การพัฒนาระบบการรู้จำลายมือออนไลน์สำหรับตัวอักษรภาษาไทยบนแท็บเล็ตพีซี

นางสาววรารัตน์ เอื้ออำนวยชัย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2548

ISBN 974-53-2869-3

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

DEVELOPMENT OF AN ONLINE HANDWRITTEN RECOGNITION SYSTEM FOR THAI CHARACTERS ON TABLET PC

Miss Vararut Auamnuaychai

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2005

ISBN 974-53-2869-3

หัวข้อวิทยานิพนธ์	การพัฒนาระบบการรู้จำลายมือออนไลน์สำหรับตัวอักษรภาษาไทยบนแท็บเล็ตพีซี
โดย	นางสาววรารัตน์ เอื้ออำนวยชัย
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ)

..... กรรมการ
(อาจารย์ ดร.อาทิตย์ ทองทักษ์)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ธาราทิพย์ สุวรรณศาสตร์)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วรารัตน์ เอื้ออำนวยชัย : การพัฒนาระบบการรู้จำลายมือออนไลน์สำหรับตัวอักษรภาษาไทยบนแท็บเล็ตพีซี. (DEVELOPMENT OF AN ONLINE HANDWRITTEN RECOGNITION SYSTEM FOR THAI CHARACTERS ON TABLET PC) อ. ที่ปรึกษา : ผศ. ดร.วิวัฒน์ วัฒนาวุฒิ, 77 หน้า. ISBN 974-53-2869-3.

วิทยานิพนธ์ฉบับนี้มีวัตถุประสงค์เพื่อออกแบบและพัฒนาระบบการรู้จำลายมือออนไลน์สำหรับตัวอักษรภาษาไทยบนแท็บเล็ตพีซี โดยใช้ข่ายงานประสาทแบบแพร่กระจายย้อนกลับในการประมวลผลการรู้จำลายมือเขียน และใช้เทคนิคในการจำแนกตัวอักษร ซึ่งประกอบด้วย การประมวลผลข้อมูลตัวอักษรเบื้องต้น การหาลักษณะเด่นของตัวอักษร และการเข้ารหัส โดยมีเทคนิคที่สำคัญคือการหาวงรอบของตัวอักษร การใช้เขตย่อย การคัดแยกตัวอักษรโดยใช้ระยะห่างของสโตรค การตรวจสอบตัวอักษรก่อนหน้า ระบบที่ได้ประยุกต์การรู้จำตัวอักษรภาษาอังกฤษเพิ่มโดยใช้เอพีไอจากไมโครซอฟท์แท็บเล็ตพีซี พร้อมทั้งรองรับลายมือเขียนจากผู้ใช้หลายคน สามารถส่งตัวอักษรที่รู้จำได้ไปยังโปรแกรมไมโครซอฟท์เวิร์ด และ ไมโครซอฟท์เอ็กซเซล รองรับการปรับเปลี่ยนไฟล์น้ำหนักและค่าพารามิเตอร์ของข่ายงานประสาทได้ตามรูปแบบของโปรแกรมข่ายงานประสาทของมหาวิทยาลัยสตุทการ์ท เพื่อเปิดให้สามารถปรับค่าการรู้จำของระบบ และรองรับการเขียนตัวอักษรครั้งละหลายตัวอักษร

จากการทดสอบระบบการรู้จำลายมือออนไลน์ สำหรับตัวอัษรภาษาไทยบนแท็บเล็ตพีซี ด้วยลายมือเขียนจากผู้ทดสอบจำนวน 3 ท่าน ท่านละ 10 ชุด พบว่าอัตราการรู้จำครั้งละตัวอักษรมีความถูกต้องร้อยละ 89.4 รู้จำผิดร้อยละ 10.6 อัตราการรู้จำครั้งละหลายตัวอักษรมีความถูกต้องร้อยละ 87.7 รู้จำผิดร้อยละ 12.3

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา..... วิศวกรรมคอมพิวเตอร์.....ลายมือชื่อนิสิต.....

สาขาวิชา.....วิทยาศาสตร์คอมพิวเตอร์....ลายมือชื่ออาจารย์ที่ปรึกษา.....

ปีการศึกษา ...2548.....

45714500 : MAJOR COMPUTER SCIENCE

KEY WORD: ONLINE HANDWRITTEN RECOGNITION / BACKPROPAGATION NEURAL NETWORK / TABLET PC

VARARUT AUAMNUAYCHAI : DEVELOPMENT OF AN ONLINE HANDWRITTEN RECOGNITION SYSTEM FOR THAI CHARACTERS ON TABLET PC. THESIS ADVISOR : ASST.PROF. WIWAT VATANAWOOD, PH.D., 77 pp. ISBN 974-53-2869-3.

The research aims to design and develop an online handwritten recognition system for Thai characters on Tablet PC by using the backpropagation neural network. The relevant techniques for defining the characters are data preprocessing, feature extraction and data encoding. The proposed techniques used in this research are circle finding, zone, character separation by using stroke displacement and previous character checking techniques. This system can recognize English characters by using Microsoft Tablet PC API. Moreover, the multiple writers handwriting is also supported and the recognized text will be sent to Microsoft Word and Microsoft Excel. The system provides user to adjust weight file and parameter of neural network compliant to Stuttgart Neural Network Simulator (SNNS) format as to enhance the recognition performance and support multiple characters writing.

The test data has been collected from 3 anonymous testers. Each tester provide 10 set of characters and words. The results show that the characters recognition rate is 89.4%, incorrect rate is 10.6%. While, the words recognition rate is 87.7%, incorrect rate is 12.3%.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Department..... Computer Engineering.....Student's signature.....

Field of study.....Computer Science.....Advisor's signature.....

Academic year ...2005.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างดียิ่งของ ผศ.ดร.วิวัฒน์ วัฒนาวุฒิ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งได้สละเวลาให้คำแนะนำ ข้อเสนอแนะ ข้อคิดเห็น และแนวทางในการค้นคว้าในการวิจัยด้วยดีตลอดมา ผู้วิจัยขอกราบขอบพระคุณในความกรุณาจากอาจารย์เป็นอย่างสูง รวมถึงอาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ทุกท่าน ที่ประสิทธิประสาทวิชาความรู้ให้ผู้วิจัย

ขอขอบคุณ รศ. ดร.บุญเสริม กิจศิริกุล อ.ดร.อาทิตย์ ทองทักษ์ ผศ. ดร.ธาราทิพย์ สุวรรณศาสตร์ ที่กรุณาเสียสละเวลามาช่วยตรวจสอบ ดำเนินการสอบ และแก้ไขวิทยานิพนธ์ฉบับนี้ให้สมบูรณ์

ขอขอบคุณคุณปกรณ์ บุพศิริ และคุณนำชัย ยิ่งนวลจันทร์ ที่ให้ข้อมูลและคำแนะนำในการใช้โปรแกรมสอนข่ายงานประสาทของคุณกันตา กิติยานันท์

ขอขอบคุณเพื่อนๆ ที่ช่วยคัดตัวอักษรภาษาไทยเพื่อใช้ในการสอนและทดสอบข่ายงานประสาท ช่วยทดสอบโปรแกรม ให้กำลังใจ และคอยถามไถ่ถึงความคืบหน้าในการทำงานวิจัย และให้ความช่วยเหลือในด้านต่าง ๆ เป็นอย่างดี

ท้ายนี้ผู้วิจัยขอกราบขอบพระคุณบิดา มารดา และทุกคนในครอบครัวที่คอยสนับสนุนในทุกๆด้าน และให้กำลังใจแก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญภาพ ..... ฌ
สารบัญตาราง ..... ญ
บทที่
1.บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ของการวิจัย 2
1.3 ขอบเขตของการวิจัย 2
1.4 วิธีดำเนินการวิจัย 3
2.งานวิจัยและทฤษฎีที่เกี่ยวข้อง 4
2.1 งานวิจัยที่เกี่ยวข้อง 4
2.1.1 การปรับปรุงระบบการรู้จำลายมือเขียนแบบออนไลน์สำหรับตัวอักษรภาษาไทยโดยใช้ข่ายงานประสาทแบบแพร่กระจายย้อนกลับ 4
2.1.2 การพัฒนาเครื่องมือซอฟต์แวร์สำหรับระบบการรู้จำลายมือเขียนแบบออนไลน์โดยใช้ข่ายงานประสาทแบบแพร่กระจายย้อนกลับ 4
2.1.3 การรู้จำลายมือเขียนแบบออนไลน์สำหรับตัวอักษรภาษาไทยโดยใช้ข่ายงานประสาทแบบแพร่กระจายย้อนกลับบนวินโดว์ส 5
2.2 เทคนิคการจำแนกลักษณะของตัวอักษรภาษาไทย 6
2.2.1 การประมวลผลข้อมูลตัวอักษรเบื้องต้น 6
2.2.2. การหาลักษณะเด่นของตัวอักษร 12
2.3 ทฤษฎีข่ายงานประสาท 16
2.3.1 สารสนเทศที่ใช้ในข่ายงานประสาท 17
2.3.2 ขั้นตอนการเรียนรู้ของข่ายงานประสาท 18
2.3.3 ประเภทการเรียนรู้ของข่ายงานประสาท 18
2.3.4 การเรียนรู้แบบย้อนกลับ 19
3.การออกแบบและพัฒนาระบบ 26

สารบัญ (ต่อ)

หน้า

3.1 เทคนิคที่ใช้ในการจำแนกตัวอักษรภาษาไทย 26
3.2 การออกแบบระบบ 27
3.2.1 การออกแบบส่วนต่อประสานกับผู้ใช้ 27
3.2.2 การออกแบบส่วนของฟังก์ชันการรู้จำลายมือ 31
3.2.3 การออกแบบส่วนของฟังก์ชันข่ายงานประสาทแบบแพร่กระจายย้อนกลับ 42
3.3 แผนภาพคลาสและความสัมพันธ์ 42
3.4 สภาพแวดล้อมในการพัฒนาเครื่องมือ 48
4.การทดสอบการใช้งาน 49
4.1 การเตรียมการทดสอบ 49
4.2 วิธีการทดสอบระบบ 50
4.3 ผลการทดสอบระบบ 51
4.4 วิเคราะห์ผลการทดสอบระบบ 54
5.สรุปผลการวิจัยและข้อเสนอแนะ 58
5.1 สรุปผลการวิจัย 58
5.2 ปัญหาที่พบจากการวิจัย 59
5.3 ประโยชน์ที่ได้รับจากงานวิจัย 60
5.4 ข้อจำกัดของระบบ 60
5.4 ข้อเสนอแนะ 61
รายการอ้างอิง 62
ภาคผนวก 64
ภาคผนวก ก.ฟังก์ชันที่ใช้ในการรู้จำลายมือเขียนครั้งละหลายตัวอักษร 65
ภาคผนวก ข.รูปแบบของไฟล์โค้ดและไฟล์น้ำหนัก 67
ภาคผนวก ค.ตัวอย่างการเรียกใช้ไลบรารีสำหรับรู้จำลายมือภาษาไทย 71
ภาคผนวก ง.ตารางรายละเอียดตัวอักษรที่ระบบรู้จำผิดและผลการรู้จำผิด 73
ประวัติผู้เขียนวิทยานิพนธ์ 77

สารบัญภาพ

หน้า

รูปที่ 2.1 การทำนอร์มอลไลซ์ของตัวอักษร น 9
รูปที่ 2.2 มุมที่ใช้ในการหามุมแหลม 11
รูปที่ 2.3 ทิศทางของรหัสลูกโซ่แบบ 8 ทิศทาง 12
รูปที่ 2.4 จุดเด่นบนตัวอักษร น 13
รูปที่ 2.5 วงรอบของตัวอักษร น 13
รูปที่ 2.6 จุดตัดวงรอบที่สองของตัวอักษร น 14
รูปที่ 2.7 ลักษณะการตัดกันของเส้นตรง 15
รูปที่ 2.8 รหัสโซนของตัวอักษรแบบ 5 โซน 16
รูปที่ 2.9 โครงสร้างของข่ายงานประสาท 17
รูปที่ 2.10 หน่วยประมวลผล 20
รูปที่ 2.11 ขั้นตอนการแพร่เดินหน้า 20
รูปที่ 2.12 ขั้นตอนการแพร่ย้อนกลับ 23
รูปที่ 3.1 ส่วนประกอบหลักของระบบ 27
รูปที่ 3.2 แผนภาพยูเคสส่วนต่อประสานกับผู้ใช้ 28
รูปที่ 3.3 ส่วนประกอบของส่วนต่อประสานกับผู้ใช้ 28
รูปที่ 3.4 การส่งชุดตัวอักษรที่รู้จำได้ไปยังโปรแกรมไมโครซอฟท์เวิร์ด 30
รูปที่ 3.5 ส่วนประกอบของฟังก์ชันในการรู้จำลายมือ 32
รูปที่ 3.6 การหาขอบเขตของสโตรค 32
รูปที่ 3.7 การแบ่งกลุ่มตัวอักษรที่มี 2 สโตรค 33
รูปที่ 3.8 การเปรียบเทียบระยะห่างระหว่างขอบเขตของแต่ละสโตรค 35
รูปที่ 3.9 การทำงานของฟังก์ชันตรวจสอบตัวอักษรก่อนหน้า 41
รูปที่ 3.10 แผนภาพความสัมพันธ์ระหว่างคลาสไลบรารีสำหรับรู้จำลายมือภาษาไทย 43
รูปที่ 3.11 แผนภาพความสัมพันธ์ระหว่างคลาสไลบรารีข่ายงานประสาทแบบแพร่กระจาย
ย้อนกลับ 46

สารบัญตาราง

หน้า

ตารางที่ 3.1 สรุปข้อมูลเข้ารหัสที่ใช้ส่งให้ข่ายงานประสาทตามลำดับที่ของอาร์เรย์..... 39
ตารางที่ 3.2 เมท็อดและหน้าที่ของเมท็อดภายในคลาส WrittingPoints 43
ตารางที่ 3.3 เมท็อดและหน้าที่ของเมท็อดภายในคลาส TStrokes 44
ตารางที่ 3.4 เมท็อดและหน้าที่ของเมท็อดภายในคลาส TRecognize..... 45
ตารางที่ 3.5 เมท็อดและหน้าที่ของเมท็อดภายในคลาส Neuron 45
ตารางที่ 3.6 เมท็อดและหน้าที่ของเมท็อดภายในคลาส Neurons 45
ตารางที่ 3.7 เมท็อดและหน้าที่ของเมท็อดภายในคลาส Connectors 46
ตารางที่ 3.8 เมท็อดและหน้าที่ของเมท็อดภายในคลาส Layer..... 47
ตารางที่ 3.9 เมท็อดและหน้าที่ของเมท็อดภายในคลาส NeuralNet..... 47
ตารางที่ 4.1 สรุปผลการรู้จำลายมือครั้งละตัวอักษร..... 51
ตารางที่ 4.2 รายละเอียดผลการรู้จำลายมือครั้งละตัวอักษร..... 52
ตารางที่ 4.3 ผลการรู้จำลายมือครั้งละหลายตัวอักษร..... 55

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันนี้เครื่องคอมพิวเตอร์ประเภทพกพา เช่น อุปกรณ์ปาล์ม ไพลอท (Palm pilot) อุปกรณ์แฮนด์เฮล์ด พีซี (Handheld PC) ได้รับความนิยมเพิ่มมากขึ้น เนื่องจากมีขนาดเล็กทำให้สามารถพกพาไปได้ทุกที่ และสามารถใช้งานได้สะดวกสบาย แต่ด้วยขนาดที่เล็กจึงมีขีดจำกัดหลายๆอย่างที่ทำให้ไม่สามารถทำงานได้เท่าเทียมกับคอมพิวเตอร์ตั้งโต๊ะ บริษัทไมโครซอฟท์จึงได้ร่วมกับผู้ผลิตอุปกรณ์คอมพิวเตอร์หลายรายด้วยกัน เพื่อผลิตอุปกรณ์คอมพิวเตอร์แบบใหม่ที่รวมคุณสมบัติที่ดีๆของ คอมพิวเตอร์ประเภทพกพาและคอมพิวเตอร์ตั้งโต๊ะเข้าไว้ด้วยกัน จึงได้ออกมาเป็นแท็บเล็ตพีซี ซึ่งเป็นเครื่องคอมพิวเตอร์ที่ออกแบบมาเพื่อช่วยให้ผู้ใช้สามารถใช้งานเครื่องคอมพิวเตอร์ได้อย่างเป็นธรรมชาติด้วยการใช้งานในรูปแบบของสมุดโน้ต ไม่ว่าจะเป็นการจดบันทึกข้อความการดูข้อมูล โดยมีคุณสมบัติเทียบเท่ากับคอมพิวเตอร์ตั้งโต๊ะทุกประการ แต่มีขนาดเล็ก น้ำหนักเบา เหมาะแก่การพกพา และเพิ่มคุณสมบัติในการรับข้อมูลเข้าโดยอาศัยปากกาหรือสไตลัส (Stylus) เขียนตัวอักษรและสัญลักษณ์ต่างๆ ลงบนหน้าจอ ซึ่งการที่เครื่องคอมพิวเตอร์จะสามารถเข้าใจถึงตัวอักษรหรือสัญลักษณ์ที่ถูกเขียนลงบนหน้าจอได้นั้นก็ต้องอาศัยโปรแกรมการรู้จำตัวอักษรลายมือเขียน เพื่อที่จะสามารถบอกได้ว่าตัวอักษรหรือสัญลักษณ์ที่เขียนนั้นเป็นตัวอะไร ปัจจุบันนี้มีผู้ให้ความสนใจ ทำการวิจัยและพัฒนาโปรแกรมประเภทนี้อยู่เป็นจำนวนมาก โดยการใช้เทคนิคและวิธีการที่แตกต่างกันออกไป หนึ่งในวิธีที่ได้รับความนิยมอย่างแพร่หลายก็คือ ระบบการรู้จำตัวอักษรลายมือเขียน โดยใช้ข่ายงานประสาท

ในปี พ.ศ. 2541 อภิชาต สัจจพงษ์ ได้เสนอการรู้จำลายมือเขียนภาษาไทยแบบออนไลน์โดยใช้ข่ายงานประสาท [1] ซึ่งเทคนิคที่นำมาใช้สามารถให้อัตราการรู้จำมีความถูกต้องร้อยละ 83.43 ต่อมา โชติ ศิริวงศ์วิเชียร และปกิตต์ นิธิวิบูลย์ ได้พัฒนาโปรแกรมสำหรับการเข้ารหัสตัวอักษรลายมือเขียนภาษาไทย เพื่อระบบการรู้จำโดยใช้ข่ายงานประสาท [2] เทคนิคที่นำมาใช้สามารถให้อัตราการรู้จำมีความถูกต้องร้อยละ 81.9 และในพ.ศ. 2545 ปกรณ์ บุพศิริ ได้ทำการปรับปรุงระบบการรู้จำลายมือเขียนแบบออนไลน์สำหรับตัวอักษรภาษาไทย โดยใช้ข่ายงานประสาท[3] ให้อัตราการรู้จำมีความถูกต้องร้อยละ 96.62 ซึ่งจะเห็นได้ว่าอัตราการรู้จำได้ถูกปรับปรุงให้มีความถูกต้องมากขึ้น

จากที่มีผู้วิจัยการรู้จำตัวอักษรลายมือเขียนภาษาไทย พร้อมทั้งได้ปรับปรุงให้อัตราการรู้จำดียิ่งขึ้น ประกอบกับการที่ในปัจจุบันการรู้จำลายมือบนแท็บเล็ตพีซี รองรับเฉพาะภาษาอังกฤษ เยอรมัน ฝรั่งเศส ญี่ปุ่น เกาหลี จีน เท่านั้น ดังนั้นในงานวิจัยนี้ จึงมีแนวคิดที่จะนำผลการวิจัยการรู้จำตัวอักษรลายมือเขียนภาษาไทยมาพัฒนาระบบการรู้จำลายมือออนไลน์สำหรับตัวอักษรภาษาไทยบนแท็บเล็ตพีซี โดยใช้เทคนิคการรู้จำจากงานวิจัยของนายปกรณ์ บุพศิริ ซึ่งให้ผลอัตราการรู้จำที่มีความถูกต้องร้อยละ 96.62

1.2 วัตถุประสงค์ของการวิจัย

เพื่อพัฒนาระบบการรู้จำลายมือออนไลน์สำหรับตัวอักษรภาษาไทยบนแท็บเล็ตพีซี

1.3 ขอบเขตของการวิจัย

1) ระบบที่พัฒนาขึ้นสามารถรู้จำตัวอักษรลายมือเขียนภาษาไทย ได้ทีละ ตัวอักษร จาก 67 ตัวอักษร ซึ่งประกอบด้วย พยัญชนะ 42 ตัว สระ 17 ตัว วรรณยุกต์ 4 ตัว และอักขระพิเศษ 4 ตัว ที่เลือกไว้ คือ ก ข ค ฆ ง จ ฉ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย ร ล ว ศ ษ ส ห ฬ อ ฮ ะ ั ิ ี ึ ื ุ ู เ แ โ ไ ใ ำ ั ฤ ฦ ็ ่ ้ ๊ ๋ ฯ ๆ
2) เป็นการพัฒนาระบบการรู้จำลายมือออนไลน์สำหรับตัวอักษรภาษาไทยบนแท็บเล็ตพีซี โดยใช้หลักการในการพัฒนาจากผลงานการวิจัยระบบการรู้จำลายมือเขียนภาษาไทยแบบออนไลน์โดยใช้ข่ายงานประสาทแบบแพร่กระจายย้อนกลับของนายปกรณ์ บุพศิริ [3]
3) ใช้เครื่องอ่านพิกัดแบบปากกาเป็นอุปกรณ์นำข้อมูลเข้า
4) ลักษณะการเขียนตัวอักษรภาษาไทยต้องเป็นไปตามรูปแบบจาก แบบฝึกหัดคัดไทยของสมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น) [4]
5) จำนวนสโตรคสูงสุดที่รับได้ไม่เกิน 2 สโตรค สำหรับตัวอักษรแต่ละตัว โดยมีระยะเวลาหน่วงระหว่างสโตรคที่ 1 กับสโตรคที่ 2 ไม่เกิน 1 วินาที ถ้าผู้ใช้เครื่องมือป้อนข้อมูลในสโตรคถัดไปช้ากว่าเวลาที่กำหนดระบบจะถือว่าข้อมูลลายมือเขียนนี้ได้สิ้นสุดการป้อนข้อมูลแล้ว
6) ใช้ลายมืออย่างน้อย 3 คนในการสอนข่ายงานประสาทและในการทดสอบ
7) ระบบที่พัฒนาสามารถรองรับการรู้จำตัวอักษรลายมือภาษาอังกฤษ ได้ตามการรู้จำตัวอักษรลายมือภาษาอังกฤษของไมโครซอฟท์วินโดว์เอ็กซ์พีสำหรับแท็บเล็ตพีซี (Microsoft Windows XP Tablet PC Recognizer Pack) [12]

8) ตัวอักษรที่รู้จำได้ สามารถส่งไปให้กับโปรแกรมไมโครซอฟท์เวิร์ด (Microsoft Word) และไมโครซอฟท์เอ็กซเซล (Microsoft Excel) ได้

1.4 วิธีดำเนินการวิจัย

1) ศึกษาทฤษฎีต่าง ๆ ของการจำแนกลักษณะของตัวอักษรไทยที่เขียนด้วยลายมือที่ใช้ในงานวิจัยของนายปกรณ์ บุพศิริ
2) ศึกษาความรู้ทางด้านข่ายงานประสาท
3) ศึกษาการใช้งานฟังก์ชันสำหรับการรู้จำลายมือภาษาอังกฤษของไมโครซอฟท์วินโดว์เอ็กซ์พีสำหรับแท็บเล็ตพีซี (Microsoft Windows XP Tablet PC Recognizer Pack) [12]
4) พัฒนาระบบการรู้จำลายมือออนไลน์สำหรับตัวอักษรภาษาไทยบนแท็บเล็ตพีซี
5) ทดสอบการใช้งานและวิเคราะห์ผล
6) สรุปผลการวิจัยและจัดทำเอกสาร

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

งานวิจัยและทฤษฎีที่เกี่ยวข้อง

2.1 งานวิจัยที่เกี่ยวข้อง

2.1.1 การปรับปรุงระบบการรู้จำลายมือเขียนแบบออนไลน์สำหรับตัวอักษรภาษาไทยโดยใช้ข่ายงานประสาทแบบแพร่กระจายย้อนกลับ(2545) [3] โดย ปกรณ์ บุพศิริ

เสนอการปรับปรุงระบบการรู้จำลายมือเขียนแบบออนไลน์ สำหรับตัวอักษรภาษาไทย โดยใช้ข่ายงานประสาทแบบแพร่กระจายย้อนกลับ ซึ่งประกอบด้วยการปรับปรุง 3 ส่วน คือการประมวลผลตัวอักษรเบื้องต้น การหาลักษณะเด่นของตัวอักษร และการเข้ารหัส โดยมีเทคนิคสำคัญที่ใช้คือ การทำนอร์มอลไลซ์ โดยใช้วิธีการปรับให้อัตราส่วนของตัวอักษรคงที่ ซึ่งอัตราส่วนที่ใช้เกิดจากการคำนวณหาอัตราส่วนของค่าใหม่เทียบกับค่าเดิม โดยใช้อัตราส่วนของค่าที่มากกว่าในการเปรียบเทียบระหว่างความสูง กับความกว้างของตัวอักษร การหาวงรอบของตัวอักษร โดยใช้การหาจุดตัดและการเปลี่ยนแปลงของทิศทางรหัสลูกโซ่ การใช้เขตย่อยแบบ 5 เขต การเพิ่มข้อมูลในการเข้ารหัสโหนด (Node) โดยเพิ่มรหัสจำนวนสโตรค และจำนวนโหนดวงรอบของตัวอักษร

จากการปรับปรุงดังกล่าวได้ข้อมูลที่ใช้ในการสอนข่ายงานประสาทประกอบด้วย จำนวนของสโตรค จำนวนรหัสเขตย่อย และทิศทางการวนของวงรอบ เปอร์เซ็นต์ของจุดในแต่ละเขตย่อย รหัสลูกโซ่และรหัสเขตย่อยของจุดเด่น รหัสเขตย่อยของแต่ละจุด และมีการเติมค่าตำแหน่งที่ว่างด้วยจำนวนสโตรคและจำนวนวงรอบ โดยใช้จำนวนชั้นอินพุต ชั้นแอบแฝง และชั้นเอาต์พุตอย่างละ 1 ชั้น จำนวนโหนด 205, 700 และ 67 โหนด ตามลำดับ จากการทดสอบกับตัวอักษร 2,010 ตัว ซึ่งได้จากลายมือเขียนของผู้วิจัยเพียงคนเดียว มีรูปแบบตัวอักษร 67 แบบ ให้ผลที่ได้มีความถูกต้อง 96.62 เปอร์เซ็นต์ รู้จำผิด 1.79 เปอร์เซ็นต์ และรู้จำไม่ได้ 1.59 เปอร์เซ็นต์

เทคนิคในการแยกลายมือเขียนภาษาไทยที่ใช้ในงานวิจัยของนายปกรณ์ บุพศิริ นี้จะถูกนำมาใช้เป็นแนวทางหลักในการพัฒนาระบบสำหรับงานวิจัยนี้

2.1.2 การพัฒนาเครื่องมือซอฟต์แวร์สำหรับระบบการรู้จำลายมือเขียนแบบออนไลน์โดยใช้ข่ายงานประสาทแบบแพร่กระจายย้อนกลับ (2544) [13] โดย กันตา กิติยานันท์

พัฒนาเครื่องมือซอฟต์แวร์สำหรับระบบการรู้จำลายมือเขียนแบบออนไลน์ โดยใช้ข่ายงานประสาทแบบแพร่กระจายย้อนกลับ โดยเครื่องมือนี้สามารถรับข้อมูลลายมือเขียน กำหนดวิธีการ

ประมวลผลข้อมูลตัวอักษรเบื้องต้น และค่าพารามิเตอร์ต่างๆที่สำคัญ ทำการประมวลผล และส่งข้อมูลที่ผ่านการประมวลผลแล้วไปยังโปรแกรมข่ายงานประสาทเพื่อทำการสอนและทดสอบข่ายงานประสาท โดยใช้โปรแกรมข่ายงานประสาทของมหาวิทยาลัยสตุทการ์ท (SNNS: Stuttgart Neural Network Simulator) [10] หลังจากทำการสอนและทดสอบข่ายงานประสาท จะได้รับข้อมูลข่ายงานประสาทและผลการทดสอบกลับมาเพื่อรายงานผล และจัดเก็บเพื่อการใช้งานต่อไป ซึ่งจะทำให้ภาระในการเขียนโปรแกรมลดลง เครื่องมือนี้ยังมีความสามารถคือ สามารถจัดเก็บข้อมูลตัวอักษรไว้ในฐานข้อมูลซึ่งสามารถรองรับลายมือเขียนจากผู้เขียนหลายๆคน สามารถทำการสร้างและแก้ไขภาษาไพธอนสคริปต์ (Python script) เพื่อใช้ในการกำหนดวิธีการประมวลผลลายมือเขียนโดยไม่มีผลกระทบกับส่วนอื่นๆ และได้จัดเตรียมส่วนต่อประสานกับโปรแกรมข่ายงานประสาทแบบแพร่กระจายย้อนกลับเพื่อให้ผู้ใช้สามารถทำการกำหนดพารามิเตอร์ต่างๆ ได้อย่างสะดวกและง่ายยิ่งขึ้น

เครื่องมือจากงานวิจัยของคุณกันตา กิติยานันท์ นี้จะถูกนำมาใช้ในการสอนและทดสอบข่ายงานประสาท เพื่อให้สามารถรู้จำลายมือเขียนจากผู้ใช้หลายคน

2.1.3 การรู้จำลายมือเขียนแบบออนไลน์สำหรับตัวอักษรภาษาไทยโดยใช้ข่ายงานประสาทแบบแพร่กระจายย้อนกลับบนวินโดว์ส (2545)[14] โดย Sutat Sac-Tang และ Ithipan Methaste

