 2 แบบ ได้แก่ การรับข้อมูลแบบ online และการรับข้อมูลแบบ Offline แต่ในโครงงานนี้จะเป็นการรับข้อมูลแบบ online เพื่อเป็นประโยชน์ในการเขียน Contour tracking graph ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับข้อมูลตัวอักษรในโครงงานนี้จะใช้ปากกาอิเล็กทรอนิกส์หรือ Electronic pen (เมาส์ที่เป็นปากกา) เขียนบนแผ่นรองที่มีมาพร้อมกับปากกาอิเล็กทรอนิกส์ ตัวอักษรที่ได้จากการเขียนก็จะปรากฏบนวินโดว์ฟอร์ม (Window Form) จากนั้นเก็บตำแหน่งของตัวอักษรทางด้านแกน X และแกน Y บนวินโดว์ฟอร์มในอาเรย์ (Array) ของตำแหน่ง X และ Y เพื่อนำไปสู่การจัดการในขั้นต่อไป

ขั้นที่2 Scaling 50 × 50 pixels

Scaling 50 × 50 pixels หรือการปรับขนาดตัวอักษร เพื่อทำให้ตัวอักษรมีขนาดเดียวกันหรือเป็นมาตรฐานเดียวกันก่อนการเข้าสู่กระบวนการ Preprocessingในขั้นตอนต่อไป โดยทำการปรับขนาดตัวอักษรให้มีขนาด 50 × 50 ในหน่วยของ Pixel

ขั้นที่ 3 Pixels Scanning

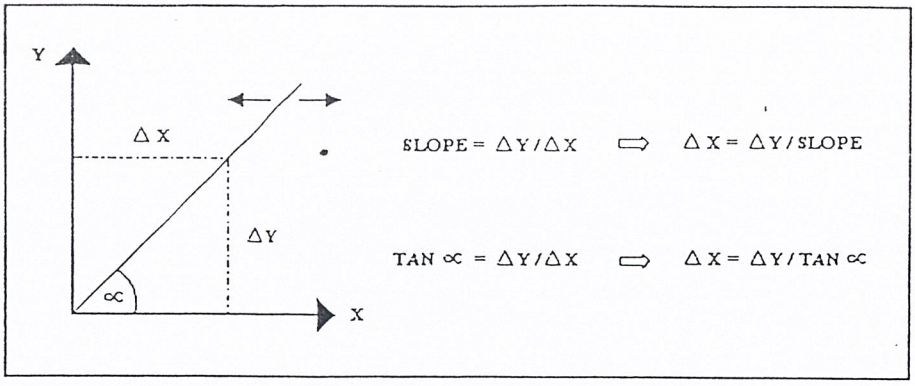
Pixel scanning คือการตัดข้อมูลของตำแหน่ง X และ Y ในอาเรย์ (Array) ที่มีการซ้ำซ้อนกันออก ซึ่งข้อมูลที่ซ้ำซ้อนกันนี้เกิดจากการกระทำการเปลี่ยนข้อมูลของตำแหน่ง X และ Y เช่นการปรับขนาด (Scaling) และการปรับมุมเอียงเพื่อหาตำแหน่งที่เสถียร

ขั้นที่ 4 ปรับมุมเอียงตัวอักษรเพื่อหาตำแหน่งที่เสถียร

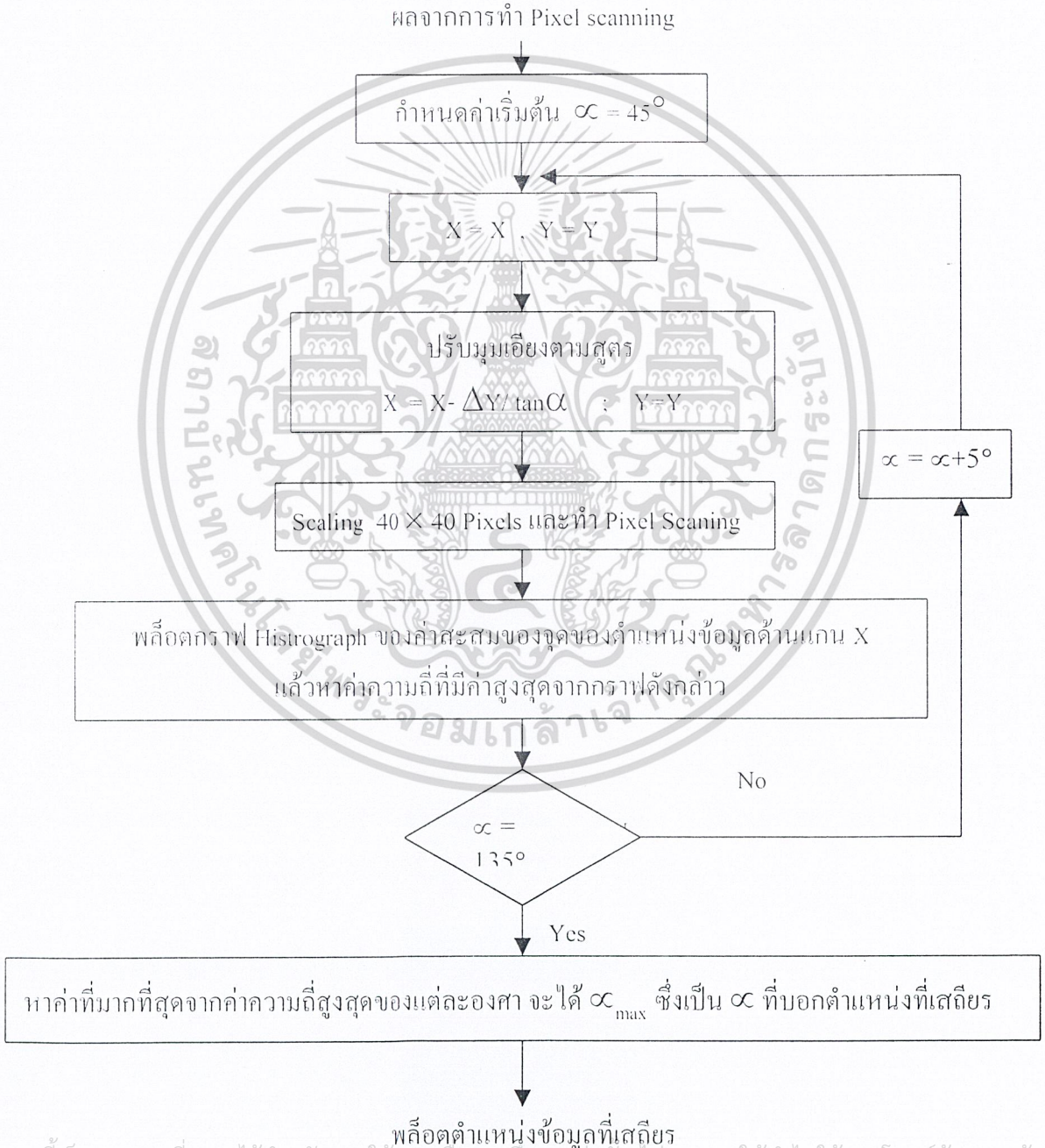
การทำให้ตัวอักษรมีตำแหน่งที่เสถียรที่สุดไม่ว่าจะด้วยวิธีใดก็ตาม จะสามารถแก้ไขความแตกต่างของลายมือได้ในระดับหนึ่ง ในโครงงานนี้ได้ใช้วิธีการเอียงตัวอักษรให้มีค่าความถี่ของกราฟ Histogram ทางด้านแกน X สูงสุดค่าๆหนึ่ง วิธีนี้จะทำให้ตัวอักษรอยู่ในตำแหน่งที่เสถียร ซึ่งข้อมูลที่เป็นลายมือต่างๆ จะอยู่ตำแหน่งนี้ได้เพียงตำแหน่งเดียว การเอียงตัวอักษรนี้จะทำได้ โดยการเลื่อนตำแหน่งของตัวอักษรเฉพาะส่วนที่อยู่ในแกน X ให้เปลี่ยนไปตามองศาต่างๆ (ตำแหน่ง Y คงที่) ซึ่งสามารถพิจารณาได้จากสูตรที่แสดงไว้ในรูปที่ 2.2 จากนั้นนำข้อมูลตำแหน่งที่ได้จากการเอียงตามองศาต่างๆ จากนั้นนำมาเปรียบเทียบหาค่าที่มากที่สุด ซึ่งค่าที่มากที่สุดจะบอกว่าที่องศาใดให้ตำแหน่งที่เสถียรที่สุด

ในโครงงานนี้ได้ทำการปรับองศาของมุมเอียงตั้งแต่ 45°-135° ซึ่งจะให้ค่า $\tan 45^{\circ} = 1$ และ $\tan 135^{\circ} = -1$ ซึ่งเป็นค่าที่ครอบคลุมของการเอียงทั้งหมด กล่าวคือ ΔX จะมีค่าเป็นบวกและลบได้ จากสูตร $\Delta X = \Delta Y / \tan\alpha$ ซึ่ง α มีค่าตั้งแต่ 45°-135° กระบวนการปรับมุมเอียงตัวอักษรเพื่อให้ได้ตำแหน่งที่เสถียรที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 สมการที่ใช้เป็นหลักในการเอียงตัวอักษร



รูปที่2.3 Flowchart แสดงขั้นตอนการหาตำแหน่งที่เสถียรของตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 5 พล็อตตำแหน่งข้อมูลที่เสถียร

การพล็อตตำแหน่งข้อมูลที่เสถียร ในส่วนกระบวนการ Preprocessing ของการฝึกสอนด้วย Handwriting การพล็อตข้อมูลจะเป็นการพล็อตตำแหน่งของข้อมูลโดยใช้สมการ

$$X = X - \Delta Y / \tan\alpha_{max} \quad ; \quad Y=Y$$

พร้อมการ Scaling 40×40 Pixels

ขั้นที่ 6 Thickening หรือการปรับความหนาตัวอักษร

การปรับความหนาตัวอักษรจากที่มีลักษณะเป็น โครงร่างบางๆให้หนาขึ้นเพื่อเพิ่มอัตราส่วนจำนวนข้อมูลระหว่างขาวกับดำ(ขาวคือส่วนที่ไม่มีข้อมูล ดำคือส่วนที่มีข้อมูล) ให้มีค่าพอๆกัน จะทำให้ข้อมูลในประสาทเทียมมีค่า 0. 1 พอๆกัน (ลดการชนกันของข้อมูล) ซึ่งจะมีประโยชน์มากต่อการประมวลผลของ neural ที่เป็นแบบ Weightless เพราะ neural จะจดจำข้อมูลจากความแตกต่าง ดังนั้นการทำข้อมูลให้มีความแตกต่างอย่างเด่นชัดจะเป็นผลดีต่อการประมวลผลของ neural อาจกล่าวได้ว่าเป็นการเพิ่มประสิทธิภาพการตัดสินใจให้กับ neural นั่นเอง

ขั้นที่ 7 Scaling 40×40 pixels

จากการ Thickening หรือการปรับความหนาของตัวอักษร จะเป็นผลให้ตัวอักษรที่มีขนาด 40×40 pixels มีขนาดใหญ่ขึ้น ดังนั้นจึงต้องทำการ Scaling 40×40 pixels หรือการปรับขนาดของตัวอักษรอีกครั้ง เพื่อทำให้ตัวอักษรมีขนาดเท่ากับ 40×40 pixels เพื่อทำให้ตัวอักษรมีขนาดเดียวกันหรือเป็นมาตรฐานเดียวกันก่อนเข้าสู่การประมวลผลของ Neural โดยทำการปรับขนาดตัวอักษรให้มีขนาด 40×40 ในหน่วยของ Pixel

ขั้นที่ 8 Pixels Scanning

Pixel scanning คือ การตัดข้อมูลของตำแหน่ง X และ Y ในอาเรย์ (Array) ที่มีการซ้ำซ้อนกันออก ซึ่งข้อมูลที่ซ้ำกันนี้เกิดจากการกระทำของการเปลี่ยนแปลงข้อมูลของตำแหน่ง X และ Y ในที่นี้เกิดจากการปรับขนาด

ขั้นที่ 9 การเก็บรูปแบบตัวอักษร

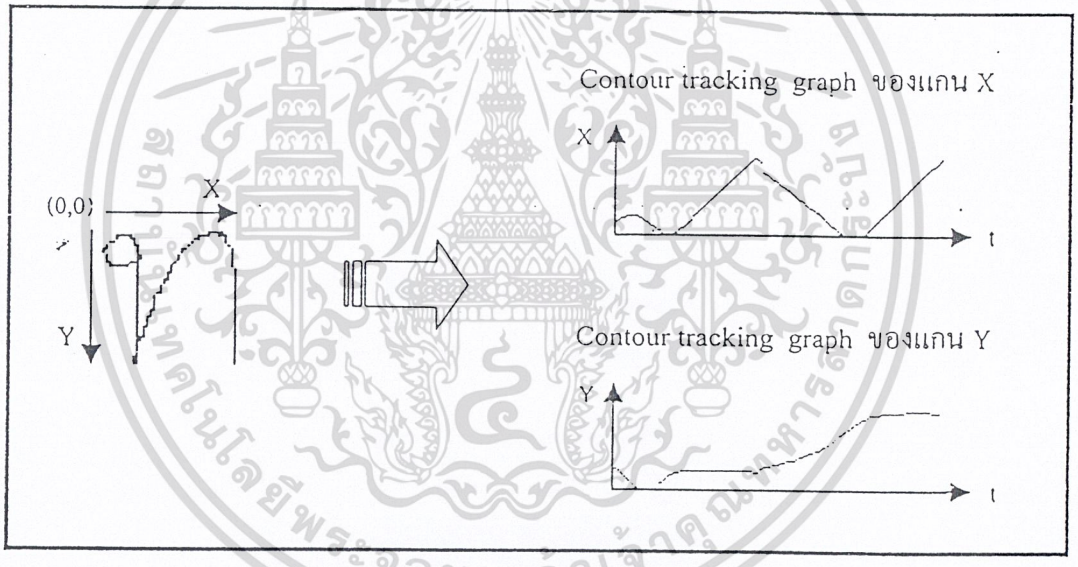
การเก็บข้อมูลจะเก็บข้อมูลตัวอักษรที่อยู่ในตำแหน่งที่เสถียรที่สุด ซึ่งตำแหน่งที่เสถียรที่สุดนี้ คือตำแหน่งที่ให้ค่าสะสมของจุดของตำแหน่งด้านแกน X (ความถี่ด้านแกน X) ของกราฟ Histogram สูงสุด โดยการเก็บข้อมูลตัวอักษรจะทำการเก็บข้อมูลอาเรย์ (Array) ของตำแหน่งทาง

เอกสารนี้มีไว้เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

ด้านแกน X และตำแหน่งทางด้านแกน Y โดยจะเก็บไว้ใน File เพื่อใช้สำหรับเป็นข้อมูลในการ Training (การฝึกสอนให้กับ Neural) ส่วนข้อมูลที่ใช้ในการทดสอบการรู้จำของ Neural ไม่ต้องจัดเก็บไว้ใน File

2.2 Preprocessing ในส่วนรูปแบบการฝึกสอนด้วย Contour tracking graph

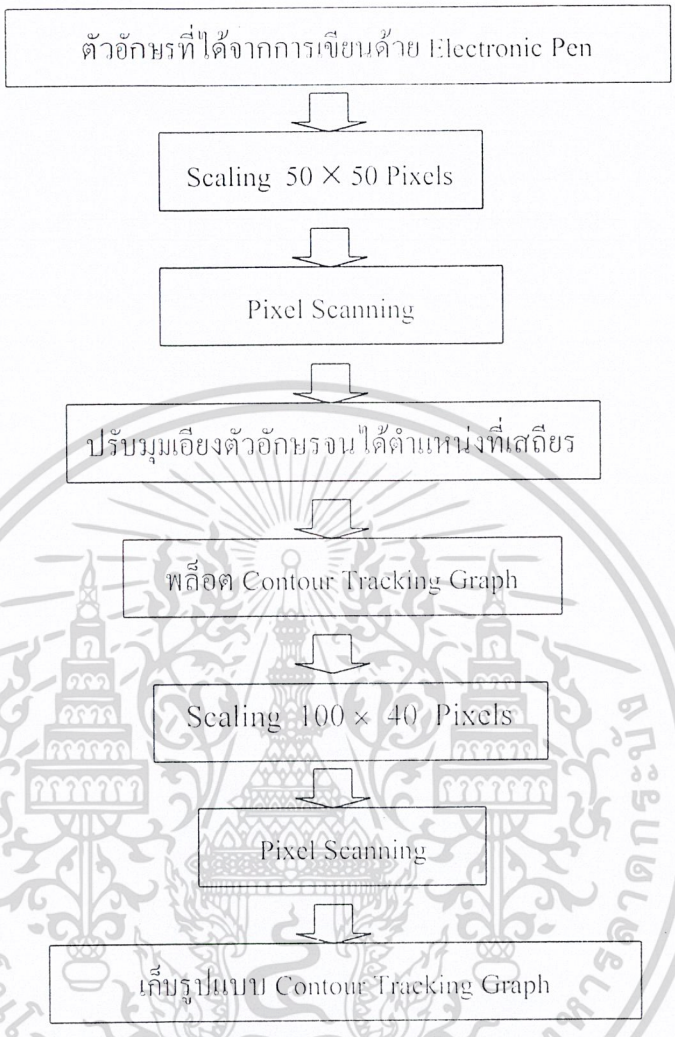
การพล็อต Contour tracking graph คือ การพล็อตค่าของตำแหน่งข้อมูลทางด้านแกน X หรือแกน Y กับแกนของเวลา หรืออาจกล่าวอีกนัยนึงว่าคือแกนบอกลำดับก่อนหลังของข้อมูล จากการเขียนข้อมูลแบบ Online โดย Electronic Pen ซึ่งจะมีขนาดของแกนเท่ากับจำนวนตำแหน่งของข้อมูลที่ต้องการพล็อตสามารถแสดงดังรูปที่ 2.4



รูปที่ 2.4 ภาพแสดงการพล็อต Contour tracking graph

กระบวนการPreprocessing ในส่วนรูปแบบการฝึกสอนด้วย Contour tracking graph จะมีขั้นตอนต่างๆเหมือนกับกระบวนการ preprocessing ในส่วนรูปแบบการฝึกสอนด้วย Handwriting จะต่างกันตรงขั้นตอนการพล็อตข้อมูลที่ได้หลังจากการเอียงข้อมูลตามองศาต่างๆ เพื่อหาตำแหน่งที่เสถียรแล้วจะเป็น Contour tracking graph แทนการพล็อตข้อมูลตามตำแหน่ง X และ Y กล่าวคือ กระบวนการ preprocessing ในขั้นตอนที่ 1-4 จะเหมือนกับการฝึกสอนด้วย Handwriting จะต่างกันตั้งแต่ขั้นตอนที่ 5 เป็นต้นไป โดยเขียนบล็อกไดอะแกรมได้ดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 บล็อกไดอะแกรมแสดงขั้นตอนการ Preprocessing ด้วยวิธี Contour tracking graph

จากบล็อกไดอะแกรมสามารถอธิบายในแต่ละส่วนได้คือ

ขั้นที่ 1-4 ในส่วนนี้จะใช้วิธีการ preprocessing เช่นเดียวกับของการฝึกสอนด้วย Handwriting

ขั้นที่ 5 พล็อต Contour tracking graph

การพล็อต Contour tracking graph จะพล็อตได้หลังจากการหาค่าของการหาตำแหน่งที่เสถียร ซึ่งหาได้จากสมการ

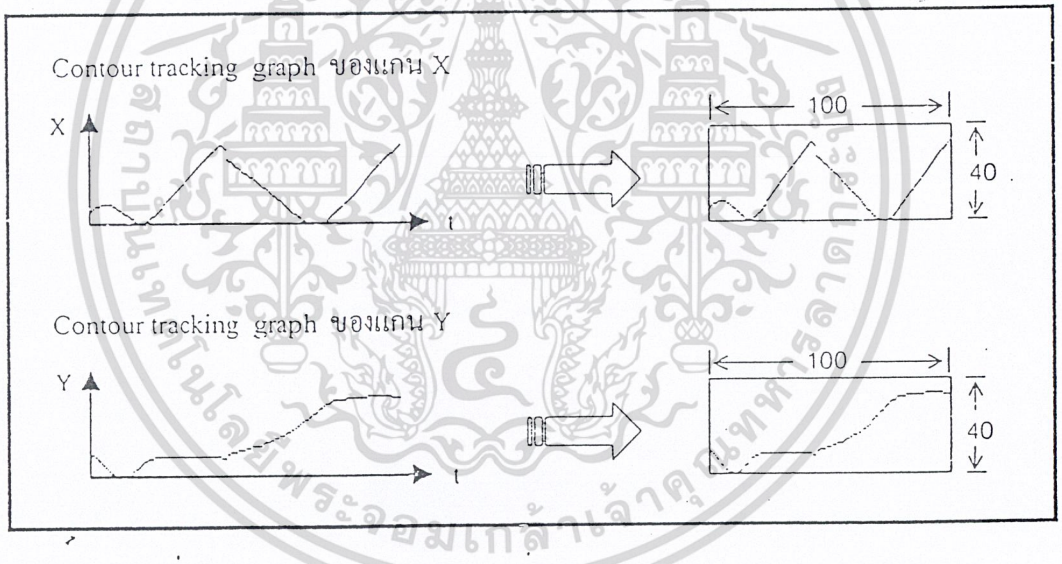
$$X = X - \Delta Y / \tan\alpha_{max}, \quad Y = Y$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วจึงนำมาพล็อต Contour tracking graph อาจกล่าวได้ว่าการพล็อต Contour tracking graph จะสามารถทำหลังจากการหาและพล็อตค่าของตำแหน่งที่เสถียรจากกระบวนการ preprocessing ในส่วนของการฝึกสอนด้วย Handwriting

ขั้นที่ 6 Scaling 100×40 pixels

การ Scaling 100×40 pixels หรือการปรับขนาดให้มีขนาด 100×40 ในหน่วยของ pixels คือการปรับขนาด Contour tracking graph จากขนาดด้านแกน X ที่เท่ากับจำนวนข้อมูลทั้งหมดและจากขนาดด้านแกน Y ที่เท่ากับขนาด 40pixels ไปเป็นขนาด 100 pixelsและ 40 pixels ตามลำดับเพื่อทำให้ Contour tracking graph มีขนาดเดียวกันหรือเป็นมาตรฐานเดียวกันก่อนการประมวลผลของ Neural



รูปที่ 2.6 การ Scaling ของ Contour tracking graph เป็นขนาด 100 × 40

ขั้นที่ 7 Pixels scanning

Pixels scanning คือการตัดข้อมูลของแกน X และ Y Contour tracking graph ที่มีการซ้ำซ้อนกันออก ซึ่งข้อมูลที่ซ้ำซ้อนนี้เกิดจากการกระทำของการเปลี่ยนแปลงข้อมูลของแกน X และ Y ในที่นี้เกิดจากการปรับขนาดหรือ Scaling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 8 เก็บรูปแบบ Contour tracking graph

การเก็บข้อมูลจะเหมือนกับการเก็บข้อมูลจากการฝึกสอนแบบ Handwriting โดยการเก็บข้อมูล Contour tracking graph จะทำการเก็บข้อมูลอาเรย์ (Array) ของตำแหน่งทางด้านแกน Y ของ Contour tracking graphเท่านั้น โดยอาเรย์ (Array)จะมีขนาดเท่ากับ 100 ตำแหน่งทางด้านแกน Y ทั้ง Contour tracking graph ของข้อมูลทางด้านแกน X และแกน Y จะเก็บไว้ใน File สำหรับข้อมูลที่ใช้ในการ Training หรือการฝึกสอนให้กับ Neural และจะนำไปประมวลผลต่อโดย Neural สำหรับข้อมูลที่ต้องการจะทดสอบการรู้จำของ Neural