ทำการพัฒนาระบบการรู้จำลายมือเขียนแบบออนไลน์สำหรับตัวอักษรภาษาไทยโดยใช้ข่ายงานประสาทแบบแพร่ย้อนกลับบนวินโดว์ส ใช้คุณสมบัติท้องถิ่นและคุณสมบัติทั่วไปสำหรับการหาลักษณะเด่นของตัวอักษรไทย การประมวลผลแบ่งเป็นการประมวลผลตัวอักษรเบื้องต้นและการหาลักษณะเด่นของตัวอักษร ขั้นตอนในการประมวลผลตัวอักษรเบื้องต้น คือการลบจุดจรดปากกาและจุดสิ้นสุดของแต่ละสโตรค การลดจุดพิกัดที่ไม่จำเป็น ขั้นตอนในการหาลักษณะเด่นของตัวอักษร ใช้คุณสมบัติท้องถิ่น (Local Features) คือ ความยาวของสโตรค ทิศทางของสโตรค จุดเริ่มต้น จุดสิ้นสุด มุมที่เริ่มแรกของสโตรค มุมที่สุดท้ายของสโตรค และ คุณสมบัติทั่วไป (Global Features) คือแบ่งตัวอักษรเป็น 3 ส่วนในแนวตั้ง และ 3 ส่วนในแนวนอนเท่าๆกันตามลำดับ และหาอัตราส่วนระหว่างจุดสีดำในแต่ละส่วน กับจำนวนจุดสีดำทั้งหมด

การทดสอบใช้ข้อมูลลายมือเขียนของผู้หญิง 4 คน กับผู้ชาย 6 คน ซึ่งแต่ละคนเขียนตัวอักษร 75 ตัว(พยัญชนะ 42 ตัว สระ 23 ตัว และตัวเลขอารบิก 10 ตัว) จำนวน 10 ครั้ง แบ่ง

ข้อมูลเป็น 2 ส่วนคือชุดที่ 1-6 สำหรับสอนข่ายงานประสาท ส่วนชุดที่ 7-10 สำหรับทดสอบ ซึ่งผลของการรู้จำตัวอักษรมีความถูกต้อง 91.74 เปอร์เซ็นต์

2.2 เทคนิคการจำแนกลักษณะของตัวอักษรภาษาไทย

2.2.1 การประมวลผลข้อมูลตัวอักษรเบื้องต้น (Preprocessing)

คือการเตรียมข้อมูลลายมือเขียนตัวอักษรให้อยู่ในรูปแบบที่มีขนาดเท่ากัน และปรับปรุงข้อมูลที่ไม่ถูกต้องให้อยู่ในรูปแบบที่เหมาะสมและง่ายต่อการประมวลผลในขั้นตอนต่อไป ซึ่งข้อมูลตัวอักษรดังกล่าวคือข้อมูลที่ได้จากการจรดปากกาสัมผัสกับพื้นที่เขียน โดยการจรดปากกาสัมผัสกับพื้นที่เขียนจนกระทั่งยกปากกาขึ้นเรียกว่า หนึ่งสโตรค ดังนั้นข้อมูลลายมือเขียนตัวอักษร จะประกอบด้วยสโตรคตั้งแต่หนึ่งสโตรคขึ้นไป ซึ่งสามารถอธิบายหนึ่งสโตรคจากลำดับของจุดบนระนาบ XY ด้วยสมการที่ 2.1

$$S = p_1 p_2 ... p_L \qquad \text{...(2.1)}$$

เมื่อ

S คือ สโตรคในการเขียนหนึ่งครั้ง

p_i คือ จุดภายในสโตรค โดยแสดงด้วยคู่ลำดับ (x_i, y_i) เมื่อ $1 \le i \le L$

p_1 คือ จุดแรกที่ปากกาเริ่มสัมผัสกับพื้นที่เขียน

p_L คือ จุดสุดท้ายก่อนที่จะยกปากกาออกจากพื้นที่เขียน

L คือ จำนวนของจุดในการเขียนหนึ่งสโตรค

ข้อมูลลายมือเขียนในหนึ่งตัวอักษร สามารถอธิบายได้ดังสมการที่ 2.2

$$C = S_1 S_2 ... S_N \qquad \text{...(2.2)}$$

เมื่อ

C คือ ลำดับสโตรคของตัวอักษร

S_i คือ สโตรคที่ i ในการเขียน

N คือ จำนวนของสโตรคในการเขียนตัวอักษรนั้น ๆ

2.2.1.1. การทำนอร์มอลไลซ์ (Normalization)

คือการปรับขนาดตัวอักษรลายมือเขียนให้มีขนาดมาตรฐานเดียวกัน และย้ายกรอบของตัวอักษรไปยังจุดกำเนิด(Origin) เพื่อให้ง่ายต่อการประมวลผล เนื่องจากขนาดของตัวอักษรจากลายมือเขียนในแต่ละครั้งมีขนาดและตำแหน่งที่ไม่แน่นอน การทำนอร์มอลไลซ์ จะทำให้ไม่เกิด

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ความแตกต่างของขนาด และตำแหน่งการเขียนในแต่ละครั้ง โดยวิธีการทำนอร์มอลไลซ์ มีอยู่ด้วยกันหลายวิธีดังนี้

2.2.1.1.1 การทำนอร์มอลไลซ์โดยกำหนดความกว้างและความสูงที่ต้องการ [6] ซึ่งเป็นการปรับให้ตัวอักษรมีขนาดความกว้างและความสูงตามที่ต้องการ โดยสามารถทำได้ดังสมการที่ 2.3

$$x_i = \frac{x_i^o - x_{\min}}{x_{\max} - x_{\min}} W \qquad \ldots(2.3)$$

$$y_i = \frac{y_i^o - y_{\min}}{y_{\max} - y_{\min}} H \qquad \ldots(2.4)$$

เมื่อ

(x_i^o, y_i^o) คือ จุดที่เป็นข้อมูลจริง

(x_i, y_i) คือ จุดที่เกิดจากการแปลงค่าแล้ว

$x_{\min}$ คือ พิกัด x ที่มีค่าน้อยที่สุดในข้อมูลจริง

$y_{\min}$ คือ พิกัด y ที่มีค่าน้อยที่สุดในข้อมูลจริง

$x_{\max}$ คือ พิกัด x ที่มีค่ามากที่สุดในข้อมูลจริง

$y_{\max}$ คือ พิกัด y ที่มีค่ามากที่สุดในข้อมูลจริง

W และ H คือ ความกว้างและความสูงของตัวอักษรที่ต้องการหลังจากปรับแล้วตามลำดับ

2.2.1.1.2 การทำนอร์มอลไลซ์โดยกำหนดความสูงที่ต้องการ [14] ซึ่งเป็นการปรับให้ตัวอักษรมีขนาดความสูงตามที่ต้องการ และมีอัตราส่วนคงที่ โดยสามารถทำได้ดังสมการที่ 2.5

$$r = \frac{h_n}{h_o} \qquad \ldots(2.5)$$

เมื่อ

r คือ อัตราส่วนของความสูงใหม่เทียบกับความสูงเดิม

h_n คือ ความสูงใหม่ของตัวอักษรลายมือเขียน

h_o คือ ความสูงเดิมของตัวอักษรลายมือเขียน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

และนำอัตราส่วนที่ได้จากสมการที่ 3.5 มาคำนวณหาค่าพิกัดใหม่ โดยการนำอัตราส่วนนี้คูณกับค่าพิกัดเดิมทั้งค่า x และ y ของทุกจุด

2.2.1.1.3 การทำนอร์มอลไลซ์โดยใช้วิธีการปรับให้อัตราส่วนของตัวอักษรคงที่[3] ซึ่งอัตราส่วนที่ใช้ ได้จากการคำนวณหาอัตราส่วนของความสูงใหม่เทียบกับความสูงเดิม หรือความกว้างใหม่เทียบกับความกว้างเดิม โดยขึ้นอยู่กับลักษณะโครงสร้างของตัวอักษรว่ามีความสูงหรือความกว้างมากกว่ากัน ถ้ามีความสูงมากกว่าความกว้าง จะใช้อัตราส่วนของความสูงในการทำนอร์มอลไลซ์ ถ้ามีความกว้างมากกว่าความสูง จะใช้อัตราส่วนของความกว้างในการทำนอร์มอลไลซ์แทน โดยสามารถทำได้ดังสมการที่ 2.6 และ 2.7

$$r = \frac{h_n}{h_o} \qquad \text{เมื่อ} \quad h_o > w_o \qquad \ldots(2.6)$$

$$r = \frac{w_n}{w_o} \qquad \text{เมื่อ} \quad w_o > h_o \qquad \ldots(2.7)$$

เมื่อ

r คือ อัตราส่วนของตัวอักษร

h_n คือ ความสูงใหม่ของตัวอักษรลายมือเขียน

h_o คือ ความสูงเดิมของตัวอักษรลายมือเขียน

w_n คือ ความกว้างใหม่ของตัวอักษรลายมือเขียน

w_o คือ ความกว้างเดิมของตัวอักษรลายมือเขียน

$$x_i = \frac{x_i^o - x_{\min}}{x_{\max} - x_{\min}} r \qquad \ldots(2.8)$$

$$y_i = \frac{y_i^o - y_{\min}}{y_{\max} - y_{\min}} r \qquad \ldots(2.9)$$

เมื่อ

(x_i^o, y_i^o) คือ จุดที่เป็นข้อมูลจริง

(x_i, y_i) คือ จุดที่เกิดจากการแปลงค่าแล้ว

$x_{\min}$ คือพิกัด x ที่มีค่าน้อยที่สุดในข้อมูลจริงที่ได้จากการเขียน

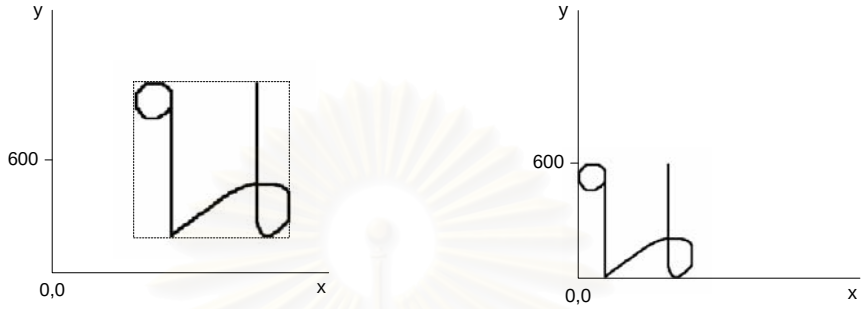
$y_{\min}$ คือ พิกัด y ที่มีค่าน้อยที่สุดในข้อมูลจริงที่ได้จากการเขียน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

x_{max} คือ พิกัด x ที่มีค่ามากที่สุดในข้อมูลจริงที่ได้จากการเขียน

y_{max} คือ พิกัด y ที่มีค่ามากที่สุดในข้อมูลจริงที่ได้จากการเขียน

ตัวอย่างการทำนอร์มอลไลซ์ของตัวอักษร น แสดงดังรูปที่ 2.1



(ก) ก่อนทำนอร์มอลไลซ์ (ข) หลังทำนอร์มอลไลซ์

รูปที่ 2.1 การทำนอร์มอลไลซ์ของตัวอักษร น [3]

2.2.1.2. การปรับแต่งและกำจัดสัญญาณรบกวน [7]

เป็นการปรับแต่งและกำจัดสัญญาณรบกวนของข้อมูลลายมือเขียน เพื่อให้ได้ข้อมูลที่อยู่ในรูปแบบที่เหมาะสมและสามารถนำไปทำการหาลักษณะเด่นได้ง่าย เนื่องจากข้อมูลลายมือเขียนที่ได้รับอาจมีข้อมูลที่ไม่สมบูรณ์ หรือสัญญารบกวนซึ่งจะส่งผลให้เกิดปัญหาในการประมวลผลได้ โดยมีขั้นตอนตามลำดับดังนี้

2.2.1.2.1 การกำจัดจุดผิดพลาดที่เกิดจากการยกปากกาโดยไม่ได้ตั้งใจ (Removal of accidental pen-lifts) ซึ่งอาจเกิดจากการเปลี่ยนแปลงทิศทางการเขียนภายในสโตรค ดังนั้นการพิจารณาจุดผิดพลาดดังกล่าว จะพิจารณาจากจุดที่มีระยะห่างระหว่างจุดยกปากกาและจุดจรดปากกาน้อยกว่า 0.5 มิลลิเมตร และมีระยะเวลาในการเขียน 2 จุดดังกล่าวต่างกันน้อยกว่า 100 มิลลิวินาที

2.2.1.2.2 การปรับเส้นให้ราบเรียบขึ้น (Modified gear backlash smoothing) เป็นการปรับระยะห่างของจุด 2 จุด ระหว่างจุดก่อนหน้ากับจุดปัจจุบัน โดยนำมาเปรียบเทียบกับค่า k ที่กำหนด เพื่อย้ายจุดปัจจุบันจากตำแหน่งเดิมให้เข้ามาอยู่ใกล้จุดก่อนหน้าเป็นระยะไม่เกิน k หน่วย โดยมีทิศทางในการเปลี่ยนแปลงไปในแนวตั้ง แนวนอน เพื่อทำให้จุดทั้งสองไม่ห่างกันจนเกินไป ด้วยวิธีการดังนี้

ทำขั้นตอนต่อไปนี้สำหรับแต่ละจุด จนครบทุกจุด :

1) ปรับในแนวแกน X :

ถ้าพิกัด X ของจุดก่อนหน้า $\leq$ พิกัด X ของจุดปัจจุบัน – K

ให้พิกัด X ของจุดปัจจุบัน = พิกัด X ของจุดปัจจุบัน – K

มิฉะนั้น ถ้าพิกัด X ของจุดก่อนหน้า $\geq$ พิกัด X ของจุดปัจจุบัน + K

ให้พิกัด X ของจุดปัจจุบัน = พิกัด X ของจุดปัจจุบัน + K

มิฉะนั้น ให้พิกัด X ของจุดปัจจุบัน = พิกัด X ของจุดก่อนหน้า

2) ปรับในแนวแกน Y :

ถ้าพิกัด Y ของจุดก่อนหน้า $\leq$ พิกัด Y ของจุดปัจจุบัน – K

ให้พิกัด Y ของจุดปัจจุบัน = พิกัด X ของจุดปัจจุบัน – K

มิฉะนั้น ถ้าพิกัด Y ของจุดก่อนหน้า $\geq$ พิกัด Y ของจุดปัจจุบัน + K

ให้พิกัด Y ของจุดปัจจุบัน = พิกัด Y ของจุดปัจจุบัน + K

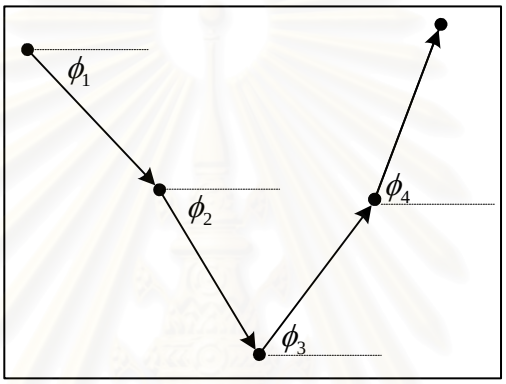
มิฉะนั้น ให้พิกัด Y ของจุดปัจจุบัน = พิกัด Y ของจุดก่อนหน้า

2.2.1.2.3 การกรองด้วยระยะห่างที่น้อยที่สุด (Minimum distance filtering) เป็นการกำจัดจุดที่มีความซ้ำซ้อนและไม่มีความสำคัญออก โดยกำจัดจุดที่อยู่ห่างจากจุดก่อนหน้าต่ำกว่าค่าที่กำหนด D ยกเว้นจุดที่เป็นจุดเริ่มต้นและสิ้นสุดของสโตรค การกำหนดค่า D ต้องเป็นค่าที่เหมาะสม เพราะถ้าค่า D มีค่ามากเกินไป จะทำให้จุดที่มีความสำคัญในการแสดงลักษณะเด่นถูกลบไปด้วย แต่ถ้าค่า D มีค่าน้อยเกินไป จะทำให้การกรองไม่มีผล โดยในขั้นตอนนี้จะใช้ค่า D ที่มีค่าน้อย เพื่อเป็นการกรองอย่างหยาบๆ

2.2.1.2.4 การทำให้เรียบโดยการเฉลี่ยให้เป็นแนวเส้นตรง (Straight line average smoothing) เป็นการทำให้เส้นมีความเรียบมากขึ้น โดยการเลื่อนจุดปัจจุบัน ให้มาอยู่ในแนวเส้นตรงที่ลากระหว่างจุดก่อนหน้าและจุดถัดไป ทำทีละจุดไล่ไปจนสุดสโตรค ขั้นตอนนี้อาจทำให้จุดที่เป็นมุมลดการหักมุมลงไปบ้าง แต่กรณีที่ใช้กับข้อมูลที่มีความละเอียดสูง จะลดความไม่สม่ำเสมอของจุดที่อยู่ใกล้เคียงลงได้

2.2.1.2.5 การกรองด้วยระยะห่างที่น้อยที่สุด ในข้อ 2.2.1.2.3 อีกรอบ โดยใช้ค่า D ที่มากขึ้น เพื่อลดจุดที่ไม่มีความสำคัญออกมากขึ้น

2.2.1.2.6 การทำให้เรียบโดยการเฉลี่ยให้เป็นแนวเส้นตรงโดยมีมุมเข้ามาร่วมในการพิจารณา (Straight line average smoothing with angle constraint) จะทำการทำให้เรียบเฉพาะจุดที่ไม่ใช่จุดที่มีการหักมุม เนื่องจากจุดที่มีการหักมุมมักจะเป็นจุดสำคัญในการหาลักษณะเด่นของตัวอักษร ดังนั้นการพิจารณาว่าจุดใดควรได้รับการปรับ พิจารณาจากจุดที่มีค่าเปลี่ยนแปลงมุม (ϕ_A) และค่าขีดแบ่ง(ϕ_T) ที่กำหนด โดยคำนวณจากสมการที่ 2.10



รูปที่ 2.2 มุมที่ใช้ในการหามุมแหลม [7]

$$\phi_A = | \phi_1 + \phi_2 - \phi_3 - \phi_4 | \quad \ldots(2.10)$$

เมื่อ

ϕ_A คือ ค่ามุมของจุดที่กำลังพิจารณา

ϕ_T คือ ค่ามุมแหลมที่กำหนด

ϕ_1 คือ ค่ามุมของจุดก่อนหน้าจุดที่พิจารณา 1 จุด

ϕ_2 คือ ค่ามุมของจุดก่อนจุดที่พิจารณา 2 จุด

ϕ_3 คือ ค่ามุมของจุดถัดจากจุดที่พิจารณา 1 จุด

ϕ_4 คือ ค่ามุมของจุดถัดจากจุดที่พิจารณา 2 จุด

ถ้า $\phi_A < \phi_T$ จุดนั้นจะถูกทำตามขั้นตอนที่ 2.2.1.2.4

2.2.1.2.7 การกำจัดเส้นที่เกิดจากการวางปากกาหรือยกปากกา (Serif removal) เป็นขั้นตอนในการกำจัดจุดส่วนเกินที่เกิดจากการเปลี่ยนทิศทางของปากกา ในช่วงที่จรดปากกา

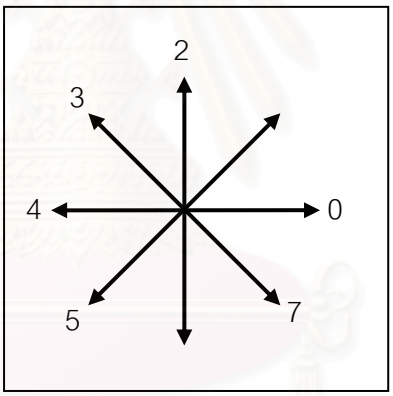
และยกปากกา โดยพิจารณาจากจุดที่อยู่ในระยะ 1 มิลลิเมตรจากจุดเริ่มต้นและจุดสุดท้ายของสโตรคที่มีมุมหักมากกว่า 60 องศา

2.2.2. **การหาลักษณะเด่นของตัวอักษร** (Feature extraction)

เป็นการนำข้อมูลลายมือเขียนที่ผ่านการประมวลผลตัวอักษรเบื้องต้นมาหาลักษณะเด่นของตัวอักษรแต่ละตัว ซึ่งเป็นการแบ่งแยกความแตกต่างของลายมือเขียนของตัวอักษรแต่ละตัว เพื่อนำข้อมูลดังกล่าวไปเข้ารหัส เพื่อนำไปสอนข่ายงานประสาท โดยมีวิธีการดังต่อไปนี้

2.2.2.1 รหัสลูกโซ่ (Chain Code) [8]

รหัสลูกโซ่นี้ใช้ในการแสดงทิศทางการเชื่อมต่อระหว่างจุด โดยแบ่งทิศทางออกเป็น 8 ทิศทาง ซึ่งแทนด้วยรหัสตัวเลข ดังรูปที่ 2.3 และใช้ตัวเลขทิศทางของรหัสลูกโซ่นี้แทนข้อมูลของตัวอักษร



รูปที่ 2.3 ทิศทางของรหัสลูกโซ่แบบ 8 ทิศทาง [8]

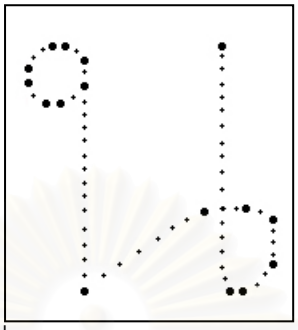
2.2.2.2 จุดเด่น (Dominant Point) [9]

จุดเด่นในสโตรค ประกอบด้วย 3 จุด ได้แก่ จุดแรกในสโตรค คือจุดแรกที่จรดปากกา จุดสุดท้ายในสโตรค คือจุดที่ยกปลายปากกาขึ้น จุดที่อยู่บนส่วนโค้งที่มีความโค้งมากที่สุด (Extrema of curvature) โดยการสังเกตทิศทางรหัสลูกโซ่ของจุดก่อนและหลังจุดที่กำลังพิจารณา 1 จุด จะต้องไม่มีทิศทางเดียวกัน ดังรูปที่ 2.4

2.2.2.3 จำนวนสโตรค[6]

เนื่องจากการเขียนตัวอักษรต่างๆอาจมีการใช้จำนวนสโตรคที่ต่างกันออกไป ดังนั้นจำนวนสโตรคจึงสามารถใช้แสดงลักษณะเด่นได้ การหาจำนวนสโตรคทำได้โดยการนับจำนวนจุดเริ่มต้น

หรือจำนวนจุดสุดท้ายอย่างใดอย่างหนึ่ง ซึ่งในภาษาไทยตัวอักษรลายมือเขียนหนึ่งตัวจะมีจำนวนสโตรคได้มากที่สุด 2 สโตรค



รูปที่ 2.4 จุดเด่นบนตัวอักษร น [3]

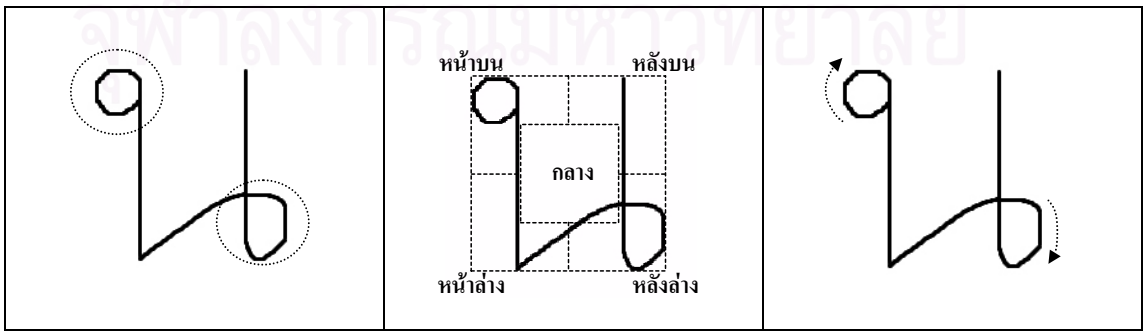
2.2.2.4 จำนวนวงรอบ[2]

วงรอบ คือ ชุดข้อมูลจุดที่มีลักษณะการเรียงลำดับของจุดเป็นวงรอบ ภายในสโตรคเดียวกัน ในภาษาไทยวงรอบเห็นได้ในลักษณะที่เป็นหัวของตัวอักษร

2.2.2.4.1. เทคนิคการหาวงรอบ [3]

เนื่องจากตัวอักษรภาษาไทย โดยส่วนใหญ่จะมีวงรอบเป็นส่วนประกอบ ซึ่งวงรอบของแต่ละตัวอักษรมีความแตกต่างกัน คือ จำนวนวงรอบในแต่ละตัวอักษร ตำแหน่งและทิศทางการเขียนวงรอบ เช่น ตัวอักษร น จะมีจำนวนวงรอบ 2 วงรอบ ตำแหน่งของวงรอบแรกจะอยู่ส่วนหน้าบน ตำแหน่งของวงรอบที่สองจะอยู่ส่วนหลังล่าง ทิศทาง การวนของวงรอบแรกและวงรอบที่สองจะมีทิศทางตามเข็มนาฬิกา แสดงดังรูปที่ 2.5

การหาวงรอบของตัวอักษรภาษาไทยสามารถหาได้ 2 วิธีคือ การหาวงรอบจากจุดตัดและการหาวงรอบจากการเปลี่ยนแปลงของทิศทางรหัสลูกโซ่

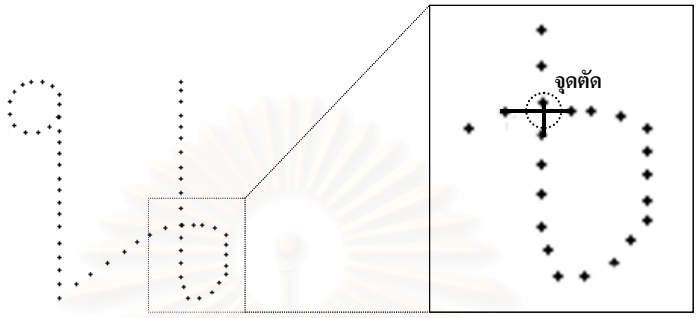


(ก) มี 2 วงรอบ (ข) ตำแหน่งของวงรอบ (ค) ทิศทางการวนของวงรอบ

รูปที่ 2.5 วงรอบของตัวอักษร น [3]

1) การหาวงรอบจากจุดตัด

พิจารณาจากจุดตัดที่เกิดจากการตัดกันของเส้นตรงสองเส้น ซึ่งเส้นตรงทั้งสองนี้จะเกิดจากการเชื่อมระหว่างจุดสองจุดที่อยู่ติดกันใดๆ จุดตัดที่ได้จะเป็นจุดเริ่มต้นและจุดสุดท้ายของวงรอบ ดังรูปที่ 2.6



รูปที่ 2.6 จุดตัดวงรอบที่สองของตัวอักษร น [3]

วิธีการหาจุดตัด สามารถหาได้จากการหาสมการเส้นตรงระหว่างจุดสองจุดที่อยู่ติดกัน โดยจะเริ่มจากจุดแรกของสโตรค แล้วหาสมการเส้นตรงเส้นที่สองจากจุดในลำดับต่อไป ในการหาสมการเส้นตรงที่เกิดจากจุดสองจุดนั้น ใช้สมการที่ 2.11 และ 2.12

$$y = mx + b \qquad \ldots(2.11)$$

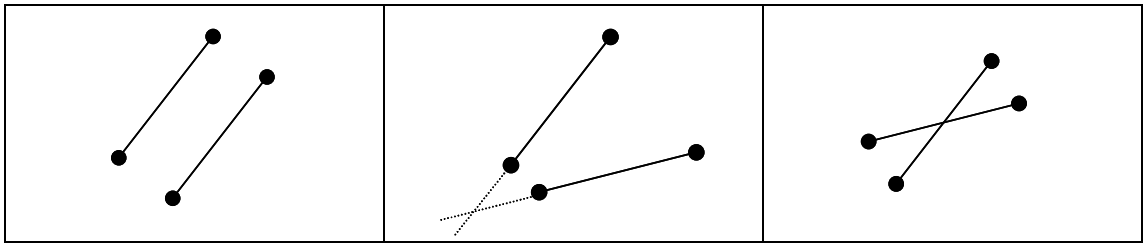
$$m = \frac{y_2 - y_1}{x_2 - x_1} \qquad \ldots(2.12)$$

เมื่อ

m	คือ ความชันของเส้นตรงที่ผ่านจุด (x1,y1) และ (x2,y2)
(x1,y1)	คือ จุดแรกของเส้นตรง
(x2,y2)	คือ จุดที่สองของเส้นตรง
b	คือ จุดตัดแกน y (y-intercept)

ซึ่งหาได้ในสมการที่ 2.11 โดยการแทนค่า (x,y) ด้วย (x1,y1) หรือ (x2,y2)

เมื่อได้สมการของเส้นตรงทั้งสองเส้นแล้ว สามารถหาจุดตัดได้จากการแก้สมการเส้นตรงทั้งสอง โดยให้ค่า y หรือ x เท่ากัน เมื่อได้จุดตัดจะต้องตรวจสอบว่าจุดตัดนั้นอยู่บนเส้นตรงทั้งสอง แสดงดังรูปที่ 2.7(ค) จึงถือว่าเป็นจุดตัดของวงรอบ



(ก) ไม่มีจุดตัด (ข) จุดตัดไม่อยู่บนเส้นตรง (ค) จุดตัดอยู่บนเส้นตรง

รูปที่ 2.7 ลักษณะการตัดกันของเส้นตรง [3]

การหาจุดตัดจะต้องหาจากจุดที่กำลังพิจารณาเทียบกับจุดในลำดับถัดไป ภายในระยะ k จุด (ในงานวิจัยนี้ให้ k = 35) โดยงานวิจัยนี้ได้กำหนดให้ขนาดตัวอักษร 600x600 หน่วย และให้ระยะห่างระหว่างจุด 25 หน่วย ซึ่งจะทำให้ขนาดของวงรอบที่เป็นไปได้ไม่เกิน 35 จุด และการหาจุดตัดจะต้องเป็นจุดตัดที่อยู่ภายในสโตรคเดียวกันเท่านั้น เพื่อป้องกันการเกิดจุดตัดที่ไม่ใช่จุดตัดของวงรอบ