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

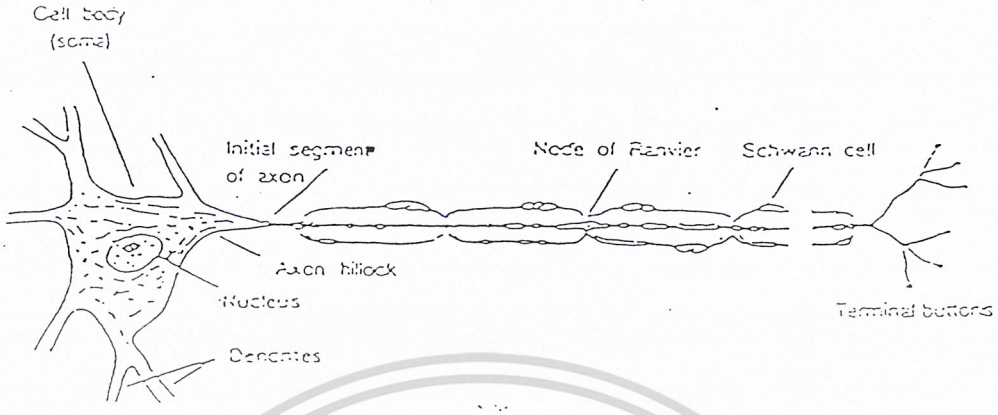
Neural Network

กระบวนการ Neural Network หรือโครงข่ายประสาทเทียมเป็นกระบวนการที่จำลองมาจากกระบวนการทางความคิดของสมองมนุษย์ซึ่งมีลักษณะพิเศษหลายประการที่เป็นประโยชน์ต่อระบบการประมวลผลดังนั้นจึงได้มีการพัฒนาและประยุกต์กระบวนการทาง Neural Network มาใช้กับงานหลายประเภท ตัวอย่างเช่น งานที่เกี่ยวกับการประมวลผลสัญญาณ (Signal Processing) , งานที่เกี่ยวกับการพัฒนาหุ่นยนต์ (Robot) และงานที่เกี่ยวกับการจดจำรูปแบบต่างๆ(Pattern Recognition) เป็นต้น ในโครงงานนี้เราจึงอาศัยหลักการทำงานของ Neural Network มาใช้ในการจดจำตัวอักษร หรือ Character Recognitionโดยปกติการพัฒนากระบวนการการรู้จำตัวอักษรเราจะใช้โครงข่ายประสาทเทียมแบบทั่วไปที่มีน้ำหนัก หรือ Artificial Neural Network: ANN แต่ในโครงงานนี้ เราใช้โครงข่ายประสาทเทียมแบบไร้น้ำหนัก หรือ Weightless Artificial Neural Network :WANN ซึ่งเป็นระบบที่ถูกพัฒนาขึ้นมาใหม่และยังไม่เป็นที่รู้จักกันอย่างแพร่หลาย แต่โครงข่ายประสาทเทียมแบบไร้น้ำหนักนี้มีข้อดีหลายประการที่น่าสนใจ อย่างเช่น ความเร็วและกระบวนการที่ง่ายกว่าเมื่อเทียบกับโครงข่ายแบบมีน้ำหนักจึงได้นำมาประยุกต์ใช้ในการรู้จำตัวอักษร ที่ได้จากการเขียนด้วยลายมือ

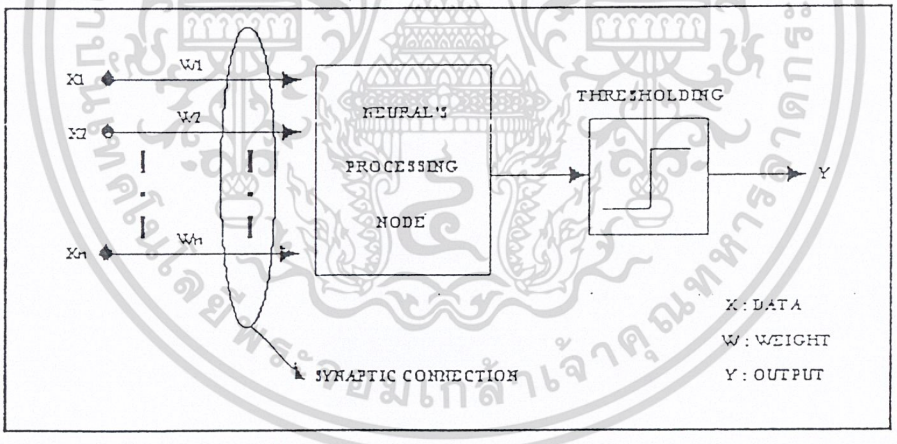
3.1 Artificial Neural Network

โครงสร้างของ Artificial Neural Network นั้นโดยทั่วไปจะประกอบไปด้วยส่วนย่อยๆที่เรียกว่า Neuron หรือ เซลล์ประสาทในทางชีวภาพซึ่งจะมีขาที่ใช้รับส่งข้อมูลหลายขาแล้วแต่การออกแบบ ซึ่งอาจเปรียบเทียบได้กับส่วนของ Synapses ในทางชีวภาพ ที่เป็นทางผ่านให้ข้อมูลหรือข่าวสารสามารถส่งผ่านจาก Neuron ตัวหนึ่งไปยัง Neuron อีกตัวหนึ่ง แต่ละ Synapses จะมีค่าน้ำหนักหรือ Weight ของมันเอง เมื่อรับข้อมูลเข้ามาแต่ละข้อมูลจะต้องนำไปคูณกับค่าน้ำหนักของ Synapses ที่รับข้อมูลนั้นๆ โดยค่าน้ำหนักนั้นได้มาจากการเรียนรู้แบบ Delta Learning Rule คือ การเปรียบเทียบกับคำตอบที่ได้จริง กับคำตอบที่ควรจะเป็นแล้วนำข้อผิดพลาดมาเข้าสมการ เพื่อหาค่าน้ำหนักที่เหมาะสม ซึ่งการทำงานของ Artificial Neural Network ในหนึ่งรอบการเรียนรู้นั้นจะต้องผ่านการคำนวณมากมาย ซึ่งแตกต่างจาก Weightless Artificial Neural Network อย่างสิ้นเชิง กล่าวคือ Weightless Artificial Neural Network นั้นจะทำงานในลักษณะของ Boolean Table 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

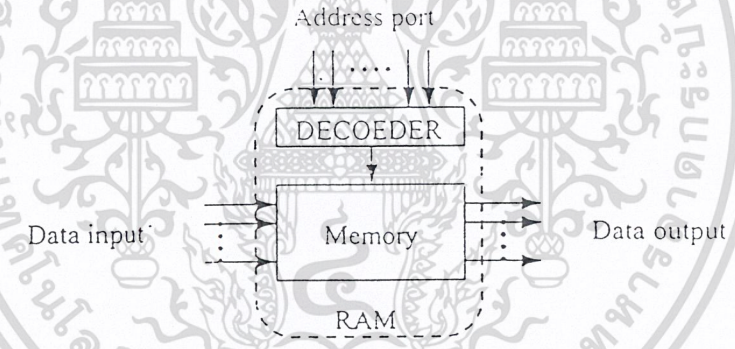
3.2 Weightless Artificial Neural Network

โครงสร้างของ Weightless Artificial Neural Network จะมีลักษณะโครงสร้างเหมือน Artificial Neural Network ต่างกันตรงที่ ไม่มีค่าน้ำหนักหรือ weight ที่ขา Synapses ซึ่งทำหน้าที่รับส่งข้อมูล ทำให้กระบวนการในการตัดสินใจง่ายกว่าและเร็วกว่าแบบ Artificial Neural Network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาก จากข้อได้เปรียบที่กล่าวมาเราจึงนำ Weightless Artificial Neural Network มาประยุกต์ใช้ในการจดจำตัวอักษร ซึ่งในการใช้งานนี้ การทำงานของโครงข่ายจะเป็นลักษณะของ Single Layer Feed Forward Neural คือมี 1 ชั้นและไม่มีการป้อนกลับ

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 ก็จะเขียนหรืออ่านข้อมูลนั้น ซึ่ง Address ของข้อมูลนั้นก็คือ ข้อมูลที่รับเข้ามาจาก Synapses ของ Neuron ซึ่งเป็นข้อมูล Binary ส่วนข้อมูลที่เขียนและอ่านที่ Address นั้นก็คือ ข้อมูลตัวอักษรที่เราต้องฝึกสอน (Training) และข้อมูลที่ได้จากการประมวลผล ตามลำดับ

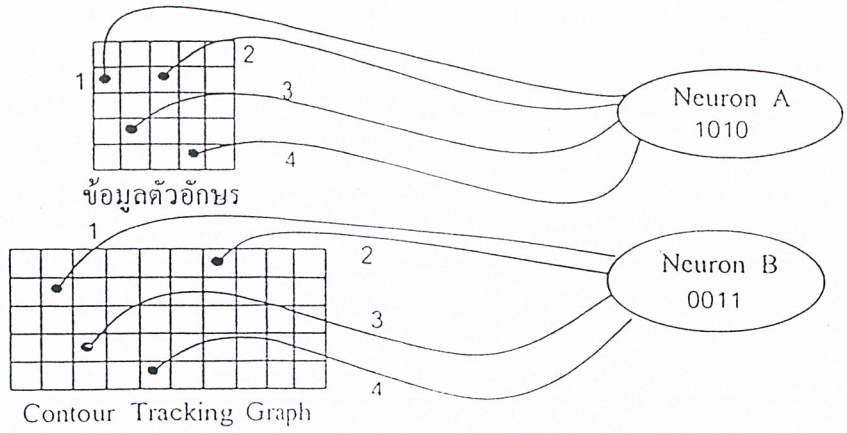


รูปที่ 3.3 Random Access Memory

3.3 กระบวนการฝึกสอน Neuron

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

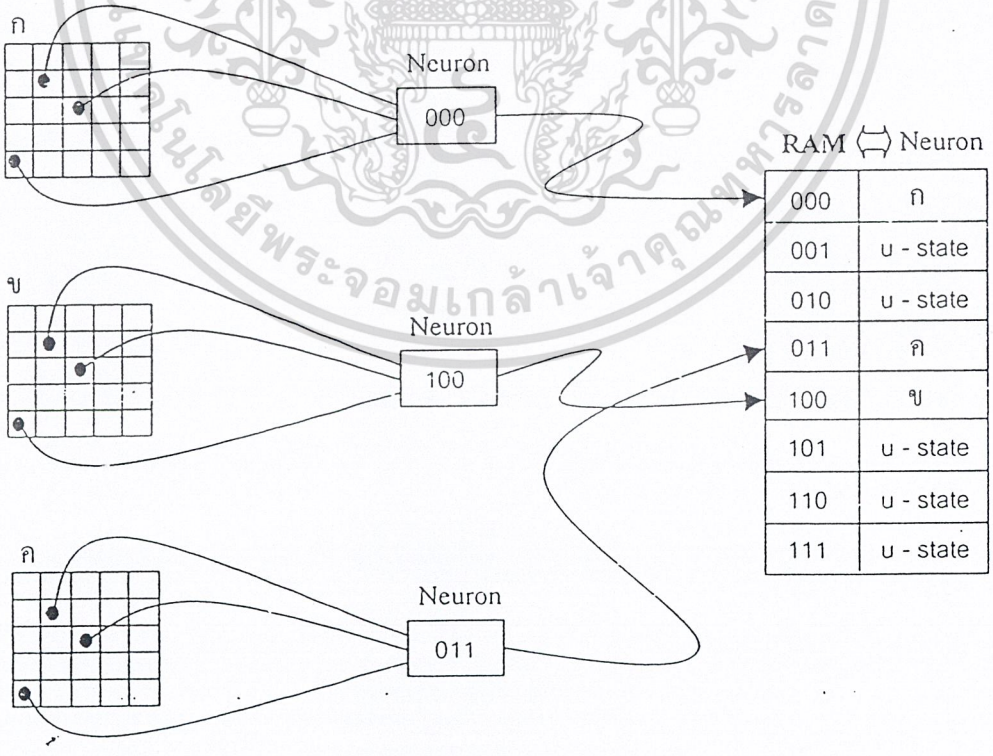
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



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 จะคล้ายการทำงานของ RAM ดังนั้นในการฝึกสอน Neuron จะกำหนดให้ทุกตำแหน่งใน Memory ก่อนการฝึกสอนอยู่ในสภาวะ u-state หรือ undefined-state คือ สภาพที่ค่าของข้อมูลใน RAM ยังไม่ถูกกำหนดชัดเจน จากนั้นจึงให้ Synapses ของ Neuron แต่ละตัวจับตามตำแหน่งที่กำหนด แล้วแปลงเป็นรหัส Binary ตามหลักที่กล่าวไปแล้ว เพื่อนำไปชี้ตำแหน่ง Address ของ RAM ที่จะถูกเขียนด้วยตัวอักษรที่ถูกจับโดย Synapses ของ Neuron นั้นๆ



รูปที่ 3.5 การฝึกสอน Neuron ใน Neural Network โดยใช้คุณสมบัติของ RAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 ตัวเดียวกับที่จับภาพตัวอักษรก็ได้

ข้อสังเกต

จากการที่แต่ละ Synapses ของ Neuron ตัวหนึ่งจับภาพของตัวอักษรหรือภาพของ Contour Tracking Graph ซึ่งเป็นไปได้ที่ค่าจากการแปลงเป็นรหัส Binary จะมีค่าซ้ำกัน ทำให้ใน 1 Adrress ของ RAM มีค่าข้อมูลมากกว่า 1 ค่า ซึ่งไม่เป็นผลดีต่อระบบการประมวลผลเท่าใดนัก ปัญหาเหล่านี้สามารถบรรเทาด้วยการเพิ่มความหนาของข้อมูลตัวอักษร (Thickening)เพื่อให้ Pixel ที่มีข้อมูล (Binary code = 1) และไม่มีข้อมูล(Binary code = 0) มีค่าพอกัน สามารถลดการชนกันของข้อมูลได้

จากประโยคที่ว่า " แต่ละ Synapses ของ Neuron จะจับตามตำแหน่งที่กำหนดบนภาพของตัวอักษรหรือภาพของ Contour Tracking Graph " เราจะพิจารณาคำว่า " ตำแหน่งที่กำหนด " ว่ามีความหมายอย่างไรในโครงงานนี้

การกำหนดตำแหน่งขาของ Neuron แต่ละตัวสามารถพิจารณาการกำหนดได้หลายวิธี อาจจะอยู่ที่ตำแหน่งที่ซ้ำกันหรือไม่ซ้ำกันก็ได้ แต่ในโครงงานนี้เราจะกำหนดให้ Synapses ของ Neuron แต่ละตัวจับในตำแหน่งที่แตกต่างกัน โดยจะใช้วิธีการหาตำแหน่งที่ Optimize ของภาพตัวอักษรและภาพของ Contour Tracking Graph ตำแหน่งที่ Optimize คือตำแหน่งที่เหมาะสม หรือตำแหน่งที่มีการเปลี่ยนแปลงของข้อมูลจากการฝึกสอนมากที่สุด หรือกล่าวอีกนัยหนึ่งคือ ตำแหน่งที่ Synapses ของ Neuron มีการเปลี่ยนแปลงข้อมูลเป็น 1 และ 0 พอๆกันในแต่ละข้อมูลการฝึกสอน หาได้โดยใช้วิธีการดังนี้

1. มองข้อมูลตัวอักษรและข้อมูล Contour Tracking Graph เป็นภาพๆหนึ่งและพิจารณาแยกกัน

2. กำหนดแบบข้อมูลที่จะทำการฝึกสอน Neuron ให้ครบทุกตัวอักษรครบทุกสระ ซึ่งในโครงงานนี้จะทำการฝึกสอน Neuron ตัวอักษรละ 3 แบบ

3. นำข้อมูลที่ทำการฝึกสอน Neuron แต่ละตัวมาพิจารณาหา Pixel ของข้อมูลที่มีการเปลี่ยนแปลงของข้อมูล Binary เป็น 1 และ 0 พอๆกันจากการพิจารณาตัวอักษรทั้งหมด ซึ่งสามารถทำได้โดยให้ทุก Pixel ที่มีข้อมูลภาพมีค่าเป็น +1 และให้ทุก Pixel ที่ไม่มีข้อมูลมีค่าเป็น –1 สำหรับการ

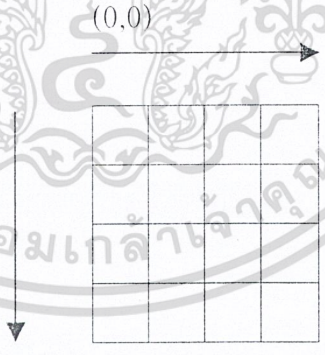
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาตัวอักษรแต่ละตัว เรียกว่าการแปลงแบบ Binary เป็น Bipolar จากนั้นนำข้อมูลที่ทำการฝึกสอน Neuron ทุกแบบมาพิจารณาตามหลักดังกล่าว

4. รวมค่าของแต่ละ Pixel ของทุกข้อมูลที่ทำการฝึกสอน Neuron ผลรวมของ Pixel ใดที่มีค่าเท่ากับศูนย์หรือใกล้เคียงศูนย์ คือ ตำแหน่งที่ Optimize

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

ตำแหน่งที่ Optimize เหล่านี้ก็จะเป็นตำแหน่งที่ให้ Synapses จับ ส่วนตำแหน่งที่เหลือก็จะทำการ Random ต่อไป นั่นคือถ้าสมมุติว่า ระบบมี Neuron 2 ตัว และ Neuron แต่ละตัวมี Synapses 4 ขา ดังนั้นจะมีตำแหน่งจับทั้งหมด 8 ตำแหน่ง ซึ่งจะต้องไม่มากกว่าจำนวน Pixel ทั้งหมดของกรอบข้อมูล และจากการที่กรอบข้อมูลมีขนาด 4×4 นั่นคือจะมีจำนวน Pixel ทั้งหมด 16 Pixel เราก็จะให้ Synapses ทั้ง 8ขาจับตำแหน่งที่ Optimize ที่สุดก่อน แล้วให้จับตำแหน่งที่ Optimize รองลงมาหรือส่วนที่เหลือจะใช้วิธีการ Random ส่วนการหาตำแหน่งที่ Optimize ของข้อมูล Contour Tracking Graph ก็ใช้หลักการเดียวกัน



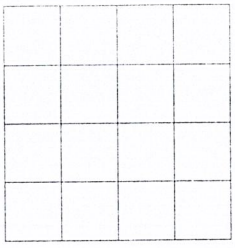
แบบที่1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

แบบที่ 2



แบบที่ 3

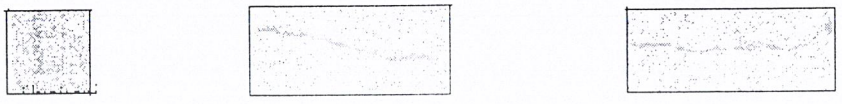
ตำแหน่ง	แบบที่ 1	แบบที่ 2	แบบที่ 3	รวม
(0,0)	-1	+1	-1	-1
(0,1)	+1	-1	+1	+1
(0,2)	-1	-1	-1	-3
(0,3)	-1	-1	-1	-3
(1,0)	+1	-1	-1	+1
(1,1)	-1	+1	-1	-1
(1,2)	+1	+1	+1	+3
(1,3)	+1	+1	-1	+1
(2,0)	+1	-1	+1	+1
(2,1)	-1	-1	-1	-3
(2,2)	-1	-1	-1	-3
(2,3)	-1	+1	+1	+1
(3,0)	-1	+1	+1	+1
(3,1)	+1	+1	+1	+3
(3,2)	+1	+1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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

รูปที่ 3.6 ภาพและตารางแสดงการหาตำแหน่งที่ Optimize

50371

ในโครงงานนี้เราได้ทำการหาตำแหน่งที่ให้ Synapses ของ Neural จับทั้งแบบการหาตำแหน่งที่ Optimize และจากการหาแบบ Random ซึ่งแสดงได้ดังรูปที่ 3.7



ตำแหน่งที่ให้ Synapses จับของข้อมูลตัวอักษร

ตำแหน่งที่ให้ Synapses จับของข้อมูล Contour Tracking Graph ด้านแกน X

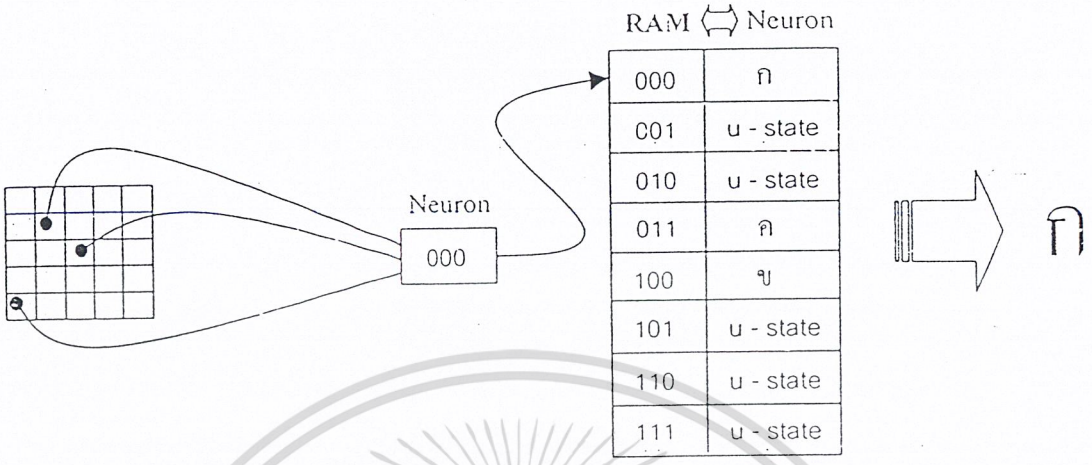
ตำแหน่งที่ให้ Synapses จับ ของข้อมูล Contour Tracking Graph ด้านแกน Y

รูปที่ 3.7 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) แต่ถ้าข้อมูลในตำแหน่ง Address ของ RAM ที่ต้องการอ่านอยู่ในสภาวะ u-state ก็จะต้องนำค่า Address นั้นมาเปรียบเทียบหาค่า Classes ที่ใกล้เคียงที่สุดทาง Hamming Distance แล้วจึงนำข้อมูลใน Address ที่ใกล้เคียงนั้นมาเป็นค่าที่ได้จากการประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 การประมวลผลของ Neuron ใน Neural Network โดยใช้คุณสมบัติของ RAM

จากการฝึกสอนของ Neuron ใน Neural Network จากภาพตัวอย่างในรูปที่ 3.5 เรามาดูการประมวลผลของ Neuron นั้นในรูปที่ 3.8 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 เป็นคำตอบที่ได้จากการประมวลผลนั่นเอง แต่ถ้าค่าที่ได้จากการแปลงเป็นรหัส Binary มีค่าเท่ากับ 001 ซึ่งข้อมูลในตำแหน่ง Address ของ RAM ที่ต้องการนั้นอยู่ในสภาวะ u – state ก็จะต้องนำค่า Address นั้นมาเปรียบเทียบหาค่า Classes ที่ใกล้เคียงที่สุดแล้วจึงนำข้อมูลใน Address ที่ใกล้เคียงที่สุดมาเป็นคำตอบ การหา Classes ที่ใกล้เคียงที่สุดสามารถหาได้โดยใช้วิธี Hamming Distance หรือวิธี H.M. Distance ซึ่งจะเป็นวิธีการเปรียบเทียบกันบิตต่อบิต โดยจะนำข้อมูลที่ต้องการเปรียบเทียบ (2 ข้อมูล) มา Excusive-or กัน แล้วนำบิตจากคำตอบที่มีค่าเป็น 1 มาบวกกันก็จะได้ค่า Hamming Distance ของข้อมูลที่เปรียบเทียบนั้น ซึ่งถ้ามีค่ามากก็จะแตกต่างกันมาก จากตัวอย่างที่กำหนดให้ค่าที่ได้จากการแปลงเป็นรหัส Binary มีค่าเท่ากับ 001 เราจะนำค่านั้นมาเปรียบเทียบกับค่า Address ของข้อมูลใน Class ต่างๆอันได้แก่ตำแหน่ง Address ที่ 000, 011 ,100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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	001
H.M. distance	1

ข้อมูลที่ทดสอบที่มีรหัส Binary เท่ากับ 001 เปรียบเทียบกับข้อมูลที่ใช้ฝึกสอนที่มีค่า Address เท่ากับ 100