2) การหาวงรอบจากการเปลี่ยนแปลงของทิศทางรหัสลูกโซ่

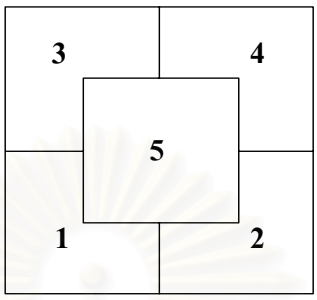
ในบางครั้งการเขียนวงรอบแรกอาจไม่เกิดจุดตัดได้ จึงจำเป็นที่จะต้องหาวงรอบแรกโดยใช้เทคนิค การหาวงรอบจากการเปลี่ยนแปลงของทิศทางรหัสลูกโซ่เข้ามาช่วย โดยการสังเกตลักษณะการเปลี่ยนแปลงของทิศทางรหัสลูกโซ่ ตั้งแต่จุดแรกของสโตรคต่อเนื่องกันไปเป็นระยะไม่เกิน L หน่วย (ในงานวิจัยนี้ให้ L= 15) ถ้าชุดของจุดใดมีลักษณะการเปลี่ยนแปลงที่น่าจะก่อให้เกิดวงรอบ ก็จะถือว่าเจอวงรอบ เช่น ลำดับรหัสลูกโซ่ 3-4-5-5-5-6-7-7-0-1-1-2 จะเป็นลักษณะของวงรอบ

ในงานวิจัยนี้ การหาวงรอบจะหาโดยใช้วิธีการหาวงรอบจากจุดตัด แต่สำหรับวงรอบแรกถ้าไม่พบจุดตัดในช่วงแรกของสโตรค (5 จุดแรกของสโตรค) จึงหาวงรอบโดยวิธีการเปลี่ยนแปลงทิศทางรหัสลูกโซ่แทน

2.2.2.5 รหัสโซนสำหรับตัวอักษรภาษาไทย [3]

การแบ่งรหัสโซนแบบ 5 โซน เนื่องจากตัวอักษรภาษาไทยโดยส่วนใหญ่จะพบวงรอบในตำแหน่งต่างๆ คือ หน้าบน หน้าล่าง หลังบน หลังล่าง และตรงกลาง ดังนั้นจึงใช้การแบ่งรหัสโซนแบบ 5 โซนสำหรับตัวอักษรภาษาไทย โดยรหัสโซนแบบ 5 โซนนั้นจะแบ่งความกว้างและความ

สูงของตัวอักษรออกเป็น 4 ส่วน เท่าๆ กัน ซึ่งจะได้พื้นที่ทั้งหมด 16 หน่วย โดยที่โซนที่ 1 ถึงโซนที่ 4 จะครอบคลุมพื้นที่โซนละ 3 หน่วย ส่วนโซนที่ 5 จะครอบคลุมพื้นที่ 4 หน่วย แสดงดังรูปที่ 2.8



รูปที่ 2.8 รหัสโซนของตัวอักษรแบบ 5 โซน [3]

2.3 **ทฤษฎีข่ายงานประสาท** [5]

ข่ายงานประสาท คือ ระบบที่ประกอบไปด้วยหน่วยประมวลแบบง่าย ๆ หลาย ๆ ตัวที่อาจเชื่อมโยงกันอย่างทั่วถึง (Fully Connection) หรือเชื่อมโยงกันแบบบางส่วน (Partial Connection) ก็ได้ การทำงานของระบบขึ้นอยู่กับลักษณะการเชื่อมโยงระหว่างหน่วยประมวลผลและค่าน้ำหนัก เป็นชุดสถาปัตยกรรมสำหรับการประมวลผลข้อมูลซึ่งอยู่ในรูปแบบต่าง ๆ อาจจะเป็นซอฟต์แวร์หรือโครงสร้างของฮาร์ดแวร์ที่ถูกสอนให้รู้จักรูปแบบข้อมูลแบบต่าง ๆ สำหรับการทำงานข่ายงานประสาทเรียนรู้จากการจำแนกข้อมูลจำนวนมากโดยเทียบกับข้อมูลเข้า-ออกที่กำหนด ปรับตัวแปรหรือค่าน้ำหนักที่จุดที่นิวรอลติดต่อซึ่งกันและกัน จากนั้นชั้นของหน่วยประมวลผลในแต่ละระดับจะสื่อสารกัน หน้าที่ของหน่วยประมวลผลเหล่านี้และโครงสร้างของการเชื่อมโยงแนวคิดจากการศึกษาระบบประสาทของมนุษย์

งานวิจัยนี้จะใช้ข่ายงานประสาทแบบแพร่กระจายย้อนกลับ โดยหน่วยประมวลมีโครงสร้างของการเชื่อมโยงกันอย่างทั่วถึง ดังรูปที่ 2.9

สมการ Output Layer สามารถแสดงได้ดังนี้

$$y_i = f(w_i^1 x_1 + w_i^2 x_2 + w_i^3 x_3 + \cdots + w_i^m x_m) \quad \ldots(2.13)$$

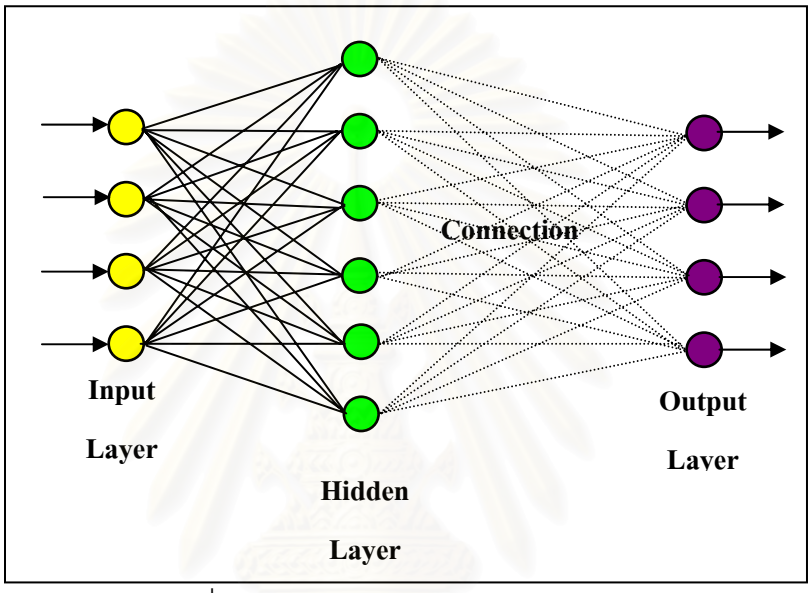
$$y_i = f(\sum_j w_i^j x_j) \quad \ldots(2.14)$$

โดยที่

y_i Output จากของแต่ละโหนด

x_j input จากแต่ละโหนด

w_i^j น้ำหนัก (weight) ของแต่ละแขน (connection)



รูปที่ 2.9 โครงสร้างของข่ายงานประสาท [5]

2.3.1 สารสนเทศที่ใช้ในข่ายงานประสาท

2.3.1.1. ข้อมูลนำเข้า (Input) ต้องเป็นค่าตัวเลข ถ้าเป็นข้อมูลเชิงคุณภาพต้องแปลงให้อยู่ในเชิงปริมาณที่ข่ายงานประสาทยอมรับเข้าไปเพื่อเรียนรู้ได้ นั่นคือกระบวนการเบื้องต้นก่อนการประเมินผล (Preprocess)

2.3.1.2. ผลลัพธ์ (Output) คือผลลัพธ์ที่เกิดขึ้นจริง (Actual Output) จากกระบวนการเรียนรู้ของข่ายงานประสาท

2.3.1.3. ค่าน้ำหนัก (Weights) คือสิ่งที่ได้จากการเรียนรู้ของข่ายงานประสาท หรือเรียกอีกอย่างหนึ่งว่าค่าความรู้ (Knowledge) ค่าน้ำหนักเป็นสิ่งที่สำคัญมากของข่ายงานประสาท และค่าเหล่านี้จะไม่มีการเปลี่ยนแปลงอีกต่อไป ค่าเหล่านี้จะใช้ในการระลึกข้อมูลอื่น ๆ ที่อยู่ในรูปแบบเดียวกัน

2.3.1.4. ฟังก์ชันผลรวม (Summation Function) เป็นผลรวมของข้อมูลนำเข้า (ai) และค่าน้ำหนัก

$$S = \sum_{i=1}^{n} a_i w_i \qquad \ldots(2.15)$$

2.3.1.5. ฟังก์ชันการแปลงค่า (Transfer Function) เป็นการคำนวณการจำลองการทำงานของนิวรอลต่าง ๆ ฟังก์ชันที่นิยมใช้คือ

ฟังก์ชันซิกมอยด์ (Sigmoid Function)

$$f(x) = 1/(1+e^{-x}) \qquad \ldots(2.16)$$

จุดประสงค์ของการแปลงค่าคือ เพื่อให้ผลลัพธ์ที่ได้อยู่ในช่วง 0 – 1 มิเช่นนั้นค่าที่ได้จะโตมาก ดั้งนั้นจึงต้องมีการแปลงค่าข้อมูลที่ใช้ในการสอนและทดสอบ เพื่อให้สอดคล้องกับฟังก์ชันที่ใช้

2.3.2 ขั้นตอนการเรียนรู้ของข่ายงานประสาท

ขั้นตอนการเรียนรู้ของข่ายงานประสาท สามารถแบ่งออกได้เป็น 3 ขั้นตอนหลัก คือ

2.3.2.1. คำนวณผลลัพธ์

2.3.2.2. เปรียบเทียบผลลัพธ์ที่เกิดขึ้นจริง และค่าเป้าหมายที่ตั้งไว้ (Desired/Target Output) ถ้าค่าความผิดพลาดอยู่ในระดับที่ยอมรับได้ ก็หยุดการสอน

2.3.2.3. ปรับปรุงค่าน้ำหนัก และทำซ้ำข้อ 1 ใหม่

2.3.3 ประเภทการเรียนรู้ของข่ายงานประสาท

การเรียนรู้ของข่ายงานประสาทสามารถแบ่งออกเป็น 3 ประเภท คือ

2.3.3.1. การเรียนรู้แบบมีครู (Supervised Learning)

การเรียนรู้แบบมีครูต้องการชุดข้อมูลนำเข้าและผลลัพธ์เป้าหมายเป็นชุดการสอนควบคู่ (Training Pair) โดยปกติการสอนข่ายงานประสาทจะใช้ชุดการสอนควบคู่หลายชุด ในระหว่างการสอนข่ายงานประสาทจะเกิดผลลัพธ์จริงขึ้น ผลต่างระหว่างผลลัพธ์จริงกับผลลัพธ์เป้าหมายคือค่าความคลาดเคลื่อนหรือค่าความผิดพลาด

การเรียนรู้ประเภทนี้ได้แก่ วิธีอะแด็พทีพลีเนียนิวรอลอีเลเม้นท์ (Adaptive Linear Neural Element - ADALINE) และวิธีแบ็คพร็อคเพเกชันรีเคอเร้นท์นิวรอลเน็ตเวิร์ค (Back propagation Recurrent Neural Network - RNN) เป็นต้น

2.3.3.2. การเรียนรู้แบบไม่มีครู (Unsupervised Learning)

การเรียนรู้แบบไม่มีครูถูกพัฒนาขึ้นเพื่อให้ใกล้เคียงกับระบบการเรียนรู้ของสมองมนุษย์มากยิ่งขึ้น โดยมีเพียงชุดข้อมูลนำเข้า กระบวนการเรียนรู้จะใช้หลักการทางสถิติ โดยหาค่าทางสถิติของชุดการสอน และจัดกลุ่มข้อมูลออกเป็นระดับต่าง ๆ ข่ายงานประสาทจะหาค่าผลลัพธ์เอง จากความสัมพันธ์ระหว่างข้อมูลนำเข้าและผลลัพธ์

การเรียนรู้ประเภทนี้ได้แก่ วิธีฮอพฟิลด์เน็ตเวิร์คเรคค็อกนิชัน (Hopfield Network Recognition) และวิธีเซลฟออแกนไนซิงฟีเจอร์แมพ (Self Organizing Feature Map - SOFM) เป็นต้น

2.3.3.3. การเรียนรู้เชิงบังคับ (Reinforcement Learning)

การเรียนรู้เชิงบังคับเป็นการเรียนรู้ทั้งแบบมีครูและไม่มีครู การเรียนรู้แบบไม่มีครูคือในระหว่างการสอนมีเพียงชุดข้อมูลนำเข้า การเรียนรู้แบบมีครูคือเมื่อได้ผลลัพธ์จะบอกว่าถูกหรือผิด แต่ไม่ได้บอกว่าผลลัพธ์ที่ถูกคืออะไร

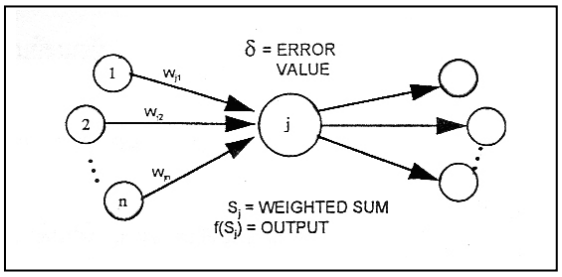
2.3.4 **การเรียนรู้แบบย้อนกลับ** (Back propagation)

การเรียนรู้แบบย้อนกลับเป็นวิธีการหนึ่งของข่ายงานประสาทที่ง่ายต่อความเข้าใจ เนื่องจากกระบวนการเรียนรู้และปรับปรุงแก้ไขนั้นเป็นไปด้วยตนเอง หรืออยู่ในเกณฑ์ที่ยอมรับได้ นั่นคือค่าที่ได้ในครั้งต่อ ๆ ไปจะมีความถูกต้องมากยิ่งขึ้น

การเรียนรู้แบบย้อนกลับมีโครงสร้างเป็นชั้น ๆ แต่ละชั้นเชื่อมโยงกันอย่างทั่วถึงกับชั้นที่อยู่ด้านบนและชั้นที่อยู่ด้านล่าง เมื่อข่ายงานประสาทได้รับข้อมูลนำเข้าจะทำการคำนวณค่าน้ำหนักของหน่วยประมวลผลนำเข้าไปยังชั้นแอบแฝงและจากชั้นแอบแฝงไปยังชั้นแสดงผลลัพธ์ เมื่อเกิดผลต่างระหว่างผลลัพธ์จริงกับผลลัพธ์เป้าหมาย ข่ายงานประสาทจะปรับค่าความผิดพลาดจากหน่วยแสดงผลลัพธ์และแพร่ย้อนกลับไปยังชั้นแอบแฝง จากชั้นแอบแฝงไปยังชั้นข้อมูลนำเข้า

2.3.4.1. โครงสร้างของการเรียนรู้แบบย้อนกลับ

การเรียนรู้แบบย้อนกลับมีโครงสร้างเป็นชั้น ๆ โดยมีโครงสร้างอย่างง่าย ๆ 3 ชั้น คือ ชั้นข้อมูลนำเข้า ชั้นแอบแฝงและชั้นแสดงผลลัพธ์ แต่ละชั้นจะติดต่อกันอย่างสมบูรณ์ดังรูปที่ 2.10



รูปที่ 2.10 หน่วยประมวลผล [5]

หน่วยประมวลผลกลางประกอบด้วย ผลรวมถ่วงน้ำหนักของข้อมูลนำเข้า (Sj) ผลลัพธ์ (f(Sj)) และค่าความผิดพลาด (δj) ที่ใช้ในการปรับค่าน้ำหนัก

ค่าน้ำหนักที่เกี่ยวเนื่องกับการติดต่อแต่ละชั้นจะถูกปรับแต่งตลอดเวลาระหว่างการเรียนรู้เพื่อลดค่าความแตกต่างระหว่างผลลัพธ์ที่ได้กับผลลัพธ์ที่ต้องการ

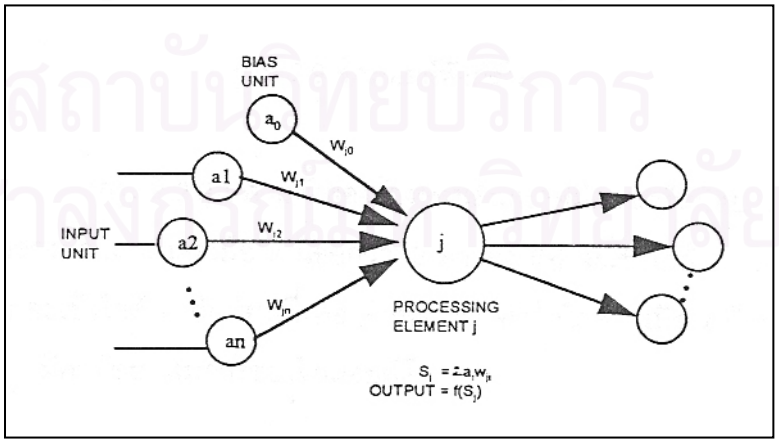
ค่าน้ำหนักจากหน่วยที่ i ไปยังหน่วยที่ j คือ Wji เมื่อจบการเรียนรู้แล้วค่าน้ำหนักนี้จะใช้ในกระบวนการการระลึก

2.3.4.2. ขั้นตอนการเรียนรู้แบบย้อนกลับ

ประกอบด้วย 2 ขั้นตอน

2.3.4.2.1 การแพร่เดินหน้า (Forward Propagation)

ขั้นตอนนี้เริ่มเมื่อข่ายงานประสาทได้รับข้อมูลนำเข้าและค่าของหน่วยประกอบผลในชั้นข้อมูลนำเข้าถูกกำหนด ชั้นอื่น ๆ จึงจะเริ่มทำการแพร่เดินหน้า



รูปที่ 2.11 ขั้นตอนการแพร่เดินหน้า [5]

จากรูปเป็นการคำนวณผลรวมของผลลัพธ์ที่เข้ามายังหน่วยที่ j ดังสมการ

$$S_j = \sum a_i w_{ji} \quad \text{...(2.17)}$$

โดยที่ ai = ข้อมูลจากหน่วยที่ i

wji = ค่าน้ำหนักจากหน่วยที่ i ไปยังหน่วยที่ j

จากนั้นจะทำการแปลงค่าข้อมูลโดยคำนวณหาค่า f(Sj) โดยซิกมอยด์ฟังก์ชัน

$$f(x) = 1/1+e^{-x} \quad \text{...(2.18)}$$

เมื่อ x เป็นค่าผลรวมของหน่วยที่ j ดังนั้น

$$f(S_j) = 1/1+e^{-S_j} \quad \text{...(2.19)}$$

เมื่อได้รับค่า f(Sj) แล้วค่า f(Sj) จะกลายเป็นผลลัพธ์ของหน่วยที่ j ซึ่งก็คือค่า aj ดังรูปที่ 2.11 โดยจะส่งค่าออกไปยังหน่วยอื่น ๆ ด้วยค่า aj ที่เท่ากัน

สำหรับชั้นข้อมูลนำเข้าจะเป็นกรณีพิเศษ โดยที่หน่วยประมวลผลในชั้นนี้จะไม่ทำการคำนวณผลรวมจากข้อมูลนำเข้าและจะไม่ทำการแปลงข้อมูล เนื่องจากแต่ละหน่วยประมวลผลจะใช้ค่าของข้อมูลนำเข้าเป็นค่าของตัวเอง

จากรูปที่ 2.11 หน่วยเอนเอียง (Bias Unit) เป็นหน่วยที่มีค่าเป็น 1 และเชื่อมโยงกับหน่วยอื่นทุก ๆ หน่วย และค่าน้ำหนักของหน่วยนี้จะถูกปรับในระหว่างการเรียนรู้เช่นกันหน่วยนี้จะให้ค่าคงที่กับหน่วยอื่นซึ่งบางครั้งทำให้การเรียนรู้ (Convergence Time) ของข่ายงานประสาทเร็วขึ้น หน่วยเอนเอียงมีผลต่อระดับการกระตุ้น (Threshold) ของหน่วยอื่น นั่นคือเมื่อค่าน้ำหนักคงที่และ

$$S_j = \sum a_i w_{ji} \quad \text{...(2.20)}$$

$$C = Wj0 \quad \text{...(2.21)}$$

ดังนั้นผลรวมของหน่วยที่ j = S + C ซึ่งจะทำให้กราฟของสมการซิกมอยด์ขยับไปทางซ้าย C หน่วย ด้วยวิธีนี้จะทำให้ระดับการกระตุ้นของหน่วยเป้าหมายเปลี่ยนไป

เหตุนี้ต้องใช้ซิกมอยด์เนื่องจากต้องการให้ระดับกระตุ้นเป็นแบบอ่อน (Soft Threshold) มากกว่าการกระตุ้นแบบแรง (Hard Threshold) นั้นคือซิกมอยด์ฟังก์ชันให้ค่าที่ต่อเนื่องกัน

2.3.4.2.2 การแพร่ย้อนกลับ (Backward Propagation)

จากรูป 2.12 แสดงขั้นตอนการแพร่ย้อนกลับ ขั้นนี้ค่าความผิดพลาด δ จะถูกคำนวณสำหรับทุกหน่วยและคำนวณค่าน้ำหนักที่จะเปลี่ยนทุกการเชื่อมโยง การคำนวณนี้เริ่มที่ชั้นแสดงผลลัพธ์และแพร่ย้อนกลับไปยังชั้นข้อมูลนำเข้า

ขั้นปรับปรุงความผิดพลาดเกิดหลังจากที่ดำเนินการแพร่เดินหน้าแล้วหน่วยประมวลผลแต่ละหน่วยในชั้นแสดงผลลัพธ์จะให้ค่าผลลัพธ์ที่จะเปรียบเทียบกับผลลัพธ์เป้าหมายในชุดการสอนดังในรูปที่ 2.12a ความแตกต่างที่เกิดคือค่าความผิดพลาดสำหรับแต่ละหน่วยในชั้นแสดงผลลัพธ์ดังในรูปที่ 2.12b ค่าน้ำหนักของทุกการเชื่อมโยงไปยังชั้นแสดงผลลัพธ์จะถูกปรับ จากนั้น ค่าความผิดพลาดของหน่วยในชั้นแอบแฝงที่ต่ำกว่าชั้นแสดงผลลัพธ์ดังในรูปที่ 2.12c จะถูกคำนวณ แล้วค่าน้ำหนักของทุกการเชื่อมโยงไปยังชั้นแอบแฝงจะถูกปรับ กระบวนการนี้จะดำเนินไปจนกว่าชั้นสุดท้ายจะถูกปรับค่าน้ำหนัก

ค่า δ คือค่าความผิดพลาดท้องถิ่น (Local Error) สามารถหาได้จากการคำนวณง่าย ๆ สำหรับชั้นแสดงผลลัพธ์ และซับซ้อนสำหรับชั้นแอบแฝง ถ้าหน่วย j เป็นหน่วยที่อยู่ในชั้นแสดงผลลัพธ์แล้วการคำนวณค่าความผิดพลาดจะได้จากสมการ

$$\delta_j = \left(t_j - \mathrm{a}_j\right)\mathrm{f}'\left(S_j\right) \qquad \ldots(2.22)$$

เมื่อ tj = ค่าเป้าหมายของหน่วยที่ j

aj = ค่าผลลัพธ์ของหน่วยที่ j

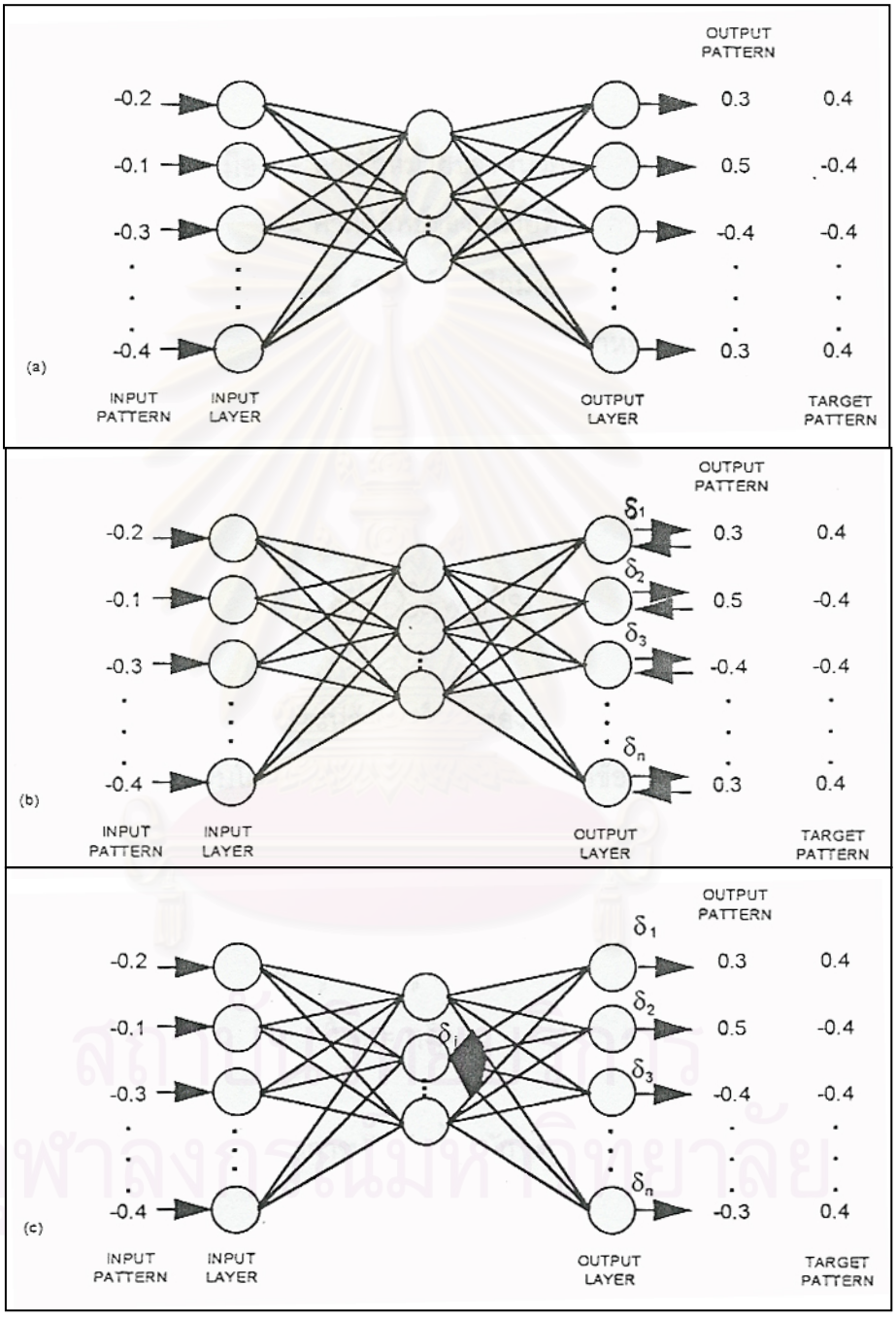
f/(x) = อนุพันธ์ของซิกมอยด์ฟังก์ชัน

Sj = ผลรวมของข้อมูลนำเข้ามาหน่วยที่ j

ถ้าเป็นหน่วย j เป็นหน่วยที่อยู่ในชั้นแอบแฝงแล้วค่าความผิดพลาดจะได้จาก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

$$\delta_j = \left[\sum \delta_k W_{kj}\right] f'(S_j) \qquad \ldots(2.23)$$



รูปที่ 2.12 ขั้นตอนการแพร่ย้อนกลับ [5]

การปรับค่าน้ำหนักจะปรับโดยใช้ค่า δ ทุกหน่วยที่รับผลลัพธ์จากหน่วยที่ j ค่าน้ำหนักแต่ละค่าจะถูกปรับโดยนำค่า δ ของหน่วยที่รับจากข้อมูลนำเข้าจากการเชื่อมโยงนั้น ค่าน้ำหนักจะถูกปรับโดยสมการ

$$W^{new}{}_{ji} = W^{old}{}_{ji} + \Delta w_{ji} \quad \text{...(2.24)}$$

$$\Delta w_{ji} = \eta \delta_j a_i \quad \text{...(2.25)}$$

ตัวแปร η คือค่าสัมประสิทธิ์การเรียนรู้ซึ่งจะถูกกำหนดโดยผู้ใช้ ถ้าค่า η มากจะทำให้ข่ายงานประสาทไม่คงที่และการเรียนรู้ไม่เป็นที่น่าพอใจ ถ้าเล็กเกินไปจะทำให้การเรียนรู้ช้า บางครั้งค่า η สามารถเปลี่ยนได้เพื่อทำให้การเรียนรู้มีประสิทธิภาพขึ้น เช่นเริ่มต้นใช้ค่า η มาก ๆ และลดลงในระหว่างการสอน

นอกจากค่า η แล้วยังมีค่าโมเมนตัม α ซึ่งคิดค้นโดย Rumelhart Hinton และ William ในปี ค.ศ. 1986 เพื่อช่วยให้การเรียนรู้เร็วขึ้น ดังนั้นค่าน้ำหนักจะถูกปรับโดยสมการ

$$\Delta w_{ji}(k+1) = \eta \delta_j a_i + \alpha \left[\Delta w_{ji}(k)\right] \quad \text{...(2.26)}$$

เมื่อ k คือ ดัชนีเวลา (Time Index) หรือจำนวนรอบของการปรับปรุงค่าน้ำหนัก

2.3.4.2.3 การสอนการเรียนรู้แบบย้อนกลับ

การสอนข่ายงานประสาทชนิดการเรียนรู้แบบย้อนกลับ เป็นการสอนแบบมีครู โดยมีการกำหนดรูปแบบข้อมูลนำเข้าควบคู่กับรูปแบบเป้าหมาย มีการกำหนดชุดการสอนหลาย ๆ รูปแบบเพื่อให้ข่ายงานประสาทสามารถเรียนรู้

ในกระบวนการสอนเพื่อให้ข่ายงานประสาทสามารถเรียนรู้ได้ดีนั้นจะต้องใช้การวนซ้ำสอนหลาย ๆ รอบ การสอนข่ายงานประสาทจะสำเร็จหรือไม่ดูจากค่า RMS ดังสมการ

$$RMS = \sqrt{\sum_p \sum_j \left(t_{jp} - a_{jp}\right)^2 / n_p n_o} \quad \text{...(2.27)}$$

โดยที่ np = จำนวนของรูปแบบในชุดการสอน

nO = จำนวนหน่วยในชั้นแสดงผลลัพธ์

tJP = ค่าเป้าหมายของหน่วยที่ j หลังจากเสนอรูปแบบ p

aJP = ค่าผลลัพธ์ที่ได้จากหน่วยที่ j หลังจากเสนอรูปแบบ p

เมื่อค่า RMS เข้าใกล้ค่าศูนย์ หรือลดต่ำกว่า 0.1 หมายความว่าข่ายงานประสาทได้เกิดการเรียนรู้แล้ว

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

การออกแบบและพัฒนาระบบ

ในบทนี้ จะกล่าวถึงการออกแบบและพัฒนาระบบการรู้จำลายมือออนไลน์สำหรับตัวอักษรภาษาไทยบนแท็บเล็ตพีซี ซึ่งได้ออกแบบให้ระบบสามารถรู้จำได้ทั้งตัวอักษรไทยและตัวอักษรภาษาอังกฤษ รองรับการเขียนตัวอักษรครั้งละตัวอักษรและครั้งละหลายตัวอักษร และสามารถส่งตัวอักษรที่รู้จำได้ไปยังโปรแกรมไมโครซอฟท์เวิร์ดและไมโครซอฟท์เอ็กซเซล โดยแบ่งเป็นหัวข้อดังต่อไปนี้ เทคนิคที่ใช้ในการจำแนกตัวอักษรภาษาไทย การออกแบบ แผนภาพคลาสและความสัมพนธ์ และสภาพแวดล้อมในการพัฒนาระบบ ซึ่งมีรายละเอียดในแต่ละหัวข้อดังนี้

3.1 เทคนิคที่ใช้ในการจำแนกตัวอักษรภาษาไทย

จากการออกแบบระบบให้สามารถรู้จำตัวอักษรที่เขียนครั้งละตัวอักษรและครั้งละหลายตัวอักษร สามารถแบ่งเทคนิคในการจำแนกลักษณะตัวอักษรออกเป็น 2 เทคนิคใหญ่ๆคือ

1) เทคนิคการจำแนกลักษณะตัวอักษรครั้งละตัวอักษร เป็นเทคนิคที่ได้จากผลการวิจัยของนายปกรณ์ บุพศิริ

2) เทคนิคการจำแนกลักษณะตัวอักษรครั้งละหลายตัวอักษร เป็นเทคนิคที่ผู้วิจัยได้เพิ่มเติมเข้ามา เพื่อให้ระบบสามารถรองรับการเขียนตัวอักษรภาษาไทยครั้งละหลายตัวอักษร โดยมี 2 เทคนิคย่อยดังนี้

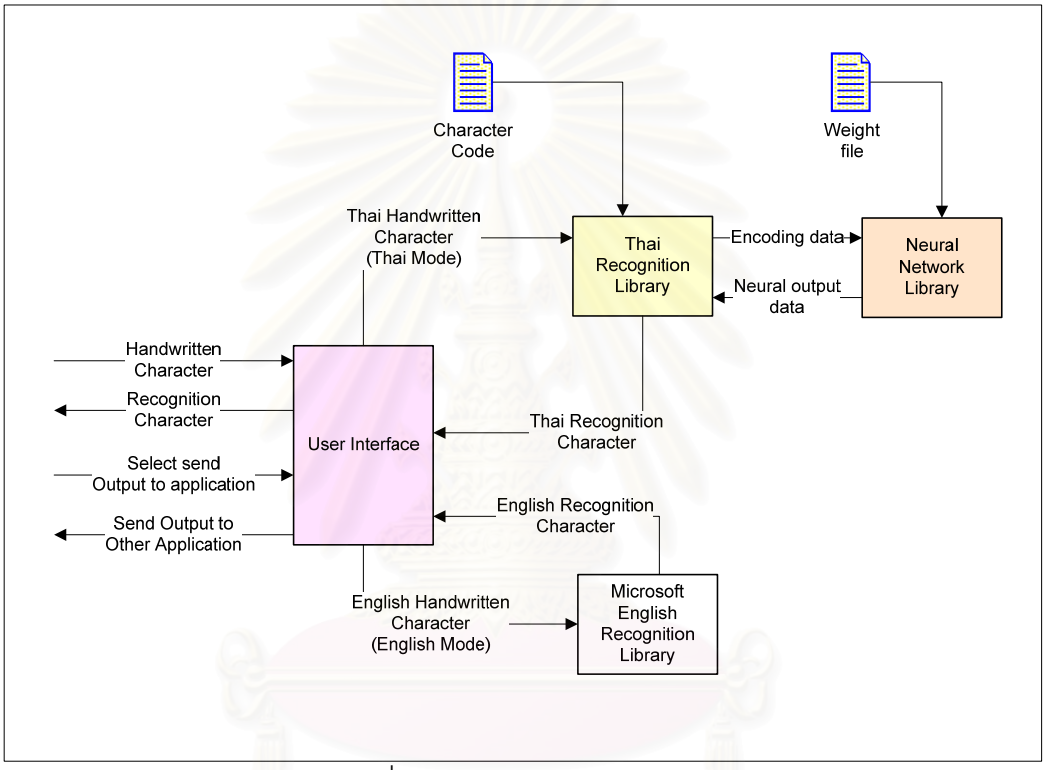
- การคัดแยกตัวอักษร (Character Separation) เพื่อแบ่งแยกสโตรคของแต่ละตัวอักษร ก่อนที่จะนำไปประมวลผลการรู้จำ เนื่องจากตัวอักษรภาษาไทย 1 ตัวอักษรมีทั้งที่ประกอบด้วย 1 และ 2 สโตรค

- การตรวจสอบตัวอักษรก่อนหน้า (Check Previous Character) เพื่อทำการปรับปรุงตัวอักษรที่รู้จำได้ให้มีความถูกต้องมากยิ่งขึ้น โดยใช้ลักษณะพิเศษของตัวอักษรภาษาไทยบางตัวและหลักการเขียนภาษาไทยเข้ามาช่วยในการปรับปรุงตัวอักษร

เทคนิคที่ใช้ทั้งหมดนี้จะถูกอธิบายอย่างละเอียดในส่วนของการออกแบบไลบรารีสำหรับรู้จำลายมือภาษาไทย

3.2 การออกแบบระบบ

ผู้วิจัยได้ออกแบบระบบโดยใช้หลักในการออกแบบระบบเพื่อให้ผู้ใช้ระบบสามารถใช้งานได้โดยง่าย และแบ่งส่วนของฟังก์ชันต่างๆแยกออกจากกัน เพื่อให้สะดวกต่อการนำฟังก์ชันต่างๆไปใช้กับระบบอื่นๆ หรือใช้พัฒนาระบบอื่นๆ ต่อไป โดยแบ่งออกเป็นส่วนประกอบหลัก 3 ส่วนดังแสดงในรูปที่ 3.1

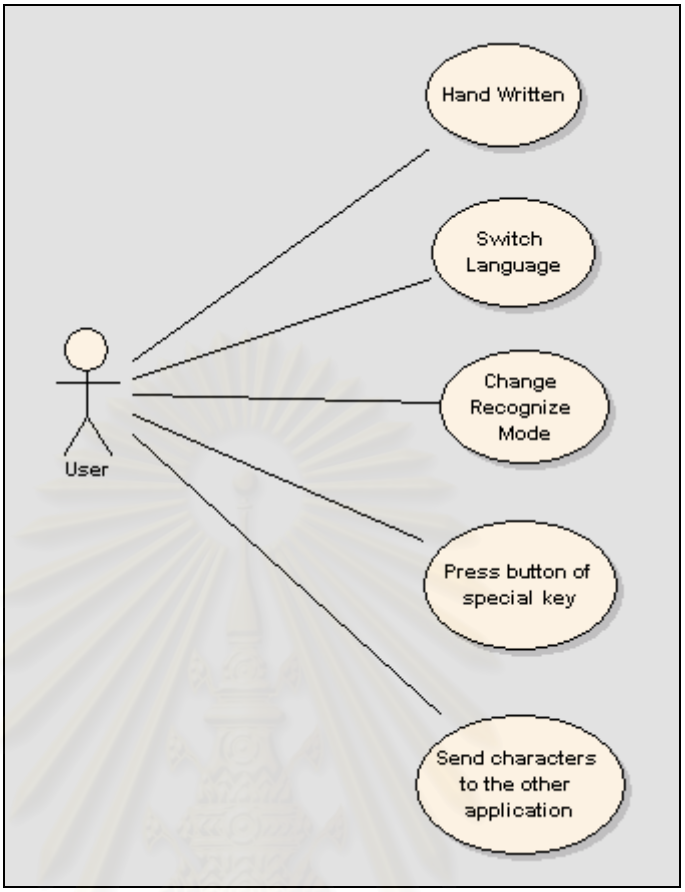


รูปที่ 3.1 ส่วนประกอบหลักของระบบ

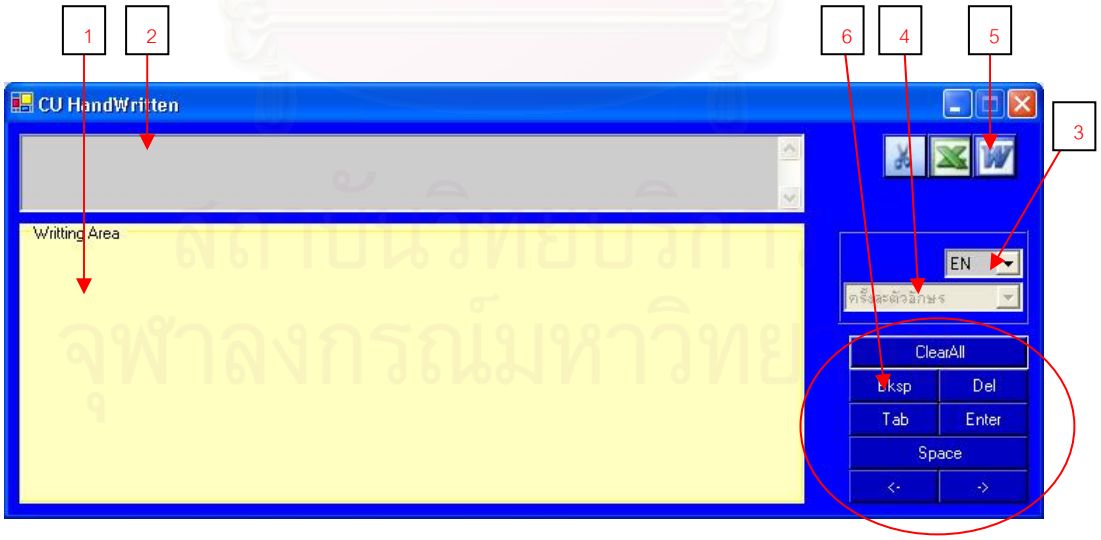
- การออกแบบส่วนต่อประสานกับผู้ใช้ (User Interface)
- การออกแบบส่วนของฟังก์ชันในการรู้จำลายมือ (Thai Recognition Library)
- การออกแบบส่วนของฟังก์ชันข่ายงานประสาทแบบแพร่กระจายย้อนกลับ (Neural Network Library)

3.2.1 การออกแบบส่วนต่อประสานกับผู้ใช้

ผู้วิจัยได้ออกแบบให้ผู้ใช้งานสามารถใช้ได้โดยง่าย โดยแบ่งออกเป็นส่วนย่อยๆ ดังรูปที่ 3.2 และ 3.3 โดยมีรายละเอียดดังต่อไปนี้



รูปที่ 3.2 แผนภาพยูเคสส่วนต่อประสานกับผู้ใช้



รูปที่ 3.3 ส่วนประกอบของส่วนต่อประสานกับผู้ใช้

3.1.1.1. ส่วนของการรับข้อมูลลายมือเขียน

เป็นพื้นที่ที่ใช้รับข้อมูลลายมือเขียนของผู้ใช้ ซึ่งได้ออกแบบให้มีขนาดกว้าง เพื่อให้รองรับลายมือเขียนตัวอักษรภาษาไทย ซึ่งมีทั้งตัวอักษร สระด้านบนและด้านล่างตัวอักษร และได้เชื่อมพื้นที่นี้เข้ากับอิงค์คอลเลเตอร์ (Ink collector) ภายใต้ไมโครซอฟท์อิงค์ไลบารี่ (Microsoft ink library) สำหรับทำการเก็บข้อมูลลายมือเขียนทั้งหมด เพื่อนำไปประมวลผล

3.2.1.2. ส่วนแสดงผลตัวอักษรที่รู้จำได้บนหน้าจอ

เป็นส่วนแสดงผลตัวอักษรที่รู้จำได้ ซึ่งได้ออกแบบให้สามารถรองรับตัวอักษรจำนวนมาก พร้อมทั้งสามารถเลื่อนขึ้น-ลง เพื่อดูตัวอักษรทั้งหมดที่รู้จำได้ และรองรับการคลิกเม้าท์ขวา เพื่อเรียกใช้เมนูย่อย (popup menu) สำหรับใช้ฟังก์ชันคัดลอกหรือตัดเฉพาะตัวอักษรที่ต้องการ นำไปวางไว้ในเอกสารอื่นๆได้

3.2.1.3. รายการเลือกสำหรับสลับการทำงานระหว่างภาษาไทยกับภาษาอังกฤษ

แสดงรายการภาษาที่สามารถใช้งานได้ภายในระบบ และให้ผู้ใช้ระบบสามารถเลือกสลับภาษาที่ต้องการจะเขียนตัวอักษร ซึ่งได้ออกแบบให้ระบบสามารถรองรับทั้งภาษาไทยและภาษาอังกฤษ เพื่อให้ผู้ใช้สามารถใช้งานได้ทั้งภาษาไทยและภาษาอังกฤษ ภายใต้ระบบเดียวกัน เมื่อผู้ใช้เลือกภาษาที่ต้องการระบบจะทำการเปลี่ยนสถานะภาษาของคีย์บอร์ดของผู้ใช้งานให้อยู่ในสถานะภาษาที่เลือกโดยอัตโนมัติ

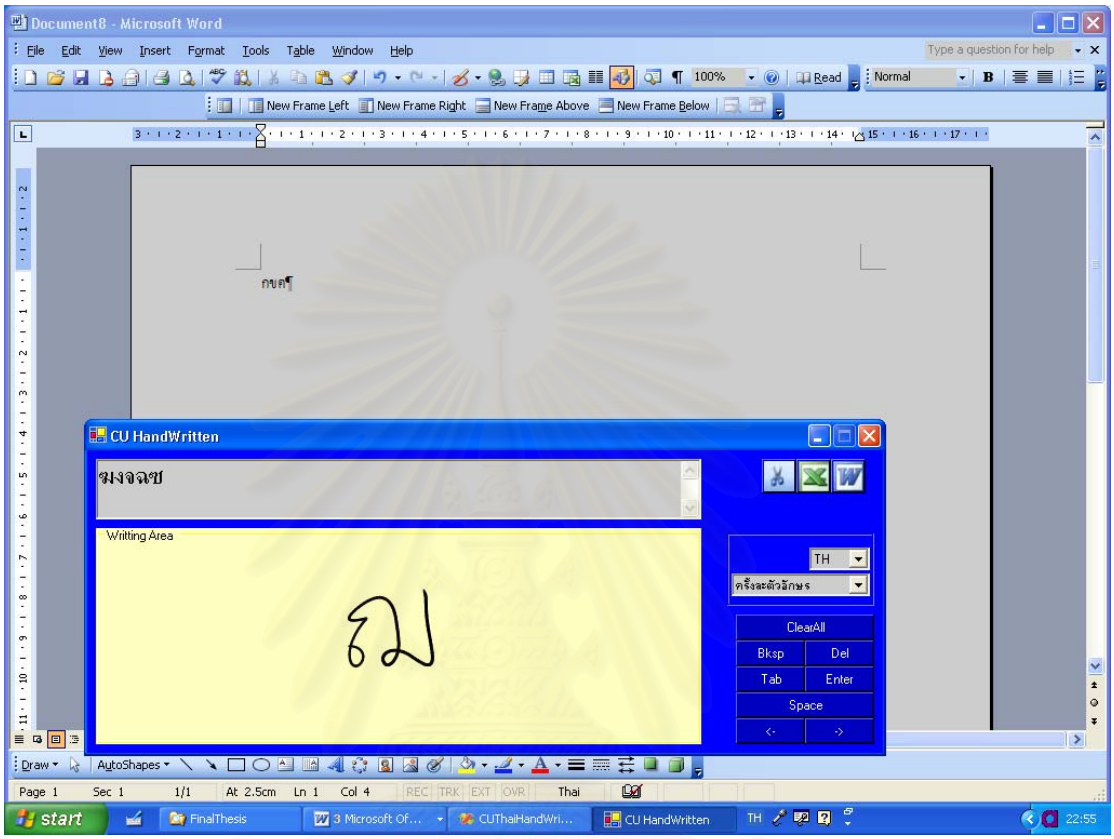
3.2.1.4. รายการเลือกสำหรับสลับการทำงานระหว่างการรับลายมือเขียนครั้งละตัวอักษรกับครั้งละหลายตัวอักษร

แสดงรายการการรับลายมือเขียนที่สามารถใช้งานได้ภายในระบบ และให้ผู้ใช้ระบบสามารถเลือกสลับลักษณะการเขียนตัวอักษร ซึ่งได้ออกแบบให้ระบบสามารถรองรับทั้งการเขียนครั้งละตัวอักษรและการเขียนครั้งละหลายตัวอักษร เพื่อให้ผู้ใช้สามารถใช้งานได้ตามความถนัด

3.2.1.5. ส่วนของการส่งชุดตัวอักษรที่รู้จำได้ ไปยังโปรแกรมอื่น

เป็นปุ่มสำหรับส่งตัวอักษรที่รู้จำได้ ไปยังโปรแกรมไมโครซอฟท์เวิร์ด ไมโครซอฟท์เอ็กซเซล เพื่อให้ผู้ใช้สามารถส่งตัวอักษรที่รู้จำได้ไปแสดงบนไฟล์เอกสารบนไมโครซอฟท์เวิร์ด หรือไมโครซอฟท์เอ็กซเซลที่เปิดใช้งานอยู่ขณะนั้น หากผู้ใช้ไม่ได้เปิดใช้งานไฟล์เอกสารใดๆอยู่ขณะนั้น

ระบบจะทำการสร้างเอกสารใหม่และเปิดเอกสารดังกล่าวขึ้นมาให้ผู้ใช้โดยอัตโนมัติต่อจากนั้นจึงทำการส่งตัวอักษรที่รู้จำได้ไปยังเอกสารดังกล่าว ดังรูปที่ 3.4



รูปที่ 3.4 การส่งชุดตัวอักษรที่รู้จำได้ไปยังโปรแกรมไมโครซอฟท์เวิร์ด

3.2.1.6. ส่วนของรหัสแป้นพิมพ์พิเศษ

เป็นปุ่มสำหรับส่งรหัสแป้นพิมพ์พิเศษไปยังส่วนแสดงผลตัวอักษรที่รู้จำได้ ซึ่งเป็นรหัสแป้นพิมพ์ที่ผู้ใช้งานจำเป็นต้องใช้ เช่น ปุ่มย่อหน้า (TAB) ปุ่มขึ้นบรรทัดใหม่ (ENTER) ปุ่มลบตัวอักษร (DELETE) ปุ่มเว้นวรรค (SPACE) เป็นต้น เนื่องจากในการใช้งานผู้ใช้อาจจะต้องการลบข้อมูลบางตัวอักษรหรือต้องการเว้นวรรคในบางช่วงของตัวอักษร ซึ่งการลบหรือการเว้นวรรคไม่ได้ถูกออกแบบเป็นสัญลักษณ์พิเศษที่ใช้ลายมือเขียน ดังนั้นจึงได้ออกแบบปุ่มรหัสแป้นพิมพ์พิเศษเหล่านี้ไว้เพื่ออำนวยความสะดวกในการใช้งาน

3.2.2 การออกแบบส่วนของฟังก์ชันการรู้จำลายมือ

ผู้ออกแบบได้ออกแบบโดยแบ่งส่วนของฟังก์ชันการรู้จำลายมือแยกออกจากส่วนต่างๆ เพื่อให้สะดวกในการปรับปรุงแก้ไขเฉพาะส่วน หรือเพื่อให้ง่ายต่อการนำไปใช้พัฒนาร่วมกับระบบอื่นๆที่ต้องการใช้คุณสมบัติในการรู้จำลายมือเขียนตัวอักษรไทยต่อไป โดยได้แบ่งออกเป็นส่วนหลักๆ ดังนี้

- ไลบรารีสำหรับรู้จำลายมือภาษาไทย
- ไลบารี่การรู้จำลายมือภาษาอังกฤษของไมโครซอฟท์

3.2.2.1 ไลบรารีสำหรับรู้จำลายมือภาษาไทย ประกอบด้วยฟังก์ชันต่างๆที่ใช้ในการรับข้อมูลลายมือเขียน ประมวลผลลายมือเขียน รวบรวมข้อมูลที่ได้ส่งไปยังข่ายงานประสาท และรับผลจากข่ายงานประสาทมาเทียบหาตัวอักษรที่รู้จำได้กับข้อมูลในไฟล์โค้ด (ตัวอย่างไฟล์โค้ดดังในภาคผนวก ข.) พร้อมทั้งส่งผลการรู้จำตัวอักษรกลับคืนให้กับโปรแกรมที่เรียกใช้งาน โดยมีส่วนประกอบของฟังก์ชันในการรู้จำลายมือดังรูปที่ 3.5

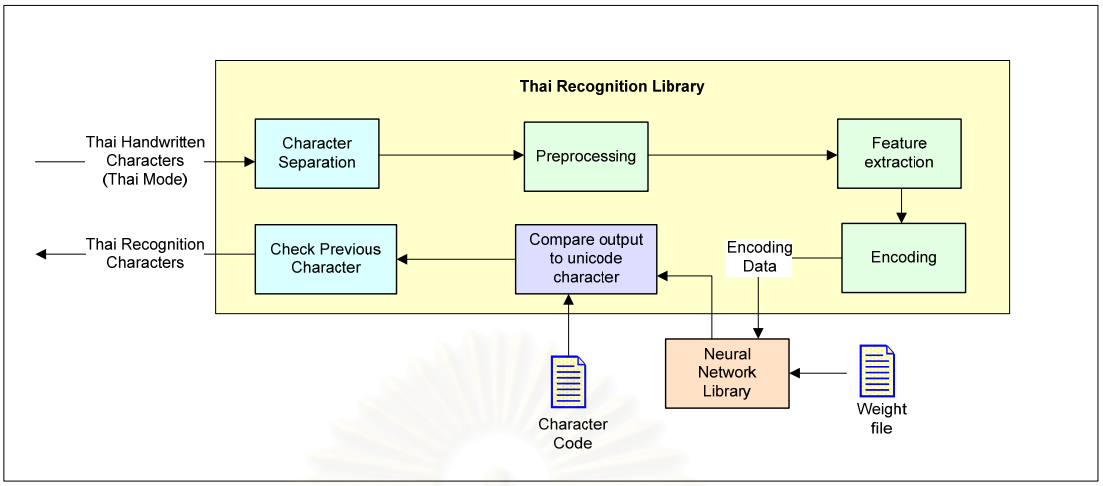
ส่วนของการประมวลผลลายมือเขียนดังกล่าวประกอบไปด้วยฟังก์ชันย่อยดังต่อไปนี้

3.2.2.1.1 ฟังก์ชันคัดแยกตัวอักษร (Character Separation)

สำหรับกรณีที่ผู้ใช้งานเลือกใช้งานในโหมดการเขียนครั้งละหลายตัวอักษร ฟังก์ชันนี้จะทำการคัดแยกสโตรคที่เป็นส่วนประกอบของตัวอักษรแต่ละตัวอักษร (ตัวอักษรไทย 1 ตัวอักษรสามารถประกอบด้วย 1 หรือ 2 สโตรค)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

เนื่องจากไฟล์น้ำหนักที่ใช้ในงานวิจัยนี้เป็นไฟล์น้ำหนักที่ได้จากการสอนข่ายงานประสาททีละตัวอักษร จึงต้องทำการคัดแยกตัวอักษรที่เขียนครั้งละหลายตัวอักษรออกเป็นตัวอักษรแต่ละตัว ต่อจากนั้นจึงส่งตัวอักษรทีละตัวอักษรไปประมวลผล และนำผลลัพธ์ของตัวอักษรแต่ละตัวอักษรที่รู้จำได้มาเรียงต่อกัน ซึ่งวิธีการคัดแยกสโตรคมีดังนี้

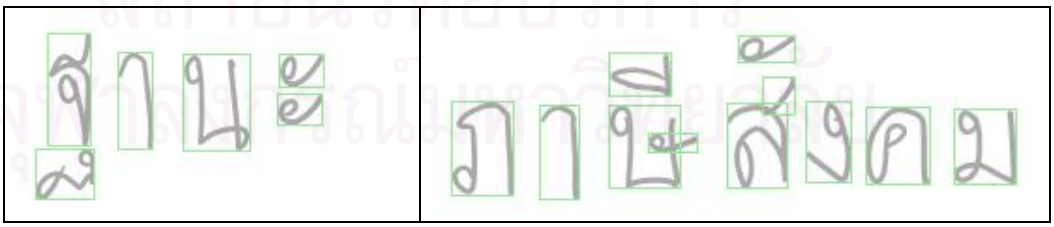


รูปที่ 3.5 ส่วนประกอบของฟังก์ชันในการรู้จำลายมือ

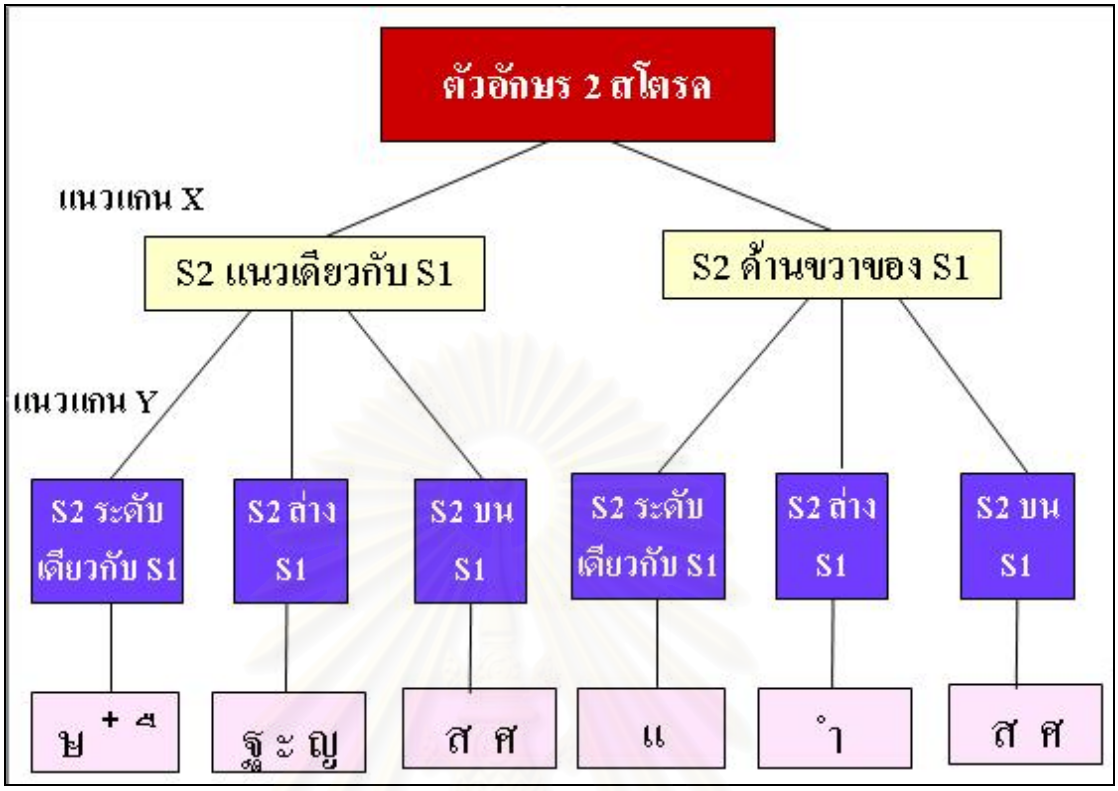
- การตรวจสอบระยะห่างระหว่างสโตรค(Stroke displacement) เป็นการแบ่งตัวอักษรโดยตรวจสอบระยะห่างของสโตรค 2 สโตรค ระหว่างสโตรคปัจจุบันกับสโตรคถัดไป โดยกำหนดให้ 2 สโตรคที่มีขอบเขตของสโตรคห่างกันไม่เกิน D หน่วย จะถือว่าทั้ง 2 สโตรคเป็นสโตรคของตัวอักษรเดียวกัน ส่วนสโตรคที่มีระยะห่างเกิน D หน่วย จะถือว่าเป็นสโตรคของตัวอักษรคนละตัว ซึ่งในงานวิจัยนี้ใช้ค่า D เท่ากับ 500 หน่วย หรือประมาณ 0.5 ซ.ม. (1 หน่วยของอิงค์คอลเลเตอร์ = 0.01 มิลลิเมตร) โดยมีขั้นตอนการตรวจสอบดังต่อไปนี้

ทำขั้นตอนต่อไปนี้สำหรับแต่ละสโตรค จนครบทุกสโตรค :

1) หาขอบเขตของแต่ละสโตรค โดยการหาค่า Xmax, Xmin, Ymax, Ymin ของแต่ละสโตรค ดังรูปที่ 3.6 และส่วนของโปรแกรมดังแสดงในภาคผนวก ก.



รูปที่ 3.6 การหาขอบเขตของสโตรค



รูปที่ 3.7 การแบ่งกลุ่มตัวอักษรที่มี 2 สโตรค

2) เปรียบเทียบระยะห่างระหว่างขอบเขตของแต่ละสโตรค โดยตรวจสอบระยะห่างระหว่างค่าขอบเขตของสโตรคปัจจุบันกับสโตรคถัดไป โดยพิจารณาทั้งแนวแกน X และแกน Y ดังรูปที่ 3.7 และ 3.8 (ส่วนของโปรแกรมดังแสดงในภาคผนวก ก.)