ข้อมูลที่ทดสอบ	001
ข้อมูลที่ฝึกสอน	100
Excusive-or	101
H.M.Distance	2

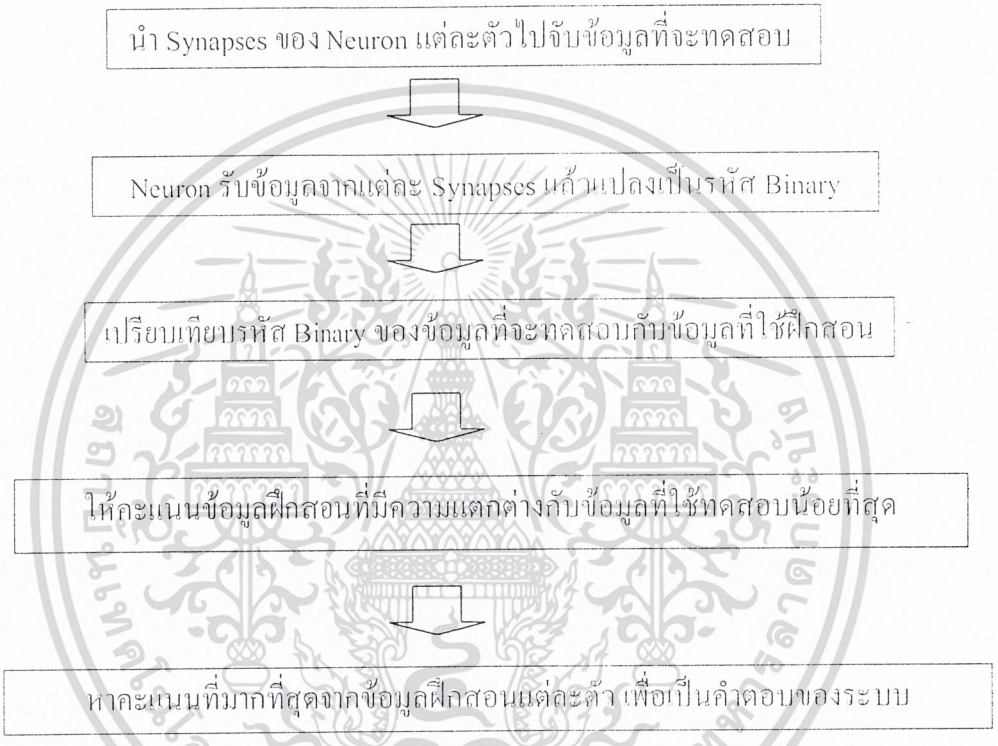
จากการเปรียบเทียบด้วยวิธีการหาค่า H.M.Distance จะได้คำตอบที่น้อยที่สุดคือ 1 ซึ่งมีอยู่ 2 ค่า นั้นคือที่ตำแหน่ง Address 011,100 ที่ชี้ข้อมูล ก และ ค อยู่ ดังนั้นการประมวลผล Neuron ตัวนี้จะให้คำตอบ 2 ค่า แต่คำตอบนี้ยังไม่ใช่คำตอบของระบบเพราะยังมี Neuron ตัวอื่นๆอีก ที่อาจให้คำตอบเป็นอย่างอื่น ซึ่งการประมวลผลของ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

ในโครงงานนี้เราจะใช้หลักการตัดสินใจของระบบ Neural Network โดยการให้ 1 Neuron แทนด้วยคะแนน 1 คะแนนของข้อมูลหรือตัวอักษรที่ได้จากการประมวลผลของ Neuron นั้นๆดังนั้น ถ้าให้ Neural Network มี Neuron ทั้งหมด x ตัว ก็สามารถกล่าวได้ว่าระบบจะมีคะแนนเต็ม x

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คะแนนซึ่งแต่ละคะแนนจะกระจายไปตามข้อมูลที่ได้จากการประมวลผลของแต่ละ Neuron และข้อมูลหรืออักษรที่มีคะแนนสูงที่สุดก็จะเป็นคำตอบของระบบนั่นเอง

ขั้นตอนการประมวลผล Weightless Neural Network ที่กล่าวในข้างต้นสามารถเขียนสรุปเป็นขั้นตอนต่างๆ อย่างคร่าวๆได้ดังนี้



รูปที่ 3.9 บล็อกไดอะแกรมแสดงการทำงานของ Neural Network

จากบล็อกไดอะแกรมสามารถอธิบายได้ดังนี้

ขั้นที่ 1 นำ Synapses ของ Neuron แต่ละตัวไปจับข้อมูลที่จะทดสอบ

ตำแหน่งที่กำหนดให้ Synapses ของ Neuron แต่ละตัวจับก็คือตำแหน่งเดียวกับตำแหน่งของ Neuron แต่ละตัวจับจากการฝึกสอน ซึ่งได้จากการหาตำแหน่งที่ Optimize และได้จากการ Random ตำแหน่ง ซึ่งโครงงานนี้เราจะใช้ Neuron ทั้งหมด 100 ตัว ตัวหนึ่งมี Synapses 30 ขา สำหรับจับภาพข้อมูลตัวอักษร 10 ขา สำหรับจับข้อมูล Contour Tracking Graph ในแต่ละแกนของการพล็อต แกนละ 10 ขา นั่นคือระบบจะมีคะแนนเต็มจากการประมวลผลทั้งหมด 100 คะแนน

เอกสารนี้สงวนลิขสิทธิ์ ห้ามนำไปใช้เพื่อการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ขอสงวนสิทธิ์ในการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

ขั้นที่ 2 Neuron รับข้อมูลจากแต่ละ Synapses แล้วแปลงเป็นรหัส Binary

การแปลงเป็นรหัส Binary จากการรับข้อมูลของ Synapses แต่ละ Synapses ของ Neuron คือ เราจะมองข้อมูลตัวอักษรและข้อมูล Contour Tracking Graph เป็นภาพๆหนึ่ง และจะกำหนดให้ Pixel ที่มีข้อมูลมีค่าเป็น 1 และ Pixel ที่ไม่มีข้อมูลมีค่าเป็น 0

ขั้นที่ 3 เปรียบเทียบรหัส Binary ของข้อมูลที่จะทดสอบกับข้อมูลที่ใช้ฝึกสอน

การเปรียบเทียบรหัส Binary ของข้อมูลที่จะทดสอบกับข้อมูลที่ใช้ฝึกสอนจะใช้วิธีหา Hamming Distance หรือวิธีเปรียบเทียบกันบิตต่อบิต 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 ที่มีค่าน้อยที่สุดก็จะเป็นค่าที่บอกว่าข้อมูลที่ใช้ทดสอบมีค่าใกล้เคียงกับข้อมูลที่ใช้ฝึกสอนตัวนั้นมากสุด เราก็จะให้คะแนนข้อมูลฝึกสอนตัวนั้น 1 คะแนน ซึ่งข้อมูลฝึกสอนที่ได้จากการเปรียบเทียบนั้นอาจมีมากกว่าหนึ่งข้อมูล เราก็จะให้คะแนนข้อมูลฝึกสอนทั้งหมดที่ได้นั้นข้อมูลละ 1 คะแนน

ขั้นที่ 5 หาคะแนนที่มากที่สุดจากข้อมูลฝึกสอนแต่ละตัว เพื่อเป็นคำตอบของระบบ

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

ในโครงงานนี้เราใช้ Neural Network จดจำตัวอักษร ก-ฮ พร้อมด้วยสระต่างๆอันได้แก่ อะ , อา , อิ , อี , อึ , อื , โอ , อำ , ไอ , ใอ , เอ , แอ , อุและ อู วรรณยุกต์ต่างๆ ได้แก่ อ่ , อ้ , อ๊ , อ๋ เครื่องหมายต่างๆได้แก่ อ์ , อ็ , อ๎ อักขระพิเศษ ได้แก่ ฯ , ๆ , ฤ และ ฤๅ ซึ่งประสิทธิภาพหรือเปอร์เซ็นต์การจดจำจะแสดงในบทต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ความเป็นมาและลักษณะของอักษรเบรลล์

4.1 ความเป็นมาของอักษรเบรลล์

อักษรเบรลล์เริ่มต้นขึ้นโดยครูตาบอดชาวฝรั่งเศส หลุยส์ เบรลล์ เป็นผู้ประดิษฐ์อักษรสำหรับคนตาบอดได้ใช้อ่านโดยการใช้นิ้วสัมผัส การประดิษฐ์อักษรของเขาได้รับความคิดมาจากการส่งข่าวสารทางทหารในเวลากลางคืนของกัปตันชาร์ลส์ บาร์บิเอร์ ซึ่งใช้กระดาษแข็งปั๊มเป็นรหัสจุดและขีด โดยได้พัฒนามาเป็นระบบ 6 จุดซึ่งสามารถจัดกลุ่มของจุดได้ถึง 63 แบบ ใช้แทนอักษรตาดี อักษรเบรลล์ได้เข้ามาในประเทศไทยและพัฒนาให้ใช้กับภาษาไทยจากการนำมาเผยแพร่ของเจนีวีฟ คอลฟิลด์ สุภาพสตรีอเมริกันตาบอด ในปี พ.ศ. 2482 จึงเป็นการเปิดศักราชแห่งการเรียนรู้หนังสือของคนตาบอดในประเทศไทย ทำให้คนตาบอดไทยได้รับความรู้จากการอ่านหนังสือผ่านอักษรเบรลล์ตั้งแต่นั้นเป็นต้นมา

4.2 ลักษณะของอักษรเบรลล์

อักษรเบรลล์จะมีลักษณะเป็นจุดนูนเล็กๆ ใน 1 ช่องประกอบด้วยจุด 6 ตำแหน่งซึ่งนำมาจัดสลับไปมาเป็นรหัสแทนอักษรตาดี หรือสัญลักษณ์ทางคณิตศาสตร์ วิทยาศาสตร์ โน้ตดนตรี ฯลฯ การเขียนใช้เครื่องมือเฉพาะเรียก สเลท และดินสอ การพิมพ์ใช้เครื่องพิมพ์เรียกว่าเบรลเล่อร์ ใช้กระดาษหนาขนาดกระดาษวาดรูป

ข้อกำหนดในการเขียนอักษรเบรลล์เปรียบเทียบกับอักษรคนตาดี

กรณีข้อมูลเป็นอักษรภาษาไทย

- คำที่ไม่อยู่ในรูปแบบเฉพาะ
- คำที่มีรูปแบบเฉพาะ ซึ่งสะกดอยู่ด้วยสระดังนี้ คือ “สระเอะ, สระแอะ , สระโอะ, สระเอาะ, สระเออะ, สระเออ, สระเอีย, สระเอือ, สระอัว,สระเอา, สระเอือ
- คำที่มีรูปแบบเฉพาะที่มีตัวควบกล้ำ
- คำที่มีรูปแบบเฉพาะที่มีวรรณยุกต์

กรณีข้อมูลเป็นอักษรภาษาอังกฤษ

- อักษรภาษาอังกฤษเป็นตัวพิมพ์ใหญ่
- อักษรภาษาอังกฤษเป็นตัวพิมพ์เล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีข้อมูลเป็นตัวเลข

- ข้อมูลตัวเลขที่ใช้ในข้อความ
- ข้อมูลตัวเลขที่เป็นทศนิยมหรือใช้ในแบบคณิตศาสตร์

ซึ่งในปริญญานิพนธ์ฉบับนี้ได้จัดทำเฉพาะกรณีข้อมูลเป็นตัวอักษรภาษาไทยเท่านั้น

4.2.1 กรณีข้อมูลเป็นตัวอักษรภาษาไทย

การเขียนเป็นหนังสือเบรลล์ขึ้นอยู่กับว่าคำภาษาไทยนั้นๆ จัดอยู่ในรูปแบบเฉพาะตามที่อักษรเบรลล์ได้กำหนดไว้หรือไม่ ถ้าไม่อยู่ในเงื่อนไขดังกล่าวก็ถือว่าคำภาษาไทยนั้นๆ เป็นคำภาษาไทยทั่วๆไป ซึ่งเมื่อเขียนเป็นหนังสือเบรลล์จะมีลำดับการเขียนเหมือนกับหนังสือคนตาดี แต่ถ้ารูปแบบของคำภาษาไทยนั้นๆ อยู่ในเงื่อนไขตามที่อักษรเบรลล์ได้กำหนดไว้ก็ถือว่าคำภาษาไทยนั้น จัดอยู่ในรูปแบบเฉพาะซึ่งก็จะทำให้การเขียนเป็นหนังสือเบรลล์จะมีลำดับการเขียนแตกต่างไปจากเดิม โดยมีกรรมวิธีที่จะอธิบายได้ดังนี้ คือ ถ้าคำภาษาไทยนั้นๆ จัดอยู่ในเงื่อนไขของรูปแบบเฉพาะก็จะเขียนอักษรเบรลล์ที่ใช้ตัวแทนพยัญชนะภาษาไทยนั้นๆก่อน แล้วจึงตามด้วยอักษรเบรลล์ที่ใช้แทนความหมายของรูปแบบของคำภาษาไทย รายละเอียดและตัวอย่างประกอบคำอธิบายจะได้แสดงในลำดับถัดไป ดังนี้

ตัวอย่างประกอบคำอธิบายรูปแบบของคำภาษาไทยที่ไม่จัดอยู่ในเงื่อนไขแบบเฉพาะคำภาษาไทยตามที่อักษรเบรลล์ได้กำหนด เช่น

คำว่า “สิ่ง” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ส-สระอิ-ไม้เอก-ง”
คำว่า “ราคา” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ร-สระอา-ค-สระอา”
คำว่า “แต่” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “สระแอ-ต-ไม้เอก”
คำว่า “ชมรม” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ช-ม-ร-ม”
คำว่า “ปฏิทิน” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ป-ฏ-สระอิ-ท-สระอิ-น”
คำว่า “หนังสือ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ห-น-ไม้หันอากาศ-ง-ส-สระอือ-อ”
คำว่า “บรรยาย” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “บ-รร-ย-สระอา-ย”
คำว่า “น้ำใจ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “น-ไม้โท-สระอำ-ไม้ม้วน-จ”

รูปแบบคำเฉพาะของภาษาไทยคือ รูปแบบที่มีการสะกดของคำที่มีสระดังต่อไปนี้
“สระเอะ, สระแอะ, สระโอะ, สระเออะ, สระเออ, สระเอีย, สระเอียะ, สระเอือ, สระเอือะ, สระอัว, สระเอา, รูปแบบพิเศษสระเออที่ใช้สระเออ แทน”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างประกอบคำอธิบาย ถึงลักษณะของคำเฉพาะภาษาไทยที่จัดอยู่ในรูปแบบของคำเฉพาะที่มีพยัญชนะเพียงตัวเดียวตามที่อักษรเบรลล์ได้กำหนดไว้ดังนี้ เช่น

คำว่า “เตะ ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ ต-สระเอะ ”
คำว่า “และ ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ ล-สระแอะ ”
คำว่า “ โละ ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ ล-สระโอะ ”
คำว่า “ เกาะ ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ ก-สระเอาะ ”
คำว่า “ เลอะ ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ ล-สระเออะ ”
คำว่า “ เธอ ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ ธ-สระเออ ”
คำว่า “ เลีย ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ ล-สระเอีย ”
คำว่า “ เรือ ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ ร-สระเอือ ”
คำว่า “ วัว ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ ว-สระอัว ”
คำว่า “ เรา ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ ร-สระเอา ”
คำว่า “ เดิน ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ ด-สระเออ-น ”

ตัวอย่างที่ได้แสดงข้างต้นเป็นคำที่มีพยัญชนะเพียงตัวเดียว ซึ่งเป็นคำพื้นฐานของคำที่จัดอยู่ในรูปแบบของคำเฉพาะภาษาไทย และในตัวอย่างที่จะแสดงต่อไปนี้ จะแสดงตัวอย่างของคำภาษาไทยที่มีพยัญชนะ 2 ตัว ที่จัดอยู่ในรูปแบบเฉพาะของคำภาษาไทย เช่น

คำว่า “ เผละ ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ ผ-ล-สระเอะ ”
คำว่า “ แคระ ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ ค-ร-สระแอะ ”
คำว่า “ โพละ ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ พ-ล-สระโอะ ”
คำว่า “ เพราะ ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ พ-ร-สระเอาะ ”
คำว่า “ เปรอะ ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ ป-ร-สระเออะ ”
คำว่า “ เผลอ ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ ผ-ล-สระเออ ”
คำว่า “ เพลีย ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ พ-ล-สระเอีย ”
คำว่า “ เครือ ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ ค-ร-สระเอือ ”
คำว่า “ กลัว ” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “ ก-ล-สระอัว ”
คำว่า “เพลา” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “พ-ล-สระเอา”
คำว่า “เพลิน” เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ “พ-ล-สระเออ-น”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และยังมีคำภาษาไทยที่จัดอยู่ในรูปแบบเฉพาะของคำภาษาไทย ตามที่อัการเบรลล์ได้กำหนด โดยที่มีวรรณยุกต์เข้ามาผสมในรูปแบบเฉพาะของคำภาษาไทยด้วย เช่น

คำว่า "โต๊ะ" เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ "ต-สระโอะ-ไม้ตรี"

คำว่า "เขี่ย" เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ "ข-สระเอีย-ไม้เอก"

คำว่า "เบื่อ" เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ "บ-สระเออ-ไม้เอก"

คำว่า "พลั่ว" เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ "พ-ล-สระอัว-ไม้เอก"

คำว่า "เหล้า" เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ "ห-ล-สระเอา-ไม้โท"

คำว่า "เพิ่ม" เมื่อเป็นหนังสือเบรลล์จะมีลำดับการเขียนดังนี้ "พ-สระเออ-ไม้เอก-ม"

สำหรับการสร้างแบบตัวอักษรในวิทยานิพนธ์นี้ได้เลือกใช้โปรแกรม Fontographer ในการสร้างแบบตัวอักษรเบรลล์ ซึ่งโปรแกรมดังกล่าวเป็นโปรแกรมสำหรับใช้ออกแบบลักษณะตัวอักษร(Fonts) ที่ต้องการใช้งาน

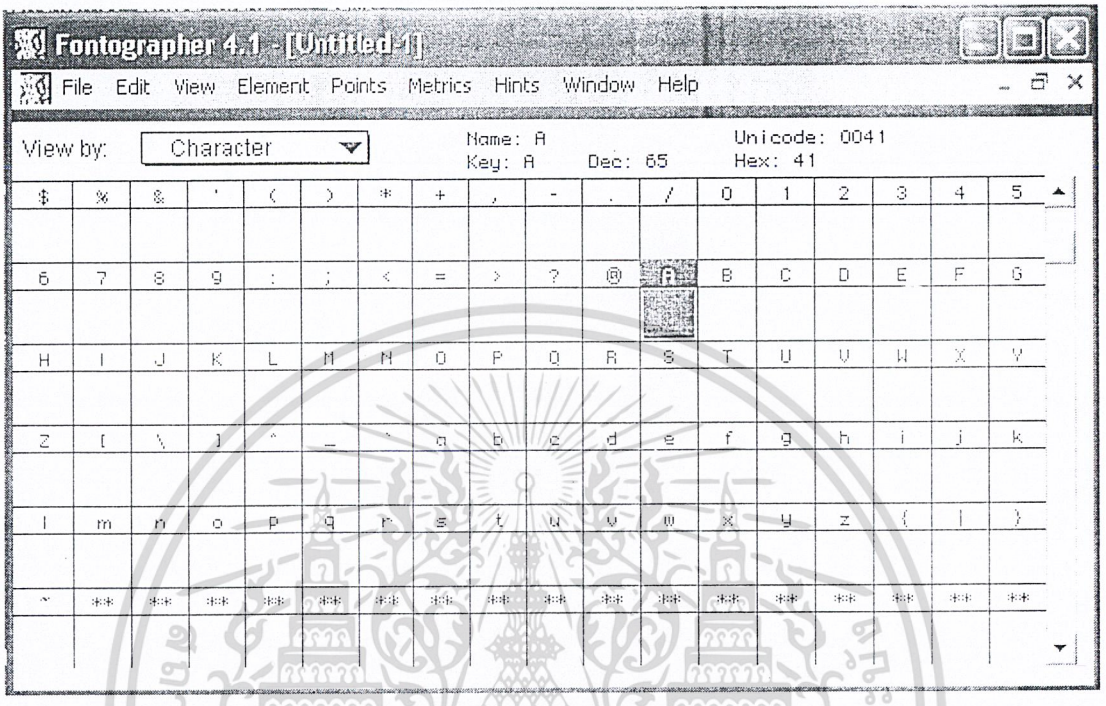
4.3 การใช้งานโปรแกรม Fontographer ในการสร้างแบบตัวอักษรเบรลล์

โปรแกรม Fontographer เป็น software ที่ใช้งานในด้านการออกแบบลักษณะตัวอักษร(Fonts) ซึ่งมีลักษณะการใช้งานง่ายและสะดวกเพียงแค่สามารถใช้เครื่องมือที่โปรแกรมมีไว้ให้เพื่อสร้างแบบตัวอักษรได้เท่านั้น โดยที่เมื่อจะทำการออกแบบตัวอักษรขึ้นใหม่ก็ให้เลือกที่เมนู File จากนั้นก็เลือกเมนู New โปรแกรมก็จะแสดงตารางรหัส ASCII เพื่อให้ผู้ใช้สร้างแบบตัวอักษรดังจะแสดงในรูปที่ 4.1 แล้วจากนั้นเมื่อต้องการออกแบบรหัส ASCII ตัวใดก็ให้คลิกไปที่ตารางรหัส ASCII นั้น โปรแกรมจะแสดงหน้าต่างที่ขยายขึ้นมาพร้อมกับแสดงแถบเครื่องมือช่วยในการออกแบบ โดยจะมีแถบเครื่องมือในการเขียนลายเส้นต่างๆ และแถบเครื่องมือ Layers ที่ใช้กำหนดขอบเขตของตัวอักษรที่จะออกแบบดังแสดงในรูปที่ 4.2 เมื่อออกแบบทั้งหมดเสร็จก็ทำการบันทึกโดยการเลือกเมนู File แล้วก็เลือกเมนู Save จากนั้นก็จะได้แฟ้มข้อมูลแบบตัวอักษรเป็นนามสกุล *.fog ดังแสดงในรูปที่ 4.3 ซึ่งแฟ้มข้อมูลแบบตัวอักษรที่ได้ไม่สามารถจะใช้งานบนระบบปฏิบัติการ Windows 98 ได้ ดังนั้น โปรแกรม Fontographer ก็มีเมนูให้เลือกที่สามารถแปลงข้อมูลที่ได้ออกแบบโดยเลือกเมนู File แล้วก็เลือกเมนู Generate Font Files จากนั้นก็เลือกข้อมูลที่จะแปลงเป็นข้อมูลแบบตัวอักษรที่เป็นแบบ True Type ซึ่งจะได้แฟ้มข้อมูลแบบตัวอักษรเป็นนามสกุล *.tiff เพื่อนำไปใช้งานต่อไป