เมื่อ

S1 คือสโตรคปัจจุบัน

S2 คือสโตรคถัดไป

S1_Xmin คือพิกัด X ที่มีค่าน้อยที่สุดในข้อมูลจริงที่ได้จากการเขียนของสโตรคปัจจุบัน

S1_Ymin คือพิกัด Y ที่มีค่าน้อยที่สุดในข้อมูลจริงที่ได้จากการเขียนของสโตรคปัจจุบัน

S1_Xmax คือพิกัด X ที่มีค่ามากที่สุดในข้อมูลจริงที่ได้จากการเขียนของสโตรคปัจจุบัน

S1_Ymax คือพิกัด Y ที่มีค่ามากที่สุดในข้อมูลจริงที่ได้จากการเขียนของสโตรคปัจจุบัน

S2_Xmin คือพิกัด X ที่มีค่าน้อยที่สุดในข้อมูลจริงที่ได้จากการเขียนของสโตรคถัดไป

S2_Ymin คือพิกัด Y ที่มีค่าน้อยที่สุดในข้อมูลจริงที่ได้จากการเขียนของสโตรคถัดไป

S2_Xmax คือพิกัด X ที่มีค่ามากที่สุดในข้อมูลจริงที่ได้จากการเขียนของสโตรคถัดไป

S2_Ymax คือพิกัด Y ที่มีค่ามากที่สุดในข้อมูลจริงที่ได้จากการเขียนของสโตรคถัดไป

3.2.2.1.2 ฟังก์ชันในการประมวลผลข้อมูลตัวอักษรเบื้องต้น(Prepocessing)

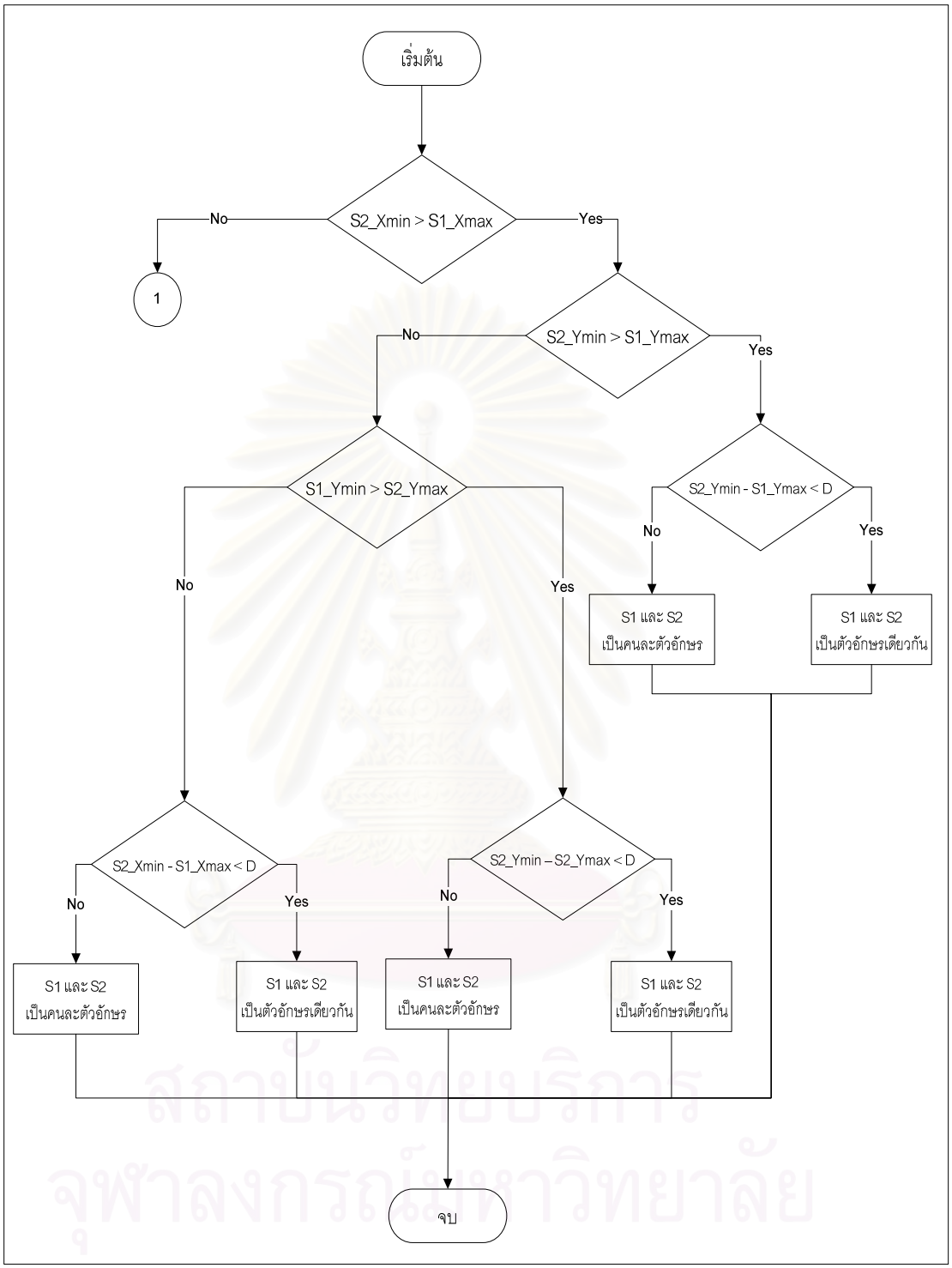
เนื่องจากข้อมูลลายมือที่ได้จากการเขียนจะประกอบด้วยจุดต่างๆในแนวระนาบ xy ซึ่งข้อมูลที่ได้นั้นอาจจะมีข้อมูลที่เกิดจากความคลาดเคลื่อนจากตัวอุปกรณ์เอง หรือความคลาดเคลื่อนในการจรดปากกา เป็นต้น ทำให้ข้อมูลที่ได้ไม่สมบรูณ์ จึงจำเป็นต้องใช้ฟังก์ชันนี้ทำการปรับปรุงข้อมูลเบื้องต้นเพื่อกำจัดความคลาดเคลื่อนต่างๆให้เหลือน้อยที่สุด และทำให้ข้อมูลอยู่ในรูปแบบที่สามารถนำไปหาลักษณะเด่นได้โดยง่าย ซึ่งมีขั้นตอนดังนี้

1) การทำนอร์มอลไลซ์ (Normalization) เป็นการปรับขนาดตัวอักษรลายมือให้มีขนาดเท่ากันและย้ายกรอบของตัวอักษรไปยังจุดกำเนิดเดียวกัน เพื่อไม่ให้เกิดความแตกต่างในด้านของขนาด และตำแหน่งการเขียนในแต่ละครั้ง โดยใช้วิธีการปรับให้อัตราส่วนของตัวอักษรคงที่

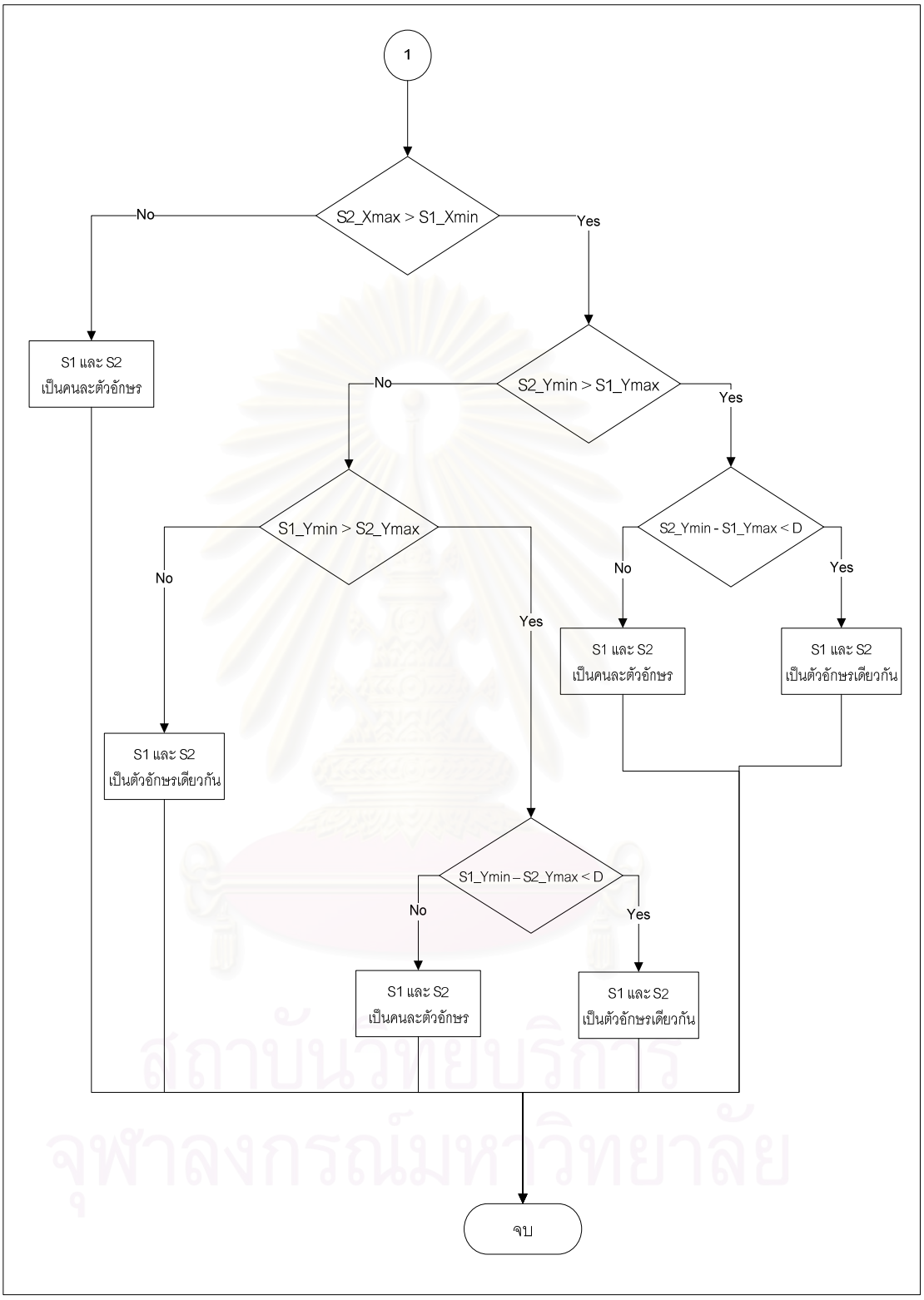
2) ทำให้เรียบโดยการเฉลี่ยให้เป็นแนวเส้นตรง (Straight line average smoothing) เป็นการทำให้เส้นมีความเรียบมากขึ้น วิธีนี้จะพิจารณาจากจุด 3 จุด โดยใช้วิธีการเลื่อนจุดที่กำลังพิจารณาไปยังตำแหน่งที่เป็นแนวระหว่างจุดหน้าและจุดหลัง

3) กรองด้วยระยะห่างที่น้อยที่สุด (Minimum distance filtering) เป็นการกำจัดจุดที่อยู่ชิดกันมากเกินไป โดยกำจัดจุดที่อยู่ติดกันต่ำกว่า 25 หน่วย (เทียบจากขนาดตัวอักษร 600x600 หน่วย) เพื่อลดจำนวนจุดที่ซ้ำซ้อนออกไป ยกเว้นจุดที่เป็นจุดเริ่มต้นและสิ้นสุด

4) ทำนอร์มอลไลซ์อีกรอบ เพื่อให้กรอบของตัวอักษรอยู่จุดกำเนิดหลังการกรองด้วยระยะห่างที่น้อยที่สุด



รูปที่ 3.8 การเปรียบเทียบระยะห่างระหว่างขอบเขตของแต่ละสโตรค



รูปที่ 3.8 การเปรียบเทียบระยะห่างระหว่างขอบเขตของแต่ละสโตรค (ต่อ)

3.2.2.1.3 ฟังก์ชันในการหาลักษณะเด่นของตัวอักษร (Feature Extraction)

เป็นการนำข้อมูลที่ผ่านการประมวลผลเบื้องต้น มาหาลักษณะเฉพาะของตัวอักษรภาษาไทยแต่ละตัว เพื่อให้สามารถจำแนกตัวอักษรแต่ละตัวได้ถูกต้อง โดยมีขั้นตอนดังต่อไปนี้

1) หารหัสลูกโซ่ เป็นการหาทิศทางของจุดแต่ละจุด เพื่อหาทิศทางในการเขียนตัวอักษร

2) หาจำนวนสโตรค เป็นการนับจำนวนสโตรคภายในตัวอักษรแต่ละตัวอักษร เนื่องจากตัวอักษรภาษาไทยแบ่งออกได้เป็น 2 ประเภทคือ เป็นตัวอักษรที่ประกอบด้วย 1 สโตรค และ ประกอบด้วย 2 สโตรค

3) หาจำนวนวงรอบ เป็นการหาจำนวนหัวของตัวอักษร โดยใช้วิธีหาจากจุดตัดของเส้นตรงสองเส้น แต่สำหรับวงรอบแรกของตัวอักษรจะใช้วิธีหาจากจุดตัดของเส้นตรงสองเส้นก่อน ถ้าไม่พบจุดตัดในช่วงแรกของสโตรค (5 จุดแรกของสโตรค) จึงหาวงรอบโดยหาจากการเปลี่ยนแปลงทิศทางรหัสลูกโซ่ แทน

4) หาโซนของวงรอบที่พบ เป็นการแบ่งตัวอักษรออกเป็น 5 โซนและตรวจสอบว่าทั้ง 5 โซน แต่ละโซนมีโซนใดบ้างที่พบวงรอบภายในโซน

5) หาทิศทางการวนของวงรอบ เป็นการหาทิศทางของวงรอบที่พบแต่ละวงรอบว่าทิศทางการวนเป็นทิศทวนเข็มหรือตามเข็มนาฬิกา โดยจำกัดจำนวนวงรอบในแต่ละตัวอักษรไม่เกิน 3 วงรอบ

6) หาเปอร์เซ็นต์ของจุดในแต่ละโซน เป็นการแบ่งตัวอักษรออกเป็น 5 โซนและหาเปอร์เซ็นต์ของจำนวนจุดที่ประกอบเป็นตัวอักษรภายในแต่ละโซน โดยเทียบจากจำนวนจุดทั้งหมดของตัวอักษร ซึ่งตัวเลขที่ได้จะเป็นเลขจำนวนเต็ม

7) หาข้อมูลเวคเตอร์จุดเด่นทั้งหมด และนำเวคเตอร์จุดเด่นมาหารหัสลูกโซ่ รหัสโซนของจุดเริ่มต้น รหัสโซนของจุดปลาย

8) หารหัสโซนของแต่ละจุดที่ผ่านการประมวลผลเบื้องต้นแล้ว

3.2.2.1.4 ฟังก์ชันเข้ารหัส (Encoding)

เป็นการนำข้อมูลที่ได้จากการหาลักษณะเด่นของตัวอักษรมาจัดเรียงให้อยู่ในรูปที่พร้อมจะส่งให้ข่ายงานประสาททำการประมวลผล โดยจัดเรียงในรูปแบบของอาร์เรย์ขนาด 205 รายการ โดยมีข้อมูลดังต่อไปนี้ และการจัดเรียงลำดับข้อมูลตามรูปแบบในตารางที่ 3.1

- จำนวนสโตรค โดยมีรหัส ดังนี้

1 เมื่อตัวอักษรมีจำนวน 1 สโตรค

2 เมื่อตัวอักษรมีจำนวน 2 สโตรค

- จำนวนวงรอบ โดยมีรหัสดังนี้

0 เมื่อตัวอักษรไม่มีวงรอบ

1 เมื่อตัวอักษรมีจำนวน 1 รอบ

2 เมื่อตัวอักษรทีจำนวน 2 วงรอบ

3 เมื่อตัวอักษรมีจำนวน 3 วงรอบ

- ตำแหน่งของวงรอบในแต่ละโซน (ทั้งหมด 5 โซน) โดยมีรหัสดังนี้

0 เมื่อไม่พบวงรอบในโซนนั้น

1 เมื่อพบวงรอบในโซนนั้น

- ทิศทางการวนของวงรอบแต่ละวงรอบ เรียงตามลำดับ (โดยจำกัดอยู่ที่ 3 ลำดับต่อตัวอักษร คือวงรอบที่1 วงรอบที่2 วงรอบที่3) โดยมีรหัสดังนี้

0 เมื่อไม่พบวงรอบ

1 เมื่อวงรอบมีทิศทางการวนตามเข็มนาฬิกา

2 เมื่อวงรอบมีทิศทางการวนทวนเข็มนาฬิกา

- เปอร์เซ็นต์ของจุดในแต่ละโซน (ทั้งหมด 5 โซน)

- ชุดข้อมูลเวคเตอร์จุดเด่น ซึ่งประกอบด้วย รหัสลูกโซ่ รหัสโซนจุดเริ่มต้นของเวคเตอร์ และรหัสโซนจุดปลายของเวคเตอร์ของจุดเด่นทั้งหมด

- รหัสโซนของทุกจุดของตัวอักษร

3.2.2.1.1.5 ฟังก์ชันเรียกใช้ข่ายงานประสาท

เป็นฟังก์ชันสำหรับส่งข้อมูลที่ผ่านขั้นตอนการเข้ารหัสเรียบร้อยให้กับข่ายงานประสาท เพื่อให้ข่ายงานประสาทนำข้อมูลไปประมวลผล และส่งตัวอักษรที่รู้จำได้เป็นผลลัพธ์กลับคืนมา

3.2.2.1.1.6 ฟังก์ชันเปรียบเทียบผลลัพธ์ของตัวอักษร

เป็นฟังก์ชันสำหรับนำผลรับที่ได้จากข่ายงานประสาท ซึ่งประกอบด้วยค่าของโหนดเอาท์พุตทั้งหมด 67 ค่า มาเปรียบเทียบเพื่อหาผลลัพธ์ของตัวอักษร โดยใช้เทคนิคการหาค่าตำแหน่งที่ของโหนดเอาท์พุตที่ให้ค่าของข้อมูลสูงที่สุด จากนั้นนำค่าตำแหน่งที่ได้มาทำการ

เปรียบเทียบกับข้อมูลในไฟล์โค้ด (ตัวอย่างไฟล์โค้ดดังแสดงในภาคผนวก ข.) เพื่อค้นหารหัส Unicode ของตัวอักษร จากการประมวลผลด้วยวิธีนี้จึงทำให้ไม่พบผลการรู้จำตัวอักษรไม่ได้ของระบบ เนื่องจากระบบจะให้ค่าตัวอักษรที่ใกล้เคียงมากที่สุดแทน

ตารางที่ 3.1 สรุปข้อมูลเข้ารหัสที่ใช้ส่งให้ข่ายงานประสาทตามลำดับที่ของอาร์เรย์

ลำดับที่	ค่าของข้อมูลในแต่ละลำดับ	หมายเหตุ
1	จำนวนสโตรค	
2	จำนวนวงรอบทั้งหมดภายในตัวอักษร	
3	ตำแหน่งของวงรอบในโซนที่ 1	
4	ตำแหน่งของวงรอบในโซนที่ 2	
5	ตำแหน่งของวงรอบในโซนที่ 3	
6	ตำแหน่งของวงรอบในโซนที่ 4	
7	ตำแหน่งของวงรอบในโซนที่ 5	
8	รหัสทิศทางการวนของวงรอบที่ 1	
9	รหัสทิศทางการวนของวงรอบที่ 2	
10	รหัสทิศทางการวนของวงรอบที่ 3	
11	เปอร์เซ็นต์ของจุดในโซนที่ 1	
12	เปอร์เซ็นต์ของจุดในโซนที่ 2	
13	เปอร์เซ็นต์ของจุดในโซนที่ 3	
14	เปอร์เซ็นต์ของจุดในโซนที่ 4	
15	เปอร์เซ็นต์ของจุดในโซนที่ 5	
16	รหัสลูกโซ่ของจุดเด่นที่ 1	ข้อมูลเวกเตอร์จุดเด่น จัดเรียงตั้งแต่ลำดับที่ 16-105 หากข้อมูลมีไม่ถึงตำแหน่งที่ 105 ให้เติมค่าตำแหน่งที่เหลือจนถึงตำแหน่งที่ 105 ด้วยจำนวนสโตรค
17	รหัสโซนจุดเริ่มต้นของเวคเตอร์ของจุดเด่นที่ 1	
18	รหัสโซนจุดปลายของเวคเตอร์ของจุดเด่นที่ 1	
19	รหัสลูกโซ่ของจุดเด่นที่ 2	
20	รหัสโซนจุดเริ่มต้นของเวคเตอร์ของจุดเด่นที่ 2	
21	รหัสโซนจุดปลายของเวคเตอร์ของจุดเด่นที่ 2	
22	รหัสลูกโซ่ของจุดเด่นที่ 3	

ตารางที่ 3.1 สรุปข้อมูลเข้ารหัสที่ใช้ส่งให้ข่ายงานประสาทตามลำดับที่ของอาร์เรย์ (ต่อ)

ลำดับที่	ค่าของข้อมูลในแต่ละลำดับ	หมายเหตุ
23	รหัสโซนจุดเริ่มต้นของเวคเตอร์ของจุดเด่นที่ 3	
24	รหัสโซนจุดปลายของเวคเตอร์ของจุดเด่นที่ 3	
25	รหัสลูกโซ่ของจุดเด่นที่ 4	
26	รหัสโซนจุดเริ่มต้นของเวคเตอร์ของจุดเด่นที่ 4	
27	รหัสโซนจุดปลายของเวคเตอร์ของจุดเด่นที่ 4	
28	รหัสลูกโซ่ของจุดเด่นที่ 5	
29	รหัสโซนจุดเริ่มต้นของเวคเตอร์ของจุดเด่นที่ 5	
30	รหัสโซนจุดปลายของเวคเตอร์ของจุดเด่นที่ 5	
31	……..	
…	……..	
…	……..	
105	……..	
106	รหัสโซนของจุดที่ 1 ของตัวอักษร	รหัสโซนของแต่ละจุดจัดเรียงตั้งแต่ลำดับที่ 106-205 หากข้อมูลมีไม่ถึงตำแหน่งที่ 205 ให้เติมค่าตำแหน่งที่เหลือจนถึงตำแหน่งที่ 205 ด้วยจำนวนวงรอบ
107	รหัสโซนของจุดที่ 2 ของตัวอักษร	
108	รหัสโซนของจุดที่ 3 ของตัวอักษร	
109	รหัสโซนของจุดที่ 4 ของตัวอักษร	
110	……..	
…	……..	
…	……..	
205	……..	

3.2.2.1.1.7 ฟังก์ชันตรวจสอบตัวอักษรก่อนหน้า (Check Previous Character)

เป็นฟังก์ชันเพื่อตรวจสอบตัวอักษรที่รู้จำได้ทั้งหมดในกรณีที่ผู้ใช้เลือกโหมดเขียนครั้งละหลายตัวอักษร โดยจะทำการตรวจสอบและปรับปรุงตัวอักษรให้เป็นตัวอักษรที่ถูกต้อง

เนื่องจากตัวอักษรบางตัวอักษรประกอบด้วยสโตรค 2 สโตรค ซึ่งเวลาเขียนปกติสโตรคทั้งสองอาจจะห่างกันได้ ทำให้ฟังก์ชันคัดแยกสโตรคไม่สามารถแบ่งแยกได้ถูกต้อง จึงจำเป็นต้องมีการปรับปรุงอีกครั้งด้วยฟังก์ชันนี้ ตัวอักษรที่ฟังก์ชันนี้ทำการปรับปรุงมีอยู่ด้วยกัน 2 ตัวอักษรคือสระอะ และ สระแอ ซึ่งเป็นตัวอักษรที่ประกอบด้วย 2 สโตรค และมีลักษณะพิเศษดังนี้

1) แต่ละสโตรคภายในตัวอักษรสามารถเขียนห่างกันได้

2) ลักษณะการเขียนของสโตรคที่หนึ่งเหมือนกับสโตรคที่สอง

3) แต่ละสโตรคมีลักษณะการเขียนตรงกันตัวอักษรภาษาไทยตัวอื่น เช่น สระแอ แต่ละสโตรคจะเหมือนสระเอ สระอะ แต่ละสโตรคจะเหมือนไม้หันอากาศ เป็นต้น

4) ข้อกำหนดของภาษาไทยไม่อนุญาตให้เขียนตัวอักษรสระเอ ติดกัน หรือไม้หันอากาศติดกัน

จึงทำให้สามารถนำลักษณะพิเศษดังกล่าวมาใช้ในการตรวจสอบตัวอักษรที่รู้จำได้ และทำการปรับเปลี่ยนให้เป็นตัวอักษรที่ถูกต้อง ดังรูปที่ 3.9 โดยมีวิธีการดังนี้ (ส่วนของโปรแกรมดังแสดงในภาคผนวก ก.)

ทำขั้นตอนต่อไปนี้สำหรับแต่ละตัวอักษร จนครบทุกตัวอักษร :

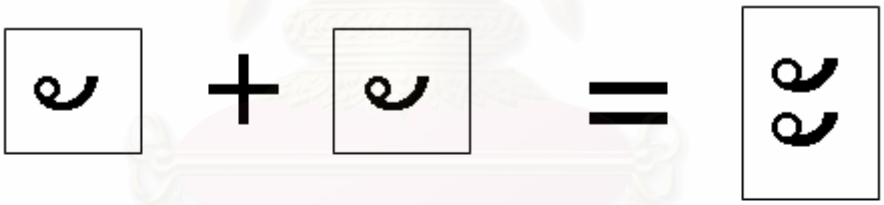
ถ้าตัวอักษรปัจจุบันเป็นไม้หันอากาศและตัวอักษรก่อนหน้าเป็นไม้หันอากาศ

ให้แทนที่ตัวอักษรทั้ง 2 ตัว ด้วยสระอะแทน

มิฉะนั้น ถ้าตัวอักษรปัจจุบันเป็นสระเอและตัวอักษรก่อนหน้าเป็นสระเอ

ให้แทนที่ตัวอักษรทั้ง 2 ตัว ด้วยสระแอแทน

มิฉะนั้น ให้คงตัวอักษรเดิม



รูปที่ 3.9 การทำงานของฟังก์ชันตรวจสอบตัวอักษรก่อนหน้า

3.2.2.2. ไลบารี่การรู้จำลายมือภาษาอังกฤษของไมโครซอฟท์

เนื่องจากไมโครซอฟท์ได้พัฒนาระบบการรู้จำลายมือภาษาต่างๆบนวินโดว์เอ็กซ์พี แท็บเล็ตพีซี และได้จัดทำเป็นไลบารี่การรู้จำลายมือเพื่อให้นักพัฒนาสามารถนำคุณสมบัติการรู้จำ

ลายมือที่ไมโครซอฟท์พัฒนาไว้นี้ไปใช้พัฒนาร่วมกับระบบต่างๆ ผู้วิจัยจึงได้นำไลบารี่ดังกล่าวมาใช้ในงานวิจัยนี้ โดยใช้ในส่วนของการรู้จำลายมือภาษาอังกฤษเท่านั้น ไลบารี่ที่เรียกใช้คือ ไมโครซอฟท์แท็บเล็ตพีซี เอพีไอ (Microsoft Tablet PC API) ซึ่งได้จากการติดตั้งไมโครซอฟท์แท็บเล็ตพีซีซอฟท์แวร์รีสอร์ซคิต (Microsoft Tablet PC Software Resource Kit)

3.2.3 การออกแบบส่วนของฟังก์ชันข่ายงานประสาทแบบแพร่กระจายย้อนกลับ (Back Propagation)

ได้ทำการพัฒนาฟังก์ชันข่ายงานประสาทแบบแพร่กระจายย้อนกลับเฉพาะส่วนที่จำเป็นต้องใช้กับระบบเท่านั้น ซึ่งก็คือส่วนของการหาผลลัพธ์ของการรู้จำเท่านั้น มิได้พัฒนาในส่วนที่ใช้สอนข่ายงานประสาท เนื่องจากในระบบไม่อนุญาติให้ผู้ใช้งานสอนข่ายงานประสาทใหม่ เพราะว่าการสอนข่ายงานประสาทใหม่จนกระทั่งได้ผลลัพธ์จะใช้เวลาในการสอนนานมาก ผู้วิจัยได้ออกแบบโดยแบ่งออกเป็นส่วนย่อยดังรายละเอียดในหัวข้อ 3.3.2