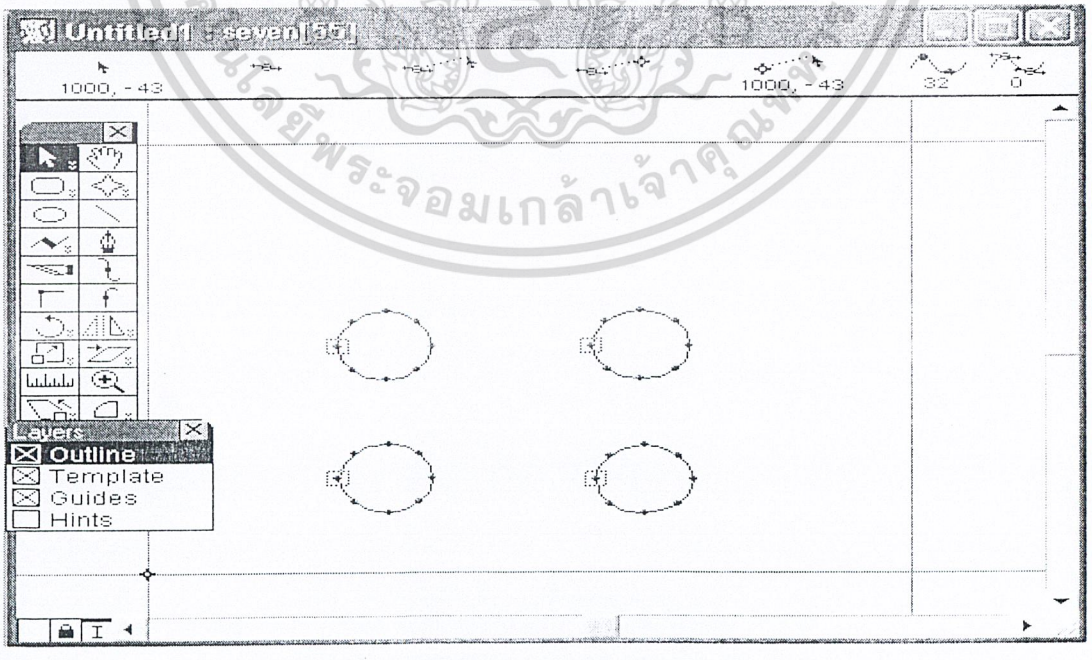
ข้อมูลแบบตัวอักษรที่เป็นแบบ True Type นั้นเป็นข้อมูลแบบตัวอักษรที่ใช้งานบนระบบปฏิบัติการ Windows 98 ซึ่งข้อมูลแบบตัวอักษรที่เราได้แปลงนี้จะนำมา Add ข้อมูลแบบตัวอักษรให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบปฏิบัติการ Windows 98 เพื่อให้ระบบได้รู้จักข้อมูลแบบตัวอักษรนี้และสามารถเรียกใช้ในการแสดงผลได้

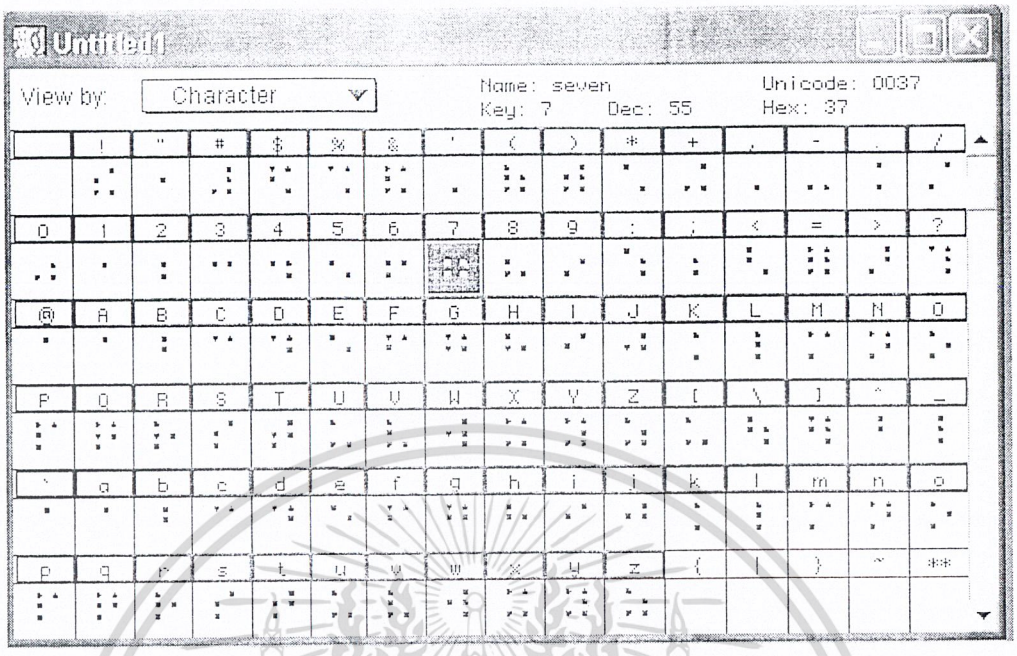


รูปที่4.1 แสดงการสร้างแบบตัวอักษรใหม่ของโปรแกรม Fontographer

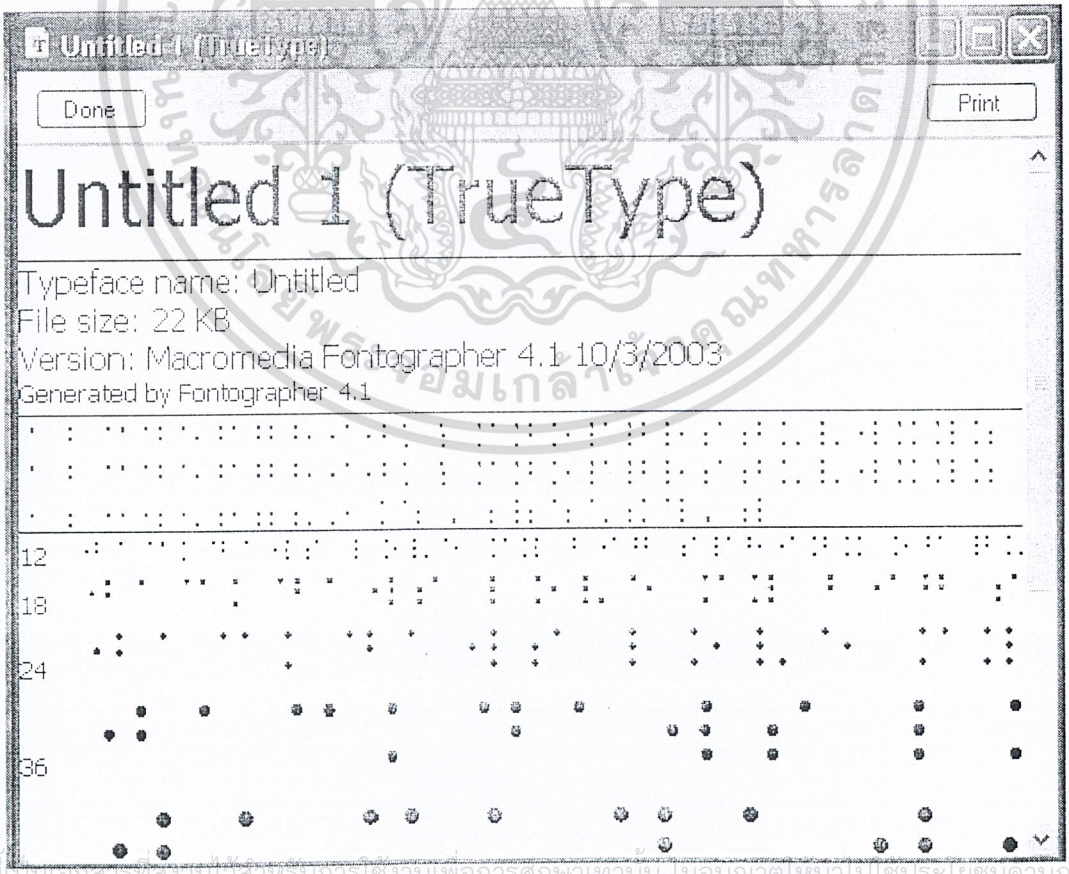


รูปที่4.2 แสดงการสร้างแบบตัวอักษรบนโปรแกรม Fontographer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงแบบอักษรที่สร้างเสร็จแล้วของโปรแกรม Fontographer



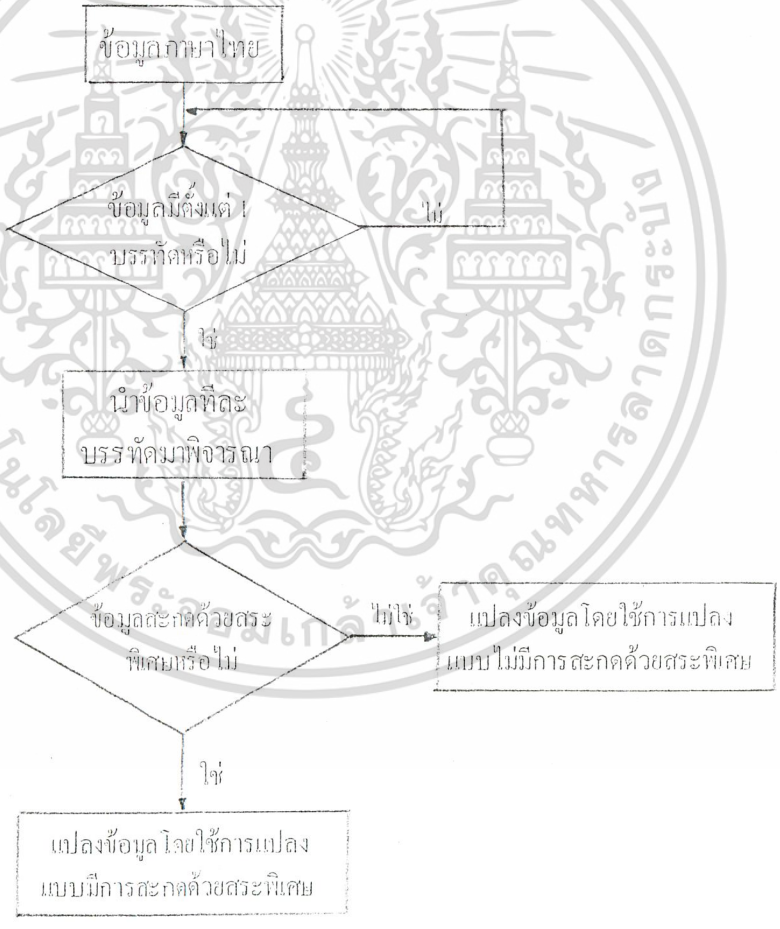
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.4 แสดงแบบตัวอักษรที่แปลงจากโปรแกรม Fontographer

4.4 หลักการในการแปลงข้อมูลภาษาไทยให้เป็นอักษรเบรลล์

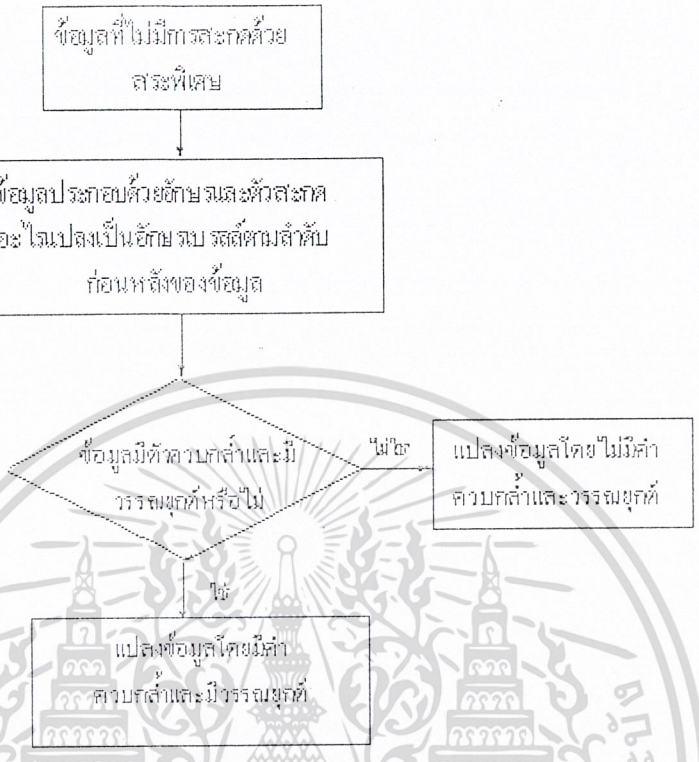
ในส่วนของการแปลงข้อมูลภาษาไทยให้เป็นอักษรเบรลล์สามารถแบ่งการทำงานออกเป็น 2 กรณี คือ

1. กรณีคำที่ไม่อยู่ในรูปแบบเฉพาะ
2. กรณีคำที่อยู่ในรูปแบบเฉพาะ ซึ่งสะกดอยู่ด้วยสระ ดังนี้คือ " สระเอะ, สระแอะ , สระโอะ, สระเอาะ, สระเออะ, สระเออ, สระเอีย, สระเอือ, สระอัว,สระเอา, สระเอิอ "

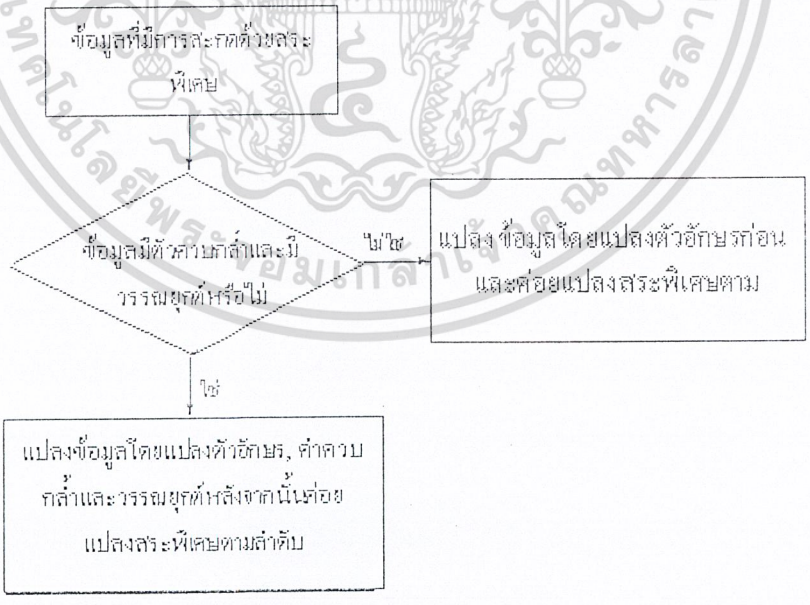


รูปที่ 4.5 Flowchart แสดงการทำงานหลักของการแปลงข้อมูลภาษาไทยเป็นอักษรเบรลล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4.6 Flowchart แสดงการทำงานของโปรแกรมในส่วนการแปลงข้อมูลแบบไม่ใช่สระพิเศษ



รูปที่ 4.7 Flowchart แสดงการทำงานของโปรแกรมในส่วนการแปลงข้อมูลแบบสระพิเศษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การทดลองการทำงานของโปรแกรมและผลการทดลอง

การทดลองการรู้จำตัวอักษร

ทำการทดลองโดยแบ่งผู้ทดลองออกเป็น 2 กลุ่ม กลุ่มแรกเป็นผู้ทดลองที่ไม่มีลายมือเขียนอยู่ในฐานข้อมูล กลุ่มที่สองเป็นผู้ทดลองที่มีลายมือเขียนอยู่ในฐานข้อมูล โดยทำการทดสอบกับฐานข้อมูลที่มีจำนวนชุดของข้อมูลที่ใช้อ้างอิงต่างๆกันไปจำนวนสามแบบคือ ข้อมูลหนึ่งชุด ข้อมูลสองชุด และข้อมูลสามชุด โดยผู้ทดลองชุดแรกมีจำนวน 5 คน ทำการเขียนตัวอักษร ก-ฮ แต่ละตัวอักษรคนละ 5 ครั้ง โดยผู้ทดลองชุดแรกแต่ละคนจะทำการเขียนตัวอักษรคนละช่วงที่ไม่ซ้ำกัน เช่น ผู้ทดลองคนแรกจะทดสอบเขียนตัวอักษร ก-ง ผู้ทดลองคนที่สองเขียน จ-ฐ แล้วสังเกตการรู้จำ ส่วนผู้ทดลองชุดที่สองทำการเขียน ก-ฮ ทั้ง 44 ตัวแล้วสังเกตการรู้จำ

ตารางที่ 1 ตารางแสดงผลการทดลองการรู้จำ

ตัวอักษรที่ทดลอง	ฐานข้อมูล 1 ชุด						ฐานข้อมูล 2 ชุด						ฐานข้อมูล 3 ชุด					
	ผู้ทดลอง1			ผู้ทดลอง2			ผู้ทดลอง1			ผู้ทดลอง2			ผู้ทดลอง1			ผู้ทดลอง2		
	ถูกต้อง	ผิด	ไม่รู้จัก	ถูกต้อง	ผิด	ไม่รู้จัก	ถูกต้อง	ผิด	ไม่รู้จัก	ถูกต้อง	ผิด	ไม่รู้จัก	ถูกต้อง	ผิด	ไม่รู้จัก	ถูกต้อง	ผิด	ไม่รู้จัก
ก	4	1	-	5	-	-	4	1	-	5	-	-	5	-	-	5	-	-
ข	5	-	-	3	1	1	3	2	-	4	1	-	4	1	-	5	-	-
ฃ	3	1	1	5	-	-	5	-	-	4	1	-	5	-	-	5	-	-
ค	3	2	-	5	-	-	4	1	-	5	-	-	5	-	-	5	-	-
ฅ	3	2	-	5	-	-	5	-	-	5	-	-	5	-	-	5	-	-
ฆ	5	-	-	5	-	-	5	-	-	5	-	-	2	3	-	4	1	-
ง	3	2	-	4	1	-	3	2	-	4	1	-	4	1	-	5	-	-
จ	5	-	-	1	4	-	4	1	-	5	-	-	2	3	-	5	-	-
ฉ	4	1	-	2	3	-	3	2	-	5	-	-	5	-	-	5	-	-
ช	5	-	-	3	2	-	5	-	-	5	-	-	5	-	-	5	-	-
ซ	4	-	1	5	-	-	4	1	-	4	1	-	3	2	-	3	2	-
ฌ	-	5	-	5	-	-	4	1	-	5	-	-	3	2	-	5	-	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษรที่ทดลอง	ฐานข้อมูล 1 ชุด						ฐานข้อมูล 2 ชุด						ฐานข้อมูล 3 ชุด					
	ผู้ทดลอง1			ผู้ทดลอง2			ผู้ทดลอง1			ผู้ทดลอง2			ผู้ทดลอง1			ผู้ทดลอง2		
	ถูก	ผิด	ไม่ออก	ถูก	ผิด	ไม่ออก	ถูก	ผิด	ไม่ออก	ถูก	ผิด	ไม่ออก	ถูก	ผิด	ไม่ออก	ถูก	ผิด	ไม่ออก
ญ	-	5	-	5	-	-	2	3	-	4	1	-	4	1	-	5	-	-
ฎ	4	1	-	5	-	-	-	5	-	5	-	-	2	3	-	4	1	-
ฏ	5	-	-	4	1	-	5	-	-	5	-	-	5	-	-	5	-	-
ฐ	2	-	3	5	-	-	2	-	3	5	-	-	5	-	-	5	-	-
ฑ	4	1	-	5	-	-	5	-	-	5	-	-	5	-	-	5	-	-
ฒ	5	-	-	5	-	-	4	1	-	5	-	-	5	-	-	5	-	-
ณ	4	1	-	4	-	1	2	2	1	3	2	-	4	1	-	4	1	-
ด	5	-	-	3	2	-	2	3	-	5	-	-	3	2	-	4	1	-
ต	1	-	4	3	2	-	-	5	-	5	-	-	-	5	-	5	-	-
ถ	4	1	-	3	2	-	4	1	-	4	1	-	3	2	-	4	1	-
ท	3	2	-	2	3	-	4	1	-	4	1	-	3	-	2	5	-	-
ธ	-	4	1	3	2	-	4	1	-	5	-	-	2	3	-	5	-	-
น	4	-	1	5	-	-	3	2	-	5	-	-	5	-	-	5	-	-
บ	3	-	2	2	3	-	1	4	-	4	1	-	5	-	-	5	-	-
ป	4	1	-	2	3	-	2	3	-	4	1	-	5	-	-	4	1	-
ผ	-	2	3	5	-	-	1	3	-	5	-	-	-	5	-	4	1	-
ฝ	2	3	-	4	1	-	5	-	-	5	-	-	5	-	-	4	1	-
พ	1	2	2	4	1	-	-	3	2	5	-	-	-	5	-	5	-	-
ฟ	1	4	-	5	-	-	-	5	-	5	-	-	-	5	-	4	1	-
ภ	5	-	-	1	2	2	5	-	-	3	2	-	5	-	-	3	2	-
ม	4	1	-	3	2	-	4	1	-	5	-	-	3	1	1	5	-	-
ย	-	4	1	5	-	-	4	1	-	4	1	-	2	2	1	5	-	-
ร	5	-	-	5	-	-	2	3	-	5	-	-	1	4	-	4	1	-
ล	4	1	-	3	2	-	5	-	-	3	2	-	3	2	-	4	1	-
ว	2	2	1	5	-	-	2	3	-	5	-	-	3	2	-	5	-	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษรที่ทดลอง	ฐานข้อมูล 1 ชุด						ฐานข้อมูล 2 ชุด						ฐานข้อมูล 3 ชุด					
	ผู้ทดลอง1			ผู้ทดลอง2			ผู้ทดลอง1			ผู้ทดลอง2			ผู้ทดลอง1			ผู้ทดลอง2		
	ถูกก	ผิดด	ไม่ระบุ	ถูกก	ผิดด	ไม่ระบุ	ถูกก	ผิดด	ไม่ระบุ	ถูกก	ผิดด	ไม่ระบุ	ถูกก	ผิดด	ไม่ระบุ	ถูกก	ผิดด	ไม่ระบุ
ศ	4	1	-	5	-	-	5	-	-	5	-	-	4	1	-	5	-	-
ษ	3	-	2	5	-	-	5	-	-	5	-	-	5	-	-	5	-	-
ส	5	-	-	5	-	-	5	-	-	5	-	-	5	-	-	4	1	-
ห	5	-	-	4	-	1	5	-	-	5	-	-	5	-	-	4	1	-
ฬ	4	-	1	5	-	-	4	-	1	5	-	-	5	-	-	4	1	-
อ	5	-	-	4	1	-	4	1	-	3	2	-	4	1	-	4	1	-
ฮ	5	-	-	5	-	-	3	2	-	5	-	-	4	1	-	4	1	-

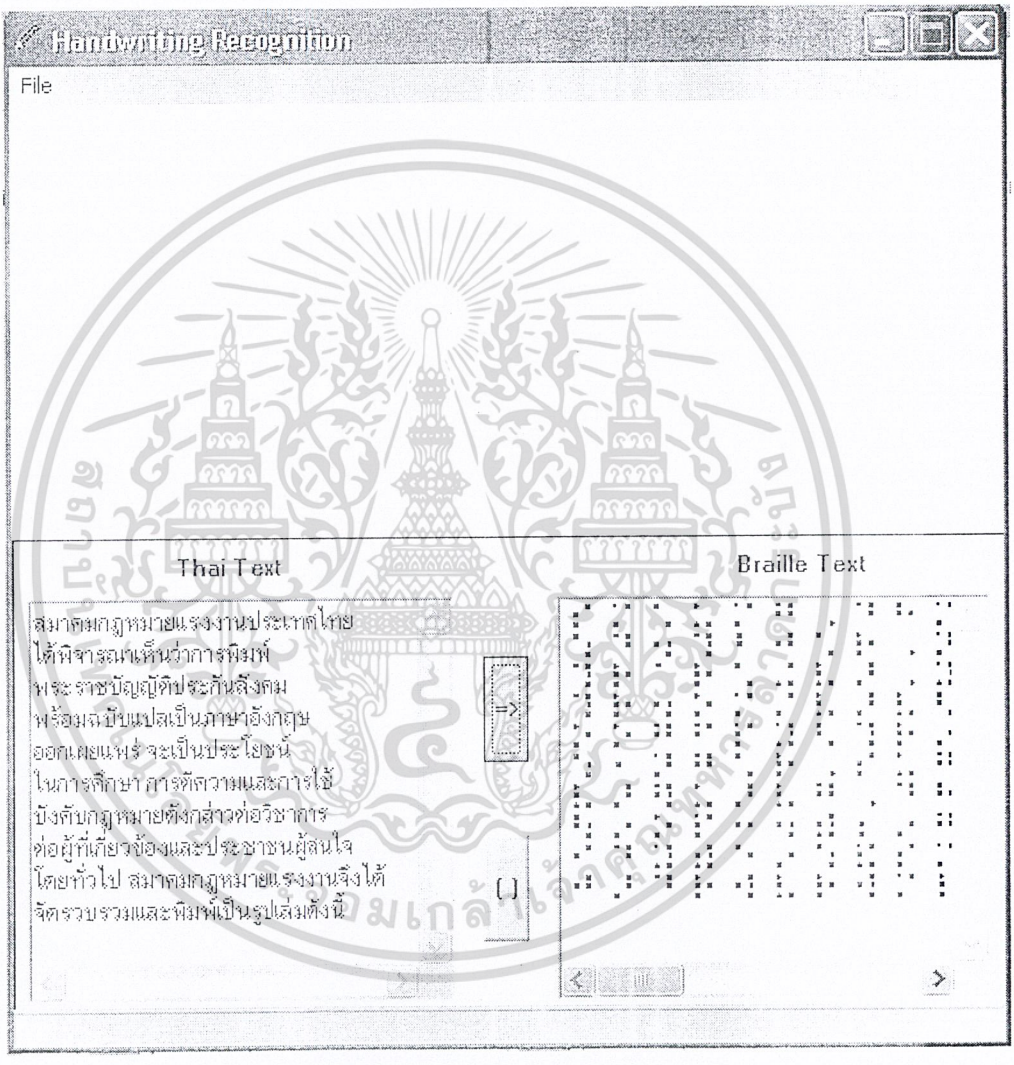
จากผลการทดลองจะได้ว่าความถูกต้องของตัวอักษรที่ผู้ทดลองในชุดแรก (ไม่มีลายมือเขียนอยู่ในฐานข้อมูล) เขียนจะอยู่ระหว่าง 66.82%-71.81% ส่วนความถูกต้องของผู้ทดลองชุดที่สอง (มีลายมือเขียนอยู่ในฐานข้อมูล) เขียนจะอยู่ระหว่าง 81.81%- 91.82% ดังนั้นหากต้องการเพิ่มความถูกต้องให้กับการเขียนตัวอักษรนั้น สามารถทำได้โดยเก็บข้อมูลลายมือเขียนของผู้ใช้ลงไปในฐานข้อมูลของโปรแกรมก่อน นอกจากนี้จะเห็นได้ว่าเมื่อเพิ่มจำนวนชุดข้อมูลในฐานข้อมูลให้เพิ่มมากขึ้น จะทำให้ความถูกต้องของผลการทดลองเพิ่มมากขึ้นทั้งกับผู้ที่มีลายมือเขียนอยู่ในฐานข้อมูลและผู้ที่ไม่มีลายมือเขียนอยู่ในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองแปลงข้อมูลจากภาษาไทยเป็นอักษรเบรลล์

การทดลองที่ 1 การทดลองการแปลงข้อมูลจากภาษาไทยเป็นอักษรเบรลล์

ทำการเขียนข้อมูลภาษาไทยดังตัวอย่างข้างล่าง และทำการแปลงเป็นข้อมูลอักษรเบรลล์ โดยทำการกดปุ่มลูกศร



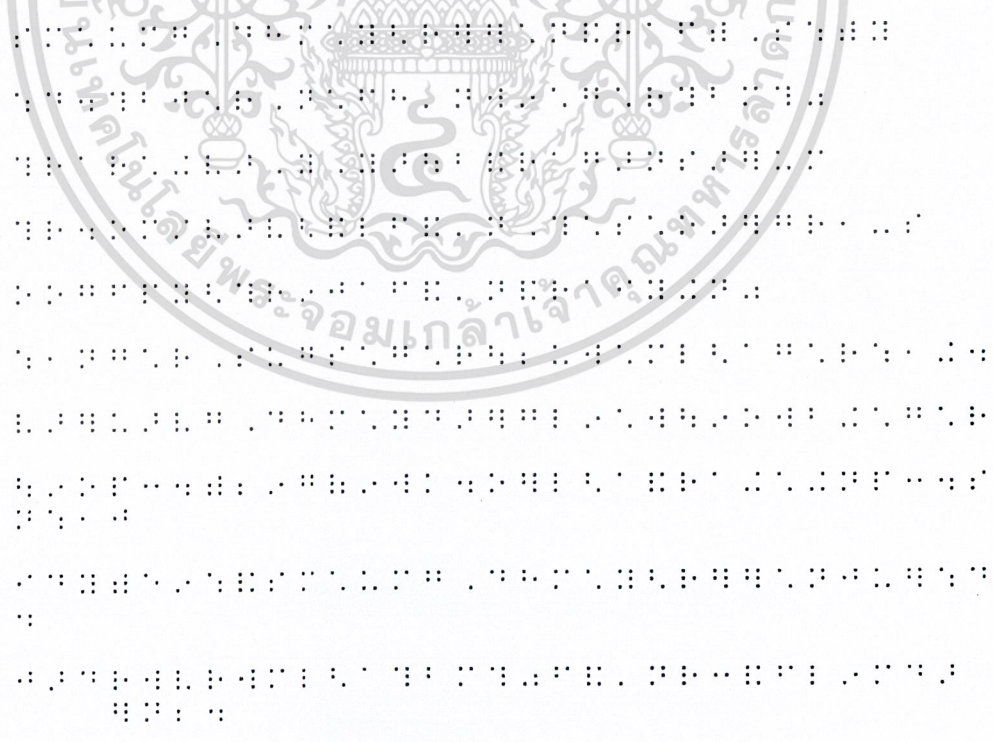
รูปที่ 5.1 แสดงการทดลองที่ 1 แปลงข้อมูลภาษาไทยเป็นข้อมูลอักษรเบรลล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อความที่ต้องการแปลง

“ สมาคมกฎหมายแรงงานประเทศไทย
ได้พิจารณาเห็นว่าการพิมพ์
พระราชบัญญัติประกันสังคม
พร้อมฉบับแปลเป็นภาษาอังกฤษ
ออกเผยแพร่ จะเป็นประโยชน์
ในการศึกษา การตีความและการใช้
บังคับกฎหมายดังกล่าวต่อวิชาการ
ต่อผู้ที่เกี่ยวข้องและประชาชนผู้สนใจ
โดยทั่วไป สมาคมกฎหมายแรงงานจึงได้
จัดรวบรวมและพิมพ์เป็นรูปเล่มดังนี้ ”

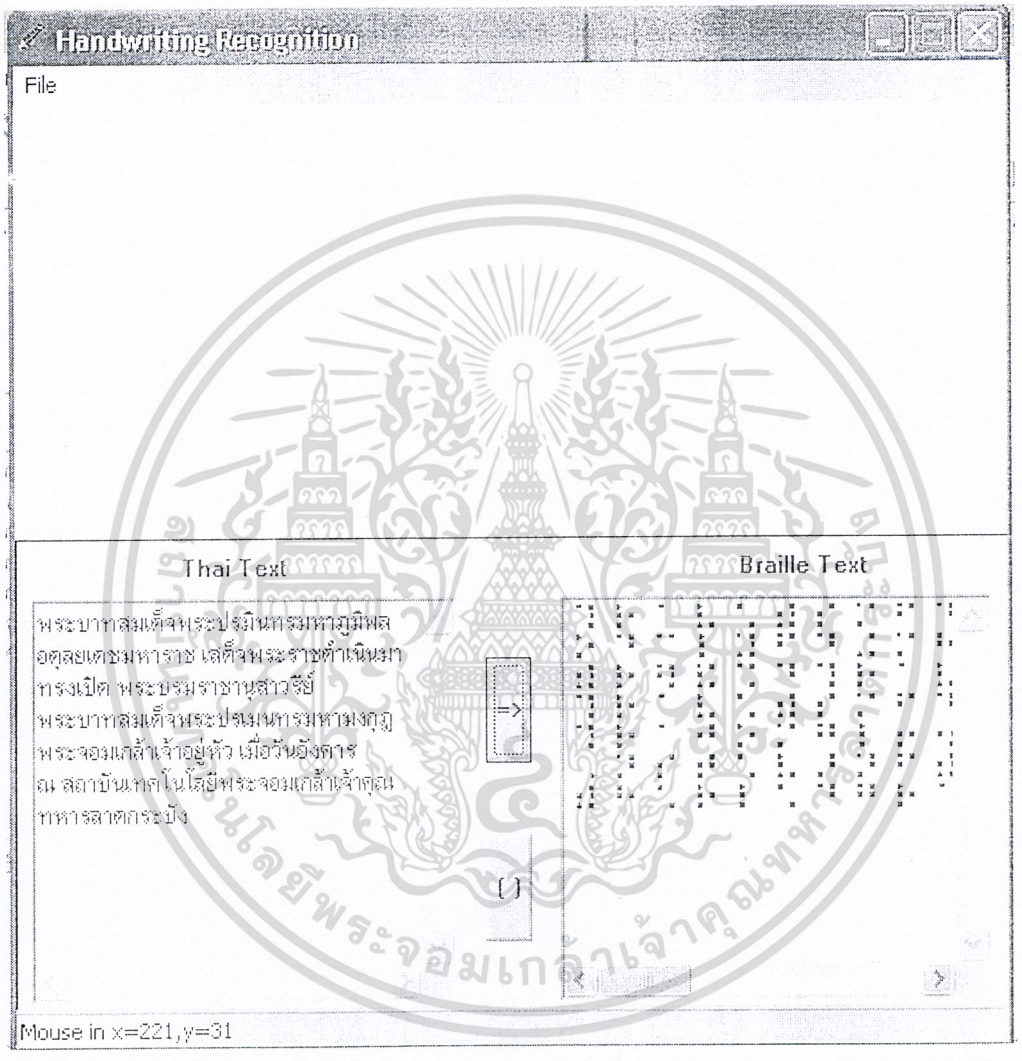
ข้อมูลอักษรเบรลล์ที่แปลงได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2 การทดลองการแปลงข้อมูลจากภาษาไทยเป็นอักษรเบรลล์

ทำการเขียนข้อมูลภาษาไทยดังตัวอย่างข้างล่าง และทำการแปลงเป็นข้อมูลอักษรเบรลล์ โดยทำการกดปุ่มลูกศร
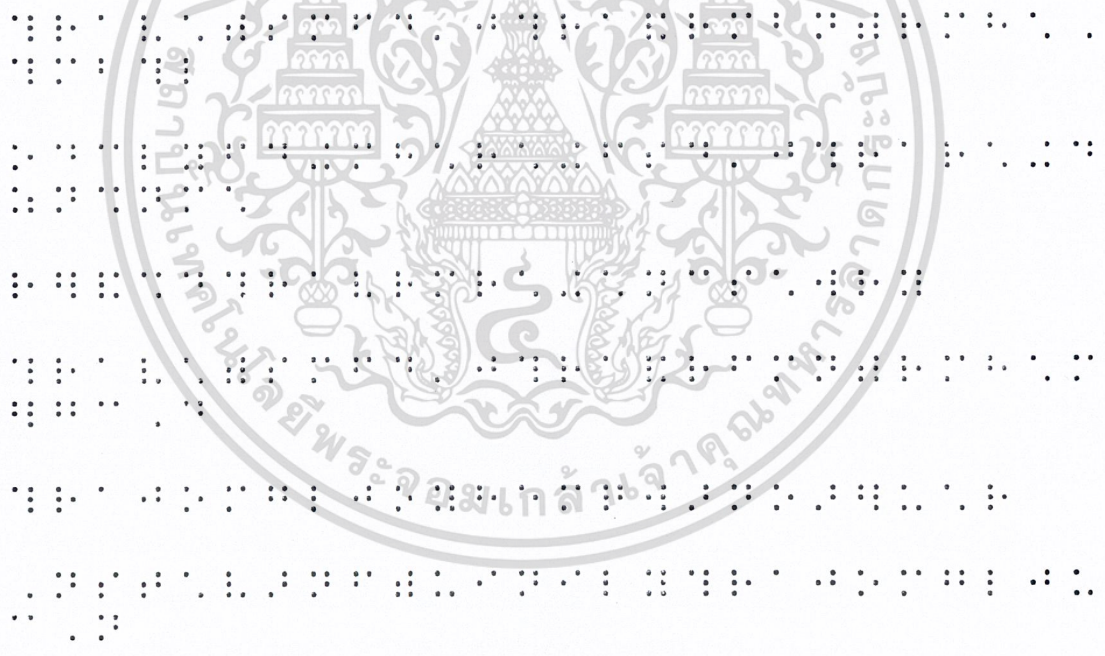


รูปที่ 5.2 แสดงการทดลองที่ 1 แปลงข้อมูลภาษาไทยเป็นข้อมูลอักษรเบรลล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อความภาษาไทย

พระบาทสมเด็จพระปรมินทรมหาภูมิพล
อดุลยเดชมหาราช เสด็จพระราชดำเนินมา
ทรงเปิด พระบรมราชานุสาวรีย์
พระบาทสมเด็จพระปรเมนทรมหามงกุฎ
พระจอมเกล้าเจ้าอยู่หัว เมื่อวันอังคาร
ณ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ
ทหารลาดกระบัง

ข้อมูลอักษรเบรลล์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
วิจารณ์และข้อเสนอแนะ

เนื่องจากในโครงงานนี้มีอินพุทเป็นลายมือเขียนซึ่งจะแตกต่างกันไปในแต่ละบุคคลทั้งลักษณะของการเขียนและรูปแบบตัวอักษรตลอดจนความคล้ายคลึงกันของอักษรแต่ละตัวเป็นเหตุให้เกิดความผิดพลาดในการรู้จำ ดังนั้นการเตรียมข้อมูลก่อนการประมวลผลเช่นการปรับขนาดตัวอักษร,การหาตำแหน่งที่เสถียรของตัวอักษร,การใช้ข้อมูลฝึกสอนแบบลายมือเขียนและข้อมูลกราฟโครงร่าง เพื่อช่วยกำจัดความแตกต่างของลายมือจึงเป็นสิ่งสำคัญ ซึ่งจากการทดลองผลการรู้จำที่ได้ก็มีความถูกต้องในระดับหนึ่ง

ความผิดพลาดที่เกิดขึ้นในการรู้จำเกิดจากสาเหตุหลายประการได้แก่ฐานข้อมูลที่ใช้ยังไม่ครอบคลุมรูปแบบการเขียนทั้งหมดซึ่งอาจแก้ไขโดยการเพิ่มฐานข้อมูลให้มากขึ้นแต่ก็ไม่ควรจะมากเกินไปเพราะจะส่งผลต่อการทำงานของโปรแกรมทั้งความถูกต้องและความเร็ว นอกจากนี้ปัญหายังเกิดจากการที่ไม่สามารถกำจัดความแตกต่างของลายมือออกไปได้อย่างเพียงพอและตำแหน่งในการจับภาพข้อมูลของ Neural ยังไม่เหมาะสมที่สุดซึ่งอาจแก้ไขโดยเปลี่ยนแปลงกระบวนการในการวิเคราะห์ตัวอักษรหรือหาวิธีการอื่นมาช่วยเสริมการวิเคราะห์ให้ดีขึ้นและเพื่อเพิ่มความถูกต้องในการใช้งานโปรแกรมควรนำลายมือเขียนของผู้ที่ทดสอบมาใช้เป็นฐานข้อมูลอ้างอิงก็จะช่วยให้ความถูกต้องเพิ่มขึ้นได้เช่นเดียวกัน

จากผลการทดลองที่ได้สามารถนำมาวิเคราะห์ความถูกต้องได้ในระดับหนึ่งเนื่องจากลายมือของผู้ทดสอบแต่ละคนในขณะทดสอบจะขึ้นกับอารมณ์และความตั้งใจในการเขียนในขณะทดสอบด้วย

ในส่วนที่สองเป็นการนำข้อมูลที่ได้จากการรู้จำไปแปลงเป็นอักษรเบรลล์ซึ่งก็ได้ผลเป็นที่น่าพอใจแต่ก็ยังมีปัญหาอยู่บ้างในคำบางคำเช่น คำว่า “เพ-ลา” , “เพลา” โปรแกรมจะมองเป็นคำว่าเพลา และแปลงตามหลักอักษรเบรลล์ ซึ่งถ้าเราต้องการคำว่า เพ-ลา ก็จะต้องใส่เครื่องหมาย “-” คั่นในตัวอักษรเพื่อไม่ให้โปรแกรมมองเป็นการสะกดแบบพิเศษที่มีการควบกล้ำเป็นต้น นอกจากนี้ยังอาจเพิ่มประสิทธิภาพของโปรแกรมได้โดยทำการเพิ่มโปรแกรมในส่วนของการแปลงข้อมูลตัวเลขและข้อมูลภาษาอังกฤษเข้าไปด้วยก็จะทำให้โปรแกรมมีความสมบูรณ์ยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ตารางแสดงพยัญชนะ , สระ , วรรณยุกต์ ที่ใช้ในโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดง พยัญชนะ , สระ , วรรณยุกต์ที่ใช้ในโครงงานนี้

row \ column	1	2	3	4	5	6	7	8	9	10
1	ก	ข	ฃ	ค	ฅ	ฆ	ง	จ	ฉ	ช
2	ซ	ฌ	ญ	ฎ	ฏ	ฐ	ฑ	ฒ	ณ	ด
3	ต	ถ	ท	ธ	น	บ	ป	ผ	ฝ	พ
4	ฟ	ภ	ม	ย	ร	ล	ว	ศ	ษ	ส
5	ห	ฬ	อ	ฮ	อะ	อา	อิ	อี	อึ	อื
6	โอ	อำ	ไอ	ใอ	เอ	แอ	อุ	อู	ฯ	ๆ
7	ฤ	ฦ	เอก	โท	ตรี	จัตวา	การันต์	ไม้หันอากาศ	ไม้ไต่คู้	D
8	E	S								

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โปรแกรมที่ใช้ฝึกสอน Neuron

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Menus, ExtCtrls, ComCtrls;

type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    New1: TMenuItem;
    N1: TMenuItem;
    Close1: TMenuItem;
    Timer1: TTimer;
    StatusBar1: TStatusBar;
    Edit1: TMenuItem;
    KeepPattern1: TMenuItem;
    Findoptimize1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    FindBinary1: TMenuItem;
    N4: TMenuItem;
    ClearFile1: TMenuItem;
    procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure FormMouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure FormMouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure Close1Click(Sender: TObject);
    procedure New1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure KeepPattern1Click(Sender: TObject);
    procedure Findoptimize1Click(Sender: TObject);
    procedure FindBinary1Click(Sender: TObject);
    procedure ClearFile1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

const
  leg=10;
  totalleg=(100*leg);
  stb=200;
  sta=200;

var
  Form1: TForm1;
  fdragging : boolean;




  FRect :TRect;
  picture:file;
  conx:file;
  cony:file;
  allpicture:file;
  optimize:file of byte;///**
  c,d,i,j,m,last:integer;
  ax,ay,aax,aaxt,aay,aayt,tempx,tempy:array[1..totalleg]of byte;   ///**
  array_nx,array_sx:array[1..2500] of integer;
  array_cx,array_tx,array_x,array_y,array_xx,array_yy :array[1..5000] of integer;
  array_xy,optimize_xy,patternpic:array[1..40,1..40] of integer;
  array_xt,array_yt,optimize_conx,optimize_cony,patterncon:array[1..40,1..100] of integer;
  bivalue:array of array of array of integer; ///***

implementation

{$R *.dfm}

function find_min(xyarray:array of integer;last:integer):integer;
var index:integer;
begin
        result:=600;
        for index:=1 to last do
                if (xyarray[index]<result) and (xyarray[index]<>600) then
                        result:=xyarray[index];
end;

function find_max(xyarray:array of integer;last:integer):integer;
var index:integer;
begin
        result:=0;
        for index:=1 to last do
                if (xyarray[index]>result) and (xyarray[index]<>600) then
                        result:=xyarray[index];
end;
function scaling(sxy,nwidth,owidth:integer):integer;
var pans:real;
begin
        if (sxy<>600) and (owidth<>0) then
        begin
                pans:=sxy*(nwidth/owidth);
                result:=round(pans);
        end
        else if (sxy<>600) and (owidth=0) then
                result:=sxy
             else
                        result:=600;
end;

procedure TForm1.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        begin
                canvas.LineTo(x,y);
                if (c<>x) or (d<>y) then
                begin
                        Array_x[i]:=x;
                        Array_y[i]:=y;
                        i:=i+1;
                end;
        end;
end;

procedure TForm1.FormMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
        if button = mbleft then
        begin
                fdragging := true;
                canvas.MoveTo(x,y);
                c:=x;
                d:=y;
                timer1.Enabled := false;
        end;
end;

procedure TForm1.FormMouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
        fdragging := false;
        array_x[i]:=600;
        array_y[i]:=600;
        i:=i+1;
        timer1.Enabled := true;
end;

procedure TForm1.Close1Click(Sender: TObject);
begin
        form1.close;
end;

procedure TForm1.New1Click(Sender: TObject);
var a,b:integer;
begin
        for a:=1 to 40 do
                for b:=1 to 40 do
                        array_xy[a,b]:=0;
        for a:=1 to 40 do
                for b:=1 to 100 do
                begin
                        array_xt[a,b]:=0;
                        array_yt[a,b]:=0;
                end;