3.3 แผนภาพคลาสและความสัมพันธ์

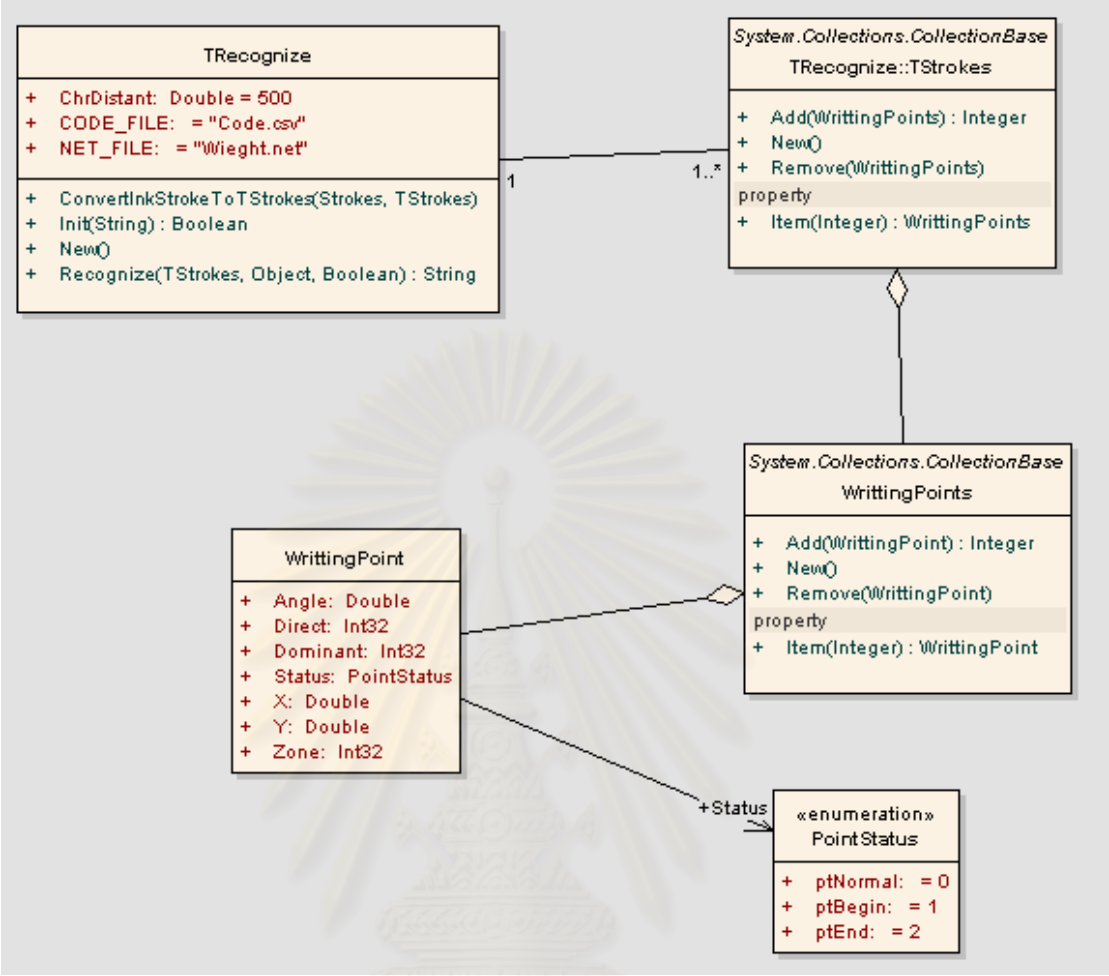
เนื่องจากผู้วิจัยได้ออกแบบไลบรารีสำหรับรู้จำลายมือภาษาไทยและไลบรารีข่ายงานประสาทแบบแพร่กระจายย้อนกลับ ในรูปแบบที่สามารถนำกลับมาใช้ใหม่ได้ โดยมีวิธีการเรียกใช้ดังตัวอย่างในภาคผนวก ค. และมีแผนภาพคลาสและความสัมพนธ์ระหว่างคลาส แบ่งออกเป็น 2 แผนภาพ ดังต่อไปนี้

- แผนภาพคลาสและความสัมพันธ์ระหว่างคลาสในไลบรารีสำหรับรู้จำลายมือภาษาไทย
- แผนภาพคลาสและความสัมพันธ์ระหว่างคลาสในไลบรารีข่ายงานประสาทแบบแพร่กระจายย้อนกลับ

3.3.1 แผนภาพคลาสและความสัมพันธ์ระหว่างคลาสในไลบรารี สำหรับรู้จำลายมือภาษาไทย

เป็นแผนภาพคลาสและความสัมพันธ์ระหว่างคลาสต่างๆที่เกี่ยวข้องกับการรู้จำลายมือภาษาไทย ดังรูปที่ 3.10

3.3.1.1 คลาส PointStatus เป็นค่าที่ใช้ในการแสดงสถานะของจุดที่ได้จากลายมือเขียน



รูปที่ 3.10 แผนภาพความสัมพันธ์ระหว่างคลาสไลบรารีสำหรับรู้จำลายมือภาษาไทย

3.3.1.2 คลาส WrittingPoint ทำหน้าที่เก็บข้อมูลของจุดแต่ละจุดจากลายมือเขียนที่รับเข้ามาและข้อมูลที่ได้จากการประมวลตัวอักษร

3.3.1.3 คลาส WrittingPoints เป็นคอลเลชันของ WritingPoint ซึ่งทำหน้าที่เก็บรวบรวมจุดต่างๆ ประกอบด้วยเมท็อดดังตารางที่ 3.2

ตารางที่ 3.2 เมท็อดและหน้าที่ของเมท็อดภายในคลาส WrittingPoints

ชื่อคลาส	เมท็อด	หน้าที่
WrittingPoints	New	สร้างอ็อบเจ็กเพื่อเก็บจุดที่เกิดจากการเขียน
WrittingPoints	Add	เพิ่มข้อมูลจุดที่เกิดจากการเขียน
WrittingPoints	Remove	ลบข้อมูลจุดที่เกิดจากการเขียน

3.3.1.4 คลาส TStrokes เป็นคอลเลชันของ WritingPoints คอลเลชัน ทำหน้าที่เก็บรวบรวมสโตรคทั้งหมดในการเขียนแต่ละครั้ง ซึ่งแต่ละสโตรคประกอบไปด้วยจุดต่างๆ ประกอบด้วยเมท็อดดังตารางที่ 3.3

ตารางที่ 3.3 เมท็อดและหน้าที่ของเมท็อดภายในคลาส TStrokes

ชื่อคลาส	เมท็อด	หน้าที่
TStrokes	New	สร้างอ็อบเจ็กเพื่อเก็บสโตรคทั้งหมด
TStrokes	Add	เพิ่มข้อมูลจุดทั้งหมดของสโตรคใหม่
TStrokes	Remove	ลบสโตรค

3.3.1.5 คลาส TRecognize ทำหน้าที่รับข้อมูลลายมือเขียนและค่าต่างๆที่เกี่ยวข้องกับการประมวลผลการรู้จำ เช่น ระยะห่างระหว่างตัวอักษรแต่ละตัวอักษร ชื่อไฟล์น้ำหนัก ชื่อไฟล์โค้ด เป็นต้น และทำหน้าที่ประมวลผลลายมือเขียนที่รับเข้ามา โดยจะส่งค่าตัวอักษรที่รู้จำได้กลับคืนไปยังฟังก์ชันที่เรียกใช้ ซึ่งประกอบด้วยเมท็อดดังตารางที่ 3.4

3.3.2 แผนภาพคลาสและความสัมพันธ์ระหว่างคลาสในไลบรารีข่ายงานประสาทแบบแพร่กระจายย้อนกลับ

เป็นแผนภาพคลาสและความสัมพันธ์ระหว่างคลาสต่างๆที่เกี่ยวข้องกับข่ายงานประสาท ดังรูปที่ 3.11

3.3.2.1 คลาส Neuron เก็บข้อมูลของโหนดแต่ละโหนดภายในข่ายงานประสาท เช่น ตำแหน่งของโหนด ค่าภายในโหนด ค่าหน่วยเอนเอียง ค่าน้ำหนัก เป็นต้น ซึ่งประกอบด้วยเมท็อดดังตารางที่ 3.5

3.3.2.2 คลาส Neurons เก็บรวบรวมโหนดทั้งหมดภายในข่ายงานประสาท ซึ่งประกอบด้วยเมท็อดดังตารางที่ 3.6

3.3.2.3 คลาส Connector เก็บโหนดอินพุต และค่าน้ำหนักระหว่างโหนดอินพุต กับโหนดเอาต์พุต

ตารางที่ 3.4 เมท็อดและหน้าที่ของเมท็อดภายในคลาส TRecognize

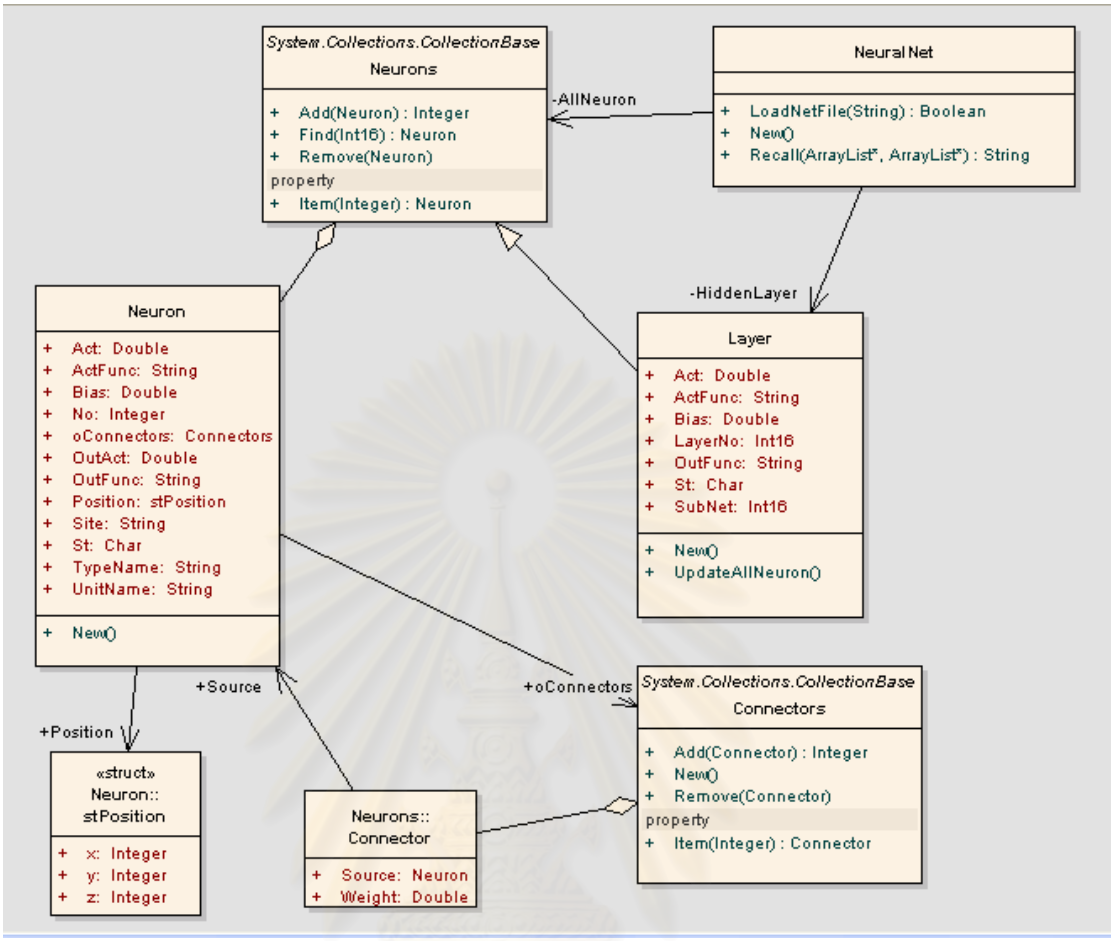
ชื่อคลาส	เมท็อด	หน้าที่
TRecognize	New	สร้างอ็อบเจ็กสำหรับประมวลผลการรู้จำ
TRecognize	Init	1. รับชื่อไดเรกทอรี่ที่เก็บไฟล์โค้ดและไฟล์น้ำหนัก 2. อ่านข้อมูลจากไฟล์โค้ดที่กำหนดให้ หากไม่มีการกำหนดให้จะอ่านจากไฟล์โค้ดที่ชื่อว่า Code.csv ซึ่งอยู่ภายใต้ไดเรกทอรี่ที่รับค่ามา 3. สั่งให้ข่ายงานประสาททำการอ่านข้อมูลจากไฟล์น้ำหนักที่กำหนดให้ หากไม่มีการกำหนดให้จะอ่านจากไฟล์น้ำหนักที่ชื่อว่า Wieght.net ซึ่งอยู่ภายใต้ไดเรกทอรี่ที่รับค่ามา
TRecognize	ConvertInkStrokeToTStrokes	ทำการแปลงสโตรคในรูปแบบของไมโครซอฟท์อิงค์คอลเลชันให้อยู่ในรูปแบบสโตรคคอลเลชันของระบบ
TRecognize	Recognize	ทำการประมวลผลสโตรคทั้งหมด โดยคืนค่ากลับมาเป็นตัวอักษรที่รู้จำได้ทั้งหมด

ตารางที่ 3.5 เมท็อดและหน้าที่ของเมท็อดภายในคลาส Neuron

ชื่อคลาส	เมท็อด	หน้าที่
Neuron	New	สร้างอ็อบเจ็กสำหรับเก็บข้อมูลของโหนดแต่ละโหนด

ตารางที่ 3.6 เมท็อดและหน้าที่ของเมท็อดภายในคลาส Neurons

ชื่อคลาส	เมท็อด	หน้าที่
Neurons	New	สร้างอ็อบเจ็กสำหรับเก็บรวบรวมโหนดทั้งหมด
Neurons	Add	เพิ่มโหนดในข่ายงานประสาท
Neurons	Find	ค้นหาโหนดในข่ายงานประสาท
Neurons	Remove	ลบโหนดจากข่ายงานประสาท



รูปที่ 3.11 แผนภาพความสัมพันธ์ระหว่างคลาสไลบรารีข่ายงานประสาทแบบแพร่กระจายย้อนกลับ

3.3.2.4 คลาส Connectors เป็นคอลเลชันของ Connector ทำหน้าที่เก็บโหนดอินพุต และน้ำหนักระหว่างโหนดอินพุตกับโหนดเอาต์พุตทั้งหมดภายในข่ายงานประสาท ซึ่งประกอบด้วยเมท็อดดังตารางที่ 3.7

ตารางที่ 3.7 เมท็อดและหน้าที่ของเมท็อดภายในคลาส Connectors

ชื่อคลาส	เมท็อด	หน้าที่
Connectors	New	สร้างอ็อบเจ็กสำหรับเก็บค่า Connector ทั้งหมด
Connectors	Add	เพิ่ม Connector อ็อบเจ็กในข่ายงานประสาท
Connectors	Remove	ลบ Connector อ็อบเจ็กจากข่ายงานประสาท

3.3.2.5 คลาส Layer เก็บข้อมูลของโหนดทั้งหมดในแต่ละเลเยอร์ เช่น โหนดทั้งหมดในเลเยอร์อินพุต เลเยอร์ซ่อน เลเยอร์เอาต์พุต เป็นต้น และทำหน้าคำนวณหาค่าของโหนดในเลเยอร์ถัดไปทั้งหมด ซึ่งประกอบด้วยเมท็อดดังตารางที่ 3.8

ตารางที่ 3.8 เมท็อดและหน้าที่ของเมท็อดภายในคลาส Layer

ชื่อคลาส	เมท็อด	หน้าที่
Layer	New	สร้างอ็อบเจ็กสำหรับเก็บข้อมูลของโหนดทั้งหมดในแต่ละเลเยอร์
Layer	UpdateAllNeuron	ทำการคำนวณหาค่าของโหนดทั้งหมดในเลเยอร์ถัดไป

3.3.2.6 คลาส NeuralNet เก็บข้อมูลเลเยอร์ทั้งหมดของข่ายงานประสาท และทำหน้าที่อ่านไฟล์น้ำหนัก ป้อนข้อมูลเข้าเลเยอร์อินพุต ประมวลผลและส่งผลจากเอาต์พุตเลเยอร์กลับคืน ซึ่งประกอบด้วยเมท็อดดังตารางที่ 3.9

ตารางที่ 3.9 เมท็อดและหน้าที่ของเมท็อดภายในคลาส NeuralNet

ชื่อคลาส	เมท็อด	หน้าที่
NeuralNet	New	สร้างอ็อบเจ็กข่ายงานประสาท
NeuralNet	LoadNetFile	อ่านข้อมูลจากไฟล์น้ำหนักที่กำหนดให้ หากไม่มีการกำหนดให้จะอ่านจากไฟล์นำหนักที่ชื่อว่า Wieght.net ภายในไดเรกทอรี่เดียวกับโปรแกรม
NeuralNet	Recall	1. ทำการป้อนข้อมูลที่รับมาเข้าแต่ละโหนดภายในเลเยอร์อินพุต 2. สั่งให้แต่ละเลเยอร์ทำการประมวลผล เพื่อให้ได้ค่าของเลเยอร์เอาต์พุต

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

3.4 สภาพแวดล้อมในการพัฒนาเครื่องมือ

ระบบการรู้จำลายมือออนไลน์สำหรับตัวอักษรภาษาไทยบนแท็บเล็ตพีซีนี้ ได้ทำการพัฒนาโปรแกรมภายใต้สภาพแวดล้อมดังนี้

1) พัฒนาระบบโดยใช้โปรแกรม

1.1) วิลชวลเบสิค ดอทเน็ต (Visual Basic .NET) ในชุดของวิลชวลสตูดิโอ เวอร์ชัน 2003 (Visual Studio 2003)

1.2) ชุดสำหรับนักพัฒนาโปรแกรมบนไมโครซอฟท์วินโดว์เอ็กซ์พี แท็บเล็ตพีซี เวอร์ชัน 1.7 (Microsoft Windows XP Tablet PC Edition Development Kit 1.7) [16]

1.3) ชุดสำหรับการรู้จำลายมือของไมโครซอฟท์วินโดว์เอ็กซ์พี เวอร์ชัน 2003 (Microsoft Windows XP Tablet PC Edition 2003 Recognizer Pack)

2) ใช้ไฟล์น้ำหนักของข่ายงานประสาทตามรูปแบบของมหาวิทยาลัยสตุทการ์ท ดังตัวอย่างในภาคผนวก ข.

3) ใช้เครื่องคอมพิวเตอร์แท็บเล็ตพีซีในการพัฒนา

4) พัฒนาบนระบบปฏิบัติการไมโครซอฟท์วินโดว์เอ็กซ์พีสำหรับแท็บเล็ตพีซี เวอร์ชัน 2005 (Microsoft Window XP Tablet PC Edition 2005 Service Pack2)

5) ใช้ไลบารี่ของไมโครซอฟท์ออฟฟิต 2003 (Microsoft Office 2003)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

การทดสอบการใช้งาน

ในบทนี้ จะกล่าวถึงการเตรียมการทดสอบ วิธีการทดสอบ ผลการทดสอบระบบ และการวิเคราะห์ผลการทดสอบระบบการรู้จำลายมือออนไลน์สำหรับตัวอักษรภาษาไทยบนแท็บเล็ตพีซี

4.1 การเตรียมการทดสอบ

เนื่องจากระบบที่พัฒนาขึ้นนี้ได้ถูกออกแบบให้สามารถรู้จำลายมือจากผู้ใช้อย่างน้อย 3 ท่าน และเพื่อเปรียบเทียบผลการรู้จำกับงานวิจัยเดิมของนายปกรณ์ บุพศิริ และเทียบผลการรู้จำในกรณีที่มีลายมือผู้ทดสอบในการสอนข่ายงานประสาทกับกรณีที่ไม่มีลายมือผู้ทดสอบในการสอนข่ายงานประสาท ผู้วิจัยจึงได้เตรียมสิ่งที่จะใช้ในการทดสอบระบบดังนี้

1) ผู้ทดสอบระบบจำนวน 3 ท่าน ซึ่งในการทดสอบระบบทั้งหมดจะใช้ลายมือของผู้ทดสอบ 3 ท่านนี้ในการทดสอบ

2) ไฟล์น้ำหนัก 2 ไฟล์ ซึ่งได้จากการสอนข่ายงานประสาท ด้วยโปรแกรมเครื่องมือซอฟต์แวร์สำหรับระบบการรู้จำลายมือเขียนแบบออนไลน์ โดยใช้ข่ายงานประสาทแบบแพร่กระจายย้อนกลับ ของคุณกันตา กิติยานันท์ ดังนี้

- ไฟล์น้ำหนักเดิม เป็นไฟล์น้ำหนักที่ไม่มีลายมือของผู้ทดสอบทั้ง 3 ท่านในการสอนข่ายงานประสาท เนื่องจากไฟล์น้ำหนักนี้เป็นไฟล์ที่ได้จากงานวิจัยเดิมของนายปกรณ์ บุพศิริ ซึ่งได้จากการสอนข่ายงานประสาทด้วยลายมือของนายปกรณ์ บุพศิริ เท่านั้น จำนวน 60 ชุด โดยแบ่งเป็น 30 ชุดสำหรับสอน 30 ชุดสำหรับทดสอบไฟล์น้ำหนัก ให้ผลการรู้จำ 96.6 เปอร์เซนต์

- ไฟล์น้ำหนักใหม่ เป็นไฟล์น้ำหนักที่มีลายมือของผู้ทดสอบทั้ง 3 ท่านในการสอนข่ายงานประสาท เนื่องจากไฟล์น้ำหนักนี้เป็นไฟล์ที่ได้จากการสอนข่ายงานประสาทด้วยลายมือของนายปกรณ์ บุพศิริ และผู้ทดสอบทั้ง 3 ท่าน ด้วยชุดของตัวอักษรทั้งหมด 90 ชุด โดยแบ่งเป็น 45 ชุดสำหรับสอน (ประกอบด้วยลายมือของนายปกรณ์ บุพศิริ 30 ชุด และลายมือของผู้ทดสอบ 3 ท่าน ท่านละ 5 ชุด) และ 45 ชุดสำหรับทดสอบไฟล์น้ำหนัก (ประกอบด้วยลายมือของนายปกรณ์ บุพศิริ 30 ชุด และลายมือของผู้ทดสอบ 3 ท่าน ท่านละ 5 ชุด) ให้ผลการรู้จำ 99.0 เปอร์เซนต์

4.2 วิธีการทดสอบระบบ

เนื่องจากระบบที่พัฒนาขึ้นนี้ได้ถูกออกแบบให้สามารถรู้จำตัวอักษรที่เขียนครั้งละตัวอักษร และครั้งละหลายตัวอักษร ผู้วิจัยจึงได้แบ่งการทดสอบออกเป็น 2 ขั้นตอนดังนี้

4.2.1 การทดสอบการรู้จำตัวอักษรครั้งละตัวอักษร

เพื่อเปรียบเทียบผลการรู้จำกับงานเดิม และเทียบผลการรู้จำในกรณีที่มีลายมือผู้ทดสอบในการสอนข่ายงานประสาทกับกรณีที่ไม่มีลายมือผู้ทดสอบในการสอนข่ายงานประสาท จึงได้แบ่งการทดสอบการรู้จำตัวอักษรครั้งละตัวอักษรออกเป็น 2 การทดสอบย่อยดังนี้

1) การทดสอบด้วยไฟล์น้ำหนักเดิม เป็นการทดสอบระบบโดยการใช้ไฟล์น้ำหนักเดิม ซึ่งไม่มีลายมือของผู้ทดสอบในการสอนข่ายงานประสาท ทดสอบโดยให้ผู้ทดสอบทั้ง 3 ท่าน เขียนตัวอักษรภาษาไทยทั้ง 67 ตัวอักษร จำนวนท่านละ 10 ชุด และทำการนับผลการรู้จำได้ และรู้จำผิด ทีละตัวอักษร

2) การทดสอบด้วยไฟล์น้ำหนักใหม่ เป็นการทดสอบระบบโดยใช้ไฟล์น้ำหนักใหม่ ซึ่งมีลายมือของผู้ทดสอบในการสอนข่ายงานประสาท ทดสอบโดยให้ผู้ทดสอบทั้ง 3 ท่านเขียนตัวอักษรภาษาไทยทั้ง 67 ตัวอักษร จำนวนท่านละ 10 ชุด และทำการนับผลการรู้จำได้ และรู้จำผิด ทีละตัวอักษร

ในการทดสอบแต่ละครั้งผู้ทดสอบจะต้องทำการเขียนตัวอักษรใหม่ทุกครั้ง

4.2.2 การทดสอบการรู้จำตัวอักษรครั้งละหลายตัวอักษร

เพื่อทดสอบเทคนิคการจำแนกตัวอักษรครั้งละหลายตัวอักษร จึงทำการทดสอบด้วยไฟล์น้ำหนักใหม่ ซึ่งมีลายมือของผู้ทดสอบในการสอนข่ายงานประสาท ทดสอบโดยให้ผู้ทดสอบทั้ง 3 ท่านเขียนคำว่า “กินนอน” “สาวสวย” “ภาษี” “ฐานะ” “รักกัน“ “ธุระ” “ศรัทธา” “กำลัง” “แม่จ๋า” “ญาติ” “แพ้ยา” “เมืองไทย” “ลายมือ” ทั้งหมด 13 คำ จำนวนท่านละ 10 ชุด และทำการนับผลการรู้จำได้ และรู้จำผิด ทีละตัวอักษร

4.3 ผลการทดสอบระบบ

จากการทดสอบระบบการรู้จำลายมือออนไลน์สำหรับตัวอักษรภาษาไทยบนแท็บเล็ตพีซีด้วยวิธีการทดสอบข้างต้น ได้ผลการทดสอบดังนี้

4.3.1 ผลการทดสอบการรู้จำตัวอักษรครั้งละตัวอักษร แบ่งออกเป็น 2 ส่วนดังนี้

4.3.1.1 ผลการรู้จำลายมือครั้งละตัวอักษรโดยใช้ไฟล์น้ำหนักเดิม สามารถรู้จำลายมือเขียนตัวอักษรไทยได้ถูกต้อง 73.4 เปอร์เซ็นต์ รู้จำผิด 26.6 เปอร์เซ็นต์

4.3.1.2 ผลการรู้จำลายมือครั้งละตัวอักษรโดยใช้ไฟล์น้ำหนักใหม่ สามารถรู้จำลายมือเขียนตัวอักษรภาษาไทยได้ถูกต้อง 89.4 เปอร์เซ็นต์ รู้จำผิด 10.6 เปอร์เซ็นต์ รายละเอียดผลการทดสอบดังแสดงในตารางที่ 4.1 และตารางที่ 4.2 ตามลำดับ รายละเอียดตัวอักษรที่รู้จำผิดดังแสดงในภาคผนวก ง.

4.3.2 ผลการทดสอบการรู้จำตัวอักษรครั้งละหลายตัวอักษร โดยใช้ไฟล์น้ำหนักใหม่ สามารถรู้จำลายมือเขียนตัวอักษรไทยได้ถูกต้อง 87.7 เปอร์เซ็นต์ รู้จำผิด 12.3 เปอร์เซ็นต์ รายละเอียดผลการทดสอบดังแสดงในตารางที่ 4.3

ตารางที่ 4.1 สรุปผลการรู้จำลายมือครั้งละตัวอักษร

ประเภทการรู้จำ	ไฟล์น้ำหนักเดิม		ไฟล์น้ำหนักใหม่	
	จำนวน (ตัวอักษร)	%	จำนวน (ตัวอักษร)	%
การรู้จำที่ถูกต้อง	1475	73.4	1796	89.4
การรู้จำที่ผิด	535	26.6	214	10.6

ตารางที่ 4.2 รายละเอียดผลการรู้จำลายมือครั้งละตัวอักษร

ตัวอักษร	ไฟล์น้ำหนักเดิม				ไฟล์น้ำหนักใหม่			
	รู้จำถูกต้อง		รู้จำผิด		รู้จำถูกต้อง		รู้จำผิด	
	จำนวน (ตัวอักษร)	%	จำนวน (ตัวอักษร)	%	จำนวน (ตัวอักษร)	%	จำนวน (ตัวอักษร)	%
ก	30	100.0	0	0.0	30	100.0	0	0.0
ข	8	26.7	22	73.3	24	80.0	6	20.0
ค	14	46.7	16	53.3	22	73.3	8	26.7
ฆ	30	100.0	0	0.0	29	96.7	1	3.3
ง	29	96.7	1	3.3	29	96.7	1	3.3
จ	30	100.0	0	0.0	30	100.0	0	0.0
ฉ	22	73.3	8	26.7	28	93.3	2	6.7
ช	4	13.3	26	86.7	24	80.0	6	20.0
ซ	25	83.3	5	16.7	29	96.7	1	3.3
ฌ	21	70.0	9	30.0	22	73.3	8	26.7
ญ	30	100.0	0	0.0	30	100.0	0	0.0
ฎ	1	3.3	29	96.7	13	43.3	17	56.7
ฏ	20	66.7	10	33.3	25	83.3	5	16.7
ฐ	22	73.3	8	26.7	30	100.0	0	0.0
ฑ	25	83.3	5	16.7	28	93.3	2	6.7
ฒ	26	86.7	4	13.3	30	100.0	0	0.0
ณ	17	56.7	13	43.3	29	96.7	1	3.3
ด	26	86.7	4	13.3	30	100.0	0	0.0
ต	24	80.0	6	20.0	29	96.7	1	3.3
ถ	29	96.7	1	3.3	26	86.7	4	13.3
ท	29	96.7	1	3.3	30	100.0	0	0.0
ธ	27	90.0	3	10.0	30	100.0	0	0.0
น	29	96.7	1	3.3	29	96.7	1	3.3
บ	22	73.3	8	26.7	30	100.0	0	0.0
ป	26	86.7	4	13.3	29	96.7	1	3.3

ตารางที่ 4.2 รายละเอียดผลการรู้จำลายมือครั้งละตัวอักษร (ต่อ)

ตัวอักษร	ไฟล์น้ำหนักเดิม				ไฟล์น้ำหนักใหม่			
	รู้จำถูกต้อง		รู้จำผิด		รู้จำถูกต้อง		รู้จำผิด	
	จำนวน (ตัวอักษร)	%	จำนวน (ตัวอักษร)	%	จำนวน (ตัวอักษร)	%	จำนวน (ตัวอักษร)	%
ผ	21	70.0	9	30.0	28	93.3	2	6.7
ฝ	18	60.0	12	40.0	20	66.7	10	33.3
พ	23	76.7	7	23.3	25	83.3	5	16.7
ฟ	21	70.0	9	30.0	29	96.7	1	3.3
ภ	15	50.0	15	50.0	21	70.0	9	30.0
ม	29	96.7	1	3.3	30	100.0	0	0.0
ย	22	73.3	8	26.7	29	96.7	1	3.3
ร	2	6.7	28	93.3	28	93.3	2	6.7
ล	30	100	0	0.0	29	96.7	1	3.3
ว	23	76.7	7	23.3	27	90.0	3	10.0
ศ	20	66.7	10	33.3	21	70.0	9	30.0
ษ	29	96.7	1	3.3	30	100.0	0	0.0
ส	29	96.7	1	3.3	29	96.7	1	3.3
ห	15	50.0	15	50.0	24	80.0	6	20.0
ฬ	19	63.3	11	36.7	29	96.7	1	3.3
อ	24	80.0	6	20.0	24	80.0	6	20.0
ฮ	15	50.0	15	50.0	24	80.0	6	20.0
ะ	28	93.3	2	6.7	30	100.0	0	0.0
า	30	100.0	0	0.0	30	100.0	0	0.0
ิ	23	76.7	7	23.3	27	90.0	3	10.0
ี	28	93.3	2	6.7	28	93.3	2	6.7
ึ	22	73.3	8	26.7	27	90.0	3	10.0
ื	23	76.7	7	23.3	25	83.3	5	16.7
ุ	28	93.3	2	6.7	28	93.3	2	6.7
ู	22	73.3	8	26.7	24	80.0	6	20.0