        for a:=1 to 5000 do
        begin




                Array_x[a]:=0;
                Array_y[a]:=0;
                Array_xx[a]:=0;
                Array_yy[a]:=0;
                array_sx[a]:=0;
                array_nx[a]:=0;
                array_cx[a]:=0;
        end;
        i:=1;
        j:=2;
        m:=1;
        invalidate;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
        i:=1;
        j:=2;
        m:=1;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
var h,k,t,u,v,z,max,xmin,ymin,xmax,ymax,alpha,angle:integer;
begin
                //////disable timer and move coordinate to origin
                //////for scaling
        timer1.Enabled:= false;
        xmin:=find_min(array_x,i-2);
        ymin:=find_min(array_y,i-2);
        for u:=1 to i-1 do
        begin
                if (array_x[u]<>600) and (array_y[u]<>600) then
                begin
                        array_x[u]:=array_x[u]-xmin;
                        array_y[u]:=array_y[u]-ymin;
                end;
        end;
                        /////find min max again for scaling
        xmin:=find_min(array_x,i-2);
        ymin:=find_min(array_y,i-2);
        xmax:=find_max(array_x,i-2);
        ymax:=find_max(array_y,i-2);
        if (xmax-xmin)>(ymax-ymin) then
                max:=xmax-xmin
        else
                max:=ymax-ymin;
                 ///////scaling to 50
        for u:=1 to i do
        begin
                array_x[u]:= scaling(array_x[u],50,max);
                array_y[u]:= scaling(array_y[u],50,max);
        end;
```

ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        for v:=1 to i do
        begin
                if (array_x[u]=array_x[v]) and
                (array_y[u]=array_y[v])
                and (u<>v) then
                begin
                        array_x[v]:=100;
                        array_y[v]:=100;
                end;
        end;
           //////pixels scaning in case of large letter to small
v:=1;
j:=2;
m:=1;
for u:=1 to 5000 do
        if (array_x[u]<>100) and (array_y[u]<>100) then
        begin
                array_x[v]:=array_x[u];
                array_y[v]:=array_y[u];
                inc(v);
        end;
        //////extend line in case of small letter to large
array_xx[1] := array_x[1];
array_yy[1] := array_y[1];
repeat
        if (array_x[m+1]=600) and (array_y[m+1]=600) then
                m:=m+2;
        h:=(array_x[m+1]-array_x[m]);
        k:=(array_y[m+1]-array_y[m]);
        if (h=0) and (k<0) then
        begin
                array_xx[j]:=array_x[m];
                array_yy[j]:=(array_y[m]-1);
        end;
        if (h>0) and (k<0) then
        begin
                array_xx[j]:=(array_x[m]+1);
                array_yy[j]:=(array_y[m]-1);
        end;
        if (h>0) and (k=0) then
        begin
                array_xx[j]:=(array_x[m]+1);
                array_yy[j]:=array_y[m];
        end;
        if (h>0) and (k>0) then
        begin
                array_xx[j]:=(array_x[m]+1);
                array_yy[j]:=(array_y[m]+1);
        end;
        if (h=0) and (k>0) then
        begin
                array_xx[j]:=array_x[m];
                array_yy[j]:=(array_y[m]+1);
        end;



        if (h<0) and (k>0) then
        begin
                array_xx[j]:=(array_x[m]-1);
                array_yy[j]:=(array_y[m]+1);
        end;
        if (h<0) and (k=0) then
        begin
                array_xx[j]:=(array_x[m]-1);
                array_yy[j]:=array_y[m];
        end;
        if (h<0) and (k<0) then
        begin
                array_xx[j]:=(array_x[m]-1);
                array_yy[j]:=(array_y[m]-1);
        end;
                repeat
                h:=(array_x[m+1]-array_xx[j]);
                k:=(array_y[m+1]-array_yy[j]);
                if (h=0) and (k<0) then
                begin
                        inc(j);
                        array_xx[j]:=array_xx[j-1];
                        array_yy[j]:=(array_yy[j-1]-1);
                end;
                if (h>0) and (k<0) then
                begin
                        inc(j);
                        array_xx[j]:=(array_xx[j-1]+1);
                        array_yy[j]:=(array_yy[j-1]-1);
                end;
                if (h>0) and (k=0) then
                begin
                        inc(j);
                        array_xx[j]:=(array_xx[j-1]+1);
                        array_yy[j]:=array_yy[j-1];
                end;
                if (h>0) and (k>0) then
                begin
                        inc(j);
                        array_xx[j]:=(array_xx[j-1]+1);
                        array_yy[j]:=(array_yy[j-1]+1);
                end;
                if (h=0) and (k>0) then
                begin
                        inc(j);
                        array_xx[j]:=array_xx[j-1];
                        array_yy[j]:=(array_yy[j-1]+1);
                end;
                if (h<0) and (k>0) then
                begin
                        inc(j);
                        array_xx[j]:=(array_xx[j-1]-1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if (h<0) and (k=0) then
                begin
                        inc(j);
                        array_xx[j]:=(array_xx[j-1]-1);
                        array_yy[j]:=array_yy[j-1];
                end;
                if (h<0) and (k<0) then
                begin
                        inc(j);
                        array_xx[j]:=(array_xx[j-1]-1);
                        array_yy[j]:=(array_yy[j-1]-1);
                end;
                until (array_x[m+1]=array_xx[j]) and (array_y[m+1]=
                array_yy[j]);
inc(j);
inc(m);
until (array_x[m+2]=array_x[m+2])and(array_y[m+3]=array_y[m+3]);
        ////find stable angle in 45-135 degree
alpha:=45;
v:=1;
repeat
for u:=1 to j do
        array_x[u]:=round(array_xx[u]-((array_yy[u]-array_yy[1])/
        (sin((alpha*pi)/180)/cos((alpha*pi)/180))));
h:=2;
repeat
        for k:=h to j+1 do
                if array_x[h-1]=array_x[k] then
                        inc(array_nx[h-1]);
        inc(h);
until h=j+1;
array_sx[v]:=find_max(array_nx,j);
inc(v);
alpha:=alpha+5;
for h:=1 to j do
        array_nx[h]:=0;
until alpha=140;
angle:=find_max(array_sx,18);
        ////keep stable coordinate
h:=1;
repeat
        if array_sx[h]<>angle then
                h:=h+1;
until array_sx[h]=angle;
angle:=45+((h-1)*5);
for u:=1 to j do
begin
        array_x[u]:=round(array_xx[u]-((array_yy[u]-array_yy[1])/
        (sin((angle*pi)/180)/cos((angle*pi)/180))));
        array_y[u]:=round(array_yy[u]+((array_xx[u]-array_xx[1])/
        (sin((angle*pi)/180)/cos((angle*pi)/180))));
end;
array_xx:=array_x;
array_yy:=array_y;



xmin:=find_min(array_xx,j-2);
ymin:=find_min(array_yy,j-2);
        ////move coordinate of x to origin again after rotate
        ////and move coordinate to center for thickening
for u:=1 to j-1 do
begin
        array_xx[u]:=array_xx[u]-xmin;
        array_xx[u]:=array_xx[u]+5;
        array_yy[u]:=array_yy[u]-ymin;
        array_yy[u]:=array_yy[u]+5;
end;
  /////scaling from 56 to 40 pixels befor thickening
for u:=1 to j-1 do
begin
        array_xx[u]:= scaling(array_xx[u],40,60);
        array_yy[u]:= scaling(array_yy[u],40,60);
end;
        ////thickening
for u:=1 to j-1 do
begin
        array_xy[array_xx[u],array_yy[u]]:=1;
        for v:=-2 to 2 do
        begin
                array_xy[array_xx[u]+v,array_yy[u]-1]:=1;
                array_xy[array_xx[u]+v+1,array_yy[u]-1]:=1;
                array_xy[array_xx[u]+v+1,array_yy[u]]:=1;
                array_xy[array_xx[u]+v+1,array_yy[u]+1]:=1;
                array_xy[array_xx[u]+v,array_yy[u]+1]:=1;
                array_xy[array_xx[u]+v-1,array_yy[u]+1]:=1;
                array_xy[array_xx[u]+v-1,array_yy[u]]:=1;
                array_xy[array_xx[u]+v-1,array_yy[u]-1]:=1;
        end;
end;
     //////////scale contour to 40*100 pixels
for u:=1 to j-1 do
        array_cx[u]:=u;
for u:=1 to 5000 do
begin
        {array_xx[u]:= scaling(array_xx[u],40,60);
        array_yy[u]:= scaling(array_yy[u],40,60);}
        array_cx[u]:= scaling(array_cx[u],100,j-1);
        array_x[u]:=0;
        array_y[u]:=0;
end;
        ////find duplicate coordinate of countour
for h:=1 to 5000 do
        for k:=1 to 5000 do
        begin
                if (array_cx[h]=array_cx[k]) and (h<>k) then
                begin
                        array_xx[k]:=200;
                        array_yy[k]:=200;
```

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                end;
                        ////pixels scaning
        t:=j;
        j:=1;
        for u:=1 to t-1 do
                if (array_xx[u]<>200) and (array_cx[u]<>200)
                and (array_yy[u]<>200) then
                begin
                        array_x[j]:=array_xx[u];
                        array_y[j]:=array_yy[u];
                        {array_tx[j]:=array_cx[u];}
                        inc(j);
                end;
                ///////transform form one dimention array to two dimention array
        for t:=1 to 100 do
        begin
                for h:=array_x[t] to 40 do
                        array_xt[h,t]:=1;
                for k:=array_y[t] to 40 do
                        array_yt[k,t]:=1;
        end;
              ///////display handwriting on from
        for u:=1 to 40 do
                for t:=1 to 40 do
                        if array_xy[u,t]=1 then
                                form1.Canvas.Pixels[u,t]:= clBlue;
             ///////display countour of x and y on from
        for t:=1 to 100 do
        begin
                for h:=2 to 41 do
                        if array_xt[h,t]=1 then
                                form1.Canvas.Pixels[t,h+100]:= clBlue;
                for k:=1 to 40 do
                        if array_yt[k,t]=1 then
                                form1.Canvas.Pixels[t,k+200]:= clBlue;
        end;
end;
procedure TForm1.KeepPattern1Click(Sender: TObject);
var a,b:integer;
begin
                ////open file & save data into file
        assignfile(picture,'picture_xy.unt');
        reset(picture,4);
        seek(picture,filesize(picture));
        blockwrite(picture,array_xy,1600);
        closefile(picture);
        assignfile(conx,'contour_x.unt');
        reset(conx,4);
        seek(conx,filesize(conx));
        blockwrite(conx,array_xt,4000);
        closefile(conx);
        assignfile(cony,'contour_y.unt');
        reset(cony,4);
        seek(cony,filesize(cony));




        blockwrite(cony,array_yt,4000);
        closefile(cony);
          ////clear array for new data after keep pattern into file
        for a:=1 to 40 do
                for b:=1 to 40 do
                        array_xy[a,b]:=0;
        for a:=1 to 40 do
                for b:=1 to 100 do
                begin
                        array_xt[a,b]:=0;
                        array_yt[a,b]:=0;
                end;

        for a:=1 to 5000 do
        begin
                Array_x[a]:=0;
                Array_y[a]:=0;
                Array_xx[a]:=0;
                Array_yy[a]:=0;
                array_sx[a]:=0;
                array_nx[a]:=0;
                array_cx[a]:=0;
        end;
        i:=1;
        j:=2;
        m:=1;
                ///clear form for new data
        invalidate;
end;

procedure TForm1.Findoptimize1Click(Sender: TObject);
var a,b,c,d,i,j,{sta,stb,}index,threshold,high,width:integer;
begin
                ////clear array befor find optimize value
        for b:=1 to 40 do
                for c:=1 to 40 do
                        optimize_xy[b,c]:=0;
        for b:=1 to 40 do
                for c:=1 to 100 do
                begin
                        optimize_conx[b,c]:=0;
                        optimize_cony[b,c]:=0;
                end;
                ////open file & load data & find optimize of picture_xy
        assignfile(picture,'picture_xy.unt');
        reset(picture,4);
        last:=trunc(filesize(picture)/1600);
        for a:=0 to trunc(filesize(picture)/1600)-1 do
        begin
                seek(picture,(a*1600));
                blockread(picture,array_xy,1600);
                for b:=1 to 40 do
                        for c:=1 to 40 do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        if array_xy[b,c]=1 then
                                optimize_xy[b,c]:=optimize_xy[b,c]+1
                        else
                                optimize_xy[b,c]:=optimize_xy[b,c]-1;
                end;
        end;
        closefile(picture);
                ////open file & load data & find optimize of contour_x
        assignfile(conx,'contour_x.unt');
        reset(conx,4);
        for a:=0 to trunc(filesize(conx)/4000)-1 do
        begin
                seek(conx,(a*4000));
                blockread(conx,array_xt,4000);
                for b:=1 to 40 do
                        for c:=1 to 100 do
                        begin
                                if array_xt[b,c]=1 then
                                        optimize_conx[b,c]:=optimize_conx[b,c]+1
                                else
                                        optimize_conx[b,c]:=optimize_conx[b,c]-1;
                        end;
        end;
        closefile(conx);
                ////open file & load data & find optimize of contour_y
        assignfile(cony,'contour_y.unt');
        reset(cony,4);
        for a:=0 to trunc(filesize(cony)/4000)-1 do
        begin
                seek(cony,(a*4000));
                blockread(cony,array_yt,4000);
                for b:=1 to 40 do
                        for c:=1 to 100 do
                        begin
                                if array_yt[b,c]=1 then
                                        optimize_cony[b,c]:=optimize_cony[b,c]+1
                                else
                                        optimize_cony[b,c]:=optimize_cony[b,c]-1;
                        end;
        end;
        closefile(cony);
                ///*****/// ////find coordinate 1..800 of best optimize and keep in array
//*1*//}{stb:=160;}
        index:=1;  threshold:=0;
        repeat
                c:=1;
                repeat
                        d:=1;
                        repeat
                                if abs(optimize_xy[c,d])=threshold then
                                        index:=index+1;
                                inc(d);
                        until (index>=stb) or (d=41);
                        inc(c);



                until (index>=stb) or (c=41);
                inc(threshold);
        until (index>=stb) or (threshold=last+1);
        dec(threshold);
        index:=1;
        for c:=1 to 40 do
                for d:=1 to 40 do
                        if abs(optimize_xy[c,d])<=threshold then
                        begin
                                ax[index]:=c;
                                ay[index]:=d;
                                inc(index);
                        end;
        index:=1;  threshold:=0;
        repeat
                c:=1;
                repeat
                        d:=1;
                        repeat
                                if abs(optimize_conx[c,d])=threshold then
                                        index:=index+1;
                                inc(d);
                        until (index>=stb) or (d=101);
                        inc(c);
                until (index>=stb) or (c=41);
                inc(threshold);
        until (index>=stb) or (threshold=last+1);
        dec(threshold);
        index:=1;
        for c:=1 to 40 do
                for d:=1 to 100 do
                        if abs(optimize_conx[c,d])<=threshold then
                        begin
                                aax[index]:=c;
                                aaxt[index]:=d;
                                inc(index);
                        end;
        index:=1;  threshold:=0;
        repeat
                c:=1;
                repeat
                        d:=1;
                        repeat
                                if abs(optimize_cony[c,d])=threshold then
                                        index:=index+1;
                                inc(d);
                        until (index>=stb) or (d=101);
                        inc(c);
                until (index>=stb) or (c=41);
                inc(threshold);
        until (index>=stb) or (threshold=last+1);
        dec(threshold);
        index:=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                for d:=1 to 100 do
                        if abs(optimize_cony[c,d])<=threshold then
                        begin
                                aay[index]:=c;
                                aayt[index]:=d;
                                inc(index);
                        end;
                ////make new file & save optimize coordinate into that file
{//*2*//}{sta:=160;}
        randomize;
        assignfile(optimize,'optimize.dat');
        rewrite(optimize);
        for c:=1 to sta do   ///**
                write(optimize,ax[c]);
        for c:=sta+1 to totalleg do
        begin
                ax[c]:=random(40);
                write(optimize,ax[c]);
        end;
        for c:=1 to sta do   ///**
                write(optimize,ay[c]);
        for c:=sta+1 to totalleg do
        begin
                ay[c]:=random(40);
                write(optimize,ay[c]);
        end;
        for c:=1 to sta do   ///**
                write(optimize,aax[c]);
        for c:=sta+1 to totalleg do
        begin
                aax[c]:=random(40);
                write(optimize,aax[c]);
        end;
        for c:=1 to sta do   ///**
                write(optimize,aaxt[c]);
        for c:=sta+1 to totalleg do
        begin
                aaxt[c]:=random(100);
                write(optimize,aaxt[c]);
        end;
        for c:=1 to sta do   ///**
                write(optimize,aay[c]);
        for c:=sta+1 to totalleg do
        begin
                aay[c]:=random(40);
                write(optimize,aay[c]);
        end;
        for c:=1 to sta do   ///**
                write(optimize,aayt[c]);
        for c:=sta+1 to totalleg do
        begin
                aayt[c]:=random(100);
                write(optimize,aayt[c]);
        end;



        closefile(optimize);
end;

procedure TForm1.FindBinary1Click(Sender: TObject);
var co_x,co_y:array[1..(3*leg)]of byte;
    zb,zd,ze,zg,sk:integer;
begin
                ///set length dynamic array
        setlength(bivalue,100,last,(3*leg));     ///*
                ///assign neural leg
        for zd:=1 to 100 do
        begin
                tempx:=ax;
                tempy:=ay;
                for zb:=0 to (leg-1) do
                begin
                        co_x[zb+1]:=tempx[zd+(100*zb)];
                        co_y[zb+1]:=tempy[zd+(100*zb)];
                end;
                tempx:=aax;
                tempy:=aaxt;
                for zb:=leg to ((2*leg)-1) do
                begin
                        co_x[zb+1]:=tempx[zd+(100*(zb mod leg))];
                        co_y[zb+1]:=tempy[zd+(100*(zb mod leg))];
                end;
                tempx:=aay;
                tempy:=aayt;
                for zb:=(2*leg) to ((3*leg)-1) do
                begin
                        co_x[zb+1]:=tempx[zd+(100*(zb mod leg))];
                        co_y[zb+1]:=tempy[zd+(100*(zb mod leg))];
                end;
                /////load test pattern in two dimension array and
                        ///// catch test pattern
                for zg:=1 to last do
                begin
                        assignfile(allpicture,'picture_xy.unt');
                        reset(allpicture,4);
                        seek(allpicture,(zg-1)*1600);
                        blockread(allpicture,patternpic,1600);
                        closefile(allpicture);
                        for ze:=1 to leg do
                                bivalue[zd-1,zg-1,ze-1]:=patternpic[co_x[ze],co_y[ze]]; ///*
                        assignfile(allpicture,'contour_x.unt');
                        reset(allpicture,4);
                        seek(allpicture,(zg-1)*4000);
                        blockread(allpicture,patterncon,4000);
                        closefile(allpicture);
                        for ze:=(leg+1) to (2*leg) do
                                bivalue[zd-1,zg-1,ze-1]:=patterncon[co_x[ze],co_y[ze]]; ///*
                        assignfile(allpicture,'contour_y.unt');
                        reset(allpicture,4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        blockread(allpicture,patterncon,4000);
                        closefile(allpicture);
                        for ze:=((2*leg)+1) to (3*leg) do
                                bivalue[zd-1,zg-1,ze-1]:=patterncon[co_x[ze],co_y[ze]];  ///*
                end;
        end;
        assignfile(allpicture,'bivalue.dat');
        rewrite(allpicture,4);
        seek(allpicture,0);
        for zd:=0 to 99 do
                for zg:=0 to last-1 do
                        for ze:=0 to ((3*leg)-1) do
                        begin
                                sk:=bivalue[zd,zg,ze];
                                blockwrite(allpicture,sk,1);
                        end;
        closefile(allpicture);
end;

procedure TForm1.ClearFile1Click(Sender: TObject);
begin
        assignfile(picture,'picture_xy.unt');
        rewrite(picture,4);
        closefile(picture);
        assignfile(conx,'contour_x.unt');
        rewrite(conx,4);
        closefile(conx);
        assignfile(cony,'contour_y.unt');
        rewrite(cony,4);
        closefile(cony);
end;

end.
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. โปรแกรมที่ใช้ทดสอบการรู้จำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Menus, ExtCtrls, ComCtrls, Printers;

type
  TForm1 = class(TForm)
    Timer1: TTimer;
    StatusBar1: TStatusBar;
    Label1: TLabel;
    Label2: TLabel;
    Memo1: TMemo;
    Memo2: TMemo;
    Shape1: TShape;
    Button1: TButton;
    Label3: TLabel;
    Label4: TLabel;
    Button2: TButton;
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    OpenDialog1: TOpenDialog;
    SaveDialog1: TSaveDialog;
    FontDialog1: TFontDialog;
    PrintDialog1: TPrintDialog;
    PrinterSetupDialog1: TPrinterSetupDialog;
    Open1: TMenuItem;
    Open2: TMenuItem;
    Save1: TMenuItem;
    Print1: TMenuItem;
    haiText1: TMenuItem;
    BrailleText1: TMenuItem;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    Exit1: TMenuItem;
    haiText2: TMenuItem;
    BrailleText2: TMenuItem;
    haiText3: TMenuItem;
    BrailleText3: TMenuItem;
    FontDialog2: TFontDialog;
    Timer2: TTimer;
    procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure FormMouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure FormMouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure FormCreate(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);




    procedure Open1Click(Sender: TObject);
    procedure haiText1Click(Sender: TObject);
    procedure BrailleText1Click(Sender: TObject);
    procedure haiText2Click(Sender: TObject);
    procedure BrailleText2Click(Sender: TObject);
    procedure haiText3Click(Sender: TObject);
    procedure BrailleText3Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

const
  leg=10;
  totalleg=(leg*100);

var
  Form1: TForm1;
  fdragging : boolean;
  FRect :TRect;
  c,d,i,j,m,ln,answer,last,score:integer;
  array_nx,array_sx:array[1..2500] of integer;
  array_cx,array_tx,array_x,array_y,array_xx,array_yy :array[1..5000] of integer;
  array_xy,patternpic:array[1..40,1..40] of integer;
  array_xt,array_yt,patterncon:array[1..40,1..100] of integer;
  allpicture:file;
  optimize:file of byte;
  ax,ay,aax,aaxt,aay,aayt,tempx,tempy:array[1..totalleg] of byte;
  optidy:array of array of integer;
  scoredy:array of integer;
  bivalue:array of array of array of integer;

implementation

uses Unit2, Unit3;

{$R *.dfm}

function find_min(xyarray:array of integer;last:integer):integer;
var index:integer;
begin
        result:=600;
        for index:=1 to last do
                if (xyarray[index]<result) and (xyarray[index]<>600) then
                        result:=xyarray[index];
end;

function find_max(xyarray:array of integer;last:integer):integer;
var index:integer;
beg.n
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        for index:=1 to last do
                if (xyarray[index]>result) and (xyarray[index]<>600) then
                        result:=xyarray[index];
end;
function scaling(sxy,nwidth,owidth:integer):integer;
var pans:real;
begin
        if (sxy<>600) and (owidth<>0) then
        begin
                pans:=sxy*(nwidth/owidth);
                result:=round(pans);
        end
        else if (sxy<>600) and (owidth=0) then
                        result:=sxy
             else
                        result:=600;
end;
function translate(data:char):string;
begin
        case data of
                ';':translate:='g';
                'ç':translate:='k';
                '£':translate:='0k';
                'o':translate:='u';
                '¥':translate:='-u';
                '¦':translate:=',u';
                '§':translate:=']';
                '¨':translate:='j';
                '©':translate:='/';
                'ª':translate:='+';
                '«':translate:='!';
                '¬':translate:=',+';
                '-':translate:=',y';
                '®':translate:=',d';
                '¯':translate:=',\';
                '°':translate:=',t';
                '±':translate:=',)';
                '²':translate:='-)';
                '³':translate:=',n';
                '´':translate:='d';
                'µ':translate:='\';
                '¶':translate:='t';
                '·':translate:='(';
                '¸':translate:='0)';
                '¹':translate:='n';
                ' ':translate:='v';
                '»':translate:='&';
                '¼':translate:='p';
                '½':translate:='x';
                '¾':translate:='?';
                '¿':translate:='$';
                'À':translate:='-?';
                'Á':translate:='m';
                'Â':translate:='y';



                'Ã':translate:='r';
                'Å':translate:='l';
                'Ç':translate:='w';
                'È':translate:=',s';
                'É':translate:='-s';
                'Ê':translate:='s';
                'Ë':translate:='h';
                'Ì':translate:=',1';
                'Í':translate:='o';
                'Î':translate:='=';
                'Ð':translate:='a';
                'Ò':translate:='*';
                'Ô':translate:='b';
                'Õ':translate:='2';
                'Ö':translate:='[';
                '×':translate:='5';
                'â':begin   ///f
                        translate:='i';
                    end;
                'Ó':translate:='z';
                'ä':translate:=':';
                'ã':translate:=':1';
                'à':begin   ///g
                        translate:='f';
                    end;
                'á':begin   ///c
                        translate:='<';
                    end;
                'Ø':translate:='c';
                'Ù':translate:='3';
                'Ï':translate:=';2';
                'æ':translate:='1';
                'Ä':translate:='rl';
                'Æ':translate:='ll';
                'è':translate:='9';
                'é':translate:='4';
                'ê':translate:='7';
                'ë':translate:='8';
                'ì':translate:='0';
                'Ñ':begin    ///y
                        translate:='>';
                    end;
                'ç':translate:='''';
        else        translate:='';
        end;
end;

procedure TForm1.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
        statusbar1.Panels[0].Text := format ('Mouse in x=%d,y=%d',[x,y]);
        if fdragging then
        begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if (c<>x) or (d<>y) then
                begin
                        Array_x[i]:=x;
                        Array_y[i]:=y;
                        i:=i+1;
                end;
        end;
end;

procedure TForm1.FormMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
        if button = mbleft then
        begin
                fdragging := true;
                canvas.MoveTo(x,y);
                c:=x;
                d:=y;
                timer1.Enabled := false;
        end;
end;

procedure TForm1.FormMouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
        if button = mbleft then
        begin
                fdragging := false;
                array_x[i]:=600;
                array_y[i]:=600;
                i:=i+1;
                timer1.Enabled := true;
        end;
end;

procedure TForm1.FormCreate(Sender: TObject);
var zd,zg,ze,sk:integer;
begin
        i:=1;
        j:=2;
        m:=1;
                 /////find amount of test pattern
        assignfile(allpicture,'picture_xy.unt');
        reset(allpicture,4);
        last:=filesize(allpicture) div 1600;
        closefile(allpicture);
                ////setlength bivalue
        setlength(bivalue,100,last,(3*leg));
                /////receive binary value of neural leg from file
        assignfile(allpicture,'bivalue.dat');
        reset(allpicture,4);
        seek(allpicture,0);
        for zd:=0 to 99 do
                for zg:=0 to last-1 do




                        for ze:=0 to ((3*leg)-1) do
                        begin
                                blockread(allpicture,sk,1);          ///*
                                bivalue[zd,zg,ze]:=sk;
                        end;
        closefile(allpicture);
end;

procedure TForm1.Timer1Timer(Sender: TObject);
var a,b,h,k,t,u,v,z,pt,max,xmin,ymin,xmax,ymax,alpha,angle,za,zb,zd,ze,zf,zg,zh,terminate:integer;
    co_x,co_y,test,pattern:array[1..(3*leg)]of byte;
    zc:byte;
    str,path,data:string;
begin
                //////disable timer and move coordinate to origin
                //////for scaling
        timer1.Enabled:= false;
        xmin:=find_min(array_x,i-2);
        ymin:=find_min(array_y,i-2);
        for u:=1 to i-1 do
        begin
                if (array_x[u]<>600) and (array_y[u]<>600) then
                begin
                        array_x[u]:=array_x[u]-xmin;
                        array_y[u]:=array_y[u]-ymin;
                end;
        end;
                        /////find min max again for scaling
        xmin:=find_min(array_x,i-2);
        ymin:=find_min(array_y,i-2);
        xmax:=find_max(array_x,i-2);
        ymax:=find_max(array_y,i-2);
        if (xmax-xmin)>(ymax-ymin) then
                max:=xmax-xmin
        else
                max:=ymax-ymin;
                 ///////scaling to 50
        for u:=1 to i do
        begin
                array_x[u]:= scaling(array_x[u],50,max);
                array_y[u]:= scaling(array_y[u],50,max);
        end;
                        /////find douplicate coordinate
        for u:=1 to i do
                for v:=1 to i do
                begin
                        if (array_x[u]=array_x[v]) and
                        (array_y[u]=array_y[v])
                        and (u<>v) then
                        begin
                                array_x[v]:=100;
                                array_y[v]:=100;
                        end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            //////pixels scaning in case of large letter to small
v:=1;
j:=2;
m:=1;
for u:=1 to 5000 do
        if (array_x[u]<>100) and (array_y[u]<>100) then
        begin
                array_x[v]:=array_x[u];
                array_y[v]:=array_y[u];
                inc(v);
        end;
        //////extend line in case of small letter to large
array_xx[1] := array_x[1];
array_yy[1] := array_y[1];
repeat
        if (array_x[m+1]=600) and (array_y[m+1]=600) then
                m:=m+2;
        h:=(array_x[m+1]-array_x[m]);
        k:=(array_y[m+1]-array_y[m]);
        if (h=0) and (k<0) then
        begin
                array_xx[j]:=array_x[m];
                array_yy[j]:=(array_y[m]-1);
        end;
        if (h>0) and (k<0) then
        begin
                array_xx[j]:=(array_x[m]+1);
                array_yy[j]:=(array_y[m]-1);
        end;
        if (h>0) and (k=0) then
        begin
                array_xx[j]:=(array_x[m]+1);
                array_yy[j]:=array_y[m];
        end;
        if (h>0) and (k>0) then
        begin
                array_xx[j]:=(array_x[m]+1);
                array_yy[j]:=(array_y[m]+1);
        end;
        if (h=0) and (k>0) then
        begin
                array_xx[j]:=array_x[m];
                array_yy[j]:=(array_y[m]+1);
        end;
        if (h<0) and (k>0) then
        begin
                array_xx[j]:=(array_x[m]-1);
                array_yy[j]:=(array_y[m]+1);
        end;
        if (h<0) and (k=0) then
        begin
                array_xx[j]:=(array_x[m]-1);
                array_yy[j]:=array_y[m];
        end;




        if (h<0) and (k<0) then
        begin
                array_xx[j]:=(array_x[m]-1);
                array_yy[j]:=(array_y[m]-1);
        end;
                repeat
                h:=(array_x[m+1]-array_xx[j]);
                k:=(array_y[m+1]-array_yy[j]);
                if (h=0) and (k<0) then
                begin
                        inc(j);
                        array_xx[j]:=array_xx[j-1];
                        array_yy[j]:=(array_yy[j-1]-1);
                end;
                if (h>0) and (k<0) then
                begin
                        inc(j);
                        array_xx[j]:=(array_xx[j-1]+1);
                        array_yy[j]:=(array_yy[j-1]-1);
                end;
                if (h>0) and (k=0) then
                begin
                        inc(j);
                        array_xx[j]:=(array_xx[j-1]+1);
                        array_yy[j]:=array_yy[j-1];
                end;
                if (h>0) and (k>0) then
                begin
                        inc(j);
                        array_xx[j]:=(array_xx[j-1]+1);
                        array_yy[j]:=(array_yy[j-1]+1);
                end;
                if (h=0) and (k>0) then
                begin
                        inc(j);
                        array_xx[j]:=array_xx[j-1];
                        array_yy[j]:=(array_yy[j-1]+1);
                end;
                if (h<0) and (k>0) then
                begin
                        inc(j);
                        array_xx[j]:=(array_xx[j-1]-1);
                        array_yy[j]:=(array_yy[j-1]+1);
                end;
                if (h<0) and (k=0) then
                begin
                        inc(j);
                        array_xx[j]:=(array_xx[j-1]-1);
                        array_yy[j]:=array_yy[j-1];
                end;
                if (h<0) and (k<0) then
                begin
                        inc(j);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        array_yy[j]:=(array_yy[j-1]-1);
                end;
                until (array_x[m+1]=array_xx[j]) and (array_y[m+1]=
                array_yy[j]);
inc(j);
inc(m);
until  (array_x[m+3]=array_x[m+2])and(array_y[m+3]=array_y[m+2]);
        ////find stable angle in 45-135 degree
alpha:=45;
v:=1;
repeat
for u:=1 to j do
        array_x[u]:=round(array_xx[u]-((array_yy[u]-array_yy[1])/
        (sin((alpha*pi)/180)/cos((alpha*pi)/180))));
h:=2;
repeat
        for k:=h to j+1 do
                if array_x[h-1]=array_x[k] then
                        inc(array_nx[h-1]);
        inc(h);
until h=j+1;
array_sx[v]:=find_max(array_nx,j);
inc(v);
alpha:=alpha+5;
for h:=1 to j do
        array_nx[h]:=0;
until alpha=140;
angle:=find_max(array_sx,18);
        ////keep stable coordinate
h:=1;
repeat
        if array_sx[h]<>angle then
                h:=h+1;
until array_sx[h]=angle;
angle:=45+((h-1)*5);
for u:=1 to j do
begin
        array_x[u]:=round(array_xx[u]-((array_yy[u]-array_yy[1])/
        (sin((angle*pi)/180)/cos((angle*pi)/180))));
        array_y[u]:=round(array_yy[u]+((array_xx[u]-array_xx[1])/
        (sin((angle*pi)/180)/cos((angle*pi)/180))));
end;
array_xx:=array_x;
array_yy:=array_y;
xmin:=find_min(array_xx,j-2);
ymin:=find_min(array_yy,j-2);
        ////move coordinate of x to origin again after rotate
        ////and move coordinate to center for thickening
for u:=1 to j-1 do
begin
        array_xx[u]:=array_xx[u]-xmin;
        array_xx[u]:=array_xx[u]+5;
        array_yy[u]:=array_yy[u]-ymin;
        array_yy[u]:=array_yy[u]+5;




end;
  /////scaling from 56 to 40 pixels befor thickening
for u:=1 to j-1 do
begin
        array_xx[u]:= scaling(array_xx[u],40,60);
        array_yy[u]:= scaling(array_yy[u],40,60);
end;
        ////thickening
for u:=1 to j-1 do
begin
        array_xy[array_xx[u],array_yy[u]]:=1;
        for v:=-2 to 2 do
        begin
                array_xy[array_xx[u]+v,array_yy[u]-1]:=1;
                array_xy[array_xx[u]+v+1,array_yy[u]-1]:=1;
                array_xy[array_xx[u]+v+1,array_yy[u]]:=1;
                array_xy[array_xx[u]+v+1,array_yy[u]+1]:=1;
                array_xy[array_xx[u]+v,array_yy[u]+1]:=1;
                array_xy[array_xx[u]+v-1,array_yy[u]+1]:=1;
                array_xy[array_xx[u]+v-1,array_yy[u]]:=1;
                array_xy[array_xx[u]+v-1,array_yy[u]-1]:=1;
        end;
end;
     //////////scale contour to 40*100 pixels
for u:=1 to j-1 do
        array_cx[u]:=u;
for u:=1 to 5000 do
begin
        {array_xx[u]:= scaling(array_xx[u],40,60);
         array_yy[u]:= scaling(array_yy[u],40,60);}
        array_cx[u]:= scaling(array_cx[u],100,j-1);
        array_x[u]:=0;
        array_y[u]:=0;
end;
        ////find douplicate coordinate of countour
for h:=1 to 5000 do
        for k:=1 to 5000 do
        begin
                if (array_cx[h]=array_cx[k]) and (h<>k) then
                begin
                        array_xx[k]:=200;
                        array_yy[k]:=200;
                        array_cx[k]:=200;
                end;
        end;
        ////pixels scaning
t:=j;
j:=1;
for u:=1 to t-1 do
        if (array_xx[u]<>200) and (array_cx[u]<>200)
        and (array_yy[u]<>200) then
        begin
                array_x[j]:=array_xx[u];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {array_tx[j]:=array_cx[u];}
                inc(j);
        end;
        ///////transform form one dimention array to two dimention array
for t:=1 to 100 do
begin
        for h:=array_x[t] to 40 do
                array_xt[h,t]:=1;
        for k:=array_y[t] to 40 do
                array_yt[k,t]:=1;
end;
      ///////display handwriting on from
{for u:=1 to 40 do
        for t:=1 to 40 do
                if array_xy[u,t]=1 then
                        form1.Canvas.Pixels[u,t]:= clBlue;
      ///////display countour of x and y on from
for t:=1 to 100 do
begin
        for h:=2 to 41 do
                if array_xt[h,t]=1 then
                        form1.Canvas.Pixels[t,h+100]:= clBlue;
        for k:=1 to 40 do
                if array_yt[k,t]=1 then
                        form1.Canvas.Pixels[t,k+200]:= clBlue;
end;}
                /////load optimize coordinate into array
path:=application.ExeName;
terminate:=pos('PROJECT1.EXE',path);
delete(path,terminate,length(path)-terminate+1);
terminate:=pos('Project1.exe',path);
delete(path,terminate,length(path)-terminate+1);
assignfile(optimize,path+'optimize.dat');
reset(optimize);
for za:=1 to totalleg do
begin
        read(optimize,zc);
        ax[za]:=zc;
end;
for za:=1 to totalleg do
begin
        read(optimize,zc);
        ay[za]:=zc;
end;
for za:=1 to totalleg do
begin
        read(optimize,zc);
        aax[za]:=zc;
end;
for za:=1 to totalleg do
begin
        read(optimize,zc);
        aaxt[za]:=zc;
end;




for za:=1 to totalleg do
begin
        read(optimize,zc);
        aay[za]:=zc;
end;
for za:=1 to totalleg do
begin
        read(optimize,zc);
        aayt[za]:=zc;
end;
closefile(optimize);
        /////setlength dynamic array
setlength(optidy,100,last);
setlength(scoredy,last);
        /////assign neural leg
for zd:=1 to 100 do
begin
        tempx:=ax;
        tempy:=ay;
        for zb:=0 to (leg-1) do
        begin
                co_x[zb+1]:=tempx[zd+(100*zb)];
                co_y[zb+1]:=tempy[zd+(100*zb)];
        end;
        tempx:=aax;
        tempy:=aaxt;
        for zb:=leg to ((2*leg)-1) do
        begin
                co_x[zb+1]:=tempx[zd+(100*(zb mod leg))];
                co_y[zb+1]:=tempy[zd+(100*(zb mod leg))];
        end;
        tempx:=aay;
        tempy:=aayt;
        for zb:=(2*leg) to ((3*leg)-1) do
        begin
                co_x[zb+1]:=tempx[zd+(100*(zb mod leg))];
                co_y[zb+1]:=tempy[zd+(100*(zb mod leg))];
        end;
                /////catch test data
        for ze:=1 to leg do
                test[ze]:=array_xy[co_x[ze],co_y[ze]];
        for ze:=(leg+1) to (2*leg) do
                test[ze]:=array_xt[co_x[ze],co_y[ze]];
        for ze:=((2*leg)+1) to (3*leg) do
                test[ze]:=array_yt[co_x[ze],co_y[ze]];
                /////load test pattern in two dimension array and
                ///// catch test pattern
        for zg:=1 to last do
        begin
                        /////xor & find H.M. of each pattern
                        /////and keep it in dynamic array
                zh:=0;
                for zf:=1 to (3*leg) do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                zh:=zh+1;
                        optidy[zd-1,zg-1]:=zh;
                end;
        end;
                ////find min optimize value & give this pattern one score
        for zd:=0 to 99 do
        begin
                zf:=last;
                for zg:=0 to last-1 do
                        if optidy[zd,zg]<zf then
                                zf:=optidy[zd,zg];
                for zg:=0 to last-1 do
                        if optidy[zd,zg]=zf then
                                scoredy[zg mod 72]:=scoredy[zg mod 72]+1;
        end;
                /////find score max & show result
        zf:=0;
        for zg:=0 to last-1 do
                if scoredy[zg]>zf then
                        zf:=scoredy[zg];
                /////
{3} {form3.show;
        for zg:=0 to 71 do
        begin
                data:=form3.memo1.Lines[((zg+1) mod 72)-1];
                form3.memo1.Lines[((zg+1) mod 72)-1]:=data+'='+inttostr(scoredy[zg]);
        end; }
        score:=zf;
{1} if score>=25 then
        begin
                for zg:=0 to last-1 do
                        if scoredy[zg]=score then
                                answer:=zg+1;
                end;
        for zd:=0 to 99 do
                for zg:=0 to last-1 do
                        optidy[zd,zg]:=0;
        for zg:=0 to last-1 do
                scoredy[zg]:=0;
{2} if score>=25 then
        begin
                answer :=answer mod 72;
                case answer of
                {1:begin
                        memo1.Lines.Text:=memo1.Lines.Text+';';
                   end;
                 1:begin
                        memo1.Lines[ln-1]:=memo1.Lines[ln-1]+';';
                   end;}
                1:begin
                        memo1.Lines.Text:=memo1.Lines.Text+';';
                        ln:=memo1.Lines.Count;
                        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
                   end;




                2:begin
                        memo1.Lines.Text:=memo1.Lines.Text+'¢';
                        ln:=memo1.Lines.Count;
                        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
                   end;
                3:begin
                        memo1.Lines.Text:=memo1.Lines.Text+'£';
                        ln:=memo1.Lines.Count;
                        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
                   end;
                4:begin
                        memo1.Lines.Text:=memo1.Lines.Text+'¤';
                        ln:=memo1.Lines.Count;
                        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
                   end;
                5:begin
                        memo1.Lines.Text:=memo1.Lines.Text+'¥';
                        ln:=memo1.Lines.Count;
                        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
                   end;
                6:begin
                        memo1.Lines.Text:=memo1.Lines.Text+'¦';
                        ln:=memo1.Lines.Count;
                        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
                   end;
                7:begin
                        memo1.Lines.Text:=memo1.Lines.Text+'§';
                        ln:=memo1.Lines.Count;
                        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
                   end;
                8:begin
                        memo1.Lines.Text:=memo1.Lines.Text+'¨';
                        ln:=memo1.Lines.Count;
                        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
                   end;
                9:begin
                        memo1.Lines.Text:=memo1.Lines.Text+'©';
                        ln:=memo1.Lines.Count;
                        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
                   end;
                10:begin
                        memo1.Lines.Text:=memo1.Lines.Text+'ª';
                        ln:=memo1.Lines.Count;
                        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
                   end;
                11:begin
                        memo1.Lines.Text:=memo1.Lines.Text+'«';
                        ln:=memo1.Lines.Count;
                        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
                   end;
                12:begin
                        memo1.Lines.Text:=memo1.Lines.Text+'¬';
                        ln:=memo1.Lines.Count;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end;
13:begin
        memo1.Lines.Text:=memo1.Lines.Text+'-';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
14:begin
        memo1.Lines.Text:=memo1.Lines.Text+'®';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
15:begin
        memo1.Lines.Text:=memo1.Lines.Text+'¯';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
16:begin
        memo1.Lines.Text:=memo1.Lines.Text+'°';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
17:begin
        memo1.Lines.Text:=memo1.Lines.Text+'±';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
18:begin
        memo1.Lines.Text:=memo1.Lines.Text+'²';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
19:begin
        memo1.Lines.Text:=memo1.Lines.Text+'³';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
20:begin
        memo1.Lines.Text:=memo1.Lines.Text+'´';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
21:begin
        memo1.Lines.Text:=memo1.Lines.Text+'µ';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
22:begin
        memo1.Lines.Text:=memo1.Lines.Text+'¶';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
23:begin
        memo1.Lines.Text:=memo1.Lines.Text+'·';
        ln:=memo1.Lines.Count;



        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
24:begin
        memo1.Lines.Text:=memo1.Lines.Text+'¸';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
25:begin
        memo1.Lines.Text:=memo1.Lines.Text+'¹';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
26:begin
        memo1.Lines.Text:=memo1.Lines.Text+'º';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
27:begin
        memo1.Lines.Text:=memo1.Lines.Text+'»';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
28:begin
        memo1.Lines.Text:=memo1.Lines.Text+'¼';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
29:begin
        memo1.Lines.Text:=memo1.Lines.Text+'½';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
30:begin
        memo1.Lines.Text:=memo1.Lines.Text+'¾';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
31:begin
        memo1.Lines.Text:=memo1.Lines.Text+'¿';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
32:begin
        memo1.Lines.Text:=memo1.Lines.Text+'À';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
33:begin
        memo1.Lines.Text:=memo1.Lines.Text+'Á';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
34:begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
35:begin
        memo1.Lines.Text:=memo1.Lines.Text+'Ã';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
36:begin
        memo1.Lines.Text:=memo1.Lines.Text+'Ä';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
37:begin
        memo1.Lines.Text:=memo1.Lines.Text+'Ç';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
38:begin
        memo1.Lines.Text:=memo1.Lines.Text+'È';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
39:begin
        memo1.Lines.Text:=memo1.Lines.Text+'É';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
40:begin
        memo1.Lines.Text:=memo1.Lines.Text+'Ê';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
41:begin
        memo1.Lines.Text:=memo1.Lines.Text+'Ë';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
42:begin
        memo1.Lines.Text:=memo1.Lines.Text+'Ì';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
43:begin
        memo1.Lines.Text:=memo1.Lines.Text+'Í';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
44:begin
        memo1.Lines.Text:=memo1.Lines.Text+'Î';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
45:begin



        memo1.Lines.Text:=memo1.Lines.Text+'Ð';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
46:begin
        memo1.Lines.Text:=memo1.Lines.Text+'Ò';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
47:begin
        memo1.Lines.Text:=memo1.Lines.Text+'Ô';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
48:begin
        memo1.Lines.Text:=memo1.Lines.Text+'Õ';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
49:begin
        memo1.Lines.Text:=memo1.Lines.Text+'Ö';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
50:begin
        memo1.Lines.Text:=memo1.Lines.Text+'×';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
51:begin
        memo1.Lines.Text:=memo1.Lines.Text+'â';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
52:begin
        memo1.Lines.Text:=memo1.Lines.Text+'Ó';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
53:begin
        memo1.Lines.Text:=memo1.Lines.Text+'ã';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
54:begin
        memo1.Lines.Text:=memo1.Lines.Text+'ä';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
    end;
55:begin
        memo1.Lines.Text:=memo1.Lines.Text+'à';
        ln:=memo1.Lines.Count;
        memo1.Lines[ln-1]:=memo1.Lines[ln-1];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        56:begin
                memo1.Lines.Text:=memo1.Lines.Text+'á';
                ln:=memo1.Lines.Count;
                memo1.Lines[ln-1]:=memo1.Lines[ln-1];
           end;
        57:begin
                memo1.Lines.Text:=memo1.Lines.Text+'Ø';
                ln:=memo1.Lines.Count;
                memo1.Lines[ln-1]:=memo1.Lines[ln-1];
           end;
        58:begin
                memo1.Lines.Text:=memo1.Lines.Text+'Ù';
                ln:=memo1.Lines.Count;
                memo1.Lines[ln-1]:=memo1.Lines[ln-1];
           end;
        59:begin
                memo1.Lines.Text:=memo1.Lines.Text+'Ï';
                ln:=memo1.Lines.Count;
                memo1.Lines[ln-1]:=memo1.Lines[ln-1];
           end;
        60:begin
                memo1.Lines.Text:=memo1.Lines.Text+'æ';
                ln:=memo1.Lines.Count;
                memo1.Lines[ln-1]:=memo1.Lines[ln-1];
           end;
        61:begin
                memo1.Lines.Text:=memo1.Lines.Text+'Ã';
                ln:=memo1.Lines.Count;
                memo1.Lines[ln-1]:=memo1.Lines[ln-1];
           end;
        62:begin
                memo1.Lines.Text:=memo1.Lines.Text+'Æ';
                ln:=memo1.Lines.Count;
                memo1.Lines[ln-1]:=memo1.Lines[ln-1];
           end;
        63:begin
                memo1.Lines.Text:=memo1.Lines.Text+'è';
                ln:=memo1.Lines.Count;
                memo1.Lines[ln-1]:=memo1.Lines[ln-1];
           end;
        64:begin
                memo1.Lines.Text:=memo1.Lines.Text+'é';
                ln:=memo1.Lines.Count;
                memo1.Lines[ln-1]:=memo1.Lines[ln-1];
           end;
        65:begin
                memo1.Lines.Text:=memo1.Lines.Text+'ê';
                ln:=memo1.Lines.Count;
                memo1.Lines[ln-1]:=memo1.Lines[ln-1];
           end;
        66:begin
                memo1.Lines.Text:=memo1.Lines.Text+'ë';
                ln:=memo1.Lines.Count;
                memo1.Lines[ln-1]:=memo1.Lines[ln-1];



           end;
        67:begin
                memo1.Lines.Text:=memo1.Lines.Text+'ì';
                ln:=memo1.Lines.Count;
                memo1.Lines[ln-1]:=memo1.Lines[ln-1];
           end;
        68:begin
                memo1.Lines.Text:=memo1.Lines.Text+'Ñ';
                ln:=memo1.Lines.Count;
                memo1.Lines[ln-1]:=memo1.Lines[ln-1];
           end;
        69:begin
                memo1.Lines.Text:=memo1.Lines.Text+'ç';
                ln:=memo1.Lines.Count;
                memo1.Lines[ln-1]:=memo1.Lines[ln-1];
           end;
        70:begin
                pt:=memo1.Lines.Count;
                if memo1.Lines.Count > 0 then
                begin
                        str:=memo1.Lines[pt-1];
                        delete(str,length(str),1);
                        memo1.Lines[pt-1]:=str;
                end
                else
                begin
                        str:=memo1.Lines[pt];
                        delete(str,length(str),1);
                        memo1.Lines[pt]:=str;
                end;

           end;
        71:begin
                memo1.Lines.Add('');
           end;
        0 :begin
                memo1.Lines.Text:=memo1.Lines.Text+' ';
                ln:=memo1.Lines.Count;
                memo1.Lines[ln-1]:=memo1.Lines[ln-1];
           end;
        end;
end
else
begin
        form2.show;
        timer2.Enabled:=true;
end;
for a:=1 to 40 do
        for b:=1 to 40 do
                array_xy[a,b]:=0;
for a:=1 to 40 do
        for b:=1 to 100 do
        begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                array_yt[a,b]:=0;
            end;
        for a:=1 to 5000 do
        begin
                Array_x[a]:=0;
                Array_y[a]:=0;
                Array_xx[a]:=0;
                Array_yy[a]:=0;
                array_sx[a]:=0;
                array_nx[a]:=0;
                array_cx[a]:=0;
        end;
        i:=1;
        j:=2;
        m:=1;
        invalidate;

nd;
rocedure TForm1.Button2Click(Sender: TObject);
egin
        memo1.Clear;
        memo2.Clear;
nd;

rocedure TForm1.Button1Click(Sender: TObject);
ar txt:string;
   a,b,long,last:integer;
egin
            ////clear memo2 and check memo1 have data or not
        memo2.Clear;
        long:=memo1.Lines.Count;
        if (long<>0) then
        begin
        a:=0;
        repeat{for a:=0 to long do
            begin}
                    ///set amount of character in each line
                txt:=memo1.Lines[a];
                if txt<>'' then
                    last:=length(txt)
                else    last:=1;
                txt:=txt+'   ';
            //// transfer thai character to braille character
            b:=1;
            repeat
                {for b:=1 to last do
                begin}
                    case txt[1] of
                    'à' :   begin
                            if (txt[1]='à') and (txt[3]='Ð') then
                            begin
                                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                                memo2.Lines.Text:=memo2.Lines.Text+'fa';
                                delete(txt,1,3);