ตารางที่ 4.2 รายละเอียดผลการรู้จำลายมือครั้งละตัวอักษร (ต่อ)

ตัวอักษร	ไฟล์น้ำหนักเดิม				ไฟล์น้ำหนักใหม่			
	รู้จำถูกต้อง		รู้จำผิด		รู้จำถูกต้อง		รู้จำผิด	
	จำนวน (ตัวอักษร)	%	จำนวน (ตัวอักษร)	%	จำนวน (ตัวอักษร)	%	จำนวน (ตัวอักษร)	%
เ	25	83.3	5	16.7	30	100.0	0	0.0
แ	29	96.7	1	3.3	30	100.0	0	0.0
โ	8	26.7	22	73.3	22	73.3	8	26.7
ไ	24	80.0	6	20.0	26	86.7	4	13.3
ใ	26	86.7	4	13.3	30	100.0	0	0.0
ำ	29	96.7	1	3.3	30	100.0	0	0.0
ั	23	76.7	7	23.3	16	53.3	14	46.7
ฤ	14	46.7	16	53.3	23	76.7	7	23.3
ฦ	11	36.7	19	63.3	26	86.7	4	13.3
่	26	86.7	4	13.3	29	96.7	1	3.3
้	11	36.7	19	63.3	21	70.0	9	30.0
๊	28	93.3	2	6.7	29	96.7	1	3.3
๋	23	76.7	7	23.3	25	83.3	5	16.7
็	16	53.3	14	46.7	30	100.0	0	0.0
์	27	90.0	3	10.0	28	93.3	2	6.7
ฯ	1	3.3	29	96.7	21	70.0	9	30.0
ๆ	29	96.7	1	3.3	30	100.0	0	0.0
รวม	1475	73.4	535	26.6	1797	89.4	213	10.6

4.4 วิเคราะห์ผลการทดสอบระบบ

จากการทดสอบระบบการรู้จำลายมือออนไลน์สำหรับตัวอักษรภาษาไทยบนแท็บเล็ตพีซี สามารถสรุปผลการทดสอบโดยแบ่งออกเป็น 2 ส่วน คือ ส่วนของการทดสอบลายมือเขียนครั้งละตัวอักษร และส่วนของการทดสอบลายมือเขียนครั้งละหลายตัวอักษร ดังต่อไปนี้

ตารางที่ 4.3 ผลการรู้จำลายมือครั้งละหลายตัวอักษร

คำ	ไฟล์น้ำหนักใหม่			
	รู้จำถูกต้อง		รู้จำผิด	
	จำนวน (ตัวอักษร)	%	จำนวน (ตัวอักษร)	%
กินนอน	171	95.0	9	5.0
สาวสวย	172	95.6	8	4.4
ภาษี	111	92.5	9	7.5
ฐานะ	118	98.3	2	1.7
รักกัน	164	91.1	16	8.9
ธุระ	88	73.3	32	26.7
ศรัทธา	153	85.0	27	15.0
กำลัง	109	72.7	41	27.3
แม่จ๋า	159	88.3	21	11.7
ญาติ	117	97.5	3	2.5
แพ้ยา	117	78.0	33	22.0
เมืองไทย	208	86.7	32	13.3
ลายมือ	156	86.7	24	13.3
รวม	1843	87.7	8.25	12.3

4.4.1 ผลการทดสอบในกรณีที่ใช้ลายมือเขียนครั้งละตัวอักษร

ผลการทดสอบโดยใช้ไฟล์น้ำหนักเดิม ซึ่งมีเพียงลายมือเขียนของนายปกรณ์ บุพศิริ เพียงท่านเดียว ให้ผลการรู้จำที่ถูกต้อง 73.4% การรู้จำที่ผิด 26.6% ส่วนผลการทดสอบโดยใช้ไฟล์น้ำหนักใหม่ ซึ่งเพิ่มลายมือเขียนของผู้ทดสอบเข้าไปในขั้นตอนการสอนข่ายงานประสาท ให้ผลการรู้จำที่ถูกต้อง 89.4% การรู้จำที่ผิด 10.6%

4.4.1.1 จากผลการทดสอบพบว่าผลการทดสอบโดยใช้ไฟล์น้ำหนักใหม่ ซึ่งมีลายมือผู้ทดสอบในการสอนข่ายงานประสาท ให้ผลการทดสอบที่ดีกว่าผลของการทดสอบโดยใช้ไฟล์น้ำหนักเดิม ดังนั้นหากจะให้ระบบสามารถรู้จำได้มากขึ้นควรจะสอนข่ายงานประสาทด้วยลายมือ

จำนวนมาก และหลากหลายลายมือเขียน เพื่อให้ข่ายงานประสาทสามารถรู้จำลายมือเขียนได้หลากหลายรูปแบบมากขึ้น

4.4.1.2 ผลการทดสอบการรู้จำที่ได้จากระบบรู้จำลายมือออนไลน์สำหรับตัวอักษรภาษาไทยบนแท็บเล็ตพีซี ให้ผลการรู้จำต่ำกว่าผลการทดสอบการรู้จำที่ได้จากการทดสอบการรู้จำภายใต้โปรแกรมของคุณกันตนา กิติยานันท์ ซึ่งอาจเกิดจากปัญหาดังต่อไปนี้

1) ผลการทดสอบที่ได้จากโปรแกรมของคุณกันตนา กิติยานันท์ เป็นการทดสอบโดยเก็บข้อมูลลายมือเขียนไว้ในฐานข้อมูล ซึ่งในระหว่างที่เขียนตัวอักษรแต่ละตัวไม่มีการจำกัดเวลาในการยกปากกา และสามารถแก้ไขตัวอักษรได้ก่อนที่จะนำไปทดสอบ ส่วนการทดสอบภายใต้งานวิจัยนี้จะเป็นการทดสอบโดยให้ผู้ใช้เขียนตัวอักษรทีละตัวอักษร โดยเป็นการเขียนตัวอักษรออนไลน์ ที่มีการจำกัดเวลาในการยกปากกาในแต่ละสโตรค ห้ามนานเกิน 1 วินาที หากเกิน 1 วินาทีจะถือว่าได้เขียนจบหนึ่งตัวอักษร จะทำการประมวลผลทันที ซึ่งผู้ทดสอบไม่สามารถกลับมาแก้ไขตัวอักษรได้อีก

2) ผู้ทดสอบไม่คุ้นกับการเขียนบนแท็บเล็ตพีซี เนื่องจากการเขียนบนแท็บเล็ตพีซีจะลื่นกว่าการเขียนบนกระดาษ และต้องลงน้ำหนักในการเขียน แต่ผู้ทดสอบบางท่านจะกลัวหน้าจอของเครื่องแท็บเล็ตพีซีเป็นรอย จึงทำให้ผู้ทดสอบไม่กล้าลงน้ำหนักในการเขียน ทำให้ลายมือที่ออกมาจึงไม่ตรงกับลายมือปกติ

3) ปากกาของแท็บเล็ตพีซีไม่ถนัดมือ และมีปุ่มสำหรับใช้แทนการคลิกเม้าท์ ซึ่งปุ่มดังกล่าวอยู่ในตำแหน่งที่สามารถกดได้โดยง่าย ผู้เขียนจึงต้องคอยระวังมิให้กดโดนปุ่มดังกล่าว ทำให้การจับปากกาไม่ถนัดเท่าที่ควร

4) ตำแหน่งที่จรดปากกาซึ่งแสดงบนหน้าจอ ไม่ตรงกับตำแหน่งจริงที่ผู้ใช้จรดปากกา โดยจะเยื้องจากตำแหน่งจริงไปเล็กน้อย ส่งผลให้ผู้เขียนที่ไม่คุ้นเคยอาจรู้สึกว่าหลอกตา ทำให้ลายมือที่ออกมาไม่ตรงกับลายมือปกติ

5) จากการวิเคราะห์ข้อมูลในตารางรายละเอียดตัวอักษรที่รู้จำผิดในภาคผนวก ง. และข้อมูลที่ผ่านการประมวลเรียบร้อย พร้อมที่จะส่งให้ข่ายงานประสาทประมวลผลการรู้จำ พบว่าตัวอักษรที่รู้จำผิด ไม่ได้มีสาเหตุมาจากการประมวลผลข้อมูลในส่วนของการหาจำนวนวงรอบและทิศทางการวนของวงรอบ แต่พบว่าสิ่งที่แตกต่างระหว่างข้อมูลของตัวอักษรที่รู้จำได้ กับที่รู้จำผิด คือ จำนวนจุดเด่น และทิศทางของจุดเด่น จากการสังเกตพบว่าลักษณะการเขียนหัวของ

ตัวอักษร มีผลต่อการรู้จำ เช่น เมื่อผู้ทดสอบพยายามจะเขียนหัวของตัวอักษรให้กลม โดยเขียนหัวด้วยจำนวนวงรอบ หนึ่งรอบครึ่ง ทำให้เกิดจำนวนจุดและจำนวนจุดเด่นของตัวอักษรมากกว่าการเขียนปกติตามหลักคัดไทย ซึ่งส่งผลให้ข้อมูลที่ส่งให้ข่ายงานประสาทมีค่าแตกต่างจากข้อมูลที่เคยใช้สอนข่ายงานประสาท จึงทำให้ผลการรู้จำผิด (สอนข่ายงานประสาทด้วยการเขียนตามหลักคัดไทย ซึ่งวงรอบของหัวจะเขียนเพียงหนึ่งรอบเท่านั้น)

4.4.2 ผลการทดสอบในกรณีที่ใช้ลายมือเขียนครั้งละหลายตัวอักษร

ผลการทดสอบให้ผลการรู้จำที่ถูกต้อง 87.7% การรู้จำที่ผิด 12.3% จากการสังเกตพบปัญหาที่อาจจะทำให้ผลที่ได้ไม่ถูกต้องดังนี้

4.4.2.1 เงื่อนไขเกี่ยวกับระยะห่างของตัวอักษร ซึ่งได้กำหนดให้แต่ละตัวต้องห่างกัน 0.5 เซนติเมตร (สโตรคของตัวอักษรเดียวกันต้องอยู่ห่างกันน้อยกว่า 0.5 เซนติเมตร สโตรคของคนละตัวอักษรต้องอยู่ห่างกัน 0.5 เซนติเมตรขึ้นไป) ทำให้ผู้ทดสอบอาจรู้สึกว่าฝืนๆ และต้องคอยระวัง หรือในบางครั้งอาจจะเขียนติดกันในบางตัวอักษรหรือเขียนสโตรคของตัวอักษรเดียวกันห่างกันเกิน 0.5 เซนติเมตร ทำให้การแบ่งตัวอักษรผิดพลาด ส่งผลให้การรู้จำผิดพลาด

4.4.2.2 เงื่อนไขเกี่ยวกับระยะเวลาในการยกปากกา หากยกปากกาขึ้นนานกว่า 1 วินาทีตัวอักษรที่เขียนทั้งหมดจะถูกประมวลทันที ทำให้ผู้ทดสอบเขียนด้วยความเร่งรีบ เนื่องจากเกรงว่าจะยกปากกานานเกินไปในแต่ละสโตรค จึงเป็นเหตุให้ลายมือเขียนที่ได้เป็นลายมือที่ไม่เหมือนลายมือเขียนปกติ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

ในบทนี้ จะเป็นการกล่าวโดยสรุปสำหรับการพัฒนาระบบการรู้จำลายมือออนไลน์สำหรับตัวอักษรภาษาไทยบนแท็บเล็ตพีซี ปัญหาที่พบจากการวิจัย ข้อดีและข้อจำกัดของระบบ และข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ผู้วิจัยได้ทำการพัฒนาระบบการรู้จำลายมือออนไลน์สำหรับตัวอักษรภาษาไทยบนแท็บเล็ตพีซี โดยใช้ข่ายงานประสาทแบบแพร่กระจายย้อนกลับในการประมวลผลการรู้จำตัวอักษรลายมือเขียนและใช้เทคนิคในการจำแนกตัวอักษร ค่าพารามิเตอร์ที่ใช้สอนข่ายงานประสาท ไฟล์น้ำหนัก จากผลงานวิจัยของนายปกรณ์ บุพศิริ มาเป็นข้อมูลเบื้องต้น ซึ่งประกอบด้วย การประมวลผลข้อมูลตัวอักษรเบื้องต้น การหาลักษณะเด่นของตัวอักษร และการเข้ารหัส โดยมีเทคนิคที่สำคัญคือการหาวงรอบของตัวอักษร การใช้เขตย่อย

ผู้วิจัยได้เพิ่มการรู้จำตัวอักษรลายมือเขียนครั้งละหลายตัวอักษรต่อเนื่องกัน โดยเพิ่มเทคนิคในการคัดแยกตัวอักษร โดยใช้ระยะห่างของสโตรค และเทคนิคในการตรวจสอบตัวอักษรก่อนหน้า ซึ่งเทคนิคในการคัดแยกตัวอักษร โดยใช้ระยะห่างของสโตรค เป็นการจำกัดระยะห่างระหว่างสโตรคของตัวอักษรเดียวกันไม่ให้เกิน 0.5 เซนติเมตร หากระยะห่างเกินกว่า 0.5 เซนติเมตรจะถือว่าสโตรคทั้งสองสโตรคเป็นคนละตัวอักษร เทคนิคในการตรวจสอบตัวอักษรก่อนหน้า เป็นการใช้ลักษณะพิเศษของตัวอักษรบางตัวอักษรและข้อกำหนดในการเขียนตัวอักษรภาษาไทยเข้ามาช่วยในการพิจารณาและปรับปรุงตัวอักษรที่รู้จำได้ให้ได้ผลการรู้จำถูกต้องแม่นยำมากขึ้น เช่น การที่ตัวอักษรบางตัวมีลักษณะของสโตรคที่หนึ่งเหมือนกับสโตรคที่สอง และแต่ละสโตรคมีลักษณะตรงกับตัวอักษรตัวอื่น ซึ่งตัวอักษรที่ตรงกันนี้มีข้อกำหนดว่าไม่สามารถเขียนตัวอักษรเดียวกันติดต่อกันได้ เป็นต้น

ผู้วิจัยได้ออกแบบระบบเป็นไลบรารี เพื่อให้เหมาะแก่การนำไปใช้พัฒนาร่วมกับระบบอื่นๆต่อไป โดยมีฟังก์ชันให้ผู้ที่นำไปใช้สามารถเรียกใช้ได้สะดวก และได้ออกแบบระบบให้สามารถรู้จำได้ทั้งตัวอักษรภาษาไทยและภาษาอังกฤษ พร้อมทั้งรองรับลายมือเขียนจากผู้ใช้หลายคน สามารถส่งตัวอักษรที่รู้จำได้ไปยังโปรแกรมไมโครซอฟท์เวิร์ด และ ไมโครซอฟท์เอ็กซเซล รองรับการ

ปรับเปลี่ยนไฟล์น้ำหนักและค่าพารามิเตอร์ของข่ายงานประสาทได้ตามรูปแบบของโปรแกรมข่ายงานประสาทของมหาวิทยาลัยสตุทการ์ท เพื่อเปิดให้สามารถปรับค่าการรู้จำของระบบ และรองรับการเขียนตัวอักษรครั้งละหลายตัวอักษรต่อเนื่องกัน

ผู้วิจัยได้ทำการทดสอบระบบโดยแบ่งการทดสอบออกเป็น 2 รูปแบบ ดังนี้

1) การทดสอบการรู้จำตัวอักษรภาษาไทยที่ใช้ลายมือเขียนครั้งละตัวอักษร แบ่งเป็น

- การทดสอบโดยใช้ไฟล์น้ำหนักเดิมจากงานวิจัยของนายปกรณ์ บุพศิริ ด้วยลายมือของผู้ทดสอบ 3 ท่าน จำนวนท่านละ 10 ชุด ซึ่งให้ผลการรู้จำที่ถูกต้อง 73.38% การรู้จำผิด 26.62%

- การทดสอบการรู้จำตัวอักษรภาษาไทยที่ใช้ลายมือเขียนครั้งละตัวอักษร โดยใช้ไฟล์น้ำหนักใหม่ ด้วยลายมือของผู้ทดสอบ 3 ท่าน จำนวนท่านละ 10 ชุด ซึ่งให้ผลการรู้จำที่ถูกต้อง 89.4% การรู้จำผิด 10.6%

2) การทดสอบการรู้จำตัวอักษรภาษาไทยที่ใช้ลายมือเขียนครั้งละหลายตัวอักษร โดยใช้ไฟล์น้ำหนักใหม่ ด้วยลายมือของผู้ทดสอบ 3 ท่าน จำนวนท่านละ 10 ชุด ซึ่งให้ผลการรู้จำที่ถูกต้อง 87.7% การรู้จำผิด 12.3%

สรุปผลจากการทดสอบพบว่าระบบที่ได้สามารถรู้จำได้ และรองรับการปรับเปลี่ยนไฟล์น้ำหนักและค่าพารามิเตอร์ของข่ายงานประสาทได้ตามรูปแบบของโปรแกรมข่ายงานประสาทของมหาวิทยาลัยสตุทการ์ท และรองรับการเขียนตัวอักษรครั้งละหลายตัวอักษร

5.2 ปัญหาที่พบจากการวิจัย

ผู้วิจัยได้พบปัญหาในระหว่างทำการวิจัยบางประการที่น่าจะเป็นประโยชน์ และสามารถนำไปแก้ไขปัญหาในการพัฒนาระบบอื่นๆต่อไป ดังนี้

1) ปัญหาในการเรียกใช้ไลบารี่ของไมโครซอฟท์เอ็กซเซล เวอร์ชัน 2003 ซึ่งเกิดจากการกำหนดรูปแบบมาตรฐานสำหรับตัวเลข วัน-เวลา จำนวนเงิน ในรูปแบบของประเทศไทย ทำให้คำสั่งที่เรียกใช้งานไมโครซอฟท์เอ็กซเซล เวอร์ชั่น 2003 ไม่สามารถใช้งานได้ โดยมีข้อความแจ้งดังนี้ “คำสั่งอยู่ในรูปแบบเก่าหรือไลบรารีไม่ถูกต้อง” ซึ่งเมื่อทดลองแก้ไขค่าดังกล่าวให้อยู่ในรูปแบบของประเทศอังกฤษ ไม่พบปัญหาใดๆที่เคยพบและสามารถเรียกใช้ไลบารี่ของไมโครซอฟท์เอ็กซเซลด้วยคำสั่งเดิมได้ แต่การแก้ไขค่านี้อาจมีพบกระทบกับโปรแกรมอื่นๆของผู้ใช้งาน เพื่อ

แก้ปัญหานี้ผู้วิจัยจึงใช้วิธีการสลับค่าดังกล่าวให้เป็นรูปแบบของประเทศอังกฤษ ก่อนที่จะเรียกใช้งานไลบารี่ของไมโครซอฟท์เอ็กซเซล และทำการสลับค่ากลับไปยังค่าเดิมทุกครั้งหลังจากใช้งานเสร็จ

2) ปัญหาของตัวอักษรสระอำ ซึ่งประกอบด้วย 2 สโตรค ที่มีขอบเขตระหว่าง 2 สโตรคห่างกันมาก และบางครั้งขอบเขตของสโตรคแรกอาจอยู่ภายในขอบเขตของสโตรคของตัวอักษรก่อนหน้า ทำให้การใช้เทคนิคการคัดแยกตัวอักษร โดยใช้ระยะห่างของสโตรค ไม่สามารถใช้ได้กับตัวอักษรสระอำ

5.3 ประโยชน์ที่ได้รับจากงานวิจัย

1) ได้ระบบการรู้จำลายมือออนไลน์สำหรับตัวอักษรภาษาไทยบนแท็บเล็ตพีซี ซึ่งมีความสามารถดังนี้

- รู้จำได้ทั้งตัวอักษรภาษาไทยและภาษาอังกฤษ
- รองรับลายมือเขียนจากผู้ใช้หลายคน
- สามารถส่งตัวอักษรที่รู้จำได้ไปยังโปรแกรมไมโครซอฟท์เวิร์ด และ ไมโครซอฟท์เอ็กซเซล
- รองรับการปรับเปลี่ยนไฟล์น้ำหนักและค่าพารามิเตอร์ของข่ายงานประสาทได้ตามรูปแบบของโปรแกรมข่ายงานประสาทของมหาวิทยาลัยสตุทการ์ท เพื่อเปิดให้สามารถปรับค่าการรู้จำของระบบ
- รองรับการเขียนตัวอักษรครั้งละหลายตัวอักษรต่อเนื่องกัน

2) เป็นแนวทางสำหรับการพัฒนาระบบงานด้านอื่น ๆ

5.4 ข้อจำกัดของระบบ

ระบบที่ได้นำเสนอนี้มีข้อจำกัด ซึ่งมีรายละเอียดดังต่อไปนี้

1) สามารถรู้จำตัวอักษรลายมือเขียนภาษาไทยจาก 67 ตัวอักษร ซึ่งประกอบด้วยพยัญชนะ 42 ตัว สระ 17 ตัว วรรณยุกต์ 4 ตัว และอักขระพิเศษ 4 ตัว ที่เลือกไว้ คือ ก ข ค ฆ ง จ ฉ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย ร ล ว ศ ษ ส ห ฬ อ ฮ ะ า ิ ี ึ ื ุ ู เ แ โ ไ ใ ำ ั ฤ ฦ ่ ้ ๊ ๋ ์ ฯ ๆ

2) จำนวนสโตรคสูงสุดที่รับได้ไม่เกิน 2 สโตรค สำหรับตัวอักษรแต่ละตัว โดยมีระยะเวลาหน่วงระหว่างสโตรคที่ 1 กับสโตรคที่ 2 ไม่เกิน 1 วินาที ถ้าผู้ใช้เครื่องมือป้อนข้อมูลในสโตรคถัดไปช้ากว่าเวลาที่กำหนดระบบจะถือว่าข้อมูลลายมือเขียนนี้ได้สิ้นสุดการป้อนข้อมูลแล้ว

3) สามารถส่งตัวอักษรที่รู้จำได้ไปยังโปรแกรมไมโครซอฟท์เวิร์ด และไมโครซอฟท์เอ็กซเซล เท่านั้น สำหรับแอพพลิเคชันอื่นๆผู้ใช้งานต้องใช้วิธีคัดลอกไปวางด้วยตนเอง

4) ตัวอักษรลายมือเขียนภาษาไทย ในโหมดของการเขียนครั้งละหลายตัวอักษร จะต้องเขียนตัวอักษรแต่ละตัวห่างกันประมาณ 0.5 เซนติเมตร (สโตรคทั้งหมดของตัวอักษรเดียวกันจะต้องอยู่ห่างกันไม่เกิน 0.5 เซนติเมตร)

5.4 ข้อเสนอแนะ

ผู้วิจัยพบว่ายังมีข้อเสนอแนะบางประการที่น่าจะเป็นประโยชน์ และสามารถนำไปใช้ในการพัฒนาต่อไป ดังนี้

1) ควรเพิ่มความยืดหยุ่นให้กับผู้ใช้งาน โดยเพิ่มส่วนที่ให้ผู้ใช้งานสามารถปรับเปลี่ยนระยะห่างในการเขียนตัวอักษรแต่ละตัวได้ เพื่อให้ผู้ใช้งานสามารถปรับเปลี่ยนระยะห่างได้ตามความถนัดของแต่ละบุคคล

2) ควรปรับปรุงค่าระยะห่างของตัวอักษร จากการกำหนดเป็นค่าคงที่ เปลี่ยนเป็นหาค่าระยะห่างจากอัตราส่วนหรือเปอร์เซนต์ของความกว้างของตัวอักษรเฉลี่ย

3) ควรปรับปรุงประสิทธิภาพในการรู้จำตัวอักษรครั้งละหลายตัวอักษรให้รู้จำได้มากขึ้น เนื่องจากในการใช้งานจริง ผู้ใช้งานต้องการที่จะเขียนตัวอักษรต่อเนื่องกันไปครั้งละหลายตัวอักษร โดยเพิ่มเทคนิคอื่นๆ เช่น เทคนิคอื่นๆที่จะมาช่วยคัดแยกสโตรคออกเป็นตัวอักษรแต่ละตัวอักษร เทคนิคการตรวจสอบคำที่รู้จำได้กับพจนานุกรม เทคนิคการรู้จำตัวอักษรทีละสโตรค เป็นต้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

[1] อภิชาต สัจจพงษ์. การรู้จำลายมือเขียนภาษาไทยแบบออนไลน์โดยใช้นิวรอลเน็ตเวิร์ก. วิทยานิพนธ์ ปริญญาวิทยาศาสตร์มหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย, ปี 2541.

[2] โชติ ศิริวงศ์วิเชียร และ ปกิตต์ นิธิวิบูลย์. การพัฒนาโปรแกรมสำหรับการเข้ารหัสตัวอักษรลายมือเขียนภาษาไทยเพื่อระบบการรู้จำโดยใช้นิวรอลเน็ตเวิร์ก. โครงงานทางวิศวกรรมปริญญาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย, ปี 2543.

[3] ปกรณ์ บุพศิริ. การปรับปรุงระบบการรู้จำลายมือเขียนแบบออนไลน์สำหรับตัวอักษรภาษาไทย โดยใช้ข่ายงานประสาทแบบแพร่กระจายย้อนกลับ. วิทยานิพนธ์ปริญญาวิทยาศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, ปี 2545.

[4] สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น). แบบฝึกหัดคัดไทย. พิมพ์ครั้งที่ 3. กรุงเทพมหานคร :, ปี 2541.

[5] Simon Haykin. NEURAL NETWORKS: A Comprehensive Foundation. Ontario. Canada. McMaster University, (1999)

[6] X. Li and D. Yeung. On-line Handwritten Alphanumeric Character Recognition using Dominant Points in Strokes. Pattern Recognition vol. 30, no.1 (1997) : 31-44.

[7] Zahedi A. Nair and C.G. Leedham. Preprocessing of Line Codes for Online Recognition Purposes. Electronics Letters vol. 27, (1991) : 1-2.

[8] I. Pitas. Digital Image Processing Algorithms. U.K. : Prentice Hall, 1993.

[9] X. Li , M. Parizeau and R. Plamondon. Segmentation and Reconstruction of On-line Handwritten Scripts. Pattern Recognition vol. 31, no.6 (1998) : 675-684.

[10] Stuttgart Neural Network Simulator (SNNS) version 4.1. Available from: <http://www-ra.informatik.uni-tuebingen.de/SNNS/>

[11] ราชบัณฑิตยสถาน. มาตรฐานโครงสร้างตัวอักษรไทย. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : อรุณการพิมพ์, ปี 2540.

[12] Microsoft Windows XP Tablet PC Edition 2005 Recognizer Pack. Available from:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=080184dd-5e92-4464-b907-10762e9f918b&DisplayLang=en> (14 Sep 2005)

[13] กันตา กิติยานันท์. การพัฒนาเครื่องมือซอฟต์แวร์สำหรับระบบการรู้จำลายมือเขียนแบบออนไลน์โดยใช้ข่ายงานประสาทแบบแพร่กระจายย้อนกลับ. วิทยานิพนธ์ ปริญญาวิทยาศาสตร์มหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย, ปี 2544.

[14] Sutat Sae-Tang, Ithipan Methasate. Thai Online Handwritten Character Recognition Using Windowing Backpropagation Neural Networks. International Conference on Modelling, Identification, and Control (MIC2002), Innsbruck, Austria, pp 337-340, (18-21 Feb 2002).