                                b:=b+3;
                            end  else
                            if (txt[1]='à') and (txt[3]='Õ') and (txt[4]='Ð') then
                            begin
                                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                                memo2.Lines.Text:=memo2.Lines.Text+'oa';
                                delete(txt,1,4);
                                b:=b+4;
                            end  else
                            if (txt[1]='à') and (txt[3]='Í') and (txt[4]='Ð') then
                            begin
                                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                                memo2.Lines.Text:=memo2.Lines.Text+'%a';
                                delete(txt,1,4);
                                b:=b+4;
                            end  else
                            if (txt[1]='à') and (txt[3]='Í') and (txt[4]<>'Ð') then
                            begin
                                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                                memo2.Lines.Text:=memo2.Lines.Text+'%';
                                delete(txt,1,3);
                                b:=b+3;
                            end  else
                            if (txt[1]='à') and (txt[3]='Õ') and (txt[4]='Â') and
                            (txt[5]='Ð') then
                            begin
                                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                                memo2.Lines.Text:=memo2.Lines.Text+'(a';
                                delete(txt,1,5);
                                b:=b+5;
                            end  else
                            if (txt[1]='à') and (txt[3]='Õ') and (txt[4]='Â') and
                            (txt[5]<>'Ð') then
                            begin
                                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                                memo2.Lines.Text:=memo2.Lines.Text+'(';
                                delete(txt,1,4);
                                b:=b+4;
                            end  else
                            if (txt[1]='à') and (txt[3]='x') and (txt[4]='Í') and
                            (txt[5]='Ð') then
                            begin
                                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                                memo2.Lines.Text:=memo2.Lines.Text+'qa';
                                delete(txt,1,5);
                                b:=b+5;
                            end  else
                            if (txt[1]='à') and (txt[3]='x') and (txt[4]='Í') and
                            (txt[5]<>'Ð') then
                            begin
                                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                                memo2.Lines.Text:=memo2.Lines.Text+'q';
                                delete(txt,1,4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end  else
if (txt[1]='à') and (txt[3]='Ò') and (txt[4]<>'Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+'6';
        delete(txt,1,3);
        b:=b+3;
end  else
if (txt[1]='à') and (txt[3]='Ô') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+'%';
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[4]);
        delete(txt,1,4);
        b:=b+4;
end  else
//////*
if (txt[1]='à') and (txt[3]in['á','é','ê','ë']) and
(txt[4]='Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+'fa';
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
        delete(txt,1,4);
        b:=b+4;
end  else
if (txt[1]='à') and (txt[3]in['è','é','ê','ë']) and
(txt[4]='Ò') and (txt[5]='Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+'oa';
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
        delete(txt,1,5);
        b:=b+5;
end  else
if (txt[1]='à') and (txt[3]in['è','é','ê','ë']) and
(txt[4]='Í') and (txt[5]='Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+'%a';
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
        delete(txt,1,5);
        b:=b+5;
end  else
if (txt[1]='à') and (txt[3]in['è','é','ê','ë']) and
(txt[4]='Í') and (txt[5]<>'Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+'%';
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
        delete(txt,1,4);
        b:=b+4;
end  else
if (txt[1]='à') and (txt[3]='Õ') and (txt[4]in['è','é','ê','ë'])
and (txt[5]='Â') and (txt[6]='Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+'(a';
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[4]);
        delete(txt,1,6);
        b:=b+6;
end  else
if (txt[1]='à') and (txt[3]='Õ') and (txt[4]in['è','é','ê','ë'])
 and (txt[5]='Â') and (txt[6]<>'Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+'(';
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[4]);
        delete(txt,1,5);
        b:=b+5;
end  else
if (txt[1]='à') and (txt[3]='×') and (txt[4]in['è','é','ê','ë'])
 and (txt[5]='Í') and (txt[6]='Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+'qa';
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[4]);
        delete(txt,1,6);
        b:=b+6;
end  else
if (txt[1]='à') and (txt[3]='×') and (txt[4]in['è','é','ê','ë'])
 and (txt[5]='Í') and (txt[6]<>'Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+'q';
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[4]);
        delete(txt,1,5);
        b:=b+5;
end  else
if (txt[1]='à') and (txt[3]in['è','é','ê','ë']) and (txt[4]='Ò')
and (txt[5]<>'Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+'6';
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
        delete(txt,1,4);
        b:=b+4;
end  else
if (txt[1]='à') and (txt[3]='Ô') and (txt[4]in['è','é','ê','ë'])
 then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+'%';
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[4]);
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[5]);
        delete(txt,1,5);
        b:=b+5;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//////*
//////**
if (txt[1]='à') and (txt[3]in['Ã','Á','Ç']) and (txt[4]='Ð')
then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
        memo2.Lines.Text:=memo2.Lines.Text+'fa';
        delete(txt,1,4);
        b:=b+4;
end  else
if (txt[1]='à') and (txt[3]in['Ã','Á','Ç']) and (txt[4]='Ò')
 and (txt[5]='Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
        memo2.Lines.Text:=memo2.Lines.Text+'oa';
        delete(txt,1,5);
        b:=b+5;
end  else
if (txt[1]='à') and (txt[3]in['Ã','Á','Ç']) and (txt[4]='Í')
and (txt[5]='Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
        memo2.Lines.Text:=memo2.Lines.Text+'%a';
        delete(txt,1,5);
        b:=b+5;
end  else
if (txt[1]='à') and (txt[3]in['Ã','Á','Ç']) and (txt[4]='Í')
 and (txt[5]<>'Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
        memo2.Lines.Text:=memo2.Lines.Text+'%';
        delete(txt,1,4);
        b:=b+4;
end  else
if (txt[1]='à')and(txt[3]in['Ã','Á','Ç']) and (txt[4]='Õ')
and (txt[5]='Â') and (txt[6]='Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
        memo2.Lines.Text:=memo2.Lines.Text+'(a';
        delete(txt,1,6);
        b:=b+6;
end  else
if (txt[1]='à') and (txt[3]in['Ã','Á','Ç']) and (txt[4]='Õ')
and (txt[5]='Â') and (txt[6]<>'Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
        memo2.Lines.Text:=memo2.Lines.Text+'(';
        delete(txt,1,5);



        b:=b+5;
end  else
if (txt[1]='à') and  (txt[3]in['Ã','Á','Ç']) and (txt[4]='×')
and (txt[5]='Í') and (txt[6]='Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
        memo2.Lines.Text:=memo2.Lines.Text+'qa';
        delete(txt,1,6);
        b:=b+6;
end  else
if (txt[1]='à') and  (txt[3]in['Ã','Á','Ç']) and (txt[4]='×')
 and (txt[5]='Í') and (txt[6]<>'Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
        memo2.Lines.Text:=memo2.Lines.Text+'q';
        delete(txt,1,5);
        b:=b+5;
end  else
if (txt[1]='à') and (txt[3]in['Ã','Á','Ç']) and  (txt[4]='Ò')
and (txt[5]<>'Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
        memo2.Lines.Text:=memo2.Lines.Text+'6';
        delete(txt,1,4);
        b:=b+4;
end  else
if (txt[1]='à') and  (txt[3]in['Ã','Á','Ç']) and (txt[4]='Ô')
 then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
        memo2.Lines.Text:=memo2.Lines.Text+'%';
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[5]);
        delete(txt,1,5);
        b:=b+5;
end  else
//////**
//////***
if (txt[1]='à') and (txt[3]in['Ã','Á','Ç']) and (txt[4]in['è','é','ê','ë'])
 and (txt[5]='Ð') then
begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
        memo2.Lines.Text:=memo2.Lines.Text+'fa';
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[4]);
        delete(txt,1,5);
        b:=b+5;
end  else
if (txt[1]='à') and (txt[3]in['Ã','Á','Ç']) and (txt[4]in['è','é','ê','ë'])
and (txt[5]='Ò') and (txt[6]='Ð') then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

```
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
                memo2.Lines.Text:=memo2.Lines.Text+'oa';
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[4]);
                delete(txt,1,6);
                b:=b+6;
        end  else
        if (txt[1]='à') and (txt[3]in['Ã','Á','Ç']) and (txt[4]in['è','é','ê','ë'])
         and (txt[5]='Í') and (txt[6]='Ð') then
        begin
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
                memo2.Lines.Text:=memo2.Lines.Text+'%a';
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[4]);
                delete(txt,1,6);
                b:=b+6;
        end  else
        if (txt[1]='à') and (txt[3]in['Ã','Á','Ç']) and (txt[4]in['è','é','ê','ë'])
        and (txt[5]='Í') and (txt[6]<>'Ð') then
        begin
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
                memo2.Lines.Text:=memo2.Lines.Text+'%';
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[4]);
                delete(txt,1,5);
                b:=b+5;
        end  else
        if (txt[1]='à') and (txt[3]in['Ã','Á','Ç']) and (txt[4]='Õ') and
        (txt[5]in['è','é','ê','ë']) and (txt[6]='Â') and (txt[7]='Ð') then
        begin
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
                memo2.Lines.Text:=memo2.Lines.Text+'(a';
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[5]);
                delete(txt,1,7);
                b:=b+7;
        end  else
        if (txt[1]='à') and (txt[3]in['Ã','Á','Ç']) and (txt[4]='Õ') and
        (txt[5]in['è','é','ê','ë']) and (txt[6]='Â') and (txt[7]<>'Ð') then
        begin
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
                memo2.Lines.Text:=memo2.Lines.Text+'(';
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[5]);
                delete(txt,1,6);
                b:=b+6;
        end  else
        if (txt[1]='à') and (txt[3]in['Ã','Á','Ç']) and (txt[4]='×') and
        (txt[5]in['è','é','ê','ë']) and (txt[6]='Í') and (txt[7]='Ð') then
        begin
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
                memo2.Lines.Text:=memo2.Lines.Text+'qa';
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[5]);



                delete(txt,1,7);
                b:=b+7;
        end  else
        if (txt[1]='à') and (txt[3]in['Ã','Á','Ç']) and (txt[4]='×') and
         (txt[5]in['è','é','ê','ë']) and (txt[6]='Í') and (txt[7]<>'Ð') then
        begin
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
                memo2.Lines.Text:=memo2.Lines.Text+'q';
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[5]);
                delete(txt,1,6);
                b:=b+6;
        end  else
        if (txt[1]='à') and (txt[3]in['Ã','Á','Ç']) and (txt[4]in['è','é','ê','ë'])
        and (txt[5]='Ò') and (txt[6]<>'Ð') then
        begin
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
                memo2.Lines.Text:=memo2.Lines.Text+'6';
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[4]);
                delete(txt,1,5);
                b:=b+5;
        end  else
        if (txt[1]='à') and (txt[3]in['Ã','Á','Ç']) and (txt[4]='Ô')
        and (txt[5]in['è','é','ê','ë']) then
        begin
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
                memo2.Lines.Text:=memo2.Lines.Text+'%';
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[5]);
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[6]);
                delete(txt,1,6);
                b:=b+6;
        end  else
        //////***
        begin
        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[1]);
        delete(txt,1,1);
        b:=b+1;
        end;
end;
'á' : begin
        if (txt[1]='á') and (txt[3]='Ð') then
        begin
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                memo2.Lines.Text:=memo2.Lines.Text+'<a';
                delete(txt,1,3);
                b:=b+3;
        end  else
        /////*
        if (txt[1]='á') and (txt[3]in['è','ë','ê','ë']) and (txt[4]='Ð') then
        begin
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

```
                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
                        delete(txt,1,4);
                        b:=b+4;
                end  else
                ////*
                ////**
                if (txt[1]='á') and  (txt[3]in['À','Á','Ç']) and (txt[4]='Ð') then
                begin
                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
                        memo2.Lines.Text:=memo2.Lines.Text+'<a';
                        delete(txt,1,4);
                        b:=b+4;
                end  else
                ////**
                ////***
                if (txt[1]='á') and (txt[3]in['À','Á','Ç']) and
                (txt[4]in['è','é','ê','ë']) and (txt[5]='Ð') then
                begin
                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
                        memo2.Lines.Text:=memo2.Lines.Text+'<a';
                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[4]);
                        delete(txt,1,5);
                        b:=b+5;
                end  else
                ////***
                begin
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[1]);
                delete(txt,1,1);
                b:=b+1;
                end;
        end;
'â' :   begin
                if (txt[1]='â') and (txt[3]='Ð') then
                begin
                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                        memo2.Lines.Text:=memo2.Lines.Text+'ia';
                        delete(txt,1,3);
                        b:=b+3;
                end  else
                /////*
                if (txt[1]='â') and (txt[3]in['à','á','è','ë']) and (txt[4]='Ð') then
                begin
                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                        memo2.Lines.Text:=memo2.Lines.Text+'ia';
                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
                        delete(txt,1,4);
                        b:=b+4;
                end  else
                /////*
                /////**
                if (txt[1]='â') and  (txt[3]in['À','Á','Ç']) and (txt[4]='Ð') then
                begin



                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
                        memo2.Lines.Text:=memo2.Lines.Text+'ia';
                        delete(txt,1,4);
                        b:=b+4;
                end  else
                /////**
                /////***
                if (txt[1]='â') and (txt[3]in['À','Á','Ç']) and
                (txt[4]in['è','é','ê','ë']) and (txt[5]='Ð') then
                begin
                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
                        memo2.Lines.Text:=memo2.Lines.Text+'ia';
                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[4]);
                        delete(txt,1,5);
                        b:=b+5;
                end  else
                /////***
                begin
                memo2.Lines.Text:=memo2.Lines.Text+translate(txt[1]);
                delete(txt,1,1);
                b:=b+1;
                end;
        end;
else
        begin
                if (txt[2]='Ñ') and (txt[3]='Ç') and (txt[4]='Ð') then
                begin
                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[1]);
                        memo2.Lines.Text:=memo2.Lines.Text+'ea';
                        delete(txt,1,4);
                        b:=b+4;
                end  else
                if (txt[2]='Ñ') and (txt[3]='Ç') and (txt[4]<>'Ð') then
                begin
                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[1]);
                        memo2.Lines.Text:=memo2.Lines.Text+'e';
                        delete(txt,1,3);
                        b:=b+3;
                end  else
                //////*
                if (txt[2]='Ñ') and (txt[3]in['è','é','ê','ë']) and
                (txt[4]='Ç') and (txt[5]='Ð') then
                begin
                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[1]);
                        memo2.Lines.Text:=memo2.Lines.Text+'ea';
                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
                        delete(txt,1,5);
                        b:=b+5;
                end  else
                if (txt[2]='Ñ') and (txt[3]in['è','é','ê','ë']) and
                (txt[4]='Ç') and (txt[5]<>'Ð') then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[1]);
                                        memo2.Lines.Text:=memo2.Lines.Text+'e';
                                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[3]);
                                        delete(txt,1,4);
                                        b:=b+4;
                                end  else
                                //////*
                                //////**
                                if  (txt[2]in['Ã','Å','Ç']) and (txt[3]='Ñ') and
                                (txt[4]='Ç') and (txt[5]='Ð') then
                                begin
                                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[1]);
                                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                                        memo2.Lines.Text:=memo2.Lines.Text+'ea';
                                        delete(txt,1,5);
                                        b:=b+5;
                                end  else
                                if (txt[2]in['Ã','Å','Ç']) and (txt[3]='Ñ') and
                                (txt[4]='Ç') and (txt[5]<>'Ð') then
                                begin
                                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[1]);
                                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                                        memo2.Lines.Text:=memo2.Lines.Text+'e';
                                        delete(txt,1,4);
                                        b:=b+4;
                                end  else
                                //////**
                                //////***
                                if (txt[2]in['Ã','Å','Ç']) and (txt[3]='Ñ') and
                                (txt[4]in['è','é','ê','ë']) and (txt[5]='Ç') and (txt[6]='Ð') then
                                begin
                                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[1]);
                                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                                        memo2.Lines.Text:=memo2.Lines.Text+'ea';
                                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[4]);
                                        delete(txt,1,6);
                                        b:=b+6;
                                end  else
                                if (txt[2]in['Ã','Å','Ç']) and (txt[3]='Ñ') and
                                  (txt[4]in['è','é','ê','ë']) and (txt[5]='Ç') and (txt[6]<>'Ð') then
                                begin
                                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[1]);
                                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[2]);
                                        memo2.Lines.Text:=memo2.Lines.Text+'e';
                                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[4]);
                                        delete(txt,1,5);
                                        b:=b+5;
                                end  else
                                //////***
                                begin
                                        memo2.Lines.Text:=memo2.Lines.Text+translate(txt[1]);
                                        delete(txt,1,1);
                                        inc(b);
                                end;



                        end;
                end;
                {end;}
                {inc(b);}
        until b=last+1;
                memo2.Lines.Add('');
        inc(a);
until a=long;
        {end;}
end;
end;
procedure TForm1.Open1Click(Sender: TObject);
var a,b:integer;
begin
        for a:=1 to 40 do
                for b:=1 to 40 do
                        array_xy[a,b]:=0;
        for a:=1 to 40 do
                for b:=1 to 100 do
                begin
                        array_xt[a,b]:=0;
                        array_yt[a,b]:=0;
                end;

        for a:=1 to 5000 do
        begin
                Array_x[a]:=0;
                Array_y[a]:=0;
                Array_xx[a]:=0;
                Array_yy[a]:=0;
                array_sx[a]:=0;
                array_nx[a]:=0;
                array_cx[a]:=0;
        end;
        i:=1;
        j:=2;
        m:=1;
        invalidate;
        {memo1.Clear;
        memo2.Clear;}
end;

Procedure TForm1.HalfText1Click(Sender: TObject);
begin
        if opendialog1.execute then
        memo1.Lines.LoadFromFile(opendialog1.filename);
        timer1.Enabled:= false;
end;

Procedure TForm1.BrailleText1Click(Sender: TObject);
begin
        if opendialog1.execute then
        memo2.Lines.LoadFromFile(opendialog1.filename);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;

procedure TForm1.haiText2Click(Sender: TObject);
begin
        if SaveDialog1.Execute then
        memo1.Lines.SaveToFile(savedialog1.FileName);
end;

procedure TForm1.BrailleText2Click(Sender: TObject);
begin
        if SaveDialog1.Execute then
        memo2.Lines.SaveToFile(savedialog1.FileName);
end;

procedure TForm1.haiText3Click(Sender: TObject);
var c,y,txthigh: Integer;
begin
        y:=10;
        application.ProcessMessages;
        if (memo1.Lines.Count=0) then
        begin
                showmessage('No data in text box'); memo1.SetFocus; exit;
        end;
        statusbar1.Panels[0].Text:='Printing';
        Printer.Canvas.Font.Assign(fontdialog1.Font);
        printer.BeginDoc;
        for c:=0 to memo1.Lines.Count-1 do
        begin
                printer.Canvas.TextOut(10,y,memo1.Lines[c]);
                txthigh:=printer.Canvas.TextHeight(memo1.Lines[c]);
                inc(y,txthigh);
        end;
        printer.EndDoc;
        statusbar1.Panels[0].Text:='';
end;

procedure TForm1.BrailleText3Click(Sender: TObject);
var c,y,txthigh: Integer;
begin
        y:=10;
        application.ProcessMessages;
        if (memo2.Lines.Count=0) then
        begin
                showmessage('No data in text box'); memo2.SetFocus; exit;
        end;
        statusbar1.Panels[0].Text:='Printing';
        Printer.Canvas.Font.Assign(fontdialog2.Font);
        printer.BeginDoc;
        for c:=0 to memo2.Lines.Count-1 do
        begin
                printer.Canvas.TextOut(10,y,memo2.Lines[c]);
                txthigh:=printer.Canvas.TextHeight(memo2.Lines[c]);
                inc(y,txthigh);
        end;




        printer.EndDoc;
        statusbar1.Panels[0].Text:='';
end;

procedure TForm1.Exit1Click(Sender: TObject);
begin
        form1.Close;
        {if fontdialog1.Execute then
        Printer.Canvas.Font.Assign(fontdialog1.Font);}
end;

procedure TForm1.Timer2Timer(Sender: TObject);
begin
        timer2.Enabled:=false;
        form2.hide;
end;

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



4. ตารางการเทียบเคียงอักษรเบรลล์กับอักษรไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางการเทียบเคียงอักษรเบรลล์กับอักษรไทย และค่ารหัส ASCII ในการแปลงข้อมูล

Braille	Character	Decimal Keyboard	Convert Character	Decimal Printer	Braille	Character	Decimal Keyboard	Convert Character	Decimal Printer
	ก	161	g	103		ศ	181	\	92
	ข	162	k	107		ถ	182	t	116
	ฃ	163	0k	48,107		ท	183	•)	41
	ค	164	u	117		ธ	184	0)	48,41
	ฅ	165	-u	45,117		น	185	n	110
	ฆ	166	,u	44,117		บ	186	v	118
	ง	167	]	93		ป	187	&	38
	จ	168	j	106		ผ	188	p	112
	ฉ	169	/	47		ฝ	189	x	120
	ช	170	+	43		พ	190	?	63
	ซ	171	!	33		ฟ	191	$	36
	ฌ	172	,+	44,43		ภ	192	,?	44,63
	ญ	173	,y	44,121		ม	193	m	109
	ฎ	174	,d	44,100		ย	194	y	121
	ฏ	175	,\	44,92		ร	195	r	114
	ฐ	176	,t	44,116		ล	197	l	108
	ฑ	177	,)	44,41		ว	199	w	119
	ฒ	178	-)	45,41		ศ	200	,s	44,115
	ณ	179	,n	44,110		ษ	201	-s	45,115
	ด	180	-l	100		ส	202	s	115

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Braille	Character	Decimal Keyboard	Convert Character	Decimal Printer	Braille	Character	Decimal Keyboard	Convert Character	Decimal Printer
	ห	203	h	104		เอา	-	6	54
	ฬ	204	,l	44,108		เออะ	-	%a	37,97
	อ	205	o	111		เอ็	-	%	37
	ฮ	206	=	61		เอียะ	-	(a	40,97
	ะ	208	a	97		เอีย	-	(	40
	า	210	*	42		เอียะ	-	qa	113,97
	ไ	228	:	58		เอีย	-	q	113
	ใ	227	;l	58,49		อัวะ	-	ca	101,97
	โ	226	i	105		อัว	-	c	101
	อิ	212	b	98		อำ	211	z	122
	อี	213	2	50		ฤ	196	rl	114,49
	อึ	214	l	91		ฦ	198	ll	108,49
	อือ	215	5	53		ไม้เอก	232	9	57
	อุ	216	c	99		ไม้โท	233	4	52
	อู	217	3	51		ไม้ตรี	234	7	55
	เอะ	-	fa	102,97		ไม้จัตวา	235	8	56
	เอ	224	f	102		ทัณฑฆาต	209	>	62
	แอะ	-	<a	60,97		การันต์	236	0	48
	แอ	225	<	60		ไม้ไต่คู้	231	'	39
	โอะ	-	ia	105,97		ๆ	230	1	49
	เอาะ	-	oa	111,97		""	34	80	56,48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงลงได้ด้วยความช่วยเหลือจาก ดร.ยุทธนา คิดใจเดียว และอาจารย์ทุกท่าน ที่กรุณาเสนอแนะแนวทางและวิธีการใหม่ๆ ตลอดจนแนวทางในการแก้ปัญหา ในการดำเนินงานของการจัดทำปริญญานิพนธ์นี้

สุดท้ายนี้ขอขอบคุณอาจารย์ทุกท่านที่ประสิทธิประสาทวิชาให้ผู้จัดทำมีความรู้เพิ่มมากขึ้น และสามารถนำไปใช้ได้ไม่ว่าจะเป็นปัจจุบันหรือในอนาคต

.....

(สุพัฒน์ ธนาศรีวิไล)

.....

(โสภณ ภัทรวงศ์หิรัญ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. ประยงค์ อู่ประสิทธ์วงศ์ , “เรียนรู้การเขียนโปรแกรมด้วยภาษา Object Pascal ” , บริษัท ซัคเซส มีเดีย จำกัด
2. สัจจะ จรัสรุ่งรวีวร ,จักรพงษ์ สุขประเสริฐ , “ เริ่มต้นอย่างมืออาชีพด้วย DELPHI 5 ” สำนักพิมพ์ อินโฟเพส , 2543
3. ดุสิต สุภาสวัสดิ์ , “ เทคนิคการสร้างฟอนด์ด้วยโปรแกรม Fontographer สำหรับวินโดวส์ ” , บริษัท ซีเอ็ดยูเคชั่น จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้