[15] Philip Su, Rob Jarrett. Building Tablet PC Applications. United States of America : Microsoft Press, (1976)

[16] Microsoft Windows XP Tablet PC Edition Development Kit 1.7. Available from:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=b46d4b83-a821-40bc-aa85-c9ee3d6e9699&DisplayLang=en> (14 Sep 2005)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

ฟังก์ชันที่ใช้ในการรู้จำลายมือเขียนครั้งละหลายตัวอักษร

ฟังก์ชันที่ใช้ในการรู้จำลายมือเขียนครั้งละหลายตัวอักษร คือฟังก์ชันที่ช่วยให้ระบบสามารถรู้จำลายมือเขียนที่เขียนครั้งละหลายตัวอักษรต่อเนื่องกัน ซึ่งมีรายละเอียดดังต่อไปนี้

1) การคัดแยกสโตรค เป็นสิ่งที่สำคัญในการรู้จำลายมือเขียนครั้งละหลายตัวอักษร ผู้วิจัยจึงได้เสนอฟังก์ชันการคัดแยกสโตรคที่ใช้ในการหาสโตรคที่ประกอบเป็นตัวอักษรแต่ละตัว ดังนี้

```
Private Function TwoStrokeDistant(ByVal Stroke1 As WrittingPoints, _
                        ByVal Stroke2 As WrittingPoints) As Boolean

      Dim S1_XMin, S1_XMax, S1_YMin, S1_YMax As Double
      Dim S2_XMin, S2_XMax, S2_YMin, S2_YMax As Double
      Dim bResult As Boolean
      Dim MaxDistant As Double

      MaxDistant = 500    '1 unit = 0.01 mm

      FindMaxMinPoint(Stroke1, S1_XMax, S1_XMin, S1_YMax, S1_YMin)
      FindMaxMinPoint(Stroke2, S2_XMax, S2_XMin, S2_YMax, S2_YMin)

      bResult = False

      If S2_XMin > S1_XMax Then 'Case S2 is right of S1
         If S2_YMin > S1_YMax Then  'Case S2 on top of S1
            If (S2_YMin - S1_YMax) < MaxDistant Then bResult = True
         ElseIf S1_YMin > S2_YMax Then 'Case S2 in below of S1
            If (S1_YMin - S2_YMax) < MaxDistant Then bResult = True
         Else
            If (S2_XMin - S1_XMax) < MaxDistant Then bResult = True
         End If
      ElseIf S2_XMax > S1_XMin Then  'Case S2 is not left of S1 and
is not right of S1
         If S2_YMin > S1_YMax Then  'Case S2 on top of S1
            If (S2_YMin - S1_YMax) < MaxDistant Then bResult = True
         ElseIf S1_YMin > S2_YMax Then 'Case S2 in below of S1
            If (S1_YMin - S2_YMax) < MaxDistant Then bResult = True
         Else
            bResult = True
         End If
      End If

      Return bResult
End Function


Private Sub FindMaxMinPoint(ByRef PointList As WrittingPoints, _
            ByRef XMax As Double, ByRef XMin As Double, _
            ByRef YMax As Double, ByRef YMin As Double)

        Dim p As New WrittingPoint
```

```
        Dim bFirst As Boolean

        bFirst = True
        For Each p In PointList
            If bFirst Then
               XMax = p.X : XMin = p.X
               YMax = p.Y : YMin = p.Y
               bFirst = False
            Else
               If p.X > XMax Then XMax = p.X
               If p.Y > YMax Then YMax = p.Y
               If p.X < XMin Then XMin = p.X
               If p.Y < YMin Then YMin = p.Y
            End If
        Next
End Sub
```

2) การตรวจสอบตัวอักษรก่อนหน้า เป็นการปรับปรุงตัวอักษรที่รู้จำได้ให้ถูกต้องมากขึ้น ในรู้จำลายมือเขียนครั้งละหลายตัวอักษร ผู้วิจัยจึงได้เสนอฟังก์ชันการตรวจสอบตัวอักษรก่อนหน้า ดังนี้

```
Private Function ChkPrevChar(ByVal strOriginal As String) As String
      Dim strResult As String
      Dim CurChar, PrevChar As Char
      Dim i As Integer
      i = 1
      Do While i <= strOriginal.Length
         CurChar = Mid(strOriginal, i, 1)
         If PrevChar <> "" Then
            If Asc(CurChar) = 209 And Asc(PrevChar) = 209 Then
               strResult = Mid(strResult, 1, strResult.Length - 1)
               CurChar = Chr(208)
            ElseIf Asc(CurChar) = 224 And Asc(PrevChar) = 224 Then
               strResult = Mid(strResult, 1, strResult.Length - 1)
               CurChar = Chr(225)
            End If
         End If
         strResult = strResult & CurChar
         PrevChar = CurChar
         i = i + 1
      Loop
      Return strResult
End Function
```

ภาคผนวก ข

รูปแบบของไฟล์โค้ดและไฟล์น้ำหนัก

ไฟล์โค้ด เป็นไฟล์ที่เก็บข้อมูลรหัสแอสกีของตัวอักษรไทย และตัวเลขรหัสผลลัพธ์ที่กำหนดให้ตัวอักษรแต่ละตัวอักษร ซึ่งรหัสแอสกีของตัวอักษรไทยและเลขรหัสผลลัพธ์ทั้งหมดเป็นข้อมูลชุดเดียวกับที่ใช้ในการสอนข่ายงานประสาท โดยกำหนดให้โค้ดไฟล์มีรูปแบบดังนี้

1) จัดเก็บในรูปแบบของไฟล์เท็กซ์ (text file)

2) ข้อมูลของตัวอักษรแต่ละตัวอักษรจัดเก็บแยกออกเป็นแต่ละบรรทัด ซึ่งในแต่ละบรรทัดประกอบด้วยรหัสผลลัพธ์ที่กำหนดให้แต่ละตัวอักษรและรหัสแอสกีของตัวอักษรที่คู่กับผลลัพธ์

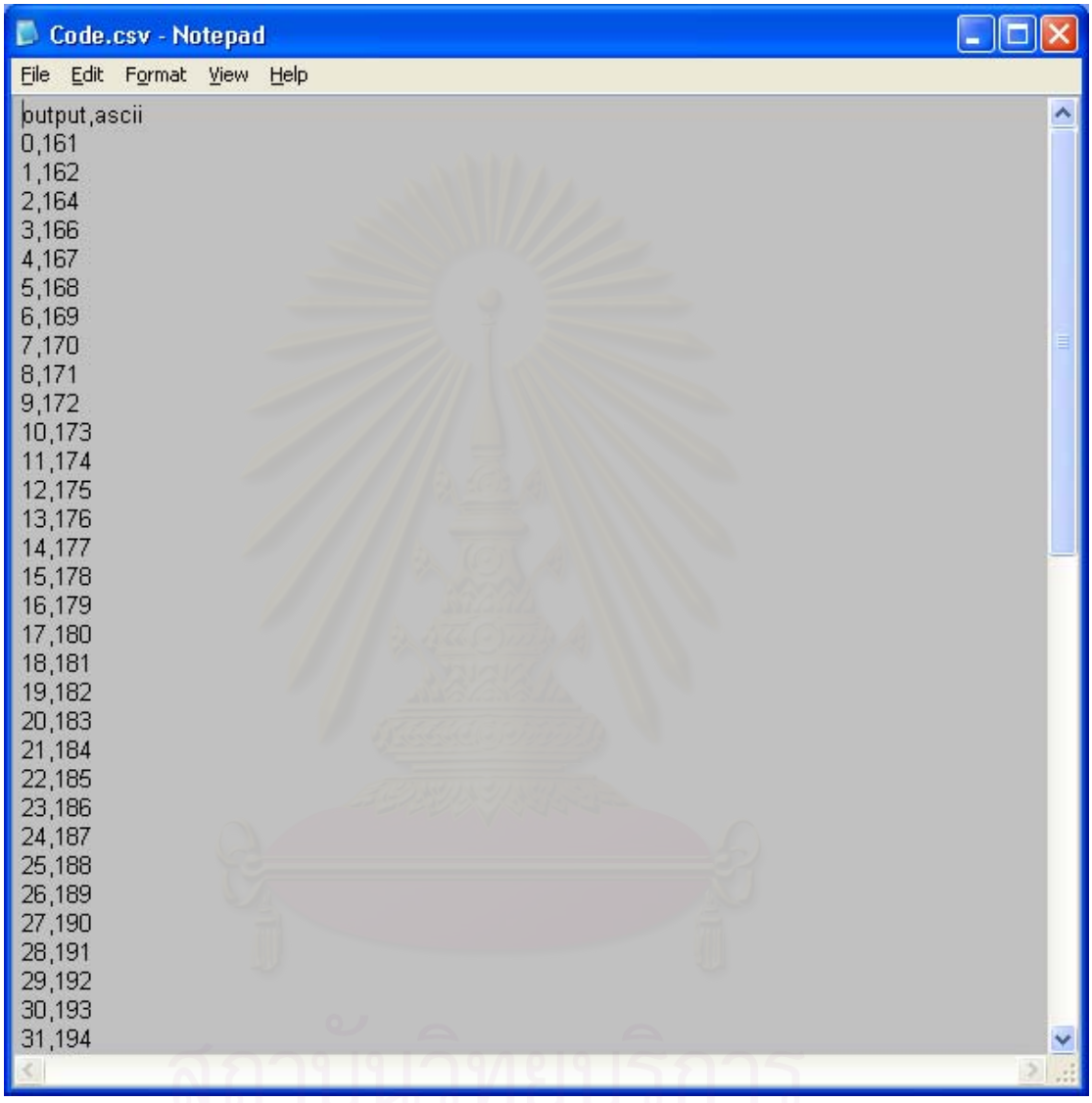
3) ระหว่างตัวเลขรหัสผลลัพธ์ที่กำหนดให้แต่ละตัวอักษร กับรหัสแอสกีของตัวอักษร คั่นด้วยเครื่องหมายคอมมา (“,”) โดยให้ตัวเลขรหัสผลลัพธ์ขึ้นก่อนแล้วตามด้วยรหัสแอสกีของตัวอักษร

4) บรรทัดแรกของข้อมูลต้องขึ้นด้วย output,ascii

5) บรรทัดที่สองเป็นต้นไปจึงเป็นข้อมูลของแต่ละตัวอักษร

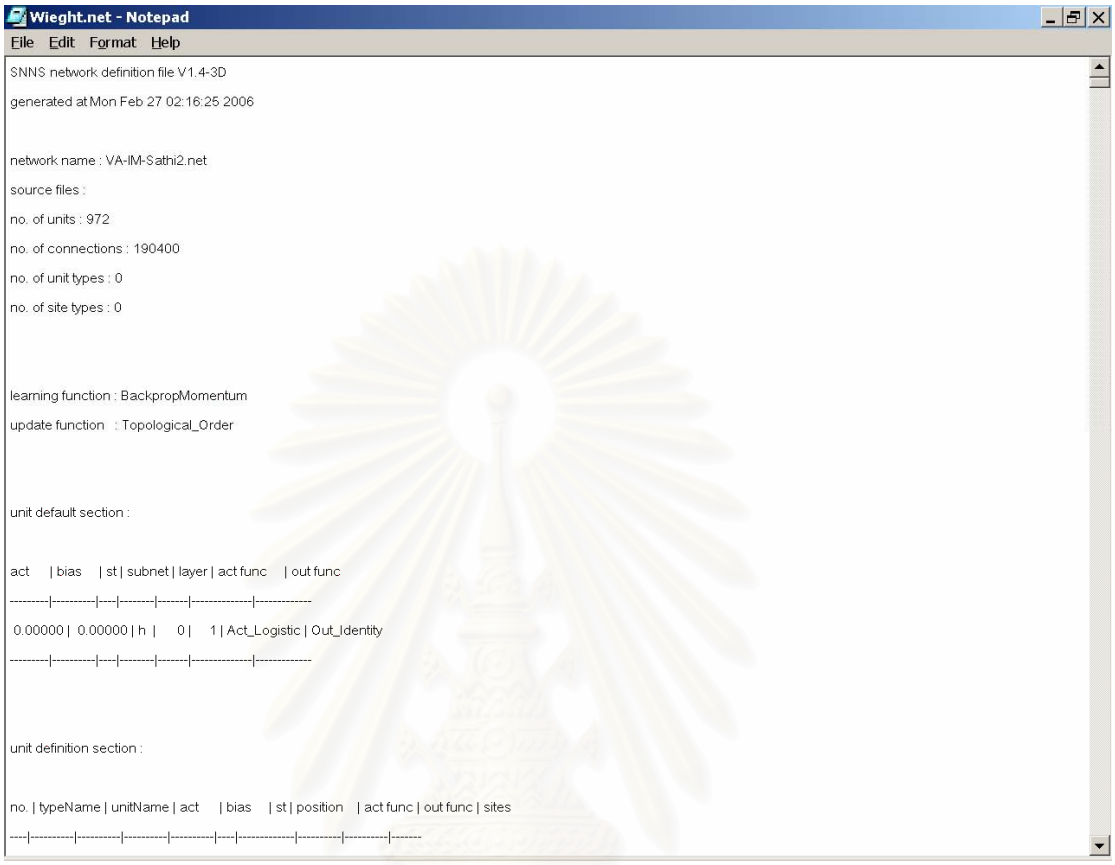
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอย่างโค้ดไฟล์



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอย่างไฟล์น้ำหนัก

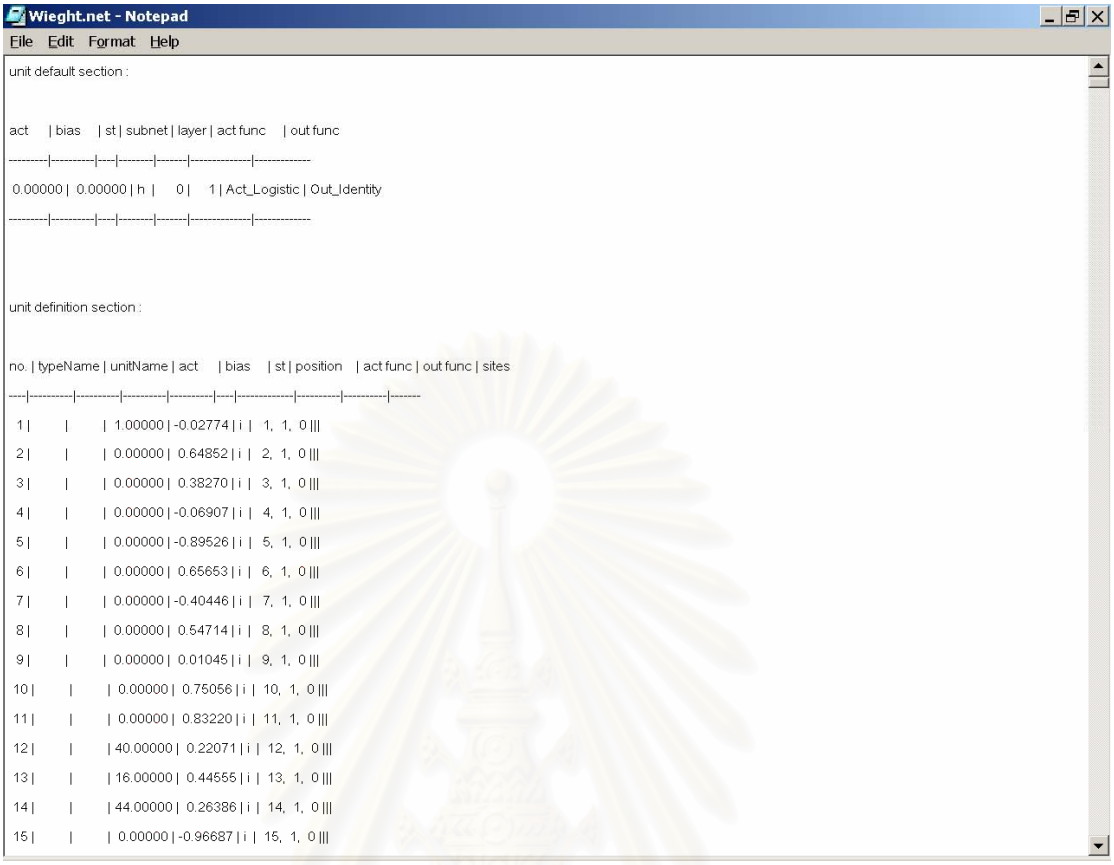


Wieght.net - Notepad

File Edit Format Help

```
SNNS network definition file V1.4-3D
generated at Mon Feb 27 02:16:25 2006

network name : VA-IM-Sathi2.net
source files :
no. of units : 972
no. of connections : 190400
no. of unit types : 0
no. of site types : 0


learning function : BackpropMomentum
update function   : Topological_Order


unit default section :

act     | bias    | st | subnet | layer | act func     | out func
---------|---------|----|--------|-------|--------------|-------------
 0.00000 |  0.00000 | h |      0 |     1 | Act_Logistic | Out_Identity
---------|---------|----|--------|-------|--------------|-------------


unit definition section :

no. | typeName | unitName | act     | bias    | st | position    | act func | out func | sites
----|----------|----------|---------|---------|----|-------------|----------|----------|-------
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



Wieght.net - Notepad

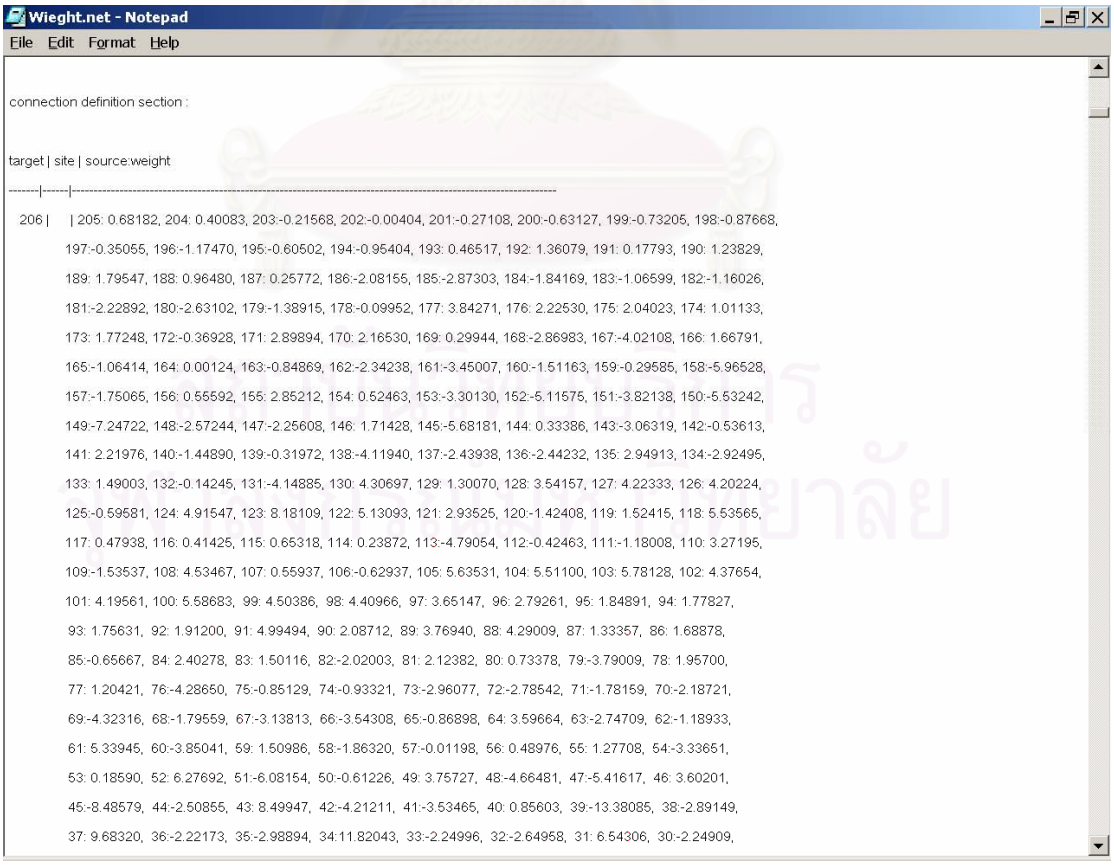
File Edit Format Help

unit default section :

act	bias	st	subnet	layer	act func	out func
0.00000	0.00000	h	0	1	Act_Logistic	Out_Identity

unit definition section :

no.	typeName	unitName	act	bias	st	position	act func	out func	sites
1			1.00000	-0.02774	i	1, 1, 0		
2			0.00000	0.64852	i	2, 1, 0		
3			0.00000	0.38270	i	3, 1, 0		
4			0.00000	-0.06907	i	4, 1, 0		
5			0.00000	-0.89526	i	5, 1, 0		
6			0.00000	0.65653	i	6, 1, 0		
7			0.00000	-0.40446	i	7, 1, 0		
8			0.00000	0.54714	i	8, 1, 0		
9			0.00000	0.01045	i	9, 1, 0		
10			0.00000	0.75056	i	10, 1, 0		
11			0.00000	0.83220	i	11, 1, 0		
12			40.00000	0.22071	i	12, 1, 0		
13			16.00000	0.44555	i	13, 1, 0		
14			44.00000	0.26386	i	14, 1, 0		
15			0.00000	-0.96687	i	15, 1, 0		



Wieght.net - Notepad

File Edit Format Help

connection definition section :

target	site	source:weight

206 | | 205: 0.68182, 204: 0.40083, 203:-0.21568, 202:-0.00404, 201:-0.27108, 200:-0.63127, 199:-0.73205, 198:-0.87668,
197:-0.35055, 196:-1.17470, 195:-0.60502, 194:-0.95404, 193: 0.46517, 192: 1.36079, 191: 0.17793, 190: 1.23829,
189: 1.79547, 188: 0.96480, 187: 0.25772, 186:-2.08155, 185:-2.87303, 184:-1.84169, 183:-1.06599, 182:-1.16026,
181:-2.22892, 180:-2.63102, 179:-1.38915, 178:-0.09952, 177: 3.84271, 176: 2.22530, 175: 2.04023, 174: 1.01133,
173: 1.77248, 172:-0.36928, 171: 2.89894, 170: 2.16530, 169: 0.29944, 168:-2.86983, 167:-4.02108, 166: 1.66791,
165:-1.06414, 164: 0.00124, 163:-0.84869, 162:-2.34238, 161:-3.45007, 160:-1.51163, 159:-0.29585, 158:-5.96528,
157:-1.75065, 156: 0.55592, 155: 2.85212, 154: 0.52463, 153:-3.30130, 152:-5.11575, 151:-3.82138, 150:-5.53242,
149:-7.24722, 148:-2.57244, 147:-2.25608, 146: 1.71428, 145:-5.68181, 144: 0.33386, 143:-3.06319, 142:-0.53613,
141: 2.21976, 140:-1.44890, 139:-0.31972, 138:-4.11940, 137:-2.43938, 136:-2.44232, 135: 2.94913, 134:-2.92495,
133: 1.49003, 132:-0.14245, 131:-4.14885, 130: 4.30697, 129: 1.30070, 128: 3.54157, 127: 4.22333, 126: 4.20224,
125:-0.59581, 124: 4.91547, 123: 8.18109, 122: 5.13093, 121: 2.93525, 120:-1.42408, 119: 1.52415, 118: 5.53565,
117: 0.47938, 116: 0.41425, 115: 0.65318, 114: 0.23872, 113:-4.79054, 112:-0.42463, 111:-1.18008, 110: 3.27195,
109:-1.53537, 108: 4.53467, 107: 0.55937, 106:-0.62937, 105: 5.63531, 104: 5.51100, 103: 5.78128, 102: 4.37654,
101: 4.19561, 100: 5.58683, 99: 4.50386, 98: 4.40966, 97: 3.65147, 96: 2.79261, 95: 1.84891, 94: 1.77827,
93: 1.75631, 92: 1.91200, 91: 4.99494, 90: 2.08712, 89: 3.76940, 88: 4.29009, 87: 1.33357, 86: 1.68878,
85:-0.65667, 84: 2.40278, 83: 1.50116, 82:-2.02003, 81: 2.12382, 80: 0.73378, 79:-3.79009, 78: 1.95700,
77: 1.20421, 76:-4.28650, 75:-0.85129, 74:-0.93321, 73:-2.96077, 72:-2.78542, 71:-1.78159, 70:-2.18721,
69:-4.32316, 68:-1.79559, 67:-3.13813, 66:-3.54308, 65:-0.86898, 64: 3.59664, 63:-2.74709, 62:-1.18933,
61: 5.33945, 60:-3.85041, 59: 1.50986, 58:-1.86320, 57:-0.01198, 56: 0.48976, 55: 1.27708, 54:-3.33651,
53: 0.18590, 52: 6.27692, 51:-6.08154, 50:-0.61226, 49: 3.75727, 48:-4.66481, 47:-5.41617, 46: 3.60201,
45:-8.48579, 44:-2.50855, 43: 8.49947, 42:-4.21211, 41:-3.53465, 40: 0.85603, 39:-13.38085, 38:-2.89149,
37: 9.68320, 36:-2.22173, 35:-2.98894, 34:11.82043, 33:-2.24996, 32:-2.64958, 31: 6.54306, 30:-2.24909,

ภาคผนวก ค

ตัวอย่างการเรียกใช้ไลบรารีสำหรับรู้จำลายมือภาษาไทย

```
Imports CU.ThaiRecognize
Imports Microsoft.Ink
Dim ThaiRecognize As New TRecognize
Const CONFIG_PATH = "\Conf\"  'ระบุไดเรคตอรี่ที่ใช้เก็บไฟล์น้ำหนักและไฟล์โค้ด
Dim myInkCollector As InkCollector
Dim g As Graphics
Dim myRecognizers As Recognizers

Private Sub Form_Load(ByVal sender As Object, _
            ByVal e As System.EventArgs) Handles MyBase.Load
        ThaiRecognize = New TRecognize
        g = gbInkArea.CreateGraphics()
        myRecognizers = New Recognizers
        myInkCollector = New InkCollector(gbInkArea.Handle)
        myInkCollector.Enabled = True
End Sub

Private Sub gbInkArea_MouseDown(ByVal sender As Object, _
            ByVal e As System.Windows.Forms.MouseEventArgs) _
            Handles gbInkArea.MouseDown
       TimerExitWrite.Enabled = False
End Sub

Private Sub gbInkArea_MouseUp(ByVal sender As Object, _
            ByVal e As System.Windows.Forms.MouseEventArgs) _
            Handles gbInkArea.MouseUp
         TimerExitWrite.Enabled = True
End Sub

Private Sub TimerExitWrite_Elapsed(ByVal sender As Object, _
            ByVal e As System.Timers.ElapsedEventArgs) _
            Handles TimerExitWrite.Elapsed
        TimerExitWrite.Enabled = False
        RecognizeWritting()
End Sub

Private Sub RecognizeWritting()
        Dim strRecognize As String
        Dim RecognizeStrokes As New TStrokes
        ' สั่งให้ไลบรารีสำหรับรู้จำลายมือภาษาไทยอ่านไฟล์น้ำหนักและไฟล์โค้ด
       ThaiRecognize.Init(Environment.CurrentDirectory & _
CONFIG_PATH)

        ' กำหนดระยะห่างของแต่ละตัวอักษรหน่วยเป็น 0.01 มิลลิเมตร
        ThaiRecognize.ChrDistant = 500

        ' สั่งให้ไลบรารีสำหรับรู้จำลายมือภาษาไทยเปลี่ยนอิงค์สโตรคคอลเลชัน (ink stroke collection) ให้
อยู่ในรูปแบบสโตรคคอลเลชันของไลบรารีสำหรับรู้จำลายมือภาษาไทย
      ThaiRecognize.ConvertInkStrokeToTStrokes( _
```

```
myInkCollector.Ink.Strokes, RecognizeStrokes)

        'สั่งให้ไลบรารีสำหรับรู้จำลายมือภาษาไทยประมวลผลลายมือเขียนตัวอักษรภาษาไทยและส่งผลลัพธ์
ตัวอักษรที่รู้จำได้กลับคืนมา
          strRecognize = ThaiRecognize.Recognize(RecognizeStrokes, _
gbInkArea, False)
End Sub
```


สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ง

ตารางรายละเอียดตัวอักษรที่ระบบรู้จำผิดและผลการรู้จำผิด

ตัวอักษรที่ระบบรู้จำผิด	ผลการรู้จำผิด																																																		
	ก	ข	ค	จ	ฉ	ช	ซ	ฌ	ฎ	ฏ	ฑ	ณ	ด	ต	ถ	ท	ธ	น	ฝ	พ	ฟ	ม	ย	ร	ล	ว	ษ	ส	ฬ	อ	อะ	อี	อึ	อื	อุ	อู	เ	แ	โ	ใ	ไ	อำ	อั	ฤ	ฦ	อ้	อ๊	อ๋	อ์	ฯ	ๆ	รวม
ข											2									2																2																6
ค													7			1																																				8
ฆ								1																																												1
ง				1																																																1
ฉ								1				1																																								2
ช							6																																													6
ซ		1																																																		1
ฌ												5																										3														8
ฎ										12												2	1								1														1							17
ฏ									5																																											5
ฑ																1											1																									2
ณ																																						1														1

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางรายละเอียดตัวอักษรที่ระบบรู้จำผิดและผลการรู้จำผิด(ต่อ)

ตัวอักษรที่ระบบรู้จำผิด	ผลการรู้จำผิด																																																			รวม
	ก	ข	ค	จ	ฉ	ช	ซ	ฌ	ฎ	ฏ	ฑ	ณ	ด	ต	ถ	ท	ธ	น	ฝ	พ	ฟ	ม	ย	ร	ล	ว	ษ	ส	ฬ	ฮ	อะ	อี	อึ	อื	อุ	อู	เ	แ	โ	ใ	ไ	อำ	อั	ฤ	ฦ	อ้	อ๊	อ๋	อ์	ฯ	ๆ	
ต													1																																							1
ถ																														1																	3					4
น					1																																															1
ป																																					1															1
ผ																				2																																2
ฝ																	1				9																															10
พ																		5																																		5
ฟ																						1																														1
ภ															2							1			1								2												1		2					9
ย																																	1																			1
ร																										1															1											2
ล												1																																								1
ว																																	1								2											3

ตารางรายละเอียดตัวอักษรที่ระบบรู้จำผิดและผลการรู้จำผิด(ต่อ)

ตัวอักษรที่ระบบรู้จำผิด	ผลการรู้จำผิด																																																			
	ก	ข	ค	จ	ฉ	ช	ซ	ฌ	ฎ	ฏ	ฑ	ณ	ด	ต	ถ	ท	ธ	น	ฝ	พ	ฟ	ม	ย	ร	ล	ว	ษ	ส	ฬ	ฮ	อะ	อี	อึ	อื	อุ	อู	เ	แ	โ	ใ	ไ	อำ	อั	ฤ	ฦ	อ้	อ๊	อ๋	อ์	ฯ	ๆ	รวม
ศ																												2	2	5																						9
ส																													1																							1
ห											3			1		1																																			1	6
ฬ								1																																												1
อ														2					1																								1					2				6
ฮ								1																					4																		1					6
อิ	1	1																																		1																3
อี																					2																															2
อึ																									2					1																						3
อื																												5																								5
อุ		1																																		1																2
อู		6																																																		6
โ																								1																7												8

ตารางรายละเอียดตัวอักษรที่ระบบรู้จำผิดและผลการรู้จำผิด(ต่อ)

ตัวอักษรที่ระบบรู้จำผิด	ผลการรู้จำผิด																																																		
	ก	ข	ค	จ	ฉ	ช	ซ	ฌ	ฎ	ฏ	ฑ	ณ	ด	ต	ถ	ท	ธ	น	ฝ	พ	ฟ	ม	ย	ร	ล	ว	ษ	ส	ฬ	ฮ	อะ	อี	อึ	อื	อุ	อู	เ	แ	โ	ใ	ไ	อำ	อั	ฤ	ฦ	อ้	อ๊	อ๋	อ์	ฯ	ๆ	รวม
ไ																										4																										4
อั																							2							1																11						14
ฤ	1																																												1					1	4	7
ฦ																																			1									2						1		4
อ่																																																1				1
อ้			3			1																																					4						1			9
อ๊																													1																							1
อ๋																											1				1			1					1			1										5
อ์																																1															1					2
ฯ											2																													1					4						2	9
รวม	2	9	3	1	1	1	6	4	5	12	7	7	8	3	2	3	1	5	1	4	11	4	3	1	3	5	2	7	8	8	2	1	4	1	1	4	1	4	1	8	3	1	5	2	7	11	7	3	1	2	7	213

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นางสาววรารัตน์ เอื้ออำนวยชัย เกิดเมื่อวันที่ 24 กุมภาพันธ์ 2514 สำเร็จการศึกษาวิทยาศาสตร์บัณฑิต สาขาสถิติประยุกต์ จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2537 เข้าศึกษาระดับปริญญามหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2545

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย